UCLA
UCLA Electronic Theses and Dissertations

Title
Distributed Adaptation over Networks with Applications to Biological Networks

Permalink
https://escholarship.org/uc/item/7vn753zt

Author
Tu, Sheng-Yuan

Publication Date
2013

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

University of California
Los Angeles

Distributed Adaptation over Networks with Applications to Biological Networks

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Electrical Engineering

by

Sheng-Yuan Tu

2013

© Copyright by
Sheng-Yuan Tu
2013

Abstract of the Dissertation

Distributed Adaptation over Networks with Applications to Biological Networks

by

Sheng-Yuan Tu

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2013

Professor Ali H. Sayed, Chair

Adaptive networks consist of a collection of nodes with adaptation and learning abilities. The nodes interact with each other on a local level and diffuse information across the network to solve estimation and inference tasks in real-time. In this dissertation, we first examine and compare the mean-square performance of two main strategies for distributed estimation over networks: consensus strategies and diffusion strategies. The analysis confirms that diffusion networks converge faster and reach lower mean-square deviation than consensus networks, and that their mean-square stability is insensitive to the choice of the combination weights. In contrast, and surprisingly, it is shown that consensus networks can become unstable even if all individual nodes are stable and able to solve the estimation task on their own. This finding motivates us to focus on the study of diffusion networks. We incorporate node mobility into the design of the networks and demonstrate that the resulting strategies are well suited to model various types of self-organized behavior observed in biological networks.

We also examine the effect of heterogeneous sources of information on network performance. In one scenario, we consider two types of agents: informed and

uninformed. Informed agents receive new data regularly and perform consultation and in-network processing tasks, while uninformed agents participate solely in the consultation tasks. It is established that if the set of informed agents is enlarged, the convergence rate of the network becomes faster albeit at the possible expense of some deterioration in mean-square performance. The arguments reveal an important interplay among three factors: the number and distribution of informed agents in the network, the convergence rate of the adaptation process, and the estimation accuracy in steady-state. In a second scenario, we study the situation in which the data observed by the agents may arise from two different distributions or models. We develop and study a procedure by which the entire network can be made to follow one objective or the other through a distributed and collaborative decision process. The results are useful to model situations where the agents in biological networks need to decide between multiple options, such as deciding between moving towards one food source or another or between moving towards a new hive or another.

The results in this dissertation reveal some interesting phenomena that relate to adaptation over networks: more information is not necessarily better and the way by which information is processed and propagated through the network matters: small variations can lead to catastrophic failures. The dissertation also reveals the convenience of using diffusion strategies to model sophisticated behavior exhibited by biological networks such as fish schooling and prey-predator behavior.

The dissertation of Sheng-Yuan Tu is approved.

Mario Gerla

Paulo Tabuada

Lieven Vandenberghe

Ali H. Sayed, Committee Chair

University of California, Los Angeles

2013

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my heartfelt gratitude to Professor Ali H. Sayed. I really thank thank him for his support and inspiration and for guiding me through my Ph.D. program. His careful reading and close reviews of my drafts have greatly enhanced the quality of my work. I was once told that a Ph.D. program is not only research, but more importantly, a process of mind training. Indeed, during the past four years, I am grateful to have Prof. Sayed help me go through these struggles.

The members of the Adaptive Systems Laboratory (ASL) have contributed immensely to my personal and professional time at UCLA. The group has been a source of friendships as well as good advice and collaboration. I am especially grateful for your company for the past four years: Zaid Towfic, Shang-Kee Ting, Jianshu Chen, and Xiaochuan Zhao. You are such nice and easygoing guys. I would also like to thank Chung-Kai Yu. Even though you have been in the group only for two years, we have known each other for seven years. I thank for your help in my personal affairs and for your useful coupons! I am also thankful for the opportunity to be a member of ASL where I have made many friends from the world: Paolo Di Lorenzo from Italy, Jingon Joung and Jae-Woo Lee from Korea, Oyvind L. Rortveit from Norway, Alexander Bertrand from Belgium, Victor Lora from France, Ricardo Merched and Cassio G. Lopes from Brazil, Reza Abdolee and Milad A. Toutounchian from Canada, Mohammad-Reza from Sweden, and Sergio Valcarcel Macua from Spain. I really enjoyed the useful discussions with all of you.

Lastly, I would like to thank my family for all their love and encouragement. For my parents who raised me with a love of science and supported me in all my pursuits. And most of all for my loving, supportive, encouraging, and patient

wife Yin-Yin whose faithful support during the final stages of this Ph.D. is so appreciated. Thank you.

Vita

1983	Born, Taichung, Taiwan.
2005	B.S. (Electrical Engineering), National Taiwan University (NTU), Taiwan.
2006–2007	Teaching Assistant, Department of Electrical Engineering, NTU.
2007	M.S. (Electrical Engineering), NTU.
2007–2008	Military Service, Army, Taiwan.
2008–2009	Research Assistant, Wireless Broadband Communication System Laboratory, NTU.
2008–2009	Teaching Assistant, Department of Electrical Engineering, NTU.
2009–present	Research Assistant, Department of Electrical Engineering, UCLA.

CHAPTER 1

Introduction

1.1 Insights from Biological Networks

Self-organization is a remarkable property in nature and has been observed in several physical and biological networks [25, 45]. Examples include fish joining together in schools, ants forming trails in foraging, and birds flying in formation. In such networks, a global pattern of behavior emerges from limited and localized interactions among the individual agents of the network. One interesting behavior is collective motion [25, 45, 48, 120, 135, 161], where animals move together in amazing synchrony such as fish schools swimming together in ball or band shaped patterns [111], birds flying in V-formation [4], or bees swarming towards a hive [70]. In fish schools, for example, the individual agents tend to have similar speeds and to move in alignment while keeping a safe distance from their neighbors to avoid collisions. Nevertheless, when predators appear, fish schools are able to react almost instantly; they reconfigure their topology to evade the predator and then regroup to continue with their schooling.

From a signal processing perspective, there are four properties exhibited by biological networks that are particularly relevant to the development of adaptive distributed strategies. First, in several of these networks, there is not generally any single agent taking central command and dictating the behavior of the other agents. Instead, the agents tend to be *homogeneous* in that they have

similar abilities. Second, due the lack of a central command unit, the agents in these networks tend to base their actions and decisions on observations of their neighbors' actions. That is, the interactions among the agents are *localized*. Interestingly, even though the individual agents tend to have limited capabilities, through localized collaboration, the network as a whole is able to exhibit sophisticated behavior. This phenomenon serves as one manifestation of the power of distributed processing. Third, agents in biological networks are constantly monitoring their environment to avoid attacks by predators. Therefore, the agents need to respond to the *continuous streaming* of data from their environment. Finally, the agents need to respond to the information they collect in *real-time*. These last two properties indicate that agents in biological networks are endowed with adaptation and learning abilities. One of the main objectives of this dissertation is to design and examine adaptive networks for distributed inference and to incorporate mobility into the operation of the agents in a manner that enables the resulting adaptive networks to model various forms of sophisticated behavior exhibited by biological networks.

There have been extensive prior studies on the collective motion of animal groups in the sciences and engineering literature (see, e.g., [25,45,66,67,120,159, 161] and [11,13,63,96]). For example, collective patterns of behavior have been exploited in the design of robotic systems and distributed control mechanisms with application to military and surveillance systems [24,56,68,100,106,107,118,171]. In these applications, it is generally assumed that the individual agents move along the average direction of their neighbors and use repulsion and attraction mechanisms to maintain safe distances from the neighbors. While such techniques help generate coordinated motion behavior, they nevertheless do not generally account for the adaptation and learning abilities of the agents. In nature, agents are rarely interested in only aligning their motion with their neighbors. The net-

work generally has more complex objectives, such as estimating the location of food sources and evading predators. Moreover, since agents have individualized assessment of their environment, it is unnatural to expect all agents to behave in coordination. For example, a fish that is closer to a predator should react differently from fish that are further away. For these reasons, it is important to incorporate learning and adaptation abilities into collective motion so that networks are able to accomplish their estimation and inference tasks while, at the same, they maintain the flexibility to adapt to the information they receive.

There are other issues that stand out in the behavior of biological networks, besides collective motion and the ability of their agents to adapt and learn in reaction to drifts in the environment. Agents also tend to have various degrees of information: some agents are more informed than others. Actually, in biological networks, the behavior of the network and its motion is often influenced heavily by a small fraction of informed agents. The experiment performed in Figure 1 of [136] provides a good example. In that experiment, when a few fish agents on the boundary of the perimeter are frightened, these agents rapidly change their direction of motion and reverse their orientation. The behavior propagates through the network very quickly. After a short period of time, the entire network ends up moving in the opposite direction relative to the original motion. Another example occurs in honeybees. It is observed in the home-site selection procedure that only 3-5% of bees [130] in the swarm have been to the new site (i.e., possess information). Interestingly, such small fraction of the swarm is still able to lead the entire swarm towards the new site. These observations reveal that although agents in a network can be homogeneous in their capabilities, they can nevertheless be heterogeneous in their access to information: some agents are more informed while others are less informed or uninformed.

Furthermore, the data observed by the agents in a network may arise from different distributions or models. It is common for biological networks to encounter situations where agents need to decide between multiple options [7, 12, 137, 138], such as fish deciding between moving towards one food source or another [46], and ants or bees deciding between moving towards a new hive or another [23, 113]. In this situation, data arriving at any particular agent could have originated from one model or the other, and the objective of the network becomes that of achieving agreement among the agents about which model to pursue as a group. Therefore, in addition to learning and adaptation abilities, agents should be endowed with decision-making abilities to resolve conflicts of interest situations. The decision-making process needs to be implemented in a fully distributed manner and in real-time as well.

1.2 Contributions

The studies in this dissertation are motivated by the aforementioned remarkable properties of biological networks. In particular, the contributions of this work belong to three broad areas. First, we examine and compare distributed adaptation strategies over networks and analyze which strategies are more resilient and stable in the face of changing topologies, and which strategies perform better. It is not uncommon in the literature to find the term "distributed" used to mean that the data are collected in a distributed manner by the agents and transferred to a fusion center for processing [36, 79, 79, 114, 162]. In this case, the qualification "distributed" is only meant to refer to the distributed nature of information gathering. It is clear though that such centralized solutions do not process data in a distributed manner; they are also prone to catastrophic failure: if the fusion center collapses, then the solution is interrupted. This

mode of operation is not characteristic of biological networks where processing is dispersed among the spatially distributed agents. There are three general approaches to enable distributed processing over networks: the incremental approach [18, 88, 93, 102, 115], the consensus approach [50, 73, 103, 144], and the diffusion approach [34,37,39,95,124,127,142,143]. In the incremental solution, a cyclic path is constructed over the agents and the data are processed in a cyclic manner through the network. However, determining a cyclic path that covers all agents is generally an NP-hard problem [75] and, in addition, cyclic trajectories need to be reconstructed in case of link or agent failures. In contrast, in the consensus and diffusion approaches, each agent shares information with its immediate neighbors and data processing is performed locally at the agents. Thus, even when any of the agents fails, the sharing of information will not be interrupted and the distributed solution continues to operate. Nevertheless, this dissertation reveals that there are some important differences in the dynamics of the consensus and diffusion networks. Specifically, consensus networks can become unstable even if all individual agents are stable and able to solve the estimation and inference tasks on their own. In contrast, diffusion networks are stable regardless of the network topology. They also deliver better mean-square performance at faster convergence rate than consensus networks. These results suggest that due to their inherent robustness to topology changes, diffusion networks are well suited to model sophisticated behavior in biological networks where topologies are continuously evolving with time.

The second broad area we consider is to examine the effect of heterogeneous sources of information on network performance. Specifically, we consider two types of agents: informed and uninformed. Informed agents receive new data regularly and perform consultation and in-network processing tasks, while uninformed agents participate solely in the consultation tasks. We establish that if

the set of informed agents is enlarged, the convergence rate of the network becomes faster albeit at the possible expense of some deterioration in mean-square performance. This implies that more information is not necessarily better. In addition, there exists an important interplay among three factors: the number and distribution of informed agents in the network, the convergence rate of the adaptation process, and the estimation accuracy in steady-state.

The third broad problem we study relates to adaptation over networks when agents have distinct objectives. We consider the situation in which the data observed by the agents may arise from two different distributions or models, but the objective of the network is still to agree and to converge to one of the models. We develop a procedure by which the entire network can be made to converge to a common model through a distributed and collaborative decision process. We also examine the rate of information transfer over the network. We show that the interactions among the agents in the network need to be information-aware. It has often been suggested in the literature that the patterns of collective motion observed in nature can be modeled by having each agent move along the average direction of its neighbors. However, recent experiments on the behavioral rules of fish schools appear to challenge this traditional uniform (or averaging) combination rule [76]. We establish that the combination weights should be information-aware: agents should assign larger weights to more informed agents. We explain how this can be achieved. By doing so, quicker transfer of information is attained through the network.

1.3 Organization

The organization of the dissertation is summarized as follows.

- **Chapter 2**: We motivate and derive two classes of distributed adaptation strategies: consensus and diffusion. It turns out that both families can be derived as special cases of the same formalism by employing steepest-descent and incremental techniques. We then carry out a detailed mean-square performance analysis of the distributed solutions in a unified manner in preparation for the following chapters.

- **Chapter 3**: In this chapter, we compare the mean-square performance of consensus and diffusion strategies. The analysis confirms that diffusion strategies allow information to diffuse more thoroughly through the network and this property has a favorable effect on the evolution of the network: diffusion networks are shown to converge faster and reach lower mean-square deviation (MSD) than consensus networks, and their mean-square stability is insensitive to the choice of the combination weights. In contrast, and surprisingly, it is shown that consensus networks can become unstable even if all the individual agents are stable and able to solve the estimation task on their own. When this occurs, cooperation over consensus network leads to a catastrophic failure.

- **Chapter 4**: One important issue in the design of adaptive networks is how to combine the information collected from the neighbors, especially when the links over which information is exchanged are subject to noise and interference. In this chapter, we propose an optimal strategy for selecting the combination weights for the diffusion implementation in order to counter the effect of noise sources from measurement and information exchanges. We also develop an effective strategy to adapt the combination weights over time, which is particularly useful when the data moments are not available and when networks are operating under non-stationary conditions. For

example, in mobile networks where agents are continuously on the move and where neighborhoods evolve over time, it is particularly critical to develop adaptive combination strategies that are able to track the dynamics of the noise profile and to perform estimation and inference successfully under such demanding and varying conditions.

- **Chapter 5**: We examine the effect of heterogeneous sources of information on network performance in this chapter. We consider two types of agents: informed agents and uninformed agents. The former receive new data regularly and perform consultation and in-network tasks, while the latter do not collect data and only participate in the consultation tasks. We examine the performance of diffusion strategies as a function of the proportion of informed agents and their distribution in space. The results reveal some interesting and surprising trade-offs between convergence rate and mean-square deviation. In particular, among other results, it is shown that the mean-square deviation of diffusion networks does not necessarily improve with a larger proportion of informed agents. Instead, it is established that if the set of informed agents is enlarged, the convergence rate of the network becomes faster albeit at the expense of some deterioration in mean-square performance. The results further establish that uninformed agents play an important role in determining the steady-state performance of the network, and that it is preferable to keep some of the highly noisy or highly connected agents uninformed.

- **Chapter 6**: In distributed processing, agents generally collect data generated by the *same* underlying unknown distribution and then solve the desired estimation and inference tasks cooperatively. In this chapter, we consider the situation in which the data observed by the agents may arise

from *different* distributions or models. Agents do not know beforehand which model accounts for their data and the objective for the network becomes that of guiding all agents towards the *same* common goal. In these situations, where agents are subject to data from unknown different sources, conventional distributed estimation strategies would lead to biased solutions. We first show how to modify existing strategies to guarantee unbiasedness. We then develop a classification scheme for the agents to identify the models that generated the data, and propose a procedure by which the entire network can be made to converge towards the same model through a collaborative decision-making process. The probability of error in the classification scheme is evaluated. We also examine the convergence rates of the estimation and decision-making processes and characterize the speed at which information diffuses through the network.

- **Chapter 7**: We add another dimension of complexity in this chapter. We incorporate mobility into the design of adaptive networks and study what we refer to as *mobile adaptive networks*. We develop control mechanisms that enable the agents to move in a coordinated manner and to simultaneously solve the estimation or inference tasks of interest. The motion of the agents is influenced by the quality of the adaptation process and vice versa, such that the two issues of adaptation and mobility become intertwined. Moreover, by combining motion coordination with adaptation in real-time, this chapter extends previous studies on the motion of coordinated agents, which generally assume that the individual agents move along the average direction of their neighbors and maintain safe distances from the neighbors. We apply the mobile adaptive networks to model sophisticated behavior observed in biological networks, such as mobile networks in the presence of

moving predators, mobile networks with two distinct targets, and mobile networks with a small fraction of informed agents. The results help provide an explanation for the agile adjustment of network patterns in the interaction between fish schools and predators, for decision-making processes among animal groups, and for the quick information transfer from informed agents to the entire network.

CHAPTER 2

Distributed Estimation Strategies

In this chapter, we motivate and derive two well-studied distributed strategies, namely, consensus and diffusion strategies, within a unified framework. The consensus strategy has been originally proposed in the statistics literature [49] and has since then been developed into an elegant procedure to enforce agreement among cooperating agents. Average consensus and gossip algorithms have also been studied extensively in recent years, especially in the control literature [8, 22, 69, 106, 109, 122, 166], and have been applied to the study of multi-agent formations [56, 68, 107, 118], distributed optimization [103, 104, 144], and distributed estimation problems [50, 73, 128, 129]. Original implementations of the consensus strategy relied on the use of two time-scales [11, 71, 168]: one time-scale for the collection of measurements across the agents and another time-scale to iterate sufficiently enough over the collected data to attain agreement before the process is repeated. Unfortunately, two time-scale implementations hinder the ability to perform real-time recursive estimation and adaptation when measurement data keep streaming in. For this reason, in this work, we focus instead on consensus implementations that operate in a single time-scale. Such implementations appear in several recent works, including [50, 73, 104, 129], and are largely motivated by the procedure developed earlier in [19, 144] for the solution of distributed optimization problems.

The second class of algorithms that we consider deals with diffusion strategies,

which were originally introduced for the solution of distributed estimation and adaptation problems in [28,31,32,34,35,92,95,125,126]. The main motivation for the introduction of diffusion strategies in these works has been the need to develop distributed schemes that are able to respond in real-time to the *continuous streaming* of data at the agents by operating over a *single* time-scale. A useful overview of diffusion strategies appears in [124]. Since their inception, diffusion strategies have been applied to model various forms of complex behavior encountered in nature [29,40,86,145,147,149,150]; they have also been adopted to solve distributed optimization problems advantageously in [37,39,116,133]; and have been studied under varied conditions in [1,41,42,87,98,140] as well. Diffusion strategies are inherently single time-scale implementations and are therefore naturally amenable to real-time and recursive implementations. It turns out that the dynamics of the consensus and diffusion strategies differ in important ways, which in turn impact the mean-square behavior of the respective networks in a fundamental manner, as we will reveal in the next chapter.

2.1 Cooperative and Non-Cooperative Strategies

Consider a network consisting of N agents distributed over a spatial domain. Agent l is said to be a neighbor of agent k if agent k can receive information from agent l. The fact that agent l is a neighbor of agent k does not necessarily imply that agent k is a neighbor of agent l. In other work, the network is assumed to be *directed*. The neighborhood of agent k is denoted by $\mathcal{N}_k$ and the number of neighbors (or the degree) of agent k is denoted by n_k. The agents in the network would like to estimate an unknown $M \times 1$ vector, w°. At every time instant, i, each agent k is able to observe realizations $\{d_k(i), u_{k,i}\}$ of a scalar random process $\boldsymbol{d}_k(i)$ and a $1 \times M$ vector random process $\boldsymbol{u}_{k,i}$ with a positive-definite covariance

matrix, $R_{u,k} = \mathbb{E}\boldsymbol{u}_{k,i}^{*}\boldsymbol{u}_{k,i} > 0$, where $\mathbb{E}$ denotes the expectation operator. The regressors $\{\boldsymbol{u}_{k,i}\}$ are assumed to be temporally white and spatially independent, i.e.,

$$\mathbb{E}\boldsymbol{u}_{k,i}^{T}\boldsymbol{u}_{l,j} = R_{u,k} \cdot \delta_{kl} \cdot \delta_{ij} \tag{2.1}$$

in terms of the Kronecker delta function, i.e.,

$$\delta_{kl} = \begin{cases} 1, \text{if } k = l \\ 0, \text{if } k \neq l \end{cases} \tag{2.2}$$

All vectors in our treatment are column vectors with the exception of the regression vector, $\boldsymbol{u}_{k,i}$, which is taken to be a row vector for convenience of presentation. It is common to assume that the random processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ are related to w° via the linear regression model [123]:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}w^{\circ} + \boldsymbol{v}_k(i) \tag{2.3}$$

where $\boldsymbol{v}_k(i)$ is measurement noise with zero mean and variance $\sigma_{v,k}^2$ and assumed to be temporally white and spatially independent. The noise $\boldsymbol{v}_k(i)$ and the regressors $\{\boldsymbol{u}_{l,j}\}$ are assumed to be independent of each other for all $\{k, l, i, j\}$. Note that we use boldface letters to denote random quantities and normal letters to denote their realizations or deterministic quantities. Models of the form (2.3) are useful in capturing many situations of interest, such as estimating the parameters of some underlying physical phenomenon, estimating or equalizing a communications channel, tracking a moving target by a collection of agents, or estimating the location of a nutrient source or predator in biological networks (see, e.g., [5, 6, 26, 29, 80, 81, 101, 123, 150, 164, 165]). For example, in biological networks, the variable $\boldsymbol{d}$ can model the noisy distance to a target and the variable $\boldsymbol{u}$ can model the noisy direction towards the target [150].

The objective of the network is to estimate w° in a distributed manner through an online learning process. The agents estimate w° by seeking to minimize the following global cost function:

$$J^{\text{glob}}(w) \triangleq \sum_{k=1}^{N} \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2. \tag{2.4}$$

A completion-of-squares argument, followed by a stochastic approximation step and an incremental approximation step, were used in [34,37] to derive a *diffusion* algorithm for the optimization of (2.4) in a decentralized manner; the algorithm is reviewed below in (2.29) and (2.32) in two of its prominent forms. It turns out that we can derive and motivate the aforementioned single time-scale consensus strategy in a similar manner, which will lead us to recursion (2.24) given further ahead. The presentation in this chapter is therefore meant to show how diffusion and consensus strategies can be derived as special cases of the same formalism. In the next chapter, we will move on to show that diffusion networks outperform consensus networks.

2.1.1 Non-Cooperative Strategy

We consider first the case in which each agent k operates individually and attempts to determine w° by minimizing its individual cost function, which is denoted by

$$J_k(w) \triangleq \mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2. \tag{2.5}$$

The traditional steepest-descent solution for determining the solution of (2.5) takes the following form [62, 123]:

$$w_{k,i} = w_{k,i-1} - \mu_k \cdot [\nabla_w J_k(w_{k,i-1})]^* \tag{2.6}$$

where $\mu_k > 0$ is the step-size used by agent k, and $\nabla_w J_k(\cdot)$ denotes the (row) gradient vector of $J_k(w)$ with respect to the variable w. Moreover, $w_{k,i}$ is the

estimate of w° by agent k at time i. Using (2.5), the above recursion becomes

$$w_{k,i} = w_{k,i-1} + \mu_k \cdot (r_{du,k} - R_{u,k} w_{k,i-1}) \tag{2.7}$$

where $r_{du,k} = \mathbb{E}\boldsymbol{d}_k(i)\boldsymbol{u}^*_{k,i}$. An adaptive implementation can be obtained by replacing $\{r_{du,k}, R_{u,k}\}$ by instantaneous approximations:

$$r_{du,k} \approx d_k(i)u^*_{k,i}, \quad R_{u,k} \approx u^*_{k,i}u_{k,i} \tag{2.8}$$

thus leading to the well-known LMS recursion [123]:

$$\text{(non-cooperative strategy)} \quad w_{k,i} = w_{k,i-1} + \mu_k \cdot u^*_{k,i}[d_k(i) - u_{k,i}w_{k,i-1}] \tag{2.9}$$

Using (2.9), each agent k can update its estimate of w° over time using its local data $\{d_k(i), u_{k,i}\}$. Note that for the underlying model where $R_{u,k} > 0$ for all k, every individual agent can employ (2.9) to estimate w° independently if desired. Studies allowing for other observability conditions for diffusion and consensus strategies, including possibly singular covariance matrices, appear in [1, 73].

2.1.2 Cooperative Strategies

We now motivate cooperative strategies where agents are allowed to interact with their neighbors. By doing so, it can be shown that performance (in terms of convergence rate and mean-square-error) over the network will improve relative to the non-cooperative mode of operation [34,154,155]. To begin with, we express the global cost (2.4) as:

$$J^{\text{glob}}(w) = \sum_{k=1}^{N} J_k(w) \tag{2.10}$$

where $J_k(w)$ is given by (2.5). Our derivation of cooperative strategies to optimize $J^{\text{glob}}(w)$ in a distributed manner is based on two steps by extending the development of [34, 37] to cover both consensus and diffusion-type methods. First, using

a completion-of-squares argument, we approximate the global cost function (2.10) by an alternative cost that is amenable to distributed optimization. Then, each agent will optimize the alternative cost via a combination of a steepest-descent step and an incremental approximation step.

Thus, note that each individual cost $J_k(w)$ given by (2.5) can be factored via a completion-of-squares argument and written in the form:

$$J_k(w) = \|w - w^\circ\|^2_{R_{u,k}} + \text{mmse}_k \tag{2.11}$$

where the notation $\|x\|^2_\Sigma$ denotes the weighted square quantity $x^*\Sigma x$ for any nonnegative-definite matrix $\Sigma \geq 0$, and mmse_k is independent of w. Using (2.11), we can replace the cost function (2.10) by the equivalent global cost:

$$J^{\text{glob}'}(w) \triangleq J_k(w) + \sum_{l \neq k} \|w - w^\circ\|^2_{R_{u,l}} \tag{2.12}$$

where we ignored the terms $\{\text{mmse}_l\}$ that do not depend on w. The second term on the right-hand side of (2.12), which corresponds to a sum of quadratic factors involving the minimizer w°, tells us how the individual cost $J_k(w)$ can be corrected to the global cost $J^{\text{glob}'}(w)$. Obviously, the minimizer w° that appears in the correction term is not known since the agents wish to determine its value. Likewise, not all weighting matrices $R_{u,l}$ are available to agent k; only those matrices that originate from its neighbors can be assumed to be available. Still, expression (2.12) motivates us to introduce a new localized cost function at agent k that is closer to the desired $J^{\text{glob}'}(w)$ and which can be minimized through local cooperation. We denote this localized cost at agent k by $J_k^{\text{glob}'}(w)$ and it is obtained from (2.12) by limiting the summation on the right-hand side of (2.12) to the neighbors of agent k, namely,

$$J_k^{\text{glob}'}(w) \triangleq J_k(w) + \sum_{l \in \mathcal{N}_k \backslash \{k\}} \|w - w^\circ\|^2_{R_{u,l}}. \tag{2.13}$$

The cost functions $J_k(w)$ and $J_k^{\text{glob}'}(w)$ are both associated with agent k; the difference between them is that the expression for the latter is closer to the global cost function (2.12) that we want to optimize.

The second-order moments $\{R_{u,l}\}$ that appear in (2.13) may or may not available. If they are known, then we can proceed with the analysis by assuming knowledge of the $\{R_{u,l}\}$. However, the more interesting case is when these moments are not known. This is generally the case in practice, especially in the context of recursive estimation and tracking problems. Usually, agents can only observe realizations $\{u_{l,i}\}$ of regression data arising from distributions whose covariance matrices are the unknown $\{R_{u,l}\}$. One way to address this difficulty is to replace each of the weighted norms $\|w - w^\circ\|^2_{R_{u,l}}$ by a scaled multiple of the form

$$\|w - w^\circ\|^2_{R_{u,l}} \approx b_{l,k}\|w - w^\circ\|^2 \tag{2.14}$$

where $b_{l,k}$ is some nonnegative coefficient; we are even allowing the value of this coefficient to change with the agent index k. The above substitution amounts to having each agent k approximate the $\{R_{u,l}\}$ from its neighbors by multiples of the identity matrix:

$$R_{u,l} \approx b_{l,k} I_M. \tag{2.15}$$

Approximation (2.14) is reasonable in view of the fact that all vector norms on finite dimensional spaces are equivalent [64]; this norm property ensures that we can always bound the weighted norm $\|w-w^\circ\|^2_{R_{u,l}}$ by some constants multiplying the unweighted norm $\|w - w^\circ\|^2$, say,

$$r_1\|w - w^\circ\|^2 \leq \|w - w^\circ\|^2_{R_{u,l}} \leq r_2\|w - w^\circ\|^2 \tag{2.16}$$

for some positive constants $\{r_1, r_2\}$. Actually, in view of the Rayleigh-Ritz char-

acterization of eigenvalues [59, 64], we have

$$\lambda_{\min}(R_{u,l}) \cdot \|w - w^{\circ}\|^2 \leq \|w - w^{\circ}\|^2_{R_{u,l}} \leq \lambda_{\max}(R_{u,l}) \cdot \|w - w^{\circ}\|^2 \tag{2.17}$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimum and the maximum eigenvalues of its Hermitian matrix argument, respectively. As the derivation will show, we do not need to worry at this stage about how the scalars $\{b_{l,k}\}$ in (2.14) are selected; they will end up being embedded into another set of coefficients $\{a_{l,k}\}$ that will be selected by the designer — see (2.23) further ahead. We will also show later (see Section 4.4 further ahead) that how to select these coefficients to obtain our desired objective. In this way, we replace (2.13) by:

$$J_k^{\text{glob}''}(w) \triangleq J_k(w) + \sum_{l \in \mathcal{N}_k \setminus \{k\}} b_{l,k} \|w - w^{\circ}\|^2 \tag{2.18}$$

With the exception of the variable w°, this alternative cost at agent k relies solely on information that is available to agent k from its neighborhood. We will soon explain how to handle the fact that w° is not known to agent k.

Now, each agent k can apply a steepest-descent iteration to minimize its localized cost $J_k^{\text{glob}''}(w)$. The iteration would be of the following form:

$$\begin{aligned} w_{k,i} &= w_{k,i-1} - \mu_k \cdot \left[\nabla_w J_k^{\text{glob}''}(w_{k,i-1})\right]^* \\ &= w_{k,i-1} + \mu_k \cdot (r_{du,k} - R_{u,k} w_{k,i-1}) - \mu_k \cdot \sum_{l \in \mathcal{N}_k \setminus \{k\}} b_{l,k}(w_{k,i-1} - w^{\circ}) \end{aligned} \tag{2.19}$$

The step-size parameters $\{\mu_k\}$ can be constant or time-variant. In this work, we consider the case of constant step-sizes because we are particularly interested in the adaptation and learning abilities of the distributed strategies. Constant step-sizes allow the resulting strategies to learn and adapt continuously, while time-variant step-sizes that decay to zero turn off the learning abilities of the networks with time.

Now, as was the case with the non-cooperative steepest-descent (2.7) and LMS strategies (2.9), an adaptive implementation of (2.19) can be obtained by using the same instantaneous approximations (2.8). Doing so leads to the following recursion:

$$w_{k,i} = w_{k,i-1} + \mu_k \cdot u_{k,i}^*[d_k(i) - u_{k,i}w_{k,i-1}] - \mu_k \cdot \sum_{l \in \mathcal{N}_k \backslash \{k\}} b_{l,k}(w_{k,i-1} - w^\circ) \quad (2.20)$$

Compared with the non-cooperative LMS strategy (2.9), recursion (2.20) indicates that the update from $w_{k,i-1}$ to $w_{k,i}$ now involves adding *two* correction terms to $w_{k,i-1}$. However, the last correction term still depends on the unknown w°. We can now use incremental-type arguments to replace w° in (2.20) by suitable approximations for it. It turns out that different replacements for w° lead to different learning strategies (such as consensus and diffusion strategies) and these replacements will affect the operation of the network in a fundamental way.

2.1.2.1 Consensus Strategy

The first substitution for w° that we consider will lead to the single time-scale consensus strategy that we alluded to before; see (2.24) below. Specifically, note that each of the agents in the network is performing steps similar to (2.20). As such, each agent l will have a readily available approximation for w°, which is its local estimate $w_{l,i-1}$. Therefore, one substitution for w° in (2.20) is to replace it by $w_{l,i-1}$. In that case, recursion (2.20) becomes

$$w_{k,i} = w_{k,i-1} - \mu_k \cdot \sum_{l \in \mathcal{N}_k \backslash \{k\}} b_{l,k}(w_{k,i-1} - w_{l,i-1}) + \mu_k \cdot u_{k,i}^*[d_k(i) - u_{k,i}w_{k,i-1}]. \quad (2.21)$$

Recursion (2.21) is in the form of the well-known consensus strategy (see, e.g., expression (19) in [50] and expression (9) in [73]). It should be noted that in most other works on consensus implementations, especially in the context of

distributed optimization problems [19, 50, 73, 104, 116], the step-sizes $\{\mu_k\}$ that are used in (2.21) depend on the time-index i and are required to satisfy

$$\sum_{i=0}^{\infty} \mu_k(i) = \infty \quad \text{and} \quad \sum_{i=0}^{\infty} \mu_k^2(i) < \infty \tag{2.22}$$

In other words, for each agent k, the step-size sequence $\mu_k(i)$ is required to vanish as $i \to \infty$. Under such conditions, it is known that consensus strategies allow the agents to reach agreement. Here, instead, in the representations (2.21), we are using *constant* step-sizes because we are interested in studying the adaptation and learning abilities of the networks. Constant step-sizes are critical to endow networks with *continuous* adaptation and tracking abilities; otherwise, under (2.22), once the step-size has decayed to zero, the network stops adapting.

We can rewrite the recursion in a more compact and revealing form by combining the first two terms on the right-hand side of (2.21) and by introducing the following coefficients:

$$a_{l,k} = \begin{cases} 1 - \displaystyle\sum_{j \in \mathcal{N}_k \setminus \{k\}} \mu_k \cdot b_{j,k}, & \text{if } l = k \\ \mu_k \cdot b_{l,k}, & \text{if } l \in \mathcal{N}_k \setminus \{k\} \\ 0, & \text{otherwise} \end{cases} \tag{2.23}$$

In this way, recursion (2.21) can be rewritten equivalently as (see, e.g., expression (7.10) in [19] and expression (1.20) in [104]):

$$\text{(consensus strategy)} \quad w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{l,k} w_{l,i-1} + \mu_k \cdot u_{k,i}^* [d_k(i) - u_{k,i} w_{k,i-1}] \tag{2.24}$$

The entry $a_{l,k}$ denotes the weight that agent k assigns to the estimate $w_{l,i-1}$ received from its neighbor l (see Fig. 2.1); note that the weights $\{a_{l,k}\}$ are nonnegative for $l \neq k$ and that $a_{k,k}$ is nonnegative for sufficiently small step-sizes. If we collect the nonnegative weights $\{a_{l,k}\}$ into an $N \times N$ matrix A,

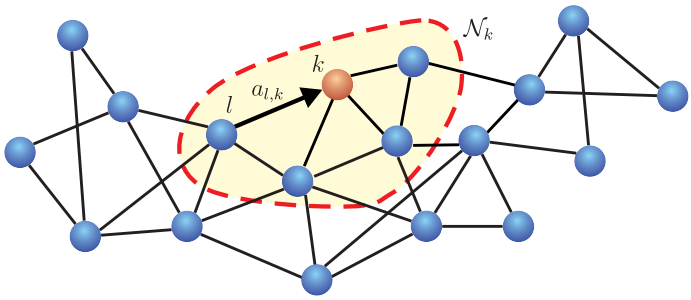


Figure 2.1: A connected network showing the neighborhood of agent k. The weight $a_{l,k}$ scales the data transmitted from agent l to agent k over the edge linking them.

then it follows from (2.23) that the combination matrix A satisfies the following properties:

$$a_{l,k} \geq 0,\ A^T \mathbb{1}_N = \mathbb{1}_N, \text{ and } a_{l,k} = 0 \text{ if } l \notin \mathcal{N}_k \tag{2.25}$$

where $\mathbb{1}_N$ is a vector of size N with all entries equal to one. That is, the weights on the links arriving at agent k add up to one, which is equivalent to saying that the matrix A is left-stochastic. Moreover, if agent l is not a neighbor of agent k, then the corresponding weight $a_{l,k}$ is zero. We summarize useful properties of the combination matrix in Appendix 2.A.

2.1.2.2 ATC Diffusion Strategy

The second substitution we consider for w° in (2.20) will lead to two forms of diffusion strategies. Thus, note first that there are two correction terms on the right-hand side of (2.20). These terms could be added one at a time. For example, we can achieve (2.20) by splitting the update into the following two steps involving

an intermediate estimate $\psi_{k,i}$:

$$\psi_{k,i} = w_{k,i-1} + \mu_k \cdot u_{k,i}^*[d_k(i) - u_{k,i}w_{k,i-1}] \tag{2.26}$$

$$w_{k,i} = \psi_{k,i} - \mu_k \cdot \sum_{l \in \mathcal{N}_k \backslash \{k\}} b_{l,k}(w_{k,i-1} - w^\circ) \tag{2.27}$$

Note that the first update (2.26) can be carried out by all agents independent of knowledge of w°. However, the unknown w° still appears in (2.27). Now, rather than replace w° by $w_{l,i-1}$, as was the case with the consensus strategy, it appears to be more advantageous to replace w° by the "improved" estimate $\psi_{l,i}$ obtained via the update (2.26); and the analysis will confirm that this is indeed the case. For each agent l, the intermediate value $\psi_{l,i}$ is generally a better estimate for w° than $w_{l,i-1}$ since it is obtained by incorporating information from its recent data $\{d_l(i), u_{l,i}\}$ in (2.26). In the same spirit, we also replace $w_{k,i-1}$ in (2.27) by $\psi_{k,i}$. This second substitution is reminiscent of incremental-type approaches to optimization, which have been widely studied in the literature [18, 102]. With these replacements, recursion (2.27) becomes

$$w_{k,i} = \psi_{k,i} - \mu_k \cdot \sum_{l \in \mathcal{N}_k \backslash \{k\}} b_{l,k}(\psi_{k,i} - \psi_{l,i}) \tag{2.28}$$

If we again introduce the same coefficients $\{a_{l,k}\}$ from (2.23), we arrive at the following alternative compact form, known as the adapt-then-combine (ATC) diffusion strategy [34]:

$$\text{(ATC diffusion strategy)} \quad \begin{aligned} \psi_{k,i} &= w_{k,i-1} + \mu_k \cdot u_{k,i}^*[d_k(i) - u_{k,i}w_{k,i-1}] \\ w_{k,i} &= \sum_{l \in \mathcal{N}_k} a_{l,k}\psi_{l,i} \end{aligned} \tag{2.29}$$

The ATC diffusion strategy consists of two steps. The first step of (2.29) involves local adaptation, where agent k uses its own data $\{d_k(i), u_{k,i}\}$ to update its weight estimate from $w_{k,i-1}$ to an intermediate value $\psi_{k,i}$. The second step of (2.29) is

a consultation (combination) step where the intermediate estimates $\{\psi_{l,i}\}$ from the neighborhood of agent k are combined through the weights $\{a_{l,k}\}$ to obtain the updated weight estimate $w_{k,i}$.

2.1.2.3 CTA Diffusion Strategy

Other variants of the diffusion strategies are possible. For example, if we reverse the order of (2.26)-(2.27), recursion (2.20) can also be achieved as follows:

$$\psi_{k,i-1} = w_{k,i-1} - \mu_k \cdot \sum_{l \in \mathcal{N}_k \backslash \{k\}} b_{l,k}(w_{k,i-1} - w^\circ) \tag{2.30}$$

$$w_{k,i} = \psi_{k,i-1} + \mu_k \cdot u_{k,i}^*[d_k(i) - u_{k,i} w_{k,i-1}] \tag{2.31}$$

We now replace w° in (2.30) by $w_{l,i-1}$ and replace $w_{k,i-1}$ in (2.31) by $\psi_{k,i-1}$. Furthermore, using the same coefficients $\{a_{l,k}\}$ from (2.23), we arrive at the following combine-then-adapt (CTA) diffusion strategy [34, 95]:

$$\text{(CTA diffusion strategy)} \qquad \begin{aligned} \psi_{k,i-1} &= \sum_{l \in \mathcal{N}_k} a_{l,k} w_{l,i-1} \\ w_{k,i} &= \psi_{k,i-1} + \mu_k \cdot u_{k,i}^*[d_k(i) - u_{k,i} \psi_{k,i-1}] \end{aligned} \tag{2.32}$$

The CTA diffusion strategy consists of two steps. The first step of (2.32) involves a consultation (combination) step, where the existing estimates $\{w_{l,i-1}\}$ from the neighbors of agent k are combined through the weights $\{a_{l,k}\}$. The second step of (2.32) is a local adaptation step, where agent k uses its own data $\{d_k(i), u_{k,i}\}$ to update its weight estimate from the intermediate value $\psi_{k,i-1}$ to $w_{k,i}$. Thus, comparing the ATC and CTA strategies, we note that the order of the consultation and adaptation steps are reversed.

2.1.2.4 Comparing Diffusion and Consensus Strategies

For ease of reference, we rewrite below the recursions that correspond to the consensus (2.24), ATC diffusion (2.29), and CTA diffusion (2.32) strategies in a single update:

$$\text{(consensus)} \quad w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{l,k} w_{l,i-1} + \mu_k \cdot u_{k,i}^* [d_k(i) - u_{k,i} w_{k,i-1}] \tag{2.33}$$

$$\text{(ATC diffusion)} \quad w_{k,i} = \sum_{l \in \mathcal{N}_k} \left\{ a_{l,k} w_{l,i-1} + \mu_l \cdot a_{l,k} u_{l,i}^* [d_l(i) - u_{l,i} w_{l,i-1}] \right\} \tag{2.34}$$

$$\text{(CTA diffusion)} \quad w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{l,k} w_{l,i-1} + \mu_k \cdot u_{k,i}^* \left[d_k(i) - u_{k,i} \left(\sum_{l \in \mathcal{N}_k} a_{l,k} w_{l,i-1} \right) \right] \tag{2.35}$$

Note that the first terms on the right hand side of these recursions are all the same. For the second terms, only variable $w_{k,i-1}$ appears in the consensus strategy (2.33), while the diffusion strategies (2.34)-(2.35) incorporate the estimates $\{w_{l,i-1}\}$ from the neighborhood of agent k into the update of $w_{k,i}$. Moreover, in contrast to the consensus (2.33) and CTA diffusion (2.35) strategies, the ATC diffusion strategy (2.34) further incorporates the influence of the data $\{d_l(i), u_{l,i}\}$ from the neighborhood of agent k into the update of $w_{k,i}$. These facts have important implications on the evolution of the weight-error vectors across consensus and diffusion networks. It is important to note that the diffusion strategies (2.34)-(2.35) are able to incorporate additional information into their processing steps *without* being more complex than the consensus strategy. All three strategies have the *same* computational complexity and require sharing the *same* amount of data (see Table 2.1), as can be ascertained by comparing the actual implementations (2.24), (2.29), and (2.32). The key fact to note is that the diffusion implementations first generate an intermediate state, which is subsequently used in the final update. This important ordering of the calculations has a critical

Table 2.1: Comparison of the number of complex multiplications and additions per iteration, as well as the number of vectors of size $M \times 1$ that are exchanged for each iteration of the algorithms at every agent k. Observe that all three implementations have *exactly* the same computational complexity.

	ATC diffusion (2.29)	**CTA diffusion** (2.32)	**Consensus** (2.24)
Multiplications	$(n_k+2)M$	$(n_k+2)M$	$(n_k+2)M$
Additions	$(n_k+1)M$	$(n_k+1)M$	$(n_k+1)M$
Vector exchanges	n_k	n_k	n_k

influence on the performance of the algorithms, as we will reveal in the next chapter.

2.2 Mean-Square Performance Analysis

The mean-square performance of diffusion networks has been studied in detail in [34]. Expressions for the network performance, and conditions for their mean-square stability, were derived there by applying energy conservation arguments [2,123]. Following [34], we will first show how to carry out the performance analysis in a unified manner that can cover both diffusion and consensus strategies (see Table 2.2 further ahead, which highlights how the parameters for both strategies differ). Subsequently, we use the resulting performance expressions to carry out detailed comparisons and to establish and highlight some surprising and interesting differences in performance.

2.2.1 Network Error Recursion

Let the error vector for an arbitrary agent k be denoted by:

$$\tilde{\boldsymbol{w}}_{k,i} \triangleq w^{\circ} - \boldsymbol{w}_{k,i} \tag{2.36}$$

We collect all error vectors and step-sizes across the network into a block vector and block matrix:

$$\tilde{\boldsymbol{w}}_i \triangleq \operatorname{col}\{\tilde{\boldsymbol{w}}_{1,i}, \tilde{\boldsymbol{w}}_{2,i}, \cdots, \tilde{\boldsymbol{w}}_{N,i}\} \tag{2.37}$$

$$\mathcal{M} \triangleq \operatorname{diag}\{\mu_1 I_M, \mu_2 I_M, \cdots, \mu_N I_M\} \tag{2.38}$$

where the notation $\operatorname{col}\{\cdot\}$ denotes the vector that is obtained by stacking its arguments on top of each other, and the notation $\operatorname{diag}\{\cdot\}$ constructs a diagonal matrix from its arguments. We further introduce the extended combination matrix:

$$\mathcal{A} \triangleq A \otimes I_M \tag{2.39}$$

where the symbol $\otimes$ denotes the Kronecker product of two matrices [59,84]. This construction replaces each entry $a_{l,k}$ in A by the block $M \times M$ matrix $a_{l,k} I_M$ in $\mathcal{A}$. Then, if we start from (2.33) and use model (2.3), some straightforward algebra in [34,124] shows that the global error vector $\tilde{\boldsymbol{w}}_i$ for the consensus strategy evolves according to the following recursion:

$$\text{(consensus)} \quad \tilde{\boldsymbol{w}}_i = (\mathcal{A}^T - \mathcal{M}\boldsymbol{\mathcal{R}}_i) \cdot \tilde{\boldsymbol{w}}_{i-1} - \mathcal{M}\boldsymbol{s}_i \tag{2.40}$$

where $\boldsymbol{\mathcal{R}}_i$ is a block diagonal matrix and $\boldsymbol{s}_i$ is a block column vector:

$$\boldsymbol{\mathcal{R}}_i \triangleq \operatorname{diag}\{\boldsymbol{u}_{1,i}^* \boldsymbol{u}_{1,i}, \boldsymbol{u}_{2,i}^* \boldsymbol{u}_{2,i}, \cdots, \boldsymbol{u}_{N,i}^* \boldsymbol{u}_{N,i}\} \tag{2.41}$$

$$\boldsymbol{s}_i \triangleq \operatorname{col}\{\boldsymbol{u}_{1,i}^* \boldsymbol{v}_{1,i}, \boldsymbol{u}_{2,i}^* \boldsymbol{v}_{2,i}, \cdots, \boldsymbol{u}_{N,i}^* \boldsymbol{v}_{N,i}\} \tag{2.42}$$

Likewise, starting from (2.34) or (2.35) and using model (2.3), the global error vector $\tilde{\boldsymbol{w}}_i$ for the diffusion strategies can be found to evolve according to the

recursions:

$$\text{(ATC diffusion)} \quad \tilde{\boldsymbol{w}}_i = \mathcal{A}^T(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i) \cdot \tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}^T\mathcal{M}\boldsymbol{s}_i \tag{2.43}$$

$$\text{(CTA diffusion)} \quad \tilde{\boldsymbol{w}}_i = (I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}^T \cdot \tilde{\boldsymbol{w}}_{i-1} - \mathcal{M}\boldsymbol{s}_i \tag{2.44}$$

Recursions (2.40), (2.43), and (2.44) can be seen to be special cases of a general recursion of the form:

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \cdot \tilde{\boldsymbol{w}}_{i-1} - \boldsymbol{y}_i \tag{2.45}$$

where the quantities $\boldsymbol{\mathcal{B}}_i$ and $\boldsymbol{y}_i$ are listed in Table 2.2 for ease of reference. The coefficient matrix $\boldsymbol{\mathcal{B}}_i$ is an $N \times N$ block matrix with blocks of size $M \times M$ each. Likewise, the driving vector $\boldsymbol{y}_i$ is an $N \times 1$ block vector with entries that are $M \times 1$ each. The matrix $\boldsymbol{\mathcal{B}}_i$ controls the evolution dynamics of the network error vector $\tilde{\boldsymbol{w}}_i$. It is obvious from Table 2.2 that this matrix is different for each of the strategies under consideration. We shall verify in the next chapter that the differences have critical ramifications when we compare consensus and diffusion strategies. Note in passing that any of these three distributed strategies degenerates to the non-cooperative strategy (2.9) when the combination matrix is set to $A = I_N$.

2.2.2 Mean Stability

We start our analysis by examining the stability in the mean of the networks, i.e., the stability of the recursion for $\mathbb{E}\tilde{\boldsymbol{w}}_i$. Thus, note that the matrices $\{\boldsymbol{\mathcal{B}}_i\}$ in Table 2.2 are random matrices due to the randomness of the regressors $\{\boldsymbol{u}_{k,i}\}$ in $\boldsymbol{\mathcal{R}}_i$. In other words, the evolution of the networks is stochastic in nature. Now, since the regressors $\{\boldsymbol{u}_{k,i}\}$ are temporally white and spatially independent, then the $\{\boldsymbol{\mathcal{B}}_i\}$ are independent of $\tilde{\boldsymbol{w}}_{i-1}$ for any of the strategies. Moreover, since the $\{\boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$ are independent of each other, then the $\{\boldsymbol{y}_i\}$ are zero mean. Taking

Table 2.2: The network weight error vector evolves according to the recursion $\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \cdot \tilde{\boldsymbol{w}}_{i-1} - \boldsymbol{y}_i$, where the variables $\{\boldsymbol{\mathcal{B}}_i, \boldsymbol{y}_i\}$, and their respective means or covariances, are listed below for three cooperative strategies and the non-cooperative strategy.

	ATC diffusion (2.29)	**CTA diffusion** (2.32)	**Consensus** (2.24)
$\boldsymbol{\mathcal{B}}_i$	$\mathcal{A}^T(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)$	$(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}^T$	$\mathcal{A}^T - \mathcal{M}\boldsymbol{\mathcal{R}}_i$
$\mathcal{B} \triangleq \mathbb{E}\boldsymbol{\mathcal{B}}_i$	$\mathcal{A}^T(I_{NM} - \mathcal{M}\mathcal{R})$	$(I_{NM} - \mathcal{M}\mathcal{R})\mathcal{A}^T$	$\mathcal{A}^T - \mathcal{M}\mathcal{R}$
$\boldsymbol{y}_i$	$\mathcal{A}^T\mathcal{M}\boldsymbol{s}_i$	$\mathcal{M}\boldsymbol{s}_i$	$\mathcal{M}\boldsymbol{s}_i$
$\mathcal{Y} \triangleq \mathbb{E}\boldsymbol{y}_i\boldsymbol{y}_i^*$	$\mathcal{A}^T\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A}$	$\mathcal{M}\mathcal{S}\mathcal{M}$	$\mathcal{M}\mathcal{S}\mathcal{M}$

expectation of both sides of (2.45), we find that the mean of $\tilde{\boldsymbol{w}}_i$ evolves in time according to the recursion:

$$\mathbb{E}\tilde{\boldsymbol{w}}_i = \mathcal{B} \cdot \mathbb{E}\tilde{\boldsymbol{w}}_{i-1} \tag{2.46}$$

where

$$\mathcal{B} \triangleq \mathbb{E}\boldsymbol{\mathcal{B}}_i \tag{2.47}$$

is shown in Table 2.2 and

$$\mathcal{R} \triangleq \mathbb{E}\boldsymbol{\mathcal{R}}_i = \text{diag}\{R_{u,1}, R_{u,2}, \cdots, R_{u,N}\} \tag{2.48}$$

The necessary and sufficient condition to ensure mean stability of the network (namely, $\mathbb{E}\tilde{\boldsymbol{w}}_i \to 0$ as $i \to \infty$) is therefore to select step-sizes $\{\mu_k\}$ that ensure [34]:

$$\rho(\mathcal{B}) < 1 \tag{2.49}$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument. Note that the coefficient matrices $\{\mathcal{B}\}$ that control the evolution of $\mathbb{E}\tilde{\boldsymbol{w}}_i$ are different in the cases listed in Table 2.2.

2.2.3 Mean-Square Stability

We now examine the stability in the mean-square sense of the consensus and diffusion strategies. Let Σ denote an arbitrary nonnegative-definite matrix that we are free to choose, and let $\sigma = \mathrm{vec}(\Sigma)$ denote the vector that is obtained by stacking the columns of Σ on top of each other. We shall interchangeably use the notation $\|x\|^2_\Sigma$ and $\|x\|^2_\sigma$ to refer to the same weighted square quantity, i.e.,

$$\|x\|^2_\sigma = \|x\|^2_\Sigma = x^*\Sigma x. \tag{2.50}$$

From (2.45), we get the following weighted variance relation:

$$\begin{aligned}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_\Sigma &= \mathbb{E}\|\boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{w}}_{i-1} - \boldsymbol{y}_i\|^2_\Sigma \\ &= \mathbb{E}\left(\tilde{\boldsymbol{w}}^*_{i-1}\boldsymbol{\mathcal{B}}^*_i\Sigma\boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{w}}_{i-1}\right) + \mathbb{E}\left(\boldsymbol{y}^*_i\Sigma\boldsymbol{y}_i\right) - \mathbb{E}\left(\tilde{\boldsymbol{w}}^*_{i-1}\boldsymbol{\mathcal{B}}^*_i\Sigma\boldsymbol{y}_i\right) - \mathbb{E}\left(\boldsymbol{y}^*_i\Sigma\boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{w}}_{i-1}\right)\end{aligned} \tag{2.51}$$

Note that the last two terms in (2.51) are zero because the $\{\boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$ are independent of each other and the $\{\boldsymbol{v}_k(i)\}$ are zero mean. Let us evaluate the first two terms on the right-hand side of (2.51). Since the $\{\boldsymbol{\mathcal{B}}_i, \tilde{\boldsymbol{w}}_{i-1}\}$ are independent of each other, the first term of (2.51) is given by

$$\begin{aligned}\mathbb{E}\left(\tilde{\boldsymbol{w}}^*_{i-1}\boldsymbol{\mathcal{B}}^*_i\Sigma\boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{w}}_{i-1}\right) &= \mathbb{E}\left[\tilde{\boldsymbol{w}}^*_{i-1}\mathbb{E}\left(\boldsymbol{\mathcal{B}}^*_i\Sigma\boldsymbol{\mathcal{B}}_i\right)\tilde{\boldsymbol{w}}_{i-1}\right] \\ &= \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\Sigma'},\end{aligned} \tag{2.52}$$

where we introduced the nonnegative-definite matrix

$$\Sigma' \triangleq \mathbb{E}\boldsymbol{\mathcal{B}}^*_i\Sigma\boldsymbol{\mathcal{B}}_i \tag{2.53}$$

Moreover, using the property $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$, we get

$$\begin{aligned}\mathbb{E}\left(\boldsymbol{y}^*_i\Sigma\boldsymbol{y}_i\right) &= \mathrm{Tr}\left(\Sigma\mathbb{E}\boldsymbol{y}_i\boldsymbol{y}^*_i\right) \\ &= \mathrm{Tr}\left(\Sigma\mathcal{Y}\right)\end{aligned} \tag{2.54}$$

where

$$\mathcal{Y} \triangleq \mathbb{E}\boldsymbol{y}_i\boldsymbol{y}_i^* \tag{2.55}$$

appears in Table 2.2 with the covariance matrix $\mathcal{S}$ defined by:

$$\mathcal{S} \triangleq \mathbb{E}\boldsymbol{s}_i\boldsymbol{s}_i^* = \text{diag}\{\sigma_{v,1}^2 R_{u,1}, \sigma_{v,2}^2 R_{u,2}, \ldots, \sigma_{v,N}^2 R_{u,N}\} \tag{2.56}$$

Then, using the following matrix identities for arbitrary matrices $\{U, W, \Sigma\}$ of compatible dimensions [123]:

$$\text{vec}(U\Sigma W) = (W^T \otimes U)\sigma \quad \text{and} \quad \text{Tr}(\Sigma W) = \left[\text{vec}(W^T)\right]^T \sigma \tag{2.57}$$

and from (2.52)-(2.54), we can rewrite relation (2.51) in the following equivalent form:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_\sigma^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 + \left[\text{vec}(\mathcal{Y}^T)\right]^T \sigma \tag{2.58}$$

where $\mathcal{F}$ is the $N^2M^2 \times N^2M^2$ matrix defined by:

$$\mathcal{F} \triangleq \mathbb{E}\left(\boldsymbol{\mathcal{B}}_i^T \otimes \boldsymbol{\mathcal{B}}_i^*\right) \tag{2.59}$$

Note that relation (2.58) is not an actual recursion; this is because the weighting matrices $\{\sigma, \mathcal{F}\sigma\}$ on both sides of the equality are different. The relation can be transformed into a true recursion by expanding it into a convenient state-space model; this argument was pursued in [34, 123] and is briefly reviewed below to arrive at conditions on the step-sizes for mean-square stability.

Let

$$L = N^2M^2 \quad \text{and} \quad y = \text{vec}(\mathcal{Y}^T) \tag{2.60}$$

and introduce the characteristic polynomial of the matrix $\mathcal{F}$, i.e.,

$$p(x) = \det(xI_L - \mathcal{F}) = x^L + p_{L-1}x^{L-1} + \cdots + p_0 \tag{2.61}$$

for some polynomial coefficients $\{p_j\}$. By the Cayley-Hamilton Theorem [64,84], we have that $p(\mathcal{F}) = 0$ so that

$$\mathcal{F}^L = -p_0 - p_1\mathcal{F} - \cdots - p_{L-1}\mathcal{F}^{L-1} \tag{2.62}$$

Applying (2.58) repeatedly and using (2.62) we arrive at the following state-space recursion:

$$\underbrace{\begin{bmatrix} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_\sigma^2 \\ \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\mathcal{F}\sigma}^2 \\ \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\mathcal{F}^2\sigma}^2 \\ \vdots \\ \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\mathcal{F}^{L-1}\sigma}^2 \end{bmatrix}}_{\mathcal{W}_i} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \\ -p_0 & -p_1 & -p_2 & \cdots & -p_{L-1} \end{bmatrix}}_{\mathcal{H}} \underbrace{\begin{bmatrix} \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_\sigma^2 \\ \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 \\ \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}^2\sigma}^2 \\ \vdots \\ \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}^{L-1}\sigma}^2 \end{bmatrix}}_{\mathcal{W}_{i-1}} + \underbrace{\begin{bmatrix} y^T\sigma \\ y^T\mathcal{F}\sigma \\ y^T\mathcal{F}^2\sigma \\ \vdots \\ y^T\mathcal{F}^{L-1}\sigma \end{bmatrix}}_{\mathcal{Z}} \tag{2.63}$$

We therefore arrive at a true recursion that describes the evolution of the state of the adaptive network:

$$\mathcal{W}_i \;=\; \mathcal{H}\cdot\mathcal{W}_{i-1} \;+\; \mathcal{Z} \tag{2.64}$$

with $\mathcal{W}_i$ denoting the state vector at time i. The matrix $\mathcal{H}$ in (2.63) is in companion form, and it is known that its eigenvalues are the roots of $p(x)$ from (2.61) [64], which are also the eigenvalues of $\mathcal{F}$. Therefore, a necessary and sufficient condition for mean-square stability of the network (namely, $\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_\sigma^2 \to c < \infty$ as $i \to \infty$) is to select the step-sizes $\{\mu_k\}$ such that $\mathcal{H}$ is a stable matrix, or equivalently, that all the eigenvalues of $\mathcal{F}$ are inside the unit circle, i.e.,

$$\rho(\mathcal{F}) < 1 \tag{2.65}$$

where $\mathcal{F}$ is defined by (2.59). A simpler condition for mean-square stability can be obtained by assuming sufficiently small step-sizes since in this case we can verify that the matrix $\mathcal{F}$ in (2.59) can be approximated as follows:

$$\mathcal{F} \approx \mathcal{B}^T \otimes \mathcal{B}^* \tag{2.66}$$

in terms of the mean matrix $\mathcal{B}$ that appears in Table 2.2. We justify approximation (2.66) for the ATC diffusion strategy (2.29); the same argument applies to CTA diffusion (2.32) and consensus (2.24). From (2.59) and Table 2.2, the matrix $\mathcal{F}$ for ATC diffusion takes the form:

$$\begin{aligned}\mathcal{F} &= \mathbb{E}\left[(I-\boldsymbol{\mathcal{R}}_i^T\mathcal{M})\mathcal{A}\otimes(I-\boldsymbol{\mathcal{R}}_i\mathcal{M})\mathcal{A}\right] \\ &= \left[I-(\mathcal{R}^T\mathcal{M})\otimes I - I\otimes(\mathcal{R}\mathcal{M}) + \mathbb{E}(\boldsymbol{\mathcal{R}}_i^T\mathcal{M})\otimes(\boldsymbol{\mathcal{R}}_i\mathcal{M})\right](\mathcal{A}\otimes\mathcal{A}) \qquad (2.67)\end{aligned}$$

where we used the following Kronecker product property for matrices $\{A, B, C, D\}$ of compatible dimensions [123]:

$$(A\otimes B)(C\otimes D) = AC\otimes BD \qquad (2.68)$$

Note that the last term on the right-hand side of (2.67) depends on the fourth order of the regressors, which is difficult to evaluate its expectation. To proceed, we introduce the following condition on the step-sizes $\{\mu_k\}$, which will be assumed throughout this work. Such conditions are prevalent in the stochastic gradient approximation literature [15, 62, 82, 83, 90, 91, 123]

Assumption 2.1. *The step-sizes $\{\mu_k\}$ are sufficiently small such that terms that depend on higher-order powers of the step-sizes can be ignored and, in particular,*

$$0 < \mu_k \cdot \rho(R_{u,k}) \ll 1 \text{ for all } k \qquad (2.69)$$

Under Assumption 2.1, we can either ignore terms that depend on higher-order powers of the step-sizes [124] or call upon a separation principle [123] to approximate these terms. The last term on the right-hand side of (2.67) is one such term since its entries depend on $\{\mu_k^2\}$. For small step-sizes, we approximate the expectation of the product by the product of expectations and write:

$$\begin{aligned}\mathbb{E}(\boldsymbol{\mathcal{R}}_i^T\mathcal{M})\otimes(\boldsymbol{\mathcal{R}}_i\mathcal{M}) &\approx \mathbb{E}(\boldsymbol{\mathcal{R}}_i^T\mathcal{M})\otimes\mathbb{E}(\boldsymbol{\mathcal{R}}_i\mathcal{M}) \\ &= (\mathcal{R}^T\mathcal{M})\otimes(\mathcal{R}\mathcal{M}) \qquad (2.70)\end{aligned}$$

Then, expression (2.67) simplifies to:

$$\begin{aligned}\mathcal{F} &\approx \left[I - (\mathcal{R}^T\mathcal{M}) \otimes I - I \otimes (\mathcal{R}\mathcal{M}) + (\mathcal{R}^T\mathcal{M}) \otimes (\mathcal{R}\mathcal{M})\right](\mathcal{A} \otimes \mathcal{A}) \\ &= \left[(I - \mathcal{R}^T\mathcal{M}) \otimes (I - \mathcal{R}\mathcal{M})\right](\mathcal{A} \otimes \mathcal{A}) \\ &= \mathcal{B}^T \otimes \mathcal{B}^* \end{aligned} \tag{2.71}$$

as desired. Now, using (2.66), we note that since

$$\rho(\mathcal{F}) \approx \rho(\mathcal{B}^T) \cdot \rho(\mathcal{B}^*) = [\rho(\mathcal{B})]^2 \tag{2.72}$$

so that sufficiently small step-sizes that satisfy (2.49) will also ensure mean-square stability.

2.2.4 Convergence Rate

Assume that the step-sizes are sufficiently small so that condition (2.49) holds and the networks is stable in the mean and mean-square sense. Under this condition, the network achieves steady-state operation. The convergence rate of the network determines the rate at which the quantity $\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2$ converge towards its steady-state value. We denote the convergence rate by r so that the smaller the value of r is, the faster the rate of convergence of $\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2$ is. As indicated by (2.63), the convergence rate is determined by the spectral radius of the matrix $\mathcal{H}$, i.e.,

$$r = \rho(\mathcal{H}) \approx [\rho(\mathcal{B})]^2 . \tag{2.73}$$

2.2.5 Mean-Square Deviation

The mean-square deviation (MSD) measure can be used to assess how well the agents in the network estimate the weight vector, w°. The MSD at agent k is defined as follows:

$$\text{MSD}_k \triangleq \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \tag{2.74}$$

where $\|\cdot\|$ denotes the Euclidean norm for vectors. The network MSD is defined as the average MSD across the network, i.e.,

$$\text{MSD} \triangleq \frac{1}{N}\sum_{k=1}^{N}\text{MSD}_k \tag{2.75}$$

Now, assume the network is mean-square stable and let the time index i tend to infinity in (2.58). Then, we obtain the steady-state relation:

$$\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2_{(I_L-\mathcal{F})\sigma} = \left[\text{vec}(\mathcal{Y}^T)\right]^T\sigma \tag{2.76}$$

Since $\rho(\mathcal{F}) < 1$, the matrix $(I_L - \mathcal{F})$ is invertible. In order for the left-hand side of (2.76) to match the definition of the MSD at agent k given by (2.74), we select σ as

$$\sigma = (I_L-\mathcal{F})^{-1}\,\text{vec}\left[(e_k e_k^T)\otimes I_M\right] \tag{2.77}$$

where e_k denotes the kth column of the identity matrix I_N. Substituting into (2.76), we arrive at the following expression for the MSD of agent k:

$$\text{MSD}_k = \left[\text{vec}(\mathcal{Y}^T)\right]^T(I_L-\mathcal{F})^{-1}\,\text{vec}\left[(e_k e_k^T)\otimes I_M\right] \tag{2.78}$$

Then, the network MSD in (2.75) is given by:

$$\text{MSD} = \frac{1}{N}\left[\text{vec}(\mathcal{Y}^T)\right]^T(I_L-\mathcal{F})^{-1}\,\text{vec}(I_{NM}) \tag{2.79}$$

Expressions (2.78)-(2.79) are in terms of the inverse matrix $(I_L - \mathcal{F})$. There are alternative representations for these two results in terms of a series expansion that will prove useful in our subsequent analysis. Recall that for any stable matrix $\mathcal{F}$, it holds:

$$(I_L-\mathcal{F})^{-1} = \sum_{j=0}^{\infty}\mathcal{F}^j \tag{2.80}$$

Substituting into (2.78)-(2.79) and using (2.57), we obtain the alternative representations:

$$\text{MSD}_k = \sum_{j=0}^{\infty}\text{Tr}\left[(e_k^T\otimes I_M)\cdot\mathcal{B}^j\mathcal{Y}\mathcal{B}^{*j}\cdot(e_k\otimes I_M)\right] \tag{2.81}$$

and

$$\text{MSD} = \frac{1}{N}\sum_{j=0}^{\infty}\text{Tr}[\mathcal{B}^j\mathcal{Y}(\mathcal{B}^*)^j] \tag{2.82}$$

Expressions (2.81)-(2.82) relate the MSDs directly to the quantities $\{\mathcal{B},\mathcal{Y}\}$ from Table 2.2.

Note that the expressions for the individual MSD in (2.81) and the network MSD in (2.82) depend on $\mathcal{B}$ in a nontrivial manner. These expressions can be further simplified if we know the eigen-structure of the matrix $\mathcal{B}$. Assume that $\mathcal{B}$ is diagonalizable and its eigen-decomposition is given by:

$$\mathcal{B} = \sum_{l=1}^{N}\sum_{m=1}^{M}\lambda_{l,m}(\mathcal{B})\cdot r_{l,m}^{b}s_{l,m}^{b*}. \tag{2.83}$$

Using (2.83), we can rewrite the MSD at agent k from (2.81) as:

$$\begin{aligned}\text{MSD}_k = \sum_{j=0}^{\infty}\sum_{l_1,l_2=1}^{N}\sum_{m_1,m_2=1}^{M}\text{Tr}\big[&\lambda_{l_1,m_1}^{j}(\mathcal{B})\lambda_{l_2,m_2}^{*j}(\mathcal{B})\cdot(e_k^T\otimes I_M)\\ &\times r_{l_1,m_1}^{b}s_{l_1,m_1}^{b*}\mathcal{Y}s_{l_2,m_2}^{b}r_{l_2,m_2}^{b*}\cdot(e_k\otimes I_M)\big]\end{aligned} \tag{2.84}$$

Using $\text{Tr}(AB) = \text{Tr}(BA)$ and the expression for the infinite sum of a geometric series, we have:

$$\text{MSD}_k = \sum_{l_1,l_2=1}^{N}\sum_{m_1,m_2=1}^{M}\frac{\big(r_{l_2,m_2}^{b*}(e_k\otimes I_M)(e_k^T\otimes I_M)r_{l_1,m_1}^{b}\big)\cdot\big(s_{l_1,m_1}^{b*}\mathcal{Y}s_{l_2,m_2}^{b}\big)}{1-\lambda_{l_1,m_1}(\mathcal{B})\lambda_{l_2,m_2}^{*}(\mathcal{B})} \tag{2.85}$$

Moreover, from (2.75) and (2.85), and using the fact that

$$\sum_{k=1}^{N}(e_k\otimes I_M)(e_k^T\otimes I_M) = I_{NM} \tag{2.86}$$

we obtain that

$$\text{MSD} = \frac{1}{N}\sum_{l_1,l_2=1}^{N}\sum_{m_1,m_2=1}^{M}\frac{\big(r_{l_2,m_2}^{b*}r_{l_1,m_1}^{b}\big)\cdot\big(s_{l_1,m_1}^{b*}\mathcal{Y}s_{l_2,m_2}^{b}\big)}{1-\lambda_{l_1,m_1}(\mathcal{B})\lambda_{l_2,m_2}^{*}(\mathcal{B})} \tag{2.87}$$

2.3 Concluding Remarks

In this chapter, we derived the diffusion and consensus strategies using the same formalism. We derived conditions for mean and mean-square stability and derived expressions for the convergence rate and mean-square-error performance in preparation for the future chapters.

2.A Properties of the Combination Matrix

From (2.25), the combination matrix A is left-stochastic. It follows that the spectral radius of A is equal to one and one of its eigenvalue is also equal to one [124]. We denote this eigenvalue by $\lambda_1(A)$. Then, we obtain that

$$\rho(A) = \lambda_1(A) = 1 \tag{2.88}$$

That is, the eigenvalues of A lie within the unit circle. However, it is possible that there are multiple eigenvalues of A with magnitude equal to one. Usually, we are interested in networks satisfying the following condition.

Assumption 2.2 (Strongly connected networks)**.** *The network topology is strongly connected so that a path with nonzero weights exists between any two arbitrary agents and at least one agent has self-loop (i.e., $a_{k,k} > 0$ for some k)*

Thus, in effect, Assumption 2.2 is requiring the network to be connected with at least one self-loop (in which case, there is at least one agent in the network that assigns a positive weight to its local information). This condition is satisfied for most networks of interest since it is difficult to envision situations where all agents in a network do not have some level of trust in their own data. Although the analysis in this chapter holds for arbitrary network topologies, strongly connected

networks exhibit interesting behavior as the analysis in the ensuing chapters will show. In particular, among other results, we establish the following result.

Lemma 2.1. *If a network is strongly connected, then the corresponding combination matrix A is primitive, i.e., there exists an integer power $j > 0$ such that*

$$[A^j]_{l,k} > 0 \quad \textit{for all } l \textit{ and } k. \tag{2.89}$$

Expression (2.89) is equivalent to the condition that for any two agents l and k, there is a path from agent l to agent k in j steps [64].

Proof. Without loss of generality, we assume that agent m has a self-loop and $a_{m,m} > 0$. For arbitrary k and l and for three nonnegative integers j_1, j_2, and j_3 satisfying

$$j_1 + j_2 + j_3 = j \tag{2.90}$$

it holds that

$$\begin{aligned}
[A^j]_{l,k} &= \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} [A^{j_1}]_{l,n_1} [A^{j_2}]_{n_1,n_2} [A^{j_3}]_{n_2,k} \\
&\geq [A^{j_1}]_{l,m} [A^{j_2}]_{m,m} [A^{j_3}]_{m,k} \\
&\geq [A^{j_1}]_{l,m} \cdot a_{m,m}^{j_2} \cdot [A^{j_3}]_{m,k}
\end{aligned} \tag{2.91}$$

Since the network is connected, there exist two finite nonnegative integers $j_1(l)$ and $j_3(k)$ such that

$$[A^{j_1(l)}]_{l,m} > 0 \quad \text{and} \quad [A^{j_3(k)}]_{m,k} > 0 \tag{2.92}$$

Let $j_1^\circ(l)$ and $j_3^\circ(k)$ be the minimum nonnegative integers such that relations in (2.92) hold. Then, we set

$$j = \max_l \{j_1^\circ(l)\} + \max_k \{j_3^\circ(k)\} < \infty \tag{2.93}$$

and we obtain that

$$[A^j]_{l,k} \geq [A^{j_1^\circ(l)}]_{l,m} \cdot a_{m,m}^{j-j_1^\circ(l)-j_3^\circ(k)} \cdot [A^{j_3^\circ(k)}]_{m,k} \tag{2.94}$$

Since $a_{m,m} > 0$ and $j - j_1^\circ(l) - j_3^\circ(k) \geq 0$, it holds that

$$a_{m,m}^{j-j_1^\circ(l)-j_3^\circ(k)} > 0 \tag{2.95}$$

Therefore, relation (2.89) holds for any l and k and the matrix A is primitive. □

From the Perron-Frobenius Theorem [17, 64, 112, 124], the eigen-structure of any *primitive* A satisfies certain prominent properties, which will be useful in the sequel, namely, that

1. A has an eigenvalue at $\lambda_1(A) = 1$
2. The eigenvalue at $\lambda_1(A) = 1$ has multiplicity one
3. All the entries of the right and left eigenvectors associated with $\lambda_1(A) = 1$ can be scaled to be positive
4. $\rho(A) = 1$ so that all other eigenvalues of A have magnitude strictly less than one

In order to simplify the MSD expressions, we sometimes require the combination matrix A to be diagonalizable, i.e., that there exists an invertible matrix U and a diagonal matrix Λ such that

$$A^T = U\Lambda U^{-1} \tag{2.96}$$

with

$$U = \begin{bmatrix} r_1 & r_2 & \cdots & r_N \end{bmatrix}, \quad U^{-1} = \mathrm{col}\{s_1^*, s_2^*, \ldots, s_N^*\} \tag{2.97}$$

$$\Lambda = \mathrm{diag}\{\lambda_1(A), \lambda_2(A), \ldots, \lambda_N(A)\}. \tag{2.98}$$

The columns of U consist of the right eigenvectors of A^T, the rows of U^{-1} consist of the left eigenvectors of A^T, and the diagonal entries of Λ consist of the eigenvalues of A^T. That is,

$$A^T \cdot r_l = \lambda_l(A) \cdot r_l \quad \text{and} \quad s_l^* \cdot A^T = \lambda_l(A) \cdot s_l^*. \tag{2.99}$$

for $l = 1, 2, \ldots, N$. We assume the vectors $\{r_l\}$ are scaled to satisfy:

$$\|r_l\| = 1 \tag{2.100}$$

for all l. In addition, it follows that

$$s_{l_2}^* r_{l_1} = \delta_{l_1 l_2} \tag{2.101}$$

since $U^{-1}U = I_N$. Under conditions (2.100)-(2.101) and from (2.25), it can be verified that the right and left eigenvector pair $\{r_1, s_1\}$ of A corresponding to the eigenvalue $\lambda_1(A) = 1$ satisfies:

$$r_1 = \frac{\mathbb{1}_N}{\sqrt{N}} \quad \text{and} \quad \frac{s_1^T \mathbb{1}_N}{\sqrt{N}} = 1. \tag{2.102}$$

Note that any *symmetric* combination matrix A is diagonalizable and therefore satisfies condition (2.96) automatically. Actually, when A is symmetric, more can be said about its eigenvectors. In that case, the matrix U will be orthogonal so that $U^{-1} = U^*$ and it will further hold that

$$r_{l_2}^* r_{l_1} = \delta_{l_1 l_2} \tag{2.103}$$

Nevertheless, condition (2.96) allows the analysis to apply to important cases in which A is not necessarily symmetric but is still diagonalizable (such as when A is constructed according to the uniform combination rule by assigning to the links of agent k weights that are equal to the inverse of its degree, n_k).

CHAPTER 3

Diffusion Strategies Outperform Consensus Strategies

In this chapter, we compare the mean-square performance of the two main strategies derived in Chapter 2 for distributed estimation over networks: consensus strategies from (2.24) and diffusion strategies from (2.29) and (2.32). The analysis in this chapter will confirm that under *constant* step-sizes, diffusion strategies allow information to diffuse more thoroughly through networks and this property has a favorable effect on the evolution of the network. It will be shown that diffusion networks converge faster and reach lower mean-square deviation than consensus networks, and their mean-square stability is insensitive to the choice of the combination weights. In comparison, and surprisingly, it is shown that consensus networks can become unstable even if all the individual agents are stable and able to solve the estimation task on their own. In other words, the learning curve of a cooperative consensus network can diverge even if the learning curves for the non-cooperative individual agents converge. When this occurs, cooperation over the network leads to a catastrophic failure of the estimation task. This behavior does not occur for diffusion networks: we will show that stability of the individual agents is sufficient to ensure stability of the diffusion network *regardless* of the combination weights. The properties revealed in this chapter indicate that there needs to be some care with the use of consensus strategies for

adaptation because they can lead to network failure even if the individual agents are stable and well-behaved. The analysis also suggests that diffusion strategies provide a proper way to enforce cooperation over networks; their operation is such that diffusion networks will always remain stable irrespective of the combination topology.

3.1 Comparison of Mean and Mean-square Stability

As mentioned in Chapter 2, for sufficiently small step-sizes, the mean and mean-square stability are governed by the spectral radius of the matrix $\mathcal{B}$, which are different in the cases listed in Table 2.2. These differences lead to interesting conclusions. To begin with, the matrix $\mathcal{B}$ is block diagonal in the non-cooperative case and equal to

$$\mathcal{B}_{\text{ncop}} = I_{NM} - \mathcal{M}\mathcal{R}. \tag{3.1}$$

where $\mathcal{M}$ and $\mathcal{R}$ were defined in (2.38) and (2.48), respectively. Therefore, for each of the individual agents to be stable in the mean, it is necessary and sufficient that the step-sizes $\{\mu_k\}$ be selected to satisfy

$$\rho(\mathcal{B}_{\text{ncop}}) = \max_{1\leq k\leq N} \rho(I_M - \mu_k R_{u,k}) < 1 \tag{3.2}$$

since the matrices $\mathcal{M}$ and $\mathcal{R}$ are block diagonal. Condition (3.2) is equivalent to

$$\text{(stability in the non-cooperative case)} \quad 0 < \mu_k < \frac{2}{\lambda_{\max}(R_{u,k})} \tag{3.3}$$

for $k = 1, 2, \ldots, N$. Sufficiently small step-sizes that meet condition (3.3) guarantee that when each agent acts individually and applies the LMS recursion (2.9), then *all* individual agents will be stable in the mean and mean-square sense.

Now consider the matrix $\mathcal{B}$ in the consensus case; it is equal to

$$\mathcal{B}_{\text{cons}} = \mathcal{A}^T - \mathcal{M}\mathcal{R} \tag{3.4}$$

where $\mathcal{A}$ was introduced in (2.39). It is seen in this case that the stability of $\mathcal{B}_{\text{cons}}$ depends on $\mathcal{A}$. The fact that the stability of the consensus strategy is sensitive to the choice of the combination matrix is known in the consensus literature for the conventional implementation for computing averages and which does not involve streaming data or gradient noise [16, 49]. Here, we are studying the more demanding case of the single time-scale consensus iteration (2.24) in the presence of both *noisy* and *streaming* data. It is clear from (3.4) that the choice of A can destroy the stability of the consensus network even when the step-sizes are chosen according to (3.3) and all agents are stable on their own. This behavior does not occur for diffusion networks where the matrices $\{\mathcal{B}\}$ for the ATC and CTA diffusion strategies are instead given by

$$\mathcal{B}_{\text{atc}} = \mathcal{A}^T(I_{NM} - \mathcal{M}\mathcal{R}) \quad \text{and} \quad \mathcal{B}_{\text{cta}} = (I_{NM} - \mathcal{M}\mathcal{R})\mathcal{A}^T. \tag{3.5}$$

The following result clarifies these statements.

Theorem 3.1 (Spectral properties of $\mathcal{B}$)**.** *It holds that*

$$\rho(\mathcal{B}_{\text{atc}}) = \rho(\mathcal{B}_{\text{cta}}) \leq \rho(\mathcal{B}_{\text{ncop}}) \tag{3.6}$$

irrespective of the choice of the left-stochastic matrices A. Moreover, if the combination matrix A is symmetric, then the eigenvalues of $\mathcal{B}_{\text{cons}}$ are less than or equal to the corresponding eigenvalues of $\mathcal{B}_{\text{ncop}}$, i.e.,

$$\lambda_l(\mathcal{B}_{\text{cons}}) \leq \lambda_l(\mathcal{B}_{\text{ncop}}) \tag{3.7}$$

for $l = 1, 2, \ldots, NM$ where the eigenvalues $\{\lambda_l(\cdot)\}$ are arranged in decreasing order, i.e., $\lambda_{l_1}(\cdot) \geq \lambda_{l_2}(\cdot)$ if $l_1 \leq l_2$.

Proof. See Appendix 3.A. □

Result (3.6) establishes the important conclusion that the coefficient matrix $\mathcal{B}$ for the diffusion strategies is stable whenever $\mathcal{B}_{\rm ncop}$ (or, from (3.2), each of the matrices $\{I_M - \mu_k R_{u,k}\}$) is stable; *this conclusion is independent of* A. The stability of the matrices $\{I_M - \mu_k R_{u,k}\}$ is ensured by any step-size satisfying (3.3). Therefore, stability of the individual agents will always guarantee the stability of $\mathcal{B}$ in the ATC and CTA diffusion cases, *regardless* of the choice of A. This is not the case for the consensus strategy (2.24); even when the step-sizes $\{\mu_k\}$ are selected to satisfy (3.3) so that all individual agents are mean stable, the matrix $\mathcal{B}_{\rm cons}$ can still be unstable depending on the choice of A (and, therefore, on the network topology as well). Therefore, if we start from a collection of agents that are behaving in a stable manner on their own, and if we connect them through a topology and then apply consensus to solve the same estimation problem through cooperation, then the network may end up being unstable and the estimation task can fail drastically (see Fig. 3.1 further ahead). Moreover, it is further shown in Appendix 3.A that when A is symmetric, the consensus strategy is stable for step-sizes satisfying:

$$0 < \mu_k < \frac{1 + \lambda_{\min}(A)}{\lambda_{\max}(R_{u,k})} \quad \text{for } k = 1, 2, \ldots, N. \tag{3.8}$$

Recall from Appendix 2.A that $\lambda_1(A) = \rho(A) = 1$. This implies that the upper bound in (3.8) is less than the upper bound in (3.3) so that diffusion networks are stable over a wider range of step-sizes. Actually, the upper bound in (3.8) can be much smaller than the one in (3.3) or even zero because $\lambda_{\min}(A)$ can be negative or equal to -1.

What if some of the agents are unstable in the mean to begin with? How would the behavior of the diffusion and consensus strategies differ? Assume that there is at least one individual unstable agent, i.e., $\lambda_l(\mathcal{B}_{\rm ncop}) \leq -1$ for some l so that $\rho(\mathcal{B}_{\rm ncop}) \geq 1$. Then, we observe from (3.6) that the spectral radius of

$\mathcal{B}_{\rm atc}$ can still be smaller than one even if $\rho(\mathcal{B}_{\rm ncop}) \geq 1$. It follows that even if some individual agent is unstable, the diffusion strategies can still be stable if we properly choose A. In other words, diffusion cooperation has a stabilizing effect on the network. In contrast, if there is at least one individual unstable agent and the combination matrix A is symmetric, then from (3.7), no matter how we choose A, the $\rho(\mathcal{B}_{\rm cons})$ will be larger than or equal to one and the consensus network will be unstable.

The above results suggest that fusing results from neighborhoods according to the consensus strategy (2.24) is not necessarily the best thing to do because it can lead to instability and catastrophic failure. On the other hand, fusing the results from neighbors via diffusion ensures stability regardless of the topology.

3.1.1 Example: Two-Agent Networks

To illustrate these important observations, let us consider an example consisting of two cooperating agents; in this case, it is possible to carry out the calculations analytically in order to highlight the various patterns of behavior. Later, in the simulations section, we illustrate the behavior for networks with multiple agents. Thus, consider a network consisting of $N = 2$ agents. For simplicity, we assume the weight vector w° is a scalar, and $R_{u,1} = \sigma_{u,1}^2$ and $R_{u,2} = \sigma_{u,2}^2$. Without loss of generality, we assume $\mu_1\sigma_{u,1}^2 \leq \mu_2\sigma_{u,2}^2$. The combination matrix for this example is of the form (see Fig. 3.1 further ahead):

$$A^T = \begin{bmatrix} 1-a & a \\ b & 1-b \end{bmatrix}. \tag{3.9}$$

with $a, b \in [0, 1]$. When desired, a symmetric A can be selected by simply setting $a = b$. Then, using (3.9), we get

$$\mathcal{B}_{\text{atc}} = \begin{bmatrix} (1-\mu_1\sigma_{u,1}^2)(1-a) & (1-\mu_2\sigma_{u,2}^2)a \\ (1-\mu_1\sigma_{u,1}^2)b & (1-\mu_2\sigma_{u,2}^2)(1-b) \end{bmatrix} \tag{3.10}$$

$$\mathcal{B}_{\text{cons}} = \begin{bmatrix} 1-a-\mu_1\sigma_{u,1}^2 & a \\ b & 1-b-\mu_2\sigma_{u,2}^2 \end{bmatrix}. \tag{3.11}$$

We first assume that

$$0 < \mu_1\sigma_{u,1}^2 \leq \mu_2\sigma_{u,2}^2 < 2 \tag{3.12}$$

so that both individual agents are stable in the mean by virtue of (3.3). Then, by Theorem 3.1, the ATC diffusion network will also be stable in the mean for any choice of the parameters $\{a, b\}$. We now verify that there are choices for $\{a, b\}$ that will turn the consensus network unstable. Specifically, we verify below that if a and b happen to satisfy

$$a + b \geq 2 - \mu_1\sigma_{u,1}^2 \tag{3.13}$$

then consensus will lead to unstable network behavior even though both individual agents are stable. Indeed, note first that the eigenvalues of $\mathcal{B}_{\text{cons}}$ are given by:

$$\lambda(\mathcal{B}_{\text{cons}}) = \frac{(2-a-b-\mu_1\sigma_{u,1}^2-\mu_2\sigma_{u,2}^2) \pm \sqrt{D}}{2} \tag{3.14}$$

where

$$\begin{aligned} D &\triangleq (-a+b-\mu_1\sigma_{u,1}^2+\mu_2\sigma_{u,2}^2)^2 + 4ab \\ &= (a+b+\mu_1\sigma_{u,1}^2-\mu_2\sigma_{u,2}^2)^2 + 4b(\mu_2\sigma_{u,2}^2-\mu_1\sigma_{u,1}^2). \end{aligned} \tag{3.15}$$

From the first equality of (3.15), we know that $D \geq 0$. Hence, the eigenvalues of $\mathcal{B}_{\text{cons}}$ are real and the minimum eigenvalue of $\mathcal{B}_{\text{cons}}$ is given by:

$$\lambda_{\min}(\mathcal{B}_{\text{cons}}) = \frac{(2-a-b-\mu_1\sigma_{u,1}^2-\mu_2\sigma_{u,2}^2) - \sqrt{D}}{2} \tag{3.16}$$

When (3.12)-(3.13) are satisfied, we have that

$$(a+b+\mu_1\sigma_{u,1}^2-\mu_2\sigma_{u,2}^2) \quad \text{and} \quad 4b(\mu_2\sigma_{u,2}^2-\mu_1\sigma_{u,1}^2) \tag{3.17}$$

in the second equality of (3.15) are nonnegative. It follows that the consensus network is unstable since

$$\begin{aligned}\lambda_{\min}(\mathcal{B}_{\text{cons}}) &\leq \frac{(2-a-b-\mu_1\sigma_{u,1}^2-\mu_2\sigma_{u,2}^2)-(a+b+\mu_1\sigma_{u,1}^2-\mu_2\sigma_{u,2}^2)}{2} \\ &\leq -1. \end{aligned} \tag{3.18}$$

In Fig. 3.1(a), we set $\mu_1\sigma_{u,1}^2=0.4$ and $\mu_2\sigma_{u,2}^2=0.6$ so that each individual agent is stable. If we now set $a=b=0.85$, then (3.13) is satisfied and the consensus strategy becomes unstable.

Next, we consider an example satisfying

$$0<\mu_1\sigma_{u,1}^2<2\leq\mu_2\sigma_{u,2}^2 \tag{3.19}$$

so that agent 1 is still stable, whereas agent 2 becomes unstable. From the first equality of (3.15), we again conclude that

$$\begin{aligned}\lambda_{\min}(\mathcal{B}_{\text{cons}}) &\leq \frac{(2-a-b-\mu_1\sigma_{u,1}^2-\mu_2\sigma_{u,2}^2)-|-a+b-\mu_1\sigma_{u,1}^2+\mu_2\sigma_{u,2}^2|}{2} \\ &= \begin{cases} 1-a-\mu_1\sigma_{u,1}^2, & \text{if } a+\mu_1\sigma_{u,1}^2\geq b+\mu_2\sigma_{u,2}^2 \\ 1-b-\mu_2\sigma_{u,2}^2, & \text{otherwise} \end{cases} \\ &\leq -1. \end{aligned} \tag{3.20}$$

That is, in this second case, no matter how we choose the parameters $\{a,b\}$, the consensus network is always unstable. In contrast, the diffusion network is able to stabilize the network. To see this, we set $b=1-a$ so that the eigenvalues of $\mathcal{B}_{\text{atc}}$ in (3.10) are

$$\{0, 1-\mu_1\sigma_{u,1}^2-(\mu_2\sigma_{u,2}^2-\mu_1\sigma_{u,1}^2)a\}. \tag{3.21}$$

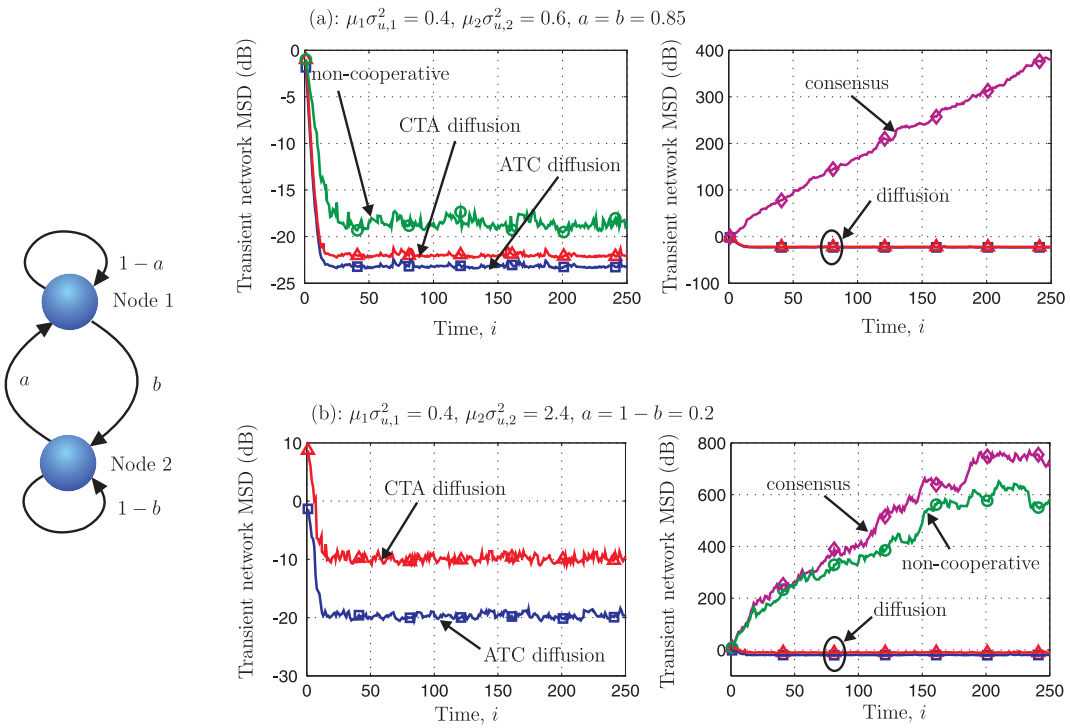


Figure 3.1: Transient network MSD over time with $N = 2$. (a) $\mu_1\sigma_{u,1}^2 = 0.4$, $\mu_2\sigma_{u,2}^2 = 0.6$, and $a = b = 0.85$. As seen in the right plot, the consensus strategy is unstable even when the individual agents are stable. (b) $\mu_1\sigma_{u,1}^2 = 0.4$, $\mu_2\sigma_{u,2}^2 = 2.4$, and $a = 1 - b = 0.2$ so that agent 2 is unstable. As seen in the right plot, the diffusion strategies are able to stabilize the network even when the non-cooperative and consensus strategies are unstable.

Some algebra shows that the diffusion network is stable if a satisfies

$$0 \leq a < \frac{2 - \mu_1 \sigma_{u,1}^2}{\mu_2 \sigma_{u,2}^2 - \mu_1 \sigma_{u,1}^2}. \tag{3.22}$$

In Fig. 3.1(b), we set $\mu_1 \sigma_{u,1}^2 = 0.4$ and $\mu_1 \sigma_{u,1}^2 = 2.4$ so that agent 1 is stable, but agent 2 is unstable. If we now set $a = 1 - b = 0.2$, then (3.22) is satisfied and the diffusion strategies become stable even when the non-cooperative and consensus strategies are unstable.

3.2 Comparison of Mean-Square Performance

In the previous section, we compared the stability of the various estimation strategies in the mean sense. In particular, we established that stability of the individual agents ensures stability of diffusion networks irrespective of the combination topology. In the sequel, we shall assume that the step-sizes are sufficiently small so that conditions (3.3) and (3.8) hold and the diffusion and consensus networks are stable both in the mean and mean-square senses; as well as the individual agents. Under these conditions, the networks achieve steady-state operation. We now establish that ATC diffusion achieves lower (and, hence, better) MSD values at faster rate than the consensus, CTA, and non-cooperative strategies. In this way, diffusion strategies do not only ensure stability of the cooperative behavior but they also lead to improved mean-square-error performance.

In order to quantify the differences in performance without biasing the results by differences in the adaptation mechanism (step-sizes) or in the covariance matrices of the regression data at the agents, we make the following reasonable condition, which assumes that all agents have similar processing abilities.

Assumption 3.1 (Homogeneous agents)**.** *All agents in the network use the same step-size, $\mu_k = \mu$, and they observe data arising from the same covariance data*

Table 3.1: Variables for cooperative and non-cooperative implementations when $\mu_k = \mu$ and $R_{u,k} = R_u$ for all k.

	ATC diffusion (2.29)	**CTA diffusion** (2.32)	**Consensus** (2.24)
$\mathcal{B}$	$A^T \otimes I_M - A^T \otimes \mu R_u$	$A^T \otimes I_M - A^T \otimes \mu R_u$	$A^T \otimes I_M - I_N \otimes \mu R_u$
$\lambda_{l,m}(\mathcal{B})$	$\lambda_l(A)(1 - \mu\lambda_m(R_u))$	$\lambda_l(A)(1 - \mu\lambda_m(R_u))$	$\lambda_l(A) - \mu\lambda_m(R_u)$
$\mathcal{Y}$	$\mu^2(A^T\Sigma_v A) \otimes R_u$	$\mu^2\Sigma_v \otimes R_u$	$\mu^2\Sigma_v \otimes R_u$
$s_{l,m}^{b*}\mathcal{Y}s_{l,m}^{b}$	$\mu^2\lambda_m(R_u)\lvert\lambda_l(A)\rvert^2 \cdot s_l^*\Sigma_v s_l$	$\mu^2\lambda_m(R_u) \cdot s_l^*\Sigma_v s_l$	$\mu^2\lambda_m(R_u) \cdot s_l^*\Sigma_v s_l$

so that $R_{u,k} = R_u$ for all k. In other words, we are dealing with a network of homogeneous agents interacting with each other.

Under Assumption 3.1, the matrices $\mathcal{M}$ from (2.38) and $\mathcal{R}$ from (2.48) simplify to

$$\mathcal{M} = \mu I_{NM} \quad \text{and} \quad \mathcal{R} = I_N \otimes R_u \tag{3.23}$$

and thus the matrices $\mathcal{B}$ and $\mathcal{Y}$ in Table 2.2 reduce to the expressions shown in Table 3.1, where we introduced the diagonal matrix

$$\Sigma_v \triangleq \text{diag}\{\sigma_{v,1}^2, \sigma_{v,2}^2, \ldots, \sigma_{v,N}^2\} > 0. \tag{3.24}$$

Note that the ATC and CTA diffusion strategies now have the same coefficient matrix $\mathcal{B}$ and all three distributed strategies degenerate to non-cooperative strategy (2.9) when $A = I_N$ and $\lambda_l(A) = 1$ for all l. We explain in the sequel the terms that appear in the last row of Table 3.1.

3.2.1 Spectral Properties of $\mathcal{B}$

As mentioned before, the stability and mean-square-error performance of the various algorithms depend on the corresponding matrix $\mathcal{B}$; therefore, we examine more closely the eigen-structure of $\mathcal{B}$. For the distributed strategies (diffusion and consensus), the eigen-structure of $\mathcal{B}$ will depend on the combination matrix A and the covariance matrix R_u. We introduce the eigen-structure of A in Appendix 2.A (see (2.99)). Moreover, we let z_m ($m = 1, 2, \ldots, M$) denote the eigenvector of the covariance matrix R_u that is associated with the eigenvalue $\lambda_m(R_u)$. That is,

$$R_u \cdot z_m = \lambda_m(R_u) \cdot z_m. \tag{3.25}$$

Since R_u is Hermitian and positive-definite, the $\{z_m\}$ are orthonormal, i.e.,

$$z_{m_2}^* z_{m_1} = \delta_{m_1 m_2} \tag{3.26}$$

and the $\{\lambda_m(R_u)\}$ are positive. The following result describes the eigen-structure of the matrix $\mathcal{B}$ in terms of the eigen-structures of $\{A^T, R_u\}$ for the diffusion and consensus algorithms of Table 3.1. Note that the results for any of these distributed strategies collapse to the result for the non-cooperative strategy when we set $\lambda_l(A) = 1$ for all l.

Lemma 3.1 (Eigen-structure of $\mathcal{B}$ under diffusion and consensus)**.** *The matrices $\{\mathcal{B}\}$ appearing in Table 3.1 for the diffusion and consensus strategies have right and left eigenvectors $\{r_{l,m}^b, s_{l,m}^b\}$ given by:*

$$r_{l,m}^b = r_l \otimes z_m \quad \textit{and} \quad s_{l,m}^b = s_l \otimes z_m \tag{3.27}$$

with the corresponding eigenvalues, $\lambda_{l,m}(\mathcal{B})$, shown in Table 3.1 for $l = 1, 2, \ldots, N$ and $m = 1, 2, \ldots, M$. Note that while the eigenvectors are the same for the diffusion and consensuses strategies, the corresponding eigenvalues are different.

Proof. We only consider the diffusion case and denote its coefficient matrix by $\mathcal{B}_{\text{diff}} = A^T \otimes I_M - A^T \otimes \mu R_u$; the same argument applies to the consensus strategy. We multiply $\mathcal{B}_{\text{diff}}$ by the $r_{l,m}^b$ defined in (3.27) from the right and obtain

$$\begin{aligned} \mathcal{B}_{\text{diff}} \cdot r_{l,m}^b &= (A^T \otimes I_M - A^T \otimes \mu R_u) \cdot (r_l \otimes z_m) \\ &= \lambda_l(A) \cdot (r_l \otimes z_m) - \lambda_l(A) \cdot \mu \lambda_m(R_u) \cdot (r_l \otimes z_m) \\ &= \lambda_l(A)(1 - \mu \lambda_m(R_u)) \cdot r_{l,m}^b \end{aligned} \tag{3.28}$$

where we used the Kronecker product property in (2.68). In a similar manner, we can verify that $\mathcal{B}_{\text{diff}}$ has left eigenvector $s_{l,m}^b$ defined in (3.27) with the corresponding eigenvalue $\lambda_{l,m}(\mathcal{B})$ from Table 3.1. □

We then arrive at the following result in comparison of the convergence rate for diffusion and consensus strategies.

Theorem 3.2 (Spectral radius of $\mathcal{B}$ under diffusion and consensus)**.** *It holds that*

$$\rho(\mathcal{B}_{diff}) = \rho(\mathcal{B}_{\text{ncop}}) \leq \rho(\mathcal{B}_{\text{cons}}) \tag{3.29}$$

where equality holds if $A = I_N$ or when the step-size satisfies:

$$0 < \mu \leq \min_{l \neq 1} \frac{1 - |\lambda_l(A)|}{\lambda_{\min}(R_u) + \lambda_{\max}(R_u)}. \tag{3.30}$$

Proof. See Appendix 3.B. □

Note that the upper bound in (3.30) is even smaller than the one in (3.8) and, therefore, can again be very small or even zero. It follows that there is generally a wide range of step-sizes over which $\rho(\mathcal{B}_{\text{cons}})$ is greater than $\rho(\mathcal{B}_{\text{diff}})$. When this happens, the convergence rate of diffusion networks is superior to the convergence rate of consensus networks; in particular, the quantities $\mathbb{E}\tilde{\boldsymbol{w}}_i$ and $\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2$ will converge faster towards their steady-state values over diffusion networks than over consensus networks.

3.2.2 Network MSD Performance

We now compare the network MSD performance. With the knowledge of the eigen-structure of the matrix $\mathcal{B}$, we can use expressions (2.85) and (2.87) to arrive at simple expressions for MSDs. To do so, we introduce the following assumption on the combination matrix.

Assumption 3.2 (Diagonalizability of A)**.** *The combination matrix A is diagonalizable.*

We can now simplify the MSD expressions by using the eigen-decomposition of $\mathcal{B}$ from Lemma 3.1 and the eigen-decomposition of A from Appendix 2.A.

Lemma 3.2 (MSD expressions for homogeneous agents)**.** *Under Assumptions 2.1-3.2, the MSD at agent k from (2.85) can be expressed as:*

$$\text{MSD}_k = \sum_{l_1=1}^{N}\sum_{l_2=1}^{N}\sum_{m=1}^{M} \frac{(e_k^T r_{l_1}) \cdot s_{l_1,m}^{b*} \mathcal{Y} s_{l_2,m}^{b} \cdot (r_{l_2}^* e_k)}{1-\lambda_{l_1,m}(\mathcal{B})\lambda_{l_2,m}^*(\mathcal{B})}. \tag{3.31}$$

Furthermore, if the right eigenvectors $\{r_l\}$ of A^T are approximately orthonormal (or, A is close-to-symmetric), i.e.,

$$r_{l_2}^* r_{l_1} \approx \delta_{l_1 l_2} \tag{3.32}$$

then the network MSD from (2.87) can be approximated by:

$$\text{MSD} \approx \sum_{l=1}^{N}\sum_{m=1}^{M} \frac{s_{l,m}^{b*} \mathcal{Y} s_{l,m}^{b}}{N \cdot (1-|\lambda_{l,m}(\mathcal{B})|^2)}. \tag{3.33}$$

Proof. Using (3.27), we have:

$$\begin{aligned} r_{l_2,m_2}^{b*}(e_k \otimes I_M)(e_k^T \otimes I_M) r_{l_1,m_1}^{b} &= (r_{l_2}^* e_k e_k^T r_{l_1}) \otimes (z_{m_2}^* z_{m_1}) \\ &= (r_{l_2}^* e_k e_k^T r_{l_1}) \cdot \delta_{m_1 m_2} \end{aligned} \tag{3.34}$$

since the eigenvectors $\{z_m\}$ are orthonormal. Substituting (3.34) into (2.85), we arrive at (3.31). For the network MSD, we obtain from (3.27) and assumption (3.32) that

$$\begin{aligned} r^{b*}_{l_2,m_2} r^{b}_{l_1,m_1} &= (r^{*}_{l_2} \otimes z^{*}_{m_2}) \cdot (rl_1 \otimes zm_1) \\ &= (r^{*}_{l_2} rl_1) \cdot (z^{*}_{m_2} zm_1) \\ &\approx \delta_{l_1 l_2} \cdot \delta_{m_1 m_2} \end{aligned} \tag{3.35}$$

Then, from (2.87) and (3.35), we can establish (3.33). □

The term $s^{b*}_{l,m} \mathcal{Y} s^{b}_{l,m}$ in (3.33) is listed in Table 3.1 for the various strategies. Note that, as indicated in (2.103), any symmetric combination matrix A satisfies condition (3.32) with an exact equality.

Using the expressions for $\lambda_{l,m}(\mathcal{B})$ and $s^{b*}_{l,m} \mathcal{Y} s^{b}_{l,m}$ from Table 3.1 and substituting into (3.33), we can obtain the network MSD expressions for the various strategies. The following result shows how these MSD values compare to each other.

Theorem 3.3 (Comparing network MSDs)**.** *If condition (3.32) is satisfied and $A \neq I_N$, then the ATC diffusion strategy achieves the lowest network MSD in comparison to the other strategies (CTA diffusion, consensus, and non-cooperative). More specifically, it holds that*

$$\text{MSD}_{\text{atc}} < \text{MSD}_{\text{cta}} < \text{MSD}_{\text{ncop}} \tag{3.36}$$

$$\text{MSD}_{\text{atc}} < \text{MSD}_{\text{cons}}. \tag{3.37}$$

Furthermore, if $1 \leq \mu \cdot \lambda_{\min}(R_u) < 2$, the consensus strategy is the worst even in comparison to the non-cooperative strategy:

$$\text{MSD}_{\text{atc}} < \text{MSD}_{\text{cta}} < \text{MSD}_{\text{ncop}} < \text{MSD}_{\text{cons}}. \tag{3.38}$$

Proof. See Appendix 3.C. □

Therefore, the ATC diffusion strategy outperforms consensus, CTA diffusion, and non-cooperative strategies when condition (3.32) is satisfied. However, the relation among MSD_{cta}, MSD_{cons}, and MSD_{ncop} depends on the combination matrix A. To illustrate this fact, we reconsider the two-agent network from Section 3.1.1 with $\sigma_{u,1}^2 = \sigma_{u,2}^2 = \sigma_u^2$, $\mu_1 = \mu_2 = \mu$, and $0 < \mu\sigma_u^2 < 1$. Furthermore, to ensure the stability of the consensus strategy and from (3.13), the parameters $\{a, b\}$ in (3.9) are now chosen to satisfy $a + b < 2 - \mu\sigma_u^2$. In this case, the eigenvalues of the combination matrix A in (3.9) are $\{1, 1 - a - b\}$. It can be verified from (3.33) and Table 3.1 that the CTA diffusion strategy achieves lower network MSD (better mean-square performance) than the consensus strategy if

$$\begin{cases} \text{MSD}_{\text{cons}} \leq \text{MSD}_{\text{cta}}, & \text{if } 0 \leq a + b \leq \frac{2(1-\mu\sigma_u^2)}{2-\mu\sigma_u^2} \\ \text{MSD}_{\text{cons}} \geq \text{MSD}_{\text{cta}}, & \text{if } \frac{2(1-\mu\sigma_u^2)}{2-\mu\sigma_u^2} \leq a + b < 2 - \mu\sigma_u^2 \end{cases} \tag{3.39}$$

Similarly, the network MSDs of the consensus and non-cooperative strategies have the following relation:

$$\begin{cases} \text{MSD}_{\text{cons}} \leq \text{MSD}_{\text{ncop}}, & \text{if } 0 \leq a + b \leq 2(1 - \mu\sigma_u^2) \\ \text{MSD}_{\text{cons}} \geq \text{MSD}_{\text{ncop}}, & \text{if } 2(1 - \mu\sigma_u^2) \leq a + b < 2 - \mu\sigma_u^2 \end{cases} \tag{3.40}$$

Combining (3.39)-(3.40), we can divide the $a \times b$ plane into three regions, as shown in Fig. 3.2, where each region corresponds to one possible relation among MSD_{cta}, MSD_{cons}, and MSD_{ncop}.

3.2.3 MSD of Individual Agents

In Theorem 3.3, we established that the ATC diffusion strategy performs the best in terms of the average network MSD. It is still not clear how well the individual

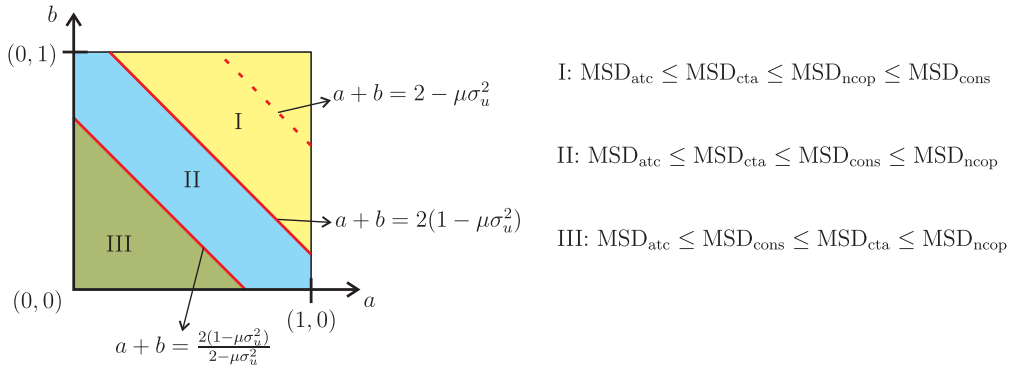


Figure 3.2: Network MSD comparison with $N = 2$ and $\mu\sigma_u^2 = 0.4$. The consensus strategy is unstable when the parameters a and b lie above the dashed line in region I.

agents perform under each strategy. It is generally more challenging to compare diffusion and consensus strategies in terms of the MSDs of their individual agents due to the structure of the matrix $\mathcal{B}$ for the consensus strategy. Nevertheless, this can be accomplished as follows. We observe from (3.31) and Table 3.1 that the $\{\mathrm{MSD}_k\}$ for the CTA diffusion and consensus strategies differ only in the value of $\lambda_{l,m}(\mathcal{B})$. From Table 3.1, the difference between the values of $\lambda_{l,m}(\mathcal{B})$ for these two strategies is

$$\lambda_{l,m}(\mathcal{B}_{\mathrm{cta}}) - \lambda_{l,m}(\mathcal{B}_{\mathrm{cons}}) = \mu\lambda_m(R_u) \cdot (1 - \lambda_l(A)) = \mathcal{O}(\mu) \tag{3.41}$$

where the term $\mathcal{O}(\mu)$ denotes a factor that is of the order of the step-size μ. It follows that for sufficiently small step-sizes (see Assumption 2.1), expression (3.41) is close to zero and the CTA diffusion and consensus strategies will exhibit similar MSDs at the individual agents, i.e., $\mathrm{MSD}_{\mathrm{cta},k} \approx \mathrm{MSD}_{\mathrm{cons},k}$ for all k. As a result, in the following, we only compare $\mathrm{MSD}_{\mathrm{atc},k}$, $\mathrm{MSD}_{\mathrm{cta},k}$, and $\mathrm{MSD}_{\mathrm{ncop},k}$. In particular, we will show that under certain conditions on the combination matrix A, the ATC diffusion strategy continues to perform the best in terms of the MSD

at the individual agents in comparison to the other strategies. To do so, starting from (3.31) and the expressions for $\{\lambda_{l,k}(\mathcal{B}), \mathcal{Y}\}$ in Table 3.1, we can express the MSD at agent k for the ATC diffusion strategy as:

$$\begin{aligned}\mathrm{MSD}_{\mathrm{atc},k} &= \sum_{m=1}^{M} \mu^2 \lambda_m(R_u) \sum_{l_1,l_2=1}^{N} \frac{\lambda_{l_1}(A)\lambda_{l_2}^*(A) \cdot (e_k^T r_{l_1} s_{l_1}^* \Sigma_v s_{l_2} r_{l_2}^* e_k)}{1 - \lambda_{l_1}(A)\lambda_{l_2}^*(A) \cdot (1-\mu\lambda_m(R_u))^2} \\ &\triangleq \sum_{m=1}^{M} \mathrm{MSD}_{\mathrm{atc},k}(m) \end{aligned} \tag{3.42}$$

where we introduced the notation $\mathrm{MSD}_{\mathrm{atc},k}(m)$ to denote the MSD component at agent k that is contributed by the mth eigenvalue of R_u, i.e.,

$$\mathrm{MSD}_{\mathrm{atc},k}(m) = \mu^2 \lambda_m(R_u) \sum_{l_1,l_2=1}^{N} \frac{\lambda_{l_1}(A)\lambda_{l_2}^*(A) \cdot (e_k^T r_{l_1} s_{l_1}^* \Sigma_v s_{l_2} r_{l_2}^* e_k)}{1 - \lambda_{l_1}(A)\lambda_{l_2}^*(A) \cdot (1-\mu\lambda_m(R_u))^2}. \tag{3.43}$$

In a similar vein, we can define the corresponding $\mathrm{MSD}_k(m)$ terms for the other strategies. We list these terms in Table 3.2 in two equivalent forms (we will use the series form later). We first have the following useful preliminary result.

Lemma 3.3 (Useful comparisons)**.** *The following ratios are positive and independent of the agent index k:*

$$\frac{\mathrm{MSD}_{\mathrm{ncop},k}(m) - \mathrm{MSD}_{\mathrm{atc},k}(m)}{\mathrm{MSD}_{\mathrm{ncop},k}(m) - \mathrm{MSD}_{\mathrm{cta},k}(m)} = \frac{1}{(1-\mu\lambda_m(R_u))^2} > 0 \tag{3.44}$$

$$\frac{\mathrm{MSD}_{\mathrm{ncop},k}(m) - \mathrm{MSD}_{\mathrm{atc},k}(m)}{\mathrm{MSD}_{\mathrm{cta},k}(m) - \mathrm{MSD}_{\mathrm{atc},k}(m)} = \frac{1}{1-(1-\mu\lambda_m(R_u))^2} > 0. \tag{3.45}$$

Proof. From the eigen-forms of $\{\mathrm{MSD}_k(m)\}$ in Table 3.2, the differences between $\mathrm{MSD}_{\mathrm{atc},k}(m)$, $\mathrm{MSD}_{\mathrm{cta},k}(m)$, and $\mathrm{MSD}_{\mathrm{ncop},k}(m)$ are given by:

$$\mathrm{MSD}_{\mathrm{ncop},k}(m) - \mathrm{MSD}_{\mathrm{atc},k}(m) = \frac{\mu^2\lambda_m(R_u)}{1-(1-\mu\lambda_m(R_u))^2} \cdot c_k(m) \tag{3.46}$$

$$\mathrm{MSD}_{\mathrm{ncop},k}(m) - \mathrm{MSD}_{\mathrm{cta},k}(m) = \frac{\mu^2\lambda_m(R_u) \cdot (1-\mu\lambda_m(R_u))^2}{1-(1-\mu\lambda_m(R_u))^2} \cdot c_k(m) \tag{3.47}$$

$$\mathrm{MSD}_{\mathrm{cta},k}(m) - \mathrm{MSD}_{\mathrm{atc},k}(m) = \mu^2\lambda_m(R_u) \cdot c_k(m) \tag{3.48}$$

Table 3.2: Expressions for $\text{MSD}_k(m)$ in series form and eigen-form.

ATC Diffusion (2.29)	Series form	$\mu^2\lambda_m(R_u)\sum_{j=0}^{\infty}(1-\mu\lambda_m(R_u))^{2j}\cdot e_k^T A^{T(j+1)}\Sigma_v A^{j+1}e_k$
	Eigen-form	$\mu^2\lambda_m(R_u)\sum_{l_1,l_2=1}^{N}\frac{\lambda_{l_1}(A)\lambda_{l_2}^*(A)\cdot(e_k^T r_{l_1}s_{l_1}^*\Sigma_v s_{l_2}r_{l_2}^* e_k)}{1-\lambda_{l_1}(A)\lambda_{l_2}^*(A)\cdot(1-\mu\lambda_m(R_u))^2}$
CTA Diffusion (2.32)	Series form	$\mu^2\lambda_m(R_u)\sum_{j=0}^{\infty}(1-\mu\lambda_m(R_u))^{2j}\cdot e_k^T A^{Tj}\Sigma_v A^j e_k$
	Eigen-form	$\mu^2\lambda_m(R_u)\sum_{l_1,l_2=1}^{N}\frac{e_k^T r_{l_1}s_{l_1}^*\Sigma_v s_{l_2}r_{l_2}^* e_k}{1-\lambda_{l_1}(A)\lambda_{l_2}^*(A)\cdot(1-\mu\lambda_m(R_u))^2}$
Non-cooperative (2.9)	Series form	$\mu^2\lambda_m(R_u)\sum_{j=0}^{\infty}(1-\mu\lambda_m(R_u))^{2j}\cdot e_k^T\Sigma_v e_k$
	Eigen-form	$\mu^2\lambda_m(R_u)\sum_{l_1,l_2=1}^{N}\frac{e_k^T r_{l_1}s_{l_1}^*\Sigma_v s_{l_2}r_{l_2}^* e_k}{1-(1-\mu\lambda_m(R_u))^2}$

where

$$c_k(m)=\sum_{l_1,l_2=1}^{N}\frac{\big[1-\lambda_{l_1}(A)\lambda_{l_2}^*(A)\big]\cdot(e_k^T r_{l_1}s_{l_1}^*\Sigma_v s_{l_2}r_{l_2}^* e_k)}{1-\lambda_{l_1}(A)\lambda_{l_2}^*(A)\cdot(1-\mu\lambda_m(R_u))^2}. \tag{3.49}$$

Then, dividing (3.46) by (3.47) and (3.46) by (3.48), we arrive at (3.44)-(3.45). □

Lemma 3.4 (Useful ordering)**.** *The relation among* $\text{MSD}_{\text{atc},k}(m)$, $\text{MSD}_{\text{cta},k}(m)$, *and* $\text{MSD}_{\text{ncop},k}(m)$ *is either*

$$\text{MSD}_{\text{atc},k}(m)\le\text{MSD}_{\text{cta},k}(m)\le\text{MSD}_{\text{ncop},k}(m) \tag{3.50}$$

or

$$\text{MSD}_{\text{atc},k}(m)\ge\text{MSD}_{\text{cta},k}(m)\ge\text{MSD}_{\text{ncop},k}(m). \tag{3.51}$$

Proof. Assume first that $\text{MSD}_{\text{atc},k}(m)\le\text{MSD}_{\text{ncop},k}(m)$. Then, using (3.44), we get $\text{MSD}_{\text{ncop},k}(m)-\text{MSD}_{\text{cta},k}(m)\ge 0$. Similarly, from (3.45), we get $\text{MSD}_{\text{cta},k}(m)-$

$\text{MSD}_{\text{atc},k}(m) \geq 0$. We conclude that relation (3.50) holds in this case. Assume instead that $\text{MSD}_{\text{atc},k}(m) \geq \text{MSD}_{\text{ncop},k}(m)$. Then, a similar argument will show that (3.51) should hold. □

The above result is useful since it allows us to deduce the relation among $\text{MSD}_{\text{atc},k}(m)$, $\text{MSD}_{\text{cta},k}(m)$, and $\text{MSD}_{\text{ncop},k}(m)$ by only knowing the relation between any two of them. To proceed, we note that we can alternatively express the $\text{MSD}_k(m)$ terms in an equivalent series form. For example, expression (3.43) can be written as:

$$\begin{aligned}
\text{MSD}_{\text{atc},k}(m) &= \mu^2 \lambda_m(R_u) \sum_{j=0}^{\infty} \sum_{l_1,l_2=1}^{N} (1-\mu\lambda_m(R_u))^{2j} \\
&\quad \times \lambda_{l_1}^{j+1}(A) \cdot \lambda_{l_2}^{*(j+1)}(A) \cdot (e_k^T r_{l_1} s_{l_1}^* \Sigma_v s_{l_2} r_{l_2}^* e_k) \\
&= \mu^2 \lambda_m(R_u) \sum_{j=0}^{\infty} (1-\mu\lambda_m(R_u))^{2j} \\
&\quad \times e_k^T \left(\sum_{l_1=1}^{N} \lambda_{l_1}^{j+1}(A) r_{l_1} s_{l_1}^* \right) \Sigma_v \left(\sum_{l_2=1}^{N} \lambda_{l_2}^{*(j+1)}(A) s_{l_2} r_{l_2}^* \right) e_k \\
&= \mu^2 \lambda_m(R_u) \sum_{j=0}^{\infty} (1-\mu\lambda_m(R_u))^{2j} \cdot e_k^T A^{T(j+1)} \Sigma_v A^{j+1} e_k. \qquad (3.52)
\end{aligned}$$

In a similar manner, we can obtain the corresponding $\text{MSD}_k(m)$ series forms for the other strategies and we list these in Table 3.2. In the following, we provide conditions to guarantee that the individual agent performance in the ATC diffusion strategy outperforms the other strategies.

Theorem 3.4 (Comparing individual MSDs)**.** *If the combination matrix A satisfies*

$$\Sigma_v - A^T \Sigma_v A \geq 0 \qquad (3.53)$$

where Σ_v is the noise variance (diagonal) matrix defined by (3.24), then:

$$\text{MSD}_{\text{atc},k} \leq \text{MSD}_{\text{cta},k} \leq \text{MSD}_{\text{ncop},k}. \qquad (3.54)$$

Proof. From the series forms of $\{\text{MSD}_k(m)\}$ in Table 3.2, the difference $\text{MSD}_{\text{cta},k}(m) - \text{MSD}_{\text{atc},k}(m)$ is given by:

$$\begin{aligned}&\text{MSD}_{\text{cta},k}(m) - \text{MSD}_{\text{atc},k}(m)\\&= \mu^2\lambda_m(R_u)\sum_{j=0}^{\infty}(1-\mu\lambda_m(R_u))^{2j}e_k^T A^{Tj}\left(\Sigma_v - A^T\Sigma_v A\right)A^j e_k. \qquad (3.55)\end{aligned}$$

Since $\Sigma_v - A^T\Sigma_v A \geq 0$, we conclude that $\text{MSD}_{\text{cta},k}(m) \geq \text{MSD}_{\text{atc},k}(m)$ for all m. Then, applying Lemma 3.4, we obtain relation (3.54). □

Condition (3.53) essentially means that the combination matrix A should not magnify the noise effect across the network. However, in general, condition (3.53) is restrictive in the sense that over the set of feasible diagonalizable left-stochastic matrices A satisfying $a_{l,k} = 0$ if $l \notin \mathcal{N}_k$, the set of combination matrices A satisfying (3.53) can be small. We illustrate this situation by reconsidering the two-agent network (3.9) for which

$$\Sigma_v - A^T\Sigma_v A = \begin{bmatrix} 2at - a^2(1+t) & -(1-a)bt - a(1-b) \\ -(1-a)bt - a(1-b) & 2b - b^2(1+t) \end{bmatrix} \qquad (3.56)$$

where $t = \sigma_{v,1}^2/\sigma_{v,2}^2$ denotes the ratio of noise variances at agents 1 and 2. Note from

$$\det(\Sigma_v - A^T\Sigma_v A) = -(a-bt)^2 \leq 0 \qquad (3.57)$$

that equality holds in (3.57) if, and only if,

$$a = tb. \qquad (3.58)$$

That is, when $a \neq tb$, the matrix $(\Sigma_v - A^T\Sigma_v A)$ has two eigenvalues with different signs. Thus, the only way to ensure $\Sigma_v - A^T\Sigma_v A \geq 0$ in this case is to set $a = tb$ and, thus, the matrix $(\Sigma_v - A^T\Sigma_v A)$ will have at least one eigenvalue at zero since

its determinant will be zero. To ensure $\Sigma_v - A^T\Sigma_v A \geq 0$, its other eigenvalue, which is equal to

$$b(1+t^2)(2-b-bt) \tag{3.59}$$

needs to be greater than or equal to zero. It follows that b must satisfy:

$$0 \leq b \leq \frac{2}{1+t}. \tag{3.60}$$

Moreover, since a and b must lie within the interval $[0,1]$, we conclude from (3.58) that b must also satisfy:

$$0 \leq b \leq \min\{1, 1/t\}. \tag{3.61}$$

It can be verified that condition (3.61) implies condition (3.60) since $\min\{1,1/t\} \leq 2/(1+t)$. That is, for any left-stochastic matrix A from (3.9) satisfying $a = tb$ and (3.61), relation (3.54) holds and both agents improve their own MSDs by employing the diffusion strategies. Note that condition (3.58) represents a line segment in the unit square $a, b \in [0,1]$ (see Fig. 3.3). In the following, we relax condition (3.53) with a mild constraint on the network topology.

In addition to Assumption 3.2, we further assume that the combination matrix A is primitive (i.e., Assumption 2.2). It follows from the Perron-Frobenius Theorem [64] that $(A^T)^j$ converges to the rank-one matrix:

$$\lim_{j\to\infty} (A^T)^j = r_1 s_1^T. \tag{3.62}$$

where r_1 and s_1 satisfy (2.102). Then, we arrive at the following result.

Theorem 3.5 (Comparing individual MSDs for primitive A)**.** *Under Assumptions 3.2 and 2.2, if*

$$\frac{s_1^T \Sigma_v s_1}{N} < \sigma_{v,k}^2 \tag{3.63}$$

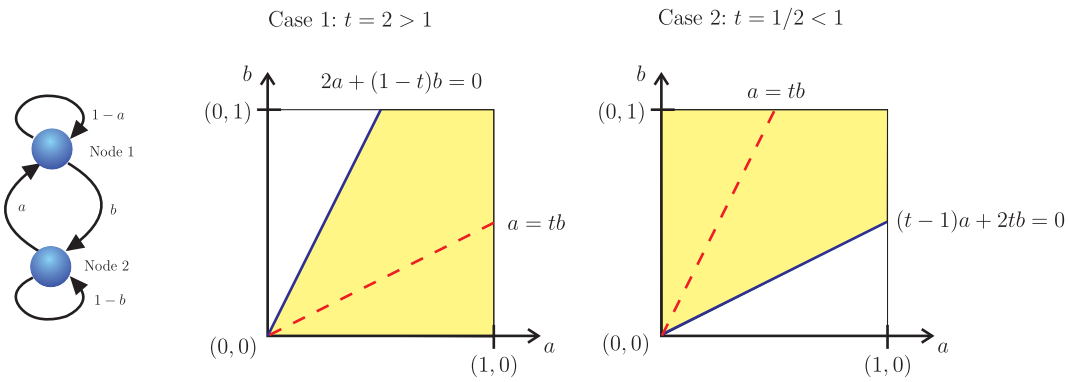


Figure 3.3: Comparison of individual agent MSD using $N = 2$ and $t = \sigma_{v,1}^2/\sigma_{v,2}^2$. There exists a step-size region such that $\text{MSD}_{\text{atc},k} < \text{MSD}_{\text{cta},k} < \text{MSD}_{\text{ncop},k}$ for $k = 1, 2$ when the parameters a and b lie in the shaded regions. The dashed lines indicate condition (3.58).

for all k, then there exists $\mu^\circ > 0$ so that for any step-size μ satisfying $0 < \mu \leq \mu^\circ$, it holds:

$$\text{MSD}_{\text{atc},k} < \text{MSD}_{\text{cta},k} < \text{MSD}_{\text{ncop},k}. \tag{3.64}$$

Proof. See Appendix 3.D. □

We show in Appendix 3.E that for any primitive A, condition (3.53) implies condition (3.63). To illustrate these two conditions, we consider again the two-agent network. It can be verified that s_1^T for A^T in (3.9) has the form

$$s_1^T = \begin{bmatrix} \sqrt{2}b/(a+b) & \sqrt{2}a/(a+b) \end{bmatrix}. \tag{3.65}$$

Then, some algebra shows that condition (3.63) becomes

$$(t-1)a + 2bt > 0 \quad \text{and} \quad 2a + (1-t)b > 0. \tag{3.66}$$

Recall that $t = \sigma_{v,1}^2/\sigma_{v,2}^2$. We illustrate condition (3.66), along with condition (3.58), in Fig. 3.3. We observe that condition (3.58), shown as the dashed lines,

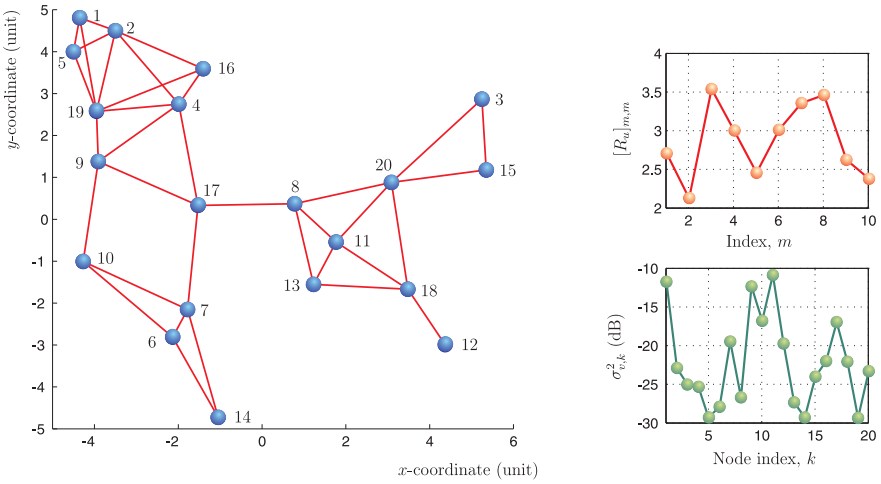


Figure 3.4: Network topology and noise and data power profiles at the agents. The number next to an agent denotes the agent index.

is contained in condition (3.66), shown as the shaded regions, and that compared to condition (3.58), condition (3.66) enlarges the region of A for which the ATC diffusion strategy performs the best in terms of the individual MSD performance.

3.3 Simulation Results

We consider a network with 20 homogeneous agents and random topology. The regression covariance matrix R_u is diagonal with entries randomly generated from $[2, 4]$, and the noise variances $\{\sigma^2_{v,k}\}$ are randomly generated over $[-30, -10]$ dB (see Fig. 3.4). The network estimates a 10×1 (i.e., $M = 10$) unknown vector w°.

The transient network MSD over time is shown on the left hand side of Fig. 3.5 with three possible combination rules: relative-variance [124, 152, 174], uniform from (4.64), and Metropolis [167] (see Table 3.3). Note that the matrix A for

Table 3.3: Combination rules used in the simulations

Name	**Rule**
Relative-variance [124, 152, 174]	$a_{l,k} = \begin{cases} \sigma_{v,l}^{-2} / \sum_{j \in \mathcal{N}_k} \sigma_{v,j}^{-2}, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}$
Uniform [124]	$a_{l,k} = \begin{cases} 1/n_k, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}$
Metropolis [167]	$a_{l,k} = \begin{cases} 1 - \sum_{j \neq k} a_{k,j}, & \text{if } l = k \\ 1/\max\{n_k, n_l\}, & \text{if } l \in \mathcal{N}_k \setminus \{k\} \\ 0, & \text{otherwise} \end{cases}$
No-cooperation	$a_{l,k} = \delta_{kl}$

the Metropolis rule is symmetric. The step-size μ is set to $\mu = 0.02$. We observe that, as expected, the ATC diffusion strategy outperforms the other strategies, especially for the relative-variance rule. It also suggests that some conventional choices of combination weights, such as the Metropolis rule, may not be the most suitable for adaptation in the presence of both noisy and streaming data because such weights do not take into account the noise profile across the agents (see, e.g., [124, 152, 174] for more details on this issue). We further show the steady-state MSD at the individual agents on the right hand side of Fig. 3.5. We observe that the ATC diffusion strategy achieves the lowest MSD at each agent in comparison to the other strategies. These observations are in agreement with the results predicted by the theoretical analysis. The theoretical expressions for MSDs from (3.31) and (3.33) are also depicted in Fig. 3.5 for the ATC diffusion strategy and match well with simulations.

We further compare the mean-square performance of the distributed strategies

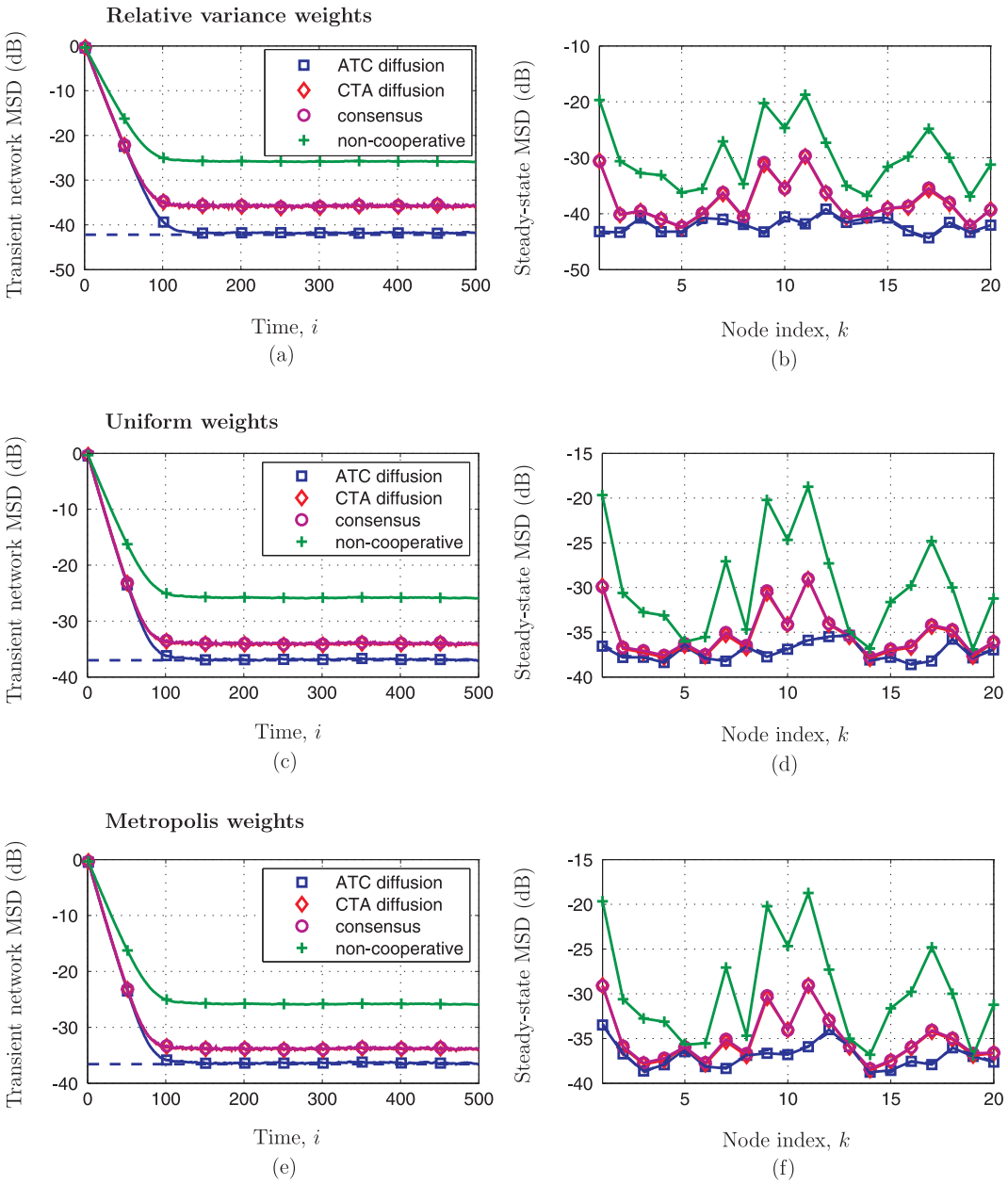


Figure 3.5: Transient network MSD over time (left, with peak values normalized to 0dB) and steady-state MSD at the individual agents (right) for (a)-(b) the relative-variance, (c)-(d) uniform, and (e)-(f) Metropolis rules. The dashed lines on the left/right hand side indicate the theoretical network/individual MSD from (3.33)/(3.31) for the ATC diffusion strategy.

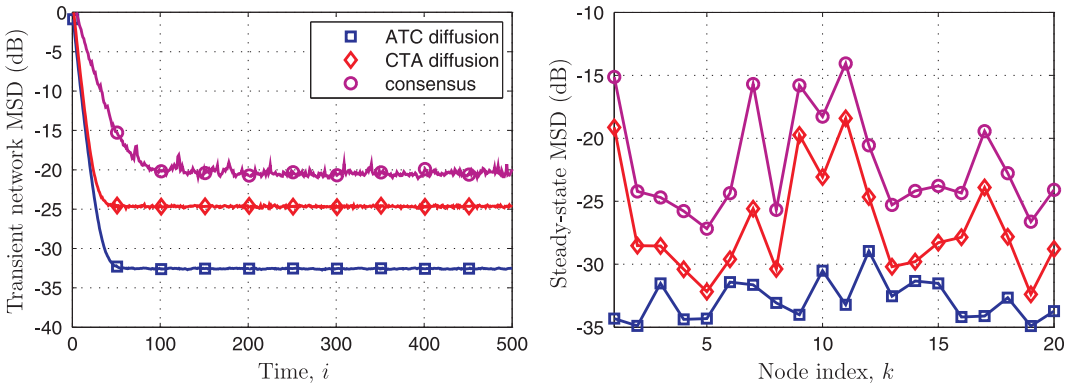


Figure 3.6: Transient network MSD over time (left) and steady-state MSD at the individual agents (right) for the relative-variance combination rule using $\mu = 0.075$.

for larger step-sizes. We set the step-size to $\mu = 0.075$ and use the relative-variance combination rule. The transient network MSD over time is shown on the left hand side of Fig. 3.6. We observe that the ATC and CTA diffusion strategies have the same convergence rate and converge faster than the consensus strategy. Moreover, the diffusion strategies achieve lower network MSD than the consensus strategy. We also show the steady-state MSD at the individual agents on the right hand side of Fig. 3.6. We see again that ATC diffusion performs the best in comparison to the other strategies at each individual agent.

3.4 Concluding Remarks

We compared analytically several cooperative estimation strategies, including ATC diffusion, CTA diffusion, and consensus for distributed estimation over networks. The results show that diffusion networks are more stable than consensus networks. Moreover, the stability of diffusion networks is independent of the

combination weights, whereas consensus networks can become unstable even if all individual agents are stable. Furthermore, in steady-state, the ATC diffusion strategy performs the best not only in terms of the network MSD, but also in terms of the MSDs at the individual agents.

3.A Proof of Theorem 3.1

First, note that the matrices $\{\mathcal{B}\}$ for the ATC and CTA diffusion strategies given by (3.5) have the same eigenvalues (and, therefore, the same spectral radius) because for any matrices X and Y of compatible dimensions, the matrix products XY and YX have the same eigenvalues [64]. So let us evaluate the spectral radius of $\mathcal{B}_{\rm atc}$. To do so, we introduce a convenient block matrix norm, and denote it by $\|\cdot\|_b$; it is defined as follows. Let $\mathcal{X}$ be an $N \times N$ block matrix with blocks of size $M \times M$ each. Its block matrix norm is defined as:

$$\|\mathcal{X}\|_b \triangleq \max_{1\le k\le N} \left(\sum_{l=1}^{N} \|\mathcal{X}_{k,l}\|_2 \right) \tag{3.67}$$

where $\mathcal{X}_{k,l}$ denotes the (k,l)th block of $\mathcal{X}$ and $\|\cdot\|_2$ denotes the 2-induced norm (largest singular value) of its matrix argument. That is, we first compute the 2-induced norm of each block $\Sigma_{k,l}$ and then find the ∞-norm of the $N \times N$ matrix formed from the entries $\{\|\Sigma_{k,l}\|_2\}$. It can be verified that (3.67) satisfies the four conditions for a matrix norm [64]. Now, since $\{I_{NM}, \mathcal{M}, \mathcal{R}\}$ are block diagonal matrices, the following property holds:

$$\begin{aligned} \|I_{NM} - \mathcal{M}\mathcal{R}\|_b &= \max_{1\le k\le N} \|I_M - \mu_k R_{u,k}\|_2 \\ &= \max_{1\le k\le N} \rho(I_M - \mu_k R_{u,k}) \\ &= \rho(\mathcal{B}_{\rm ncop}) \end{aligned} \tag{3.68}$$

where we used the fact that the 2-induced norm of any Hermitian matrix coincides with its spectral radius. In addition, since A is a left-stochastic matrix, it holds that

$$\begin{aligned}\|\mathcal{A}^T\|_b &= \max_{1\le k\le N}\left(\sum_{l=1}^{N}\|a_{l,k}I_M\|_2\right)\\ &= \max_{1\le k\le N}\left(\sum_{l=1}^{N}a_{l,k}\right)\\ &= 1. \end{aligned} \tag{3.69}$$

Accordingly, using the fact that the spectral radius of a matrix is upper bounded by any norm of the matrix [64], we get:

$$\begin{aligned}\rho(\mathcal{B}_{\rm atc}) &\le \|\mathcal{A}^T(I_{NM}-\mathcal{M}\mathcal{R})\|_b\\ &\le \|\mathcal{A}^T\|_b\cdot\|I_{NM}-\mathcal{M}\mathcal{R}\|_b\\ &= \rho(\mathcal{B}_{\rm ncop})\end{aligned} \tag{3.70}$$

which establishes (3.6).

Now, assume A is symmetric. Since it is also left-stochastic, it follows that its eigenvalues are real and lie inside the interval $[-1,1]$. Therefore, $(I_{NM}-\mathcal{A}^T)$ is nonnegative-definite. Moreover, since $\mathcal{M}$ and $\mathcal{R}$ commute, i.e., $\mathcal{R}\mathcal{M}=\mathcal{M}\mathcal{R}$, it can be verified that $\mathcal{B}_{\rm cons}$ in (3.4) and $\mathcal{B}_{\rm ncop}$ in (3.1) are Hermitian. In addition, the matrices $\mathcal{B}_{\rm cons}$ and $\mathcal{B}_{\rm ncop}$ are related as follows:

$$\mathcal{B}_{\rm ncop}=\mathcal{B}_{\rm cons}+(I_{NM}-\mathcal{A}^T) \tag{3.71}$$

with $(I_{NM}-\mathcal{A}^T)\ge 0$. Using Weyl's Theorem[1] [64], we arrive at (3.7). Following

[1]Let $\{D',D,\Delta D\}$ be $M\times M$ Hermitian matrices with ordered eigenvalues $\{\lambda_m(D'),\lambda_m(D),\lambda_m(\Delta D)\}$, i.e., $\lambda_1(D)\ge\lambda_2(D)\ge\ldots\ge\lambda_M(D)$, and likewise for the eigenvalues of $\{D',\Delta D\}$. Weyl's Theorem states that if $D'=D+\Delta D$, then

$$\lambda_m(D)+\lambda_M(\Delta D)\le\lambda_m(D')\le\lambda_m(D)+\lambda_1(\Delta D)$$

for $1\le m\le M$. When $\Delta D\ge 0$, it holds that $\lambda_m(D')\ge\lambda_m(D)$.

a similar argument, it holds for symmetric A that

$$\lambda_l\left\{\lambda_{\min}(A)\cdot I_{NM}-\mathcal{M}\mathcal{R}\right\}\leq\lambda_l(\mathcal{B}_{\rm cons})\quad\text{for } l=1,2,\ldots,NM. \tag{3.72}$$

Thus, the matrix $\mathcal{B}_{\rm cons}$ is stable (namely, $-1<\lambda_l(\mathcal{B}_{\rm cons})<1$ for $l=1,2,\ldots,NM$) if

$$\lambda_l\left(\lambda_{\min}(A)\cdot I_{NM}-\mathcal{M}\mathcal{R}\right)>-1 \tag{3.73}$$

$$\lambda_l(\mathcal{B}_{\rm ncop})<1 \tag{3.74}$$

for $l=1,2,\ldots,NM$, or, equivalently,

$$\lambda_{\min}(A)-\mu_k\lambda_m(R_{u,k})>-1 \tag{3.75}$$

$$1-\mu_k\lambda_m(R_{u,k})<1 \tag{3.76}$$

for $k=1,2,\ldots,N$ and $m=1,2,\ldots,M$. We then arrive at (3.8).

3.B Proof of Theorem 3.2

For the diffusion strategies, from Table 3.1 and since $\rho(A)=1$, we have

$$\begin{aligned}\rho(\mathcal{B}_{\rm diff})&=\rho[A^T\otimes(I_M-\mu R_u)]\\&=\rho(A)\cdot\rho(I_M-\mu R_u)\\&=\rho(I_M-\mu R_u)\\&=\rho(\mathcal{B}_{\rm ncop}).\end{aligned} \tag{3.77}$$

Moreover, since $1\in\{\lambda_l(A)\}$, we have

$$\begin{aligned}\rho(\mathcal{B}_{\rm ncop})&=\max_{1\leq m\leq M}|1-\mu\lambda_m(R_u)|\\&\leq\max_{1\leq l\leq N}\max_{1\leq m\leq M}|\lambda_l(A)-\mu\lambda_m(R_u)|\\&=\rho(\mathcal{B}_{\rm cons})\end{aligned} \tag{3.78}$$

and we arrive at (3.29). It is obvious that when $A = I_N$, then equality in (3.78) holds and $\rho(\mathcal{B}_{\text{ncop}}) = \rho(\mathcal{B}_{\text{cons}})$. We now consider the case when $A \neq I_N$. Note that the spectral radius of $\mathcal{B}_{\text{ncop}}$ is given by

$$\rho(\mathcal{B}_{\text{ncop}}) = \max\{1 - \mu\lambda_{\min}(R_u), -1 + \mu\lambda_{\max}(R_u)\}. \tag{3.79}$$

We first verify that equality in (3.78) holds only when $\rho(\mathcal{B}_{\text{ncop}}) = 1 - \mu\lambda_{\min}(R_u)$. Indeed, if

$$\rho(\mathcal{B}_{\text{ncop}}) = -1 + \mu\lambda_{\max}(R_u) \geq 0 \tag{3.80}$$

we have that $\mu\lambda_{\max}(R_u) \geq 1$ and we get from (3.78) that

$$\begin{aligned}
\rho(\mathcal{B}_{\text{cons}}) &= \max_{1\leq l\leq N} \max_{1\leq m\leq M} |\lambda_l(A) - \mu\lambda_m(R_u)| \\
&\geq |\lambda_l(A) - \mu\lambda_{\max}(R_u)| \\
&\geq |\text{Re}\{\lambda_l(A)\} - \mu\lambda_{\max}(R_u)| \\
&= -\text{Re}\{\lambda_l(A)\} + \mu\lambda_{\max}(R_u)
\end{aligned} \tag{3.81}$$

since $\text{Re}\{\lambda_l(A)\} \leq 1$ where $\text{Re}\{\cdot\}$ denotes the real part of its argument. Since $A \neq I_N$, there exists some l such that $\text{Re}\{\lambda_l(A)\} < 1$ and then

$$\rho(\mathcal{B}_{\text{cons}}) > -1 + \mu\lambda_{\max}(R_u) = \rho(\mathcal{B}_{\text{ncop}}). \tag{3.82}$$

Now, assume that

$$\rho(\mathcal{B}_{\text{ncop}}) = 1 - \mu\lambda_{\min}(R_u). \tag{3.83}$$

Then, equality in (3.78) holds if

$$|\lambda_l(A) - \mu\lambda_m(R_u)| \leq \rho(\mathcal{B}_{\text{ncop}}) \tag{3.84}$$

for all l and m. It is obvious that relation (3.84) holds for $l = 1$ since $\lambda_1(A) = 1$ and

$$\begin{aligned}
\rho(\mathcal{B}_{\text{ncop}}) &= \max_{1\leq m\leq M} |1 - \mu\lambda_m(R_u)| \\
&\geq |\lambda_1(A) - \mu\lambda_m(R_u)|.
\end{aligned} \tag{3.85}$$

For $l = 2, 3, \ldots, N$, by the triangular inequality of norms, we have that

$$|\lambda_l(A) - \mu\lambda_m(R_u)| \leq |\lambda_l(A)| + \mu\lambda_{\max}(R_u). \tag{3.86}$$

Hence, the inequality in (3.84) holds if

$$|\lambda_l(A)| + \mu\lambda_{\max}(R_u) \leq 1 - \mu\lambda_{\min}(R_u) \tag{3.87}$$

for $l = 2, 3, \ldots, N$ and we arrive at (3.30).

3.C Proof of Theorem 3.3

We first verify that $\text{MSD}_{\text{atc}} < \text{MSD}_{\text{cta}}$, $\text{MSD}_{\text{cta}} < \text{MSD}_{\text{ncop}}$, and $\text{MSD}_{\text{atc}} < \text{MSD}_{\text{cons}}$. We show the result by verifying that the individual terms on the right hand side of (3.33) for the various strategies have the same ordering. That is, from (3.33) and Table 3.1, we verify that the following ratios, which correspond to $\text{MSD}_{\text{atc}} < \text{MSD}_{\text{cta}}$, $\text{MSD}_{\text{cta}} < \text{MSD}_{\text{ncop}}$, and $\text{MSD}_{\text{atc}} < \text{MSD}_{\text{cons}}$, respectively, are upper bounded by one:

$$|\lambda_l(A)|^2 \leq 1 \tag{3.88}$$

$$\frac{1 - (1 - \mu\lambda_m(R_u))^2}{1 - |\lambda_l(A)|^2 \cdot (1 - \mu\lambda_m(R_u))^2} \leq 1 \tag{3.89}$$

$$\frac{|\lambda_l(A)|^2 \left(1 - |\lambda_l(A) - \mu\lambda_m(R_u)|^2\right)}{1 - |\lambda_l(A)|^2 \cdot (1 - \mu\lambda_m(R_u))^2} \leq 1 \tag{3.90}$$

for all l and m. Note that relations (3.88)-(3.89) hold since $|\lambda_l(A)| \leq 1$ for all l in view of the fact that A is left-stochastic and, hence, $\rho(A) = 1$. On the other hand, relation (3.90) would hold if, and only if,

$$|\lambda_l(A)|^2 \left[1 + (1 - \mu\lambda_m(R_u))^2 - |\lambda_l(A) - \mu\lambda_m(R_u)|^2\right] \leq 1. \tag{3.91}$$

To establish that (3.91) is true for all l and m, we introduce the compact notation

$$\lambda = \lambda_l(A) \quad \text{and} \quad \delta = \mu\lambda_m(R_u) \tag{3.92}$$

and consider the following function of two variables:

$$\begin{aligned} f(\lambda,\delta) &\triangleq |\lambda|^2\left[1+(1-\delta)^2-|\lambda-\delta|^2\right] \\ &\text{with } |\lambda|\leq 1,\ \delta\in(0,2), \text{ and } |\lambda-\delta|<1. \end{aligned} \tag{3.93}$$

The range for δ ensures condition (3.3) and the stability of the diffusion network, while the range for $|\lambda-\delta|$ ensures that the consensus network is stable, i.e., $|\lambda_{l,m}(\mathcal{B}_{\text{cons}})|<1$ for all l and m. Then, we would like to show that $f(\lambda,\delta)\leq 1$. Since λ is generally complex-valued, we denote the real part of λ by λ_r. Then, the term $|\lambda-\delta|^2$ in (3.93) is given by

$$|\lambda-\delta|^2=|\lambda|^2+\delta^2-2\lambda_r\delta \tag{3.94}$$

and $f(\lambda,\delta)$ from (3.93) becomes

$$f(\lambda,\delta)=-|\lambda|^4+2(1-\delta+\lambda_r\delta)|\lambda|^2. \tag{3.95}$$

Since $f(\lambda,\delta)$ is linear in δ, the maximum value of $f(\lambda,\delta)$ in (3.95) over δ occurs at the end points of δ. Since $\delta\in(0,2)$ and $|\lambda_r-\delta|\leq|\lambda-\delta|<1$, we conclude that

$$0<\delta<1+\lambda_r. \tag{3.96}$$

Substituting the end points of δ into (3.95), we have

$$f(\lambda,0)=-(|\lambda|^2-1)^2+1\leq 1 \tag{3.97}$$

$$f(\lambda,1+\lambda_r)=-|\lambda|^4+2\lambda_r^2|\lambda|^2\leq|\lambda|^4\leq 1 \tag{3.98}$$

where we used the fact that $\lambda_r^2\leq|\lambda|^2$ and $|\lambda|\leq 1$. Note that when $A\neq I_N$, there is at least one eigenvalue of A (say, λ_l) such that

$$|\lambda_l(A)|<1 \tag{3.99}$$

Therefore, strict inequality in (3.88)-(3.90) holds. We therefore established (3.36) and (3.37).

Let us now examine what happens when the step-size is such that $1 \leq \mu\lambda_{\min}(R_u) < 2$. Again, from (3.33) and Table 3.1, we establish that $\text{MSD}_{\text{ncop}} < \text{MSD}_{\text{cons}}$ this conclusion by showing that the ratio of the individual terms appearing in the sums (3.33) is upper bounded by one:

$$\frac{1 - |\lambda_l(A) - \mu\lambda_m(R_u)|^2}{1 - (1 - \mu\lambda_m(R_u))^2} \leq 1 \tag{3.100}$$

for all l and m. Condition (3.100) is equivalent to showing that

$$|\lambda_l(A) - \mu\lambda_m(R_u)|^2 - (1 - \mu\lambda_m(R_u))^2 = |\lambda|^2 - 2\lambda_r\delta - (1 - 2\delta) \geq 0 \tag{3.101}$$

where we used the notation from (3.93). Relation (3.101) holds since

$$\delta \geq \mu\lambda_{\min}(R_u) \geq 1 \geq |\lambda| \geq |\lambda_r| \tag{3.102}$$

and then

$$|\lambda|^2 - 2\lambda_r\delta - (1 - 2\delta) \geq \lambda_r^2 - 2\lambda_r\delta - (1 - 2\delta) \tag{3.103}$$

$$= (1 - \lambda_r)(2\delta - 1 - \lambda_r) \tag{3.104}$$

$$\geq 0. \tag{3.105}$$

Again, when $A \neq I_N$, strict inequality in (3.100) holds for some l.

3.D Proof of Theorem 3.5

From the series forms of $\{\text{MSD}_k(m)\}$ in Table 3.2, the difference between $\text{MSD}_{\text{cta},k}(m)$ and $\text{MSD}_{\text{ncop},k}(m)$ can be expressed as:

$$\begin{aligned}&\text{MSD}_{\text{ncop},k}(m) - \text{MSD}_{\text{cta},k}(m) \\ &= \mu^2\lambda_m(R_u)\sum_{j=0}^{\infty}(1 - \mu\lambda_m(R_u))^{2j} e_k^T \left(\Sigma_v - A^{Tj}\Sigma_v A^j\right) e_k.\end{aligned} \tag{3.106}$$

From (3.62), we have

$$\lim_{j\to\infty} e_k^T \left(\Sigma_v - A^{Tj}\Sigma_v A^j\right) e_k = \sigma_{v,k}^2 - e_k^T r_1 s_1^T \Sigma_v s_1 r_1^T e_k. \tag{3.107}$$

Therefore, there exists an integer J_m such that for any $\varepsilon > 0$,

$$e_k^T \left(\Sigma_v - A^{Tj}\Sigma_v A^j\right) e_k \geq \sigma_{v,k}^2 - e_k^T r_1 s_1^T \Sigma_v s_1 r_1^T e_k - \varepsilon \triangleq \Delta \tag{3.108}$$

for all $j \geq J_m$. From (2.102), Δ in (3.108) becomes

$$\Delta = \sigma_{v,k}^2 - s_1^T \Sigma_v s_1 / N - \varepsilon. \tag{3.109}$$

From condition (3.63), we are able to choose ε small enough such that Δ is strictly greater than zero. Therefore, expression (3.106) is lower bounded by:

$$\mathrm{MSD}_{\mathrm{ncop},k}(m) - \mathrm{MSD}_{\mathrm{cta},k}(m) \geq \mu^2 \lambda_m(R_u) \left[-z + \Delta \cdot \sum_{j=J_m}^{\infty} (1 - \mu\lambda_m(R_u))^{2j}\right] \tag{3.110}$$

where the term $z \geq 0$ is an upper bound for the first J_m terms of the summation in (3.106), i.e.,

$$\left|\sum_{j=0}^{J_m-1} (1 - \mu\lambda_m(R_u))^{2j} e_k^T \left(\Sigma_v - A^{Tj}\Sigma_v A^j\right) e_k\right| \leq z < \infty. \tag{3.111}$$

It can be verified that the series inside the brackets of (3.110) is strictly decreasing in $\mu \in (0, 1/\lambda_m(R_u))$. In addition,

$$\lim_{\mu\to 0} \left(\sum_{j=J_m}^{\infty} (1 - \mu\lambda_m(R_u))^{2j}\right) = \infty. \tag{3.112}$$

Thus, there exists a $\mu_m^\circ > 0$ such that the sum inside the bracket of (3.110) becomes positive and, hence,

$$\mathrm{MSD}_{\mathrm{ncop},k}(m) - \mathrm{MSD}_{\mathrm{cta},k}(m) > 0 \tag{3.113}$$

for all $0 < \mu \leq \mu_m^\circ$. Repeating the above argument, we will obtain a collection of step-size bounds $\{\mu_1^\circ, \mu_2^\circ, \ldots, \mu_M^\circ\}$. We then choose $\mu^\circ = \min\{\mu_1^\circ, \mu_2^\circ, \ldots, \mu_M^\circ\}$ so that relation (3.113) holds for all m. Then, applying Lemma 3.4, we arrive at (3.64) for any μ satisfying $0 < \mu \leq \mu^\circ$.

3.E Condition (3.53) Implies (3.63) when A is Primitive

It follows from (3.53) that $A^{Tj}\Sigma_v A^j - A^{T(j+1)}\Sigma_v A^{j+1} \geq 0$ for any nonnegative integer j and then

$$\sum_{j=0}^{J}\left(A^{Tj}\Sigma_v A^j - A^{T(j+1)}\Sigma_v A^{j+1}\right) = \Sigma_v - A^{T(J+1)}\Sigma_v A^{J+1} \geq 0. \tag{3.114}$$

Since A is primitive, as J tends to infinity, we get from (3.62) that

$$\lim_{J\to\infty}\left(\Sigma_v - A^{T(J+1)}\Sigma_v A^{J+1}\right) = \Sigma_v - r_1 s_1^T \Sigma_v s_1 r_1^T \geq 0. \tag{3.115}$$

Using (2.102), we conclude that

$$\det(\Sigma_v - r_1 s_1^T \Sigma_v s_1 r_1^T) = \det(\Sigma_v)\cdot\det\left(I_N - \Sigma_v^{-1}\mathbb{1}_N \cdot \frac{s_1^T\Sigma_v s_1}{N}\mathbb{1}_N^T\right) \geq 0. \tag{3.116}$$

Since for any column vectors $\{x, y\}$ of size N, it holds that

$$\det(I_N - x\cdot y^T) = 1 - y^T\cdot x \tag{3.117}$$

relation (3.116) implies that the following must hold:

$$\left(1 - \frac{s_1^T\Sigma_v s_1}{N}\mathbb{1}_N^T\cdot\Sigma_v^{-1}\mathbb{1}_N\right) \geq 0. \tag{3.118}$$

However, by the Cauchy-Schwarz inequality [64] and using the fact that $s_1^T\mathbb{1}_N/\sqrt{N} = 1$ from (2.102), we have

$$\begin{aligned}
\frac{s_1^T\Sigma_v s_1}{N}\mathbb{1}_N^T\cdot\Sigma_v^{-1}\mathbb{1}_N &= \left(\sum_{l=1}^{N}\sigma_{v,l}^2\frac{s_{1,l}^2}{N}\right)\cdot\left(\sum_{l=1}^{N}\sigma_{v,l}^{-2}\right)\\
&\geq \left(\sum_{l=1}^{N}\frac{s_{1,l}}{\sqrt{N}}\right)^2\\
&= \left(\frac{s_1^T\mathbb{1}_N}{\sqrt{N}}\right)^2\\
&= 1
\end{aligned} \tag{3.119}$$

where $s_{1,l}$ denotes the lth entry of s_1. Therefore, relation (3.118) can hold only with equality in (3.119). In turn, equality in (3.119) holds if, and only if, there exists a constant c such that $s_{1,l}/\sqrt{N} = c \cdot \sigma_{v,l}^{-2}$ for all l. By the fact that $s_1^T \mathbb{1}_N/\sqrt{N} = 1$, we get:

$$\frac{s_{l,1}}{\sqrt{N}} = \frac{\sigma_{v,l}^{-2}}{\sum_{m=1}^{N} \sigma_{v,m}^{-2}} \tag{3.120}$$

and arrive at (3.63) since

$$\begin{aligned} \sigma_{v,k}^2 - \frac{s_1^T \Sigma_v s_1}{N} &= \sigma_{v,k}^2 - \frac{1}{\sum_{l=1}^{N} \sigma_{v,l}^{-2}} \\ &> \sigma_{v,k}^2 - \frac{1}{\sigma_{v,k}^{-2}} \\ &= 0. \end{aligned} \tag{3.121}$$

CHAPTER 4

Design of Combination Rules

Although the stability of diffusion networks is insensitive to the topology, their mean-square performance is still dependent on the combination weights that are used by the agents to run the adaptation process. This motivates us to design combination rules that are able to mitigate the effect of measurement noise. Several combination rules have been proposed in the literature to fuse the data, especially in the context of consensus-based iterations [69, 166, 167], such as the Metropolis rule and the maximum-degree rule. However, these schemes ignore the noise profile that exists at the agents and usually result in performance degradation [141]. This is because the signal-to-noise ratio (SNR) generally varies across the agents, and some agents may have considerably lower SNR than other agents. Therefore, designing optimal combination rules that are aware of the variation in noise profile across the network is an important task. Moreover, in mobile networks such as those studied in Chapter 7 where agents are continuously on the move and where neighborhoods evolve over time, it is particularly critical to develop *adaptive* combination strategies that are able to track the dynamics of the noise profile and to perform estimation and inference successfully under such demanding and varying conditions.

In addition to measurement noise at the individual agents, we further consider the effect of noise over the links over which information is shared among neighbors [148, 174]. Noise over the communications links is due to various fac-

tors, including thermal noise, imperfect channel information, or inference and estimation errors. We shall refer to the noise over the communication links as information exchange noise to distinguish it from the measurement noise at the individual agents. There have been a couple of useful studies on the influence of link noise on the performance of distributed adaptation strategies [61,72,77,133,134]. However, these studies focused on assessing the degradation in performance that results from exchanging information over noisy links. In comparison, we are instead interested in addressing a more critical problem, namely, how to design the combination weights used by the network in order to counter the effect of noise over the communication links. Interestingly, the analysis will establish the revealing fact that nodes may need to trust their local data more heavily than the data collected from their neighbors over the noisy communication channels even when the quality of the node's own measurement is worse than that of its neighbors' data.

Some earlier works (such as [30, 34, 69]) have considered the problem of optimizing combination weights to minimize the MSD of a network under the assumption of noise-free exchange links. The optimization problems in these works turn out to be generally non-convex (see (4.22) further ahead) and it is not possible to determine closed-form expressions for the optimal combination weights in terms of the data and network parameters. In this work, we address this difficulty. Under some reasonable assumptions, we show how to formulate a convex optimization problem and determine closed-form expressions for the combination weights. Nevertheless, the solution requires knowledge of the second-order statistics of the regression and noise sources. This observation leads us to develop an adaptive procedure for adjusting these weights on the fly. Doing so helps resolve two issues. First, there is no need to know beforehand the second-order moments of the data; these are replaced by useful instantaneous approximations.

And, second, we end up with an adaptive rule for adjusting the weights themselves. As such, the network is able to continue to deliver performance even when the data and noise statistical profile changes with time as happens under non-stationary environments. In this way, besides the standard adaptation step to solve the desired estimation or inference problem, each agent also runs an adaptation step to adjust its combination weights on the fly. We thus end up with a multi-level adaptive solution, where agents react to the dynamics in their environment in real-time. We may remark that an earlier adaptive construction for the combination weights was proposed in [33, 141]; however, that solution relied on approximating a certain covariance matrix and results in higher computational complexity. We compare further with this solution in the simulation section (see Figs. 4.6 and 4.8).

4.1 Diffusion Strategies with Information Exchange Noise

From Theorem 3.2, diffusion networks ensure stability and converge faster than consensus networks, regardless of the network topology. Nevertheless, the MSD expression (see Theorems 3.3-3.5) depends on the selection of the combination matrix A. In the following, we show how to select the combination weights. We focus only on diffusion networks due to their performance over consensus networks. We further consider the situation when the links over which information is exchanged are subject to noise and interference. The objective of this chapter is to design the combination matrix to minimize the network MSD for diffusion networks in the presence of information exchange noise.

To model noisy links and in a manner similar to [61,72,77,133,134], we modify

the ATC diffusion strategy (2.29) as follows:

$$\begin{cases} \psi_{k,i} = w_{k,i-1} + \mu_k \cdot u_{k,i}^*[d_k(i) - u_{k,i} w_{k,i-1}] & \text{(4.1a)} \\ \phi_{l,k,i} = \psi_{l,i} + n_{l,k,i} & \text{(4.1b)} \\ w_{k,i} = a_{k,k}\psi_{k,i} + \displaystyle\sum_{l \in \mathcal{N}_k \backslash \{k\}} a_{l,k}\phi_{l,k,i} & \text{(4.1c)} \end{cases}$$

where $n_{l,k,i}$ models the exchange noise over the communications link connecting agent l to agent k. The term $n_{l,k,i}$ refers to a realization of a stationary random process, $\boldsymbol{n}_{l,k,i}$, with zero mean and covariance matrix $R_{n,l,k}$; we assume it is spatially and temporally independent and is independent of all other random variables. Note that in the combination step (4.1c), agent k is able to use its own noise-free intermediate estimate $\psi_{k,i}$, while the other estimates $\{\phi_{l,k,i}\}$ from the neighbors are affected by information exchange noise. Other variants of the algorithm are possible, for example, by reversing the order of steps (4.1a) and (4.1c), we obtain the CTA diffusion solution [34, 95]:

$$\begin{cases} \phi_{l,k,i-1} = w_{l,i-1} + n_{l,k,i} & \text{(4.2a)} \\ \psi_{k,i-1} = a_{k,k} w_{k,i-1} + \displaystyle\sum_{l \in \mathcal{N}_k \backslash \{k\}} a_{l,k}\phi_{l,k,i-1} & \text{(4.2b)} \\ w_{k,i} = \psi_{k,i-1} + \mu_k \cdot u_{k,i}^*[d_k(i) - u_{k,i}\psi_{k,i-1}] & \text{(4.2c)} \end{cases}$$

The main question that we wish to study is to examine how the combination weights $\{a_{l,k}\}$ should be selected in order to counter the effect of the measurement noise $v_k(i)$ and the information exchange noise $n_{l,k,i}$. Moreover, the optimal weights should be computable locally, so that each agent can evaluate what weights it should use based only on its interactions with its neighbors. Before we study this problem, we derive an expression for the network MSD in the presence of information exchange noise — see expression (4.11) further ahead. Subsequently, we motivate an optimization problem that will allow each agent to

select the optimal combination weights and, more importantly, to adjust them in real-time.

4.1.1 Mean-Square Performance

Expressions for the network performance, and conditions for its mean and mean-square stability, in the absence of information exchange noise, were derived in Chapter 2. The results can be readily extended to ATC and CTA networks with noisy links in a straightforward manner, as was already done in Chapter 20 of [123] for adaptive filters operating under random walk models. We carry out the performance analysis in this brief section (see Tables 2.2 and 4.1 further ahead and the analysis that goes with it) in preparation for our study of optimal combination weights.

From (4.1a)-(4.1c) and model (2.3), some algebra will show that the global error vector $\tilde{\boldsymbol{w}}_i$ for the ATC diffusion strategy evolves according to the relation:

$$\tilde{\boldsymbol{w}}_i = \mathcal{A}^T(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}^T\mathcal{M}\boldsymbol{s}_i - \boldsymbol{n}_i \tag{4.3}$$

where $\mathcal{M}$, $\mathcal{A}$, $\boldsymbol{\mathcal{R}}_i$, and $\boldsymbol{s}_i$ were defined in (2.38), (2.39), (2.41), and (2.42), respectively, and

$$\boldsymbol{n}_i \triangleq \text{col}\left\{\sum_{l\neq 1} a_{l,1}\boldsymbol{n}_{l,1,i}, \sum_{l\neq 2} a_{l,2}\boldsymbol{n}_{l,2,i}, \cdots, \sum_{l\neq N} a_{l,N}\boldsymbol{n}_{l,N,i}\right\} \tag{4.4}$$

Note that the error recursion (4.3) is the same as the one in (2.43) except for the last term $\boldsymbol{n}_i$, which is contributed by the information exchange noise. Similarly, for the CTA diffusion strategy, starting from (4.2a)-(4.2c) and using model (2.3), we obtain the following recursion for $\tilde{\boldsymbol{w}}_i$:

$$\tilde{\boldsymbol{w}}_i = (I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\mathcal{A}^T\tilde{\boldsymbol{w}}_{i-1} - \mathcal{M}\boldsymbol{s}_i - (I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\boldsymbol{n}_i \tag{4.5}$$

Table 4.1: Variables that control the dynamics of ATC and CTA networks.

	ATC diffusion (4.1)	**CTA diffusion** (4.2)
$\boldsymbol{z}_i$	$\boldsymbol{n}_i$	$(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\boldsymbol{n}_i$
$\mathcal{Z} \triangleq \mathbb{E}\boldsymbol{z}_i\boldsymbol{z}_i^*$	$\mathcal{G}$	$(I_{NM} - \mathcal{M}\mathcal{R})\mathcal{G}(I_{NM} - \mathcal{M}\mathcal{R})$

We observe from recursions (4.3) and (4.5) that the recursions evolve in a similar manner. We therefore introduce the following general form:

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \cdot \tilde{\boldsymbol{w}}_{i-1} - \boldsymbol{y}_i - \boldsymbol{z}_i \tag{4.6}$$

The ATC and CTA diffusion strategies can be obtained by choosing the matrix $\boldsymbol{\mathcal{B}}_i$ and the vectors $\{\boldsymbol{y}_i, \boldsymbol{z}_i\}$ appropriately, as indicated in Tables 2.2 and 4.1.

Since the matrix $\boldsymbol{\mathcal{B}}_i$ is independent of $\tilde{\boldsymbol{w}}_{i-1}$, and the vectors $\{\boldsymbol{y}_i, \boldsymbol{z}_i\}$ are independent of other variables and have zero mean, by following the same arguments from Chapter 2, we can derive the following mean and variance relations for $\mathbb{E}\tilde{\boldsymbol{w}}_i$:

$$\mathbb{E}\tilde{\boldsymbol{w}}_i = \mathcal{B} \cdot \mathbb{E}\tilde{\boldsymbol{w}}_{i-1} \tag{4.7}$$

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_\sigma^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 + \left[\mathrm{vec}(\mathcal{Y}^T)\right]^T \sigma + \left[\mathrm{vec}(\mathcal{Z}^T)\right]^T \sigma \tag{4.8}$$

where $\mathcal{B}$, $\mathcal{Y}$ and

$$\mathcal{Z} \triangleq \mathbb{E}\boldsymbol{z}_i\boldsymbol{z}_i^* \tag{4.9}$$

are defined in Tables 2.2 and 4.1. Moreover, the matrices $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{F}$ are defined in (2.48), (2.56), and (2.59), respectively, and

$$\mathcal{G} \triangleq \mathbb{E}\boldsymbol{n}_i\boldsymbol{n}_i^* = \mathrm{diag}\left\{\sum_{l\neq 1} a_{l,1}^2 R_{n,l,1}, \sum_{l\neq 2} a_{l,2}^2 R_{n,l,2}, \cdots, \sum_{l\neq N} a_{l,N}^2 R_{n,l,N}\right\} \tag{4.10}$$

Relations (4.7)-(4.8) are identical to the mean and variance relations of the traditional ATC and CTA diffusion strategies in the noiseless link case (see (2.46) and

(2.58)), except for the last term of (4.8) with $\mathcal{Z}$, which results from the presence of information exchange noise. Note that, as argued in Chapter 2, the mean and mean-square stability only depend on the matrices $\{\mathcal{B}, \mathcal{F}\}$. For sufficiently small step-sizes, the ATC and CTA diffusion strategies in the presence of information exchange noise are likewise mean and mean-square stable.

Moreover, from starting (4.8), we find that the network MSD is now given by

$$\text{MSD} = \frac{1}{N}\sum_{j=0}^{\infty} \text{Tr}\left[\mathcal{B}^j(\mathcal{Y}+\mathcal{Z})\mathcal{B}^{*j}\right] \tag{4.11}$$

By substituting the corresponding matrices $\{\mathcal{B}, \mathcal{Y}, \mathcal{Z}\}$ in Tables 2.2 and 4.1, we can arrive at the MSD expressions for the ATC and CTA diffusion strategies. Note that these expressions depend on the step-sizes $\{\mu_k\}$, the regression covariance matrices $\{R_{u,k}\}$, and the noise profiles $\{\sigma_{v,k}^2, R_{n,l,k}\}$. It is useful to relate these two MSD expression. First, note that

$$\text{MSD}_{\text{cta}} = \frac{1}{N}\text{Tr}(\mathcal{Y}_{\text{cta}}) + \frac{1}{N}\sum_{j=0}^{\infty} \text{Tr}\left[\mathcal{B}_{\text{cta}}^{j+1}\mathcal{Y}_{\text{cta}}\mathcal{B}_{\text{cta}}^{*(j+1)} + \mathcal{B}_{\text{cta}}^{j}\mathcal{Z}_{\text{cta}}\mathcal{B}_{\text{cta}}^{*j}\right] \tag{4.12}$$

Moreover, it holds that

$$\mathcal{B}_{\text{cta}}^{j+1}\mathcal{Y}_{\text{cta}}\mathcal{B}_{\text{cta}}^{*(j+1)} = (I_{NM} - \mathcal{M}\mathcal{R})\mathcal{B}_{\text{atc}}^{j}\mathcal{Y}_{\text{atc}}\mathcal{B}_{\text{atc}}^{*j}(I_{NM} - \mathcal{M}\mathcal{R}) \tag{4.13}$$

$$\mathcal{B}_{\text{cta}}^{j}\mathcal{Z}_{\text{cta}}\mathcal{B}_{\text{cta}}^{*j} = (I_{NM} - \mathcal{M}\mathcal{R})\mathcal{B}_{\text{atc}}^{j}\mathcal{Z}_{\text{atc}}\mathcal{B}_{\text{atc}}^{*j}(I_{NM} - \mathcal{M}\mathcal{R}) \tag{4.14}$$

Therefore, we obtain that

$$\begin{aligned}\text{MSD}_{\text{cta}} = &\frac{1}{N}\text{Tr}(\mathcal{M}\mathcal{S}\mathcal{M}) \\ &+ \frac{1}{N}\text{Tr}\left[(I_{NM} - \mathcal{M}\mathcal{R})\left(\sum_{j=0}^{\infty}\mathcal{B}_{\text{atc}}^{j}(\mathcal{Y}_{\text{atc}} + \mathcal{Z}_{\text{atc}})\mathcal{B}_{\text{atc}}^{*j}\right)(I_{NM} - \mathcal{M}\mathcal{R})\right]\end{aligned} \tag{4.15}$$

That is, we can rewrite the network MSD for the CTA diffusion strategy in terms of the matrices $\{\mathcal{B}, \mathcal{Y}, \mathcal{Z}\}$ for the ATC diffusion strategy. Relation (4.15) helps

us relate the optimal combination weights of these two strategies. It turns out that the optimal combination weights that we derive apply to both strategies.

4.2 Influence of Link Noise on Performance

Expression (4.11) reveals in an interesting way how the noise sources originating from any particular agent end up influencing the overall network performance. To see this, we start from (4.6) and expand the recursion. Then, we arrive at

$$\tilde{\boldsymbol{w}}_i = \left(\prod_{m=1}^{i} \boldsymbol{\mathcal{B}}_m\right) \tilde{\boldsymbol{w}}_0 - \sum_{j=0}^{i-1} \boldsymbol{\theta}_j \tag{4.16}$$

where $\boldsymbol{\theta}_j$ denotes the aggregate effect of noise sources j steps ahead in time and is defined as:

$$\boldsymbol{\theta}_j \triangleq \left(\prod_{m=i-j+1}^{i} \boldsymbol{\mathcal{B}}_m\right) (\boldsymbol{y}_{i-j} + \boldsymbol{z}_{i-j}) \tag{4.17}$$

where the symbol $\prod$ denotes the product of a sequence of matrices:

$$\prod_{m=j}^{i} \boldsymbol{\mathcal{B}}_m \triangleq \begin{cases} \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{B}}_{i-1} \cdots \boldsymbol{\mathcal{B}}_j, & \text{if } i \geq j \\ I_{NM}, & \text{otherwise} \end{cases} \tag{4.18}$$

From (4.16), the mean-square of $\tilde{\boldsymbol{w}}_i$ is given by

$$\begin{aligned} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2 &= \text{Tr}\left[\mathbb{E}\left(\prod_{m=1}^{i} \boldsymbol{\mathcal{B}}_m\right) \tilde{\boldsymbol{w}}_0 \tilde{\boldsymbol{w}}_0^* \left(\prod_{m=1}^{i} \boldsymbol{\mathcal{B}}_m\right)^*\right] + \sum_{j=0}^{i-1} \mathbb{E}\|\boldsymbol{\theta}_j\|^2 \\ &\approx \text{Tr}\left(\mathcal{B}^i \mathbb{E}\|\tilde{\boldsymbol{w}}_0\|^2 \mathcal{B}^{*i}\right) + \sum_{j=0}^{i-1} \mathbb{E}\|\boldsymbol{\theta}_j\|^2 \end{aligned} \tag{4.19}$$

where we used the fact that the noise sources $\{\boldsymbol{\theta}_j\}$, for $j = 0, \ldots, i-1$, are independent of each other and have zero mean, and where we applied approximation (2.66) for sufficiently small step-sizes. Now, let the time index i tend to infinity. Since the matrix $\mathcal{B}$ is stable, the first term in (4.19) converges to zero,

and expression (4.19) leads to

$$\lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2 = \sum_{j=0}^{\infty} \mathbb{E}\|\boldsymbol{\theta}_j\|^2 \tag{4.20}$$

Next, we verify that $\mathbb{E}\|\boldsymbol{\theta}_j\|^2$ is equal to the jth term of the summation in (4.11). Using approximation (2.66) and from (4.17), we obtain that

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{\theta}_j\|^2 &= \mathrm{Tr}\left[\mathbb{E}\left(\prod_{m=i-j+1}^{i} \boldsymbol{\mathcal{B}}_m\right)(\boldsymbol{y}_{i-j}+\boldsymbol{z}_{i-j})(\boldsymbol{y}_{i-j}+\boldsymbol{z}_{i-j})^*\left(\prod_{m=i-j+1}^{i} \boldsymbol{\mathcal{B}}_m\right)^*\right] \\
&= \mathrm{Tr}\left[\mathbb{E}\left(\prod_{m=i-j+1}^{i} \boldsymbol{\mathcal{B}}_m\right)(\mathcal{Y}+\mathcal{Z})\left(\prod_{m=i-j+1}^{i} \boldsymbol{\mathcal{B}}_m\right)^*\right] \\
&\approx \mathrm{Tr}\left[\mathcal{B}^j(\mathcal{Y}+\mathcal{Z})\mathcal{B}^{*j}\right]
\end{aligned} \tag{4.21}$$

Therefore, the jth term of the summation in (4.11) can be interpreted as capturing the contribution of the noise sources j steps ahead in time. We also observe from (4.11) that at each time step, the spread of the noise sources is dictated by the choice of the combination matrix A. We now discuss how to select the combination matrix A in order to minimize the effect of these noises on the network MSD.

4.3 Optimal Combination Matrix: Special Case

The selection of the optimal combination weights is formulated as the following optimization problem:

$$\begin{aligned}
&\min_{A} \quad \text{MSD (4.11)} \\
&\text{subject to (2.25)}
\end{aligned} \tag{4.22}$$

As already indicated in [30,34], the network MSD is not convex in A, and therefore the solution to the above optimization problem is generally intractable. Therefore, we start our discussions by considering some initial special cases in order

to shed some light on the behavior of the diffusion strategies in the presence of noisy links. Later, we consider more general scenarios. We therefore start by replacing (4.22) by the problem of minimizing an upper bound on the MSD; this formulation leads to closed-form expressions that are shown to perform well. The solution will further suggest an adaptive construction for the combination weights. We start with the following special case.

Theorem 4.1. *Suppose that $I_{NM} - \mathcal{M}\mathcal{R} = \alpha I_{NM}$ for some constant α (this means that $R_{u,k}$ has the form $R_{u,k} = \sigma_{u,k}^2 I_M$ and $1 - \mu_k \sigma_{u,k}^2 = \alpha$ for all k). Then, the optimal combination weights for the ATC and CTA diffusion strategies are the same.*

Proof. Since $I_{NM} - \mathcal{M}\mathcal{R} = \alpha I_{NM}$, relation (4.15) can be simplified to

$$\mathrm{MSD}_{\mathrm{atc}} = \frac{1}{N}\mathrm{Tr}(\mathcal{M}\mathcal{S}\mathcal{M}) + \alpha^2 \mathrm{MSD}_{\mathrm{cta}} \tag{4.23}$$

Note that the first term in (4.23) is independent of A. Therefore, the matrix A that minimizes $\mathrm{MSD}_{\mathrm{atc}}$ also minimizes $\mathrm{MSD}_{\mathrm{cta}}$. □

Thus, assume that the covariance matrices of the regressors $\{\boldsymbol{u}_{k,i}\}$ have the form $R_{u,k} = \sigma_u^2 I_M$ for all k. Assume further that the covariance matrix of the information exchange noise $\boldsymbol{n}_{l,k,i}$ has a similar form, namely, $R_{n,l,k} = \sigma_{n,l,k}^2 I_M$. We also assume that all nodes use the same step-size (i.e., $\mu_k = \mu$ for all k). From Theorem 4.1, the ATC and CTA strategies will have the same optimal combination weights and we therefore focus on the CTA strategy in this section. The aforementioned assumptions lead to the following expressions:

$$\mathcal{B}_{\mathrm{cta}} = (1 - \mu\sigma_u^2) \cdot A^T \otimes I_M \tag{4.24}$$

$$\mathcal{Y}_{\mathrm{cta}} = \mu^2 \sigma_u^2 \cdot \Sigma_v \otimes I_M \tag{4.25}$$

$$\mathcal{Z}_{\mathrm{cta}} = (1 - \mu\sigma_u^2)^2 \cdot \Sigma_n \otimes I_M \tag{4.26}$$

where Σ_v is defined in (3.24) and

$$\Sigma_n = \text{diag}\left\{\sum_{l\neq 1} a_{l,1}^2\sigma_{n,l,1}^2, \sum_{l\neq 2} a_{l,2}^2\sigma_{n,l,2}^2, \cdots, \sum_{l\neq N} a_{l,N}^2\sigma_{n,l,N}^2\right\} \tag{4.27}$$

Thus, from (4.11), the MSD of the CTA strategy can be expressed as

$$\text{MSD}_{\text{cta}} = \frac{M}{N}\sum_{j=0}^{\infty}(1-\mu\sigma_u^2)^{2j}\cdot \text{Tr}\left[\mu^2\sigma_u^2\cdot(A^T)^j\Sigma_v A^j + (1-\mu\sigma_u^2)^2\cdot(A^T)^j\Sigma_n A^j\right] \tag{4.28}$$

where we used the equalities $(A\otimes B)(C\otimes D) = AC\otimes BD$ and $\text{Tr}(A\otimes B) = \text{Tr}(A)\cdot\text{Tr}(B)$.

Even though the MSD of the CTA strategy is simplified to (4.28), the MSD is still not convex in A. In the following, we focus on the case of two nodes ($N = 2$) to highlight some interesting patterns of behavior. Let the combination matrix A be

$$A = \begin{bmatrix} 1-a & b \\ a & 1-b \end{bmatrix} \tag{4.29}$$

In addition, we let

$$\Sigma_v = \sigma_v^2\begin{bmatrix} r & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \Sigma_n = \sigma_n^2\begin{bmatrix} a^2 & 0 \\ 0 & sb^2 \end{bmatrix} \tag{4.30}$$

where r and s represent the ratios of the noise variances at nodes 1 and 2, respectively (r for the measurement noise and s for the information exchange noise). Moreover, for notational simplicity, we introduce the two variables:

$$\eta \triangleq (1-\mu\sigma_u^2)^2 \approx 1-2\mu\sigma_u^2 \quad \text{and} \quad t \triangleq \sigma_n^2/\sigma_v^2 \tag{4.31}$$

Note that t is the ratio between the variances of the two noise sources at node 2. In this case, the eigenvalue decomposition for the combination matrix is $A = U\Lambda U^{-1}$ where

$$U = \begin{bmatrix} b & 1 \\ a & -1 \end{bmatrix} \quad \text{and} \quad \Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 1-a-b \end{bmatrix}$$

In addition, $\mathcal{Y}_{\text{cta}}$ in (4.25) and $\mathcal{Z}_{\text{cta}}$ in (4.26) become

$$\mathcal{Y}_{\text{cta}} \approx \frac{\sigma_v^2}{2}\mu(1-\eta)\begin{bmatrix} r & 0 \\ 0 & 1 \end{bmatrix} \otimes I_M \quad \text{and} \quad \mathcal{Z}_{\text{cta}} = \sigma_v^2 t\eta \begin{bmatrix} a^2 & 0 \\ 0 & sb^2 \end{bmatrix} \otimes I_M \tag{4.32}$$

where we approximated $\mu^2\sigma_u^2$ to $\mu(1-\eta)/2$ by (4.31). Some algebra shows that (4.28) becomes

$$\begin{aligned} \text{MSD}_{\text{cta}} &= \frac{M\sigma_v^2}{2(a+b)^2} \\ &\times \left\{ \frac{1}{2}\mu(1-\eta)\left[2(rb^2+a^2)\lambda_1 + (a^2+b^2)(r+1)\lambda_3 + 2(a-b)(rb-a)\lambda_2\right] \right. \\ &\left. + t\eta\left[2a^2b^2(s+1)\lambda_1 + (a^2+b^2)(a^2+sb^2)\lambda_3 + 2ab(a-b)(a-sb)\lambda_2\right] \right\} \end{aligned} \tag{4.33}$$

where

$$\lambda_1 = \frac{1}{1-\eta}, \quad \lambda_2 = \frac{1}{1-\eta(1-a-b)}, \quad \lambda_3 = \frac{1}{1-\eta(1-a-b)^2} \tag{4.34}$$

Note that the first term inside the brackets of (4.33) is contributed by the measurement noise, while the second term originates from the information exchange noise. For $a = b$, the MSD in (4.33) simplifies to

$$\text{MSD}_{\text{cta}} = \frac{M\sigma_v^2}{4}[\mu(1-\eta) + 2t\eta a^2] \times \left(\frac{1}{1-\eta} + \frac{1}{1-\eta(1-2a)^2}\right) \tag{4.35}$$

To determine a set of weights $\{a, b\}$ such that the network MSD given by (4.33) is minimized, we set the partial derivatives of the MSD with respect to a and b to zero, i.e.,

$$\frac{\partial \text{MSD}_{\text{cta}}}{\partial a} = \frac{\partial \text{MSD}_{\text{cta}}}{\partial b} = 0 \tag{4.36}$$

From the first equality in (4.36), we obtain that the optimal values for $\{a, b\}$

should satisfy the relation:

$$\begin{aligned} h(a,b) \triangleq\ & \mu(1-\eta)\left[2(a-rb)(\lambda_1-\lambda_2)+(r+1)(a-b)(\lambda_3-\lambda_2)\right] \\ & + 2t\eta(a+b)^2(a-sb)\lambda_2 + 4t\eta(s+1)ab(b-a)(\lambda_1-\lambda_2) \\ & + 2t\eta\left[(s+1)ab(b-a)+2(a^3-sb^3)\right](\lambda_3-\lambda_2) \\ =\ & 0 \end{aligned} \tag{4.37}$$

The function $h(a,b)$ in (4.37) is nonlinear in the parameters $\{a,b\}$. However, if we are considering noise-free links (i.e., $t=0$), then weights $\{a,b\}$ satisfying the relations

$$a+b=1 \quad \text{and} \quad a-rb=0 \tag{4.38}$$

would satisfy relation (4.37). Thus, we have

$$a=\frac{r}{r+1}=\frac{\sigma_{v,2}^{-2}}{\sigma_{v,1}^{-2}+\sigma_{v,2}^{-2}} \quad \text{and} \quad b=\frac{1}{r+1}=\frac{\sigma_{v,1}^{-2}}{\sigma_{v,1}^{-2}+\sigma_{v,2}^{-2}} \tag{4.39}$$

Note that we arrive at the same result from [173]. In this way, node k combines the information from its neighbors in proportion to the inverse of the variances of the measurement noise. The result is physically meaningful. Nodes with smaller noise variance will be given larger weights.

To further examine the influence of the information exchange noise, we consider the example corresponding to the case $r=s=1$ (i.e., equal noise levels at both nodes for both measurement noise and information exchange noise). Relation (4.37) between $\{a,b\}$ simplifies to

$$a=b \tag{4.40}$$

This result is expected since both nodes have the same noise levels. In this case, setting the derivative of the MSD in (4.35) with respect to a to zero, we find that

the optimal value for a should satisfy the following fifth-order equation:

$$\begin{aligned} f(a) &\triangleq 16\eta^2 t a^5 - 32\eta^2 t a^4 + 8\eta(3\eta-1)t a^3 + 10\eta(1-\eta)t a^2 \\ &\quad + 2(1-\eta)^2(t+\mu)a - \mu(1-\eta)^2 \\ &= 0 \end{aligned} \tag{4.41}$$

In the following, we are going to show that there is a single value of a that minimizes the MSD (4.35) and that this value for a lies between $(0, 0.5]$. We first establish some useful properties as follows.

Lemma 4.1. *Suppose that $r = s = 1$ and $a = b$. For $0 < \eta < 1$, $t \geq 0$ and $a \in [0, 1]$, the network MSD in (4.35) and the function $f(a)$ in (4.41) satisfy the following properties*

1. $\frac{\partial MSD_{cta}}{\partial a}\big|_{a=0} < 0$ *and* $\frac{\partial MSD_{cta}}{\partial a}\big|_{a=1} > 0$.
2. $f(0) < 0$ *and* $f(0.5) \geq 0$ *with equality to zero if, and only if,* $t = 0$.
3. $f(a) > 0$ *if* $0.5 < a \leq 1$.
4. $f'(a) > 0$ *if* $0 \leq a \leq 0.5$.

Proof. See Appendix 4.A. □

With these properties, we arrive at the following result.

Theorem 4.2. *In case of $r = s = 1$, the optimal combination weights $\{a, b\}$ that minimize the network MSD in (4.33) satisfy $a = b$ and that the weight a $(= b)$ lies in $(0, 0.5]$ and is unique.*

Proof. When $r = s = 1$, we have shown in (4.40) that $a = b$ and then the MSD expression simplifies to (4.35). Since the MSD in (4.35) is a bounded function of a when $a \in [0, 1]$, there exists a value for a that minimizes the MSD. From the

first property in the previous lemma, we see that the optimal a cannot be at the end points, 0 or 1, and has to satisfy equation (4.41). From the second and third properties, we conclude that there exists at least one value of $a \in (0, 0.5]$ satisfying (4.41) and for any value of a greater than 0.5, equation (4.41) does not hold. To establish the uniqueness of the minimizing a, from the fourth property, we know that $f(a)$ is a strictly increasing function of a when $0 \leq a \leq 0.5$. Therefore, there is only one value of $a \in (0, 0.5]$ satisfying (4.41) and that minimizes the MSD.

□

The above result implies that when both nodes have the same noise levels, the averaging strategy (i.e., $a = b = 1/2$) is the optimal strategy if, and only if, there are noisy links. If there is noise from information exchange, the nodes have to lower the value of the combination weight, a. That is, a node has to place more weight on its own estimate.

In this section, we considered a very specific case ($\mu_k = \mu$, $R_{u,k} = \sigma_u^2 I_M$, and $R_{n,l,k} = \sigma_{n,l,k}^2 I_M$) corresponding to $N = 2$ to highlight the above interesting fact. Even in the $N = 2$ case, there is no closed-form expression for the optimal combination weights. Nevertheless, conditions (4.37) and (4.41) for the combination weights for two-node networks serve as a benchmark when we proceed to approximate optimal solutions for the networks with an arbitrary number of nodes and help us examine quantitatively the effectiveness of the approximations.

4.4 Optimal Combination Matrix: General Case

We observe from (4.21) that minimizing the network MSD is equivalent to reducing the effect of the noise sources across the network. We first have the following preliminary result.

Lemma 4.2. *Assume the spectral radius of the matrix $\mathcal{B}^*\mathcal{B}$ satisfies*

$$\rho(\mathcal{B}^*\mathcal{B}) \leq r \tag{4.42}$$

for some positive constant r. Then, for any nonnegative-definite matrix $\mathcal{Y}$,

$$\mathrm{Tr}(\mathcal{B}\mathcal{Y}\mathcal{B}^*) \leq r \cdot \mathrm{Tr}(\mathcal{Y}) \tag{4.43}$$

Proof. Since $\mathcal{B}^*\mathcal{B}$ is Hermitian and nonnegative-definite, there exists a unitary matrix U and a diagonal matrix Λ with entries in the range $[0, r]$ such that

$$\mathcal{B}^*\mathcal{B} = U_B \Lambda_B U_B^* \tag{4.44}$$

Then, using the fact that $\mathcal{Y}$ is nonnegative-definite, we have:

$$\begin{aligned}
\mathrm{Tr}(\mathcal{B}\mathcal{Y}\mathcal{B}^*) &= \mathrm{Tr}(\mathcal{Y}\mathcal{B}^*\mathcal{B}) \\
&= \mathrm{Tr}(\mathcal{Y}U_B\Lambda_B U_B^*) \\
&= \mathrm{Tr}(\Lambda U_B^*\mathcal{Y}U_B) \\
&\leq r \cdot \mathrm{Tr}(U_B^*\mathcal{Y}U_B) \\
&= r \cdot \mathrm{Tr}(\mathcal{Y})
\end{aligned} \tag{4.45}$$

□

We now examine the network MSD (4.11) for the ATC diffusion strategy. The following result provides an upper bound on the effect of the noise sources j steps ahead in time, i.e, $\mathbb{E}\|\boldsymbol{\theta}_j\|^2$ in (4.21).

Theorem 4.3. *It holds for nonnegative integer j that:*

$$\mathrm{Tr}\left[\mathcal{B}_{atc}^j(\mathcal{Y}_{atc} + \mathcal{Z}_{atc})\mathcal{B}_{atc}^{j*}\right] \leq N \cdot [\rho(I_{NM} - \mathcal{M}\mathcal{R})]^{2j} \cdot \mathrm{Tr}(\mathcal{Y}_{atc} + \mathcal{Z}_{atc}) \tag{4.46}$$

Proof. See Appendix 4.B. □

Therefore, from (4.11) and using Theorem 4.3, the network MSD of the ATC diffusion strategy is upper bounded by

$$\begin{aligned}\mathrm{MSD}_{\mathrm{atc}} &\leq \frac{1}{N}\sum_{j=0}^{\infty} N\cdot[\rho(I_{NM}-\mathcal{M}\mathcal{R})]^{2j}\cdot\mathrm{Tr}(\mathcal{Y}_{\mathrm{atc}}+\mathcal{Z}_{\mathrm{atc}})\\ &= \frac{\mathrm{Tr}(\mathcal{Y}_{\mathrm{atc}}+\mathcal{Z}_{\mathrm{atc}})}{1-[\rho(I_{NM}-\mathcal{M}\mathcal{R})]^2}\end{aligned} \tag{4.47}$$

Note from the above result that the upper bound depends on the combination matrix A only through the term $\mathrm{Tr}(\mathcal{Y}_{\mathrm{atc}}+\mathcal{Z}_{\mathrm{atc}})$. Motivated by this result, we consider the problem of minimizing this term, namely,

$$\begin{aligned}&\min_{A} \quad g(A)=\mathrm{Tr}(\mathcal{Y}_{\mathrm{atc}}+\mathcal{Z}_{\mathrm{atc}})=\mathrm{Tr}(\mathcal{A}^T\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A}+\mathcal{G})\\ &\text{subject to (2.25)}\end{aligned} \tag{4.48}$$

For the network MSD of the CTA diffusion strategy, we use expression (4.15) and obtain from Lemma 4.2 and Theorem 4.3 that

$$\begin{aligned}\mathrm{MSD}_{\mathrm{cta}} &\leq \frac{1}{N}\mathrm{Tr}(\mathcal{M}\mathcal{S}\mathcal{M})+\frac{[\rho(I_{NM}-\mathcal{M}\mathcal{R})]^2}{N}\mathrm{Tr}\left(\sum_{j=0}^{\infty}\mathcal{B}_{\mathrm{atc}}^{j}(\mathcal{Y}_{\mathrm{atc}}+\mathcal{Z}_{\mathrm{atc}})\mathcal{B}_{\mathrm{atc}}^{*j}\right)\\ &\leq \frac{1}{N}\mathrm{Tr}(\mathcal{M}\mathcal{S}\mathcal{M})+\sum_{j=0}^{\infty}[\rho(I_{NM}-\mathcal{M}\mathcal{R})]^{2+2j}\,\mathrm{Tr}\left(\mathcal{Y}_{\mathrm{atc}}+\mathcal{Z}_{\mathrm{atc}}\right)\\ &= \frac{1}{N}\mathrm{Tr}(\mathcal{M}\mathcal{S}\mathcal{M})+\frac{[\rho(I_{NM}-\mathcal{M}\mathcal{R})]^2}{1-[\rho(I_{NM}-\mathcal{M}\mathcal{R})]^2}\mathrm{Tr}(\mathcal{Y}_{\mathrm{atc}}+\mathcal{Z}_{\mathrm{atc}})\end{aligned} \tag{4.49}$$

Again, the upper bound depends on A only through the term $\mathrm{Tr}(\mathcal{Y}_{\mathrm{atc}}+\mathcal{Z}_{\mathrm{atc}})$. Therefore, we arrive at the same optimization problem for the ATC diffusion strategy as problem (4.48).

To solve problem (4.48), we expand the objective function $g(A)$ and obtain

$$g(A)=\sum_{k=1}^{N}\left\{\mu_k^2\sigma_{v,k}^2\mathrm{Tr}(R_{u,k})a_{k,k}^2+\sum_{j\neq k}[\mu_j^2\sigma_{v,j}^2\mathrm{Tr}(R_{u,j})+\mathrm{Tr}(R_{n,j,k})]a_{j,k}^2\right\} \tag{4.50}$$

Note that the objective function can be written as the sum of local functions with each function depending on the combination weights at one agent. That is, let

us introduce the functions:

$$g_k(a_{l,k}, l = 1, \ldots, N) = \mu_k^2 \sigma_{v,k}^2 \mathrm{Tr}(R_{u,k}) a_{k,k}^2 + \sum_{j \neq k} [\mu_j^2 \sigma_{v,j}^2 \mathrm{Tr}(R_{u,j}) + \mathrm{Tr}(R_{n,j,k})] a_{j,k}^2 \tag{4.51}$$

Then,

$$g(A) = \sum_{k=1}^{N} g_k(a_{l,k}, l = 1, \ldots, N) \tag{4.52}$$

Therefore, the original optimization problem can be decoupled into N optimization problems and each problem is used to find the optimal combination weights at a particular agent. Thus, we introduce the following optimization problem:

$$\begin{aligned} \min_{a_{l,k}, l=1,\ldots,N} \quad & \mu_k^2 \sigma_{v,k}^2 \mathrm{Tr}(R_{u,k}) a_{k,k}^2 + \sum_{j \neq k} [\mu_j^2 \sigma_{v,j}^2 \mathrm{Tr}(R_{u,j}) + \mathrm{Tr}(R_{n,j,k})] a_{j,k}^2 \\ \text{subject to } & (2.25) \end{aligned} \tag{4.53}$$

The problem can be solved using a Lagrangian multiplier, λ_k. Introduce the Lagrangian function

$$\begin{aligned} L(a_{l,k}, l \in \mathcal{N}_k, \lambda_k) = & \mu_k^2 \sigma_{v,k}^2 \mathrm{Tr}(R_{u,k}) a_{k,k}^2 + \sum_{j \in \mathcal{N}_k \setminus \{k\}} [\mu_j^2 \sigma_{v,j}^2 \mathrm{Tr}(R_{u,j}) + \mathrm{Tr}(R_{n,j,k})] a_{j,k}^2 \\ & - \lambda_k \left(\sum_{j \in \mathcal{N}_k} a_{j,k} - 1 \right) \end{aligned} \tag{4.54}$$

Setting the derivation of $L(a_{l,k}, l \in \mathcal{N}_k, \lambda_k)$ with respect to $a_{l,k}$ to zero, we obtain an analytical solution for $\{a_{l,k}\}$ as follows:

$$a_{l,k} = \begin{cases} \gamma_{l,k}^{-2} / \sum_{j \in \mathcal{N}_k} \gamma_{j,k}^{-2}, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \quad \text{(relative-product-variance rule)} \tag{4.55}$$

where $\gamma_{l,k}^2$ denotes product variance measure and is defined as:

$$\gamma_{l,k}^2 = \begin{cases} \mu_k^2 \sigma_{v,k}^2 \mathrm{Tr}(R_{u,k}), & \text{if } l = k \\ \mu_l^2 \sigma_{v,l}^2 \mathrm{Tr}(R_{u,l}) + \mathrm{Tr}(R_{n,l,k}), & \text{otherwise} \end{cases} \tag{4.56}$$

We also to the combination rule (4.55) as the relative-product-variance rule. We observe that for agent k, the weights $\{a_{l,k}\}$ depend on the step-sizes $\{\mu_l\}$, the traces of the covariance matrices $\{\mathrm{Tr}(R_{u,l}), \mathrm{Tr}(R_{n,l,k})\}$, and the variances of the measurement noise $\{\sigma_{v,l}^2\}$ in the neighborhood $\mathcal{N}_k$. The result has a similar form to (4.57) but is more general. Note from (4.56) that the network MSD will be sensitive to the information exchange noise for small step-sizes. In the case of homogeneous agents (see Assumption 3.1) without information exchange noise, the combination weights $\{a_{l,k}\}$ from (4.55) simplify to

$$a_{l,k} = \begin{cases} \sigma_{v,l}^{-2} / \sum_{j \in \mathcal{N}_k} \sigma_{v,j}^{-2}, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \qquad \text{(relative-variance rule)} \tag{4.57}$$

We refer to this combination rule as the relative-variance rule [152,174]. Note that the combination rule in (4.57) extends the result in (4.39) to general networks.

Consider again the two-node case from the last section to see how well the approximate solution in (4.55) matches the optimal solution (4.37). In this case, the combination weights $\{a, b\}$ given by (4.55) simplify to

$$a \approx \frac{(\mu(1-\eta)+2t)^{-1}}{(\mu(1-\eta)r)^{-1} + (\mu(1-\eta)+2t)^{-1}} \tag{4.58}$$

$$b \approx \frac{(\mu(1-\eta)r+2st)^{-1}}{(\mu(1-\eta)r+2st)^{-1} + (\mu(1-\eta))^{-1}} \tag{4.59}$$

where the parameters $\{r, s, t, \eta\}$ are introduced in (4.30)-(4.31). The weights converge to relative-variance rule in (4.39) when the parameter t converges to zero (i.e., noise-free links). In addition, the combination weights $\{a, b\}$ in (4.58)-(4.59) satisfy the properties we derived in the previous section. For example, when two nodes have the same noise levels ($r = s = 1$), we have $a = b$ and $a \leq 0.5$ with equality if, and only if, $t = 0$. To see how well the approximate solution matches with the optimal solution, we illustrate how much the weights in (4.58)-(4.59) deviate from equation (4.37) in Fig. 4.1 by showing $|h(a, b)|^2$ over

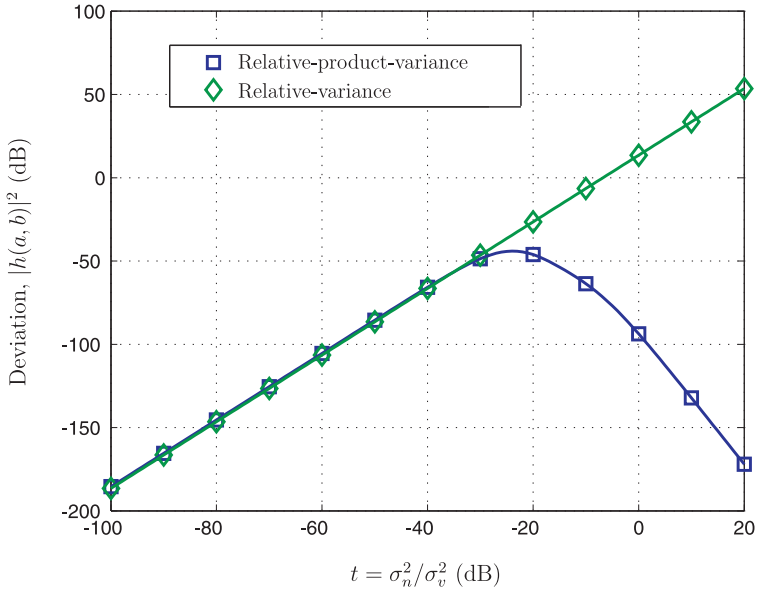


Figure 4.1: Deviation of the approximate solutions from the optimal solution, as a function of t.

different values of t. We set $\mu = 0.05$, $\sigma_u^2 = 1$, $r = 2$, and $s = 1$. We also show the curve with the weights in (4.39) where we ignored the noisy links. We observe that the weights in (4.58)-(4.59) match equation (4.37), whereas the equation with the weights in (4.39) increases rapidly after some value of t.

4.5 Adaptive Combination Policy

To evaluate the relative-product-variance rule (4.55), every agent needs to have information about the step-sizes $\{\mu_l\}$ and the second-order statistics $\{\sigma_{v,l}^2, R_{u,l}, R_{n,l,k}\}$ in its neighborhood. However, the latter quantities are usually not available and may be even varying over time. Therefore, an adaptive solution, where every agent is able to learn its combination weights using available instantaneous data, is desirable. One adaptive solution for the combination weights was proposed ear-

lier in [33, 141], albeit for the case of noise-free exchange links. Before presenting the new solution, we summarize the solution of [141] for comparison purposes. Let n_k denote the number of neighbors of agent k. Introduce the $M \times N$ matrix $\Delta\Phi_{k,i}$, formed by the differences of the intermediate estimates available at agent k, i.e., the lth column of $\Delta\Phi_{k,i}$ has the form

$$[\Delta\Phi_{k,i}]_l = \begin{cases} \psi_{k,i} - \psi_{k,i-1}, & \text{if } l = k \\ \phi_{l,k,i} - \phi_{l,k,i-1}, & \text{if } l \in \mathcal{N}_k \setminus \{k\} \\ 0, & \text{otherwise} \end{cases} \tag{4.60}$$

Then, the combination weights $\{a_{l,k}(i)\}$ are updated according to the rule

$$a_{l,k}(i) = \begin{cases} a_{l,k}(i-1) - \beta_k(i) g_{l,k}(i), & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{4.61}$$

where $\beta_k(i)$ is used to guarantee the non-negativeness of $\{a_{l,k}(i)\}$ and has the form

$$\beta_k(i) = \nu_k \cdot \frac{\min\{a_{l,k}(i-1), l \in \mathcal{N}_k\}}{\max\{g_{l,k}(i), l \in \mathcal{N}_k\} + \varepsilon} \tag{4.62}$$

with a positive step-size ν_k for agent k and a small positive constant ε to prevent singularity. Moreover, the quantity $g_{l,k}(i)$ in (4.61) is the lth entry of the vector $g_{k,i}$, which is evaluated by the relation

$$g_{k,i} = \left(I_N - \frac{\mathbb{1}_N \mathbb{1}_N^T}{n_k} \right) \mathrm{Re}\left\{ (\Delta\Phi_{k,i})^* \, \Delta\Phi_{k,i} \right\} a_{k,i} \tag{4.63}$$

where $a_{k,i}$ denotes the kth column of the matrix A. In addition, to guarantee that the entries of $a_{k,i}$ add up to one, the initial values, $\{a_{k,-1}\}$, should be chosen such that $a_{k,-1}^T \mathbb{1}_N = 1$ for all k. One simple way to do this is to use the uniform combination rule, which has the form

$$a_{l,k}(-1) = \begin{cases} 1/n_k, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{4.64}$$

That is, agent k simply averages the intermediate estimates from its neighbors.

Here, motivated by the relative-product-variance rule from (4.55), we propose an alternative and effective adaptive solution for adjusting the combination weights; this solution takes into account the presence of information exchange noise over the communication links [174]. Recalling the ATC diffusion strategy (4.1) and the data model (2.3), we can write the intermediate estimates $\{\boldsymbol{\psi}_{k,i}, \boldsymbol{\phi}_{l,k,i}\}$ at agent k as:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^{*}[\boldsymbol{u}_{k,i}\tilde{\boldsymbol{w}}_{k,i-1} + \boldsymbol{v}_k(i)] \tag{4.65}$$

$$\boldsymbol{\phi}_{l,k,i} = \boldsymbol{w}_{l,i-1} + \mu_l \boldsymbol{u}_{l,i}^{*}[\boldsymbol{u}_{l,i}\tilde{\boldsymbol{w}}_{l,i-1} + \boldsymbol{v}_l(i)] + \boldsymbol{n}_{l,k,i} \tag{4.66}$$

In view of the independence assumptions on the regression data and noise sources, we obtain

$$\lim_{i\to\infty} \mathbb{E}\|\boldsymbol{\psi}_{k,i} - \boldsymbol{w}_{k,i-1}\|^2 = \mu_k^2 \sigma_{v,k}^2 \mathrm{Tr}(R_{u,k}) + \mu_k^2 \cdot \Big(\lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i-1}\|^2_{\mathbb{E}(\boldsymbol{u}_{k,i}^{*}\|\boldsymbol{u}_{k,i}\|^2\boldsymbol{u}_{k,i})}\Big) \tag{4.67}$$

$$\begin{aligned}\lim_{i\to\infty} \mathbb{E}\|\boldsymbol{\phi}_{l,k,i} - \boldsymbol{w}_{l,i-1}\|^2 &= \mu_l^2 \sigma_{v,l}^2 \mathrm{Tr}(R_{u,l}) + \mathrm{Tr}(R_{n,l,k}) \\ &\quad + \mu_l^2 \cdot \Big(\lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{l,i-1}\|^2_{\mathbb{E}(\boldsymbol{u}_{l,i}^{*}\|\boldsymbol{u}_{l,i}\|^2\boldsymbol{u}_{l,i})}\Big)\end{aligned} \tag{4.68}$$

We can evaluate the limits on the right-hand side of (4.67)-(4.68) by using the steady-state result from (4.8):

$$\lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_{\sigma}^2 = \lim_{i\to\infty} \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 + \big[\mathrm{vec}(\mathcal{Y}^T)\big]^T \sigma + \big[\mathrm{vec}(\mathcal{Z}^T)\big]^T \sigma \tag{4.69}$$

and by setting the vector σ in (4.69) satisfying:

$$(I - \mathcal{F})\sigma = \mathrm{vec}\left[J_l \otimes \mathbb{E}(\boldsymbol{u}_{l,i}^{*}\|\boldsymbol{u}_{l,i}\|^2\boldsymbol{u}_{l,i})\right] \tag{4.70}$$

where J_l is an $N \times N$ diagonal matrix with all diagonal entries equal to zero

except the lth diagonal entry, which is equal to one. Then, from (4.69),

$$\begin{aligned}&\lim_{i\to\infty}\mathbb{E}\|\tilde{\boldsymbol{w}}_{l,i-1}\|^2_{\mathbb{E}(\boldsymbol{u}^*_{l,i}\|\boldsymbol{u}_{l,i}\|^2\boldsymbol{u}_{l,i})}\\&=\left[\mathrm{vec}(\mathcal{Y}^T_{\mathrm{atc}}+\mathcal{Z}^T_{\mathrm{atc}})\right]^T(I-\mathcal{F})^{-1}\mathrm{vec}\left[J_l\otimes\mathbb{E}(\boldsymbol{u}^*_{l,i}\|\boldsymbol{u}_{l,i}\|^2\boldsymbol{u}_{l,i})\right]\end{aligned}\tag{4.71}$$

From the terms $\mathcal{Y}_{\mathrm{atc}}$ and $\mathcal{Z}_{\mathrm{atc}}$ in Tables 2.2 and 4.1, the limit in (4.71) is of the order of $\mu_l^2\sigma_{v,l}^2\mathrm{Tr}(R_{u,l})+\mathrm{Tr}(R_{n,l,k})$. We first assume that $\mu_l^2\sigma_{v,l}^2\mathrm{Tr}(R_{u,l})$ and $\mathrm{Tr}(R_{n,l,k})$ have the same order or that $\mu_l^2\sigma_{v,l}^2\mathrm{Tr}(R_{u,l})\gg\mathrm{Tr}(R_{n,l,k})$ so that $\mathrm{Tr}(R_{n,l,k})$ is ignorable (e.g., noise-free links). Then, the limits on the right hand side of (4.67)-(4.68) can be ignored because of the factors μ_k^2 and μ_l^2 in front of them. It follows that expressions (4.67)-(4.68) can be approximated to:

$$\lim_{i\to\infty}\mathbb{E}\|\boldsymbol{\psi}_{k,i}-\boldsymbol{w}_{k,i-1}\|^2\approx\mu_k^2\sigma_{v,k}^2\mathrm{Tr}(R_{u,k})=\gamma_{k,k}^2\tag{4.72}$$

$$\lim_{i\to\infty}\mathbb{E}\|\boldsymbol{\phi}_{l,k,i}-\boldsymbol{w}_{l,i-1}\|^2\approx\mu_l^2\sigma_{v,l}^2\mathrm{Tr}(R_{u,l})+\mathrm{Tr}(R_{n,l,k})=\gamma_{l,k}^2\tag{4.73}$$

Using the following instantaneous approximation at agent k:

$$\mathbb{E}\|\boldsymbol{\psi}_{k,i}-\boldsymbol{w}_{k,i-1}\|^2\approx\|\psi_{k,i}-w_{k,i-1}\|^2\tag{4.74}$$

$$\mathbb{E}\|\boldsymbol{\phi}_{l,k,i}-\boldsymbol{w}_{l,i-1}\|^2\approx\|\phi_{l,k,i}-w_{k,i-1}\|^2\tag{4.75}$$

where $w_{l,i-1}$ is replaced by $w_{k,i-1}$ in (4.75), we can motivate an algorithm that enables agent k to estimate the variance product measure $\gamma_{l,k}^2$ of its neighbor l. Thus, let $\gamma_{l,k}^2(i)$ denote an estimate for $\gamma_{l,k}^2$ that is computed by agent k at time i. Then, one way to evaluate $\gamma_{l,k}^2(i)$ is through the following recursion:

$$\gamma_{l,k}^2(i)=\begin{cases}(1-\nu_k)\cdot\gamma_{k,k}^2(i-1)+\nu_k\cdot\|\psi_{k,i}-w_{k,i-1}\|^2, & \text{if } l=k\\(1-\nu_l)\cdot\gamma_{l,k}^2(i-1)+\nu_l\cdot\|\phi_{l,k,i}-w_{k,i-1}\|^2, & \text{if } l\in\mathcal{N}_k\setminus\{k\}\end{cases}\tag{4.76}$$

Note that under expectation, expression (4.76) becomes

$$\mathbb{E}\boldsymbol{\gamma}_{l,k}^2(i)=\begin{cases}(1-\nu_k)\cdot\mathbb{E}\boldsymbol{\gamma}_{k,k}^2(i-1)+\nu_k\cdot\mathbb{E}\|\boldsymbol{\psi}_{k,i}-\boldsymbol{w}_{k,i-1}\|^2, & \text{if } l=k\\(1-\nu_l)\cdot\mathbb{E}\boldsymbol{\gamma}_{l,k}^2(i-1)+\nu_l\cdot\mathbb{E}\|\boldsymbol{\phi}_{l,k,i}-\boldsymbol{w}_{k,i-1}\|^2, & \text{if } l\in\mathcal{N}_k\setminus\{k\}\end{cases}\tag{4.77}$$

so that in steady-state (as $i \to \infty$):

$$\lim_{i\to\infty} \mathbb{E}\boldsymbol{\gamma}_{l,k}^2(i) \approx (1-\nu_l) \cdot \lim_{i\to\infty} \mathbb{E}\boldsymbol{\gamma}_{l,k}^2(i-1) + \nu_l \cdot \gamma_{l,k}^2 \tag{4.78}$$

Hence, we obtain

$$\lim_{i\to\infty} \mathbb{E}\boldsymbol{\gamma}_{l,k}^2(i) \approx \gamma_{l,k}^2 \tag{4.79}$$

That is, the estimate $\gamma_{l,k}^2(i)$ converges on average to the desired variance product $\gamma_{l,k}^2$. In this way, we can replace the combination rule in (4.55) by the adaptive implementation:

$$a_{l,k}(i) = \begin{cases} \gamma_{l,k}^{-2}(i) / \sum_{j\in\mathcal{N}_k} \gamma_{j,k}^{-2}(i), & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{4.80}$$

Equations (4.76) and (4.80) provide an adaptive implementation for the combination weights $\{a_{l,k}\}$. We now examine the case when the information exchange noise dominates the noise sources, i.e., $\mu_l^2\sigma_{v,l}^2 \mathrm{Tr}(R_{u,l}) \ll \mathrm{Tr}(R_{n,l,k})$. From (4.67)-(4.68), we observe that in this case, we have

$$\lim_{i\to\infty} \mathbb{E}\|\boldsymbol{\psi}_{k,i} - \boldsymbol{w}_{k,i-1}\|^2 \ll \lim_{i\to\infty} \mathbb{E}\|\boldsymbol{\phi}_{l,k,i} - \boldsymbol{w}_{l,i-1}\|^2 \tag{4.81}$$

so that $a_{l,k} \approx \delta_{lk}$. That is, when the information exchange noise dominates the noise sources, the ATC diffusion strategy degenerates to the non-cooperative operation. Note that the same degeneration is observed for the relative-product-variance rule (4.55) when $\mu_l^2\sigma_{v,l}^2 \mathrm{Tr}(R_{u,l}) \ll \mathrm{Tr}(R_{n,l,k})$.

We remark that the proposed adaptive solution in (4.76) and (4.80) applies to the CTA diffusion strategy with the terms $\psi_{k,i} - w_{k,i-1}$ and $\phi_{l,k,i} - w_{k,i-1}$ replaced by $w_{k,i} - \psi_{k,i-1}$ and $w_{k,i} - \phi_{l,k,i-1}$, respectively. Moreover, the proposed adaptive solution has simpler structure than (4.61). For example, expression (4.76) requires n_k vector multiplications at agent k to compute the $\{\gamma_{l,k}^2(i)\}$, rather than at least n_k^2 vector multiplications to obtain $g_{k,i}$ in (4.63). Moreover, the

Table 4.2: Combination rules used in the simulations

Name	**Rule**
Relative-product-variance	(4.55)
Proposed adaptive	(4.76) and (4.80)
Relative-variance [124, 152, 174]	(4.57)
Adaptive solution of [141]	(4.61)
Uniform [124]	(4.64)
Metropolis [167]	$a_{l,k} = \begin{cases} 1 - \sum_{j \neq k} a_{k,j}, & \text{if } l = k \\ 1/\max\{n_k, n_l\}, & \text{if } l \in \mathcal{N}_k \setminus \{k\} \\ 0, & \text{otherwise} \end{cases}$
No-cooperation	$a_{l,k} = \delta_{kl}$

solution in (4.76) and (4.80) only needs the estimate $w_{k,i-1}$ at the previous time instant and does not require $\psi_{k,i-1}$ and $\{\phi_{l,k,i-1}\}$ to establish $\Delta\Phi_{k,i}$ in (4.60). In the next section, we further illustrate by simulations that the proposed adaptive combination rule achieves lower MSD than the solution of [141].

4.6 Simulation Results

In this section, we present simulation results and compare several combination rules with the network MSD as a measure. The combination rules are summarized in Table 4.2. The network wants to estimate a 10×1 unknown vector w° (i.e., $M = 10$) using the ATC diffusion strategy (4.1). The covariance matrices for the information exchange noise have the form $R_{n,l,k} = \sigma_{n,l}^2 I_{10}$. In addition, the step-sizes are the same for all agents ($\mu_k = 0.05$ and $\nu_k = 0.3$ for all k).

We first consider the two-node case with regression covariance matrix $R_{u,k} =$

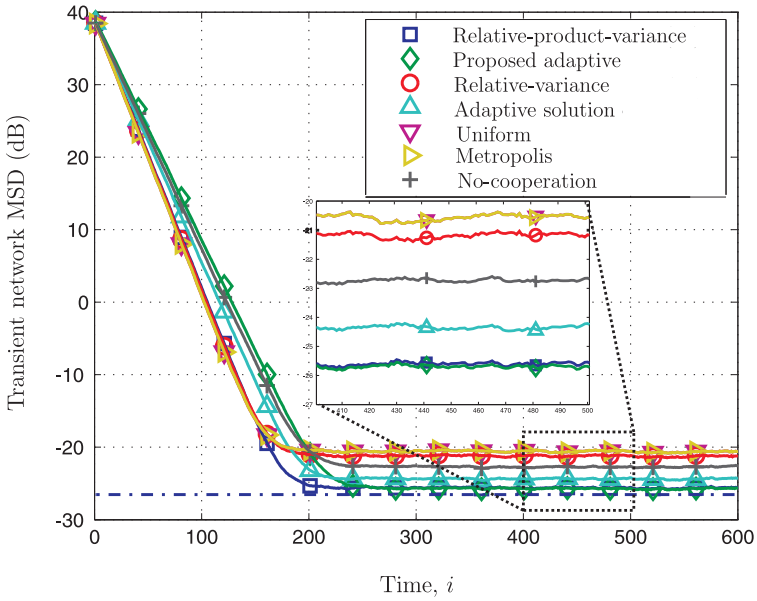


Figure 4.2: Transient network MSD over time. The dashed line indicates the theoretical steady-state MSD (4.11).

I_{10} and the variances of the noise sources set to $\sigma_v^2 = 0.01$ and $\sigma_n^2 = 4 \times 10^{-4}$ (namely, $t = -14$ dB from (4.31)). In Fig. 4.2, we show the transient network MSD over time. We observe that in the presence of the noisy links, the network without cooperation can possibly outperform the network with cooperation if the nodes ignore the information exchange noise (i.e., using the relative-variance rule from (4.57)). We also see that the relative-variance rule from (4.55) and the proposed adaptive combination rule have similar performance and perform the best in comparison to the other choices. The theoretical steady-state network MSD from (4.11) for the relative-variance rule from (4.55) is also shown in Fig. 4.2 and matches well with simulations.

To examine how the information exchange noise affects the combination weights and the network MSD, we show the optimal combination weights $\{a, b\}$ and the corresponding MSD as a function of t in the two-node case in Fig. 4.3. We first set $r = s = 1$ and compare two values of step-sizes: $\mu = 0.01$ and $\mu = 0.05$. In this case, we have $a = b$ and the values of a can be evaluated by solving the fifth-order function in (4.41). We observe that the nodes are sensitive to the information exchange noise, especially for a small step-size, and a drops significantly when t is small. Using (4.41), we can find the value of t when $a = 0.25$, where a drops to half of its initial optimal value, and obtain $t = -48.6$ dB for $\mu = 0.01$ and $t = -28.2$ dB for $\mu = 0.05$. In the right plot of Fig. 4.3, the optimal MSD is compared to the MSD with the relative-variance rule from (4.57) ($a = b = 1/2$), which are the optimal solution when there are no link errors. The figure shows bifurcation. When the ratio t is small, these two networks have similar performance. However, after some critical value of t, the MSD of the network using the relative-variance rule increases linearly in t, whereas the optimal MSD increases slightly and saturates (converges to the network without cooperation). We also see that the critical value of t is smaller for a smaller step-size. The results point

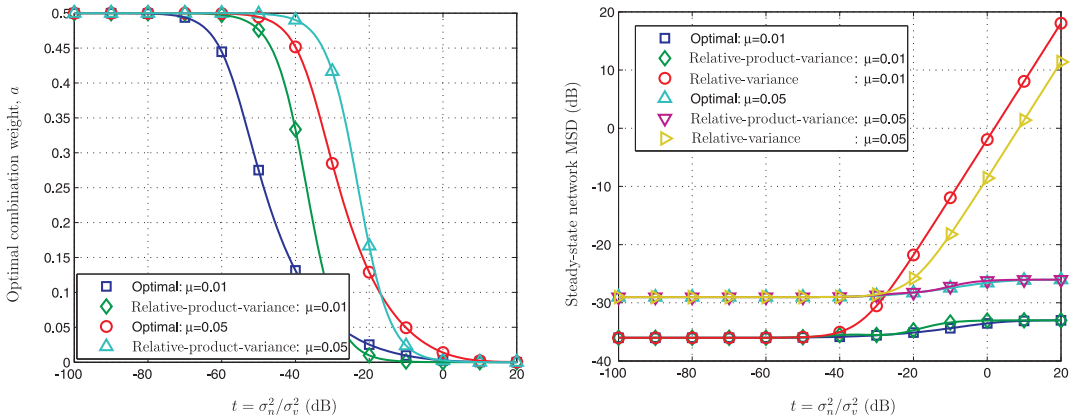


Figure 4.3: Optimal combination weight a (left) and steady-state network MSD (right) as a function of t.

to a trade-off in choosing the step-size parameter: smaller step-sizes achieve lower MSD but are more sensitive to link errors. We also show the relative-variance rule from (4.55) and its corresponding MSD (see again Fig. 4.3). The network MSD is close to the optimal MSD. The results reveals that the relative-variance rule from (4.55) is a good approximation for the optimal combination weights.

Similar curves are observed in Fig. 4.4, where we examine the situation when $r = 2$ and $s = 1$. That is, node 2 has better quality of measurement than node 1 while the noise levels due to information exchange are the same for both nodes. The optimal MSD is compared to the MSD with relative-variance rule from (4.57), i.e., $(a, b) = (2/3, 1/3)$. It is interesting to note from the left plot of Fig. 4.4 that node 1 places more weight on its own estimate (i.e., $a < 0.5$) when t is about -30 dB. The results indicate that a node should place more weight on its estimate in the combination step (4.1c) even if the node has worse quality of measurement than its neighbors. We further examine two adaptive combination rules. Fig. 4.5 shows the transient behavior of one of the combination weights, a. We also show

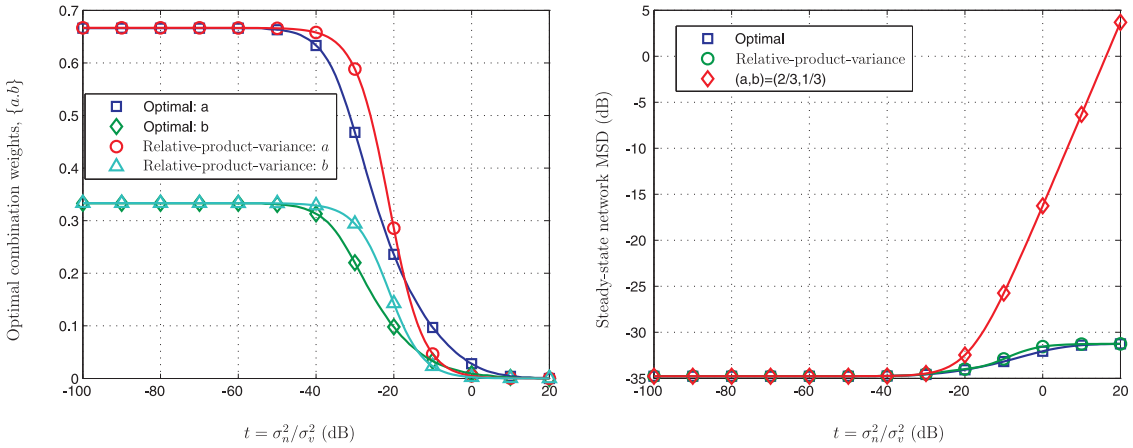


Figure 4.4: Optimal combination weights $\{a, b\}$ (left) and steady-state network MSD (right) over t.

the relative-variance rule from (4.55) and optimal weight ($a = 0.143$ from Fig. 4.4) for reference. We observe that the proposed adaptive solution converges, and the steady-state value is closer to the optimal value of a, compared to the adaptive solution in (4.61).

Next, we consider a network with 20 nodes in Fig. 3.4. The covariance matrices for the regressors are diagonal matrices with diagonal entries randomly generated from $[0.5, 2.5]$, and the noise profiles are randomly generated with $\sigma_{v,k}^2$ and $\sigma_{n,k}^2$ uniformly distributed over $[-30, -10]$ and $[-40, -25]$ dB, respectively. We show the transient network MSD over time in Fig. 4.6 and observe that the proposed combination rules perform the best in comparison to the other choices in the presence of the information exchange noise. We further show the steady-state MSD at each node in Fig. 4.7. We observe that the proposed combination rules improve the MSD at each node, compared to the non-cooperative combination rule. However, the adaptive solution in (4.61) may degrade performance at some nodes (see nodes 18 and 20 in Fig. 4.7) since the nodes may fail to adjust their

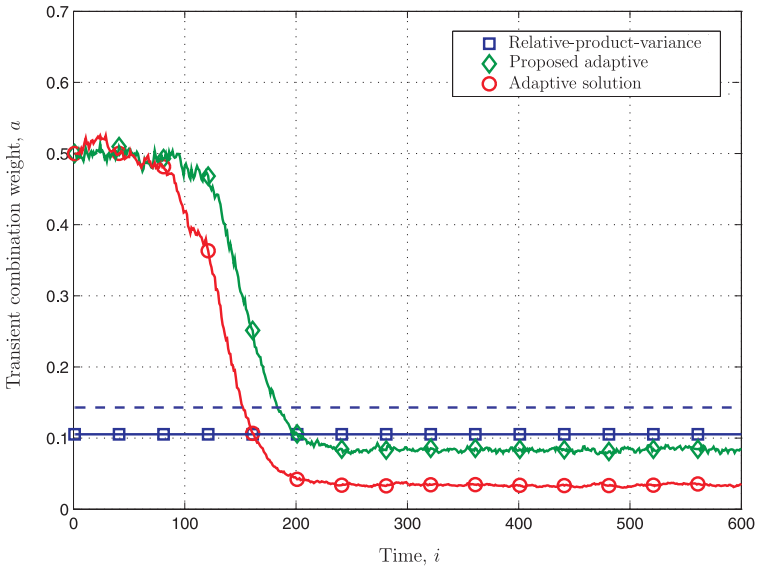


Figure 4.5: Transient combination weight a over time. The dashed line indicates the optimal combination weight.

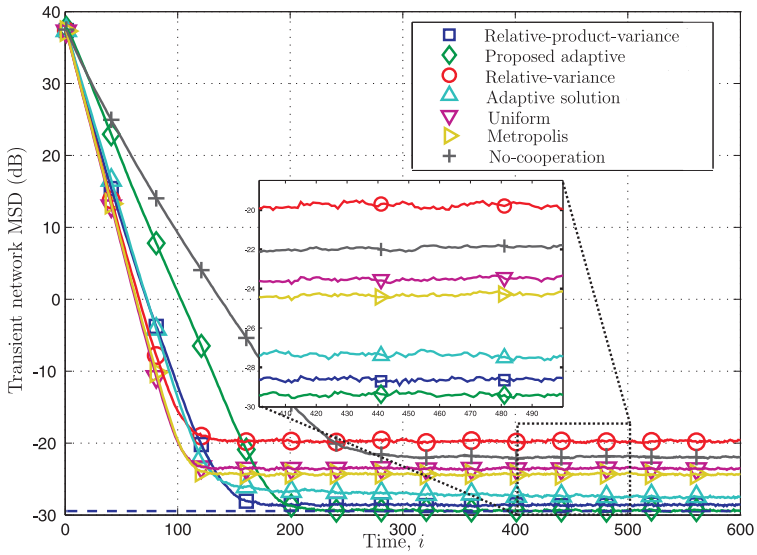


Figure 4.6: Transient network MSD over time with information exchange noise. The dashed line indicates the theoretical steady-state MSD (4.11).

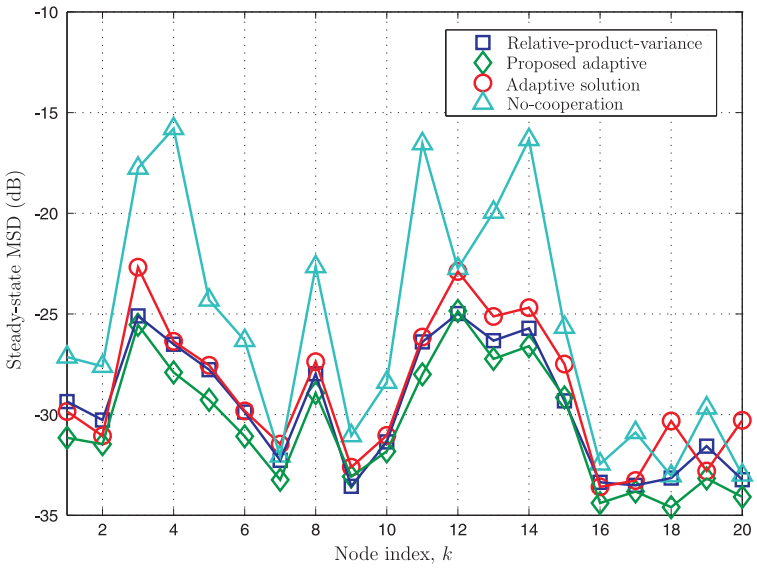


Figure 4.7: Steady-state MSD at the nodes of the network.

combination weights correctly in the presence of the noisy links.

In practice, the regressors $\{\boldsymbol{u}_{k,i}\}$ may not be independent over time. We introduce the time-correlation of $\boldsymbol{u}_{k,i}$ by means of the first-order auto-regressive model. That is, the $\boldsymbol{u}_{k,i}$ is generated by the following rule:

$$\boldsymbol{u}_{k,i} = \alpha \cdot \boldsymbol{u}_{k,i-1} + \sqrt{1-\alpha^2} \cdot \boldsymbol{r}_{k,i} \tag{4.82}$$

where $\alpha \in (0,1)$ is a constant and $\boldsymbol{r}_{k,i}$ is a random vector of size M. Moreover, $\boldsymbol{r}_{k,i}$ is assumed to be independent of $\boldsymbol{u}_{k,i-1}$ and has zero mean and covariance matrix $R_{u,k}$. It can be verified that $\boldsymbol{u}_{k,i}$ given by (4.82) is zero mean and has covariance matrix $R_{u,k}$. The transient network MSD over time is shown in Fig. 4.8 with $\alpha = 0.7$. We observe that the proposed combination rules perform the best and that the analytical results for the network MSD (4.11) still holds when the regressors are time-dependent.

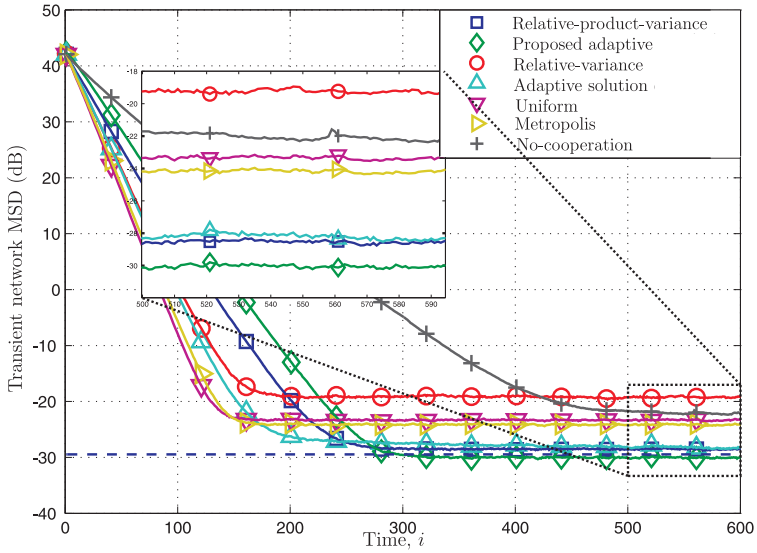


Figure 4.8: Transient network MSD over time with information exchange noise. The dashed line indicates the theoretical steady-state MSD (4.11).

4.7 Concluding Remarks

In this chapter, we studied adaptive networks with noisy information exchange. The results show that, with noise from information exchange, a node may need to place more weight on its own estimate during the aggregation and consultation step, even if the quality of the data from the node is worse than that of its neighbors. We derived optimal constructions for the combination weights and showed how these weights can be adapted as well, in order to counter the degradation caused by noisy links. Simulation results show that the proposed rules not only improve the network MSD, but they also improve the MSD at all nodes in the network, compared to the non-cooperation combination rule.

4.A Proof of Lemma 4.1

1. Taking the derivative of (4.35) with respect to a, we obtain

$$\frac{\partial \mathrm{MSD}_{\mathrm{cta}}}{\partial a} = M\sigma_v^2\eta\Bigg\{ta\left(\frac{1}{1-\eta}+\frac{1}{1-\eta(1-2a)^2}\right) - [\mu(1-\eta)+2t\eta a^2]\frac{1-2a}{[1-\eta(1-2a)^2]^2}\Bigg\} \tag{4.83}$$

Then,

$$\left.\frac{\partial \mathrm{MSD}_{\mathrm{cta}}}{\partial a}\right|_{a=0} = M\sigma_v^2\eta\frac{-\mu}{1-\eta} < 0 \tag{4.84}$$

$$\left.\frac{\partial \mathrm{MSD}_{\mathrm{cta}}}{\partial a}\right|_{a=1} = M\sigma_v^2\eta[\mu(1-\eta)+2t]\frac{1}{(1-\eta)^2} > 0 \tag{4.85}$$

2. From the expression (4.41) for $f(a)$ we get:

$$f(0) = -\mu(1-\eta)^2 < 0 \tag{4.86}$$

$$f(0.5) = \frac{1}{2}(2-\eta)t \geq 0 \tag{4.87}$$

Note that since $0 < \eta < 1$, we would obtain $f(0.5) = 0$ if, and only if, $t = 0$.

3. Some algebra shows that the function $f(a)$ can be rewritten as

$$\begin{aligned} f(a) =& 2\eta(1-\eta)ta(1-a)(2a-1)\left[(2a-1)^2+2a\right] \\ &+ 16\eta t a^3(1-a)^2 + 2(1-\eta)ta + \mu(1-\eta)^2(2a-1) \end{aligned} \tag{4.88}$$

Therefore, for $0.5 < a \leq 1$, $f(a) > 0$.

4. Taking the derivative of $f(a)$ and rearranging the result, we obtain

$$\begin{aligned} f'(a) =& 4\eta(1-\eta)ta\left[(1-2a)^2(3-5a)+2-a\right] \\ &+ 16\eta t a^2(1-a)(3-5a) + 2(1-\eta)^2(t+\mu) \end{aligned} \tag{4.89}$$

It is clear that $f'(a) > 0$ if $0 \leq a \leq 0.5$.

4.B Proof of Theorem 4.3

From Lemma 4.2, it suffices to verify that

$$\rho\left(\mathcal{B}_{\text{atc}}^{*j}\mathcal{B}_{\text{atc}}^{j}\right) \leq N \cdot [\rho(I_{NM} - \mathcal{M}\mathcal{R})]^{2j} \tag{4.90}$$

Let

$$\mathcal{B} \triangleq \mathcal{B}_{\text{atc}} = \mathcal{A}^T(I_{NM} - \mathcal{M}\mathcal{R}) \tag{4.91}$$

$$\mathcal{C} \triangleq r \cdot \mathcal{A}^T \tag{4.92}$$

where r is defined as:

$$r \triangleq \rho(I_{NM} - \mathcal{M}\mathcal{R}) = \max_{1\leq k\leq N} \rho(I_M - \mu_k R_{u,k}) \tag{4.93}$$

We first show the following inequality holds for any integer j:

$$\left\|\mathcal{B}^j\right\|_b \leq \left\|\mathcal{C}^j\right\|_b = r^j \cdot \|(A^T)^j\|_\infty \tag{4.94}$$

By definition of the block matrix norm in (3.67), it suffices to show that

$$\left\|\left[\mathcal{B}^j\right]_{k,l}\right\|_2 \leq \left\|\left[\mathcal{C}^j\right]_{k,l}\right\|_2 \tag{4.95}$$

where $[\mathcal{B}^j]_{k,l}$ denotes the (k,l)th block (of size $M \times M$) of the matrix $\mathcal{B}^j$. By the rules of matrix multiplication, $[\mathcal{B}^j]_{k,l}$ is given by:

$$\left[\mathcal{B}^j\right]_{k,l} = \sum_{m_1=1}^{N}\sum_{m_2=1}^{N}\cdots\sum_{m_{j-1}=1}^{N} \mathcal{B}_{k,m_1}\mathcal{B}_{m_1,m_2}\cdots\mathcal{B}_{m_{j-1},l} \tag{4.96}$$

where $\mathcal{B}_{k,l}$ is the (k,l)th block (of size $M \times M$) of the matrix $\mathcal{B}$ from (4.91) and is given by

$$\mathcal{B}_{k,l} = a_{l,k} \cdot (I_M - \mu_l R_{u,l}) \tag{4.97}$$

Then, by the triangular inequality and submultiplicative property of norms, the 2-induced norm of $[\mathcal{B}^j]_{k,l}$ in (4.96) is bounded by:

$$\left\|\left[\mathcal{B}^j\right]_{k,l}\right\|_2 \leq \sum_{m_1=1}^{N}\sum_{m_2=1}^{N}\cdots\sum_{m_{j-1}=1}^{N} \|\mathcal{B}_{k,m_1}\|_2 \cdot \|\mathcal{B}_{m_1,m_2}\|_2 \cdots \|\mathcal{B}_{m_{j-1},l}\|_2 \tag{4.98}$$

Note from (4.97) that

$$\begin{aligned}\|\mathcal{B}_{k,l}\|_2 &= \|a_{l,k}\cdot(I_M-\mu_l R_{u,l})\|_2\\ &= a_{l,k}\cdot\rho(I_M-\mu_l R_{u,l})\\ &\le a_{l,k}\cdot r\end{aligned}\tag{4.99}$$

Then, combining (4.98) with (4.99), we have

$$\begin{aligned}\left\|\left[\mathcal{B}^j\right]_{k,l}\right\|_2 &\le r^j\cdot\sum_{m_1=1}^{N}\sum_{m_2=1}^{N}\cdots\sum_{m_{j-1}=1}^{N}a_{m_1,k}\cdot a_{m_2,m_1}\cdots a_{l,m_{j-1}}\\ &= \left\|\left[\mathcal{C}^j\right]_{k,l}\right\|_2\end{aligned}\tag{4.100}$$

and arrive at inequality (4.94). Likewise, it can be verified that

$$\left\|\mathcal{B}^{*j}\right\|_b \le \left\|\mathcal{C}^{*j}\right\|_b = r^j\cdot\|A^j\|_\infty\tag{4.101}$$

Then, from (4.94) and (4.101), we have

$$\begin{aligned}\rho\left(\mathcal{B}^{*j}\mathcal{B}^j\right) &\le \left\|\mathcal{B}^j_{\text{atc}}\right\|_b\cdot\left\|\mathcal{B}^{*j}_{\text{atc}}\right\|_b\\ &\le \left\|\mathcal{C}^j\right\|_b\cdot\left\|\mathcal{C}^{*j}\right\|_b\\ &= r^{2j}\cdot\left\|(A^T)^j\right\|_\infty\cdot\left\|A^j\right\|_\infty\end{aligned}\tag{4.102}$$

Since $\left\|(A^T)^j\right\|_\infty = 1$ and

$$\begin{aligned}\left\|A^j\right\|_\infty &= \left\|(A^T)^j\right\|_1\\ &\le N\cdot\left\|(A^T)^j\right\|_\infty\\ &= N\end{aligned}\tag{4.103}$$

then we arrive at (4.90).

CHAPTER 5

Role of Informed Agents

In previous works on distributed estimation over adaptive networks [34, 50, 73, 95, 124, 127, 129, 134, 140], and in other related studies on distributed algorithms [6, 11, 42, 65, 78, 94, 96, 114], on consensus/gossip strategies [8, 22, 68, 109, 122, 170], and on distributed optimization [37, 104, 133], the agents are usually assumed to be homogeneous in that they all have similar processing capabilities and are able to have continuous access to information and measurements. However, it is generally observed in nature that the behavior of a biological network is often driven more heavily by a small fraction of the agents as happens, for example, with bees and fish [7, 117, 130]. This phenomenon motivates us to study adaptive networks where only a *fraction* of the agents are assumed to be informed, while the remaining agents are uninformed [151, 157]. Informed agents collect data regularly and perform in-network processing tasks, while uninformed agents only participate in consultation tasks in the manner explained in the sequel.

Specifically, we examine the performance of diffusion strategies for distributed estimation and adaptation over networks. We examine how the transient and steady-state behavior of the network are dependent on its topology and on the *proportion* of the informed agents and their distribution in space. The results reveal some interesting and surprising trade-offs between convergence rate and mean-square performance. In particular, among other results, the analysis will show that the mean-square performance of adaptive networks does not necessar-

ily improve with a larger proportion of informed agents. Instead, it is discovered that if the set of informed agents is enlarged, the convergence rate of the network becomes faster albeit at the expense of some deterioration in mean-square performance. The results also establish that uninformed agents play an important role in determining the steady-state performance of the network, and that it is preferable to maintain some of the highly noisy or highly connected agents uninformed. The analysis in the chapter reveals the important interplay that exists among three factors: the number of informed agents in a network, the convergence rate of the learning process, and the estimation accuracy. We illustrate the findings by considering two topology models from the complex network literature [20, 105], namely, the Erdos-Renyi and scale-free models.

To arrive at the aforementioned results, a detailed mean-square-error analysis of the network behavior is pursued. However, the difficulty of the analysis is compounded by the fact that agents interact with each other and, therefore, they influence each other's learning process and performance. Nevertheless, for sufficiently small step-sizes and for combination matrices that are either symmetric or close-to-symmetric, we will be able to derive an expression for the network MSD. By examining this expression, we will establish that the MSD is influenced by the eigen-structure of two matrices: the covariance matrix representing the data statistical profile and the combination matrix representing the network topology. We then study the eigen-structure of these matrices and derive useful approximate expressions for their eigenvalues and eigenvectors. The expressions end up revealing that the network MSD can be decomposed into two components. We study the behavior of each component as a function of the proportion of informed agents; both components show important differences in their behavior. When the components are added together, a picture emerges that shows how the performance of the network depends on the proportion of informed agents

in an manner that supports analytically the popular wisdom that *more information is not necessarily better* [139]. We further explain that the deterioration in mean-square-error performance is mainly caused by the inappropriate selection of combination weights by the agents. Therefore, the results highlight the importance of the manner in which the agents in the network fuse the information from their neighbors; otherwise, the mean-square-error performance will suffer from additional information.

5.1 Uninformed Agents

To model uninformed agents over the network, we set $\mu_k = 0$ if agent k is uninformed in the ATC diffusion strategy (2.29). We assume that the network contains at least one informed agent. In this model, uninformed agents do not collect data $\{d_k(i), u_{k,i}\}$ (or simply set data to zero all the time) and, therefore, do not perform the adaptation step (the first step in (2.29)); they, however, continue to perform the combination or consultation step (the second step in (2.29)). In this way, informed agents have access to data and participate in the adaptation and consultation steps, whereas uninformed agents play an auxiliary role through their participation in the consultation step only. This participation is nevertheless important because it helps diffuse information across the network. One of the main results of this chapter is to examine how the proportion of informed agents, and how the spatial distribution of these informed agents, influence the learning and adaptation abilities of the network in terms of its convergence rate and mean-square performance. It will follow from the analysis that uninformed agents also play an important role in determining the network performance.

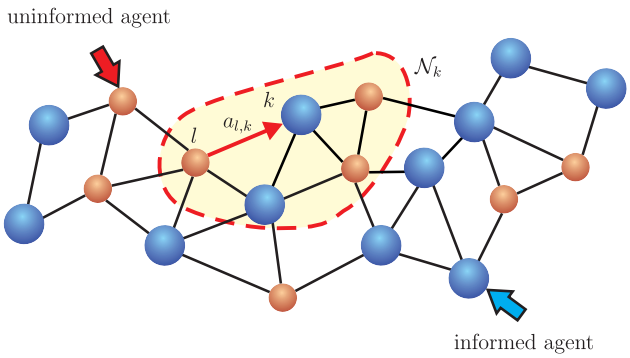


Figure 5.1: A connected network with informed and uninformed agents. The weight $a_{l,k}$ scales the data transmitted from agent l to agent k over the edge linking them.

5.1.1 Conditions for Mean and Mean-Square Stability

The mean-square performance of diffusion networks has been studied in detail in Chapters 2 for the case where all agents are informed. Expressions for the convergence rate (2.73) and the network mean-square deviation (2.82) can directly apply to the case with uninformed agents. However, the condition for mean-square stability will need to be properly adjusted as explained below in (5.2) and (5.3).

As argued in Chapter 2, for sufficiently small step-sizes, the mean and mean-square stability are guaranteed if the spectral radius of the matrix $\mathcal{B}$ is less than one. For the ATC diffusion strategy, the matrix $\mathcal{B}$ is given by:

$$\mathcal{B} = \mathcal{A}^T(I_{NM} - \mathcal{M}\mathcal{R}) \tag{5.1}$$

where $\mathcal{M}$, $\mathcal{A}$, and $\mathcal{R}$ are defined in (2.38), (2.39), and (2.48), respectively. In the following statement, we provide conditions to ensure mean and mean-square stability of the network even in the presence of uninformed agents.

Theorem 5.1 (Stability)**.** *The ATC diffusion network (2.29) with at least one informed agent converges in the mean and mean-square senses if the step-sizes* $\{\mu_k\}$ *and the combination matrix* A *satisfy the following two conditions:*

1. *For every informed agent* l*, its step-size* μ_l *satisfies:*

$$0 < \mu_l < \frac{2}{\lambda_{\max}(R_{u,l})} \tag{5.2}$$

2. *There exists a finite integer* j *such that for every agent* k*, there exists an informed agent* l *satisfying:*

$$\left[A^j\right]_{l,k} > 0 \tag{5.3}$$

That is, the (l,k)*th entry of* A^j *is positive.*

Proof. See Appendix 5.A. □

Note that condition (5.3) is automatically satisfied if the network is strongly connected (see Assumption 2.2). As such, the ATC diffusion strategy (2.29) will converge in the mean and mean-square whenever there exists at least one informed agent with its step-size satisfying condition (5.2).

5.2 Mean-Square Performance

We are now ready to examine in some detail the effect of network topology and agent distribution on the convergence rate given by (2.73) and the network MSD given by (2.87). The analysis is carried out under Assumptions 3.1 and 2.2, i.e., agents are homogeneous so that

$$\mu_k = \mu \text{ if agent } k \text{ is informed} \quad \text{and} \quad R_{u,k} = R_u \text{ for all } k \tag{5.4}$$

and the network is strongly connected so that the ATC diffusion strategy (2.29) is mean and mean-square stable. In order to reflect clearly how the network performance varies as a function of the proportion of informed agents, in the sequel, we use the eigen-decompositions of the combination matrix A (see Appendix 2.A) and the covariance matrix R_u (see Section 3.2.1) to arrive at the alternative expressions (5.37)-(5.38). To derive these alternative expressions from (2.73) and (2.87), we need two assumptions that run across our arguments, both of which are common in the literature on adaptation [62, 82, 83, 123] and distributed consensus strategies [8, 22, 109]. First, we rely on the sufficiently small step-size (see Assumption 2.1) and use it to ignore terms that depend on higher-order powers of the step-sizes. Second, we need the combination matrix A to be diagonalizable (see Assumption 3.2); this is automatically satisfied for any symmetric A (which is common, for example, in studies on consensus-type algorithms [8, 22, 109]). However, there are important non-symmetric choices for the combination matrices that are also diagonalizable that fit into our analysis, such as when uniform combination weights are used. For this reason, our analysis will not be limited to symmetric combination matrices.

5.2.1 Eigen-structure of $\mathcal{B}$

To start our expansion of expressions (2.73) and (2.87), we first examine the eigen-structure of $\mathcal{B}$. To begin with, we observe from (2.73) and (2.87) that the convergence rate and network MSD depend on the matrix $\mathcal{B}$ from (5.1) in a non-trivial manner. To gain insight into the network performance, we therefore need to examine closely the eigen-structure of $\mathcal{B}$, which is related to the combination matrix A and the covariance matrix R_u. The eigen-structure of A is summarized in Appendix 2.A. Without loss of generality, we order the eigenvalues of A^T in

decreasing order and assume

$$1 = \lambda_1(A) > |\lambda_2(A)| \geq \cdots \geq |\lambda_N(A)| \tag{5.5}$$

where the strict inequality is from Assumption 2.2 (see Appendix 2.A). To facilitate the analysis, we assume that A is close-to-symmetric, i.e., the right eigenvectors of A^T satisfy:

$$r_{l_2}^* r_{l_1} \approx \delta_{l_1 l_2}. \tag{5.6}$$

Condition (5.6) is actually automatically satisfied with an exact equality (and is not an assumption) for any symmetric A (see (2.103)). We could simplify our analysis and assume A to be symmetric throughout. However, there are important non-symmetric choices for A that are diagonalizable (such as the uniform combination matrix). This choice would be excluded from the analysis if we restrict A to being symmetric. Condition (5.6) covers all symmetric choices for A and, additionally, some useful non-symmetric choices for which the eigenvectors are closely orthogonal. The eigen-structure of the covariance matrix R_u was introduced in Section 3.2.1. Again, we arrange $\{\lambda_m(R_u)\}$ in decreasing order with

$$\lambda_{\max}(R_u) = \lambda_1(R_u) \geq \lambda_2(R_u) \geq \cdots \geq \lambda_M(R_u) = \lambda_{\min}(R_u) > 0. \tag{5.7}$$

In the sequel, for any vector x, we use the notation $x_{k:l}$ to denote a subvector of x formed from the kth up to the lth entries of x. In addition, we let $\mathcal{N}_I$ denote the set of informed agents, i.e., $k \in \mathcal{N}_I$ if agent k is informed and let N_I denote the number of informed agents in the network. Without loss of generality, we label the network agents such that the first N_I agents are informed, i.e., $\mathcal{N}_I = \{1, 2, \ldots, N_I\}$. The next result establishes a useful approximation for the eigen-structure of the matrix $\mathcal{B}$ defined in (5.1); it shows how the eigenvectors

and eigenvalues of $\mathcal{B}$ can be constructed from the eigenvalues and eigenvectors for $\{A^T, R_u\}$.

Lemma 5.1 (Eigen-structure of $\mathcal{B}$)**.** *For a ATC diffusion network (2.29) with at least one informed agent, the matrix $\mathcal{B} = \mathcal{A}^T(I - \mathcal{M}\mathcal{R})$ has the eigenvalue $\lambda_{l,m}(\mathcal{B})$ given by:*

$$\lambda_{l,m}(\mathcal{B}) = \lambda_l(A) \cdot \left[1 - \mu\lambda_m(R_u) \cdot s^*_{l,1:N_I} r_{l,1:N_I}\right] + \mathcal{O}(\mu^2) \tag{5.8}$$

for $l = 1, \ldots, N$ and $m = 1, \ldots, M$. In addition, the corresponding right and left eigenvectors $\{r^b_{l,m}, s^b_{l,m}\}$ are given by:

$$r^b_{l,m} = r_l \otimes z_m + \mu \cdot \tilde{r}^b_{l,m} + \mathcal{O}(\mu^2) \tag{5.9}$$

$$s^b_{l,m} = s_l \otimes z_m + \mu \cdot \tilde{s}^b_{l,m} + \mathcal{O}(\mu^2) \tag{5.10}$$

where $\tilde{r}^b_{l,m}$ and $\tilde{s}^b_{l,m}$ satisfy:

$$(\lambda_l(A) I_{NM} - \mathcal{A}^T)\tilde{r}^b_{l,m} = \left[(r_l \otimes z_m)(s_l \otimes z_m)^* - I_{NM}\right] \mathcal{A}^T \mathcal{D}\mathcal{R}(r_l \otimes z_m) \tag{5.11}$$

$$\tilde{s}^{b*}_{l,m}(\lambda_k(A) I_{NM} - \mathcal{A}^T) = (s_l \otimes z_m)^* \mathcal{A}^T \mathcal{D}\mathcal{R}\left[(r_l \otimes z_m)(s_l \otimes z_m)^* - I_{NM}\right] \tag{5.12}$$

The matrix $\mathcal{D}$ in (5.11)-(5.12) is defined as $\mathcal{D} \triangleq \mathcal{M}/\mu$ and is independent of μ.

Proof. See Appendix 5.C. □

5.2.2 Convergence Rate

From the eigen-structure of the matrix $\mathcal{B}$ in Lemma 5.1, we have the following useful result for the convergence rate.

Lemma 5.2 (Faster convergence rate)**.** *Consider two configurations of the same network: one with $\mathcal{N}_{I,1}$ informed agents and another with $\mathcal{N}_{I,2}$ informed agents. Let r_1 and r_2 denote the corresponding convergence rates for these two informed configurations. If $\mathcal{N}_{I,2} \supseteq \mathcal{N}_{I,1}$, then $r_2 \leq r_1$.*

Proof. See Appendix 5.B. □

The above result shows that if we enlarge the set of informed agents, the convergence rate decreases and convergence becomes faster (one important question that we consider in the next section is whether this faster convergence is occurring towards a better or worse MSD value). The following result provides bounds for the convergence rate.

Lemma 5.3 (Bound on convergence rate)**.** *The convergence rate is bounded by*

$$[1-\mu\cdot\lambda_{\min}(R_u)]^2 \leq r < 1 \tag{5.13}$$

Moreover, when all agents are informed, the convergence rate is independent of the combination matrix A and equal to the lower bound in (5.13).

Proof. Since the diffusion network is mean-square stable, i.e., $\rho(\mathcal{B}) < 1$, the upper bound is obvious. On the other hand, from Lemma 5.2, the value of $\rho(\mathcal{B})$ achieves its minimum value when all agents are informed, i.e., the matrix $\mathcal{M}$ in (2.38) becomes $\mathcal{M} = \mu I_{NM}$. In this case, the matrix $\mathcal{B}$ in (5.1) can be written as:

$$\mathcal{B}^{\circ} = A^T \otimes (I_M - \mu R_u) \tag{5.14}$$

where the superscript is used to denote the matrix $\mathcal{B}$ when all agents are informed. Then,

$$\rho(\mathcal{B}) \geq \rho(\mathcal{B}^{\circ}) = \rho(A^T)\cdot\rho(I_M - \mu R_u) \tag{5.15}$$

We already know that $\rho(A^T) = 1$. In addition, because $(I_M - \mu R_u) > 0$ in view of Assumption 2.1, we have that

$$\rho(I_M - \mu R_u) = 1 - \mu\cdot\lambda_{\min}(R_u) \tag{5.16}$$

and we arrive at the lower bound in (5.13). We also observe that the spectral radius of $\mathcal{B}^{\circ}$ is independent of A and equal to $1-\mu\cdot\lambda_{\min}(R_u)$. □

Now we are in a position to simplify the convergence rate expression (2.73) in order to highlight its dependence on the network topology more prominently. From (5.8), $|\lambda_{l,m}(\mathcal{B})|$ can be expressed as:

$$|\lambda_{l,m}(\mathcal{B})| \approx |\lambda_l(A)| \cdot |1 - \mu\lambda_m(R_u) \cdot s^*_{k,1:N_I} r_{k,1:N_I}| \tag{5.17}$$

where we ignored the term $\mathcal{O}(\mu^2)$ for sufficiently small step-sizes. Since $|\lambda_l(A)| < \lambda_1(A) = 1$ for $l > 1$, and for sufficiently small step-sizes, the maximum value of $|\lambda_{l,m}(\mathcal{B})|$ (namely, $\rho(\mathcal{B})$) occurs when $l = 1$. Recall from Appendix 2.A that, under Assumption 2.2, all entries of r_1 and s_1 are positive, it holds that

$$0 < s^T_{1,1:N_I} r_{1,1:N_I} \leq s^T_1 r_1 = 1 \tag{5.18}$$

which implies that $|\lambda_{1,m}(\mathcal{B})|$ increases as m increases (i.e. smaller $\lambda_m(R_u)$). Then, we arrive at the following expression for $\rho(\mathcal{B})$:

$$\rho(\mathcal{B}) = |\lambda_{1,M}(\mathcal{B})| \approx 1 - \mu\lambda_{\min}(R_u) \cdot s^T_{1,1:N_I} r_{1,1:N_I}. \tag{5.19}$$

The square of this expression determines the rate of convergence of the diffusion strategy (2.29). Note that expression (5.19) satisfies Lemmas 5.2 and 5.3. Also, since all entries of r_1 and s_1 are positive, we conclude that $\rho(\mathcal{B}) < 1$ and the network is mean-square stable.

5.2.3 Simplifying the MSD Expression (2.87)

Expression (2.87) relates the network MSD to the eigen-structure of $\mathcal{B}$ from Lemma 5.1 and the matrix $\mathcal{Y}$, which is given by

$$\mathcal{Y} = \mathcal{A}^T \mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A} \tag{5.20}$$

where $\mathcal{S}$ is defined in (2.56). These quantities contain information about the data statistical profile, the spatial distribution of informed agents, and the network

topology through their dependence on $\{\mathcal{R}, \mathcal{M}, \mathcal{A}\}$. From (5.9) and assumption (5.6), we have that

$$\begin{aligned} r_{l,n}^{b*} r_{k,m}^{b} &= [(r_l \otimes z_n) + \mathcal{O}(\mu)]^*[(r_k \otimes z_m) + \mathcal{O}(\mu)] \\ &= (r_l^* r_k) \otimes (z_n^* z_m) + \mathcal{O}(\mu) \\ &\approx \delta_{kl} \cdot \delta_{mn} \end{aligned} \tag{5.21}$$

where we ignored the terms depending on μ under sufficiently small step-sizes. Then, expression (2.87) simplifies to:

$$\text{MSD} \approx \sum_{l=1}^{N} \sum_{m=1}^{M} \frac{s_{l,m}^{b*} \mathcal{Y} s_{l,m}^{b}}{N \cdot [1 - |\lambda_{l,m}(\mathcal{B})|^2]}. \tag{5.22}$$

Expression (5.22) can be simplified further once we evaluate the term in the numerator. Since the agents are homogeneous from (5.4), we can express the matrix $\mathcal{Y}$ from (5.20) as:

$$\mathcal{Y} = \mathcal{Z} \Omega^{-1} \mathcal{Z}^* \tag{5.23}$$

where

$$\mathcal{Z} = \mathcal{A}^T \mathcal{M} \mathcal{R} \tag{5.24}$$

$$\begin{aligned} \Omega &= \text{diag}\{\sigma_{v,1}^{-2} R_u, \sigma_{v,2}^{-2} R_u, \ldots, \sigma_{v,N}^{-2} R_u\} \\ &= \Sigma_v^{-1} \otimes R_u \end{aligned} \tag{5.25}$$

with Σ_v defined in (3.24). Then, we get

$$s_{l,m}^{b*} \mathcal{Y} s_{l,m}^{b} = \| s_{l,m}^{b*} \mathcal{Z} \Omega^{-1/2} \|^2. \tag{5.26}$$

Note that the matrix $\mathcal{Z}$ in (5.24) can be written as:

$$\begin{aligned} \mathcal{Z} &= \sum_{k=1}^{N} \lambda_k(A) (r_k \otimes I_M)(s_k^* \otimes I_M) \text{diag}\{I_{N_I} \otimes \mu R_u, I_{N-N_I} \otimes 0_M\} \\ &= \sum_{k=1}^{N} \lambda_l(A) (r_k \otimes I_M) \begin{bmatrix} s_{k,1:N_I}^* \otimes \mu R_u & s_{k,N_I+1:N}^* \otimes 0_M \end{bmatrix}. \end{aligned} \tag{5.27}$$

We then obtain from the approximate expression for $s_{l,m}^{b*}$ in (5.10) and relations (5.24) and (5.27) that:

$$\begin{aligned} s_{l,m}^{b*}\mathcal{Z}\Omega^{-1/2} &= \sum_{k=1}^{N}\lambda_k(A)\cdot[(s_l\otimes z_m)+\mathcal{O}(\mu)]^*(r_k\otimes I_M) \\ &\quad\times\begin{bmatrix} s_{k,1:N_I}^*\otimes\mu R_u & s_{k,N_I+1:N}^*\otimes 0_M\end{bmatrix}\Omega^{-1/2} \\ &= \lambda_l(A)\cdot(1\otimes z_m^*)\begin{bmatrix} s_{l,1:N_I}^*\Sigma_{v,1:N_I}^{1/2}\otimes\mu R_u^{1/2} & 0_{1\times(N-N_I)M}\end{bmatrix}+\mathcal{O}(\mu^2) \\ &\approx \lambda_l(A)\cdot\begin{bmatrix} s_{l,1:N_I}^*\Sigma_{v,1:N_I}^{1/2}\otimes\mu\lambda_m^{1/2}(R_u)z_m^* & 0_{1\times(N-N_I)M}\end{bmatrix} \end{aligned} \tag{5.28}$$

where we used the fact that $s_{l_2}^* r_{l_1} = \delta_{l_1 l_2}$ from (2.101) and ignored the terms depending on μ^2 under sufficiently small step-sizes. Therefore, the term $s_{l,m}^{b*}\mathcal{Y}s_{l,m}^{b}$ in (5.26) becomes

$$\begin{aligned} s_{l,m}^{b*}\mathcal{Y}s_{l,m}^{b} &= \left(s_{l,m}^{b*}\mathcal{Z}\Omega^{-1/2}\right)\left(s_{l,m}^{b*}\mathcal{Z}\Omega^{-1/2}\right)^* \\ &\approx \mu^2\lambda_m(R_u)|\lambda_l(A)|^2\cdot s_{l,1:N_I}^*\Sigma_{v,1:N_I}s_{l,1:N_I} \end{aligned} \tag{5.29}$$

where we used that $\|z_m\|^2 = 1$. Then, substituting (5.8) and (5.29) into (5.22), we arrive at the following expression for the MSD in terms of the eigenvalues and eigenvectors of A^T and the eigenvalues of R_u.

Theorem 5.2 (Network MSD)**.** *The network MSD of the ATC diffusion strategy (2.29) can be approximately expressed as*

$$\text{MSD}\approx\sum_{l=1}^{N}\sum_{m=1}^{M}\frac{\mu^2\lambda_m(R_u)|\lambda_l(A)|^2\cdot s_{l,1:N_I}^*\Sigma_{v,1:N_I}s_{l,1:N_I}}{N\left[1-|\lambda_l(A)|^2\cdot\left|1-\mu\lambda_m(R_u)\cdot s_{l,1:N_I}^* r_{l,1:N_I}\right|^2\right]} \tag{5.30}$$

Since the matrix A has a single eigenvalue at $\lambda_1(A)=1$, and its value is greater than the remaining eigenvalues, we can decompose the MSD in (5.30) into two components. The first component is determined by $\lambda_1(A)$, i.e., $l=1$ in (5.30), and is denoted by $\text{MSD}_{\ell=1}$. The second component is due to the contribution from the remaining eigenvalues of A, i.e., $l>1$ in (5.30), and is denoted by

$\text{MSD}_{\ell>1}$. Since $\lambda_1(A) = 1$, and for sufficiently small step-sizes, we introduce the approximation for the denominator in (5.30):

$$\begin{aligned} &|\lambda_1(A)|^2 \cdot \left|1 - \mu\lambda_m(R_u) \cdot s^*_{1,1:N_I} r_{1,1:N_I}\right|^2 \\ &\approx 1 - 2\mu\lambda_m(R_u) \cdot s^T_{1,1:N_I} r_{1,1:N_I}. \end{aligned} \tag{5.31}$$

Then, the term $\text{MSD}_{\ell=1}$ becomes

$$\text{MSD}_{\ell=1} \approx \frac{M\mu}{2N} \cdot \frac{\sum_{k=1}^{N_I} \sigma^2_{v,k} s^2_{1,k}}{\sum_{k=1}^{N_I} r_{1,k} s_{1,k}} \tag{5.32}$$

where $\{r_{l,k}, s_{l,k}\}$ denote the kth entries of $\{r_l, s_l\}$. Expression (5.32) reveals several interesting properties. First, we observe that the term $\text{MSD}_{\ell=1}$ does not depend on the matrix R_u, which is also a property of the MSD expression for stand-alone adaptive filters [123]. Second, *if the number of informed agents increases by one, the value of* $\text{MSD}_{\ell=1}$ *may increase or decrease* (i.e., it does not necessarily decrease). This can be seen as follows. From (5.32) we see that $\text{MSD}_{\ell=1}$ will decrease (and, hence, improve) only if

$$\frac{\sum_{k=1}^{N_I} \sigma^2_{v,k} s^2_{1,k} + \sigma^2_{v,N_I+1} s^2_{1,N_I+1}}{\sum_{k=1}^{N_I} r_{1,k} s_{1,k} + r_{1,N_I+1} s_{1,N_I+1}} < \frac{\sum_{k=1}^{N_I} \sigma^2_{v,k} s^2_{1,k}}{\sum_{k=1}^{N_I} r_{1,k} s_{1,k}} \tag{5.33}$$

or, if the noise variance of the added agent satisfies:

$$\sigma^2_{v,N_I+1} s_{1,N_I+1} < \frac{\sum_{k=1}^{N_I} \sigma^2_{v,k} s^2_{1,k}}{\sum_{k=1}^{N_I} s_{1,k}} \tag{5.34}$$

where we used the fact that $r_1 = \mathbb{1}_N/\sqrt{N}$ from (2.102).

For the second part, $\text{MSD}_{\ell>1}$, since $|\lambda_l(A)| < 1$ for $l > 1$, and for sufficiently small step-sizes, the denominator in (5.30) can be approximated by:

$$1 - |\lambda_l(A)|^2 \cdot \left|1 - \mu\lambda_m(R_u) \cdot s^*_{l,1:N_I} r_{l,1:N_I}\right|^2 \approx 1 - |\lambda_l(A)|^2. \tag{5.35}$$

Comparing to (5.31), we further ignore the term $2\mu\lambda_m(R_u)|\lambda_l(A)|^2 \cdot s^*_{l,1:N_I} r_{l,1:N_I}$ in (5.35) since this term is generally much less than $1 - |\lambda_l(A)|^2$ due to small

step-size. Then, $\text{MSD}_{\ell>1}$ becomes

$$\text{MSD}_{\ell>1} \approx \frac{\mu^2 \text{Tr}(R_u)}{N} \sum_{l=2}^{N} \left[\frac{|\lambda_l(A)|^2}{1-|\lambda_l(A)|^2} \cdot \sum_{k=1}^{N_I} \sigma_{v,k}^2 |s_{l,k}|^2 \right] \tag{5.36}$$

It is important to note that, in contrast to $\text{MSD}_{\ell=1}$ in (5.32), $\text{MSD}_{\ell>1}$ *in (5.36) always increases (i.e., worse mean-square performance) when the number of informed agents increases.*

5.2.4 Behavior of the Network

Combining expressions (5.19), (5.32), and (5.36), we arrive at the following result for diffusion networks.

Theorem 5.3 (Diffusion networks)**.** *The ATC diffusion network (2.29) has approximate convergence rate:*

$$r \approx \left(1 - \mu \lambda_{\min}(R_u) \cdot \sum_{k \in \mathcal{N}_I} s_{1,k} r_{1,k} \right)^2 \tag{5.37}$$

and approximate network MSD:

$$\text{MSD} \approx \underbrace{\frac{M\mu}{2N} \cdot \frac{\sum_{k \in \mathcal{N}_I} \sigma_{v,k}^2 s_{1,k}^2}{\sum_{k \in \mathcal{N}_I} r_{1,k} s_{1,k}}}_{\text{MSD}_{\ell=1}} + \underbrace{\frac{\mu^2 \text{Tr}(R_u)}{N} \sum_{l=2}^{N} \left[\frac{|\lambda_l(A)|^2}{1-|\lambda_l(A)|^2} \cdot \sum_{k \in \mathcal{N}_I} \sigma_{v,k}^2 |s_{l,k}|^2 \right]}_{\text{MSD}_{\ell>1}}. \tag{5.38}$$

Note that the summations in (5.37) and (5.38) are over the set of informed agents, $\mathcal{N}_I$. Expressions (5.37) and (5.38) reveal important information about the behavior of the network. As the set of informed agents, $\mathcal{N}_I$, increases, we observe from (5.37) that the rate of convergence becomes faster (a desirable effect). However, as we will illustrate in simulations, the behavior of the terms $\text{MSD}_{\ell=1}$ and $\text{MSD}_{\ell>1}$ ends up causing the network MSD given by (5.38) to increase (an

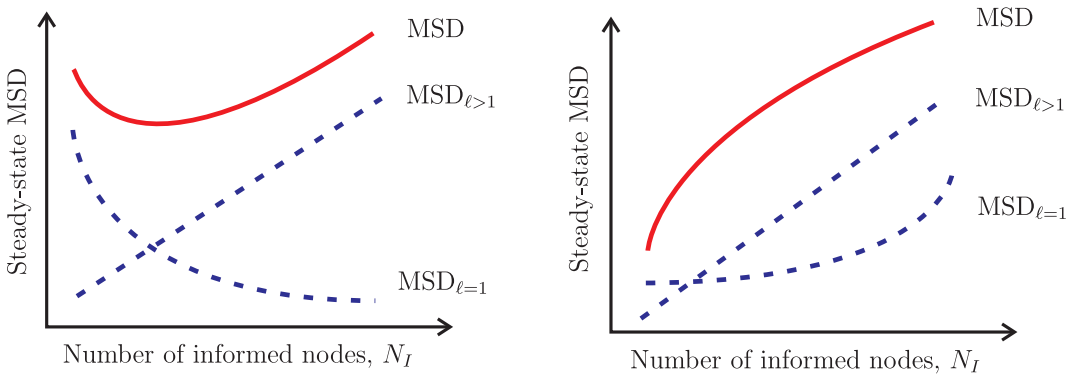


Figure 5.2: Sketch of the behavior of the network MSD as a function of the number of informed agents, N_I, depending on whether relation (5.34) is satisfied (left) or not (right).

undesirable effect) as $\mathcal{N}_I$ increases. Figure 5.2 illustrates the two possible trends in the behavior of the network MSD and its components, $\text{MSD}_{\ell=1}$ and $\text{MSD}_{\ell>1}$. Two scenarios are shown in the figure corresponding to the case whether the added informed agents satisfy (5.34) or not. The figure shows that depending on condition (5.34), the curve for $\text{MSD}_{\ell=1}$ can increase or decrease with N_I. Nevertheless, the overall network MSD generally increases (i.e., becomes worse) with increasing N_I. These facts reveal an important trade-off between the convergence rate and the network MSD in relation to the proportion of informed agents. We summarize the behavior of the diffusion network in Table 5.1 and show how the rate of convergence and the MSD respond when the parameters $\{N_I, \text{Tr}(R_u)\}$ increase. We remark that slower convergence rate and worse estimation correspond to increasing values of r and MSD (an undesirable effect).

The fact that the network MSD ends up increasing as the set of informed agents is enlarged is an interesting phenomenon. In the following, we show that the deterioration of the network MSD can be controlled through proper selection

Table 5.1: Behavior of the diffusion network in response to increases in any of the parameters $\{N_I, \mathrm{Tr}(R_u)\}$.

	$N_I \uparrow$	$\mathrm{Tr}(R_u) \uparrow$
convergence rate r (5.76)	faster	faster
MSD (5.77)	worse in general	worse
$\mathrm{MSD}_{\ell=1}$ (5.55)	may be better or worse (see (5.34))	independent of $\mathrm{Tr}(R_u)$
$\mathrm{MSD}_{\ell>1}$ (5.54)	worse	worse

of the combination weights. To see this, we assume the step-size is small enough so that the term $\mathrm{MSD}_{\ell>1}$ in (5.38), which is of the order of μ^2, can be ignored and that the network MSD from (5.38) is close to $\mathrm{MSD}_{\ell=1}$. In addition, we assume the agents select the combination weights to satisfy

$$s_{1,k} = \frac{\sqrt{N} \cdot \sigma_{v,k}^{-2}}{\sum_{j=1}^{N} \sigma_{v,j}^{-2}} \tag{5.39}$$

That is, the kth entry of the left eigenvector s_1 of A^T associated with the eigenvalue one is inversely proportional to the noise variance at agent k. One possible selection of combination rules that satisfy this property is the so-called Hastings rule [173]:

$$a_{l,k} = \begin{cases} \sigma_{v,k}^2 \,/\max\{n_k\sigma_{v,k}^2, n_l\sigma_{v,l}^2\}\,, & l \in \mathcal{N}_k \setminus \{k\} \\ 1 - \sum_{l\in\mathcal{N}_k\setminus\{k\}} a_{l,k}, & l = k \end{cases} \tag{5.40}$$

In this case and by the fact that $r_1 = \mathbb{1}_N/\sqrt{N}$, the convergence rate and network

MSD from (5.37)-(5.38) will be given by:

$$r \approx \left(1 - \mu\lambda_{\min}(R_u) \cdot \frac{\sum_{k\in\mathcal{N}_I} \sigma_{v,k}^{-2}}{\sum_{k=1}^{N} \sigma_{v,k}^{-2}}\right)^2 \tag{5.41}$$

$$\text{MSD} \approx \frac{M\mu}{2} \cdot \frac{1}{\sum_{k=1}^{N} \sigma_{v,k}^{-2}}. \tag{5.42}$$

We see that when the agents employ the Hastings combination rule, the network MSD remains constant, while the convergence rate decreases (becomes faster), as the set of informed agents is enlarged. The result highlights the importance of selecting combination weights; otherwise, the mean-square performance may suffer from additional information.

5.3 Network Behavior under Uniform Combinations

As shown by (5.37)-(5.38), the convergence rate and network MSD depend strongly on the eigenvalues and eigenvectors of the combination matrix A. In this section, we examine more closely the eigen-structure of A when it is chosen as the following uniform combination matrix:

$$a_{l,k} = \begin{cases} 1/n_k, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{5.43}$$

The uniform combination matrix is generally non-symmetric; nevertheless, it is diagonalizable under certain assumption on the network topology. In addition to Assumption 2.2, we further introduce the following assumption.

Assumption 5.1 (Undirected networks)**.** *The network topology undirected (where if agent l is a neighbor of agent k, then agent k is also a neighbor of agent l).*

Assumption 5.1 does not imply that the combination matrix A is symmetric; although agents (k, l) are neighbors of each other, they can still assign different

weights to the information shared over their links, i.e., $a_{l,k}$ can be different from $a_{k,l}$. In this case, agent k would assign a different weight to the information received from agent l, than the weight assigned by agent l to the information received from agent k. We first verify the following result.

Lemma 5.4 (Diagonalization of uniform combination matrix)**.** *Under Assumptions 5.1, the matrix A defined by (5.43) is diagonalizable and has real eigenvalues. Moreover, condition (5.6) is satisfied by the eigenvectors of A when all agents have approximately similar degrees, i.e., when $n_k \approx \eta$ where η denotes the degree of the network and is defined as*

$$\eta \triangleq \frac{1}{N}\sum_{k=1}^{N} n_k. \tag{5.44}$$

Proof. We introduce the degree matrix, D, and the adjacency matrix, C, of the network graph, whose entries are defined as follows:

$$D = \mathrm{diag}\{n_1, \ldots, n_N\} \quad \text{and} \quad [C]_{k,l} = \begin{cases} 1, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}. \tag{5.45}$$

Then, it is straightforward to verify that the matrix A^T in (5.43) can be written as:

$$A^T = D^{-1}C \tag{5.46}$$

which shows that A^T is similar to the real-valued matrix A_s defined by:

$$\begin{aligned} A_s &\triangleq D^{1/2}A^T D^{-1/2} \\ &= D^{-1/2}CD^{-1/2} \end{aligned} \tag{5.47}$$

where $D^{1/2} = \mathrm{diag}\{\sqrt{n_1}, \ldots, \sqrt{n_N}\}$. Since the topology is assumed to be undirected, the matrix C is symmetric, and so is A_s. Therefore, there exists an

orthogonal matrix, U_s, and a diagonal matrix with real diagonal entries, Λ, such that

$$A_s = U_s \Lambda U_s^T. \tag{5.48}$$

From (5.47), we let

$$U = D^{-1/2} U_s \quad \text{and} \quad U^{-1} = U_s^T D^{1/2} \tag{5.49}$$

and we obtain (2.96).

Note that since the matrices U_s and $D^{1/2}$ in (5.49) are real-valued, so are the eigenvectors of the uniform combination matrix, $\{r_l, s_l\}$. Furthermore, from (5.49), we can express $\{r_l, s_l\}$ in terms of the eigenvectors of A_s defined in (5.47). Let r_l^s denote the lth eigenvector of A_s and let $r_{l,k}^s$ denote the kth entry of r_l^s. Then, we have

$$r_{l,k} = \frac{r_{l,k}^s}{c_l \cdot \sqrt{n_k}} \quad \text{and} \quad s_{l,k} = c_l \cdot \sqrt{n_k} \cdot r_{l,k}^s \tag{5.50}$$

where c_l is used to normalize the norm of r_l so that condition (2.100) is satisfied. Then, when $n_k \approx \eta$, the right eigenvectors $\{r_l\}$ of the uniform combination matrix defined by (5.43) satisfy condition (5.6) since

$$\begin{aligned} r_{l_2}^T r_{l_1} &= \sum_{k=1}^{N} \frac{r_{l_2,k}^s r_{l_1,k}^s}{c_{l_2} \cdot c_{l_1} \cdot n_k} \\ &\approx \frac{1}{c_{l_1} \cdot c_{l_2} \cdot \eta} \sum_{k=1}^{N} r_{l_2,k}^s r_{l_1,k}^s \\ &= \frac{1}{c_{l_1}^2 \cdot \eta} \delta_{l_1 l_2}. \end{aligned} \tag{5.51}$$

To normalize the norms of $\{r_l\}$, we select

$$c_l = \frac{1}{\sqrt{\eta}} \text{ for all } l. \tag{5.52}$$

□

5.3.1 Convergence Rate Expression

For the uniform combination matrix A in (5.43), it can be verified that the right eigenvector for A_s defined in (5.47) corresponding to the eigenvalue one has the following form:

$$r_1^s = \frac{1}{\sqrt{N\eta}}\text{col}\{\sqrt{n_1}, \sqrt{n_2}, \ldots, \sqrt{n_N}\} \tag{5.53}$$

we obtain from (5.19), (5.50), and (5.53) that

$$\rho(\mathcal{B}) = 1 - \mu\lambda_{\min}(R_u) \cdot \frac{\sum_{k=1}^{N_I} n_k}{N\eta}. \tag{5.54}$$

Expression (5.54) can be motivated intuitively by noting that the decay of $\rho(\mathcal{B})$ will be larger as informed agents have higher degrees. Simulations further ahead show that expression (5.54) matches well with simulated results.

5.3.2 Expression for $\text{MSD}_\ell = 1$

From (5.32), $\text{MSD}_{\ell=1}$ depends on the eigenvectors $\{r_1, s_1\}$. From (5.50), (5.52), and (5.53), expression (5.32) becomes

$$\text{MSD}_{\ell=1} \approx \frac{M\mu}{2N} \cdot \frac{\sum_{k=1}^{N_I} \sigma_{v,k}^2 n_k^2}{\eta \sum_{k=1}^{N_I} n_k}. \tag{5.55}$$

Note that expression (5.55) is inversely proportional to the degree of the network, η. That is, when the network is more connected (i.e. higher network degree), the network will have lower $\text{MSD}_{\ell=1}$. Moreover, expression (5.55) depends on the distribution of *informed* agents through its dependence on the degree and noise profile of the informed agents.

5.3.3 Eigenvalues of Uniform Combinations

Before we proceed to the expression for $\text{MSD}_{\ell>1}$, we examine the eigenvalues of the uniform combination matrix A from (5.43). There are useful results in the

literature on the spectral properties of complex networks [43, 44, 51, 54, 58], such as networks corresponding to the Erdos-Renyi and scale free models. We shall use these results to infer properties about the spectral distribution of the uniform combination matrix. First, we note from (2.25) that one is an eigenvalue of A, i.e., $\rho(A) = \lambda_1(A) = 1$. In the following, we use the results of [44] to characterize the remaining eigenvalues (namely, $\lambda_l(A)$ for $l > 1$) of the uniform combination matrix.

Theorem 5.4 (Eigenvalue distribution of uniform combination matrix)**.** *Let $\bar{n}_k$ denote the average degree of agent k in a random graph. Let*

$$\bar{\eta} \triangleq \frac{1}{N}\sum_{k=1}^{N} \bar{n}_k \tag{5.56}$$

denote the average degree of the graph. Then, for random graphs with expected degrees satisfying

$$\bar{n}_{\min} \triangleq \min_{1\leq k\leq N}\{\bar{n}_k\} \gg \sqrt{\bar{\eta}} \tag{5.57}$$

the density function, $f(\lambda)$, of the eigenvalues of A converges in probability, as $N \to \infty$, to the semicircle law (see Fig. 5.3), i.e.,

$$f(\lambda) = \begin{cases} \frac{2}{\pi R}\sqrt{1-\left(\frac{\lambda}{R}\right)^2}, & \text{if } \lambda \in [-R, R] \\ 0, & \text{otherwise} \end{cases} \tag{5.58}$$

where

$$R = \frac{2}{\sqrt{\bar{\eta}}}. \tag{5.59}$$

Moreover, if $\bar{n}_{\min} \gg \sqrt{\bar{\eta}}\log^3(N)$, the second largest eigenvalue of A converges almost surely to

$$|\lambda_2(A)| = R. \tag{5.60}$$

Proof. See Thms. 5 and 6 in [44]. □

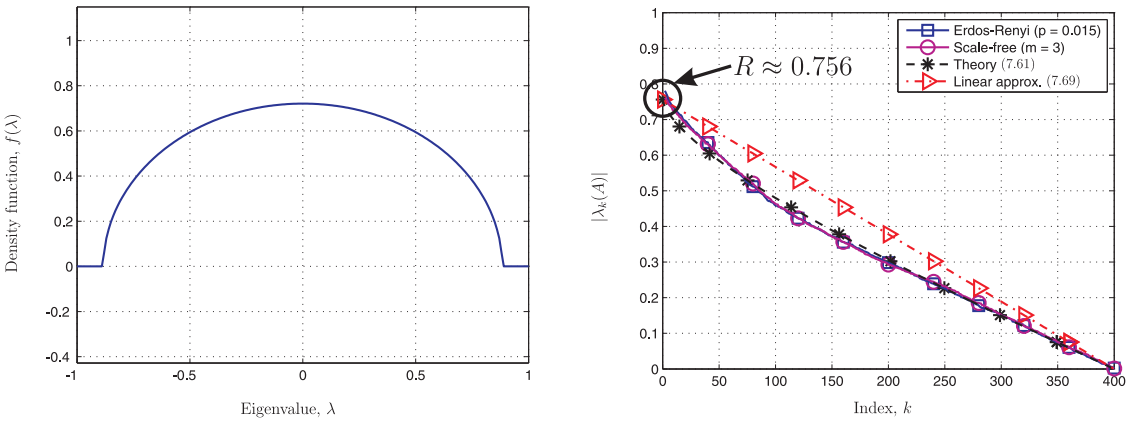


Figure 5.3: Density function (left) for the eigenvalues of A as given by (5.58) for $N \to \infty$, and the eigenvalues (right) of the combination matrix A defined by (5.43) with $N = 400$ and $\eta = 7$. The dashed line on the right represents theory from (5.62) and the dash-dot line represents linear approximation given further ahead by (5.70).

Simulations in Fig. 5.3 show that expressions (5.58) and (5.60) provide accurate approximations for the Erdos-Renyi and scale-free network models. In addition, for ergodic distributions, the value of $\bar{\eta}$ in (5.56) will be close to its realization η for large N. In the following, we determine an expression for $|\lambda_l(A)|$ by using (5.58). To do so, we let l denote the number of eigenvalues of A that are greater than some value y in magnitude for $0 \leq y \leq R$. Then, the value of l is given by:

$$\begin{aligned} l &= N \cdot \left[1 - \int_{-y}^{y} f(\lambda) d\lambda\right] \\ &\triangleq N \cdot g(y) \end{aligned} \tag{5.61}$$

where we denote the expression inside the brackets by $g(y)$. Note that the integral $\int_{-y}^{y} f(\lambda)d\lambda$ in (5.61) computes the proportion of eigenvalues of A within the region $[-y, y]$. Then, the lth eigenvalue of A can be approximated by evaluating the value of y in (5.61), i.e.,

$$|\lambda_l(A)| \approx g^{-1}\left(\frac{l}{N}\right). \tag{5.62}$$

From (5.58) and using the change of variables $\lambda/R = \sin\theta$, we obtain that $g(y)$ in (5.61) has the form:

$$g(y) = 1 - \frac{2}{\pi}\sin^{-1}\left(\frac{y}{R}\right) - \frac{2}{\pi}\frac{y}{R}\sqrt{1 - \left(\frac{y}{R}\right)^2}. \tag{5.63}$$

In Fig. 5.3, we show the eigenvalue distribution of $|\lambda_l(A)|$ for Erdos-Renyi and scale-free models. We observe that for both network models, the theoretical results in (5.60) and (5.62) match well with simulations.

5.3.4 Expression for $\text{MSD}_\ell > 1$

For $\text{MSD}_{\ell>1}$, we apply relations (5.50) and (5.52). Then, expression (5.36) can be approximated by:

$$\text{MSD}_{\ell>1} \approx \frac{\mu^2 \text{Tr}(R_u)}{N\eta} \sum_{l=2}^{N} \left[\frac{\lambda_l^2(A)}{1-\lambda_l^2(A)} \cdot \left(\sum_{k=1}^{N_I} \sigma_{v,k}^2 n_k \cdot (r_{l,k}^s)^2 \right) \right]. \tag{5.64}$$

Expression (5.64) requires knowledge of the eigenvectors $\{r_l^s\}$ of A_s in (5.47). Note that for $l = 1$ and from (5.53), we have

$$(r_{1,k}^s)^2 = \frac{n_k}{N\eta} \approx \frac{1}{N} \tag{5.65}$$

since, by assumption, $n_k \approx \eta$. We are therefore motivated to introduce the following approximation:

$$(r_{l,k}^s)^2 \approx \frac{1}{N} \tag{5.66}$$

for all l. Observe that expression (5.66) is independent of l, and we find that expression (5.64) simplifies to:

$$\text{MSD}_{\ell>1} \approx \frac{\mu^2 \text{Tr}(R_u)}{N\eta} \cdot \left(\sum_{k=1}^{N_I} \sigma_{v,k}^2 n_k \right) \cdot \frac{1}{N} \sum_{l=2}^{N} \frac{\lambda_l^2(A)}{1-\lambda_l^2(A)}. \tag{5.67}$$

Furthermore, from (5.62), we can approximate the summation over l in (5.67) by the following integral:

$$\frac{1}{N} \sum_{l=2}^{N} \frac{\lambda_l^2(A)}{1-\lambda_l^2(A)} \approx \int_0^1 \frac{[g^{-1}(x)]^2}{1-[g^{-1}(x)]^2} dx \tag{5.68}$$

where we replaced l/N by x. However, evaluating the integral in (5.68) is generally intractable. We observe though from the right plot in Fig. 5.3 that $|\lambda_l(A)|$ (and also $g^{-1}(l/N)$) decreases in a rather linear fashion for $l > 1$. Note that the function $g(y)$ in (5.63) has values 1 at $y = 0$ and 0 at $y = R \approx 2/\sqrt{\eta}$. We therefore approximate $g(y)$ by the linear function

$$g(y) \approx 1 - \frac{\sqrt{\eta}}{2} y. \tag{5.69}$$

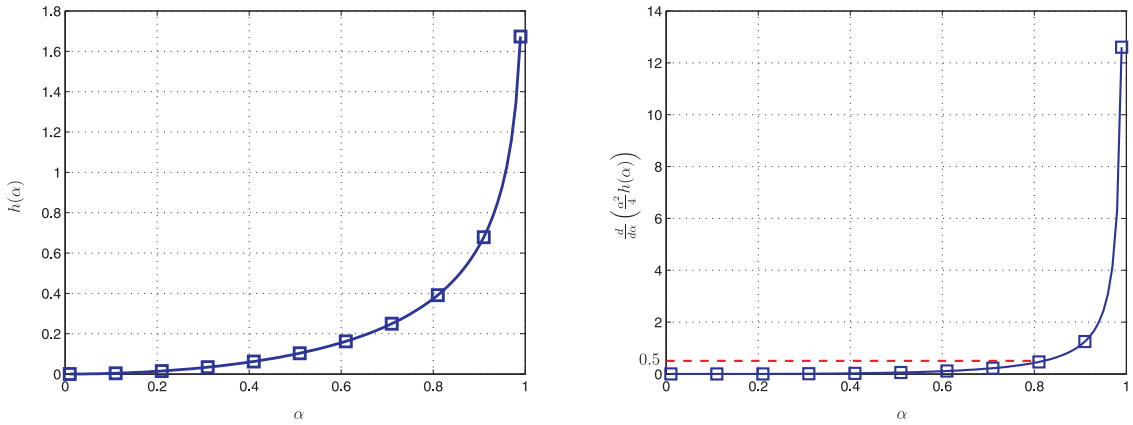


Figure 5.4: The function $h(\alpha)$ (left) from (5.72) and the derivative of $\alpha^2 h(\alpha)/4$ with respect to α (right).

Then,

$$g^{-1}(x) \approx \frac{2}{\sqrt{\eta}}(1-x) \tag{5.70}$$

and expression (5.68) becomes

$$\begin{aligned}\frac{1}{N}\sum_{l=2}^{N}\frac{\lambda_l^2(A)}{1-\lambda_l^2(A)} &\approx \int_0^1 \frac{4/\eta\cdot(1-x)^2}{1-4/\eta\cdot(1-x)^2}dx \\ &= h\left(\frac{2}{\sqrt{\eta}}\right)\end{aligned} \tag{5.71}$$

where the function $h(\alpha)$ is defined as

$$h(\alpha) \triangleq \int_0^1 \frac{\alpha^2 x^2}{1-\alpha^2 x^2}dx = \left[\frac{1}{2\alpha}\log\left(\frac{1+\alpha}{1-\alpha}\right)-1\right] \tag{5.72}$$

for $\alpha \in (0,1)$. Substituting expression (5.71) into (5.67), we find that the MSD contributed by the remaining terms ($l>1$) has the following form:

$$\text{MSD}_{\ell>1} \approx \frac{\mu^2 \text{Tr}(R_u)}{N\eta}\cdot\left(\sum_{k=1}^{N_I}\sigma_{v,k}^2 n_k\right)\cdot h\left(\frac{2}{\sqrt{\eta}}\right). \tag{5.73}$$

Note that the function $h(\alpha)$, shown in Fig. 5.4, has the following property.

Lemma 5.5. *The function $h(\alpha)$ defined in (5.72) is strictly increasing and convex in $\alpha \in (0,1)$.*

Proof. Taking the derivative of $h(\alpha)$ in (5.72) with respect to α, we obtain:

$$\frac{dh(\alpha)}{d\alpha} = \int_0^1 \frac{2\alpha x^2}{(1-\alpha^2 x^2)^2} dx > 0 \tag{5.74}$$

for $\alpha \in (0,1)$. To show convexity, we take the second derivative of $h(\alpha)$ for $\alpha \in (0,1)$ and find that

$$\frac{d^2h(\alpha)}{d\alpha^2} = \int_0^1 \frac{2x^2 + 6\alpha^2 x^4}{(1-\alpha^2 x^2)^3} dx > 0. \tag{5.75}$$

□

The result of Lemma 5.5 implies that when η increases, $\text{MSD}_{\ell>1}$ in (5.73) decreases. That is, in a manner similar to $\text{MSD}_{\ell=1}$ in (5.55), the value of $\text{MSD}_{\ell>1}$ is lower if the network is more connected. In addition, we observe that when η is too low (or, α is too large in Fig. 5.4), the value of $h(2/\sqrt{\eta})$ will increase rapidly and so does the value of $\text{MSD}_{\ell>1}$. Note from (5.73) that $\text{MSD}_{\ell>1}$ depends on η through the function $h(2/\sqrt{\eta})/\eta$, or equivalently, $\alpha^2 h(\alpha)/4$ by replacing $2/\sqrt{\eta}$ with α. We show the derivative of $\alpha^2 h(\alpha)/4$ with respect to α in the right plot of Fig. 5.4. It is seen that the derivative function increases rapidly beyond $\alpha = 0.8$. To maintain acceptable levels of accuracy, it is preferable for the derivative to be bounded by a relative small value, say, 0.5. Then, the value of α should be less than 0.8, or $\eta \geq 6.25$. That is, the average neighborhood sizes should be kept around 6-7 or larger.

5.3.5 Behavior of the Network

Combining expressions (5.54), (5.55), and (5.73), we arrive at the following result for diffusion networks using the uniform combination matrix.

Theorem 5.5 (Diffusion networks under uniform combination weights)**.** *The ATC diffusion network (2.29) with the uniform combination matrix A in (5.43) has approximate convergence rate:*

$$r \approx \left(1 - \mu\lambda_{\min}(R_u) \cdot \frac{\sum_{k\in\mathcal{N}_I} n_k}{N\eta}\right)^2 \tag{5.76}$$

and approximate network MSD:

$$\text{MSD} \approx \underbrace{\frac{M\mu}{2N\eta} \cdot \frac{\sum_{k\in\mathcal{N}_I} \sigma_{v,k}^2 n_k^2}{\sum_{k\in\mathcal{N}_I} n_k}}_{\text{MSD}_{\ell=1}} + \underbrace{\frac{\mu^2 \text{Tr}(R_u)}{N\eta} \cdot h\left(\frac{2}{\sqrt{\eta}}\right) \cdot \sum_{k\in\mathcal{N}_I} \sigma_{v,k}^2 n_k}_{\text{MSD}_{\ell>1}} \tag{5.77}$$

where η and $h(\cdot)$ are defined in (5.44) and (5.72), respectively.

Note that expressions (5.76)-(5.77) for the convergence rate and network MSD depend on the network topology only through the agent degrees, $\{n_k\}$, and the network degree, η. In general, the higher values of η are, the slower the convergence rate is (an undesirable effect) and the lower the network MSD is (a desirable effect). This reveals again a trade-off between convergence rate and network MSD.

5.3.6 Mean-Square Performance under Fixed Convergence Rate

For a proper evaluation of how the proportion of informed agents influences network behavior, we shall adjust the step-size parameter such that the convergence rate remains fixed as the set of informed agents is enlarged and then compare the resulting network MSDs. To do so, we set the step-size to the following normalized value:

$$\mu = \frac{\mu_0}{\sum_{k\in\mathcal{N}_I} n_k} \tag{5.78}$$

for some $\mu_0 > 0$. Note that this choice normalizes μ_0 by the sum of the degrees of the informed agents. In this way, the convergence rate given by (5.76) becomes

$$r \approx \left(1 - \frac{\mu_0 \lambda_{\min}(R_u)}{N\eta}\right)^2 \tag{5.79}$$

which is independent of the set of informed agents. Moreover, the network MSD in (5.77) becomes

$$\text{MSD} \approx \frac{M\mu_0}{2N\eta} \cdot \frac{\sum_{k\in\mathcal{N}_I} \sigma_{v,k}^2 n_k^2}{\left(\sum_{k\in\mathcal{N}_I} n_k\right)^2} + \frac{\mu_0^2 \text{Tr}(R_u)}{N\eta} \cdot h\left(\frac{2}{\sqrt{\eta}}\right) \cdot \frac{\sum_{k\in\mathcal{N}_I} \sigma_{v,k}^2 n_k}{\left(\sum_{k\in\mathcal{N}_I} n_k\right)^2}. \tag{5.80}$$

Using the same argument we used before in (5.33), if we increase the number of informed agents by one, the first term in (5.80) (namely, $\text{MSD}_{\ell=1}$) will increase if the degree of the added agent satisfies:

$$n_{N_I+1} \geq \underbrace{2\left[\frac{\sigma_{v,N_I+1}^2 \left(\sum_{k\in\mathcal{N}_I} n_k\right)^2}{\sum_{k\in\mathcal{N}_I} \sigma_{v,k}^2 n_k^2} - 1\right]^{-1}}_{c_1} \sum_{k\in\mathcal{N}_I} n_k \tag{5.81}$$

and the second term in (5.80) (namely, $\text{MSD}_{\ell>1}$) will increase if the degree of the added agent satisfies:

$$n_{N_I+1} \leq \underbrace{\left(\frac{\sigma_{v,N_I+1}^2 \sum_{k\in\mathcal{N}_I} n_k}{\sum_{k\in\mathcal{N}_I} \sigma_{v,k}^2 n_k} - 2\right)}_{c_2} \sum_{k\in\mathcal{N}_I} n_k. \tag{5.82}$$

In the following, we show that there exist conditions under which both requirements (5.81) and (5.82) are satisfied. That is, when this happens and interestingly, the network MSD ends up increasing (an undesirable effect) when we add one more informed agent in the network. In the first example, we assume that the degrees of all agents are the same, i.e., set $n_k = n$ for all k. Then, c_1 and c_2 in (5.81)-(5.82) become

$$c_1 = 2(N_I\beta - 1)^{-1}, \quad c_2 = \beta - 2 \tag{5.83}$$

where

$$\beta = \frac{\sigma_{v,N_I+1}^2}{\sum_{k\in\mathcal{N}_I}\sigma_{v,k}^2/N_I}. \tag{5.84}$$

It can be verified that if

$$\beta \geq 2 + \frac{1}{N_I} \tag{5.85}$$

(or, if the noise variance at the added agent is large enough), both (5.81) and (5.82) are satisfied and then the MSD will increase (i.e., become worse). A second example is obtained by setting the noise variances to a constant level, i.e., $\sigma_{v,k}^2 = \sigma_v^2$ for all k. Then, c_1 and c_2 in (5.81)-(5.82) become

$$c_1 = 2\left[\frac{\left(\sum_{k\in\mathcal{N}_I} n_k\right)^2}{\sum_{k\in\mathcal{N}_I} n_k^2} - 1\right]^{-1}, \quad c_2 = -1. \tag{5.86}$$

In this case, the second term in (5.80) always decreases, whereas the first term in (5.80) will increase if the degree of the added informed agent is high enough. However, as the number of informed agents increases, the step-size in (5.78) will become smaller and the first term in (5.80) becomes dominant. As a result, the network MSD worsens if (5.81) is satisfied, i.e., when the added agent has large degree. *These results suggest that it is beneficial to let few highly noisy or highly connected agents remain uninformed and participate only in the consultation step (the second step in (2.29)).*

5.4 Simulation Results

We consider networks with 400 agents. The weight vector, w°, is a randomly generated 5×1 vector (i.e., $M = 5$). The regressor covariance matrix R_u is a diagonal matrix with each diagonal entry uniformly generated from $[0.8, 1.8]$, and noise variances are set to $\sigma_{v,k}^2 = 0.01$ for all k. The step-size for informed agents is set to $\mu = 0.02$. Without loss of generality, we assume that the agents are indexed

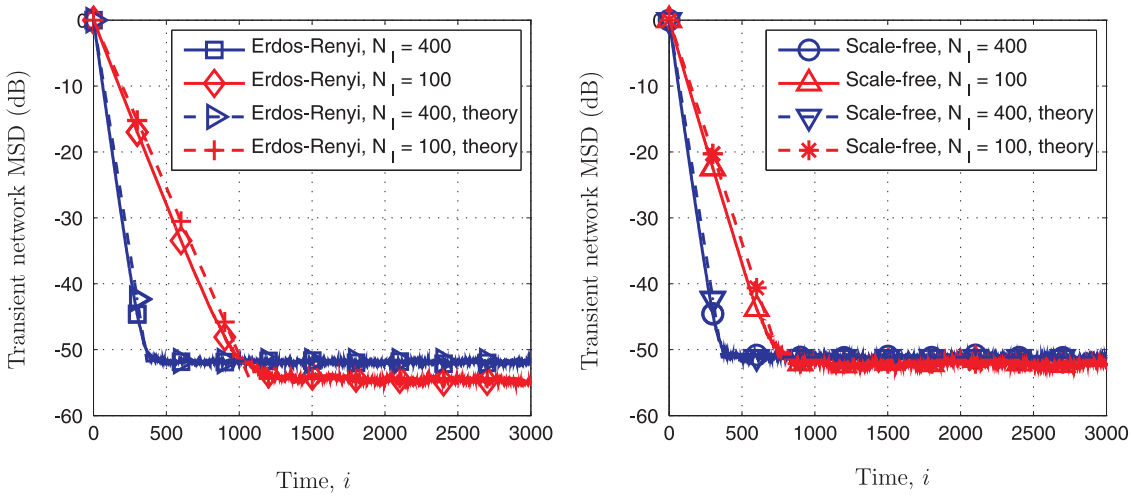


Figure 5.5: Transient network MSD over the Erdos-Renyi (left) and scale-free (right) networks with 400 agents. The dashed lines represent the theoretical results (2.73) and (2.87).

in decreasing order of degree, i.e., $n_1 \geq n_2 \geq \cdots \geq n_N$. In the simulations, we consider two network models as follows.

1. Erdos-Renyi model [20,52]: The Erdos-Renyi model has been studied widely in random graph theory [20]. In the Erdos-Renyi model, there is a single parameter called *edge probability* and is denoted by $p \in [0, 1]$. Each edge is connected with probability p independent from every other edge.

2. Scale-free model [3,9,10,105]: The scale-free model captures several prominent features of real networks (e.g., the Internet) such as the small-world phenomenon and the power-low degree distribution [3, 105]. The model starts with a small connected network with N_0 agents. At every iteration, we add a new agent, which will connect to $m \leq N_0$ distinct agents besides itself. The probability of connecting to an agent is proportional to its

degree. As time evolves, agents with higher degree are more likely to be connected to new agents.

We first verify theoretical expressions (2.73) and (2.87) for the convergence rate and network MSD. Figure 5.5 shows the MSD over time for two network models with parameters $p = 0.015$, $m = 3$, and $N_0 = 10$ so that both models have network degree around $\eta = 7$. For each network model, we consider two cases: 400 or 100 (agents 1 to 100) informed agents. We observe that when the number of informed agents decreases, the convergence rate increases, as expected, but interestingly, the MSD decreases. The theoretical results are also depicted in Fig. 5.5. The MSD decays at rate r in (2.73) during the transient stage. When the MSD is lower than the steady-state MSD value from (2.82), the MSD stays constant at (2.87). We observe that the theoretical results match well with simulations. The theoretical results (2.73) and (2.87) will be used to verify the effectiveness of the approximate expressions (5.76) and (5.77).

5.4.1 MSD and Convergence Rate with Fixed Step-Size

We examine the effect of the proportion and distribution of informed agents on the convergence rate and MSD of the network. We increase the number of informed agents from the agent with the highest degree, i.e., from agent 1 to agent N, so that condition (5.34) is satisfied for every iteration. The convergence rate and MSD are shown in Fig. 5.6. As expected, the convergence rates decrease when we add more informed agents and expression (5.76) matches well with expression (2.73). In addition, the convergence rates in the scale-free model are lower in the beginning because there are some agents with very high degrees.

Interesting patterns are seen in the MSD behavior. We further illustrate $\text{MSD}_{\ell=1}$ from (5.55) and $\text{MSD}_{\ell>1}$ from (5.73) in Fig. 5.7. We observe from Fig.

5.7 that $\text{MSD}_{\ell=1}$ decreases since condition (5.34) is satisfied, whereas $\text{MSD}_{\ell>1}$ increases with N_I. Additionally, the scale-free model has higher values of $\text{MSD}_{\ell=1}$ and $\text{MSD}_{\ell>1}$ than the Erdos-Renyi model, and therefore higher values of MSD. This is because the scale-free model has higher values of n_l. Since $\text{MSD}_{\ell=1}$ decreases and $\text{MSD}_{\ell>1}$ increases, the resulting MSD in (5.77) can either increase or decrease. The curve of MSD depends on the values of $\text{MSD}_{\ell=1}$ and $\text{MSD}_{\ell>1}$. We observe from Fig. 5.6 that as we expected in Fig. 5.2, the MSD decreases when N_I is small, and then increases with N_I. As in the case of a stand-alone adaptive filter, there exists a trade-off between the convergence rate and the MSD. We also see that the approximation for the MSD in (5.77) matches well with expression (2.87). The gap between approximation (5.77) and expression (2.87) is caused by approximations (5.51) and (5.66) and is about 0.5 dB ($\approx$ 12% deviation). Therefore, even though the approximations are not generally valid, simulations indicate that the approximations still lead to good match between theory and practice.

5.4.2 MSD with Fixed Convergence Rate

We vary the value of step-size as in (5.78) with $\mu_0 = 0.1$ and show the network MSD over the number of informed agents in Fig. 5.8. To show the MSD possibly increases with N_I, we reverse the order in adding informed agents, i.e., from agent N to agent 1. It is interesting to note that for the scale-free model, the MSD increases when the number of informed agents is large. This is because in the scale-free model, there are few agents connected to most agents in the network and condition (5.81) is satisfied. The results suggest that in the scale-free model, we should let few highly connected agents remain uninformed and perform only the consultation step.

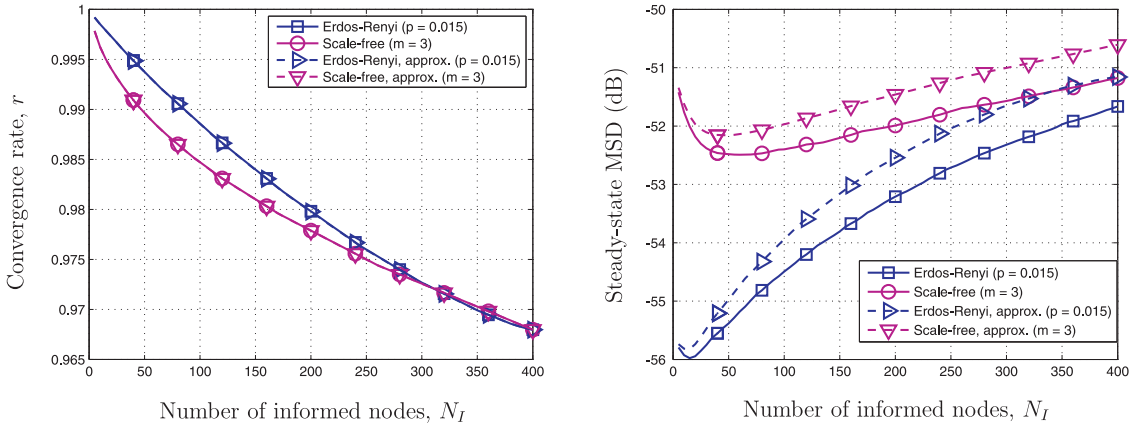


Figure 5.6: Convergence rate (left) and steady-state MSD (right) for Erdos-Renyi and scale-free models with the addition of informed agents in decreasing order of degree. The dashed lines represent approximate expressions (5.76) and (5.77).

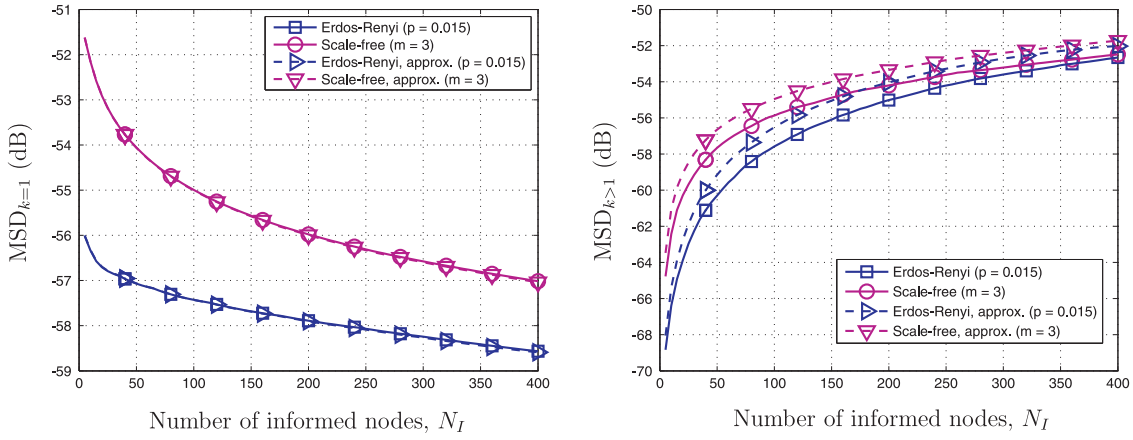


Figure 5.7: $\text{MSD}_{l=1}$ (left) and $\text{MSD}_{\ell>1}$ (right) for Erdos-Renyi and scale-free models with the addition of informed agents in decreasing order of degree. The dashed lines represent approximate expressions (5.55) and (5.73).

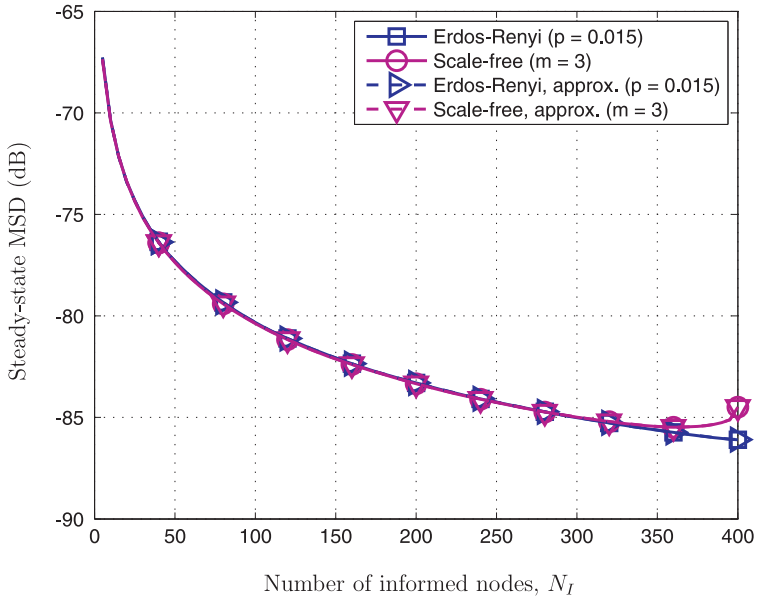


Figure 5.8: Steady-state MSD with the deployment for agent N to agent 1 for Erdos-Renyi and scale-free models. The dashed lines represent approximate expression (5.80).

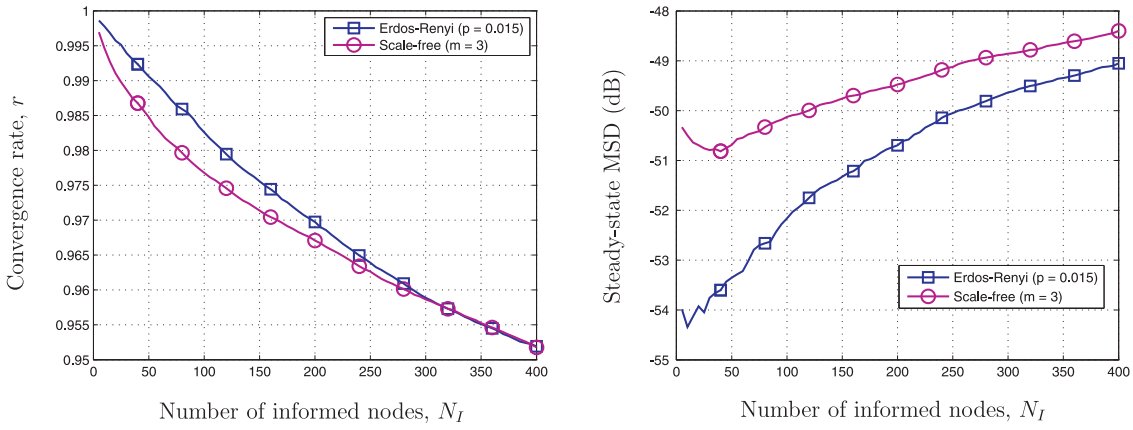


Figure 5.9: Convergence rate (left) and steady-state MSD (right) for Erdos-Renyi and scale-free models with general form of step-sizes and regression covariance matrices.

5.4.3 Nonuniform Step-Sizes and Regression Covariance Matrices

The results in Section 5.2 are derived under the assumption that the agents are homogeneous (namely, uniform step-sizes and regression covariance matrices). In this part, we allow the step-sizes and covariance matrices to vary across the agents in the network. The step-sizes are generated uniformly and independently from $[0.01, 0.04]$. The regression covariance matrices are diagonal with each diagonal entry uniformly generated from $[0.8, 1.8]$. The theoretical expressions for convergence rate (2.73) and network MSD (2.87) in terms of the number of informed agents are shown in Fig. 5.9. The number of informed agents is increased from agent 1 to agent N. We observe that the convergence rate decreases as N_I increases. The network MSD exhibits similar behavior in the right plot of Fig. 5.6, i.e., it decreases in the beginning and increases thereafter. The results show that the trade-off between the convergence rate and network MSD in terms of the number of informed agents also occurs in this scenario.

5.5 Concluding Remarks

In this chapter, we derived useful expressions for the convergence rate and mean-square performance of the ATC diffusion strategies in the presence of uninformed agents. The analysis examines analytically how the convergence rate and mean-square performance of the network vary with the network topology and with the proportion of informed agents. The results reveal interesting and surprising patterns of behavior. The analysis shows that there exists a trade-off between convergence rate and mean-square deviation in terms of the proportion of informed agents. It is not always the case that increasing the proportion of informed agents is beneficial.

5.A Proof of Theorem 5.1

To prove mean stability of the diffusion network (2.29), we need to show that conditions (5.2)-(5.3) guarantee $\rho(\mathcal{B}) < 1$, or equivalently, $\rho(\mathcal{B}^j) < 1$ for some integer j. Now, note that

$$\rho(\mathcal{B}^j) \leq \|\mathcal{B}^j\|_b = \max_{1\leq k\leq N} \left(\sum_{l=1}^{N} \left\| \left[\mathcal{B}^j\right]_{k,l} \right\|_2 \right). \tag{5.87}$$

By the rules of matrix multiplication, the (k,l)th block (of size $M \times M$) of the matrix $\mathcal{B}^j$ is given by:

$$\left[\mathcal{B}^j\right]_{k,l} = \sum_{m_1=1}^{N} \sum_{m_2=1}^{N} \cdots \sum_{m_{j-1}=1}^{N} \mathcal{B}_{k,m_1} \mathcal{B}_{m_1,m_2} \cdots \mathcal{B}_{m_{j-1},l} \tag{5.88}$$

where $\mathcal{B}_{k,l}$ is the (k,l)th block (of size $M \times M$) of the matrix $\mathcal{B}$ from (5.1) and is given by

$$\mathcal{B}_{k,l} = a_{l,k} \cdot (I_M - \mu_l R_{u,l}). \tag{5.89}$$

Then, using the triangle inequality and the submultiplicative property of norms [64], the 2-induced norm of $[\mathcal{B}^j]_{k,l}$ in (5.88) is bounded by:

$$\left\| \left[\mathcal{B}^j\right]_{k,l} \right\|_2 \leq \sum_{m_1=1}^{N} \sum_{m_2=1}^{N} \cdots \sum_{m_{j-1}=1}^{N} \|\mathcal{B}_{k,m_1}\|_2 \times \|\mathcal{B}_{m_1,m_2}\|_2 \cdots \|\mathcal{B}_{m_{j-1},l}\|_2. \tag{5.90}$$

Note that in the case where $l \in \mathcal{N}_m$, we obtain from condition (5.2) and expression (5.89) that

$$\begin{aligned} \|\mathcal{B}_{m,l}\|_2 &= a_{l,m} \cdot \rho\left(I_M - \mu_l R_{u,l}\right) \\ &\begin{cases} < a_{l,m}, & \text{if agent } l \text{ is informed} \\ = a_{l,m}, & \text{if agent } l \text{ is uninformed} \end{cases} \end{aligned} \tag{5.91}$$

where we replaced the 2-induced norm with the spectral radius because covariance matrices are Hermitian. Relation (5.91) implies that

$$\left\| \left[\mathcal{B}^j\right]_{k,l} \right\|_2 \leq \sum_{m_1=1}^{N} \sum_{m_2=1}^{N} \cdots \sum_{m_{j-1}=1}^{N} a_{m_1,k} \cdot a_{m_2,m_1} \cdots a_{l,m_{j-1}}. \tag{5.92}$$

From condition (2.25), strict inequality holds in (5.92) if, and only if, the sequence $(l, m_{j-1}, \ldots, m_1, k)$ forms a path from agent l to agent k using j edges and there exists at least one informed agent along the path. Since we know from condition (5.3) that there is an informed agent, say, agent l°, such that a path with j edges exists from agent l° to agent k, we then get from (5.87) and (5.92) that

$$\begin{aligned} \rho(\mathcal{B}^j) &\leq \max_{1\leq k\leq N} \left(\left\| \left[\mathcal{B}^j\right]_{k,l^\circ} \right\|_2 + \sum_{l\neq l^\circ} \left\| \left[\mathcal{B}^j\right]_{k,l} \right\|_2 \right) \\ &< \max_{1\leq k\leq N} \sum_{l=1}^{N} \left(\sum_{m_1=1}^{N} \sum_{m_2=1}^{N} \cdots \sum_{m_{j-1}=1}^{N} a_{m_1,k} a_{m_2,m_1} \cdots a_{l,m_{j-1}} \right) \\ &= \max_{1\leq k\leq N} \sum_{l=1}^{N} \left[A^j\right]_{l,k} \\ &= 1 \end{aligned} \tag{5.93}$$

where the last equality is from condition (2.25) because $(A^T)^j \mathbb{1}_N = \mathbb{1}_N$ if $A^T \mathbb{1}_N = \mathbb{1}_N$.

5.B Proof of Lemma 5.2

Since the agents are homogeneous from (5.4), we have that

$$I_M - \mu_l R_{u,l} = \begin{cases} I_M - \mu R_u, & \text{if agent } l \text{ is informed} \\ I_M, & \text{if agent } l \text{ is uninformed} \end{cases} \tag{5.94}$$

Then, the matrix $[\mathcal{B}^j]_{k,l}$ in (5.88) can be written as:

$$\left[\mathcal{B}^j\right]_{k,l} = \sum_{m_1=1}^{N} \sum_{m_2=1}^{N} \cdots \sum_{m_{j-1}=1}^{N} a_{m_1,k} \cdot a_{m_2,m_1} \cdots a_{l,m_{j-1}} \cdot (I_M - \mu R_u)^{q_{l,k}} \tag{5.95}$$

where the exponent $q_{l,k}$ denotes the number of informed agents along the path $(l, m_{j-1}, \ldots, m_1, k)$. Note that $[\mathcal{B}^j]_{k,l}$ is a nonnegative-definite matrix because $(I_M - \mu R_u) > 0$ in view of sufficiently small step-sizes. In fact, all eigenvalues of $(I_M - \mu R_u)$ lie within the line segment $(0, 1)$. Moreover, since $\mathcal{N}_{I,1} \subseteq \mathcal{N}_{I,2}$, we have that $q_{l,k}^{(1)} \leq q_{l,k}^{(2)}$ and, therefore, the matrix difference

$$\begin{aligned} &\left[\mathcal{B}^{(1)j}\right]_{k,l} - \left[\mathcal{B}^{(2)j}\right]_{k,l} \\ &= \sum_{m_1=1}^{N} \sum_{m_2=1}^{N} \cdots \sum_{m_{j-1}=1}^{N} a_{m_1,k} \cdot a_{m_2,m_1} \cdots a_{l,m_{j-1}} \cdot \left[(I - \mu R_u)^{q_{l,k}^{(1)}} - (I - \mu R_u)^{q_{l,k}^{(2)}}\right] \end{aligned} \tag{5.96}$$

is a nonnegative-definite matrix, where the superscripts denote the indices of the informed configurations, $\mathcal{N}_{I,1}$ or $\mathcal{N}_{I,2}$. Since $[\mathcal{B}^{(1)j}]_{k,l}$, $[\mathcal{B}^{(2)j}]_{k,l}$, and $[\mathcal{B}^{(1)j}]_{k,l} - [\mathcal{B}^{(2)j}]_{k,l}$ are all nonnegative-definite, then it must hold that

$$\left\| \left[\mathcal{B}^{(1)j}\right]_{k,l} \right\|_2 \geq \left\| \left[\mathcal{B}^{(2)j}\right]_{k,l} \right\|_2. \tag{5.97}$$

Relation (5.97) can be established by contradiction. Suppose that (5.97) does not hold, i.e., $\rho([\mathcal{B}^{(1)j}]_{k,l}) < \rho([\mathcal{B}^{(2)j}]_{k,l})$ as $[\mathcal{B}^{(1)j}]_{k,l}$ and $[\mathcal{B}^{(2)j}]_{k,l}$ are Hermitian from (5.95). In addition, let x denote the eigenvector that is associated with the largest eigenvalue of $[\mathcal{B}^{(2)j}]_{k,l}$, i.e., $([\mathcal{B}^{(2)j}]_{k,l})x = \rho([\mathcal{B}^{(2)j}]_{k,l})x$. Then, we obtain the following contradiction to the nonnegative-definiteness of $[\mathcal{B}^{(1)j}]_{k,l} - [\mathcal{B}^{(2)j}]_{k,l}$:

$$x^* \left(\left[\mathcal{B}^{(1)j}\right]_{k,l} - \left[\mathcal{B}^{(2)j}\right]_{k,l} \right) x = x^* \left(\left[\mathcal{B}^{(1)j}\right]_{k,l} \right) x - \rho \left(\left[\mathcal{B}^{(2)j}\right]_{k,l} \right) x^* x < 0 \quad (5.98)$$

by the Rayleigh-Ritz Theorem [64]. By the definition of the block matrix norm in (3.67), we arrive at

$$\left(\left\| \left[\mathcal{B}^{(1)j}\right] \right\|_b \right)^{1/j} \geq \left(\left\| \left[\mathcal{B}^{(2)j}\right] \right\|_b \right)^{1/j} \quad (5.99)$$

for all j. Let j tend to infinity and we obtain that

$$\rho\left(\mathcal{B}^{(1)}\right) \geq \rho\left(\mathcal{B}^{(2)}\right) \quad (5.100)$$

where we used the fact that $\rho(\mathcal{B}) = \lim_{j\to\infty}(\|\mathcal{B}^j\|)^{1/j}$ for any matrix norm [64].

5.C Proof of Lemma 5.1

We first note that the matrix $\mathcal{B} = \mathcal{A}^T(I_{NM} - \mathcal{M}\mathcal{R})$ can be written as a function of the step-size μ:

$$\mathcal{B}(\mu) = \mathcal{A}^T - \mu\mathcal{A}^T\mathcal{D}\mathcal{R} \quad (5.101)$$

where $\mathcal{D} = \mathcal{M}/\mu$. Therefore, the derivative of $\mathcal{B}(\mu)$ with respect to μ becomes

$$\mathcal{B}'(\mu) = -\mathcal{A}^T\mathcal{D}\mathcal{R}. \quad (5.102)$$

Then, according to the perturbation theory of eigenvalues [160], the derivative of the eigenvalues of $\mathcal{B}(\mu)$ with respect to μ can be expressed as:

$$\lambda'_{l,m}(B(\mu)) = s^{b*}_{l,m}(0)\mathcal{B}'(\mu)r^b_{l,m}(0) \quad (5.103)$$

where $r^b_{l,m}(0)$ and $s^b_{l,m}(0)$ are the right and left eigenvectors of $\mathcal{B}(0)$. Note from (5.101) that $\mathcal{B}(0) = \mathcal{A}^T$ and we have that $r^b_{l,m}(0) = r_l \otimes z_m$ and $s^b_{l,m}(0) = s_l \otimes z_m$ corresponding to the eigenvalue $\lambda_{l,m}(B(0)) = \lambda_l(A)$. Then, from (5.102), expression (5.103) becomes

$$\begin{aligned}\lambda'_{l,m}(B(\mu)) &= -(s_l \otimes z_m)^* \mathcal{A}^T \mathcal{D}\mathcal{R}(r_l \otimes z_m) \\ &= -\lambda_l(A)\lambda_m(R_u) \cdot s^*_{l,1:N_I} r_{l,1:N_I}. \end{aligned} \tag{5.104}$$

Therefore, the Taylor series expansion for the eigenvalues of $\mathcal{B}(\mu)$ around $\mu = 0$ is given by:

$$\begin{aligned}\lambda_{l,m}(B(\mu)) &= \lambda_{l,m}(B(0)) + \mu \cdot \lambda'_{l,m}(B(\mu)) + \mathcal{O}(\mu^2) \\ &= \lambda_l(A) - \mu\lambda_l(A)\lambda_m(R_u) \cdot s^*_{l,1:N_I} r_{l,1:N_I} + \mathcal{O}(\mu^2) \\ &= \lambda_l(A) \cdot \left[1 - \mu\lambda_m(R_u) \cdot s^*_{l,1:N_I} r_{l,1:N_I}\right] + \mathcal{O}(\mu^2) \end{aligned} \tag{5.105}$$

and then we arrive at (5.8).

To verify that the right and left eigenvectors of $\mathcal{B}$ have the form (5.9)-(5.10), we need to show that $\mathcal{B} r^b_{l,m} = \lambda_{l,m}(B) r^b_{l,m}$ and $s^{b*}_{l,m}\mathcal{B} = \lambda_{l,m}(B) s^{b*}_{l,m}$. We show the former in the following and the latter can be deduced in a similar manner. From (5.9) and (5.101), we have that

$$\begin{aligned}\mathcal{B} \cdot r^b_{l,m} &= (\mathcal{A}^T - \mu\mathcal{A}^T\mathcal{D}\mathcal{R})[(r_l \otimes z_m) + \mu \cdot \tilde{r}^b_{l,m} + \mathcal{O}(\mu^2)] \\ &= \mathcal{A}^T(r_l \otimes z_m) + \mu \cdot \left[-\mathcal{A}^T\mathcal{D}\mathcal{R}(r_l \otimes z_m) + \mathcal{A}^T\tilde{r}^b_{l,m}\right] + \mathcal{O}(\mu^2) \end{aligned} \tag{5.106}$$

On the other hand, we obtain from (5.9), (5.104), and (5.105) that

$$\begin{aligned}&\lambda_{l,m}(B) r^b_{l,m} \\ &= [\lambda_l(A) - \mu \cdot (s_l \otimes z_m)^* \mathcal{A}^T \mathcal{D}\mathcal{R}(r_l \otimes z_m) + \mathcal{O}(\mu^2)] \\ &\quad \times [(r_l \otimes z_m) + \mu \cdot \tilde{r}^b_{l,m} + \mathcal{O}(\mu^2)] \\ &= \lambda_l(A)(r_l \otimes z_m) \\ &\quad + \mu \cdot \left[- (s_l \otimes z_m)^* \mathcal{A}^T \mathcal{D}\mathcal{R}(r_l \otimes z_m)(r_l \otimes z_m) + \lambda_l(A)\tilde{r}^b_{l,m}\right] + \mathcal{O}(\mu^2) \end{aligned} \tag{5.107}$$

Note that

$$\mathcal{A}^T(r_l \otimes z_m) = \lambda_l(A)(r_l \otimes z_m). \tag{5.108}$$

Therefore, expressions (5.106)-(5.107) equate if $\tilde{r}^b_{l,m}$ satisfies

$$\begin{aligned} &- \mathcal{A}^T\mathcal{D}\mathcal{R}(r_l \otimes z_m) + \mathcal{A}^T\tilde{r}^b_{l,m} \\ &= -(s_l \otimes z_m)^*\mathcal{A}^T\mathcal{D}\mathcal{R}(r_l \otimes z_m)(r_l \otimes z_m) + \lambda_l(A)\tilde{r}^b_{l,m} \end{aligned} \tag{5.109}$$

and then we arrive at (5.11).

CHAPTER 6

Distributed Decision-Making

Self-organized behavior is a remarkable property of biological networks [25, 45], where various forms of complex and sophisticated behavior are evident and result from decentralized interactions among agents with limited capabilities. One example of sophisticated behavior is the group decision-making process by animal groups [138]. For example, it is common for biological networks to encounter situations where agents need to decide between multiple options, such as fish deciding between following one food source or another [46] and bees or ants deciding between moving towards a new hive or another [23,113]. Although multiple options may be available, the agents are still able to achieve agreement in a decentralized manner and move towards a common destination (e.g., [7, 12, 137]).

In previous chapters, we described several useful diffusion strategies [34, 37, 95, 124] that allow agents to adapt and learn through a process of in-network collaboration [41, 42, 87, 97, 164]. Diffusion networks consist of a collection of agents that are able to respond to excitations in real-time. As shown in Chapter 3 and compared with the class of consensus strategies [50,73,74,78,103,144,170], adaptive diffusion networks have been shown to remain stable *regardless* of the network topology, while adaptive consensus networks can become unstable even when all individual agents are stable [154,155]. This fact is a serious hindrance to the study of biological networks where the network topology is in continuous flux. Diffusion strategies are particularly well-suited to model such networks

(see Chapter 7 and [29, 127, 150]) because of their robustness to changes in the network topology. Diffusion strategies have also been shown in Chapter 3 to lead to improved convergence rate and superior mean-square-error performance [127, 154, 155]. For these reasons, we focus on the use of diffusion strategies for decentralized decision-making.

Motivated by the behavior of biological networks, we study in this chapter distributed decision-making over networks where agents may have distinct objectives [153]. In distributed processing, agents generally collect data generated by the same underlying unknown distribution or model and then solve the estimation and inference tasks cooperatively. We consider the situation in which the data observed by the agents may arise from different distributions or models. Agents do not know beforehand which model accounts for their data and the objective of the network becomes that of guiding all agents towards a *common* goal. In these situations, where agents are subject to data from unknown different sources, conventional distributed (consensus and diffusion) strategies would lead to biased solutions in that the agents will end up converging towards a linear combination of the underlying models (see (6.17) and (6.18) further ahead). For example, in the context of fish schools, such outcome would mean that the school will fail to reach any of the food sources.

The task of encouraging agreement over a network of agents with varied backgrounds (i.e., models) is more challenging than earlier works on inference under uniform data models. The difficulty is due to various reasons. First, as we are going to show in Lemma 6.2, traditional distributed strategies will converge to a linear combination of the underlying models, and therefore, the estimates will be biased. We then need to compensate for the bias in the adaptive learning process in real-time. Second, each agent now needs to distinguish between which

model each of its neighbors is collecting data from (this is called the *observed* model) and which model the network is evolving to (this is called the *desired* model). In other words, in addition to the learning and adaptation process, the agents should be equipped with a classification scheme to distinguish between the observed and desired models. Finally, in order for the network to converge to a common objective, the agents should be endowed with a decision-making process that would enable them to reach agreement on the desired model. Moreover, the classification scheme and the decision-making process will need to be implemented in a fully distributed manner and in real-time.

We analyze the performance of the proposed algorithms and examine the probability of errors in the classification scheme. We show that with high probability, the agents are able to correctly identify the observed models of their neighbors in the proposed solution. We also examine the rate of convergence of the modified diffusion strategy and propose rules to improve convergence. This is especially important in biological networks. For example, in a fish school, the convergence rate determines how quickly the fish arrive at a food source when they have disagreement on food locations. In addition, when there is a predator that is observable to a small fraction of the fish school, the convergence rate determines how quickly the fish react to danger and reverse direction.

6.1 Diffusion Strategy

In a manner that is different from the data model (2.3) used in Section 2.1, the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ collected at agent k are now assumed to originate from one of two unknown *column* vectors $\{w_0^\circ, w_1^\circ\}$ of size M (see Fig. 6.1). Agent k does not know beforehand which model is responsible for its data. We denote the generic model by $z_k^\circ \in \{w_0^\circ, w_1^\circ\}$. The data at agent k are related to its observed model

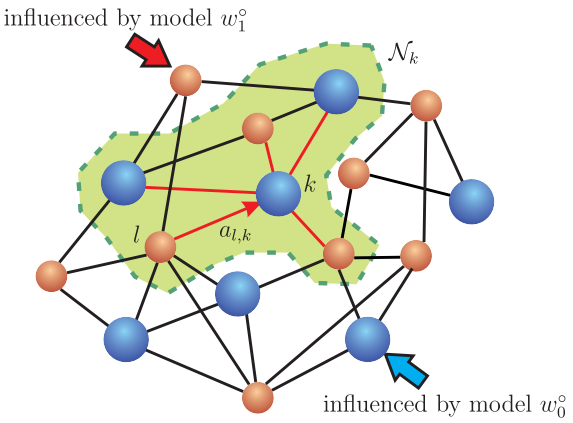


Figure 6.1: A connected network where data collected by the agents are influenced by one of two models. The weight $a_{l,k}$ scales the data transmitted from agent l to agent k over the edge linking them.

z_k° via a linear regression model of the form:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} z_k^\circ + \boldsymbol{v}_k(i) \tag{6.1}$$

Although the agents are subjected to data arising from different models, the objective of the network is still to have *all* agents converge to an estimate for *one* of the models. For example, if the models happen to represent the location of food sources [150], then this agreement will make all agents move towards one particular food source in lieu of the other sources. More specifically, let $w_{k,i}$ denote the estimate for z_k° at agent k at time i. The network would like to achieve

$$w_{k,i} \to w_q^\circ \text{ for } q = 0 \text{ or } q = 1 \text{ and for all } k \text{ as } i \to \infty \tag{6.2}$$

where convergence is in some desirable sense (such as the mean-square-error sense).

When the data arriving at the agents could have risen from one model or another, as is the case under study, the ATC diffusion strategy (2.29) will not be

able to achieve agreement among the agents and the resulting weight estimates will tend towards a biased value. We first explain how this degradation arises and subsequently explain how it can be remedied. In order to achieve agreement, it is reasonable to assume that the network is strongly connected (see Assumption 2.2). From the Perron-Frobenius Theorem [17, 64, 112, 124], the matrix A will have a unique eigenvalue at one while all other eigenvalues will be strictly less than one in magnitude. Moreover, if we let c denote the right-eigenvector of A that is associated with the eigenvalue at one and normalize the entries of c to add up to one, i.e.,

$$Ac = c \quad \text{and} \quad \mathbb{1}_N^T c = 1 \tag{6.3}$$

then all individual entries of c lie within $(0, 1)$, i.e.,

$$0 < c_k < 1 \quad \text{and} \quad c = \text{col}\{c_k\} \tag{6.4}$$

6.1.1 Biased Estimators

Let us assume for the time being that the agents in the network have agreed on converging towards one of the models (but they do not know beforehand which model it will be). We denote the desired model generically by w_q°. In Section 6.3, we explain how this agreement process can be attained. Here we explain that even when agreement is present, the diffusion strategy (2.29) leads to biased estimates unless it is modified in a proper way. To see this, we introduce the following error vectors for any agent k:

$$\tilde{\boldsymbol{w}}_{k,i} \triangleq w_q^\circ - \boldsymbol{w}_{k,i} \quad \text{and} \quad \tilde{z}_k^\circ \triangleq w_q^\circ - z_k^\circ. \tag{6.5}$$

Observe that these quantities measure the error relative to the desired objective, w_q°. Moreover, this desired model may or may not be the model that is influencing the data received by agent k. Then, using model (6.1), we obtain that the update

vector in (2.29) becomes

$$\begin{aligned}
\boldsymbol{h}_{k,i} &\triangleq \boldsymbol{u}_{k,i}^T[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \\
&= \boldsymbol{u}_{k,i}^T[\boldsymbol{u}_{k,i}z_k^\circ + \boldsymbol{v}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \\
&= \boldsymbol{u}_{k,i}^T\boldsymbol{u}_{k,i}(z_k^\circ - w_q^\circ + w_q^\circ - \boldsymbol{w}_{k,i-1}) + \boldsymbol{u}_{k,i}^T\boldsymbol{v}_k(i) \\
&= \boldsymbol{u}_{k,i}^T\boldsymbol{u}_{k,i}\tilde{\boldsymbol{w}}_{k,i-1} - \boldsymbol{u}_{k,i}^T\boldsymbol{u}_{k,i}\tilde{z}_k^\circ + \boldsymbol{u}_{k,i}^T\boldsymbol{v}_k(i)
\end{aligned} \tag{6.6}$$

We collect all error vectors across the network into block vectors:

$$\tilde{\boldsymbol{w}}_i \triangleq \operatorname{col}\{\tilde{\boldsymbol{w}}_{1,i}, \tilde{\boldsymbol{w}}_{2,i}, \cdots, \tilde{\boldsymbol{w}}_{N,i}\} \tag{6.7}$$

$$\tilde{z}^\circ \triangleq \operatorname{col}\{\tilde{z}_1^\circ, \tilde{z}_2^\circ, \cdots, \tilde{z}_N^\circ\} \tag{6.8}$$

Then, starting from (2.29) and using relation (6.6), we can verify that the global error vector $\tilde{\boldsymbol{w}}_i$ of the network evolves over time according to the recursion:

$$\tilde{\boldsymbol{w}}_i = \mathcal{A}^T(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\tilde{\boldsymbol{w}}_{i-1} + \mathcal{A}^T\mathcal{M}\boldsymbol{\mathcal{R}}_i\tilde{z}^\circ - \mathcal{A}^T\mathcal{M}\boldsymbol{s}_i \tag{6.9}$$

where $\boldsymbol{\mathcal{R}}_i$ and $\boldsymbol{s}_i$ are defined in (2.41) and (2.42), respectively. We can rewrite recursion (6.9) in the compact form:

$$\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \cdot \tilde{\boldsymbol{w}}_{i-1} + \boldsymbol{y}_i \tag{6.10}$$

where the matrix $\boldsymbol{\mathcal{B}}_i$ and the vector $\boldsymbol{y}_i$ are defined in Table 6.1.

Since the matrix $\boldsymbol{\mathcal{B}}_i$ is independent of $\tilde{\boldsymbol{w}}_{i-1}$ and the vector $\boldsymbol{s}_i$ in $\boldsymbol{y}_i$ has zero mean, taking expectation of both sides of (6.9), we find that the mean of $\tilde{\boldsymbol{w}}_i$ evolves over time according to the recursion:

$$\mathbb{E}\tilde{\boldsymbol{w}}_i = \mathcal{B} \cdot \mathbb{E}\tilde{\boldsymbol{w}}_{i-1} + y \tag{6.11}$$

where $\mathcal{B} \triangleq \mathbb{E}\boldsymbol{\mathcal{B}}_i$ and $y \triangleq \mathbb{E}\boldsymbol{y}_i$ are defined in Table 6.1 with $\mathcal{R}$ defined in (2.48). The following result provides conditions to ensure the convergence of (6.11).

Table 6.1: The network weight error vector evolves according to the recursion $\tilde{\boldsymbol{w}}_i = \boldsymbol{\mathcal{B}}_i \cdot \tilde{\boldsymbol{w}}_{i-1} + \boldsymbol{y}_i$, where the variables $\{\boldsymbol{\mathcal{B}}_i, \boldsymbol{y}_i\}$ and their respective means are listed below for the conventional and modified diffusion strategies.

	Diffusion (2.29)	**Modified diffusion** (6.23)-(6.24)
$\boldsymbol{\mathcal{B}}_i$	$\mathcal{A}^T(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)$	$\mathcal{A}_1^T(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i) + \mathcal{A}_2^T$
$\mathcal{B} \triangleq \mathbb{E}\boldsymbol{\mathcal{B}}_i$	$\mathcal{A}^T(I_{NM} - \mathcal{M}\mathcal{R})$	$\mathcal{A}_1^T(I_{NM} - \mathcal{M}\mathcal{R}) + \mathcal{A}_2^T$
$\boldsymbol{y}_i$	$\mathcal{A}^T\mathcal{M}\boldsymbol{\mathcal{R}}_i\tilde{z}^\circ - \mathcal{A}^T\mathcal{M}\boldsymbol{s}_i$	$\mathcal{A}_1^T\mathcal{M}\boldsymbol{\mathcal{R}}_i\tilde{z}^\circ - \mathcal{A}_1^T\mathcal{M}\boldsymbol{s}_i$
$y \triangleq \mathbb{E}\boldsymbol{y}_i$	$\mathcal{A}^T\mathcal{M}\mathcal{R}\tilde{z}^\circ$	$\mathcal{A}_1^T\mathcal{M}\mathcal{R}\tilde{z}^\circ$

Lemma 6.1. *Recursion (6.11) for $\mathbb{E}\tilde{\boldsymbol{w}}_i$ converges to zero if, and only if,*

$$\rho(\mathcal{B}) < 1 \;\; \textit{and} \;\; y = 0 \tag{6.12}$$

where $\rho(\cdot)$ denotes the spectral radius of its argument.

Proof. If (6.12) holds, then it is straightforward to verify that mean convergence holds. Conversely, it is clear that $\mathbb{E}\tilde{\boldsymbol{w}}_i$ will not converge if $\rho(\mathcal{B}) \geq 1$. Thus assume $\rho(\mathcal{B}) < 1$. Then $\mathbb{E}\tilde{\boldsymbol{w}}_i$ converges to the steady-state value

$$\lim_{i\to\infty} \mathbb{E}\tilde{\boldsymbol{w}}_i = (I_{NM} - \mathcal{B})^{-1} \cdot y \tag{6.13}$$

and it follows that y must be the zero vector to enforce convergence in the mean. □

Therefore, to guarantee (unbiased) mean convergence, the agents need to select the step-sizes $\{\mu_k\}$ and the combination matrix A so that condition (6.12) is satisfied. It was verified in Section 3.1 that a sufficient condition to ensure

$\rho(\mathcal{B}) < 1$ is to select

$$0 < \mu_k < \frac{2}{\rho(R_{u,k})} \quad \text{for all } k. \tag{6.14}$$

This conclusion is independent of A. However, for the second condition in (6.12), we note that in general, the vector $y = \mathcal{A}^T \mathcal{M} \mathcal{R} \tilde{z}^\circ$ cannot be zero no matter how the agents select the combination matrix A. When this happens, the weight estimate will be biased in the mean. Let us consider an example with three agents in Fig. 6.2 where agent 1 observes data from model w_0°, while agents 2 and 3 observe data from another model w_1°. The combination matrix in this case is given by

$$A^T = \begin{bmatrix} 1-a & a & 0 \\ b & 1-b-c & c \\ 0 & d & 1-d \end{bmatrix} \tag{6.15}$$

with the parameters $\{a, b, c, d\}$ lying in the interval $[0, 1]$ and $b + c \leq 1$. For simplicity, we assume that the step-sizes and regression covariance matrices are the same, i.e., $\mu_k = \mu$ and $R_{u,k} = R_u$ for all k. If the desired model of the network is $w_q^\circ = w_0^\circ$, then the vector y becomes

$$y = \mu \cdot \begin{bmatrix} a \\ 1-b \\ 1 \end{bmatrix} \otimes R_u(w_0^\circ - w_1^\circ) \tag{6.16}$$

We observe that no matter how we select the parameters $\{a, b, c, d\}$, the third entry of y can never become zero for different models. More generally, using results on the limiting behavior of diffusion estimation errors $\{\tilde{\boldsymbol{w}}_i\}$ from [38], we can characterize the limiting point of the current distributed strategy as follows.

Lemma 6.2. *For the diffusion strategy from (2.29) and for sufficiently small step-sizes, all weight estimates $\{\boldsymbol{w}_{k,i}\}$ converge to a limit point w° in the mean-square sense, i.e., $\mathbb{E}\|w^\circ - \boldsymbol{w}_{k,i}\|^2$ is bounded and of the order of $\mu_{\max} = \max\{\mu_k\}$,*

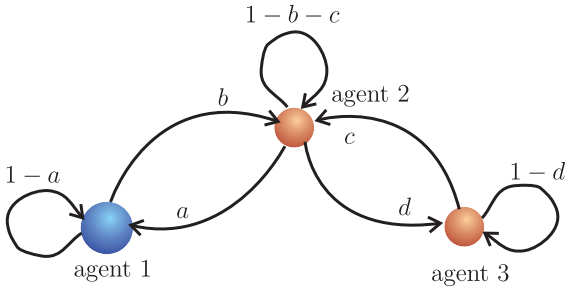


Figure 6.2: A three-agent network. Agent 1 observes data from w_0° while agents 2 and 3 observe data from w_1°.

where w° is given by

$$w^\circ = \left(\sum_{k=1}^{N} c_k \mu_k R_{u,k}\right)^{-1} \left(\sum_{k=1}^{N} c_k \mu_k R_{u,k} z_k^\circ\right) \tag{6.17}$$

where the vector c was defined in (6.3)-(6.4). Moreover, when the agents are homogeneous so that $\mu_k = \mu$ and $R_{u,k} = R_u$ for all k, w° is given by

$$w^\circ = \sum_{k=1}^{N} c_k z_k^\circ \tag{6.18}$$

Proof. The main theorem in [38] asserts that, for sufficiently small step-sizes, the limit point w° of the network is the unique solution of the following equation:

$$\sum_{k=1}^{N} c_k \mu_k \nabla_w J_k(w^\circ) = 0 \tag{6.19}$$

where $\nabla_w J_k(w^\circ)$ is defined as the expectation of the update vector $\boldsymbol{h}_{k,i}$ from (6.6) evaluated at $\boldsymbol{w}_{k,i-1} = w^\circ$, i.e.,

$$\nabla_w J_k(w) = R_{u,k}(z_k^\circ - w^\circ) \tag{6.20}$$

Substituting (6.20) into (6.19), we have that

$$\sum_{k=1}^{N} c_k \mu_k R_{u,k}(z_k^\circ - w^\circ) = 0 \tag{6.21}$$

which leads to (6.17). When the agents are homogeneous, expression (6.17) simplifies to (6.18). □

The above result shows that when the agents collect data from different models, the estimates using the diffusion strategy (2.29) converge to a linear combination of these models, as in (6.17) or (6.18). This linear combination is different from any of the individual models. For example, in the case of (6.18), if the matrix A happens to be doubly-stochastic (i.e., $A^T\mathbb{1}_N = \mathbb{1}_N$ and $A\mathbb{1}_N = \mathbb{1}_N$), then $c_k = 1/N$ and expression (6.18) shows that the network in that case will approach the average model,

$$w^\circ \triangleq \frac{1}{N}\sum_{k=1}^{N} z_k^\circ \tag{6.22}$$

which is not the intended objective.

6.2 Modified Diffusion Strategy

To deal with the problem of bias, we now show how to modify the diffusion strategy (2.29). We observe from (6.16) that the vector y cannot be zero because of agent 3 whose neighbors (agent 2 and itself in Fig. 6.2) observe data arising from a model that is different from the desired model. In addition, note from (6.6) that the bias term arises from the gradient direction used in computing the intermediate estimates in (2.29). These observations suggest that to ensure mean convergence, an agent should not combine intermediate estimates from neighbors whose observed model is different from the desired model. For this reason, we shall replace the intermediate estimates from these neighbors by their previous estimates $\{w_{l,i-1}\}$ in the combination step in (2.29). Specifically, we shall adjust

the diffusion strategy (2.29) in the following manner:

$$\psi_{k,i} = w_{k,i-1} + \mu_k \cdot u_{k,i}^T [d_k(i) - u_{k,i} w_{k,i-1}] \tag{6.23}$$

$$w_{k,i} = \sum_{l \in \mathcal{N}_k} \left(a_{l,k}^{(1)} \psi_{l,i} + a_{l,k}^{(2)} w_{l,i-1} \right) \tag{6.24}$$

where we are now using two sets of nonnegative entries $\{a_{l,k}^{(1)}\}$ and $\{a_{l,k}^{(2)}\}$. Their respective combination matrices A_1 and A_2 must satisfy

$$A_1 + A_2 = A \tag{6.25}$$

with A being the original left-stochastic matrix in (2.25). In other words, starting from the same combination matrix A used in (2.29), we are going to split its entries into two sets: some entries will be assigned to the matrix A_1 and the remaining entries will be assigned to the matrix A_2. The choice of which entries of A go into A_1 or A_2 will depend on which of the neighbors of agent k are observing data arising from a model that agrees with the desired objective for agent k (in the procedure to be developed in the sequel, each agent will be continuously updating two pieces of information related to what the agent thinks its own model is and to what the agent thinks the network's desired model is; these two models could be the same or different). Note that step (6.23) is the same as the first step in (2.29). However, in the second step (6.24), agents aggregate the $\{\psi_{l,i}, w_{l,i-1}\}$ from their neighborhood. With such adjustment, we will verify that by properly selecting $\{a_{l,k}^{(1)}, a_{l,k}^{(2)}\}$, mean convergence can be guaranteed for any connected network even in the presence of multiple source models.

6.2.1 Selection of Combination Matrices A_1 and A_2

To construct the matrices $\{A_1, A_2\}$ we associate two vectors with the network, f and g_i. Both vectors are of size N. The vector f is fixed and its kth entry, $f(k)$,

is set to $f(k) = 0$ when the observed model for agent k is w_0°; otherwise, it is set to $f(k) = 1$. In other words, the vector f reflects the models that are influencing the various agents in the network. On the other hand, the vector g_i is evolving with time; its kth entry is set to $g_i(k) = 0$ when the desired model for agent k is w_0°; otherwise, it is set equal to $g_i(k) = 1$. The decision by each agent about what the desired model should be is an evolving decision that changes with time and that is why we are indexing g with a time subscript. We will be describing a procedure for updating the vector g_i in a distributed manner so that all agents in the network will ultimately converge to an agreement about which model they want to converge to. This procedure runs in parallel with the diffusion strategy. Let us assume for the time being that the agents have achieved agreement on the desired model, which we are denoting by w_q°, so that

$$g_i(1) = g_i(2) = \cdots = g_i(N) = q, \quad \text{for all } i. \tag{6.26}$$

Obviously, the vectors $\{f, g_i\}$ still need to be determined. Nevertheless, assuming they are known, then we shall set the entries of A_1 and A_2 according to the following rules:

$$a_{l,k}^{(1)} = \begin{cases} a_{l,k}, & \text{if } l \in \mathcal{N}_k \text{ and } f(l) = g_i(k) \\ 0, & \text{otherwise} \end{cases} \tag{6.27}$$

$$a_{l,k}^{(2)} = \begin{cases} a_{l,k}, & \text{if } l \in \mathcal{N}_k \text{ and } f(l) \neq g_i(k) \\ 0, & \text{otherwise} \end{cases} \tag{6.28}$$

That is, agents that observe data arising from the same model that agent k wishes to converge to will be reinforced and their intermediate estimates $\{\psi_{l,i}\}$ will be used (and, hence, their combination weights are collected into matrix A_1). On the other hand, agents that observe data arising from a different model than the objective for agent k will be de-emphasized and their prior estimates $\{w_{l,i-1}\}$ will

be used in the combination step (6.24) (their combination weights are collected into matrix A_2).

6.2.2 Mean-Error Analysis

Now, we are ready to examine how the above construction helps remove the bias in the mean weight-error vector. Using relation (6.6) and the modified diffusion strategy (6.23)-(6.24), the recursion for the global error vector $\tilde{\boldsymbol{w}}_i$ is now given by:

$$\tilde{\boldsymbol{w}}_i = \underbrace{\left[\mathcal{A}_1^T(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i) + \mathcal{A}_2^T\right]}_{\boldsymbol{\mathcal{B}}_i} \cdot \tilde{\boldsymbol{w}}_{i-1} + \underbrace{\mathcal{A}_1^T\mathcal{M}\boldsymbol{\mathcal{R}}_i\tilde{z}^\circ - \mathcal{A}_1^T\mathcal{M}\boldsymbol{s}_i}_{\boldsymbol{y}_i} \tag{6.29}$$

where $\mathcal{A}_1$ and $\mathcal{A}_2$ are defined in a manner similar to $\mathcal{A}$ in (2.39). Taking the expectation of both sides of (6.29), we get the same recursion as (6.11) with the matrix $\mathcal{B}$ and the vector y defined in Table 6.1, i.e.,

$$\mathbb{E}\tilde{\boldsymbol{w}}_i = \left[\mathcal{A}_1^T(I_{NM} - \mu\mathcal{R}) + \mathcal{A}_2^T\right] \cdot \mathbb{E}\tilde{\boldsymbol{w}}_{i-1} + \mu\mathcal{A}_1^T\mathcal{R}\tilde{z}^\circ \tag{6.30}$$

The following result states that by constructing the combination weights according to (6.27)-(6.28), the modified diffusion strategy (6.23)-(6.24) converges in the mean.

Theorem 6.1. *Under Assumption 2.2, the mean recursion in (6.30) converges to zero if the matrices A_1 and A_2 are chosen according to (6.27)-(6.28) and the step-sizes $\{\mu_k\}$ satisfy condition (6.14) for those agents that observe data arising from the same model as the desired model w_q° for the network.*

Proof. See Appendix 6.A. □

We conclude from the arguments in the proof in Appendix 6.A that the net effect of the construction (6.27)-(6.28) is the following. Let w_q° denote the desired

model that the network wishes to converge to. We denote by $\mathcal{N}_q$ the subset of agents that receive data arising from the *same* model; these are the agents labeled $\{1, 2, \ldots, N_0\}$ in Appendix 6.A; here, we are designating the set more generically by $\mathcal{N}_q$. The remaining agents belong to the set $\mathcal{N}_q^c \triangleq \mathcal{N} \setminus \mathcal{N}_q$; these agents observe data arising from the other model. Agents that belong to the set $\mathcal{N}_q$ run the traditional diffusion strategy (2.29) using the combination matrix A and their step-sizes; these step-sizes are required to satisfy

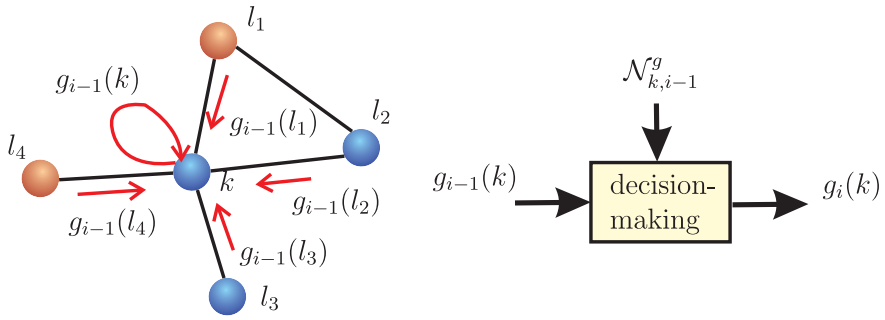
$$\mu_k < \frac{2}{\rho(R_{u,k})}, \quad \text{for all } k \in \mathcal{N}_q \tag{6.31}$$

The remaining agents set their step-sizes to zero and run only combination step of the diffusion strategy (2.29). These agents do not perform the adaptive update and therefore their estimates satisfy $\psi_{k,i} = w_{k,i-1}$ for all $k \in \mathcal{N}_q^c$.

6.3 Distributed Decision-Making

Theorem 6.1 establishes that it is possible for strongly connected networks to converge on average to a common desired model by using (6.23)-(6.24). However, the analysis so far has been based on the assumption that the agents know what are the observed models influencing their neighbors (i.e., they know $f(l)$ for their neighbors); they also need to know how to update their objective in $g_i(k)$ so that the $\{g_i(k)\}$ converge to the same value. This information is needed in (6.27)-(6.28) to construct the combination weights. In this section, we describe a distributed decision-making procedure by which the agents are able to achieve agreement on $\{g_i(k)\}$. In the next section, we develop a classification scheme to estimate $\{f(l)\}$ using available data.

The decision-making procedure is motivated by the process used by animal groups to reach agreement, and which is known as quorum sensing [23, 113, 138].



(a) Exchange (b) Decision-making

Figure 6.3: Decision-making process (a) agent k receives the desired models from its neighbors (b) agent k updates its desired model using (6.33)-(6.34).

The process is illustrated in Fig. 6.3 and described as follows. At time i, every agent k has its previous desired model $g_{i-1}(k)$. Agent k exchanges $g_{i-1}(k)$ with its neighbors and constructs the set

$$\mathcal{N}^g_{k,i-1} = \{l \mid l \in \mathcal{N}_k, g_{i-1}(l) = g_{i-1}(k)\} \tag{6.32}$$

That is, the set $\mathcal{N}^g_{k,i-1}$ contains the subset of agents that are in the neighborhood of k and have the same desired model at time $i-1$ as agent k. This set changes over time. Let $n^g_k(i-1)$ denote the number of agents in $\mathcal{N}^g_{k,i-1}$. Since at least one agent (agent k) belongs to $\mathcal{N}^g_{k,i-1}$, we have that $n^g_k(i-1) \geq 1$. Then, one way for agent k to participate in the quorum sensing process is to update its target model in $g_i(k)$ according to the following rule:

$$g_i(k) = \begin{cases} g_{i-1}(k), & \text{with probability } q_{k,i-1} \\ 1 - g_{i-1}(k), & \text{with probability } 1 - q_{k,i-1} \end{cases} \tag{6.33}$$

where the probability measure is computed as:

$$q_{k,i-1} = \frac{[n^g_k(i-1)]^K}{[n^g_k(i-1)]^K + [n_k - n^g_k(i-1)]^K} > 0 \tag{6.34}$$

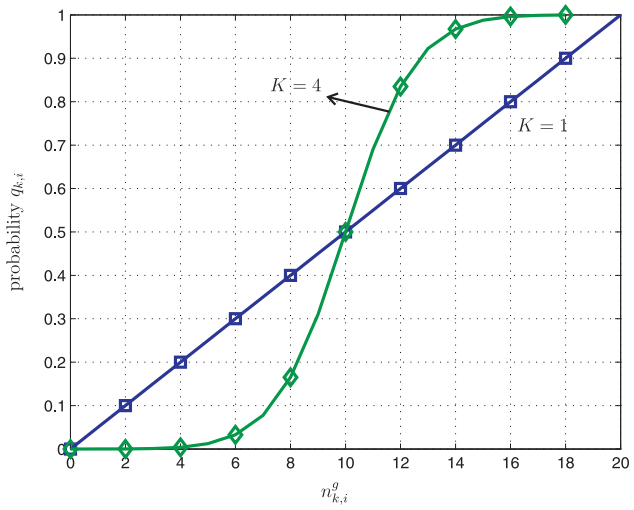


Figure 6.4: Illustration of the probability function $q_{k,i}$ in (6.34) for $n_k = 20$. The curves correspond to $K = 1$ and $K = 4$.

and the exponent K is a positive constant (e.g., $K = 4$). That is, agent k determines its desired model in a probabilistic manner, and the probability that agent k maintains its desired target is proportional to the Kth power of the number of neighbors having the same desired model. Figure 6.4 illustrates the sigmoidal shape of the probability function (6.34), which is similar to the form used by animal groups in their quorum response mechanisms (specifically, animals tend to follow a particular pattern of behavior when more of their neighbors switch to this same pattern [23, 113, 138]). We show later in Theorem 6.4 that choosing larger values for K speeds the rate at which the agents reach agreement on pursuing the same model. Using the above stochastic formulation, we are able to establish agreement on the desired model among the agents.

Theorem 6.2. *For a connected network starting from an arbitrary initial selection for the desired models vector g_i at time $i = -1$, and applying the update rule (6.33), then all agents in the network will eventually achieve agreement on the*

desired model, i.e.,

$$g_i(1) = g_i(2) = \ldots = g_i(N), \quad \text{as } i \to \infty \tag{6.35}$$

Proof. See Appendix 6.B. □

Although rule (6.32)-(6.34) ensures agreement on the decision vector, this construction is still not a distributed solution for one subtle but critical reason: agents need to agree on which index (0 or 1) to use to refer to either model $\{w_0^\circ, w_1^\circ\}$. This task would in principle require the agents to share some global information. We circumvent this difficulty and develop a distributed solution to the problem as follows. We associate with each agent k two local vectors $\{f_k, g_{k,i}\}$; these vectors will play the role of local estimates for the network vectors $\{f, g_i\}$. Each agent will then assign the index value of one to its observed model, i.e., each agent k sets $f_k(k) = 1$. Then, for every $l \in \mathcal{N}_k$, the entries $f_k(l)$ and $g_{k,i-1}(l)$ are set to one if they represent the same model as the one observed by agent k; otherwise, $f_k(l)$ and $g_{k,i-1}(l)$ are set to zero. The question still remains about how agent k knows whether its neighbors have the same observed and desired models as its observed model. To begin with, agent k knows its desired model value $g_{k,i-1}(k)$ from time $i-1$. To assign the remaining neighborhood entries in the vector $g_{k,i-1}$, the agents in the neighborhood of agent k first exchange their desired model indices with agent k, that is, they send the information $\{g_{l,i-1}(l),\ l \in \mathcal{N}_k\}$ to agent k. However, since $g_{l,i-1}(l)$ from agent l is set relative to its $f_l(l)$, agent k needs to set $g_{k,i-1}(l)$ based on the value of $f_k(l)$. Specifically, agent k will set $g_{k,i-1}(l)$ according to the rule:

$$g_{k,i-1}(l) = \begin{cases} g_{l,i-1}(l), & \text{if } f_k(l) = f_k(k) \\ 1 - g_{l,i-1}(l), & \text{otherwise} \end{cases} \tag{6.36}$$

That is, if agent l has the same observed model as agent k, then agent k simply assigns the value of $g_{l,i-1}(l)$ to $g_{k,i-1}(l)$.

In this way, computations that depend on the network vectors $\{f, g_i\}$ will be replaced by computations using the local vectors $\{f_k, g_{k,i}\}$. Note that since the operation of the network depends on the vectors $\{f, g_i\}$ through (6.27)-(6.28) and (6.32)-(6.33), these rules are now replaced by:

$$\mathcal{N}_{k,i-1}^{g} = \{l \mid l \in \mathcal{N}_k, g_{k,i-1}(l) = g_{k,i-1}(k)\} \tag{6.37}$$

$$g_{k,i}(k) = \begin{cases} g_{k,i-1}(k), & \text{with probability } q_{k,i-1} \\ 1 - g_{k,i-1}(k), & \text{with probability } 1 - q_{k,i-1} \end{cases} \tag{6.38}$$

$$a_{l,k}^{(1)} = \begin{cases} a_{l,k}, & \text{if } l \in \mathcal{N}_k \text{ and } f_k(l) = g_{k,i}(k) \\ 0, & \text{otherwise} \end{cases} \tag{6.39}$$

$$a_{l,k}^{(2)} = \begin{cases} a_{l,k}, & \text{if } l \in \mathcal{N}_k \text{ and } f_k(l) \neq g_{k,i}(k) \\ 0, & \text{otherwise} \end{cases} \tag{6.40}$$

In the following statement, we verify that using the network vectors $\{f, g_i\}$ is equivalent to using the local vectors $\{f_k, g_{k,i}\}$. From (6.37)-(6.40), it suffices to verify the following result.

Lemma 6.3. *It holds that*

$$f(l) \oplus g_i(k) = f_k(l) \oplus g_{k,i}(k) \tag{6.41}$$

$$g_i(l) \oplus g_i(k) = g_{k,i}(l) \oplus g_{k,i}(k) \tag{6.42}$$

where the symbol $\oplus$ denotes the exclusive-OR operation.

Proof. Since the values of $\{f_k(l), g_{l,i}(l), g_{k,i}(l)\}$ are set relative to $f_k(k)$, it holds

that

$$f(k) \oplus f(l) = f_k(k) \oplus f_k(l) \tag{6.43}$$

$$f(k) \oplus g_i(k) = f_k(k) \oplus g_{k,i}(k) \tag{6.44}$$

$$f(k) \oplus g_i(l) = f_k(k) \oplus g_{k,i}(l) \tag{6.45}$$

Then relations (6.41) and (6.42) hold in view of the fact:

$$(a \oplus b) \oplus (a \oplus c) = b \oplus c \tag{6.46}$$

for any a, b, and $c \in \{0, 1\}$. □

To implement (6.36), (6.39), and (6.40), agent k still needs to set the entries $\{f_k(l)\}$ that correspond to its neighbors, i.e., it needs to differentiate between their underlying models and whether their data arise from the same model as agent k or not. We propose next a procedure to determine f_k at agent k using the available estimates $\{w_{l,i-1}, \psi_{l,i}\}$ for $l \in \mathcal{N}_k$.

6.4 Model Classification Scheme

To determine the vector f_k, we introduce the belief vector $b_{k,i}$, whose lth entry, $b_{k,i}(l)$, will be a measure of the belief by agent k that agent l has the same observed model. The value of $b_{k,i}(l)$ lies in the range $[0, 1]$. The higher the value of $b_{k,i}(l)$ is, the more confidence agent k has that agent l is subject to the same model as its own model. In the proposed construction, the vector $b_{k,i}$ will be changing over time according to the estimates $\{w_{l,i-1}, \psi_{l,i}\}$. Agent k will be adjusting $b_{k,i}(l)$ according to the rule:

$$b_{k,i}(l) = \begin{cases} \alpha b_{k,i-1}(l) + (1 - \alpha), & \text{to increase belief} \\ \alpha b_{k,i-1}(l), & \text{to decrease belief} \end{cases} \tag{6.47}$$

for some positive scalar $\alpha \in (0,1)$; for example, $\alpha = 0.95$. That is, agent k increases the belief by linearly combining the belief from the previous time instant with one. Agent k then estimates $f_k(l)$ according to the rule:

$$\hat{f}_{k,i}(l) = \begin{cases} 1, & \text{if } b_{k,i}(l) \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{6.48}$$

where $\hat{f}_{k,i}(l)$ denotes the estimate of $f_k(l)$ at time i. Note that the value of $\hat{f}_{k,i}(l)$ may change over time due to the variations of $b_{k,i}(l)$. Since all agents have similar processing abilities, we introduce the assumption of homogeneous agents (see Assumption 3.1). We now develop a procedure that allows us to estimate the vectors $\{f_k\}$ by focusing on the behavior of the agents in the *far-field* regime when their weight estimates are far from their observed models. The far-field regime generally occurs during the initial stages of adaptation and, therefore, the vectors $\{f_k\}$ can be determined quickly during these initial iterations. We shall establish later in Eq. (6.97) in Theorem 6.3 that the agents are able to classify correctly the observed models of their neighbors with high probability approaching the value one by following the construction explained below.

To begin with, in order to determine whether the belief should be increased or decreased to implement (6.47), we refer to the update vector from (6.6), which can be written as follows for agent l:

$$\begin{aligned} \boldsymbol{h}_{l,i} &= \mu^{-1}(\boldsymbol{\psi}_{l,i} - \boldsymbol{w}_{l,i-1}) \\ &= \boldsymbol{u}_{l,i}^T \boldsymbol{u}_{l,i}(z_l^\circ - \boldsymbol{w}_{l,i-1}) + \boldsymbol{u}_{l,i}^T \boldsymbol{v}_l(i) \end{aligned} \tag{6.49}$$

Taking expectation of both sides conditioned on $\boldsymbol{w}_{l,i-1} = w_{l,i-1}$, we have that

$$\bar{h}_{l,i} \triangleq \mathbb{E}[\boldsymbol{h}_{l,i} \mid \boldsymbol{w}_{l,i-1} = w_{l,i-1}] = R_u(z_l^\circ - w_{l,i-1}) \tag{6.50}$$

That is, the expected update direction given the previous estimate, $w_{l,i-1}$, is a scaled vector pointing from $w_{l,i-1}$ towards z_l° with scaling matrix R_u (see Fig.

6.5). Note that since R_u is positive-definite, then the term $\bar{h}_{l,i}$ lies in the same half plane of the vector $z_l^\circ - w_{l,i-1}$. Therefore, the update vector provides useful information about the observed model, z_l°, at agent l. In addition, this term tells us how close the estimate at agent l is to its observed model. When the magnitude of $\bar{h}_{l,i}$ is large, or the estimate at agent l is far from z_l°, then we say that agent l is in a *far-field* regime (it is far from its observed model). On the other hand, when the magnitude of $\bar{h}_{l,i}$ is small, then the estimate $w_{l,i-1}$ is close to z_l° and we say that the agent is operating in a *near-field* regime (close to its observed model). The vector $\bar{h}_{l,i}$ can be estimated by the first-order recursion:

$$\hat{h}_{l,i} = (1-\nu)\hat{h}_{l,i-1} + \nu\mu^{-1}(\psi_{l,i} - w_{l,i-1}) \tag{6.51}$$

where we are denoting the estimate for $\bar{h}_{l,i}$ by $\hat{h}_{l,i}$ and ν is a positive step-size much smaller than one. Note that since the value of $\bar{h}_{l,i}$ varies with $w_{l,i-1}$, which is updated using the step-size μ, then, the value of ν should be set large enough compared to μ (i.e., $\nu \gg \mu$) so that recursion (6.51) can track variations in $\bar{h}_{l,i}$ over time. Note that since agents exchange $\{\psi_{l,i}, w_{l,i-1}\}$ in the combination step (6.24), agent k can compute $\hat{h}_{l,i}$ using (6.51) on its own if agent l is in its neighborhood. In the following, we describe how agent k updates the belief $b_{k,i}(l)$ for $l \in \mathcal{N}_k$.

During the initial stages adaptation, the agents k and l are away from their respective observed models and both agents are in the far-field. This state is characterized by the conditions $\|\hat{h}_{k,i}\| > \eta$ and $\|\hat{h}_{l,i}\| > \eta$ for some threshold η. If both agents have the same observed model, then the estimates $\hat{h}_{k,i}$ and $\hat{h}_{l,i}$ are expected to have similar direction towards the observed model (see Fig. 6.6(a)). Agent k will increase the belief value $b_{k,i}(l)$ using (6.47) if

$$\hat{h}_{k,i}^T \hat{h}_{l,i} > 0 \tag{6.52}$$

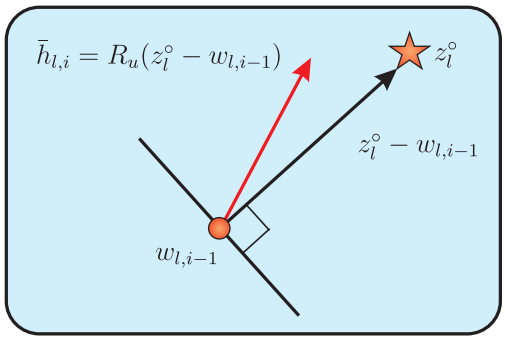


Figure 6.5: Illustration of the expected update vector $\bar{h}_{l,i}$ in (6.50).

Otherwise, agent k will decrease the belief $b_{k,i}(l)$. That is, when both agents are in the far-field, then agent k increases its belief that agent l shares the same observed model when the vectors $\hat{h}_{k,i}$ and $\hat{h}_{l,i}$ lie in the same quadrant. Note that, as shown in Fig. 6.6 (b), it is possible for agent k to increase $b_{k,i}(l)$ even when agents k and l have distinct models. This is because it is difficult to differentiate between the models during the initial stages of adaptation. This situation is handled by the evolving network dynamics as follows. If agent k considers that the data from agent l originate from the same model, then agent k will use the intermediate estimate $\psi_{l,i}$ from agent l in (6.24). Eventually, from Lemma 6.2, the estimates at these agents get close to a linear combination of the underlying models, which would then enable agent k to distinguish between the two models and to decrease the value of $b_{k,i}(l)$. Therefore, the belief $b_{k,i}(l)$ is updated according to the following rule:

$$b_{k,i}(l) = \begin{cases} \alpha b_{k,i-1}(l) + (1-\alpha), & \text{if } E_1 \\ \alpha b_{k,i-1}(l), & \text{if } E_1^c \end{cases} \tag{6.53}$$

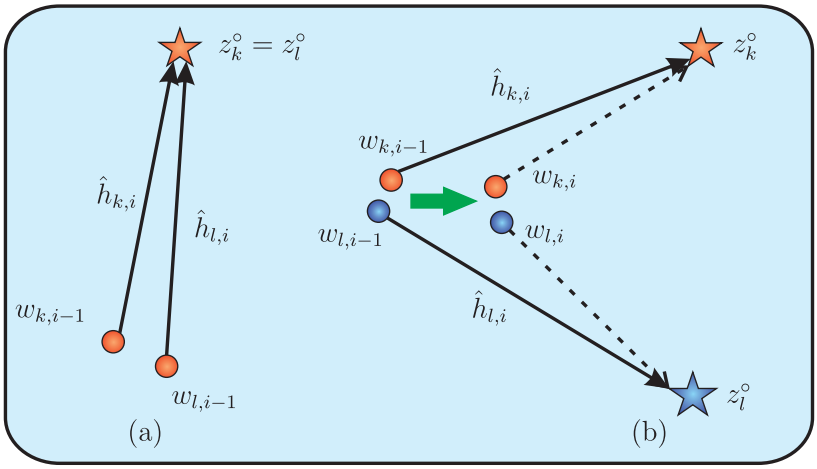


Figure 6.6: Illustration of the vectors $\hat{h}_{k,i}$ and $\hat{h}_{l,i}$ when both agents are in far-field and have (a) the same observed model or (b) different observed models.

where E_1 and E_1^c are two events of the form:

$$E_1 : \|\hat{h}_{k,i}\| > \eta, \|\hat{h}_{l,i}\| > \eta, \text{ and } \hat{h}_{k,i}^T \hat{h}_{l,i} > 0 \tag{6.54}$$

$$E_1^c : \|\hat{h}_{k,i}\| > \eta, \|\hat{h}_{l,i}\| > \eta, \text{ and } \hat{h}_{k,i}^T \hat{h}_{l,i} \leq 0 \tag{6.55}$$

Note that agent k updates the belief $b_{k,i}(l)$ only when both agents k and l are in the far-field.

6.5 Diffusion Strategy with Decision-making

Combining the modified diffusion strategy (6.23)-(6.24) and (6.39)-(6.40) with the decision-making process (6.34) and (6.37)-(6.38) and the classification scheme (6.48) and (6.53), we arrive at the listing in Algorithm. It is seen from the algorithm that the adaptation and combination steps of diffusion, which correspond to steps 1) and 8), are now separated by several steps. The purpose of these intermediate steps is to select the combination weights properly to carry out the

aggregation required by step 8). To implement the algorithm, agents need to exchange the quantities $\{w_{k,i-1}, \psi_{k,i}, g_{k,i-1}(k)\}$ with their neighbors. Note that if the agents can afford to extra information exchange, instead of every agent connected to agent l computing the term $\hat{h}_{l,i}$ in step 2), this term can be computed locally by agent l and shared with its neighbors.

Note that the combination weights are now time-varying because the agents are also updating the estimates $\{\hat{f}_{k,i}(l), g_{k,i}\}$ in steps 4) and 6). These updates run in parallel with the diffusion strategy, and their presence makes the analysis of the behavior of the algorithm more challenging due to the dependency among the steps. However, by examining the various steps closely, some useful facts stand out and help us address these challenges reasonably well. Specifically, it is observed that the convergence of the algorithm occurs in three phases as follows:

1. Convergence of the classification scheme: The first phase of convergence happens during the initial stages of adaptation. It is generally assumed that in this stage, all weight estimates are away from their respective models and the agents are operating in the far-field regime. Then, the agents use steps 2)-5) to determine the observed models $\{\hat{f}_{k,i}(l)\}$ of their neighbors. We establish later in Eq. (6.97) in Theorem 6.3 that this construction is able to identify the observed models correctly with probability approaching the value one. In other words, the classification scheme is able to converge well and fast during the initial stages of adaptation.

2. Convergence of the decision-making process: The second phase of convergence happens right after the convergence of the classification scheme, once the $\{\hat{f}_{k,i}(l)\}$ have converged. Because the agents now have correct information about their neighbor's observed models, they use steps 5)-6) to

Algorithm 6.1 Diffusion strategy with decision-making

For each agent k, initialize $w_{k,-1} = 0$, $\hat{h}_{k,-1} = 0$, $b_{k,-1}(l) = 0.5$, and $g_{k,-1}(k) = 1$.

for $i \geq 0$ and $k = 1$ to N **do**

1) Perform an adaptation step using the local data $\{d_k(i), u_{k,i}\}$:

$$\psi_{k,i} = w_{k,i-1} + \mu u_{k,i}^T [d_k(i) - u_{k,i} w_{k,i-1}]$$

2) Update the average update vectors $\{\hat{h}_{l,i}\}$ for $l \in \mathcal{N}_k$:

$$\hat{h}_{l,i} = (1-\nu)\hat{h}_{l,i-1} + \nu\mu^{-1}(\psi_{l,i} - w_{l,i-1})$$

3) Update the beliefs $\{b_{k,i}(l)\}$ for $l \in \mathcal{N}_k \setminus \{k\}$:

$$b_{k,i}(l) = \begin{cases} \alpha b_{k,i-1}(l) + (1-\alpha), & \text{if } E_1 \\ \alpha b_{k,i-1}(l), & \text{if } E_1^c \end{cases}$$

where E_1 and E_1^c are defined in (6.54)-(6.55).

4) Identify the observed models $\{\hat{f}_{k,i}(l)\}$ for $l \in \mathcal{N}_k \setminus \{k\}$:

$$\hat{f}_{k,i}(l) = \begin{cases} 1, & \text{if } b_{k,i}(l) \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

5) Collect the desired models $\{g_{k,i-1}(l)\}$ for $l \in \mathcal{N}_k \setminus \{k\}$ and construct the set $\mathcal{N}_{k,i-1}^g$ as follows:

$$g_{k,i-1}(l) = \begin{cases} g_{l,i-1}(l), & \text{if } \hat{f}_{k,i}(l) = 1 \\ 1 - g_{l,i-1}(l), & \text{otherwise} \end{cases}$$

$$\mathcal{N}_{k,i-1}^g = \{l \mid l \in \mathcal{N}_k, g_{k,i-1}(l) = g_{k,i-1}(k)\}$$

6) Update the desired model $g_{k,i}(k)$:

$$g_{k,i}(k) = \begin{cases} g_{k,i-1}(k), & \text{w.p. } q_{k,i-1} \\ 1 - g_{k,i-1}(k), & \text{w.p. } 1 - q_{k,i-1} \end{cases}$$

where the probability $q_{k,i-1}$ is defined in (6.34).

Algorithm 6.1 Diffusion strategy with decision-making (continued)

7) Adjust the combination weights $\{a_{l,k}^{(1)}\}$ and $\{a_{l,k}^{(2)}\}$:

$$a_{l,k,i}^{(1)} = \begin{cases} a_{l,k}, & \text{if } l \in \mathcal{N}_k \text{ and } \hat{f}_{k,i}(l) = g_{k,i}(k) \\ 0, & \text{otherwise} \end{cases}$$

$$a_{l,k,i}^{(2)} = \begin{cases} a_{l,k}, & \text{if } l \in \mathcal{N}_k \text{ and } \hat{f}_{k,i}(l) \neq g_{k,i}(k) \\ 0, & \text{otherwise} \end{cases}$$

8) Perform the combination step:

$$w_{k,i} = \sum_{l \in \mathcal{N}_k} \left(a_{l,k,i}^{(1)} \psi_{l,i} + a_{l,k,i}^{(2)} w_{l,i-1} \right)$$

end for

determine their own desired models $\{g_{k,i}(k)\}$. The convergence of this step is ensured by Eq. (6.35) in Theorem 6.2 and also by expression (6.114) in Theorem 6.4, which establishes that the convergence of this stage can be further improved by using larger values of the parameter K in the sigmoidal function of Figure 6.4.

3. Convergence of the diffusion strategy: After the classification and decision-making processes converge, the estimates $\{\hat{f}_{k,i}(l), g_{k,i}(l)\}$ do not change and the combination weights in step 7) remain fixed. Then, the diffusion strategy becomes unbiased and converges in the mean according to Theorem 6.1. Moreover, when the estimates are close to steady-state, those agents whose observed models are the same as the desired model enter the near-field regime and they stop updating their belief vectors (this will be justified by (6.93)).

6.6 Performance of Classification Procedure

It is clear that the success of the decision-making process depends on the reliability of the classification scheme (6.48). Therefore, it is important to examine the performance of model classification. In this section, we evaluate the probability of error for the classification step. There are two types of error. Specifically, when agents k and l are subject to the same observed model (i.e., $z_k^\circ = z_l^\circ$ and $f_k(l) = 1$), then one probability of error is defined as:

$$\begin{aligned} P_{e,1} &= \Pr\left(\hat{\boldsymbol{f}}_{k,i}(l) = 0 \mid f_k(l) = 1\right) \\ &= \Pr\left(\boldsymbol{b}_{k,i}(l) < 0.5 \mid z_k^\circ = z_l^\circ\right) \end{aligned} \tag{6.56}$$

where we used rule (6.48); note that we are denoting the variables $\hat{\boldsymbol{f}}_{k,i}(l)$ and $\boldsymbol{b}_{k,i}(l)$ in boldface to highlight the fact that they are now treated as random variables for the evaluation of the error probabilities. The second type of probability of error occurs when both agents have different observed models (i.e., when $z_k^\circ \neq z_l^\circ$ and $f_k(l) = 0$) and refers to the case:

$$\begin{aligned} P_{e,0} &= \Pr\left(\hat{\boldsymbol{f}}_{k,i}(l) = 1 \mid f_k(l) = 0\right) \\ &= \Pr\left(\boldsymbol{b}_{k,i}(l) > 0.5 \mid z_k^\circ \neq z_l^\circ\right) \end{aligned} \tag{6.57}$$

To evaluate the error probabilities in (6.56)-(6.57), we need to examine the probability distribution of the belief variable $\boldsymbol{b}_{k,i}$. Note from (6.53) that the belief variable (6.53) can be expressed as:

$$\boldsymbol{b}_{k,i}(l) = \alpha \boldsymbol{b}_{k,i-1}(l) + (1-\alpha)\boldsymbol{\xi}_{k,i}(l) \tag{6.58}$$

where $\boldsymbol{\xi}_{k,i}(l)$ is a Bernoulli random variable with

$$\boldsymbol{\xi}_{k,i}(l) = \begin{cases} 1, & \text{with probability } p \\ 0, & \text{with probability } 1-p \end{cases} \tag{6.59}$$

The value of p depends on whether the agents have the same observed models or not. First, when both agents have the same observed model (i.e., when $z_k^\circ = z_l^\circ$), the belief $\boldsymbol{b}_{k,i}(l)$ is supposed to be increased. The probability of detection, P_d, characterizes the probability that the belief $\boldsymbol{b}_{k,i}(l)$ is increased when $z_k^\circ = z_l^\circ$, i.e.,

$$P_d = \Pr(\boldsymbol{\xi}_{k,i}(l) = 1 \mid z_k^\circ = z_l^\circ) \tag{6.60}$$

In this case, the probability p will be replaced by P_d. On the other hand, when the observed models for the agents are distinct (i.e., when $z_k^\circ = z_l^\circ$), the probability of false alarm, P_f, characterizes the probability that the belief $\boldsymbol{b}_{k,i}(l)$ is increased when the belief is supposed to be decreased, i.e.,

$$P_f = \Pr(\boldsymbol{\xi}_{k,i}(l) = 1 \mid z_k^\circ \neq z_l^\circ) \tag{6.61}$$

and we replace the probability p by P_f. We will show later (see Lemma 6.5) how to evaluate the two probabilities P_d and P_f. In the sequel we denote them generically by p.

Now, expanding (6.58), we obtain

$$\boldsymbol{b}_{k,i}(l) = \alpha^{i+1}\boldsymbol{b}_{k,-1}(l) + (1-\alpha)\sum_{j=0}^{i}\alpha^j \boldsymbol{\xi}_{k,i-j}(l) \tag{6.62}$$

We assume that the $\{\boldsymbol{\xi}_{k,i}(l)\}$ are independent and identical distributed (i.i.d.) random variables with distribution (6.59). As i is large enough, the distribution of $\boldsymbol{b}_{k,i}(l)$ can be approximated by the distribution of the following random variable, which takes the form of a random geometric series:

$$\boldsymbol{\zeta}_k(l) \triangleq (1-\alpha)\sum_{j=0}^{\infty}\alpha^j \boldsymbol{\xi}_{k,j}(l) \tag{6.63}$$

where we replaced the index $i-j$ in (6.62) by j because the $\{\boldsymbol{\xi}_{k,i}(l)\}$ are i.i.d. There have been several useful works on the distribution function of random

geometric series [21, 53, 132]. However, it is generally untractable to express the distribution function in close form. We instead resort to the following two inequalities to establish bounds for the error probabilities (6.56)-(6.57). First, for any two generic events E_1 and E_2, if E_1 implies E_2, then the probability of event E_1 is less than the probability of event E_2 [110], i.e.,

$$\Pr(E_1) \leq \Pr(E_2) \quad \text{if} \quad E_1 \subseteq E_2. \tag{6.64}$$

The second inequality is the Markov inequality [110], i.e., for any nonnegative random variable $\boldsymbol{x}$ and positive scalar δ, it holds that

$$\Pr(x \geq \delta) = \Pr(x^2 \geq \delta^2) \leq \frac{\mathbb{E}x^2}{\delta^2} \tag{6.65}$$

Note that to apply the Markov inequality (6.65), we need the second-order moment of $\boldsymbol{\zeta}_k(l)$ in (6.63). However, since the $\{\boldsymbol{\xi}_{k,j}(l)\}$ are not zero mean, it is difficult to evaluate the moment. To circumvent this difficulty, let us introduce the change of variable:

$$\boldsymbol{\xi}^{\circ}_{k,j}(l) \triangleq \frac{\boldsymbol{\xi}_{k,j}(l) - p}{\sqrt{p(1-p)}} \tag{6.66}$$

It can be verified that the $\{\boldsymbol{\xi}^{\circ}_{k,j}(l)\}$ are i.i.d. Bernoulli random variables with zero mean and unit variance. Then, we can write

$$\boldsymbol{\xi}_{k,j}(l) = p + \sqrt{p(1-p)}\boldsymbol{\xi}^{\circ}_{k,j}(l) \tag{6.67}$$

so that

$$\begin{aligned}\boldsymbol{\zeta}_k(l) &= (1-\alpha)\sum_{j=0}^{\infty}\alpha^j[p + \sqrt{p(1-p)}\boldsymbol{\varepsilon}_{k,j}(l)] \\ &= p + \sqrt{p(1-p)}\boldsymbol{\zeta}^{\circ}_k(l)\end{aligned} \tag{6.68}$$

where

$$\boldsymbol{\zeta}^{\circ}_k(l) \triangleq (1-\alpha)\sum_{j=0}^{\infty}\alpha^j\boldsymbol{\xi}^{\circ}_{k,j}(l) \tag{6.69}$$

Since the $\{\boldsymbol{\xi}^{\circ}_{k,j}(l)\}$ are i.i.d. Bernoulli random variables with *zero* mean and unit variance, the mean and variance of $\boldsymbol{\zeta}^{\circ}_{k}$ are given by

$$\mathbb{E}\boldsymbol{\zeta}^{\circ}_{k}(l) = (1-\alpha)\sum_{j=0}^{\infty}\alpha^{j}\mathbb{E}\boldsymbol{\xi}^{\circ}_{k,j}(l) = 0 \tag{6.70}$$

$$\mathbb{E}(\boldsymbol{\zeta}^{\circ}_{k}(l))^{2} = (1-\alpha)^{2}\sum_{j=0}^{\infty}\alpha^{2j}\mathbb{E}(\boldsymbol{\xi}^{\circ}_{k,j}(l))^{2} = \frac{1-\alpha}{1+\alpha} \tag{6.71}$$

Then, from (6.56) and (6.68) and replacing the probability p by P_d, we obtain that

$$\begin{aligned} P_{e,1} &\approx \Pr(\boldsymbol{\zeta}_{k}(l) < 0.5 \mid z^{\circ}_{k} = z^{\circ}_{l}) \\ &= \Pr\left(\boldsymbol{\zeta}^{\circ}_{k}(l) < \frac{-(P_d - 0.5)}{\sqrt{P_d(1-P_d)}} \mid z^{\circ}_{k} = z^{\circ}_{l}\right) \\ &\leq \Pr\left(|\boldsymbol{\zeta}^{\circ}_{k}(l)| > \frac{|P_d - 0.5|}{\sqrt{P_d(1-P_d)}} \mid z^{\circ}_{k} = z^{\circ}_{l}\right) \\ &\leq \frac{1-\alpha}{1+\alpha}\cdot\frac{P_d(1-P_d)}{(P_d-0.5)^{2}} \end{aligned} \tag{6.72}$$

where we used (6.64) and the Markov inequality (6.65) in the last two inequalities. Note that in (6.72), we assume the value of P_d is greater than 0.5. Indeed, we will show in Lemma 6.5 that the value of P_d is close to one. Similarly, replacing the probability p by P_f and assuming that $P_f < 0.5$, we have from (6.57) and (6.68) that

$$\begin{aligned} P_{e,0} &\approx \Pr(\boldsymbol{\zeta}_{k}(l) > 0.5 \mid z^{\circ}_{k} \neq z^{\circ}_{l}) \\ &\leq \Pr\left(|\boldsymbol{\zeta}^{\circ}_{k}(l)| > \frac{|0.5 - P_f|}{\sqrt{P_f(1-P_f)}} \mid z^{\circ}_{k} \neq z^{\circ}_{l}\right) \\ &\leq \frac{1-\alpha}{1+\alpha}\cdot\frac{P_f(1-P_f)}{(0.5-P_f)^{2}} \end{aligned} \tag{6.73}$$

To evaluate the upper bounds in (6.72)-(6.73), we need the probabilities of detection and false alarm in (6.60)-(6.61). Since the update of $\boldsymbol{b}_{k,i}(l)$ in (6.53) depends

on $\{\hat{\boldsymbol{h}}_{k,i}, \hat{\boldsymbol{h}}_{l,i}\}$, we need to rely on the statistical properties of these latter quantities. In the following, we first examine the statistics of $\hat{\boldsymbol{h}}_{k,i}$ constructed via (6.51). We then evaluate the probability of detection and the probability of false alarm defined by (6.60)-(6.61), which will be subsequently used in determining the upper bounds for the probabilities of error (6.72)-(6.73).

6.6.1 Statistics of Update Direction

We first summarize the assumptions that are required in modeling $\hat{\boldsymbol{h}}_{k,i}$. As we mentioned following (6.51), since the step-sizes $\{\mu, \nu\}$ satisfy $\mu \ll \nu$, the variation of $\boldsymbol{w}_{k,i-1}$ can be assumed to be much slower than the variation of $\hat{\boldsymbol{h}}_{k,i}$. Therefore, the estimate $\boldsymbol{w}_{k,i-1}$ can be assumed to remain approximately constant during repeated updates of $\hat{\boldsymbol{h}}_{k,i}$. For this reason, the analysis in this section will be conditioned on $\boldsymbol{w}_{k,i-1} = w_{k,i-1}$, as we did in (6.50), and we introduce the following assumption.

Assumption 6.1 (Small step-size)**.** *The step-sizes $\{\mu, \nu\}$ are sufficiently small, i.e.,*

$$0 < \mu \ll \nu \ll 1 \tag{6.74}$$

so that

$$w_{k,i} \approx w_{k,i-1} \quad \textit{for all } k \tag{6.75}$$

In addition, since the update vector from (6.50) depends on the covariance matrix R_u, we assume R_u is well-conditioned so that the following is justified.

Assumption 6.2 (Regression model)**.** *The regression covariance matrix R_u is well-conditioned such that it holds that*

$$\textit{If } \|z_k^\circ - w_{k,i-1}\| \gg 1, \textit{ then } \|\bar{h}_{k,i}\| \gg \eta \tag{6.76}$$

$$\textit{If } \|z_k^\circ - w_{k,i-1}\| \ll 1, \textit{ then } \|\bar{h}_{k,i}\| \ll \eta \tag{6.77}$$

Moreover, the fourth-order moment of the regression data $\{\boldsymbol{u}_{k,i}\}$ *is assumed to be bounded such that*

$$\nu\tau \ll 1 \tag{6.78}$$

where the scalar τ *is a bound for*

$$\frac{\mathbb{E}\|\boldsymbol{u}_{k,i}^T\boldsymbol{u}_{k,i}(z_k^\circ - w_{k,i-1}) - \bar{h}_{k,i}\|^2}{\|\bar{h}_{k,i}\|^2} \leq \tau \tag{6.79}$$

and its value measures the randomness in variables involving fourth-order products of entries of $\boldsymbol{u}_{k,i}$.

Combining conditions (6.74) and (6.78), we obtain the following constraint on the step-sizes $\{\mu, \nu\}$:

$$0 \ll \mu \ll \nu \ll \min\{1, 1/\tau\} \tag{6.80}$$

To explain more clearly what conditions (6.76)-(6.77) entail, we obtain from (6.50) that $\|\bar{h}_{k,i}\|^2$ can be written as the weighted square Euclidean norm:

$$\|\bar{h}_{k,i}\|^2 = \|z_k^\circ - w_{k,i-1}\|_{R_u^2}^2 \tag{6.81}$$

We apply the Rayleigh-Ritz characterization of eigenvalues [64] to conclude that

$$\lambda_{\min}(R_u) \cdot \|z_k^\circ - w_{k,i-1}\| \leq \|\bar{h}_{k,i}\| \leq \lambda_{\max}(R_u) \cdot \|z_k^\circ - w_{k,i-1}\| \tag{6.82}$$

where $\lambda_{\min}(R_u)$ and $\lambda_{\max}(R_u)$ denote the minimum and maximum eigenvalues of R_u. Then, conditions (6.76)-(6.77) indicate that whenever agent k is operating in the far-field regime, i.e., whenever $\|z_k^\circ - w_{k,i-1}\| \gg 1$, then we would like to have

$$\lambda_{\min}(R_u) \cdot \|z_k^\circ - w_{k,i-1}\| \gg \eta \tag{6.83}$$

Likewise, whenever agent k is operating in the near-field regime, i.e., whenever $\|z_k^\circ - w_{k,i-1}\| \ll 1$, then we would like to have

$$\lambda_{\max}(R_u) \cdot \|z_k^\circ - w_{k,i-1}\| \ll \eta \tag{6.84}$$

Therefore, the scalars $\lambda_{\min}(R_u)/\eta$ and $\lambda_{\max}(R_u)/\eta$ cannot be too small or too large, i.e., the matrix R_u needs to be well-conditioned.

We are now ready to model the average update vector $\hat{\boldsymbol{h}}_{k,i}$. From (6.49) and (6.51), we have that

$$\hat{\boldsymbol{h}}_{k,i} = (1-\nu)\hat{\boldsymbol{h}}_{k,i-1} + \nu \boldsymbol{u}_{k,i}^T \boldsymbol{u}_{k,i}(z_k^\circ - w_{k,i-1}) + \nu \boldsymbol{u}_{k,i}^T \boldsymbol{v}_k(i) \tag{6.85}$$

According to Assumption 6.1, the estimate $w_{k,i-1}$ remains approximately constant over a certain period of time. To model $\hat{\boldsymbol{h}}_{k,i}$, we first remove the time index in $w_{k,i-1}$ and examine the statistics of $\hat{\boldsymbol{h}}_{k,i}$ under the condition $w_{k,i-1} = w_k$. From (6.85), the expected value of $\hat{\boldsymbol{h}}_{k,i}$ given $w_{k,i-1} = w_k$ converges to

$$\lim_{i\to\infty} \mathbb{E}\hat{\boldsymbol{h}}_{k,i} = R_u(z_k^\circ - w_k) \triangleq \bar{h}_k \tag{6.86}$$

We can also obtain from (6.49) and (6.51) that the limiting second-order moment of $\hat{\boldsymbol{h}}_{k,i}$, which is denoted by $\sigma_{\hat{h},k}^2$, satisfies:

$$\begin{aligned} \sigma_{\hat{h},k}^2 &\triangleq \lim_{i\to\infty} \mathbb{E}\|\hat{\boldsymbol{h}}_{k,i} - \bar{h}_k\|^2 \\ &= (1-\nu)^2\sigma_{\hat{h},k}^2 + \nu^2\sigma_{h,k}^2 \end{aligned} \tag{6.87}$$

where

$$\begin{aligned} \sigma_{h,k}^2 &\triangleq \mathbb{E}\|\boldsymbol{h}_{k,i} - \bar{h}_k\|^2 \\ &= \mathbb{E}\|\boldsymbol{u}_{k,i}^T \boldsymbol{u}_{k,i}(z_k^\circ - w_k) - \bar{h}_k\|^2 + \sigma_{v,k}^2 \mathrm{Tr}(R_u) \end{aligned} \tag{6.88}$$

Note that the cross term on the right-hand side of (6.87) is zero because the terms $\hat{\boldsymbol{h}}_{k,i-1} - \bar{h}_k$ and $\boldsymbol{h}_{k,i} - \bar{h}_k$ are independent and $\boldsymbol{h}_{k,i} - \bar{h}_k$ has zero mean. Then, from (6.87) and Assumption 6.1, the matrix $R_{\hat{h},k}$ is given by

$$R_{\hat{h},k} = \frac{\nu}{2-\nu} R_{h,k} \approx \frac{\nu}{2} R_{h,k} \tag{6.89}$$

Since $w_{k,i-1}$ remains approximately constant, the average update vector $\hat{\boldsymbol{h}}_{k,i}$ has mean and covariance matrix close to expressions (6.86) and (6.89) with w_k replaced by $w_{k,i-1}$. We then arrive at the following approximate model for $\hat{\boldsymbol{h}}_{k,i}$.

Assumption 6.3 (Model for $\hat{\boldsymbol{h}}_{k,i}$)**.** *The estimate $\hat{\boldsymbol{h}}_{k,i}$ is modeled as:*

$$\hat{\boldsymbol{h}}_{k,i} = \bar{h}_{k,i} + \boldsymbol{n}_{k,i} \tag{6.90}$$

where $\boldsymbol{n}_{k,i}$ is a random perturbation process with zero mean and

$$\mathbb{E}\|\boldsymbol{n}_{k,i}\|^2 \leq \frac{\nu[\tau\|\bar{h}_{k,i}\|^2 + \sigma_{v,k}^2 \mathrm{Tr}(R_u)]}{2} \tag{6.91}$$

with the scalar τ defined by (6.78).

Note that since the perturbation $\boldsymbol{n}_{k,i}$ is from the randomness of the regressor and noise processes $\{\boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$, then the $\{\boldsymbol{n}_{k,i}\}$ are independent of each other.

Before we proceed to the probability of detection and the probability of false alarm (6.60)-(6.61), we note that the update of the belief $\boldsymbol{b}_{k,i}(l)$ happens only when both agents k and l are in the far-field regime, which is determined by the magnitude of $\hat{\boldsymbol{h}}_{k,i}$ and $\hat{\boldsymbol{h}}_{k,i}$ being greater than the threshold η. The following result evaluates the probability that an agent is correctly classified to be in the far-field or near-field.

Lemma 6.4. *Under Assumptions 6.1-6.3, it holds that*

$$\Pr(\|\hat{\boldsymbol{h}}_{k,i}\| > \eta \mid \|z_k^\circ - w_{k,i-1}\| \gg 1) \geq 1 - \frac{\nu\tau}{2} \tag{6.92}$$

$$\Pr(\|\hat{\boldsymbol{h}}_{k,i}\| > \eta \mid \|z_k^\circ - w_{k,i-1}\| \ll 1) \leq \frac{\nu\sigma_{v,k}^2 \mathrm{Tr}(R_u)}{2\eta^2} \tag{6.93}$$

Proof. See Appendix 6.C. □

Under Assumptions 6.1 and 6.2, the probability in (6.92) is close to one and the probability in (6.93) is close to zero. Therefore, during the initial stage of

adaptation, the magnitude of $\{\|\hat{\boldsymbol{h}}_{k,i}\|\}$ successfully determines that the agents are in the far-field state and they update the belief using rule (6.53). When the estimates approach to steady-state, the agents whose observed models are the same as the desired model satisfy condition $\|z_k^\circ - w_{k,i-1}\| \ll 1$ and, therefore, they stop updating their belief vectors in view of (6.93). On the other hand, when both agents k and l have observed models that are different from the desired model (and, therefore, their estimates are away from their observed models), they will continue to update their beliefs. For this reason, we examine the probabilities of detection and false alarm in (6.60)-(6.61) and error probabilities in (6.56)-(6.57) under the following assumption.

Assumption 6.4 (Far-field regime)**.** *The estimates $\{w_{k,i-1}\}$ are far away from their corresponding observed models so that*

$$\|z_k^\circ - w_{k,i-1}\| \gg 1 \quad \textit{for all } k \tag{6.94}$$

The proof in Appendix 6.D then establishes the following bounds on P_d and P_f.

Lemma 6.5. *Under Assumptions 6.1-6.4, the probabilities of detection and false alarm defined by (6.60)-(6.61) are bounded by*

$$P_d \geq 1 - \frac{16\nu\tau}{\pi^2} \tag{6.95}$$

$$P_f \leq \frac{16\nu\tau}{\pi^2} \tag{6.96}$$

The above result establishes that the probability of detection is close to one and the probability of false alarm is close to zero in view of $\nu\tau \ll 1$. That is, with high probability, agent k will correctly adjust the value of $\boldsymbol{b}_{k,i}(l)$. We then arrive at the following bound for error probabilities.

Theorem 6.3. *Under Assumptions 6.1-6.4, the error probabilities* $\{P_{e,1}, P_{e,0}\}$ *are upper bounded by*

$$P_u = \frac{1-\alpha}{1+\alpha} \cdot \frac{16\nu\tau}{\pi^2} \cdot \frac{1-16\nu\tau/\pi^2}{(1/2-16\nu\tau/\pi^2)^2} = \mathcal{O}(\nu) \tag{6.97}$$

Proof. Let the function $f(p)$ be defined as

$$f(p) = \frac{p(1-p)}{(p-0.5)^2} \tag{6.98}$$

It can be verified that the function $f(p)$ is strictly increasing when $p \in [0, 0.5)$ and strictly decreasing when $p \in (0.5, 1]$. From Lemma 6.5, we conclude that $P_d > 0.5$ and $P_f < 0.5$. Therefore, an upper bound for $P_{e,1}$ can be obtained by replacing P_d in (6.72) by the lower bound in (6.95). Similarly, an upper bound for $P_{e,0}$ can be obtained by replacing P_f in (6.73) by the upper bound in (6.96). With these replacements, we obtain the upper bound P_u in (6.97). □

This result reveals that the error probabilities $\{P_{e,1}, P_{e,0}\}$ are upper bounded by the order of ν. In addition, the upper bound P_u also depends on the value of α used to update the belief in (6.53). We observe that the larger the value of α, the smaller the values of the error probabilities. In simulations, we choose $\nu = 0.05$ and $\alpha = 0.95$, which will give the upper bound in (6.97) the value $P_u \approx 0.008\tau < \nu\tau$. This implies that the classification scheme (6.48) identifies the observed models with high probability.

6.7 Rate of Convergence

The rate of convergence characterizes how fast information is transferred over the network [156]. It also characterizes the adaptation ability of the network. There are two rates of convergence to consider for adaptive networks running a

decision-making process of the form described in the earlier sections. First, we need to analyze the rate at which the agents arrive at agreement on a desired model (which corresponds to the speed of the decision-making process). Second, we analyze the rate at which the estimates by the agents converge to the desired model (which corresponds to the speed of the learning process). For both situations, we first evaluate the expressions for the convergence rates, as in (6.105) and (6.119), and then study how to improve the rates.

6.7.1 Convergence Rate of Decision-Making Process

From the proof of Theorem 6.2 (see Appendix 6.B), the decision-making process can be modeled as a Markov chain with $N+1$ states $\{\chi_i\}$ corresponding to the number of agents whose desired vectors are w_1°. The Markov chain has two absorbing states $\{0, N\}$ and its transition probability matrix P is shown in (6.148). The convergence rate of the decision-making process is then determined by the rate at which, starting at any arbitrary transient state, the Markov chain converges to one of the absorbing states. The argument that follows is meant to show that the rate of convergence of the decision making process improves with the parameter K used in (6.34); the larger the value of K the faster the convergence.

To arrive at this conclusion, we first remark that to assess the rate of convergence, we need to compute the jth power of P from (6.148) to find that

$$P^j = \begin{bmatrix} 1 & 0 & 0 \\ \bar{b} & Q^j & \bar{c} \\ 0 & 0 & 1 \end{bmatrix} \tag{6.99}$$

where $\{\bar{b}, \bar{c}\}$ are two $N \times 1$ vectors. Let the Markov chain start from any arbitrary

initial state distribution, y, of the form

$$y^T = \begin{bmatrix} 0 & y_Q^T & 0 \end{bmatrix} \tag{6.100}$$

where y_Q is a vector of size $N-1$ and its entries add up to one, i.e.,

$$y_Q^T \mathbb{1}_{N-1} = 1 \tag{6.101}$$

We shall select y_Q in a manner that enables us to determine how the convergence rate depends on K. Thus, note that the state distribution after j transitions becomes

$$y^T P^j = \begin{bmatrix} y_Q^T \bar{b} & y_Q^T Q^j & y_Q^T \bar{c} \end{bmatrix} \tag{6.102}$$

Therefore, the convergence rate is measured by the rate at which the matrix Q^j converges to zero, which is determined by the spectral radius of Q. Since Q is the sub-matrix of the transition probability matrix, all entries of Q are nonnegative, then by the Perron-Frobenius Theorem [64], the vector y_Q can be selected to be the left eigenvector of Q corresponding to the eigenvalue $\rho(Q)$, i.e.,

$$y_Q^T Q = \rho(Q) y_Q^T \tag{6.103}$$

Moreover, from (6.145)-(6.146), the matrix Q is primitive and, therefore, all entries of y_Q are positive. Furthermore, since the matrix P is right-stochastic (i.e., $P\mathbb{1}_{N+1} = \mathbb{1}_{N+1}$), from (6.148) it holds that

$$b + c + Q\mathbb{1}_{N-1} = \mathbb{1}_{N-1} \tag{6.104}$$

Pre-multiplying the vector y_Q on both sides of (6.104) and using (6.101) and (6.103), we obtain that the convergence rate of the decision-making process can be determined by

$$\begin{aligned} \rho(Q) &= y_Q^T Q \mathbb{1}_{N-1} \\ &= 1 - y_Q^T (b + c) \end{aligned} \tag{6.105}$$

We now determine the value of the vector sum $b+c$. We note from (6.143)-(6.144) that the transition probability matrix P and its sub-matrix Q are determined by the probability $q_{k,i-1}$ from (6.34), so is the spectral radius of Q. We further note from (6.34) that there is a single parameter K dictating the value of $q_{k,i-1}$. In the following, we examine the dependence of the convergence rate $\rho(Q)$ on the parameter K. It is generally challenging to develop the relation because the transition probability $p_{n,m}$ needs to be computed in a compounded way where we need to evaluate the summation of the products of $\{q_{k,i-1}\}$. Nevertheless, suppose that there are $\chi_{i-1} = n$ out of N agents with desired model w_1°, then, on average, agent k with n_k neighbors will have $n_k n/N$ neighbors whose desired model is w_1° and have $n_k(1 - n/N)$ neighbors whose desired model is w_0°. Then, from rule (6.33)-(6.34), every agent k chooses w_1° as its desired model with probability

$$\begin{aligned} q_n &\triangleq \frac{(n_k n/N)^K}{(n_k n/N)^K + (n_k(N-n)/N)^K} \\ &= \frac{n^K}{n^K + (N-n)^K} \end{aligned} \tag{6.106}$$

Note that the probability in (6.106) is independent of the agent index k and we denote this probability by q_n. Then, the probability in (6.144) can be evaluated in a way that there are m out of N agents choosing w_1° as their desired model and the remaining $N - m$ agents choosing w_0°, i.e., the probability in (6.144) is given by:

$$\Pr\left(\sum_{l=1}^{N} g_i(l) = m \mid g_{i-1}\right) = \binom{N}{m} q_n^m (1-q_n)^{N-m} \tag{6.107}$$

Note that the probability in (6.107) depends on g_{i-1} only through its sum, which is equal to n. Therefore, the transition probability $p_{n,m}$ in (6.143) has the same form as (6.107), i.e.,

$$p_{n,m} = \binom{N}{m} q_n^m (1-q_n)^{N-m} \tag{6.108}$$

To evaluate the spectral radius of Q from (6.105), we are interested in the value of $p_{n,0} + p_{n,N}$ (i.e., the nth entry of $b + c$), which is given by:

$$p_{n,0} + p_{n,N} = \frac{n^{NK} + (N-n)^{NK}}{(n^K + (N-n)^K)^N} \tag{6.109}$$

The following result establishes a monotonicity property of the sum in (6.109).

Lemma 6.6. *Let $f(x)$ be the function of the form*

$$f(x) = \frac{a^{Nx} + b^{Nx}}{(a^x + b^x)^N} \tag{6.110}$$

for some positive scalars $\{a, b, N\}$ with $N > 1$. Then, $f(x)$ is a non-decreasing function, i.e.,

$$f'(x) \geq 0 \tag{6.111}$$

with equality if, and only if, $a = b$.

Proof. Using

$$\frac{da^x}{dx} = a^x \ln(a) \tag{6.112}$$

we obtain that

$$f'(x) = \frac{a^{Nx} b^x}{(a^x + b^x)^{N+1}} \left[\left(\frac{b}{a} \right)^{(N-1)x} - 1 \right] \ln \left(\frac{b}{a} \right) \tag{6.113}$$

If $a > b$, then $b/a < 1$ and $\ln(b/a) < 0$. Therefore, $f'(x) > 0$. Conversely, if $a < b$, we still have that $f'(x) > 0$. Finally, if $a = b$, then $\ln(b/a) = 0$ and $f'(x) = 0$. □

Since the spectral radius of Q depends on the value of K in (6.33), we will index the quantities with the parameter K. For example, we denote the spectral radius of Q by $\rho[Q(K)]$. The following result relates the convergence rate of the decision-making process to the parameter K.

Theorem 6.4. *The spectral radius $\rho[Q(K)]$ is a strictly decreasing function of K for $N > 2$, i.e.,*

$$\rho[Q(K+1)] < \rho[Q(K)] \tag{6.114}$$

Proof. From (6.105), the spectral radius $\rho[Q(K)]$ is given by:

$$\rho[Q(K)] = 1 - \sum_{n=1}^{N-1} y_{Q,n}[p_{n,0}(K) + p_{n,N}(K)] \tag{6.115}$$

where $y_{Q,n}$ is the nth entry of the vector y_Q and the sum inside the brackets is shown in (6.109). From Lemma 6.6, we have that

$$p_{n,0}(K+1) + p_{n,N}(K+1) \geq p_{n,0}(K) + p_{n,N}(K) \tag{6.116}$$

with equality if, and only if, $n = N/2$. Therefore, if $N > 2$, there exists $n \in 1, 2, \cdots, N-1$ such that strict inequality holds in (6.116). Moreover, since the matrix Q is primitive, the $\{y_{Q,n}\}$ are positive and we arrive at (6.114). □

We therefore conclude that to improve the convergence rate of the decision-making process, the agents should use large K. When the value of K tends to infinity, the decision rule (6.33) converges to the majority rule, i.e.,

$$g_i(k) = \begin{cases} g_{i-1}(k), & \text{if } n_k^g(i) > n_k/2 \\ 1 - g_{i-1}(k), & \text{if } n_k^g(i) < n_k/2 \end{cases} \tag{6.117}$$

and $g_i(k)$ is set to 0 or 1 with equal probability if $n_k^g(i) = n_k/2$. Nevertheless, it may not be beneficial for the network to seek fast convergence during the decision making process because the network (e.g., a fish school) may converge to a bad model (e.g., a food source of poor quality). There exists a trade-off between exploration and exploitation, as in the case of multi-armed bandit problem [57, 89, 172]. Such trade-off can be taken into account by introducing some weighting scalar $\beta_k(i-1)$ that measures the quality of the desired model of agent k at time

$i-1$ relative to the other model. The higher values of $\beta_k(i-1)$, the better the quality of the model and the higher probability that agent k will maintain its desired model. Therefore, agent k adjusts the probability $q_{k,i-1}$ from (6.34) to

$$q_{k,i-1} = \frac{[\beta_k(i-1)n_k^g(i-1)]^K}{[\beta_k(i-1)n_k^g(i-1)]^K + [n_k - n_k^g(i-1)]^K} \tag{6.118}$$

6.7.2 Convergence Rate of Learning Process

We showed in Section 6.6 that with high probability, the agents are able to identify their neighbors' observed models during the initial stage of adaptation. Then, by the decision-making process in Section 6.3, the agents achieve agreement on a desired model. Therefore, in the following, we assume that the agents have achieved agreement on the desired model, say, w_q° as in (6.26). We know from the proof of Theorem 6.1 (see Appendix 6.A) that the evolution of the error recursion from (6.29) is equivalent to the error recursion of a network with a mixture of informed and uninformed agents, as studied in Chapter 5. That is, agents whose observed model is identical to its desired model ($f(l) = q$) are informed; otherwise they are uninformed. The convergence rate of the learning process specifies the rate at which the mean-square error converges to steady-state. In the context of networks with informed and uninformed agents, the convergence rate serves as a measure of how fast information is transferred (from informed agents to uninformed ones) over the networks. Using the results of Chapter 5, we can deduce that the convergence rate, denoted by r, of the modified diffusion strategy (6.23)-(6.24) is given by:

$$r = [\rho(\mathcal{B})]^2 \tag{6.119}$$

where $\mathcal{B}$ is defined in Table 6.1 and has the form

$$\mathcal{B} = \mathcal{A}^T(I_{NM} - \mathcal{M}_e\mathcal{R}) \tag{6.120}$$

with $\mathcal{M}_e$ defined in (6.140). The smaller the value of r is, the faster the rate of convergence is. Note that the value of r depends on the combination matrix A and on the spatial distribution of the informed agents through the matrix $\mathcal{M}_e$. Under Assumptions 3.1 and 6.1, it can be shown that the convergence rate is bounded by [157]:

$$(1-\mu\lambda_{\min}(R_u))^2 \leq r < 1 \tag{6.121}$$

To improve the efficiency of information transfer, it is desirable for the agents to select their combination weights so that the network has lower value of r. We will show that by appropriately selecting the combination matrix A in any connected network, the convergence rate (6.119) can achieve the lower bound provided by (6.121), that is, the network is able to converge to steady-state at the fastest rate.

Let $\mathcal{N}_I$ denote the set of informed agents, i.e.,

$$\mathcal{N}_I = \{l \mid f(l) = q\} \tag{6.122}$$

and let N_I denote the number of informed agents in the network. Without loss of generality, we assume the first N_I agents are informed, i.e., $\mathcal{N}_I = \{1, \ldots, N_I\}$. The combination matrix A can be partitioned in the following manner:

$$A = \left[\begin{array}{c|c} A_{II} & A_{IU} \\ \hline A_{UI} & A_{UU} \end{array}\right] \tag{6.123}$$

where the sub-matrices A_{II} and A_{UU} have size $N_I \times N_I$ and $(N-N_I)\times(N-N_I)$, respectively. Thus, the matrix A_{II} collects the weights among the informed agents and A_{UI} collects the weights from uninformed to informed agents; likewise for $\{A_{UU}, A_{IU}\}$. Then under Assumption 3.1, the matrix $\mathcal{B}$ from (6.120) can then be written as:

$$\mathcal{B} = \begin{bmatrix} A_{II}^T \otimes (I_M - \mu R_u) & A_{UI}^T \otimes I_M \\ A_{IU}^T \otimes (I_M - \mu R_u) & A_{UU}^T \otimes I_M \end{bmatrix} \tag{6.124}$$

The following result provides a condition on A so that the convergence rate achieves its lower bound.

Lemma 6.7. *Under Assumptions 3.1 and 6.1, if the sub-matrices $\{A_{UI}, A_{UU}\}$ of the combination matrix A in (6.123) satisfy:*

$$A_{UI} = 0 \quad \textit{and} \quad \rho(A_{UU}) \leq 1 - \mu\lambda_{\min}(R_u) \tag{6.125}$$

then the convergence rate r (6.119) of the modified diffusion strategy (6.23)-(6.24) achieves its lower bound, i.e., $r = (1 - \mu\lambda_{\min}(R_u))^2$.

Proof. Since $A_{UI} = 0$, the matrix $\mathcal{B}$ from (6.124) becomes lower block triangular, and its spectral radius is the maximum of $\rho(A_{II}^T \otimes (I_M - \mu R_u))$ and $\rho(A_{UU}^T \otimes I_M)$. Moreover, since $A_{UI} = 0$ and using (2.25), we conclude that A_{II} becomes left-stochastic. Hence, $\rho(A_{II}) = 1$ and it follows that

$$\begin{aligned}\rho(A_{II}^T \otimes (I_M - \mu R_u)) &= \rho(A_{II}^T) \cdot \rho(I_M - \mu R_u) \\ &= 1 - \mu\lambda_{\min}(R_u)\end{aligned} \tag{6.126}$$

where we used Assumption 6.1. Moreover, since $\rho(A_{UU}) \leq 1 - \mu\lambda_{\min}(R_u)$, we get

$$\rho(A_{UU}^T \otimes I_M) = \rho(A_{UU}^T) \cdot \rho(I_M) \leq 1 - \mu\lambda_{\min}(R_u) \tag{6.127}$$

Then, $\rho(\mathcal{B}) = 1 - \mu\lambda_{\min}(R_u)$. □

The following result shows that the lower bound of the convergence rate is achievable for any connected networks.

Theorem 6.5. *Under Assumptions 3.1 and 6.1 and for any connected network, there exists a combination matrix A such that the modified diffusion strategy (6.23)-(6.24) achieves the fastest convergence rate.*

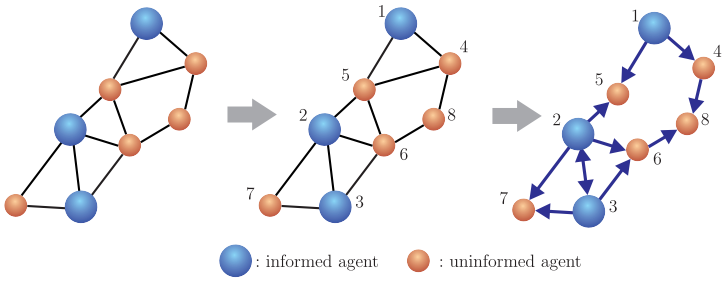


Figure 6.7: An illustration of a connected network with three informed agents (left) and one way to achieve the fastest convergence rate (right).

Proof. From Lemma 6.7, it suffices to show that we are able to construct a combination matrix A satisfying (6.125). First, we index the agents such that the smaller the distance (number of hops) from an agent to the set $\mathcal{N}_I$ is, the smaller the index of the agent is. This can be done by first indexing informed agents in any order, and then indexing the uninformed agents next to the informed agents in any order, and so on (see the middle plot of Fig. 6.7). Second, besides condition (2.25), we further require the weights $\{a_{l,k}\}$ to satisfy the following rules:

$$\begin{cases} \displaystyle\sum_{l\in\mathcal{N}_I\cap\mathcal{N}_k} a_{l,k} = 1, & \text{if } \mathcal{N}_I\cap\mathcal{N}_k \neq \emptyset \\ \displaystyle\sum_{l<k} a_{l,k} = 1, & \text{otherwise} \end{cases} \tag{6.128}$$

That is, if there are informed agents in the neighborhood of agent k, then it will assign positive combination weights to those agents only; otherwise, agent k will assign positive combination weights to neighbors with lower indices than k (i.e., those closer to informed agents). The example in Fig. 6.7 leads to a matrix A^T of the form below, where the directions of the arrows in the right plot of Fig. 6.7 indicate the allowed direction of information flow, i.e., the combination weights

in the reverse directions are zero:

$$A^T = \left[\begin{array}{ccc|ccccc} 1 & 0 & 0 & & & & & \\ 0 & a & 1-a & & & & & \\ 0 & b & 1-b & & & & & \\ \hline 1 & 0 & 0 & 0 & & & & \\ c & 1-c & 0 & 0 & 0 & & & \\ 0 & d & 1-d & 0 & 0 & 0 & & \\ 0 & e & 1-e & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & f & 0 & 1-f & 0 & 0 \end{array}\right] \tag{6.129}$$

with $a, b, c, d, e, f \in [0, 1]$. Since the weight at an informed agent assigned to an uninformed agent is always zero, $A_{UI} = 0$. In addition, for an uninformed agent k, the weight $a_{l,k}$ is equal to zero if $l \geq k$. Then, the matrix A_{UU}^T in (6.123) is a lower triangular matrix with zero diagonal entries and $\rho(A_{UU}) = 0 < 1 - \mu\lambda_{\min}(R_u)$. □

The proof of Theorem 6.5 suggests a way to construct the combination matrix so that the convergence rate achieves the minimum value (i.e., fastest convergence). However, there are two issues in such construction. First, the construction rules in (6.128) is difficult to be implemented in a distributed manner because the second rule in (6.128) requires spatial distribution of informed agents. The situation is more severe when the agents are able to move and the network topology evolves as well. Second, the constructed combination matrix is not primitive (i.e., Assumption 2.2 does not hold) because there are no links from uninformed agents to informed agents. Therefore, Theorem 6.1 does not apply here. In the following, we first propose a way to select combination weights that approximate rule (6.128) and then show that the approximate weights ensure mean convergence.

Let $\mathcal{N}_k^f$ denote the set of agents that are in the neighborhood of k and whose observed model is the same as the desired model w_q° (or, informed neighbors),

i.e.,

$$\mathcal{N}_k^f = \{l \mid l \in \mathcal{N}_k, l \in \mathcal{N}_I\} \tag{6.130}$$

Also, let n_k^f denote the number of agents in the set $\mathcal{N}_k^f$. The selection of combination weights is specified based on three types of agents: informed agents ($f(k) = q$), uninformed agents with informed neighbors ($f(k) \neq q$ and $n_k^f \neq 0$), and uninformed agents without informed neighbors ($f(k) \neq q$ and $n_k^f = 0$). The first two types correspond to the first case in (6.128) and their weights can satisfy rule (6.128) by setting

$$a_{l,k} = \begin{cases} 1/n_k^f, & \text{if } l \in \mathcal{N}_k^f \\ 0, & \text{otherwise} \end{cases} \tag{6.131}$$

That is, agent k places uniform weights on the informed neighbors and zero weights on the others. The last type of agents corresponds to the second case in (6.128). Since these agents do not know the distribution of informed agents, a convenient choice for the approximate weights they can select is for them to place zero weights on themselves and uniform weights on the others, i.e.,

$$a_{l,k} = \begin{cases} 1/(n_k - 1), & \text{if } l \in \mathcal{N}_k \text{ and } l \neq k \\ 0, & \text{otherwise} \end{cases} \tag{6.132}$$

Note that the weights from (6.131)-(6.132) can be set in a fully distributed manner and in real-time. To show the mean convergence of the modified diffusion strategy using the combination matrix A constructed from (6.131)-(6.132), we resort to Theorem 5.1. It states that the strategy converges in the mean for sufficiently small step-sizes if for any agent k, there exists an informed agent l and an integer power j such that

$$[A^j]_{l,k} > 0 \tag{6.133}$$

That is, for any agent, there exists a path with nonzero weight from an informed agent to itself. Condition (6.133) is clearly satisfied for the first two types of

agents. For any agent belonging to the last type, since the network is connected and from (6.132), there exists a path with nonzero weight from an agent of the second type (uninformed with informed neighbors) to itself. In addition, there exist direct links from informed agents to the agents of the second type, condition (6.133) is also satisfied. This implies that the modified diffusion strategy using the combination weights from (6.131)-(6.132) converges in the mean.

6.8 Simulation Results

We consider a network with 40 agents randomly connected. The model vectors are set to $w_0^\circ = [5; -5; 5; 5]$ and $w_1^\circ = [5; 5; -5; 5]$ (i.e. $M = 4$). Assume that the first 20 agents (agents 1 through 20) observe data originating from model w_0°, while the remaining agents observe data originating from model w_1°. The step-sizes are set to $\mu = 0.005$, $\nu = 0.05$, and $\alpha = 0.95$. The network employs the uniform combination rule: $a_{l,k} = 1/n_k$ if $l \in \mathcal{N}_k$.

In Fig. 6.8, we illustrate the network mean-square deviation (MSD) with respect to the two model vectors over time, i.e.,

$$\text{MSD}_q(i) = \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\|w_q^\circ - \boldsymbol{w}_{k,i}\|^2 \tag{6.134}$$

for $q = 0$ and $q = 1$. We compare the conventional ATC diffusion strategy (2.29) and the modified ATC diffusion strategy (6.23)-(6.24) with decision-making. We observe the bifurcation in MSD curves of the modified ATC diffusion strategy. Specifically, the MSD curve relative to the model w_0° converges to 23 dB, while the MSD relative to w_1° converges to -50 dB. This illustrates that the agents using the modified ATC diffusion are able to agree on a model and to converge to steady-state (to model w_1° in this case). We also show in Fig. 6.9 the evolution of the beliefs $\{b_{k,i}(l)\}$ for a particular agent using the update rule (6.53). The

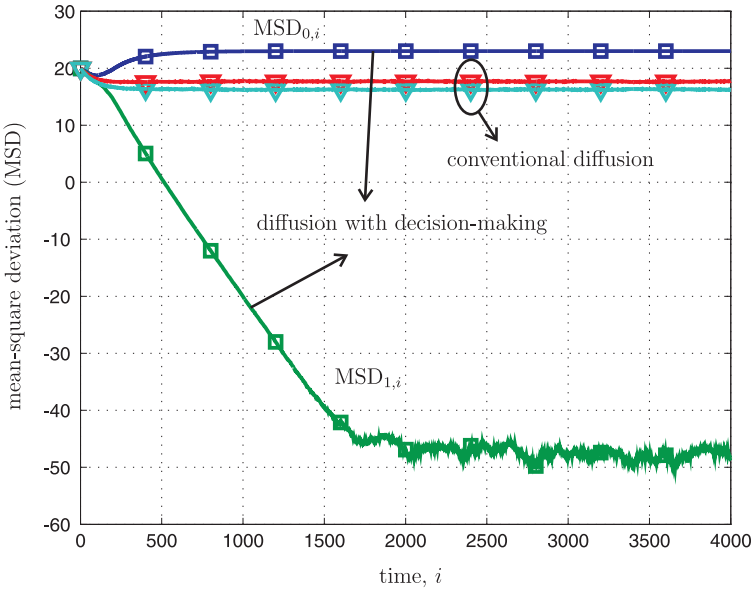


Figure 6.8: Transient network MSD over a network using the conventional diffusion strategy (2.29) and using the modified diffusion strategy (6.23)-(6.24). The network with decision-making converges to the model w_1° while the network without decision making converges to a vector that is not identical to either of the model vectors.

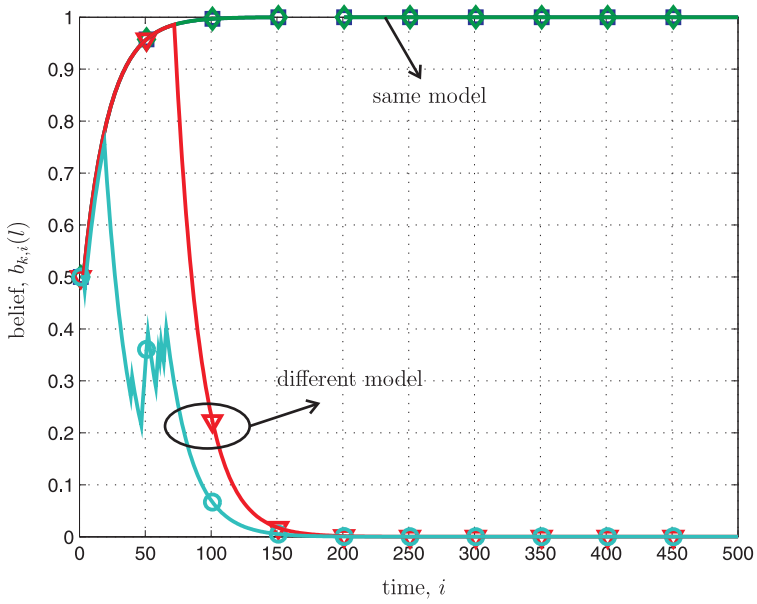


Figure 6.9: Evolution of beliefs using (6.53) at a particular agent. The agent has four neighbors; two of them collect data from the same model while the other two collect data from a different model.

agent has two neighbors observing data that originate from the same model and two neighbors observing data from a different model. We observe that, at the initial stage of adaptation, all beliefs increase. Nevertheless, as time evolves, the agent is able to differentiate between the two models and the beliefs for the latter two neighbors decrease. Note that the belief converges to one if an agent has the same observed model; otherwise, it converges to zero. This indicates that the classification scheme successfully identifies the observed models of neighboring agents. On the other hand, for the conventional diffusion strategy, the agents also converge because the MSD curves in Fig. 6.8 remain flat. However, the MSD values are large (about 18 dB). This implies that the agents converge to a common vector that does not coincide with either of the model vectors.

We also show the dependence of the convergence rate on the parameter K and the combination weights $\{a_{l,k}\}$. We first consider two modified diffusion strategies using decision-making with $K = 1$ and $K = 4$ in (6.34). The network MSD curves for these two strategies are shown in Fig. 6.10. We observe that the MSD curves relative to the model w_1° decrease at the same rate and converge to the same steady-state value. However, there is a shift in time domain between these curves: the $\text{MSD}_{1,i}$ with $K = 4$ is 75 time steps ahead of the MSD curve with $K = 1$. As the analytical results revealed, the decision-making processes with larger values of K achieve agreement at faster rate. We also consider the effect of the combination weights on the convergence rate of the adaptation strategies. Figure 6.11 illustrates the modified diffusion strategies with different combination weights: one with the uniform combination rule and the other one with the combination rule in (6.131)-(6.132). We observe that the diffusion strategy using the proposed rule converges at faster rate with some degradation in steady-state MSD. Note that the trade-off between convergence rate and MSD is also indicated in Chapter 5.

6.9 Concluding Remarks

In the presence of distinct objectives among the agents in a network, conventional distributed estimation strategies would lead to biased solutions. In this chapter, we proposed a modified strategy to address this issue. To do so, we allow the agents to exchange not only intermediate estimates, but also previous estimates. We also developed a classification scheme and a decision-making process for the agents to identify the underlying models that generate data and to achieve agreement among the agents on the desired objective. The proposed algorithms help model the mechanisms behind decision-making among biological networks.

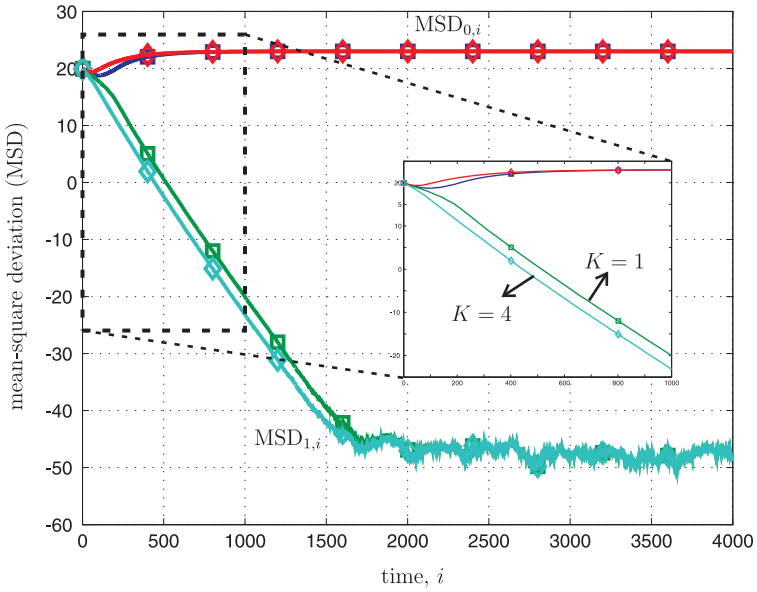


Figure 6.10: Transient network MSD over the modified diffusion strategies (6.23)-(6.24) with decision-making process for $K = 1$ and $K = 4$ in (6.34).

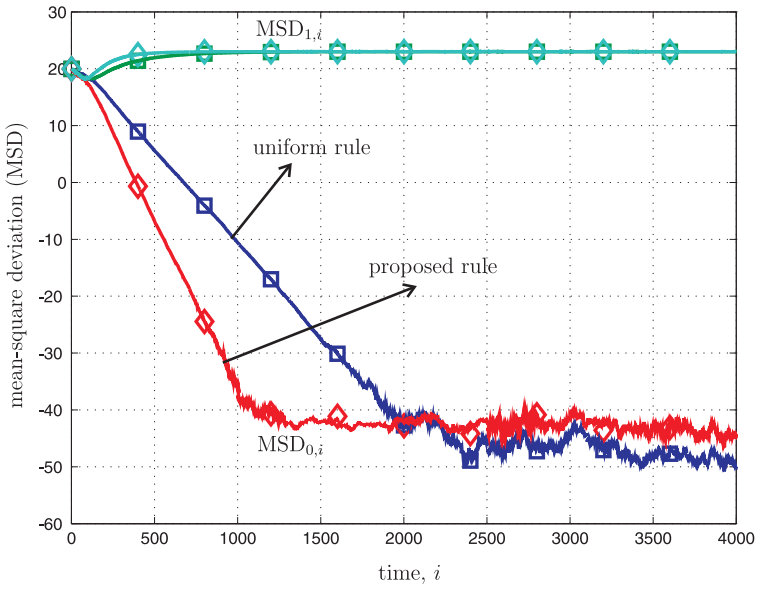


Figure 6.11: Transient network MSD over the modified diffusion strategy (6.23)-(6.24) using the uniform combination rule and the proposed rule (6.131)-(6.132).

6.A Proof of Theorem 6.1

From Lemma 6.1, it suffices to show that the vector $y = \mathcal{A}_1^T \mathcal{M}\mathcal{R}\tilde{z}^\circ$ is zero and the matrix $\mathcal{B} = \mathcal{A}_1^T(I_{NM} - \mathcal{M}\mathcal{R}) + \mathcal{A}_2^T$ has spectral radius strictly less than one. Without loss of generality, let w_0° be the desired model for the network (i.e., $q = 0$ in (6.26)) and assume there are N_0 agents with indices $\{1, 2, \ldots, N_0\}$ observing data arising from the model w_0°, while the remaining $N - N_0$ agents observe data arising from the model w_1°. Then, we obtain from (6.5), (6.27), and (6.28) that

$$\tilde{z}_k^\circ = \begin{cases} 0, & \text{if } k \leq N_0 \\ w_0^\circ - w_1^\circ, & \text{if } k > N_0 \end{cases} \tag{6.135}$$

$$a_{l,k}^{(1)} = 0 \text{ if } l > N_0 \tag{6.136}$$

$$a_{l,k}^{(2)} = 0 \text{ if } l \leq N_0 \tag{6.137}$$

Since the matrix $\mathcal{M}\mathcal{R}$ is block diagonal, we conclude that the vector y is zero. Moreover, we can write the (k,l)th block of $\mathcal{B}$ as

$$\begin{aligned} \mathcal{B}_{k,l} &= a_{l,k}^{(1)}(I_M - \mu_l R_{u,l}) + a_{l,k}^{(2)} I_M \\ &= \begin{cases} a_{l,k}(I_M - \mu_l R_{u,l}), & \text{if } l \in \mathcal{N}_k \text{ and } l \leq N_0 \\ a_{l,k} I_M, & \text{if } l \in \mathcal{N}_k \text{ and } l > N_0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{6.138}$$

Therefore, we can rewrite $\mathcal{B}$ as

$$\mathcal{B} = \mathcal{A}^T(I_{NM} - \mathcal{M}_e\mathcal{R}) \tag{6.139}$$

where $\mathcal{M}_e$ is an $N \times N$ block diagonal matrix of the form

$$\mathcal{M}_e \triangleq \text{diag}\{\mu_1 I_M, \cdots, \mu_{N_0} I_M, 0, \cdots, 0\} \tag{6.140}$$

That is, recursion (6.30) is equivalent to the mean recursion (6.11) of a network running the traditional diffusion strategy (2.29) with N_0 agents (agents 1 to N_0) using positive step-sizes and $N - N_0$ agents (agents $N_0 + 1$ to N) having zero step-sizes. Using the terminology from Chapter 5, we say that agents 1 to N_0 are informed while agents $N_0 + 1$ to N are uninformed. Then, according to Theorem 5.1 and under the assumption that the matrix A is primitive, if the step-sizes $\{\mu_1, \mu_2, \cdots, \mu_{N_0}\}$ are set to satisfy (6.14), then the spectral radius of $\mathcal{B}$ will be strictly less than one.

6.B Proof of Theorem 6.2

For a given vector g_{i-1}, we denote by χ_{i-1} the number of agents whose desired model is w_1° at time $i-1$, i.e.,

$$\chi_{i-1} = \sum_{k=1}^{N} g_{i-1}(k) \tag{6.141}$$

From (6.32)-(6.34), the vector g_i depends only on g_{i-1}. Thus, the value of χ_i depends only on χ_{i-1}. Therefore, the evolution of χ_i forms a Markov chain with $N+1$ states corresponding to the values $\{0, 1, 2, \ldots, N\}$ for χ_i and with the transition probability matrix P. The (n, m)th entry of P, denoted by $p_{n,m}$, represents the probability of transition from state $\chi_{i-1} = n$ to state $\chi_i = m$. To compute the transition probability $p_{n,m}$, let us denote by $\mathcal{G}_n$ the set of g_i satisfying $\chi_i = n$, i.e.,

$$\mathcal{G}_n = \left\{ g_i \mid \sum_{k=1}^{N} g_i(k) = n \right\} \tag{6.142}$$

Then, the $p_{n,m}$ can be written as:

$$\begin{aligned} p_{n,m} &= \Pr(\chi_i = m \mid \chi_{i-1} = n) \\ &= \sum_{g_{i-1} \in \mathcal{G}_n} \Pr(g_{i-1}) \Pr\left(\sum_{l=1}^{N} g_i(l) = m \mid g_{i-1} \right) \end{aligned} \tag{6.143}$$

where $\Pr(g_{i-1})$ is *a priori* probability and where

$$\begin{aligned} \Pr\left(\sum_{l=1}^{N} g_i(l) = m \mid g_{i-1} \right) &= \sum_{g_i \in \mathcal{G}_m} \Pr(g_i \mid g_{i-1}) \\ &= \sum_{g_i \in \mathcal{G}_m} \prod_{l=1}^{N} \Pr(g_i(l) \mid g_{i-1}(l)) \end{aligned} \tag{6.144}$$

with the probability $\Pr(g_i(l) \mid g_{i-1}(l))$ determined by (6.34). Note that for a fixed network topology, the transition probability $p_{n,m}$ is independent of the time index i, i.e., the Markov chain is *homogeneous* [85].

Now we assume that $\chi_{i-1} = n \neq 0, 1$ and $g_{i-1} \in \mathcal{G}_n$. Since the network is connected, at least one agent, say, agent k, has desired model w_1° and has a neighbor with distinct desired model w_0° so that $n_k^g(i-1) < n_k$ and $1 - q_{k,i-1} > 0$ from (6.34). In addition, since $n_l^g(i-1) > 0$ for all l, we have from (6.34) that $q_{l,i-1} > 0$ for all l. Then, we obtain from (6.143)-(6.144) that

$$\begin{aligned} p_{n,n-1} &= \sum_{g_{i-1} \in \mathcal{G}_n} \Pr(g_{i-1}) \Pr\left(\sum_{l=1}^{N} g_i(l) = n-1 \mid g_{i-1} \right) \\ &\geq \sum_{g_{i-1} \in \mathcal{G}_n} \Pr(g_{i-1}) \Pr(g_i(k) = 0 \mid g_{i-1}) \prod_{l \neq k} \Pr(g_i(l) = g_{i-1}(l) \mid g_{i-1}) \\ &= \sum_{g_{i-1} \in \mathcal{G}_n} \Pr(g_{i-1})(1 - q_{k,i-1}) \prod_{l \neq k} q_{l,i-1} \\ &> 0 \end{aligned} \tag{6.145}$$

Similarly, it can be verified that

$$p_{n,n} > 0 \quad \text{and} \quad p_{n,n+1} > 0 \tag{6.146}$$

for $n \neq 0, N$. On the other hand, for $n = 0$ or $n = N$, we have that

$$p_{0,0} = p_{N,N} = 1 \tag{6.147}$$

This indicates that the Markov chain has two absorbing states: $\chi_i = 0$ (or, $g_i(1) = g_i(2) = \cdots = g_i(N) = 0$) and $\chi_i = N$ (or, $g_i(1) = g_i(2) = \cdots = g_i(N) = 1$), and for any state χ_i different from 0 and N, there is a nonzero probability traveling from an arbitrary state χ_i to state 0 and state N. That is, the transition probability matrix P can be written as:

$$P = \begin{bmatrix} 1 & 0 & 0 \\ b & Q & c \\ 0 & 0 & 1 \end{bmatrix} \tag{6.148}$$

where the matrix Q of size $(N-1) \times (N-1)$ is the transition matrix among the transient states $\{1, 2, \cdots, N-1\}$ and the vectors $\{b, c\}$ of size $N-1$ are the transition probabilities from the transient states to the absorbing states. This implies that no matter which state the Markov chain starts from, the state converges to state 0 or state N [85, p.26], i.e., all agents reach agreement on the desired model.

6.C Proof of Lemma 6.4

Let C_1 denote the following far-field condition:

$$C_1 : \|z_k^\circ - w_{k,i-1}\| \gg 1 \tag{6.149}$$

We obtain from (6.90) that

$$\begin{aligned}\Pr(\|\hat{\boldsymbol{h}}_{k,i}\| > \eta \mid C_1) &= \Pr(\|\bar{h}_{k,i} + \boldsymbol{n}_{k,i}\| > \eta \mid C_1) \\ &\overset{(a)}{\geq} \Pr(\|\bar{h}_{k,i}\| - \|\boldsymbol{n}_{k,i}\| > \eta \mid C_1) \\ &= \Pr(\|\boldsymbol{n}_{k,i}\| < \|\bar{h}_{k,i}\| - \eta \mid C_1) \\ &= 1 - \Pr(\|\boldsymbol{n}_{k,i}\| \geq \|\bar{h}_{k,i}\| - \eta \mid C_1) \\ &\overset{(b)}{\geq} 1 - \frac{\mathbb{E}\|\boldsymbol{n}_{k,i}\|^2}{(\|\bar{h}_{k,i}\| - \eta)^2} \\ &\overset{(c)}{=} 1 - \frac{\nu[\tau\|\bar{h}_{k,i}\|^2 + \sigma_{v,k}^2 \mathrm{Tr}(R_u)]}{2(\|\bar{h}_{k,i}\| - \eta)^2} \end{aligned} \tag{6.150}$$

where step (a) follows from the triangle inequality of norm that $\|\bar{h}_{k,i} + \boldsymbol{n}_{k,i}\| \geq \|\bar{h}_{k,i}\| - \|\boldsymbol{n}_{k,i}\|$ and (6.64), step (b) is by the Markov inequality (6.65) and Assumption 6.2, and step (c) is by (6.91). Moreover, under Assumption 6.4 and condition C_1, we can ignore the term η in the denominator of (6.150). In addition, from condition C_1 and (6.81), and since the variance $\sigma_{v,k}^2$ is generally small, we may ignore the term $\nu\sigma_{v,k}^2 \mathrm{Tr}(R_u)$ in (6.150) and obtain (6.92).

On the other hand, we consider the alternative near-field condition

$$C_2 : \|z_k^\circ - w_{k,i-1}\| \ll 1 \tag{6.151}$$

Using similar arguments that led to (6.150), we obtain that

$$\begin{aligned}\Pr(\|\hat{\boldsymbol{h}}_{k,i}\| > \eta \mid C_2) &= \Pr(\|\bar{h}_{k,i} + \boldsymbol{n}_{k,i}\| > \eta \mid C_2) \\ &\leq \Pr(\|\bar{h}_{k,i}\| + \|\boldsymbol{n}_{k,i}\| > \eta \mid C_2) \\ &= \Pr(\|\boldsymbol{n}_{k,i}\| > \eta - \|\bar{h}_{k,i}\| \mid C_2) \\ &\leq \frac{\mathbb{E}\|\boldsymbol{n}_{k,i}\|^2}{(\eta - \|\bar{h}_{k,i}\|)^2} \\ &= \frac{\nu[\tau\|\bar{h}_{k,i}\|^2 + \sigma_{v,k}^2 \mathrm{Tr}(R_u)]}{2(\eta - \|\bar{h}_{k,i}\|)^2} \end{aligned} \tag{6.152}$$

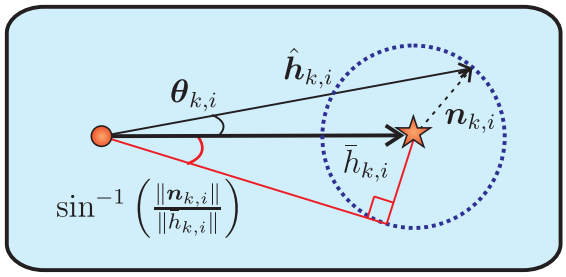


Figure 6.12: Illustration of the angle $\boldsymbol{\theta}_{k,i}$ between $\bar{h}_{k,i}$ and $\hat{\boldsymbol{h}}_{k,i}$ due to the noise $\boldsymbol{n}_{k,i}$.

Under Assumption 6.4, we can ignore the term $\|\bar{h}_{k,i}\|^2$ in the denominator of (6.152). Moreover, when agent k is in near-field, the square error $\|z_k^\circ - w_{k,i-1}\|^2$ is of the order of $\mu\sigma_{v,k}^2$ [157]. Then, using the sub-multiplicative property of norms [64], we obtain from (6.81) that

$$\begin{aligned}\|\bar{h}_{k,i}\|^2 &\leq \|R_u\|^2 \cdot \|z_k^\circ - w_{k,i-1}\|^2 \\ &= \mathcal{O}(\mu\sigma_{v,k}^2\rho^2(R_u))\end{aligned} \tag{6.153}$$

Therefore, under the small step-size assumption on μ, we can also ignore the term $\tau\|\bar{h}_{k,i}\|$ in (6.152) and obtain (6.93).

6.D Proof of Lemma 6.5

6.D.1 Probability of detection when $z_k^\circ = z_l^\circ$

Under Assumption 6.4 and from (6.53), the probability P_d in (6.60) becomes

$$\begin{aligned}P_d &= \Pr(\|\hat{\boldsymbol{h}}_{k,i}\| > \eta,\ \|\hat{\boldsymbol{h}}_{l,i}\| > \eta,\ \hat{\boldsymbol{h}}_{k,i}^T\hat{\boldsymbol{h}}_{l,i} > 0 \mid z_k^\circ = z_l^\circ) \\ &\approx \Pr(\hat{\boldsymbol{h}}_{k,i}^T\hat{\boldsymbol{h}}_{l,i} > 0 \mid z_k^\circ = z_l^\circ)\end{aligned} \tag{6.154}$$

where we used Lemma 6.4 and the fact that $\hat{\boldsymbol{h}}_{k,i}$ and $\hat{\boldsymbol{h}}_{l,i}$ are independent. Note that the event $\hat{\boldsymbol{h}}_{k,i}^{T}\hat{\boldsymbol{h}}_{l,i} > 0$ is equivalent to the fact that the angle between these two vectors is less than $\pi/2$. Let $\boldsymbol{\theta}_{k,i}$ denote the angle between the vectors $\bar{h}_{k,i}$ and $\hat{\boldsymbol{h}}_{k,i}$ due to the noise $\boldsymbol{n}_{k,i}$ (see Fig. 6.12). The value of $\boldsymbol{\theta}_{k,i}$ is positive if the vector $\hat{\boldsymbol{h}}_{k,i}$ rotates counter-clockwise relative to $\bar{h}_{k,i}$; otherwise, its value is negative. Then, we have that the angle $\boldsymbol{\theta}_{k,i}$ is upper bounded by:

$$|\boldsymbol{\theta}_{k,i}| \leq \sin^{-1}\left(\frac{\|\boldsymbol{n}_{l,i}\|}{\|\bar{h}_{k,i}\|}\right) \tag{6.155}$$

That is, the maximum value of $\boldsymbol{\theta}_{k,i}$ occurs when the vectors $\hat{\boldsymbol{h}}_{k,i}$ and $\boldsymbol{n}_{k,i}$ are perpendicular (see Fig. 6.12). Under Assumption 6.4 and from (6.91), it holds that

$$\frac{\mathbb{E}\|\boldsymbol{n}_{k,i}\|^2}{\|\bar{h}_{k,i}\|^2} \leq \frac{\nu[\tau\|\bar{h}_{k,i}\|^2 + \sigma_{v,k}^2\mathrm{Tr}(R_u)]}{2\|\bar{h}_{k,i}\|^2} \approx \frac{\nu\tau}{2} \tag{6.156}$$

and, therefore, the right-hand side of (6.155) can be approximated by

$$\sin^{-1}\left(\frac{\|\boldsymbol{n}_{k,i}\|}{\|\bar{h}_{k,i}\|}\right) \approx \frac{\|\boldsymbol{n}_{k,i}\|}{\|\bar{h}_{k,i}\|} \tag{6.157}$$

Moreover, since all agents start from the same initial estimate (i.e., $w_{k,-1} = 0$ for all k), the estimates $\{w_{k,i-1}\}$ are close to each other during the initial stages of adaptation and it is reasonable to assume that

$$\|w_{k,i-1} - w_{l,i-1}\| \ll \|z_k^\circ - w_{k,i-1}\| \tag{6.158}$$

Therefore, we arrive at the following approximation (note that $z_k^\circ = z_l^\circ$) for computing P_d:

$$\begin{aligned}
\bar{h}_{l,i} &= R_u(z_l^\circ - w_{l,i-1}) \\
&= R_u(z_k^\circ - w_{k,i-1} + w_{k,i-1} - w_{l,i-1}) \\
&\approx R_u(z_k^\circ - w_{k,i-1}) \\
&= \bar{h}_{k,i}
\end{aligned} \tag{6.159}$$

This implies that the vectors $\hat{\boldsymbol{h}}_{k,i}$ and $\hat{\boldsymbol{h}}_{l,i}$ can be modeled as starting at the same location $w_{k,i-1}$ but having deviated by angles $\boldsymbol{\theta}_{k,i}$ and $\boldsymbol{\theta}_{l,i}$, respectively. Therefore, as shown in Fig. 6.13(a), the angle between $\hat{\boldsymbol{h}}_{k,i}$ and $\hat{\boldsymbol{h}}_{l,i}$ is equal to $|\boldsymbol{\theta}_{k,i} - \boldsymbol{\theta}_{l,i}|$. From (6.154), we obtain that

$$\begin{aligned}
P_d &\approx \Pr\left(|\boldsymbol{\theta}_{k,i} - \boldsymbol{\theta}_{l,i}| < \frac{\pi}{2} \mid z_k^\circ = z_l^\circ\right) \\
&\overset{(a)}{\geq} \Pr\left(|\boldsymbol{\theta}_{k,i}| + |\boldsymbol{\theta}_{l,i}| < \frac{\pi}{2} \mid z_k^\circ = z_l^\circ\right) \\
&\overset{(b)}{\geq} \Pr\left(\frac{\|\boldsymbol{n}_{k,i}\|}{\|\bar{h}_{k,i}\|} + \frac{\|\boldsymbol{n}_{l,i}\|}{\|\bar{h}_{k,i}\|} < \frac{\pi}{2}\right) \\
&= \Pr(\|\boldsymbol{n}_{k,i}\| + \|\boldsymbol{n}_{l,i}\| < \pi\|\bar{h}_{k,i}\|/2) \\
&= 1 - \Pr(\|\boldsymbol{n}_{k,i}\| + \|\boldsymbol{n}_{l,i}\| \geq \pi\|\bar{h}_{k,i}\|/2)
\end{aligned} \tag{6.160}$$

where step (a) is by the triangle inequality of norms and (6.64) and step (b) is by (6.155) and (6.157). To evaluate the probability in (6.160), we resort to the following fact. For any two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ and for any constant η, it holds from (6.64) that

$$\begin{aligned}
\Pr(\boldsymbol{x} + \boldsymbol{y} > \eta) &\leq \Pr(\boldsymbol{x} > \eta/2 \ \text{ or } \ \boldsymbol{y} > \eta/2) \\
&\leq \Pr(\boldsymbol{x} > \eta/2) + \Pr(\boldsymbol{y} > \eta/2)
\end{aligned} \tag{6.161}$$

where in the last inequality, we used the union bound of probabilities [110]. Therefore, we arrive at (6.95) because

$$\begin{aligned}
P_d &\geq 1 - \Pr(\|\boldsymbol{n}_{k,i}\| > \pi\|\bar{h}_{k,i}\|/4) - \Pr(\|\boldsymbol{n}_{l,i}\| > \pi\|\bar{h}_{k,i}\|/4) \\
&\geq 1 - \frac{16(\mathbb{E}\|\boldsymbol{n}_{k,i}\|^2 + \mathbb{E}\|\boldsymbol{n}_{l,i}\|^2)}{\pi^2\|\bar{h}_{k,i}\|^2} \\
&\geq 1 - \frac{16\nu\tau}{\pi^2}
\end{aligned} \tag{6.162}$$

where we used the Markov inequality (6.65) and expressions (6.156) and (6.159).

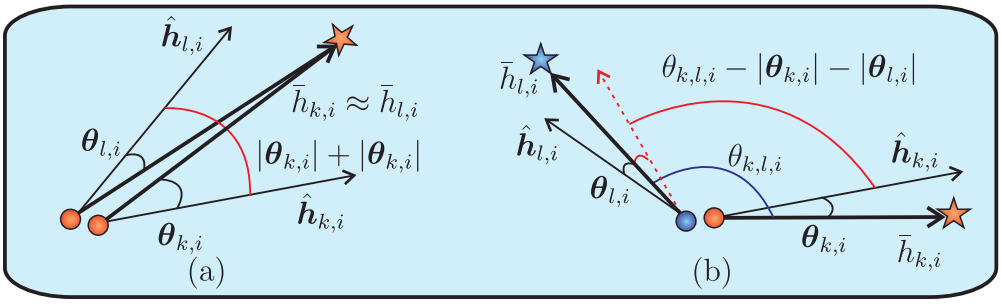


Figure 6.13: Illustration of the angle between $\hat{\boldsymbol{h}}_{k,i}$ and $\hat{\boldsymbol{h}}_{l,i}$ when both agents have (a) the same observed model ($z_k^\circ = z_l^\circ$) or (b) different observed models ($z_k^\circ \neq z_l^\circ$).

6.D.2 Probability of false alarm when $z_k^\circ \neq z_l^\circ$

Similarly, the probability of false alarm from (6.61) is upper bounded by:

$$\begin{aligned} P_f &= \Pr(\|\hat{\boldsymbol{h}}_{k,i}\| > \eta,\ \|\hat{\boldsymbol{h}}_{l,i}\| > \eta,\ \hat{\boldsymbol{h}}_{k,i}^T \hat{\boldsymbol{h}}_{l,i} > 0 \mid z_k^\circ \neq z_l^\circ) \\ &\leq \Pr(\hat{\boldsymbol{h}}_{k,i}^T \hat{\boldsymbol{h}}_{l,i} > 0 \mid z_k^\circ \neq z_l^\circ) \end{aligned} \tag{6.163}$$

where we used (6.64). We again convert to the angular domain. First, under Assumption 6.4 and since agents k and l have similar estimates during the initial stages of adaptation, we approximate $\bar{h}_{l,i}$ to

$$\begin{aligned} \bar{h}_{l,i} &= R_u(z_l^\circ - w_{k,i-1} + w_{k,i-1} - w_{l,i-1}) \\ &\approx R_u(z_l^\circ - w_{k,i-1}) \end{aligned} \tag{6.164}$$

That is, the vectors $\bar{h}_{k,i}$ and $\bar{h}_{l,i}$ can again be modeled as starting at the same location $w_{k,i-1}$, but pointing towards different directions: $\bar{h}_{k,i}$ towards z_k and $\bar{h}_{l,i}$ towards z_l. Let $\theta_{k,l,i}$ denote the angle between the vectors $\bar{h}_{k,i}$ and $\bar{h}_{l,i}$. Then, the angle between $\hat{\boldsymbol{h}}_{k,i}$ and $\hat{\boldsymbol{h}}_{l,i}$ is greater than $\theta_{k,l,i} - |\boldsymbol{\theta}_{k,i}| - |\boldsymbol{\theta}_{l,i}|$ (see Fig. 6.13(b)). Thus, following the arguments that led to (6.160) and using (6.161), the proba-

bility P_f is bounded by

$$\begin{aligned}
P_f &\leq \Pr\left(\theta_{k,l,i} - |\boldsymbol{\theta}_{k,i}| - |\boldsymbol{\theta}_{l,i}| < \frac{\pi}{2} \mid z_k^\circ \neq z_l^\circ\right) \\
&= \Pr\left(|\boldsymbol{\theta}_{k,i}| + |\boldsymbol{\theta}_{l,i}| > \theta_{k,l,i} - \frac{\pi}{2} \mid z_k^\circ \neq z_l^\circ\right) \\
&\leq \Pr\left(\frac{\|\boldsymbol{n}_{k,i}\|}{\|\bar{h}_{k,i}\|} + \frac{\|\boldsymbol{n}_{l,i}\|}{\|\bar{h}_{l,i}\|} > \theta_{k,l,i} - \frac{\pi}{2}\right) \\
&\leq \Pr\left(\|\boldsymbol{n}_{k,i}\| > \frac{\|\bar{h}_{k,i}\|}{2}\left(\theta_{k,l,i} - \frac{\pi}{2}\right)\right) \\
&\quad + \Pr\left(\|\boldsymbol{n}_{l,i}\| > \frac{\|\bar{h}_{l,i}\|}{2}\left(\theta_{k,l,i} - \frac{\pi}{2}\right)\right) \\
&\leq \frac{4\mathbb{E}\|\boldsymbol{n}_{k,i}\|^2}{\|\bar{h}_{k,i}\|^2(\theta_{k,l,i} - \pi/2)^2} + \frac{4\mathbb{E}\|\boldsymbol{n}_{l,i}\|^2}{\|\bar{h}_{l,i}\|^2(\theta_{k,l,i} - \pi/2)^2} \\
&\leq \frac{4\nu\tau}{(\theta_{k,l,i} - \pi/2)^2}
\end{aligned} \tag{6.165}$$

We observe that the probability in (6.96) depends on the angle $\theta_{k,l,i}$. As we mentioned in Section 6.4, it is possible for agent k to increase $\boldsymbol{b}_{k,i}(l)$ even when $z_k^\circ \neq z_l^\circ$. Nevertheless, as time evolves, the estimates of agents k and l move close to a linear combination of the underlying models and from Lemma 6.2 it is reasonable to assume that

$$w_{k,i-1} \approx \sum_{m=1}^{N} c_m z_m^\circ \tag{6.166}$$

for all k. Since $z_m^\circ \in \{w_0^\circ, w_1^\circ\}$ and $z_k^\circ \neq z_l^\circ$, we can rewrite (6.166) as:

$$w_{k,i-1} \approx b z_k^\circ + (1-b) z_l^\circ \tag{6.167}$$

for all k with the scalar $b \in (0,1)$ in view of (6.3)-(6.4). Then, the angle between $z_k^\circ - w_{k,i-1}$ and $z_l^\circ - w_{l,i-1}$ assumes a value close to π because

$$\begin{aligned}
z_k^\circ - w_{k,i-1} &\approx (1-b)(z_k^\circ - z_l^\circ) \\
&\approx -\frac{1-b}{b}(z_l^\circ - w_{l,i-1})
\end{aligned} \tag{6.168}$$

This implies that

$$\begin{aligned}\bar{h}_{k,i} &= R_u(z_k^\circ - w_{k,i-1}) \\ &\approx -\frac{1-b}{b} R_u(z_l^\circ - w_{l,i-1}) \\ &= -\frac{1-b}{b}\bar{h}_{l,i}\end{aligned} \tag{6.169}$$

and that the angle $\theta_{k,l,i}$ is close to π. We then obtain (6.96).

CHAPTER 7

Mobile Adaptive Networks

In this chapter, we add another dimension of complexity to the diffusion strategies [31, 34, 35, 88, 95, 124, 141] and incorporate agent mobility into the design of adaptive networks. Our objective is to develop and study what we refer to as *mobile adaptive* networks. These are networks that possess distributed adaptation abilities in addition to collective patterns of motion. The motion of the agents is influenced by the quality of the estimation process and vice versa, such that the two issues of adaptation and mobility become intertwined. For example, mobility enables agents to move to locations that improve the quality of their local measurements (e.g., by moving to locations with better signal-to-noise ratio (SNR) conditions). Mobility also enables agents to move towards the desired location when the network is working on tracking a target. We incorporate control mechanisms that enable the agents to move in a coordinated manner while at the same time solving the estimation or tracking problem of interest. The control mechanisms are implemented in a fully distributed manner, just like the diffusion mechanism that we use for the processing of information at the agents. In this way, our work extends previous studies on the motion of coordinated agents [56, 68, 106, 107, 118] by combining motion coordination with adaptation, learning and tracking in real-time. One example that we shall use to illustrate the application of mobile adaptive networks is the foraging behavior of fish schools [127, 145–147, 149, 150, 156]. Fish form schools and move together in

remarkable harmony. Using mobile adaptive networks, we will be able to emulate how the fish move together while at the same time tracking the location of a food source.

7.1 Measurement Model

For presentation purposes, the vector w° will denote in the following sections the location of a target (relative to some global coordinate system) that the network wishes to estimate and track (such as the location of a food source or a predator). However, the resulting algorithm is more widely applicable and w° could be used to represent other parameters of interest in more general optimization problems.

To begin with, we first explain how data $\{d_k(i), u_{k,i}\}$ satisfying model (2.3) can be motivated in the context of mobile networks so that the network can employ the ATC diffusion strategy (2.29) to estimate w°. As Fig. 7.1 shows, the distance between a target located at w° and a agent k located at $x_{k,i}$ at time i can be expressed as the inner product

$$d_k^\circ(i) = u_{k,i}^\circ(w^\circ - x_{k,i}) \tag{7.1}$$

where $u_{k,i}^\circ$ denotes the unit direction vector pointing to w° from $x_{k,i}$; this vector is defined in terms of the azimuth angle, $\theta_k(i)$, and the elevation angle, $\varphi_k(i)$, i.e.,

$$u_{k,i}^\circ = \Big[\cos\theta_k(i)\cos\varphi_k(i) \quad \sin\theta_k(i)\cos\varphi_k(i) \quad \sin\varphi_k(i)\Big] \tag{7.2}$$

In $\mathbb{R}^2$, we would only need to consider the azimuth angle in which case $u_{k,i}^\circ$ becomes

$$u_{k,i}^\circ = \Big[\cos\theta_k(i) \quad \sin\theta_k(i)\Big] \tag{7.3}$$

Expressing the distance in the inner product form (7.1), instead of using a quadratic Euclidean distance formula, allows us to relate the data $\{d_k^\circ(i), u_{k,i}^\circ\}$

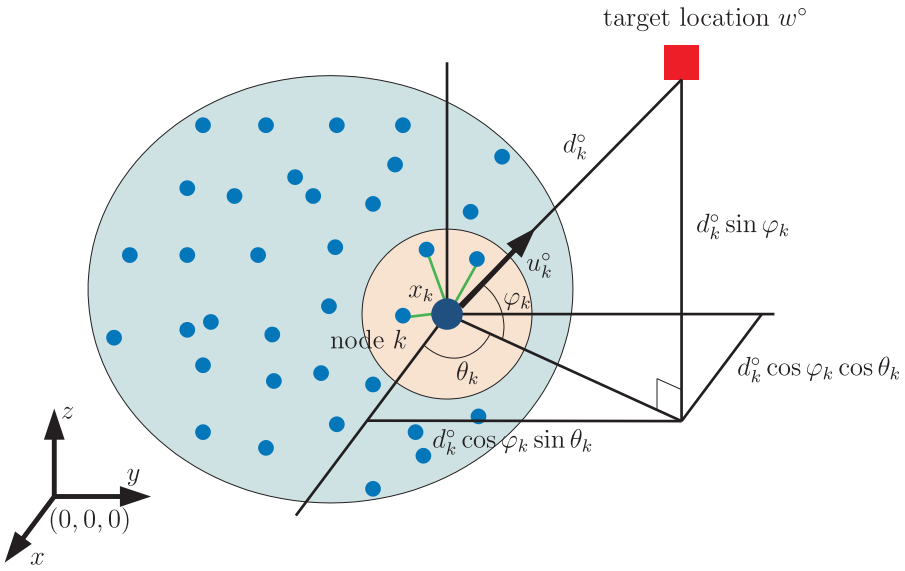


Figure 7.1: Distance and direction of the target w° from agent k at location x_k. The unit direction vector u_k° points towards w°.

in the desired linear model form (2.3). Note from the definitions of (7.1) and (7.2) in this example that $\|u_{k,i}^\circ\| = 1$ and $d_k^\circ(i) \geq 0$. These conditions on the data are *not* necessary for the operation of diffusion adaptation and mobile adaptive networks; they are simply properties that result when the variables $u_{k,i}^\circ$ and $d_k^\circ(i)$ are taken to represent the unit direction vector and the distance to the target in the current example. In more general scenarios, where $d_k(i)$ need not correspond to distances, we simply expect the data to satisfy model (2.3).

Continuing with (7.1)-(7.2), the superscript $\circ$ is used to indicate true values. However, agents observe noisy measurements of the direction $u_{k,i}^\circ$ and the distance $d_k^\circ(i)$ to the target, say,

$$u_{k,i} = u_{k,i}^\circ + v_{k,i}^u \quad \text{and} \quad d_k(i) = d_k^\circ(i) + v_k^d(i) \tag{7.4}$$

where $v_{k,i}^u$ and $v_k^d(i)$ denote additive noise terms of sizes M and 1, respectively.

Introduce

$$\begin{aligned}\hat{d}_k(i) &\triangleq d_k(i) + u_{k,i}x_{k,i} \\ &= u_{k,i}w^\circ + v_k(i)\end{aligned} \tag{7.5}$$

where the scalar noise term $v_k(i)$ is given by

$$v_k(i) \triangleq -v_{k,i}^u(w^\circ - x_{k,i}) + v_k^d(i) \tag{7.6}$$

Relation (7.5) represents a linear model of the same form (2.3) between the variables $\{\hat{d}_k(i), u_{k,i}\}$; having such a linear relation facilitates the application of the results of [34, 95, 124] to the adaptive diffusion algorithms of this chapter. In addition, the noisy location of the target is denoted by $p_{k,i}$ and is related to the variables $\{d_k(i), u_{k,i}, x_{k,i}\}$ as follows:

$$\begin{aligned}p_{k,i} &= x_{k,i} + d_k(i)u_{k,i}^T \\ &= w^\circ + \eta_{k,i}\end{aligned} \tag{7.7}$$

where the vector noise term is given by

$$\eta_{k,i} = v_k^d(i)u_{k,i}^{\circ T} + d_k^\circ(i)v_{k,i}^{uT} + v_k^d(i)v_{k,i}^{uT} \tag{7.8}$$

In the sequel, we assume that $\eta_{k,i}$ is a zero mean white random process with covariance matrix $R_{\eta,k,i}$ and let $\sigma_k^2(i) = \mathrm{Tr}(R_{\eta,k,i})$ denote the trace of $R_{\eta,k,i}$.

7.2 Motion Control Mechanism

In a mobile network, every agent k will update its location vector from $x_{k,i}$ to $x_{k,i+1}$ according to the rule:

$$x_{k,i+1} = x_{k,i} + \triangle t \cdot q_{k,i+1} \tag{7.9}$$

where $\triangle t$ represents the time step and $q_{k,i+1}$ is the velocity vector of the agent. Several factors influence the determination of the velocity vector $q_{k,i+1}$ of agent

k such as the desire to move to locations with better SNR conditions, the desire to move towards the target w°, the desire to move in coordination with the other agents, and the desire to avoid collisions. We now explain the mechanism by which $q_{k,i+1}$ can be set by agent k to achieve these goals.

To begin with, agents would like to move towards regions that have better SNR conditions to improve their estimation accuracy of w°. Suppose that at time i, the agents are able to assess (or estimate) the noise variances, σ_k^2, at their locations (we are omitting the time index for simplicity and writing σ_k^2 instead of $\sigma_k^2(i)$). To improve performance, the network may consider moving in a direction that reduces $\sum_{k=1}^N \sigma_k^2$. One way to achieve this objective is for each agent to set its motion direction towards the neighbors with lowest noise variance by setting the velocity vector as follows (we are omitting the time subscript $i+1$ from v for simplicity):

$$q_k = -\sum_{l \in \mathcal{N}_k \backslash \{k\}} (\sigma_l^2 - \sigma_k^2) \frac{x_l - x_k}{\|x_l - x_k\|} \tag{7.10}$$

For example, if the noise variance at agent l is larger than the noise variance at agent k, then (7.10) indicates that the velocity vector q_k will tend to point in the opposite direction moving away from x_l; the same observation holds for the other terms appearing in (7.10).

The agents in the network would also like to move towards the target w°. This objective will help ensure that the center of mass of the network will approach w° as time progresses. The location of the center of mass at time i is defined as the average location of all agents:

$$x_i^g \triangleq \frac{1}{N} \sum_{k=1}^N x_{k,i} \tag{7.11}$$

It is sufficient for our purposes for the convergence of x_i^g towards w° to occur in some mean-square error sense (as we discuss in Appendix 7.B). One way by

which agents can move towards w° is by having each agent set its velocity vector to point along the direction $w^\circ - x_k$, say,

$$q_k = h(w^\circ - x_k) = \begin{cases} w^\circ - x_k, & \text{if } \|w^\circ - x_k\| \leq s \\ s \cdot \frac{w^\circ - x_k}{\|w^\circ - x_k\|}, & \text{otherwise} \end{cases} \tag{7.12}$$

for some positive scaling factor s used to bound the speed in pursuing the target. Naturally, agents cannot use (7.12) directly to adjust their velocity vectors because they do not know w°. Instead, agents will replace w° in (7.12) by local estimates $w_{k,i}$ computed via a diffusion adaptation step, as shown further ahead in (7.18). Replacing w° in (7.12) by local estimates $w_{k,i}$ adds a new layer of complexity to the operation of the network in comparison, for example, to flocking models that assume beforehand that w° is known to the agents [107]. The use of $w_{k,i}$ adds estimation errors, in addition to measurement noise, into the operation of each agent and into the operation of the network. These errors influence the behavior of the agents. This is because velocities determine future locations, which in turn determine the quality of the SNR conditions at these locations and how well the network moves together towards the desired target w°. Studying how the network is able to adjust in the presence of estimation errors and noisy data, and how agents are able to process the information adaptively to achieve their desired objective, is one of the objectives of this chapter.

The agents in the network do not only want to move in the direction of w° or away from noisy regions, they also want their movements to be done in an organized manner. The agents would like to move in coherence and would like to avoid collisions while maintaining a certain safe distance r from their neighbors. Thus, consider the following cost function at agent k at time i:

$$J_k(q_k) = \sum_{l=1}^{N} \|q_k - q_l\|^2 + \sum_{l \in \mathcal{N}_k \backslash \{k\}} \left[\|(x_k + \triangle t \cdot q_k) - (x_l + \triangle t \cdot q_l)\| - r\right]^2 \tag{7.13}$$

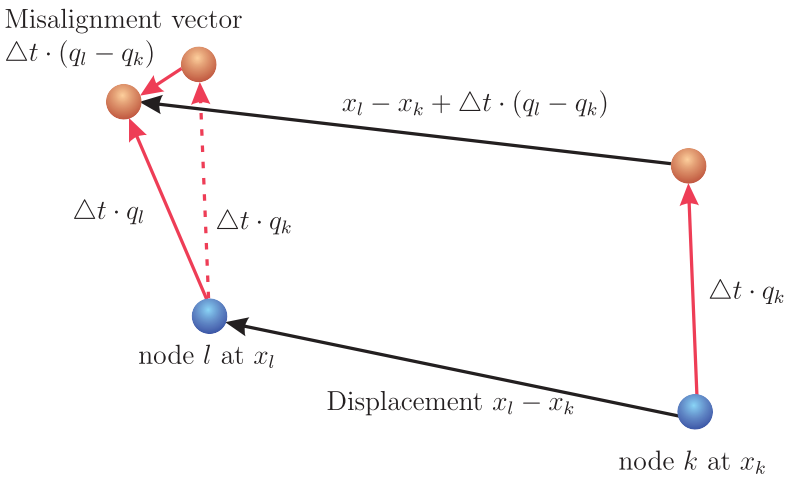


Figure 7.2: Location of agent l relative to agent k before and after the displacement update.

where the term $x_k + \triangle t \cdot q_k$ represents the updated location of agent k and the terms $\{x_l + \triangle t \cdot q_l\}$ represent the updated locations of its neighbors. The minimization of (7.13) over v_k is meant to ensure that the difference between the velocities is minimized and the distance between the updated locations stays close to r. To determine the optimal v_k, we differentiate (7.13) with respect to v_k and get

$$\frac{dJ_k(q_k)}{dq_k} = 2\sum_{l=1}^{N}(q_k - q_l) + 2\sum_{l\in\mathcal{N}_k\setminus\{k\}}\left[\triangle t(q_k - q_l) - (x_l - x_k) + r\frac{x_l - x_k + \triangle t\cdot(q_l - q_k)}{\|x_l - x_k + \triangle t\cdot(q_l - q_k)\|}\right] \tag{7.14}$$

Note that $x_l - x_k$ denotes the location of agent l relative to agent k. To determine q_k from (7.14) we first investigate the last term in (7.14). Figure 7.2 depicts the current locations of agents k and l and their updated locations. The term $\triangle t \cdot (q_l - q_k)$ is a measure of how misaligned the displacement of agents l and k are after the update relative to their displacement before the update. It is

reasonable to assume that this misalignment is sufficiently small relative to the distance $\|x_l - x_k\|$ because, in general, the velocity of agent k is close to the velocities of its neighbors (or the time step $\triangle t$ is small enough). Thus, we may introduce the approximation:

$$\frac{x_l - x_k + \triangle t(q_l - q_k)}{\|x_l - x_k + \triangle t(q_l - q_k)\|} \approx \frac{x_l - x_k}{\|x_l - x_k\|} \tag{7.15}$$

The result is a normalized direction vector (a unit vector) along the direction connecting agents l and k. Moreover, since the time step is small, the term $\triangle t \cdot (q_k - q_l)$, compared to the first term in (7.14), can be ignored. By setting the derivative in (7.14) to zero, the velocity vector is found to be

$$q_k = \frac{1}{N}\sum_{l=1}^{N} q_l + \frac{1}{N}\sum_{l \in \mathcal{N}_k \setminus \{k\}} \left(1 - \frac{r}{\|x_l - x_k\|}\right)(x_l - x_k) \tag{7.16}$$

Expression (7.16) consists of two terms. The first term represents the velocity of the center of mass of the network, q^g, i.e.,

$$q^g = \frac{1}{N}\sum_{l=1}^{N} q_l \tag{7.17}$$

This term suggests that, in addition to moving towards w°, as required by (7.12), agent k should also attempt to adjust its velocity vector to be consistent with the average velocity of the network. Doing so results in a pattern of collective motion. Obviously agents cannot use this term directly to adjust their velocity vectors because they do not have access to all velocities $\{q_l\}$ across the network. Instead, the agents replace q^g by local estimates $q^g_{k,i}$ (see (7.18) ahead) defined later in (7.28). The second term in (7.16) is a linear combination of the displacement vectors $\{x_l - x_k\}$; the magnitudes of these vectors are scaled by subtracting from them unit vectors of size r. This term suggests that agents should adjust their velocities to be consistent with the average displacement vector in the neighborhood while maintaining a distance r from their neighbors. A structure similar

to (7.16) was suggested before to induce harmonic motion and repulsion and attraction behavior among agents in a network (see [70, 107]). Here, we motivated (7.16) by starting from the optimization problem (7.13) and by appealing to the geometric approximation of Fig. 7.2.

Motivated by the previous discussions, the final structure that we adopt for updating the velocity vector appears in (7.18) below is more general than (7.16) in two respects. First, expression (7.18) incorporates the term $w_{k,i} - x_{k,i}$, which relates to the ultimate objective of having the network move towards the unknown target w°. Second, expression (7.18) involves the term $g_{k,i}$ (defined in (7.19)), which relates to moving towards regions of lower noise level to improve estimation performance. These two terms couple the adaptation problem to mobility. Therefore, based on (7.10), (7.12), and (7.16), we assume that agents adjust their velocity vectors according to four criteria as follows:

$$q_{k,i+1} = \lambda \cdot h(w_{k,i} - x_{k,i}) + \alpha \frac{g_{k,i}}{\|g_{k,i}\|} + \beta q^g_{k,i} + \gamma \delta_{k,i} \tag{7.18}$$

where $\{\lambda, \alpha, \beta, \gamma\}$ are non-negative weighting factors and

$$g_{k,i} = -\sum_{l \in \mathcal{N}_k \backslash \{k\}} [\sigma_l^2(i) - \sigma_k^2(i)] \frac{x_{l,i} - x_{k,i}}{\|x_{l,i} - x_{k,i}\|} \tag{7.19}$$

$$\delta_{k,i} = \sum_{l \in \mathcal{N}_k \backslash \{k\}} (\|x_{l,i} - x_{k,i}\| - r) \frac{x_{l,i} - x_{k,i}}{\|x_{l,i} - x_{k,i}\|} \tag{7.20}$$

Note that to prevent singularity, we define $x/\|x\| \triangleq 0$ whenever $x = 0$. Moving forward, we assume that every agent in the network is adjusting its velocity vector according to the general rule (7.18). Special cases can be obtained by setting some of the scalars $\{\lambda, \alpha, \beta, \gamma\}$ to zero. For example, if the agents do not have access to information about the noise level, then the term involving $g_{k,i}$ in (7.18) can be dropped by setting $\alpha = 0$. Also, in M dimensions, the vector w° need not denote location but may denote some state of the system that the

agents are interested in estimating rather than moving towards it. The objective of the network becomes moving towards a location with the lowest noise level. In this case, the first term in (7.18) can be removed by setting $\lambda = 0$.

As mentioned earlier, one difference in (7.18) relative to previous flocking models is the reliance on local estimates $w_{k,i}$ and $q_{k,i}^g$. Due to local estimation errors and measurement noise, these estimates influence the dynamics of the network and its learning and tracking performance. Another difference is the use of diffusion adaptation schemes to estimate both $w_{k,i}$ and $q_{k,i}^g$ sequentially and in real-time. We move on to describe the diffusion mechanisms that allow the agents to obtain the local estimates $w_{k,i}$ and $q_{k,i}^g$ in a distributed manner.

7.3 Distributed Estimation of Global Variables

Since the ATC diffusion strategy (2.29) has been shown to perform the best among the other distributed strategies, we employ the ATC diffusion strategy to solve the estimation tasks.

7.3.1 Estimating Location of the Target

At every time instant i, every agent k has access to the local measurements $\{\hat{d}_k(i), u_{k,i}\}$ related by (7.5). Using these local data, as well as data shared with their neighbors, the agents would like to estimate in a distributed manner the global parameter w° that minimizes the following cost function (similar to (2.4)):

$$J^{\text{glob}}(w) = \sum_{k=1}^{N} E|\hat{\boldsymbol{d}}_k(i) - \boldsymbol{u}_{k,i}w|^2 \tag{7.21}$$

Applying the ATC diffusion strategy (2.29), we have

$$\begin{aligned} \psi_{k,i} &= w_{k,i-1} + \mu_k \cdot u_{k,i}^T[\hat{d}_k(i) - u_{k,i}w_{k,i-1}] \\ w_{k,i} &= \sum_{l\in\mathcal{N}_k} a_{l,k}\psi_{l,i} \end{aligned} \tag{7.22}$$

where the coefficients $\{a_{l,k}\}$ satisfy property (2.25).

Though unnecessary, since model (7.5) is geometry-bearing (see (7.1)), we can exploit this fact to simplify (7.22) under reasonable approximations. We use (7.5) and (7.7) to rewrite the update vector from (7.22) as:

$$u_{k,i}^T[\hat{d}_k(i) - u_{k,i}w_{k,i-1}] = (p_{k,i} - x_{k,i}) - u_{k,i}^T u_{k,i}(w_{k,i-1} - x_{k,i}) \tag{7.23}$$

The first term, $p_{k,i} - x_{k,i}$, represents the measurement of w° by agent k while the second term, $u_{k,i}^T u_{k,i}(w_{k,i-1} - x_{k,i})$, represents the projection of the estimate of w° by agent k onto the direction, $u_{k,i}$. That is, the second term only considers the difference along the measured direction by agent k. Now, if the direction of the vector $w_{k,i-1} - x_{k,i}$ is close to $u_{k,i}$, we can use the approximation:

$$w_{k,i-1} - x_{k,i} \approx \|w_{k,i-1} - x_{k,i}\| u_{k,i}^T \tag{7.24}$$

That is, we assume $w_{k,i-1} - x_{k,i}$ aligns with $u_{k,i}$. Under this approximation, the update vector becomes

$$u_{k,i}^T[\hat{d}_k(i) - u_{k,i}w_{k,i-1}] \approx p_{k,i} - w_{k,i-1} \tag{7.25}$$

i.e., the difference between of the measurement of w° and the estimate of w° by agent k, and we rewrite (7.22) as

$$\begin{aligned} \psi_{k,i} &= w_{k,i-1} + \mu_k \cdot (q_{k,i} - w_{k,i-1}) \\ w_{k,i} &= \sum_{l\in\mathcal{N}_k} a_{l,k}\psi_{l,i} \end{aligned} \tag{7.26}$$

We will use this form in the algorithm and performance analysis.

7.3.2 Estimating Velocity of the Center of the Network

The velocity of the center of gravity, q^g, should be also estimated in a distributed way to enable (7.18). By definition, q^g is the average velocity of all agents in the network. However, since the velocities of the agents are changing and so is q^g, we need to keep track of q^g over time. Motivated by diffusion techniques, we introduce the cost function

$$J^{\text{glob}}(q^g) = \sum_{k=1}^{N} \|q_{k,i} - q^g\|^2 \tag{7.27}$$

Using the similar diffusion structure (2.29) and the arguments in [34], we can derive the following diffusion algorithm for computing $q^g_{k,i}$:

$$\begin{aligned} \phi_{k,i} &= q^g_{k,i-1} + \nu_k \sum_{l \in \mathcal{N}_k} c_{l,k}(q_{l,i} - q^g_{k,i-1}) \\ q^g_{k,i} &= \sum_{l \in \mathcal{N}_k} a^q_{l,k} \phi_{l,i} \end{aligned} \tag{7.28}$$

where $\{c_{l,k}\}$ and $\{a^q_{l,k}\}$ are two sets of non-negative real coefficients satisfying (2.25). Note that we are allowing agent k to use the velocity vectors $\{q_{l,i}\}$ in its neighborhood because it is able to observe these vectors. We see that the structure of (7.28) is the same as the structure in (7.26) and that the update vector, $q_{l,i} - q^g_{k,i-1}$, contributed by agent l is simply the difference of the velocity of agent l and the estimate of agent k. Note that by selecting $\nu_k = 1$, $a^q_{l,k} = \delta_{lk}$ in terms of the Kronecker delta function, and $c_{l,k} = 1/n_k$ if $l \in \mathcal{N}_k$, the algorithm (7.28) reduces to determining $q^g_{k,i}$ by simply averaging the velocities of the neighbors of agent k, which is usually applied in previous studies [70, 107, 161]. As we will show by simulation (see Appendix 7.B), the diffusion algorithm achieves better coherent motion.

We therefore end up with two diffusion mechanisms. Equation (7.26) describes the diffusion mechanism for estimating the unknown parameter, w°, while equa-

tion (7.28) describes the diffusion mechanism for tracking the velocity of the center of mass of the network.

7.4 Diffusion Strategy with Self-Organization

Combining the diffusion steps (7.26) and (7.28) with the velocity adjustment (7.18), we arrive at the following ATC diffusion algorithm to coordinate the adaptation and motion activities of a mobile adaptive network.

Note that the neighborhood of agent k is now denoted by $\mathcal{N}_{k,i}$ to emphasize that the network topology may change with time due to movement. The block diagrams of the above algorithm is illustrated in Fig. 7.3. The analysis of the algorithm is provided in Appendix 7.B. We also survey the previous works [56, 68, 100, 107, 109] on multi-agent formation control in 7.A.

7.4.1 Simulation Results

7.4.1.1 Mobile Adaptive Networks with Single Target

We simulate the motion of mobile networks with 50 agents in pursuing a target. The simulation parameters are set as follows. The unit length is the body length of an agent (e.g., body length of a fish). The step sizes for diffusion strategies (7.26) and (7.28) are set to $\mu_k = \nu_k = 0.5$ for all k. The combination weights are set to $a_{l,k} = a^q_{l,k} = 1/n_k$ if $l \in \mathcal{N}_k$ and $c_{l,k} = \delta_{lk}$. That is, the agents in the network only combine intermediate estimates using the uniform combination rule. The measurement $p_{k,i}$ is generated from the model (7.7) and noise covariance matrix $R_{\eta,k,i}$ is assumed to satisfy a model of the form

$$R_{\eta,k,i} = \kappa \cdot \|w^\circ - \boldsymbol{x}_{k,i}\|^2 I_M \tag{7.29}$$

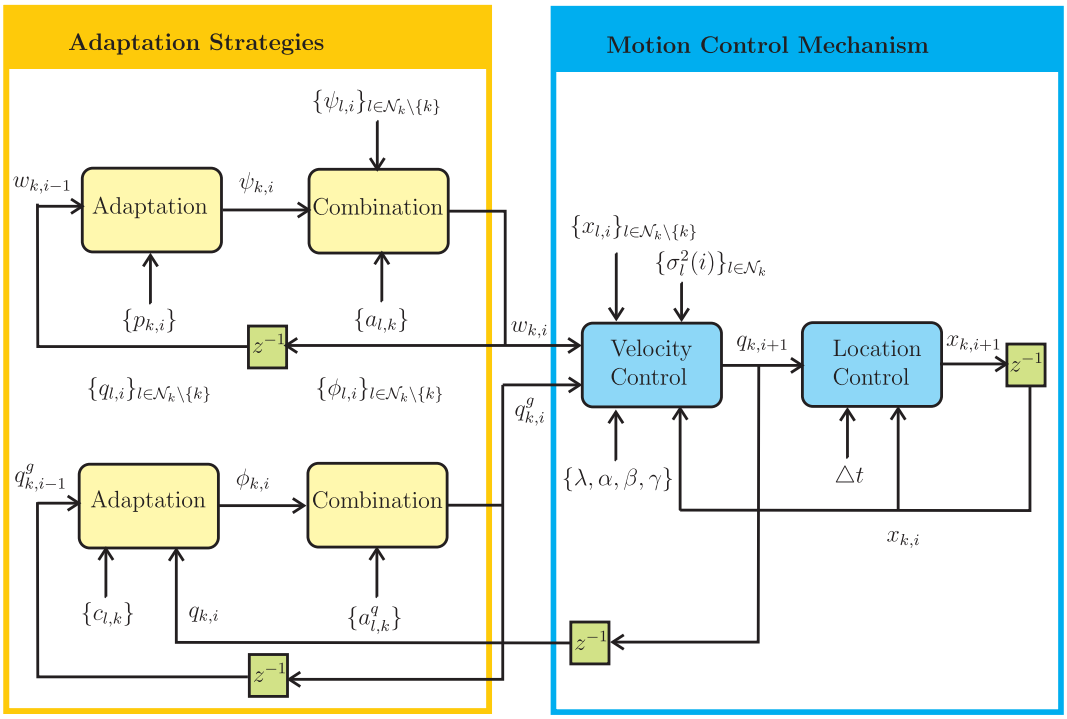


Figure 7.3: ATC diffusion strategy for mobile adaptive networks. The implementation consists of two diffusion mechanisms: one for estimating the target w° and another for tracking the velocity of the center of mass of the network q^g. In addition, a control mechanism controls the motion of the agents (their velocities and locations).

Algorithm 7.1 ATC Diffusion Strategy for Mobile Networks

For each node k, initialize $w_{k,-1} = 0$ and $q^g_{k,-1} = 0$.

for $i \geq 0$ and $k = 1$ to N **do**

1) The agent knows the location and velocity vectors $\{x_{l,i}, q_{l,i}\}$ for $l \in \mathcal{N}_{k,i}$ and has access to the local data $\{d_k(i), u_{k,i}, \sigma^2_k(i)\}$.

2) Find $p_{k,i} = x_{k,i} + d_k(i)u^T_{k,i}$.

3) Perform two local adaptation steps, one for the weight vector, w°, and the other for the velocity of the center of mass, q^g:

$$\psi_{k,i} = w_{k,i-1} + \mu_k \cdot (q_{k,i} - w_{k,i-1})$$
$$\phi_{k,i} = q^g_{k,i-1} + \nu_k \cdot \sum_{l \in \mathcal{N}_{k,i}} c_{l,k}(q_{l,i} - q^g_{k,i-1})$$

4) Perform two local combination steps: one combines weight estimates for w° and the other combines velocity estimates for q^g:

$$w_{k,i} = \sum_{l \in \mathcal{N}_{k,i}} a_{l,k}\psi_{l,i}$$
$$q^g_{k,i} = \sum_{l \in \mathcal{N}_{k,i}} a^q_{l,k}\phi_{l,i}$$

5) Update the agent velocity and its location:

$$q_{k,i+1} = \lambda \cdot h(w_{k,i} - x_{k,i}) + \alpha \frac{g_{k,i}}{\|g_{k,i}\|} + \beta q^g_{k,i} + \gamma \delta_{k,i}$$
$$x_{k,i+1} = x_{k,i} + \triangle t \cdot q_{k,i+1}$$

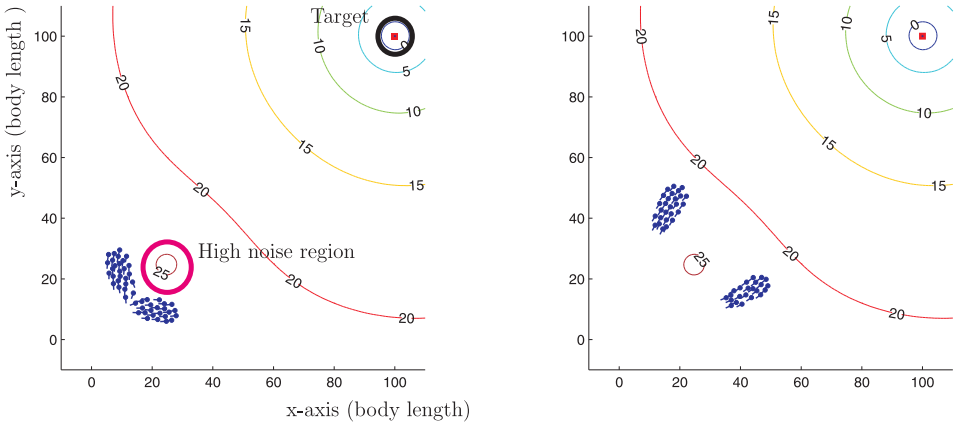
where $g_{k,i}$ and $\delta_{k,i}$ are defined by (7.19)-(7.20) and $\{\lambda, \alpha, \beta, \gamma\}$ are non-negative parameters.

end for

for some small constant κ. We set $\kappa = 0.01$. The model is reasonable since the signal power usually decreases proportionally to the square of the propagation distance, which increases the noise variance accordingly. For motion control mechanism in (7.9) and (7.18), the parameters are $(\lambda, \alpha, \beta) = (0.5, 0.5, 0.5)$ and $\gamma_k = 1/(n_k - 1)$ and the time duration and speed bound are $\triangle t = 0.1$ sec and $s = 1$, respectively. In addition, the optimal distance between two neighbors is $r = 3$. Due to computational and communication limitation, the number of neighbors will be constrained, say to N_B. An agent chooses $N_B = 6$ nearest neighbors from agents within the radius $R = 5$.

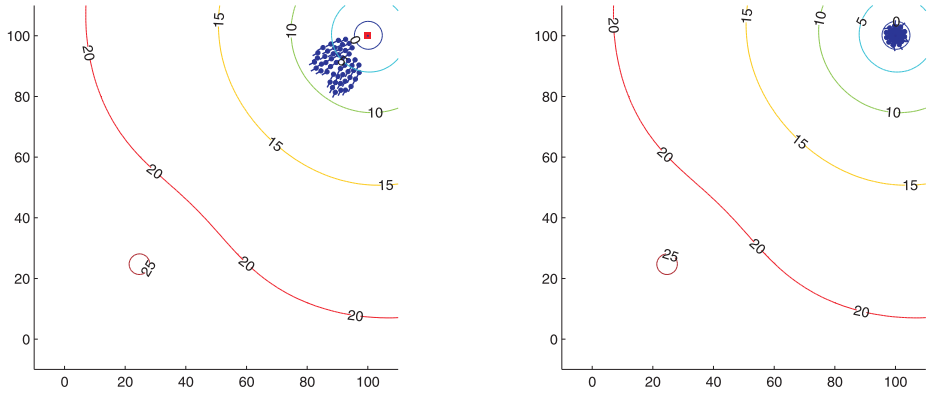
Figure 7.4 illustrates the maneuver of a mobile network in $\mathbb{R}^2$ over time. The symbol, "■", denotes the target of interest. In addition, "•" and "–" indicate the locations and moving directions of the agents, respectively. We also place a region with high noise variance along the way to the target. The noise variance distribution is illustrated in Fig. 7.5 over the plane. As Fig. 7.4 shows, the network exhibits harmonious movement. When the network approaches the region with high noise, the agents bypass the region to get better signal reception. Note that it is possible for the network to spread out and then regroup to continue their schooling. This depends on the location and the area of the high noise region. Finally, the network successfully gets to the target. In addition, due to the effects of attraction and repulsion, the agents maintain a rather uniform distance from their neighbors.

Now let w° be a general unknown vector. The motion of the agents is controlled by (7.18) with $\alpha = 0$ and $\{\lambda, \beta, \gamma\}$ set the same as before. Assume that there is a base station located at $(25, 25)$. Fig. 7.6 shows initial and final locations of the agents and we compare the network MSD of a mobile network with that of a static network in Fig. 7.7. It is clear that the mobile network successfully



(a) t=15 sec

(b) t=30 sec



(c) t=75 sec

(d) t=90 sec

Figure 7.4: Maneuvers of mobile networks in $\mathbb{R}^2$ over time: (a) $t = 15$ sec, (b) $t = 30$ sec, (c) $t = 75$ sec, and (d) $t = 90$ sec. The lines are the contour curves of noise variance in dB.

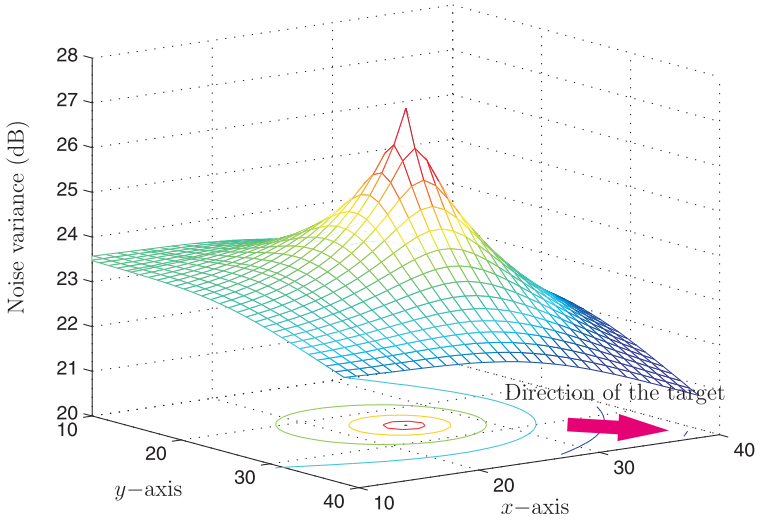
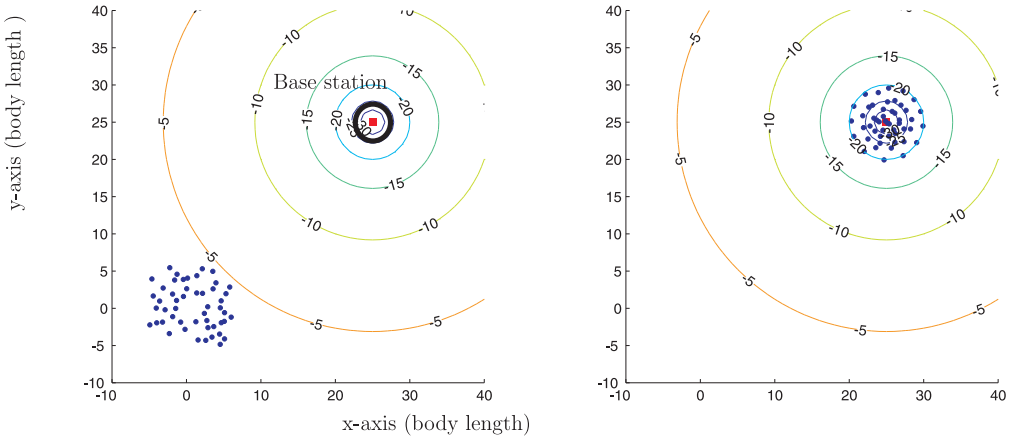


Figure 7.5: Noise variance over the plane. There is a spike at $(25, 25)$



(a) t=0 sec

(b) t=100 sec

Figure 7.6: Locations of mobile agents in $\mathbb{R}^2$ at (a) $t = 0$ sec, (b) $t = 100$ sec. The lines are the contour curves of noise variance in dB.

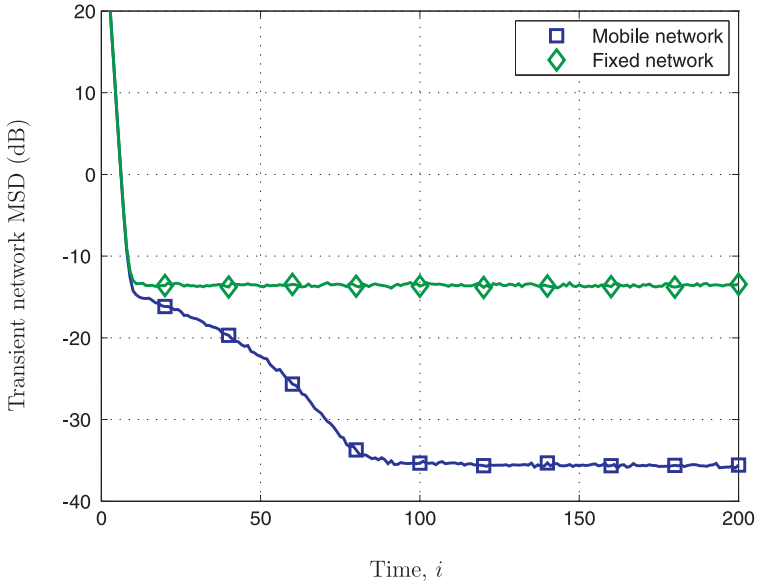


Figure 7.7: Transient network MSD for estimating a weight vector w°.

finds a region with good signal reception and achieves lower MSD.

7.4.1.2 Mobile Adaptive Networks with Two Targets

We consider the case when there are two possible targets located at $w_0^{\circ} = [40, -40]$ and $w_1^{\circ} = [40, 40]$. We illustrate the motion of the agents in Fig. 7.8, where agents that would like to move towards w_0° are shown in blue dot and agents that would like to move towards w_1° are shown in red circle. Initially, there are 100 agents uniformly distributed in a 20×20 square area around the origin; 50 of them collect data that originate from target w_0° and the other 50 agents collect data arising from the other target w_1°. The agents would like to move towards a common target by applying the diffusion strategy with decision-making in Algorithm 6.1. The step sizes for diffusion strategies (7.26) and (7.28) are set to $\mu_k = \nu_k = 0.05$ for all k. The combination weighs $\{a_{l,k}, a_{l,k}^q\}$ are set according to (6.131)-(6.132)

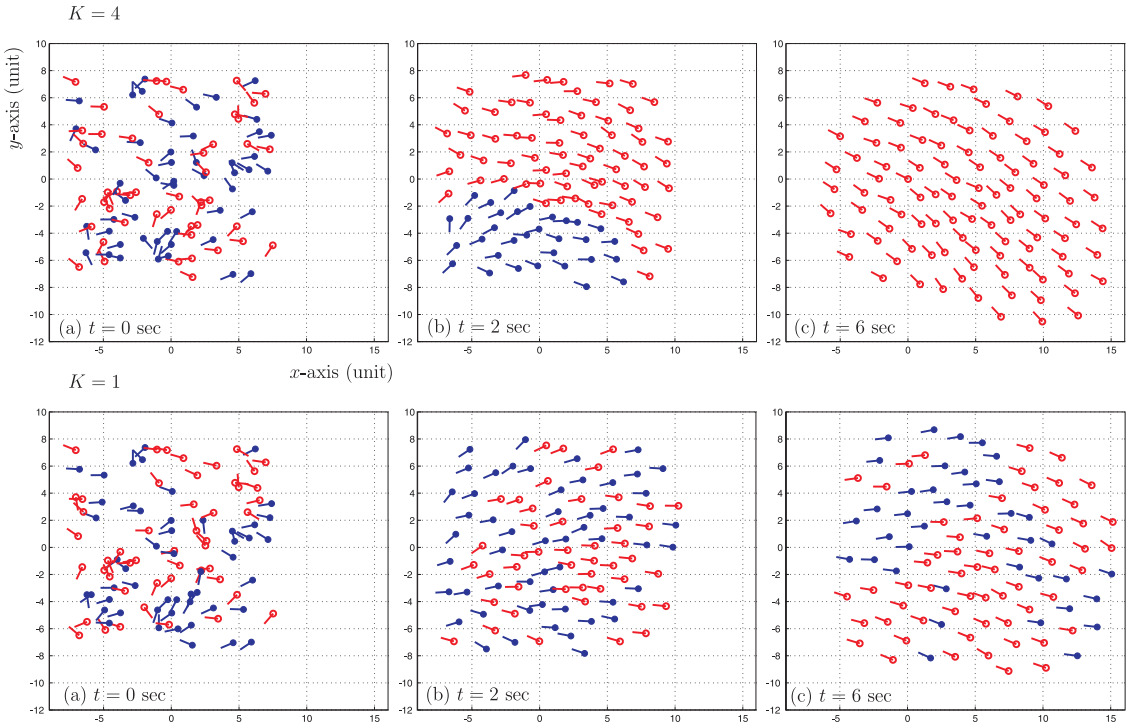


Figure 7.8: Maneuver of fish schools with two food sources over time (a) $t = 0$ (b) $t = 2$ (c) $t = 6$ sec. The networks employ decision-making with $K = 4$ and $K = 1$ in (6.34).

and $c_{l,k} = \delta_{lk}$. The parameters in the motion control mechanism (7.9) and (7.18) are set to $(\lambda, \alpha, \beta) = (0.3, 0, 0.7)$ with the other parameter fixed. We compare the decision-making process in (6.32)-(6.34) using two different parameters: $K = 4$ and $K = 1$. We observe that the network with higher value of K achieves agreement on a desired target at faster rate.

7.5 Adaptive Mobile Network with Predators

In this section, we introduce predators into mobile adaptive networks. It is observed in nature that fish schools spread out to escape from predators and regroup to continue with their schooling. There are several biological hypotheses to explain how fish take advantage of schooling to avoid the presence of their predators. Fish schooling confuses predators [99]; leading to a phenomenon known as the dilution effect [60] [158]. Another advantage of schooling is the many-eyes effect [121]. Fish within a school collaboratively detect predators such that the probability of detection increases appreciably. The school as a group can react and take action earlier than what would be possible by a single fish. In this section, we apply the ATC diffusion strategy (2.29) to explain how fish cooperatively pursue a food source while at the same time avoiding attack from predators [147, 149].

7.5.1 Motion Control Mechanism for Mobile Agents

We apply the same measurement model (7.5) in 2-dimensions. In the application we are studying here, the agents wish to track two separate targets: the location of a food source and the location of a predator. To distinguish between them, we shall write w^f for the actual location of the food source and x^p for the actual location of the predator. In addition, variables without superscript p will denote quantities that are related to the agents of the adaptive network. There are four factors influence the velocity vector of agent k such as (a) the desire to move away from a predator at location x^p, (b) the desire to move towards a food source at w^f, (c) the desire to move in coordination with the other agents, and (d) the desire to avoid collisions. We assume the predator is moving as well, so that its actual location should be denoted by x_i^p. The agents in the network are therefore

interested in estimating the fixed quantity w^f and in tracking the time-variant quantity x_i^p. Since the agents do not have access to the actual locations of the food and the predator, we will use $w_{k,i}^f$ and $w_{k,i}^p$ to denote the local estimates at agent k at time i via the ATC diffusion strategy (7.26). We now explain the mechanism by which the velocity vector $q_{k,i+1}$ can be set by agent k.

Before we proceed, we introduce two operators on 2×1 vectors. Let

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{7.30}$$

be a 2×1 vector. Then we define

$$\mathfrak{u}(q) \triangleq \frac{q}{\|q\|} \quad \text{and} \quad q^{\perp} \triangleq \begin{bmatrix} -q_2 \\ q_1 \end{bmatrix} \tag{7.31}$$

That is, $\mathfrak{u}(q)$ normalizes the vector and the notation $\perp$ finds a vector perpendicular to q.

7.5.1.1 Pursuing Food and Avoiding Predators

To begin with, the agents in the network would like to get to the food source and avoid the predator. The action of each agent depends on the location of the predator. Figure 7.9 shows three regions around the predator. There are two concentric circles with their centers origin at the predator and with radii r_1 and $2r_1$. The three regions represent the areas outside the circle of radius $2r_1$, inside the circle of radius r_1, and within the disc $r_1 < r < 2r_1$. If the predator is far away (i.e., if the distance from agent k to the predator, $d_k^p(i)$, is larger than $2r_1$, meaning $d_k^p(i) > 2r_1$), the fish stays in region I and focuses on exploring the food location. In this case, the velocity vector is set along the direction of the food, i.e.,

$$q_{k,i+1}^a = \mathfrak{u}(w_{k,i}^f - x_{k,i}) \quad \text{(region I)} \tag{7.32}$$

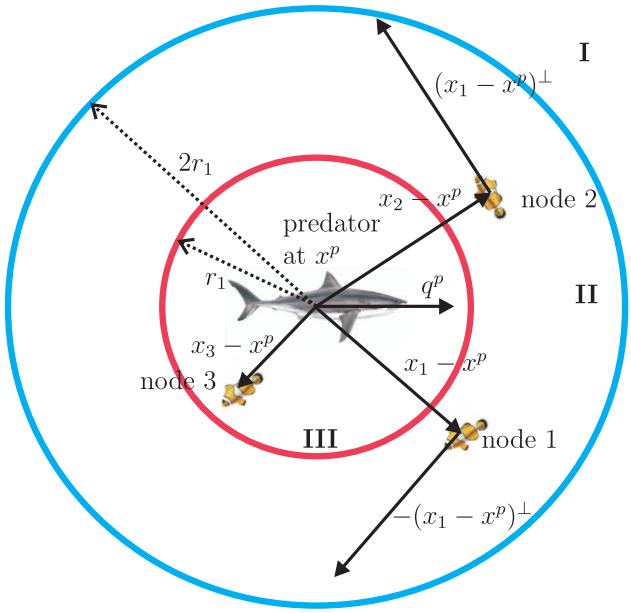


Figure 7.9: Four regions around the predator.

On the other hand, if the predator is close, (i.e., if $d_k^p(i) < r_1$ so that the agent is in region III), then the agent focuses on escaping the attack by the predator by moving away from it. In this case, the velocity vector is chosen as

$$q_{k,i+1}^a = \left(2r_1 - \|x_{k,i} - w_{k,i}^p\|\right) \mathfrak{u}(x_{k,i} - w_{k,i}^p) \quad \text{(region III)} \tag{7.33}$$

In (7.33), the speed of the agent depends on the distance to the predator. The agent will move faster if the predator is closer to it. The final situation we need to consider is when the predator is located at a distance between r_1 and $2r_1$ from the agent. The velocity vector in this case becomes

$$q_{k,i+1}^a = c_1 \cdot \mathfrak{u}[(x_{k,i} - w_{k,i}^p)^\perp] \quad \text{(region II)} \tag{7.34}$$

where c_1 is a scalar with

$$c_1 = \begin{cases} 1, & \text{if } (x_{k,i} - w_{k,i}^p)^T (w_{k,i}^q)^\perp \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{7.35}$$

That is, as shown in Fig. 7.9, the agent will move clockwise (when $c_1 = -1$) or counter-clockwise (when $c_1 = 1$) depending on the location and velocity of the predator, and if the agent is above the motion of the predator (e.g., agent 2 in Fig. 7.9), it moves counter-clockwise. Moreover, in (7.34), $w^q_{k,i}$ is the local estimate of the predator's velocity by agent k at time i, which can be estimated as

$$w^q_{k,i} = \frac{1}{\triangle t}(w^p_{k,i} - w^p_{k,i-1}) \tag{7.36}$$

For multiple predators, each agent in the fish school tracks the location of the nearest predator. This allows the agents to have the flexibility to adapt to their local situation.

7.5.1.2 Reunion

Following an attack by a predator, a network becomes fragmented with some agents lying at the outer edges of the new smaller groups that resulted from the fragmentation (see Fig. 7.10). To reunite, agents on the outer boundaries have to estimate the location of the other groups and move towards them. To do so, an agent first needs to detect whether it is on the edge of the fragmented groups. We consider three kinds of edges - frontal edge, left edge and right edge. The agent computes the number of its neighbors in each direction according to the coordinate of agent l with respect to agent k:

$$x^{(k)}_{l,i} = W^T(q_{k,i})(x_{l,i} - x_{k,i}) \tag{7.37}$$

where

$$W(q) = \begin{bmatrix} q_1/\|q\| & -q_2/\|q\| \\ q_2/\|q\| & q_1/\|q\| \end{bmatrix} \tag{7.38}$$

is an orthonormal matrix for a local coordinate system centered at a agent that is moving with velocity vector $q = (q_1, q_2)$. If the first coordinate of $x^{(k)}_{l,i}$ is greater

than zero, agent l lies in front of agent k. Similarly, if the second coordinate of $x_{l,i}^{(k)}$ is greater than zero, agent l lies to the left of agent k; otherwise, it lies to the right side of agent k. We say agent k belongs to the frontal edge of the fragmented groups if the number of neighbors in the front is less than one. Likewise for the left and right edges. Agents on the edge then search for other groups. For example, agents in the front edge will find the nearest frontal agent outside its neighborhood and move towards that agent. That is, agent k would perform the following operation:

$$\hat{l} = \arg\min_{l} \{\|x_{l,i}^{(k)}\| \mid l \in \mathcal{N}_k^F, l \notin \mathcal{N}_k\} \tag{7.39}$$

$$q_{k,i+1}^{b} = \begin{cases} 0, & \text{if } \hat{l} = \phi \\ \frac{x_{\hat{l},i} - x_{k,i}}{\|x_{\hat{l},i} - x_{k,i}\|}, & \text{otherwise} \end{cases} \tag{7.40}$$

where $\mathcal{N}_k^F$ is the set of indexes of agents that lie in the front of agent k and ϕ denotes the empty set. Agents in the left and right edges conduct the same procedure.

7.5.1.3 Coherent motion

The agents do not only want to approach to the food source and avoid the predator, they also want to move in harmony to confuse the predator and would like to avoid collisions by maintaining a safe distance r from their neighbors. According to expression (7.16), this can be achieved if the agent updates its velocity vector as follows:

$$q_{k,i+1}^{c} = q_{k,i}^{g} + \gamma \delta_{k,i} \tag{7.41}$$

where γ is a nonnegative scalar and the term $\delta_{k,i}$ is defined in (7.19). Expression (7.41) also incorporates the local estimate for the velocity of the center of mass of the network, $q_{k,i}^{g}$, at agent k, which can be implemented using (7.28).

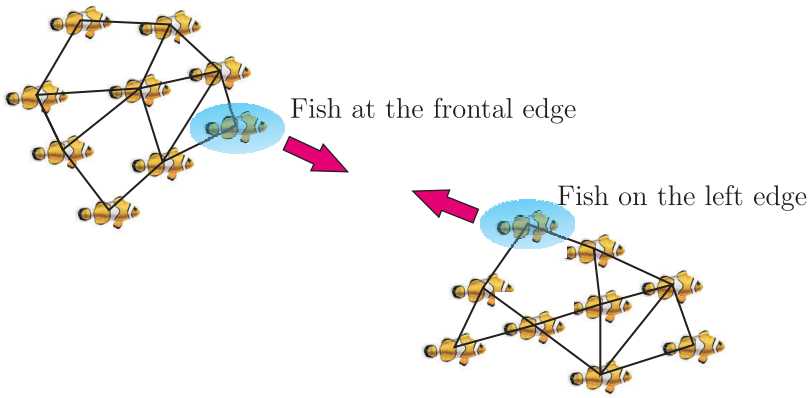


Figure 7.10: Two fragmental groups. Connections among the fish are indicated by lines. One fish at the frontal edge (left group) and one fish on the left edge (right group) are highlighted. They will move along the arrow directions to cause regrouping.

7.5.1.4 Control Mechanism

Based on the preceding criteria, we assume that agents adjust their velocity vectors as follows:

$$q_{k,i+1} = \lambda(\alpha_1 q^a_{k,i+1} + \alpha_2 q^b_{k,i+1}) + \beta q^g_{k,i} + \gamma \delta_{k,i} \tag{7.42}$$

where $\{\lambda, \alpha_1, \alpha_2, \beta, \gamma\}$ are non-negative weighting factors. In addition, we bound the maximum speed of agents by s. That is, the magnitude of $q_{k,i+1}$ will be scaled to s if it is larger than s. Compared to (7.18), the last two terms in (7.42) are the same as (7.18). However, expression (7.42) has more complicated control mechanism in the first term. When there are predators, the agents not only pursue the target but also escape from predators. They also have to reform the group after the fragmentation.

7.5.2 Motion Control Mechanism for Predators

Cooperative behavior can be observed among predators as well. For example, dolphins encircle their prey [14] and killer whales cooperatively herd herring into a tight ball close to the surface [131]. Cooperative hunting plays a role in increasing foraging efficiency. In this section, we use the ATC diffusion strategy to explain how predators cooperate with each other to surround a fish school and trap the school while attacking.

We first consider a single predator. The predator tracks the location of one agent at a time. We assume that the predator keeps tracking the nearest agent. At each time instant i, the predator measures the distance, $d(i)$, and direction, u_i, of the nearest agent. The predator then updates the location of the agent according to

$$w_i = w_{i-1} + \mu^p \cdot u_i^T [d(i) - u_i(w_{i-1} - x_i^p)] \tag{7.43}$$

where μ_p is a positive step-size for the predator. After estimating the location, the predator moves towards the agent. That is, the velocity vector of the predator are updated as:

$$q_{i+1}^p = s^p \cdot \mathfrak{u}(w_i - x_i^p) \tag{7.44}$$

for some positive scalar s^p.

Now, consider N^p predators that would like to hunt the fish agents in the network cooperatively. The location of predator l at time i is denoted by $x_{l,i}^p$. The motion of the predators is influenced by two quantities related to the network: the location of the center gravity of the network and the location of an agent of interest. Let $x_{l,i}^g$ and $w_{l,i}$ denote the estimated locations of the network center and the agent of interest by predator l at time i. Predators cooperatively estimate $x_{l,i}^g$ by using (7.26). However, since predators spread around the network,

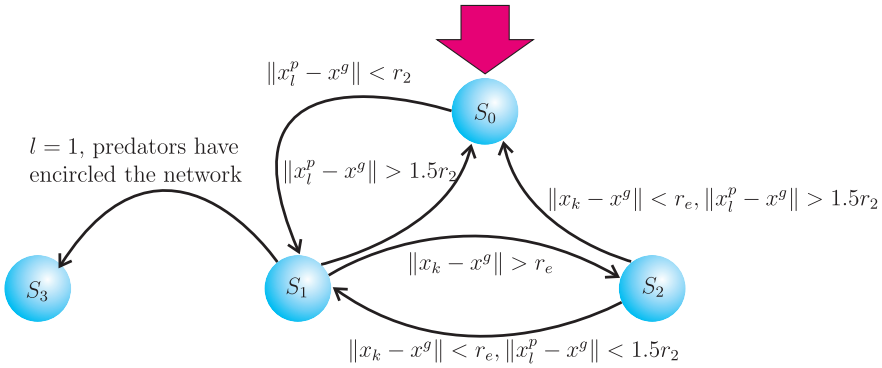


Figure 7.11: State transition diagram of the motion model of predators.

they have different agents of interest. Therefore, each predator has to track $w_{l,i}$ independently by (7.43).

We model the behavior of predators as a finite-state machine with four possible states, S_0 to S_3. The state transition diagram is depicted in Fig. 7.11. Predator l initially enters state S_0 and moves towards the network (i.e., fish school) until it is close to the network, say, $\|x_{l,i}^p - x_{l,i}^g\| < r_2$. Then the predator moves to state S_1 and tries to encircle the network by moving around it. The predator monitors the agent that is within a distance r_s and is the farthest from the network center. If the agent is far away from the network center (i.e., an outlier), say, $\|w_{l,i} - x_{l,i}^g\| > r_e$, the predator enters state S_2 and drives the agent back until it is within the distance r_e from the network center. After that, the predator may go back to state S_1 or S_0 depending on the distance to the network center. If the distance is greater than $1.5r_2$, the predator moves to state S_0; otherwise it moves to state S_1. Similarly, the predator in state S_1 may transit to state S_0 if it is far away from the center, i.e., $\|x_{l,i}^p - x_{l,i}^g\| > 1.5r_2$. Finally, after predators have encircled the network, predators take turns to attack the network. We assume that only one predator, say predator 1, will launch an attack to focus on the agile

adjustment of network patterns in the network and predators.

7.5.2.1 State S_0: Chase

Chasing happens when the distance to the network center is large. To get closer to the network, the predator sets the velocity vector towards the network center, i.e.,

$$q^d_{l,i+1} = \mathfrak{u}(x^g_{l,i} - x^p_{l,i}) \tag{7.45}$$

7.5.2.2 State S_1: Encircle

In S_1, predators would like to encircle the network within a disc with the origin at the network center and radius r_e by moving around the network center. In addition, predators would like to distribute evenly around the network in order to make it difficult for the agents in the network to escape. To avoid staying together, predator l first checks if there are other predators within distance r_s. If yes, say predator j, predator l then determines the direction of predator j by the inner product

$$(x^p_{j,i} - x^g_{l,i})^T (x^p_{l,i} - x^g_{l,i})^\perp \tag{7.46}$$

We say predator j lies in the right semicircle of predator l if the inner product is greater than zero (see Fig. 7.12). Predator l sets the velocity vector towards the empty semicircle; otherwise, predator l randomly chooses a direction, i.e.,

$$q^d_{l,i+1} = c_2 \cdot \mathfrak{u}[(x^p_{l,i} - x^g_{l,i})^\perp] \tag{7.47}$$

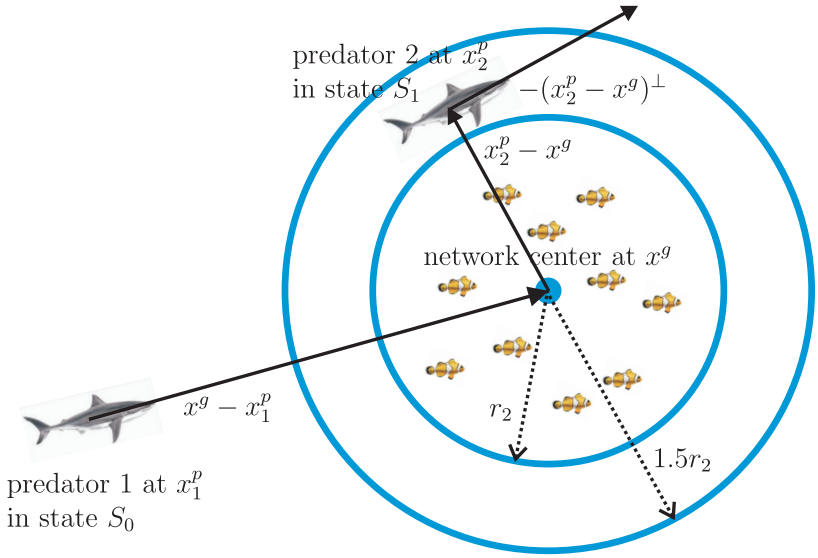


Figure 7.12: Location relations between the network and predators in states S_0 and S_1.

where c_2 determines the direction and is equal to 1 or -1 (i.e., counter-clockwise or clockwise) and

$$c_2 = \begin{cases} 1, & \text{if } (x_{j,i}^p - x_{l,i}^g)^T (x_{l,i}^p - x_{l,i}^g)^\perp < 0 \\ -1, & \text{if } (x_{j,i}^p - x_{l,i}^g)^T (x_{l,i}^p - x_{l,i}^g)^\perp \geq 0 \\ -1 \text{ or } 1 \text{ equally likely}, & \text{if } j = \phi \end{cases} \tag{7.48}$$

That is, if the right semicircle is empty, the predator moves counter-clockwise ($c_2 = 1$). Similarly, the predator moves clockwise ($c_2 = -1$) if the left semicircle is empty. Otherwise, the predator moves towards either direction with equal probability.

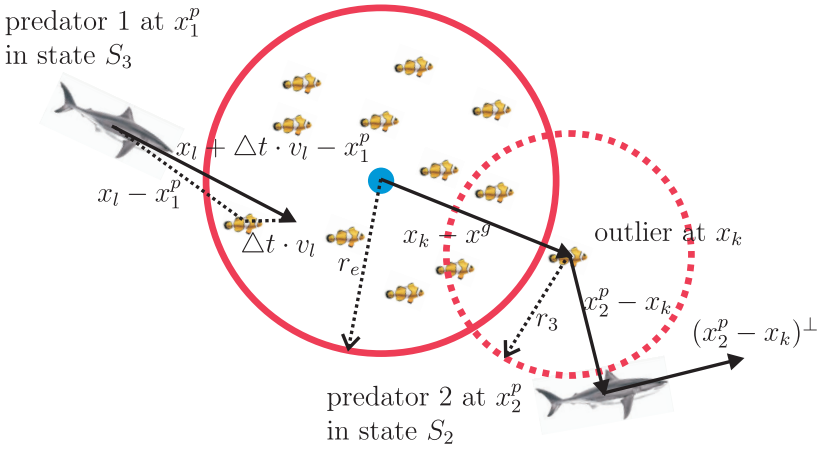


Figure 7.13: Location relations between the network and predators in states S_2 and S_3.

7.5.2.3 State S_2: Trap

Outliers occur when they try to escape from an attack. To push the outlier back, the predator moves to the front of the outlier and blocks it. However, if the predator directly approaches the outlier, it may move further away to escape from the predator. To avoid this situation, the predator keeps a certain distance to the outlier and moves around the outlier until it blocks the way out (see Fig. 7.13). To do so, the predator sets the velocity vector as follows:

$$q_{l,i+1}^d = \begin{cases} \mathfrak{u}(w_{l,i} - x_{l,i}^p) & \text{if } \|w_{l,i} - x_{l,i}^p\| > 1.5r_3 \\ -\mathfrak{u}(w_{l,i} - x_{l,i}^p) & \text{if } \|w_{l,i} - x_{l,i}^p\| < r_3 \\ c_3 \cdot \mathfrak{u}[(w_{l,i} - x_{l,i}^p)^\perp] & \text{otherwise} \end{cases} \tag{7.49}$$

where

$$c_3 = \begin{cases} 1, & \text{if } (w_{l,i} - x_{l,i}^g)^T (x_{l,i}^p - w_{l,i})^\perp \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{7.50}$$

7.5.2.4 State S_3: Attack

When attacking, the predator tracks the location and velocity of the nearest agent and moves towards the predicted location of that agent. That is, the velocity vector of the predator is updated as:

$$q^d_{l,i+1} = \mathfrak{u}(w_{l,i} + \triangle t \cdot q_{l,i} - x^p_{k,i}) \tag{7.51}$$

where $q_{l,i}$ is the estimated velocity of the agent by predator l at time i and is estimated in the same way as (7.36).

7.5.2.5 Control Mechanism

Predators also avoid collisions and will move apart when they are too close. The final adjustment of the velocity and location vectors by predator l is as follows:

$$\begin{aligned} q^p_{l,i+1} &= s^p q^d_{l,i+1} + \gamma^p \delta^p_{l,i} \\ x^p_{l,i+1} &= x^p_{l,i} + \triangle t \cdot v^p_{l,i+1} \end{aligned} \tag{7.52}$$

where $\{s^p, \gamma^p\}$ are non-negative weighting scalars and

$$\delta^p_{l,i} = \frac{1}{|\mathcal{N}^p_l| - 1} \sum_{j \in \mathcal{N}^p_l \setminus \{l\}} \left(r_p - \|x^p_{l,i} - x^p_{j,i}\|\right) \mathfrak{u}(x^p_{l,i} - x^p_{j,i}) \tag{7.53}$$

In (7.53), $\mathcal{N}^p_l$ is a set of predators within distance r_p, i.e.,

$$\mathcal{N}^p_l = \{j : \|x^p_{l,i} - x^p_{j,i}\| < r_p\} \tag{7.54}$$

7.5.3 Simulation Results

For velocity control in (7.42) and (7.52), the parameters are $(\alpha_1, \alpha_2) = (1, 2)$ and $(s^p, \gamma^p) = (2.4, 1)$; the other coefficients remain the same as Section 7.4.1.1. In addition, the distance parameters are set to $r = 3$, $r_p = 5$, $r_s = 30$, $r_e = 15$ and

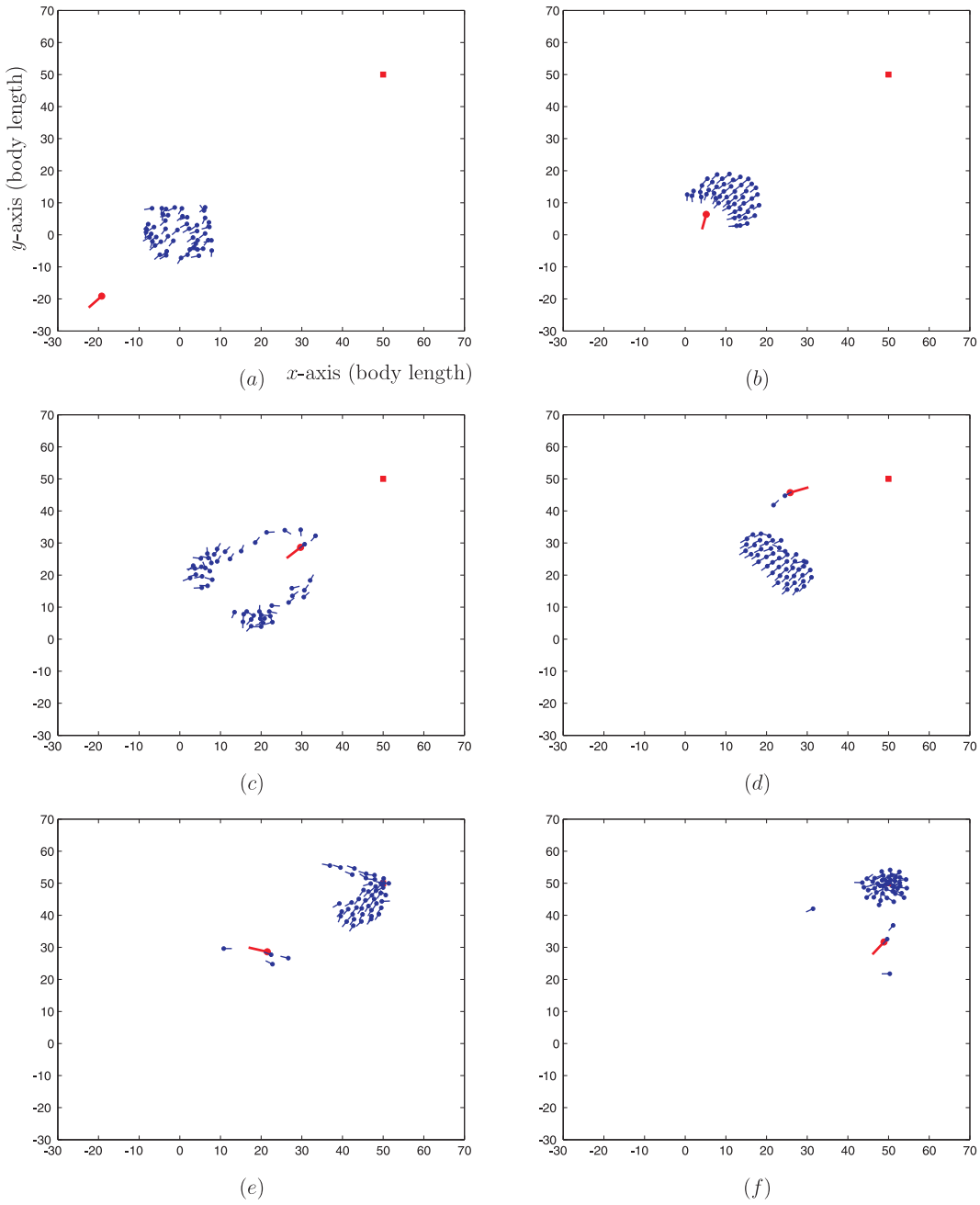


Figure 7.14: Maneuvers of mobile networks in $\mathbb{R}^2$ over time: (a) 0 sec, (b) 15 sec, (c) 30 sec, (d) 45 sec, (e) 105 sec, and (e) 120 sec.

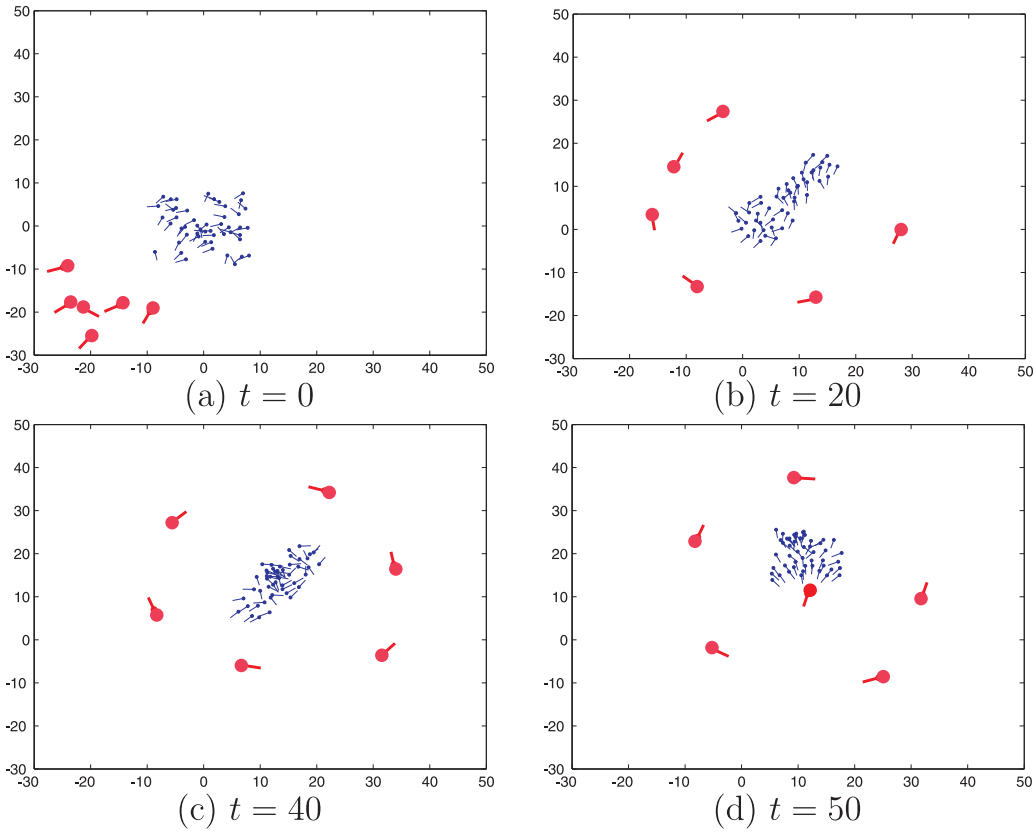


Figure 7.15: A simulation showing how predators coordinate their behavior to encircle a fish school. The behavior of the fish school and the predators are modeled using diffusion adaptation over networks.

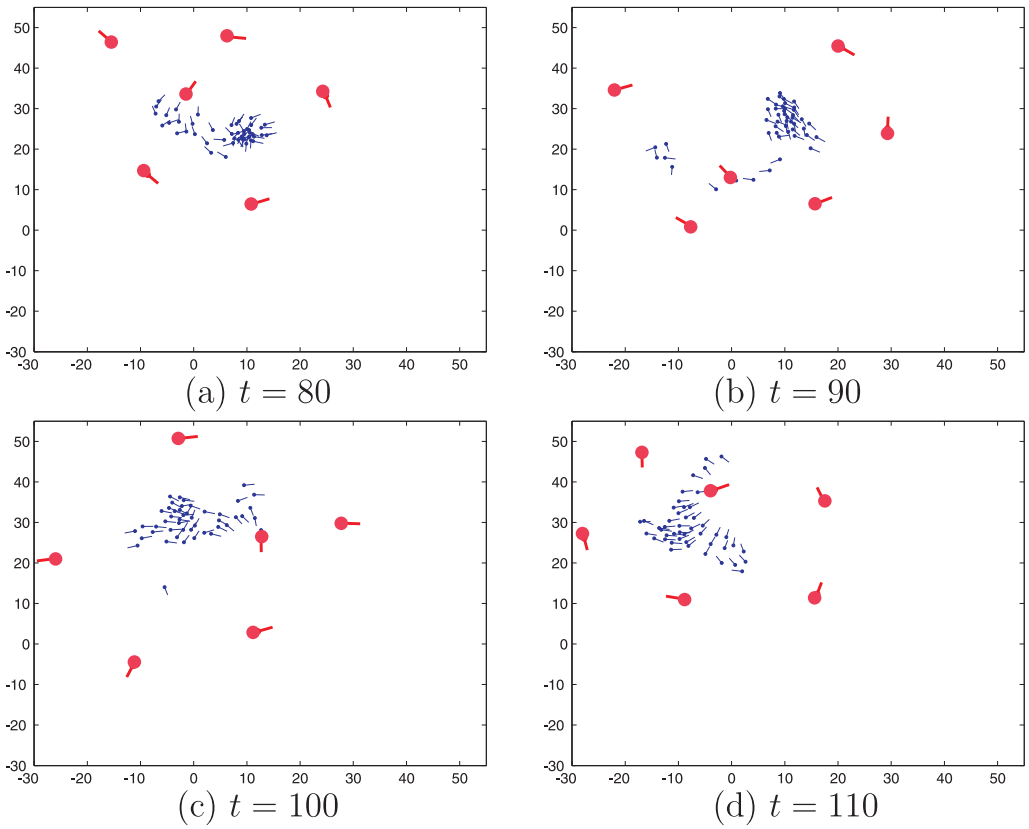


Figure 7.16: Attacking behavior of predators over time.

$(r_1, r_2, r_3) = (10, 20, 15)$. Figure 7.14 illustrates the maneuver of a mobile network with single predator over time. The red symbol with bigger size represents a predator. We observe that the agents in the network move harmoniously and approach the food source. When the predator tries to attack them, the agents spread out and regroup after the attack. The simulation results regenerate the behavior of fish schools in nature.

We also show the maneuver of a mobile network with multiple predators over time in Fig. 7.15 and 7.16. Figure 7.15 shows that predators encircle the network in the beginning and then one predator launches an attack. In Fig. 7.16, we

observe the predators trapping the network while one predator is attacking. The simulation results emulate the interactions between fish schools and predators in nature.

7.6 Information Transfer over Adaptive Networks

In this section, we examine the flow of information through a mobile network when only a small fraction of the agents possess information (e.g., location of a food source or a predator) and know the desired direction of motion. Here, we have a restriction that the agents do not exchange information (e.g., intermediate estimates) so that they do not know whether their neighbors are informed or not. Nevertheless, at time i, each agent k is able to collect the location and velocity vectors $\{x_{l,i}, q_{l,i}\}$ in its neighborhood through observation. The objective of the mobile network in such context becomes how to identify informed agents and how to select the combination weights so that the entire network moves towards the desired direction quickly. In the following, we first reformulate the motion control mechanism (7.18) under this restricted condition. We then develop a procedure for agents to quantify the information their neighbors possess and to select the combination weights so that effective information transfer is attained.

7.6.1 Motion Control Mechanism

We modify rule (7.18) so that the update of the velocity vector at agent k depends only on its available data: $\{x_{l,i}, q_{l,i}\}$ for $l \in \mathcal{N}_k$. The first term in (7.18) represents the desired direction of motion (e.g., direction towards a desired target or direction away from danger) at agent k. For simplicity, we assume this desired

motion is the same for every agent and denote it by

$$h(w_{k,i} - x_{k,i}) = q^d \tag{7.55}$$

Also, for simplicity, we do not consider noisy regions, so we set $\alpha = 0$ to eliminate the effect of the second term in (7.18). The third term in (7.18) denotes the local estimate of the average velocity of the network. Since agents are not allowed to exchange information, one way to approximate $q^g_{k,i}$ is by linearly combining the velocity vectors in the neighborhood, i.e.,

$$q^g_{k,i} = \sum_{l \in \mathcal{N}_k} a_{l,k} q_{l,i} \tag{7.56}$$

where the $\{a_{l,k}\}$ are nonnegative combination weights satisfying (2.25). We also set the parameter β to $1 - \lambda$. Finally, the fourth term in (7.18) relates to the repulsion and attraction mechanisms. As we will show in Appendix 7.B, this term converges to zero in the mean. Therefore, we denote this term by a realization of a random process $\boldsymbol{n}_{k,i}$, i.e.,

$$n_{k,i} = \gamma \delta_{k,i} \tag{7.57}$$

to account for the perturbation/noise due to repulsion and attraction and assume that the random process $\boldsymbol{n}_{k,i}$ has zero mean. Based on the above reformulation, we arrive at the following CTA diffusion strategy for agent k to adjust its velocity vector:

$$\begin{aligned} q^g_{k,i} &= \sum_{l \in \mathcal{N}_k} a_{l,k} q_{l,i} \\ q_{k,i+1} &= q^g_{k,i} + \lambda(q^d - q^g_{k,i}) + n_{k,i} \end{aligned} \tag{7.58}$$

Note that the parameter λ serves as the step-size μ_k in the CTA diffusion strategy (2.32). As we showed in Chapter 5, when agent k is uninformed, the parameter λ is set to zero so that its velocity vector simply becomes a linear combination of the velocity vectors in its neighborhood along with the repulsion and attraction

mechanisms. That is, we have that

$$\mu_k = \begin{cases} \lambda, & \text{if agent } k \text{ is informed} \\ 0, & \text{if agent } k \text{ is uninformed} \end{cases} \tag{7.59}$$

In the next section, we will derive a combination rule to select the weights $\{a_{l,k}\}$ to attain effective information transfer, i.e., to converge to the desired velocity v^d as quickly as possible. Before proceeding, we examine the dependence of convergence of motion control mechanism (7.58) on the combination weights. We introduce the following error vectors

$$\tilde{\boldsymbol{q}}_{k,i} = q^d - \boldsymbol{q}_{k,i} \tag{7.60}$$

for all k and collect them into the network error vector

$$\tilde{\boldsymbol{q}}_i = \text{col}\{\tilde{\boldsymbol{q}}_{1,i}, \tilde{\boldsymbol{q}}_{2,i}, \cdots, \tilde{\boldsymbol{q}}_{N,i}\} \tag{7.61}$$

Then, from (7.58), we arrive at the following recursion for the network error vector:

$$\tilde{\boldsymbol{q}}_{i+1} = (I_{NM} - \mathcal{M})\mathcal{A}^T \tilde{\boldsymbol{q}}_i - \tilde{\boldsymbol{n}}_i \tag{7.62}$$

where

$$\tilde{\boldsymbol{n}}_i = \text{col}\{\tilde{\boldsymbol{n}}_{1,i}, \tilde{\boldsymbol{n}}_{2,i}, \cdots, \tilde{\boldsymbol{n}}_{N,i}\} \tag{7.63}$$

As indicated in Section 2.2.4, the convergence rate of the error recursion (7.62) is determined by

$$r = [\rho((I_{NM} - \mathcal{M})\mathcal{A}^T)]^2 \tag{7.64}$$

Moreover, following the arguments in Lemma 5.3, the convergence rate in (7.64) is bounded by

$$(1-\lambda)^2 \leq r < 1 \tag{7.65}$$

As suggested in Section 6.7.2, in order to attain the fastest convergence rate (i.e., $r = (1-\lambda)^2$), agents can employ rules (6.131) and (6.132). However, in general, agents do not know whether their neighbors are informed or not due to lack of information exchange. We resolve this issue in the next section by noting that the speed of an agent usually reflects the quality of its private information. For example, fish tend to move faster when they sense food or feel danger.

7.6.2 Combination Rules

The choice of the combination weights $\{a_{l,k}\}$ in (7.58) influences the way the agents interact with each other. Different choices for the weights not only lead to different patterns of behavior, but they also influence the flow of information through the network, as revealed by (7.64). In earlier works [48,68], the uniform combination rule (or averaging strategy) has been employed to regenerate the collective motion exhibited by fish schooling or bird flocking. That is, the weights were set to

$$a_{l,k} \begin{cases} 1/(n_k - 1), & \text{if } l \in \mathcal{N}_k \setminus \{k\} \\ 0, & \text{otherwise} \end{cases} \tag{7.66}$$

if agent k is uninformed. On the other hand, if agent k is informed, the weights $\{a_{l,k}\}$ change to the rule employed in [46,47]:

$$a_{l,k} = \begin{cases} a/(a + n_k - 1), & \text{if } l = k \\ 1/(a + n_k - 1), & \text{if } l \in \mathcal{N}_k \setminus \{k\} \\ 0, & \text{otherwise} \end{cases} \tag{7.67}$$

where a is a positive weighting factor. That is, agent k places higher weight on itself if it is informed. However, we will show in simulations that the uniform combination rule fails to transfer information throughout the network in demanding situations.

In the following, we develop a combination rule that is suggested in Section 6.7.2 to improve the convergence rate in (7.64). Even though agents do not know whether their neighbors are informed or not, they may use the speed of their neighbors to infer how informed they are. To explain this idea, we drop the time index, denote the velocity for agent k in the $2D-$plane by $q_k = (q_{1,k}, q_{2,k})$, and let s_k denote its speed:

$$s_k = \|q_k\| = \sqrt{q_{1,k}^2 + q_{2,k}^2} \tag{7.68}$$

In addition, let $\mathbb{I}_k$ be an indicator function for agent k whose value is equal to 1 if agent k is informed; otherwise the value is equal to 0. Then, the combination weights $\{a_{l,k}\}$ can be selected in proportion to the probability that agent l is informed given its speed s_l:

$$a_{l,k} \propto \Pr(\mathbb{I}_l = 1 \mid s_l) \tag{7.69}$$

Note that rule (7.69) simplifies to rules (6.131) and (6.132) if agents know whether their neighbors are informed or not (i.e., $\Pr(\mathbb{I}_l = 1 \mid s_l) = 1$ or 0). Since agent k knows whether it possesses information or not, the probability $\Pr(\mathbb{I}_k = 1 \mid s_k)$ is simply zero or one. In the following, we give a model for the velocity vector q_k and evaluate the probability $\Pr(\mathbb{I}_l = 1 \mid s_l)$.

To begin with, in the absence of neighbors, we assume that the speed of any agent k is set as follows:

$$s^\circ = \begin{cases} c_0, & \text{if } \mathbb{I}_k = 0 \\ c_1, & \text{if } \mathbb{I}_k = 1 \end{cases} \tag{7.70}$$

with $c_1 > c_0$. In this way, when neighbors are not present, agent k moves at speed c_0 when it is uninformed; otherwise, it moves faster at speed $c_1 = \|q^d\|$ towards a food source or away from danger. However, when neighbors are present, the motion of agent k will be affected by its neighbors according to (7.58); in this

case, agent k will not move at a constant speed (c_0 or c_1). To take this effect into account, we introduce the following model:

$$\boldsymbol{q}_{1,k} = s^\circ \cos \boldsymbol{\theta}_k + \boldsymbol{n}_{1,k} \tag{7.71}$$

$$\boldsymbol{q}_{2,k} = s^\circ \sin \boldsymbol{\theta}_k + \boldsymbol{n}_{2,k} \tag{7.72}$$

where $\boldsymbol{\theta}_k$ is the moving direction of agent k, and $\boldsymbol{n}_{1,k}$ and $\boldsymbol{n}_{2,k}$ are Gaussian random variables with zero mean and variance σ_n^2. Moreover, $\boldsymbol{\theta}_k$, $\boldsymbol{n}_{1,k}$, and $\boldsymbol{n}_{2,k}$ are assumed to be independent of each other. Expressions (7.71)-(7.72) model the perturbation caused by the neighbors of agent k. Therefore, the velocity vector q_k, given $\boldsymbol{\theta}_k = \theta_k$, becomes a Gaussian random vector with mean $\begin{bmatrix} s^\circ \cos\theta_k & s^\circ \sin\theta_k \end{bmatrix}^T$ and covariance matrix $\sigma_n^2 I_2$. Thus, the speed s_k in (7.68) is a Rician random variable with parameters $\{s^\circ, \sigma_n^2\}$ [110] and the probability density function (pdf) of s_k, given $\boldsymbol{\theta}_k = \theta_k$, can be written as:

$$f(s_k \mid s^\circ, \sigma_n^2, \boldsymbol{\theta}_k = \theta_k) = \frac{s_k}{\sigma_n^2} \exp\left[\frac{-(s_k^2 + s^{\circ 2})}{2\sigma_n^2}\right] I_0\left(\frac{s_k s^\circ}{\sigma_n^2}\right) \tag{7.73}$$

where $I_0(z)$ is the modified Bessel function of the first kind with order zero, or

$$I_0(z) = \sum_{m=0}^{\infty} \left[\frac{(z/2)^m}{m!}\right]^2 \tag{7.74}$$

Note that expression (7.73) is independent of θ_k. Then, the pdf $f(s_k \mid s^\circ, \sigma_n^2)$ is identical to (7.73), i.e.,

$$f(s_k \mid s^\circ, \sigma_n^2) = \frac{s_k}{\sigma_n^2} \exp\left[\frac{-(s_k^2 + s^{\circ 2})}{2\sigma_n^2}\right] I_0\left(\frac{s_k s^\circ}{\sigma_n^2}\right) \tag{7.75}$$

Using Bayes' rule, the probability $\Pr(\mathbb{I}_l = 1 \mid s_l)$ can be evaluated by

$$\Pr(\mathbb{I}_l = 1 \mid s_l) = \frac{\Pr(\mathbb{I}_l = 1) f(s_l \mid \mathbb{I}_l = 1)}{\sum_{m=0}^{1} \Pr(\mathbb{I}_l = m) f(s_l \mid \mathbb{I}_l = m)} \tag{7.76}$$

where

$$f(s_l \mid \mathbb{I}_l = m) = f(s_l \mid s^\circ = c_m, \sigma_n^2) \tag{7.77}$$

Since agents do not have prior information about whether other agents are informed or not, they simply set the prior probabilities, $\Pr(\mathbb{I}_l = 1)$ and $\Pr(\mathbb{I}_l = 0)$, to equal values (namely, 1/2). Substituting the pdf from (7.73) into (7.76), we arrive at

$$\Pr(\mathbb{I}_l = 1 \mid s_l) = \left[1 + \exp\left(\frac{c_1^2 - c_0^2}{2\sigma_n^2}\right) \frac{I_0\left(\frac{s_l c_0}{\sigma_n^2}\right)}{I_0\left(\frac{s_l c_1}{\sigma_n^2}\right)}\right]^{-1} \tag{7.78}$$

However, the Bessel function (7.74) is difficult to compute. We can use the following approximation

$$\frac{I_0\left(\frac{s_l c_0}{\sigma_n^2}\right)}{I_0\left(\frac{s_l c_1}{\sigma_n^2}\right)} \approx \exp\left[\frac{-s_l(c_1 - c_0)}{\sigma_n^2}\right] \tag{7.79}$$

and expression (7.78) simplifies to

$$\Pr(\mathbb{I}_l = 1 \mid s_l) \approx \left\{1 + \exp\left[\frac{c_1 - c_0}{\sigma_n^2}\left(\frac{c_1 + c_0}{2} - s_l\right)\right]\right\}^{-1} \tag{7.80}$$

Expressions (7.78) and (7.80) are depicted in Fig. 7.17 with parameters $(c_1, c_0, \sigma_n^2) = (4, 1, 1)$. We observe that the two curves are close to each other. Note that expression (7.80) admits a physical interpretation. The probability (7.80) attains the value of 0.5 when $s_l = (c_1 + c_0)/2$, which is the middle point of c_0 and c_1. That is, when the speed of one agent passes the middle point, it has higher probability of being informed. In addition, the slope of the curve near the middle point is determined by $(c_1 - c_0)/\sigma_n^2$. If the difference between the speed of informed and uninformed agents is large, agents have better ability of distinguishing whether other agents are informed or not. We assume that every agent knows the values of the parameters (c_1, c_0, σ_n^2).

We conclude from (7.69) and Fig. 7.17 that the resulting combination rule exhibits a sigmoidal shape so that a agent places higher weights on faster-moving agents. In this way, when a fast-moving agent takes a sharp turn, for example,

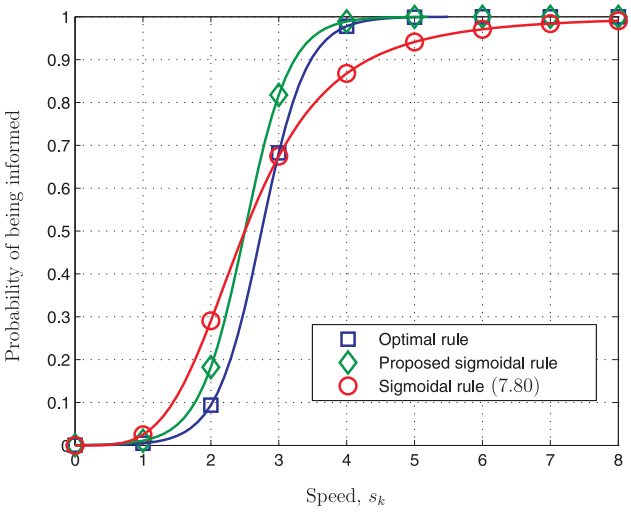


Figure 7.17: Sigmoidal combination rules: the larger the speed of a agent, the larger the weight assigned to it.

the effect of this behavior will ripple through the network at a faster rate and the remaining agents will follow suit. The choice of the combination rule (7.80) is similar to the decision-making process in animal groups, as we studied in Chapter 6. Motivated by the quorum response in [138], another way (recall (6.34)) to determine the probability $\Pr(\mathbb{I}_l = 1 \mid s_l)$ is

$$\Pr(\mathbb{I}_l = 1 \mid s_l) = \frac{s_l^K}{s_l^K + c^K} \tag{7.81}$$

Expression (7.81) is also shown in Fig. 7.17 for $(c, K) = (2.5, 4)$. Similar to (7.80), expression (7.81) attains the value of 0.5 when $s_l = c$ and the slope of the curve near the middle point is determined by K. We also compare the performance of the two sigmoidal rules (7.80) and (7.81) in simulations.

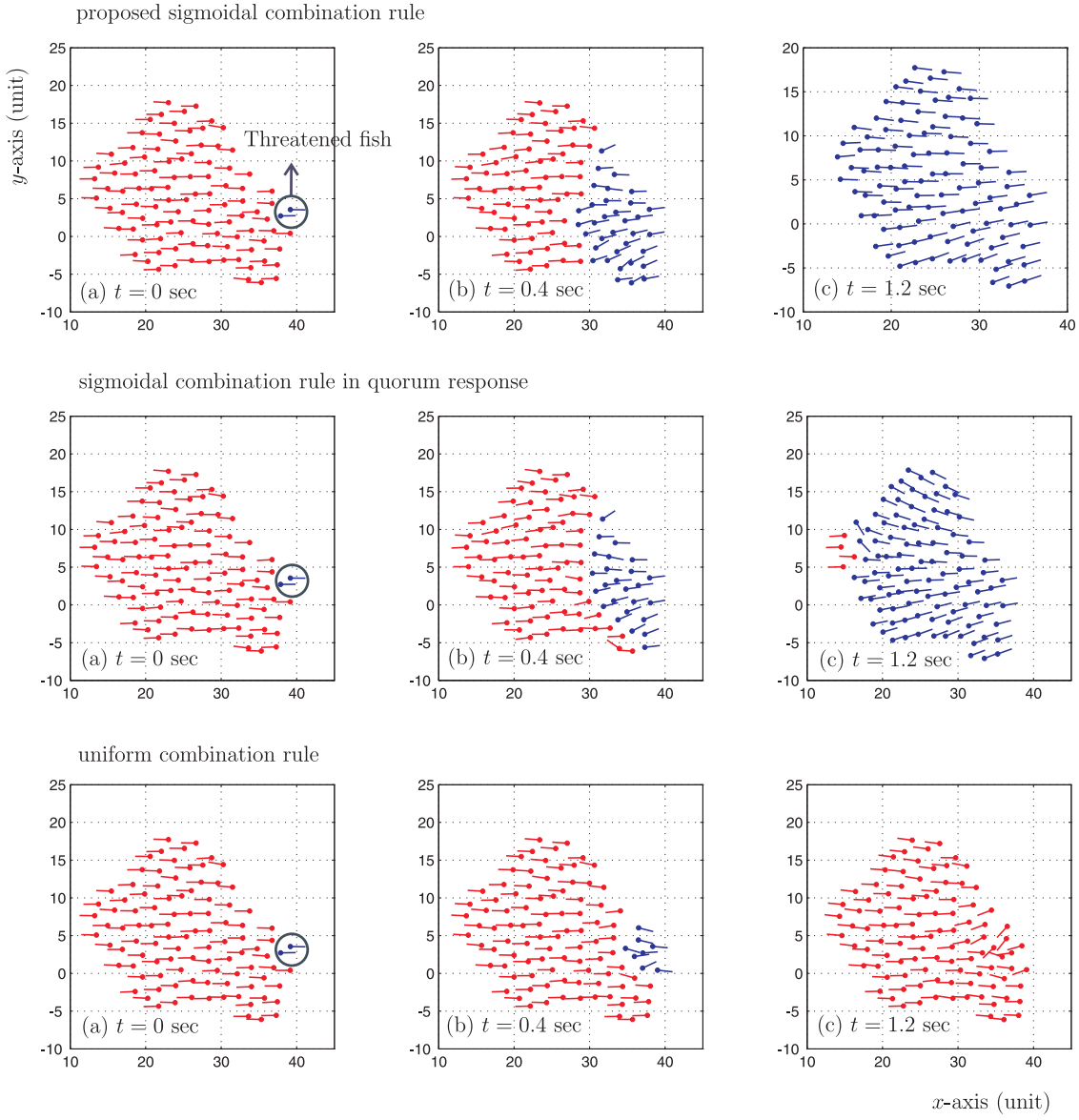


Figure 7.18: Maneuver of fish schools with the two sigmoidal (top (7.80) and middle (7.81)) and uniform (bottom (7.66)-(7.67)) combination rules over time (a) $t = 0$ (b) $t = 0.4$ (c) $t = 1.2$ sec.

7.6.3 Simulation Results

We model the information propagation in fish schools in [136] and compare three combination rules: two sigmoidal rules from (7.80) with $(c_1, c_0, \sigma_n^2) = (4, 1, 1)$ and from (7.81) with $(c, K) = (2.5, 4)$, and the uniform rule (7.66) and (7.67) with $a = 5$. The step-sizes are set to $\lambda = 0.6$ for informed agents. Figure 7.18 shows simulation results for a network with $N = 100$ agents. Initially, the velocities of the agents are set to $q_k = (1, 0)$ for all k. To choose N_I threatened (informed) agents, we first pick up the agent with the largest x-coordinate and then choose $N_I - 1$ agents that are closest to the chosen agent. In simulations, we set $N_I = 2$. The desired velocities of the informed agents are set to $q^d = (-4, 0)$ for 5 time steps with $\triangle t = 0.1$ sec. The resulting maneuver of the networks using the sigmoidal and uniform combination rules are shown in Fig. 7.18. The agents moving towards the positive (negative) x-direction are shown in red (blue). We observe that the motion of the informed agents propagates rapidly through the entire network if the network employs the sigmoidal combination rules (7.80) or (7.81), while the network using the uniform combination rule (7.66) fails to transfer the motion through the entire network. To compare these three combination rules quantitatively, we measure the magnitude and orientation of the velocity of the center of mass of the network, q_i^g in (7.17), relative to the desired velocity, q^d. That is, we introduce

$$\triangle s(i) = (\|q_i^g\| - \|q^d\|)^2 \tag{7.82}$$

$$\triangle o(i) = [\angle(q_i^g) - \angle(q^d)]^2 \tag{7.83}$$

These two quantities are averaged over 100 experiments and shown in Fig. 7.19. We observe that the desired velocity of the informed agents is successfully transferred through the network if the network adopts the sigmoidal combination rules. Moreover, comparing the two sigmoidal rules, we observe that rule (7.80) out-

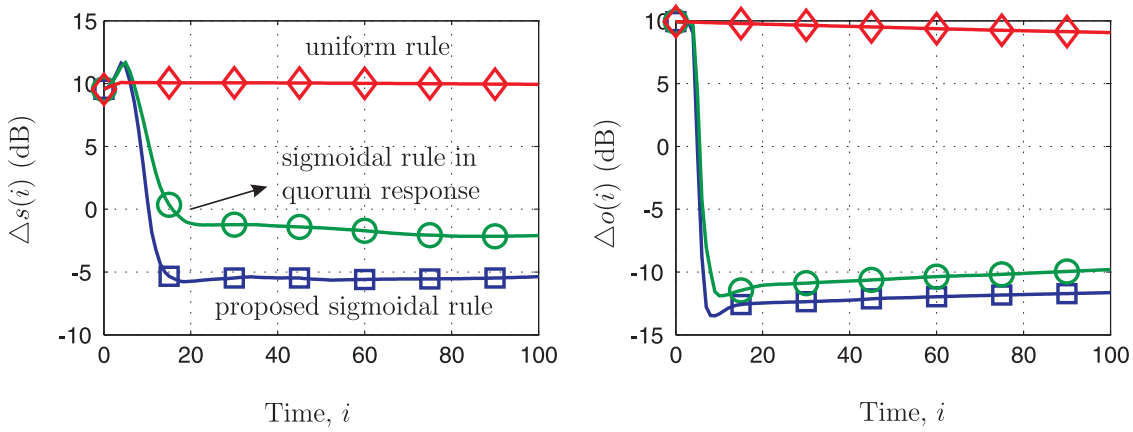


Figure 7.19: Magnitude, $\triangle s(i)$, and orientation, $\triangle o(i)$, of the velocity of the center of mass relative the the trigger velocity.

performs rule (7.81). *The results indicate that if the information of the agents is modulated according to their speed, this mechanism improves the efficiency of information transfer over the network.*

7.7 Concluding Remarks

In this chapter, we developed diffusion strategies for mobile networks. The strategies involve two diffusion steps: one for estimating the location a target and another for tracking the center of mass of the network. The strategies also include velocity and location control mechanisms to dictate the motion of the agents. The resulting motion mechanism is able to bypass regions of bad signal conditions and help the network move toward the desired target. We illustrated via simulations how the algorithms can emulate the coherent motion of fish schools and bypass obstacles (described as high noise regions). The adaptation and motion control mechanisms are further used to emulate the interactive behavior

between fish schools and predators. The algorithm helps explain how fish schools avoid attacks from predators and how predators cooperatively hunt prey. We also examined mobile networks with small fraction of informed agents to model the effective information transfer among fish. We showed how the speed information can be exploited and incorporated into the design of the combination rules for mobile networks. The analysis leads to a sigmoidal function construction, the simulation show that the proposed combination rule leads to more effective information flow over networks of mobile agents.

7.A Formation Control Using Averaging Consensus

Collective motion with localized interactions has been extensively studied in the literature [25, 27, 45, 48, 135, 161, 169]. One useful model for generating collective motion over multi-agent networks has been proposed by [120] and is based on the agents applying three rules of behavior:

1. Velocity matching: agents attempt to match the velocity of their neighboring agents.

2. Flock centering: agents attempt to stay close to neighboring agents.

3. Collision avoidance: agents avoid collision with neighboring agents.

These motion control mechanisms were initially verified by means of computer simulations until more formal studies appeared in the control literature [55, 56, 68, 100, 107, 109, 118]. References [108, 119] provide useful overviews

In this appendix, we examine the motion control mechanism from [107] and compare it with the proposed mechanism in this Chapter (see Algorithm 7.1). The mechanism in [107] is described in continuous-time where we denote by

$\{x_k(t), q_k(t)\}$ the location and velocity vectors of node k at time t. In order to achieve the three rules proposed by [120], every node k adjusts its location and velocity vectors according to the following rule:

$$\dot{x}_k = q_k \tag{7.84}$$

$$\dot{q}_k = -\nabla_{x_k} U(x) - \nabla_{x_k} V(x) - \nabla_{q_k} W(q) \tag{7.85}$$

where the vectors $\{x, q\}$ collect the $\{x_k, q_k\}$ into block vectors, i.e.,

$$x = \text{col}\{x_k\} \quad \text{and} \quad q = \text{col}\{q_k\}. \tag{7.86}$$

There are three components in updating the velocity of node k in (7.85). The first component in (7.85), $-\nabla_{x_k} U(x)$, helps the agents move towards a direction of interest. For example, when the agents would like to move towards a target located at w°, we then set the function $U(x)$ to

$$U(x) = \frac{1}{2} \sum_{k=1}^{N} \|w^\circ - x_k\|^2 \tag{7.87}$$

and the vector $-\nabla_{x_k} U(x)$ becomes the vector pointing to w° from x_k, i.e.,

$$-\nabla_{x_k} U(x) = w^\circ - x_k \tag{7.88}$$

The second component in (7.85), $-\nabla_{x_k} V(x)$, enforces attraction and repulsion between two agents and the function $V(x)$ is constructed as [56]:

$$V(x) = \frac{1}{2} \sum_{k=1}^{N} \sum_{l \neq k} J(\|x_l - x_k\|) \tag{7.89}$$

with the function $J(\|x_l - x_k\|)$ representing the attraction and repulsion between agents k and l, and its value depending on the distance between these two agents. Then, the term $-\nabla_{x_k} V(x)$ becomes

$$\begin{aligned} -\nabla_{x_k} V(x) &= -\sum_{l \neq k} \nabla_{x_k} J(\|x_l - x_k\|) \\ &= \sum_{l \neq k} \delta(x_l - x_k) \end{aligned} \tag{7.90}$$

where we denoted the function $-\nabla_{x_k} J(\|x_l - x_k\|)$ by $\delta(x_l - x_k)$. In [109], the function $J(\|x_l - x_k\|)$ has a global minimum at $\|x_l - x_k\| = r$. Note that even though the summation in (7.90) is over all agent indices (except index k), the function $\delta(y)$ shall be selected to satisfy

$$\delta(y) = 0 \quad \text{for } \|y\| \geq R \tag{7.91}$$

so that the interaction between any two agents k and l vanishes to zero if the distance between them is greater than a certain distance R. The third component in (7.85), $W(q)$, enforces alignment by setting

$$W(q) = \frac{1}{2} \sum_{k=1}^{N} \sum_{l \in \mathcal{N}_k} \|q_l - q_k\|^2 \tag{7.92}$$

so that

$$-\nabla_{q_k} W(q) = \sum_{l \in \mathcal{N}_k} (q_l - q_k) \tag{7.93}$$

That is, the function $-\nabla_{q_k} W(q)$ is a continuous-time averaging consensus step studied in [109]; a discrete-time counterpart is studied in [68, 100, 118]. As a result, agent k updates its velocity vector according to the rule:

$$\dot{q}_k = (w^\circ - x_k) + \sum_{l \neq k} \delta(x_l - x_k) + \sum_{l \in \mathcal{N}_k} (q_l - q_k) \tag{7.94}$$

The following result is established in [107].

Theorem 7.1. *If every agent k employs the motion control mechanism in (7.84) and (7.94), then it holds that*

1. *The location vectors $\{x_k\}$ converge to a local minimum of $U(x) + V(x)$.*
2. *The velocity vectors $\{q_k\}$ converge to the zero vector.*

Proof. See Theorem 3 in [107]. □

The first statement in Theorem 7.1 implies that the location vectors $\{x_k\}$ converge to the vectors $\{x_k^\circ\}$ satisfying

$$(w^\circ - x_k^\circ) + \sum_{l \neq k} \delta(x_l^\circ - x_k^\circ) = 0 \tag{7.95}$$

for all k. That is, in steady-state, all agents achieve balance between two forces: the force to move towards the target and the force to keep a safe distance to each other. Moreover, in steady-state, all agents stop moving.

For ease of comparison, we rewrite the motion control mechanism from the body of the chapter as (we set $\alpha = 0$ in (7.18)):

$$q_{k,i+1} = \lambda h(w_{k,i} - x_{k,i}) + \gamma \sum_{l \neq k} \delta(x_l - x_k) + \beta q_{k,i}^g \tag{7.96}$$

$$x_{k,i+1} = x_{k,i} + \Delta t \cdot q_{k,i+1} \tag{7.97}$$

where from (7.18) and (7.20), the function $\delta(y)$ in (7.96) is defined as

$$\delta(y) = \begin{cases} (\|y\| - r)\,\frac{y}{\|y\|}, & \text{if } \|y\| \leq R \\ 0, & \text{otherwise} \end{cases} \tag{7.98}$$

Comparing (7.96) with (7.94), we observe that there are three differences. First, in this chapter, we studied motion control in discrete-time with *constant* parameters $\{\lambda, \beta, \gamma, \Delta t\}$. To ensure the convergence of (7.96)-(7.97), we needed to examine conditions on the parameters $\{\lambda, \beta, \gamma, \Delta t\}$. Second, in the first term in (7.96), we replace the location of the desired target w° in (7.94) by the local estimate $w_{k,i}$ and we also constrain the speed at agent k using the function $h(\cdot)$ in (7.12). Therefore, the agents using (7.96)-(7.97) are endowed with learning and adaptation abilities to estimate the location of the target on the fly. In this way, the agents do not simply perform collective motion, they have an objective and they attain the objective through distributed learning strategies. Such

generalization enables us to model sophisticated behavior in nature, such as the prey-predator behavior. Finally, in the third term in (7.96), we replace the averaging consensus step for velocity by the local estimate $q_{k,i}^g$ of the velocity of the center of mass of the network. It is shown in Appendix 7.B that agents move more coherently by using the term $q_{k,i}^g$. In sum, the analysis in Appendix 7.B extends Theorem 7.1 to the motion control mechanism in discrete-time and with stochastic approximations (see Theorems 7.2 and 7.3). These extensions compound the complexity of the analysis. This is because the data observed by the agents are noisy and the estimates at the agents are random. In other words, relation (7.96) is inherently stochastic, rather than deterministic, as in (7.94). In addition, the learning process and motion control mechanism are intertwined with each other. One of the main contributions in this chapter is that we are able to study both processes under reasonable conditions.

7.B Performance Analysis of Mobile Diffusion Networks

Studying the performance of the mobile adaptive network is challenging for a couple of reasons: (a) the control of the velocity vector (7.18) depends generally in a nonlinear manner on the target estimate, $\boldsymbol{w}_{k,i}$, and on the agent locations, $\{\boldsymbol{x}_{k,i}\}$; (b) the adaptive updates for the estimates $\boldsymbol{w}_{k,i}$ and $\boldsymbol{q}_{k,i}^g$ represent stochastic updates with nonlinear dependencies on the regression and measurement data; (c) the local estimation errors and measurement noise propagate through the network during the cooperation process; and (d) the agents influence each other through the network topology. For these reasons, some simplifying assumptions are necessary to help reveal the essence of the network dynamics to first-order and for sufficiently small step-sizes. Simplifying assumptions are common even in the literature of stand-alone adaptive filters (such as LMS) because, by their very

nature, adaptive filters are nonlinear, stochastic, and time-varying systems [123]. When a multitude of adaptive agents are connected through a topology and allowed to influence each other's behavior, then the difficulty of the performance analysis is compounded relative to stand-alone filters. For this reason, we motivate and introduce some approximations to facilitate the analysis. Later, the simulations reveal that there is good match between the theoretical results obtained under the assumptions and the simulated behavior of the network.

There are several error quantities of interest to us which are defined below:

$$\begin{aligned}
\tilde{\boldsymbol{w}}_{k,i} &= w^\circ - \boldsymbol{w}_{k,i} \quad \text{(error by agent } k \text{ in estimating } w^\circ\text{)} \\
\tilde{\boldsymbol{q}}^g_{k,i} &= \boldsymbol{q}^g_i - \boldsymbol{q}^g_{k,i} \quad \text{(error by agent } k \text{ in tracking } \boldsymbol{q}^g_i\text{)} \\
\tilde{\boldsymbol{q}}_{k,i} &= \boldsymbol{q}^g_i - \boldsymbol{q}_{k,i} \quad \text{(velocity of agent } k \text{ relative to } \boldsymbol{q}^g_i\text{)} \\
\tilde{\boldsymbol{x}}_{k,i} &= \boldsymbol{x}^g_i - \boldsymbol{x}_{k,i} \quad \text{(location of agent } k \text{ relative to } \boldsymbol{x}^g_i\text{)} \\
\tilde{\boldsymbol{x}}^g_i &= w^\circ - \boldsymbol{x}^g_i \quad \text{(location of the center of mass relative to } w^\circ\text{)}
\end{aligned}$$

Note that we are now using boldface letters to represent random quantities. The first two error quantities are related to estimation problems (estimation of w° and tracking of q^g) and they measure how far the estimates at the agents are away from the true values. The remaining error quantities pertain to the motion of the network. The third quantity measure the coherence of the motion by the mismatch towards the velocity of the center of mass while the last two quantities measure the locations of the agents relative to the center of mass and the location of the center of mass relative to w°. Among other factors, we are initially interested in deriving conditions that help ensure that $\mathbb{E}\boldsymbol{w}_{k,i} \rightarrow w^o$, $\mathbb{E}\boldsymbol{x}^g_i \rightarrow w^o$ $\mathbb{E}[\boldsymbol{q}^g_{k,i} - \boldsymbol{q}^g_i] \rightarrow 0$, and $\mathbb{E}[\boldsymbol{q}_{k,i} - \boldsymbol{q}^g_i] \rightarrow 0$. Under small step-size conditions, these results would mean that, on average, the agents in the network are able to estimate and approach w° (global objective) and are able to estimate and align their motion with the center of mass (synchronized motion). Note that we do

not expect the error quantity $\tilde{\boldsymbol{x}}_{k,i}$ to converge to zero in the mean because of the assumed repulsion between the agents. We are also interested in deriving expressions for the mean-square errors $\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2$, $\mathbb{E}\|\tilde{\boldsymbol{q}}^g_{k,i}\|^2$, and $\mathbb{E}\|\tilde{\boldsymbol{q}}_{k,i}\|^2$ in order to assess closely how well the agents approach w° and $\boldsymbol{q}^g_i$.

As is evident from equations (7.19)-(7.20) and Algorithm 7.1, the evolution of the above error quantities influence each other. For example, the velocity update (7.18) depends on the velocity estimate, $\boldsymbol{q}^g_{k,i}$, which in turn relies on the current velocity vector. For this reason, we find it useful to derive a state-space model that relates the evolution of several quantities of interest. Subsequently, we derive conditions on the parameters $\{\mu_k, \nu_k, \lambda, \alpha, \beta, \gamma\}$ to ensure the stability of the model. Since we need to keep track of several error quantities, it is understandable that the notation can quickly become cumbersome. Nevertheless, the main objective is clear: we are deriving recursions for the error quantities in order to capture their inter-dependencies through a state-space model. We can interpret the various error quantities across all agents as entries in a state vector for the network. By studying the evolution of the state vector, we can gain insights into the operation of the network.

7.B.1 Recursions for Estimation Errors

We begin by evaluating recursions for the estimation errors $\tilde{\boldsymbol{w}}_{k,i}$ and $\tilde{\boldsymbol{q}}^g_{k,i}$. The recursion for $\tilde{\boldsymbol{w}}_{k,i}$ can be obtained from the results of Section 2.2.1. Some algebra shows that the network error vector $\tilde{\boldsymbol{w}}_i$ evolves according to the relation:

$$\tilde{\boldsymbol{w}}_i = \mathcal{A}^T(I_{NM} - \mathcal{M})\tilde{\boldsymbol{w}}_{i-1} - \mathcal{A}^T\mathcal{M}\boldsymbol{s}_i \tag{7.99}$$

where

$$\boldsymbol{s}_i = \operatorname{col}\left\{\boldsymbol{\eta}_{1,i}, \boldsymbol{\eta}_{2,i}, \ldots, \boldsymbol{\eta}_{N,i}\right\} \tag{7.100}$$

The recursion for $\tilde{\boldsymbol{q}}_{k,i}^{g}$ can be derived in the similar manner. We introduce the diagonal matrix of step-sizes across the network:

$$\mathcal{M}_q = \text{diag}\{\nu_1 I_M, \nu_2 I_M, \ldots, \nu_N I_M\} \tag{7.101}$$

as well as the extended matrices of the combination matrices $\{C, A_q\}$ (containing the combination weights $\{c_{l,k}\}$ and $\{a_{l,k}^q\}$, respectively):

$$\mathcal{C} = C \otimes I_M \quad \text{and} \quad \mathcal{A}_q = A_q \otimes I_M \tag{7.102}$$

We also introduce the network error vectors:

$$\tilde{\boldsymbol{q}}_i^g = \text{col}\left\{\tilde{\boldsymbol{q}}_{1,i}^g, \tilde{\boldsymbol{q}}_{2,i}^g, \ldots, \tilde{\boldsymbol{q}}_{N,i}^g\right\} \tag{7.103}$$

$$\tilde{\boldsymbol{q}}_i = \text{col}\left\{\tilde{\boldsymbol{q}}_{1,i}, \tilde{\boldsymbol{q}}_{2,i}, \ldots, \tilde{\boldsymbol{q}}_{N,i}\right\} \tag{7.104}$$

Then, using (7.28) and applying the slow-variation approximation $\boldsymbol{q}_i^g \approx \boldsymbol{q}_{i-1}^g$ (this approximation is justified in the next section), we can verify that $\tilde{\boldsymbol{q}}_i^g$ evolves according to the recursion:

$$\tilde{\boldsymbol{q}}_i^g = \mathcal{A}_q{}^T(I - \mathcal{M}_q)\tilde{\boldsymbol{q}}_{i-1}^g + \mathcal{A}_q{}^T\mathcal{M}_q\mathcal{C}^T\tilde{\boldsymbol{q}}_i \tag{7.105}$$

Note that in contrast to the recursion for $\tilde{\boldsymbol{w}}_i$, the recursion for $\tilde{\boldsymbol{q}}_i^g$ in (7.105) depends on the motion of the agents.

7.B.2 Recursions for Motion Behavior

7.B.2.1 Far-Field Approximation

By far field, we mean that the agents are far from the target so that $\|w^\circ - \boldsymbol{x}_{k,i}\| \gg s$. In addition, we assume that the network satisfies $\|\tilde{\boldsymbol{x}}_i^g\| \gg \|\tilde{\boldsymbol{x}}_{k,i}\|$ for all k. This means that the agents are close to each other compared to the distance of the target to the center of mass. Note that this assumption imposes a constraint

on the safe distance r in (7.20) that an agent keeps from its neighbors. This is because r influences the size of the network, which is defined as the maximum $\|\tilde{\boldsymbol{x}}_{k,i}\|$ over all k. Therefore, the distance r should be small enough such that the assumption $\|\tilde{\boldsymbol{x}}_i^g\| \gg \|\tilde{\boldsymbol{x}}_{k,i}\|$ holds in the far field. Another assumption we will use in the analysis is that $\|\tilde{\boldsymbol{x}}_i^g\| \gg \|\tilde{\boldsymbol{w}}_{k,i}\|$. This assumption imposes a constraint on how the noise level varies over the spatial region and it can be motivated as follows. Consider the measurement $\boldsymbol{p}_{k,i}$ from model (7.7). The noise covariance matrix is expected to vary in proportion to the distance between the target and the agent since measurements tend to be noisier at farther distances. We assume that $R_{\eta,k,i}$ satisfies a model of the form (7.29). In this case, as the agent gets closer to the target, i.e., as $\|w^\circ - \boldsymbol{x}_{k,i}\|$ becomes smaller, the noise term $\boldsymbol{\eta}_{k,i}$ in (7.7) will become smaller. Since through filtering and processing, agent k obtains an estimate $\boldsymbol{w}_{k,i}$ for w° that is expected to be an improvement over the raw measurement, $\boldsymbol{p}_{k,i}$, then the variance of $w^\circ - \boldsymbol{w}_{k,i}$ is expected to be smaller than the variance of $w^\circ - \boldsymbol{p}_{k,i}$, especially as $i \to \infty$. That is, we expect

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \leq \mathbb{E}\|\boldsymbol{\eta}_{k,i}\|^2 = \mathrm{Tr}(R_{\eta,k,i}) \tag{7.106}$$

Dividing both sides by $\|w^\circ - \boldsymbol{x}_{k,i}\|^2$, we find that the following relation should hold

$$\frac{\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2}{\|w^\circ - \boldsymbol{x}_{k,i}\|^2} \leq \frac{\mathbb{E}\|\boldsymbol{\eta}_{k,i}\|^2}{\|w^\circ - \boldsymbol{x}_{k,i}\|^2} = \kappa M \tag{7.107}$$

If κ is small enough (we choose $\kappa = 0.01$ in the simulation), then it follows that the variance $\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2$ is small in relation to $\|w^\circ - \boldsymbol{x}_{k,i}\|^2$. Using the fact that $\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 \geq (\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|)^2$ and

$$w^\circ - \boldsymbol{x}_{k,i} = \tilde{\boldsymbol{x}}_i^g + \tilde{\boldsymbol{x}}_{k,i} \approx \tilde{\boldsymbol{x}}_i^g \tag{7.108}$$

we see that it is reasonable to assume that $\|\tilde{\boldsymbol{x}}_i^g\| \gg \mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|$.

To examine the motion behavior we simplify (7.18) by setting $\alpha \approx 0$. That is, we ignore the influence of the SNR levels in (7.18) (this assumption reduces the influence of the time variations in the noise levels and facilitates our first-order analysis). Referring to the velocity adjustment rule (7.18) in the far field, i.e., when $\|w^\circ - \boldsymbol{x}_{k,i}\| \gg s$, we rewrite it as:

$$\begin{aligned} \boldsymbol{q}_{k,i+1} &= \lambda \cdot s \frac{\boldsymbol{w}_{k,i} - \boldsymbol{x}_{k,i}}{\|\boldsymbol{w}_{k,i} - \boldsymbol{x}_{k,i}\|} + \beta \boldsymbol{q}_{k,i}^g + \gamma \boldsymbol{\delta}_{k,i} \\ &\approx \lambda \cdot s \frac{\tilde{\boldsymbol{x}}_i^g}{\|\tilde{\boldsymbol{x}}_i^g\|} + \beta \boldsymbol{q}_i^g - \beta \tilde{\boldsymbol{q}}_{k,i}^g + \gamma \boldsymbol{\delta}_{k,i} \end{aligned} \tag{7.109}$$

where we used

$$\boldsymbol{w}_{k,i} - \boldsymbol{x}_{k,i} = \tilde{\boldsymbol{x}}_i^g + \tilde{\boldsymbol{x}}_{k,i} - \tilde{\boldsymbol{w}}_{k,i} \tag{7.110}$$

and the assumptions $\|\tilde{\boldsymbol{x}}_i^g\| \gg \|\tilde{\boldsymbol{x}}_{k,i}\|$ and $\|\tilde{\boldsymbol{x}}_i^g\| \gg \|\tilde{\boldsymbol{w}}_{k,i}\|$. The first two terms in (7.109) represent the motion of the center of mass, which helps the agents move coherently, while the last two terms in (7.109) refer to the disturbance due to the estimation error $\tilde{\boldsymbol{q}}_{k,i}^g$ and the attraction and repulsion effect $\boldsymbol{\delta}_{k,i}$. Using

$$\sum_{k=1}^{N} \boldsymbol{\delta}_{k,i} = 0 \tag{7.111}$$

and the approximation

$$\frac{1}{N} \sum_{k=1}^{N} \tilde{\boldsymbol{q}}_{k,i}^g \approx 0 \tag{7.112}$$

for large number of agents, we obtain from (7.109) a recursion for $\boldsymbol{q}_i^g$ (defined in (7.17)) as follows:

$$\boldsymbol{q}_{i+1}^g \approx \lambda \cdot s \frac{\tilde{\boldsymbol{x}}_i^g}{\|\tilde{\boldsymbol{x}}_i^g\|} + \beta \boldsymbol{q}_i^g \tag{7.113}$$

Moreover, since the speed of the agents is constrained and $\triangle t$ is small, it is reasonable to assume that $\tilde{\boldsymbol{x}}_i^g \approx \tilde{\boldsymbol{x}}_{i-1}^g$ and then the approximation $\boldsymbol{q}_{i+1}^g \approx \boldsymbol{q}_i^g$. Thus, $\boldsymbol{q}_i^g$ can be simplified to

$$\boldsymbol{q}_{i+1}^g \approx \frac{\lambda \cdot s}{1-\beta} \frac{\tilde{\boldsymbol{x}}_i^g}{\|\tilde{\boldsymbol{x}}_i^g\|} \tag{7.114}$$

That is, $\boldsymbol{q}_i^g$ converges to the direction of the target from the center of mass. Note that to make the network move towards the target we require at least that $\lambda \cdot s/(1-\beta) > 0$. To obtain a recursion for $\tilde{\boldsymbol{q}}_i$ in the far field, we introduce the network vectors:

$$\tilde{\boldsymbol{x}}_i = \operatorname{col}\{\tilde{\boldsymbol{x}}_{1,i}, \tilde{\boldsymbol{x}}_{2,i}, \ldots, \tilde{\boldsymbol{x}}_{N,i}\} \tag{7.115}$$

$$\boldsymbol{\delta}_i = \operatorname{col}\{\boldsymbol{\delta}_{1,i}, \boldsymbol{\delta}_{2,i}, \ldots, \boldsymbol{\delta}_{N,i}\} \tag{7.116}$$

From relations (7.109) and (7.114), we obtain

$$\tilde{\boldsymbol{q}}_{i+1} = -\gamma \boldsymbol{\delta}_i + \beta \tilde{\boldsymbol{q}}_i^g \tag{7.117}$$

It remains to figure out a recursion for $\boldsymbol{\delta}_i$. To do this, we note that

$$\boldsymbol{x}_{k,i} - \boldsymbol{x}_{l,i} = -\tilde{\boldsymbol{x}}_{k,i} + \tilde{\boldsymbol{x}}_{l,i} \tag{7.118}$$

and that we can write

$$\boldsymbol{\delta}_i = \nabla_{\tilde{\boldsymbol{x}}_i} f(\tilde{\boldsymbol{x}}_i) \tag{7.119}$$

in terms of the gradient vector of the function:

$$f(\tilde{\boldsymbol{x}}_i) = \frac{1}{4} \sum_{k=1}^{N} \sum_{l \in \mathcal{N}_k \backslash \{k\}} \left(\|\tilde{\boldsymbol{x}}_{k,i} - \tilde{\boldsymbol{x}}_{l,i}\| - r \right)^2 \tag{7.120}$$

In addition, from (7.9) and the definitions of the error quantities, $\{\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{v}}_i\}$, we have

$$\tilde{\boldsymbol{x}}_{i+1} = \tilde{\boldsymbol{x}}_i + \triangle t \cdot \tilde{\boldsymbol{q}}_{i+1} \tag{7.121}$$

Using a first-order Taylor series approximation and (7.121), we obtain a recursion for $\boldsymbol{\delta}_i$:

$$\begin{aligned} \boldsymbol{\delta}_{i+1} &\approx \boldsymbol{\delta}_i + \mathcal{D} \cdot (\triangle t \cdot \tilde{\boldsymbol{q}}_{i+1}) \\ &= (I - \triangle t \cdot \gamma \mathcal{D}) \, \boldsymbol{\delta}_i + \triangle t \cdot \beta \mathcal{D} \tilde{\boldsymbol{q}}_i^g \end{aligned} \tag{7.122}$$

where $\mathcal{D} = \nabla^2_{\tilde{\boldsymbol{x}}_i} f(\tilde{\boldsymbol{x}}_i)$.

7.B.2.2 Near-Field Approximation

Under the assumptions we used to establish the recursions (7.117) and (7.122) in the far field, we conclude that, under certain conditions (see (7.136) and (7.139) further ahead), the agents of the network will move coherently along the direction to the target relative to the center of mass in both the mean and mean-square sense. Therefore, the agents eventually enter the near field of the target, where $\|w^\circ - \boldsymbol{x}_{k,i}\| < s$. In this case, we can simplify (7.12) by setting $h(\boldsymbol{w}_{k,i} - \boldsymbol{x}_{k,i}) \approx \boldsymbol{w}_{k,i} - \boldsymbol{x}_{k,i}$. We also ignore the influence of the SNR levels by setting $\alpha = 0$. Then, the velocity update (7.18) simplifies to

$$\begin{aligned}\boldsymbol{q}_{k,i+1} &\approx \lambda \cdot (\boldsymbol{w}_{k,i} - \boldsymbol{x}_{k,i}) + \beta \boldsymbol{q}_{k,i}^g + \gamma \boldsymbol{\delta}_{k,i} \\ &= \underbrace{\lambda \tilde{\boldsymbol{x}}_i^g + \beta \boldsymbol{q}_i^g}_{\text{Center of mass}} + \underbrace{\lambda \tilde{\boldsymbol{x}}_{k,i} + \gamma \boldsymbol{\delta}_{k,i}}_{\text{Internal disturbances}} - \underbrace{(\lambda \tilde{\boldsymbol{w}}_{k,i} + \beta \tilde{\boldsymbol{q}}_{k,i}^g)}_{\text{Estimation errors}}\end{aligned} \tag{7.123}$$

It is seen that the velocity vector consists of three components. The first component relates to the motion of the center of mass (its velocity and its location relative to the target). This factor helps the center of mass approach the target. The second component in (7.123) refers to internal disturbances in the positioning of the agents (their locations relative to each other and relative to the center of mass). These sums reflect a competing trend. The term $\lambda \tilde{\boldsymbol{x}}_{k,i}$ represents the desire by agent k to get to the center of mass, which can also be interpreted as the desire to approach the target when the center of mass is close to the target, while the second term $\gamma \boldsymbol{\delta}_{k,i}$ reflects the desire by this same agent to maintain a safe distance from its neighbors. For notation simplicity, we introduce

$$\tilde{\boldsymbol{\delta}}_{k,i} = \lambda \tilde{\boldsymbol{x}}_{k,i} + \gamma \boldsymbol{\delta}_{k,i} \tag{7.124}$$

and its corresponding network vector

$$\tilde{\boldsymbol{\delta}}_i = \lambda \tilde{\boldsymbol{x}}_i + \gamma \boldsymbol{\delta}_i. \tag{7.125}$$

Finally, the third term in (7.123) is caused by the errors in estimating w° and tracking q_i^g. Using

$$\sum_{k=1}^{N} \tilde{\boldsymbol{\delta}}_{k,i} = \mathbf{0} \tag{7.126}$$

and approximations (7.112) and

$$\frac{1}{N}\sum_{k=1}^{N} \tilde{\boldsymbol{w}}_{k,i} \approx \mathbf{0} \tag{7.127}$$

we can obtain from (7.123) a recursion for $\boldsymbol{q}_i^g$ in near field as follows:

$$\boldsymbol{q}_{i+1}^g \approx \beta \boldsymbol{q}_i^g + \lambda \tilde{\boldsymbol{x}}_i^g \tag{7.128}$$

Note that since the center of mass is close to the target, under certain conditions (see (7.168)), both $\tilde{\boldsymbol{x}}_i^g$ and $\boldsymbol{q}_i^g$ will be close to zero vector in the mean. Therefore, it is also reasonable to apply the approximation $\boldsymbol{q}_{i+1}^g \approx \boldsymbol{q}_i^g$ in the near field. To obtain a recursion for $\tilde{\boldsymbol{q}}_i$, from relations (7.123) and (7.128), we obtain

$$\tilde{\boldsymbol{q}}_{i+1} = -\tilde{\boldsymbol{\delta}}_i + \lambda \tilde{\boldsymbol{w}}_i + \beta \tilde{\boldsymbol{q}}_i^g \tag{7.129}$$

That is, under the assumed approximations, the velocity of agent k relative to the center of mass is totally determined by its own internal disturbance and estimation errors. Moreover, a recursion for $\tilde{\boldsymbol{\delta}}_i$ can be obtained from (7.121) and (7.122) as follows:

$$\tilde{\boldsymbol{\delta}}_{i+1} = \tilde{\boldsymbol{\delta}}_i + \triangle t \left(\lambda I + \gamma \mathcal{D}\right) \tilde{\boldsymbol{q}}_{i+1} \tag{7.130}$$

7.B.3 State-Space Model in the Far Field

7.B.3.1 Mean Stability

We are now able to collect the results into a state-space model for the far field. Introduce the state vector:

$$\tilde{\boldsymbol{y}}_i \triangleq \mathrm{col}\{\tilde{\boldsymbol{q}}_i^g, \boldsymbol{\delta}_i\}$$

Using (7.105), (7.117), and (7.122), we find that $\tilde{\boldsymbol{y}}_i$ satisfies the state-space model:

$$\tilde{\boldsymbol{y}}_i = \mathcal{A}_y \cdot \tilde{\boldsymbol{y}}_{i-1} + \boldsymbol{v}_{y,i} \tag{7.131}$$

where

$$\mathcal{A}_y = \begin{bmatrix} \mathcal{G} & -\gamma \cdot \mathcal{H} \\ \triangle t \cdot \beta\mathcal{D} & I - \triangle t \cdot \gamma\mathcal{D} \end{bmatrix} \tag{7.132}$$

$$\mathcal{G} \triangleq \mathcal{A}_q{}^T(I - \mathcal{M}_q) + \beta\mathcal{A}_q{}^T\mathcal{M}_q\mathcal{C}^T \tag{7.133}$$

$$\mathcal{H} \triangleq \mathcal{A}_q{}^T\mathcal{M}_q\mathcal{C}^T \tag{7.134}$$

and the noise term $\boldsymbol{v}_{y,i}$ is used to capture the error introduced by the approximations and is assumed to be zero mean. Note from the last row that the coefficient matrix $\mathcal{A}_y$ depends on the data through the dependence of the entries in this row on $\tilde{\boldsymbol{x}}_i$ (through the Hessian matrix $\mathcal{D}$). Nevertheless, the factor $\mathcal{D}$ appears multiplied by $\triangle t$ and γ or β. For sufficiently small parameters, the influence of this data dependence can be neglected. Therefore, we shall assume that the matrix $\mathcal{A}_y$ is independent of the error vector $\tilde{\boldsymbol{y}}_{i-1}$; this condition is reminiscent of the independence assumption in the adaptive filtering literature, which has been shown to match well with practical results for sufficiently small step-sizes in the adaptation process for stand-alone filters [123]. Under this condition, we take expectations of both sides of (7.131) and get:

$$\mathbb{E}[\tilde{\boldsymbol{y}}_i] = \mathbb{E}[\mathcal{A}_y] \cdot \mathbb{E}[\tilde{\boldsymbol{y}}_{i-1}] \tag{7.135}$$

We conclude that, under the aforementioned independence assumption, the network is stable in the mean (i.e., $\mathbb{E}\tilde{\boldsymbol{q}}_{k,i}^q$ and $\mathbb{E}\boldsymbol{\delta}_{k,i}$ converge to zero) if the parameters $\{\nu_k, \beta, \gamma\}$ satisfy

$$\rho\left(\mathbb{E}[\mathcal{A}_y]\right) < 1 \tag{7.136}$$

in terms of the spectral radius of $\mathbb{E}[\mathcal{A}_y]$. Although the evolutions of the state variables in $\tilde{\boldsymbol{y}}_i$ are intertwined, an approximate way to set the adaptation parameters to satisfy (7.136) is to choose $\{\nu_k, \beta, \gamma\}$ such that the diagonal components of the matrix $\mathcal{A}_y$ have spectral radii less than one, i.e.,

$$\rho\left(\mathcal{G}\right) < 1 \quad \text{and} \quad \rho\left(I - \triangle t \cdot \gamma \mathcal{D}\right) < 1 \tag{7.137}$$

The rationale is that the off-diagonal submatrix $-\gamma \cdot \mathcal{H}$ in $\mathcal{A}_y$ may be ignored when ν_k and γ are small. Then, the matrix $\mathcal{A}_y$ becomes approximately lower block triangular. Now, recall that the norm $\|A\|_\infty$ denotes the maximum absolute row sum of a matrix and that it holds that $\rho(A) \leq \|A\|_\infty$ for any A. Using the triangular inequality of norms and the sub-multiplicative property of the ∞-norm, we have

$$\begin{aligned}
\rho\left(\mathcal{G}\right) &\leq \|\mathcal{A}_q{}^T(I - \mathcal{M}_q) + \beta \mathcal{A}_q{}^T \mathcal{M}_q \mathcal{C}^T\|_\infty \\
&\leq \|\mathcal{A}_q{}^T(I - \mathcal{M}_q)\|_\infty + \|\beta \mathcal{A}_q{}^T \mathcal{M}_q \mathcal{C}^T\|_\infty \\
&\leq \|\mathcal{A}_q{}^T\|_\infty \left(\|(I - \mathcal{M}_q)\|_\infty + |\beta| \|\mathcal{M}_q\|_\infty \|\mathcal{C}^T\|_\infty\right) \\
&= \|(I - \mathcal{M}_q)\|_\infty + |\beta| \|\mathcal{M}_q\|_\infty
\end{aligned} \tag{7.138}$$

where we used property $\|\mathcal{A}_q{}^T\|_\infty = \|\mathcal{C}^T\|_\infty = 1$ in the last equality. Therefore, condition (7.137) is satisfied if

$$|1 - \nu_{\min}| + |\beta \nu_{\max}| < 1 \tag{7.139}$$

where $\nu_{\min} = \min\{\nu_k\}$ and $\nu_{\min} = \max\{\nu_k\}$. To derive an approximate condition on γ that ensures the convergence of $\boldsymbol{\delta}_i$ to zero we argue as follows. Observe that under the assumptions $\|\tilde{\boldsymbol{x}}_i^g\| \gg \|\tilde{\boldsymbol{x}}_{k,i}\|$ and $\|\tilde{\boldsymbol{x}}_i^g\| \gg \|\tilde{\boldsymbol{w}}_{k,i}\|$, the noise term $\boldsymbol{v}_{y,i}$ in (7.131) is close to zero, and so is $\tilde{\boldsymbol{q}}_i^g$ in steady state. Thus, we can ignore the term $\tilde{\boldsymbol{q}}_i^g$ in (7.117) and expression (7.121) can be approximately written as

$$\tilde{\boldsymbol{x}}_{i+1} \approx \tilde{\boldsymbol{x}}_i - \triangle t \cdot \gamma \boldsymbol{\delta}_i \tag{7.140}$$

Now consider the function $f(\tilde{\boldsymbol{x}}_i)$ in (7.120). By Taylor series expansion, we have

$$f(\tilde{\boldsymbol{x}}_{i+1}) \approx f(\tilde{\boldsymbol{x}}_i) + \nabla_{\tilde{\boldsymbol{x}}_i} f(\tilde{\boldsymbol{x}}_i)^T (\tilde{\boldsymbol{x}}_{i+1} - \tilde{\boldsymbol{x}}_i) + \frac{1}{2}(\tilde{\boldsymbol{x}}_{i+1} - \tilde{\boldsymbol{x}}_i)^T \mathcal{D} (\tilde{\boldsymbol{x}}_{i+1} - \tilde{\boldsymbol{x}}_i) \quad (7.141)$$

Using the fact that $\mathcal{D} \leq NI$ from Appendix 7.C and (7.119), we obtain the following inequality

$$f(\tilde{\boldsymbol{x}}_{i+1}) - f(\tilde{\boldsymbol{x}}_i) \leq \triangle t \cdot \gamma \left(-1 + \frac{1}{2} \triangle t \cdot \gamma N \right) \|\boldsymbol{\delta}_i\|^2 \quad (7.142)$$

If the parameter γ is chosen such that

$$0 < \triangle t \cdot \gamma N < 2 \quad (7.143)$$

then $f(\tilde{\boldsymbol{x}}_i)$ is a decreasing sequence. But since $f(\tilde{\boldsymbol{x}}_i) \geq 0$, we conclude that $f(\tilde{\boldsymbol{x}}_i)$ is a convergent sequence and $\boldsymbol{\delta}_i$ will be close to zero in steady state.

From the previous discussion, we find that the state vector $\tilde{\boldsymbol{y}}_i$ converge to zero in the mean. From (7.117), we conclude that the velocity vector for each agent converge to $\boldsymbol{q}_i^g$ in the mean as well. We then arrive at the following result.

Theorem 7.2 (Far-field convergence)**.** *Assume the agents are in the far-field so that*

$$\|w^\circ - \boldsymbol{x}_{k,i}\| \gg s \quad \textit{and} \quad \|w^\circ - \boldsymbol{x}_i^g\| \gg \|\boldsymbol{x}_i^g - \boldsymbol{x}_{k,i}\| \quad (7.144)$$

for all k. If the parameters $\{\nu_k, \beta, \gamma\}$ are selected to satisfy

1. $|1 - \nu_{\min}| + |\beta \nu_{\max}| < 1$
2. $0 < \Delta t \cdot \gamma N < 2$

then it holds that

$$\mathbb{E}[\boldsymbol{q}_i^g - \boldsymbol{q}_{k,i}^g] \to 0 \quad \textit{and} \quad \mathbb{E}[\boldsymbol{q}_i^g - \boldsymbol{q}_{k,i}] \to 0 \quad (7.145)$$

for all k where $\boldsymbol{q}_i^g$ has the form

$$\boldsymbol{q}_{i+1}^g \approx \frac{\lambda s}{1-\beta} \frac{w^\circ - \boldsymbol{x}_i^g}{\|w^\circ - \boldsymbol{x}_i^g\|} \quad (7.146)$$

The above result verifies that in the far-field, the agents achieve coherent motion and they moves along the direction to the target relative to the center of mass. In the following, we proceed to evaluate the mean-square performance for error quantities in order to assess how close to zero the errors get.

7.B.3.2 Mean-Square Performance for $\tilde{\boldsymbol{q}}_i^g$ and $\tilde{\boldsymbol{q}}_i$

To examine the mean-square performance of the algorithm, we rely on the derivations in Section 2.2.3. In the following, we argue that $\boldsymbol{q}_{k,i}^g$ and $\boldsymbol{q}_{k,i}$ converge to the direction of the target from the center of mass of the network in the mean-square sense and derive expressions for the resulting mean-square errors. From (7.105) and (7.117), we rewrite the recursion for $\tilde{\boldsymbol{q}}_i^g$ as:

$$\tilde{\boldsymbol{q}}_i^g \approx \mathcal{G}\tilde{\boldsymbol{q}}_{i-1}^g - \gamma\mathcal{H}\boldsymbol{\delta}_{i-1} \tag{7.147}$$

Now, we start from (7.147) and verify that the following relation holds under expectation (see also [34]):

$$\mathbb{E}\|\tilde{\boldsymbol{q}}_i^g\|_\sigma^2 = \mathbb{E}\|\tilde{\boldsymbol{q}}_{i-1}^g\|_{\mathcal{F}_q\sigma}^2 + [\mathrm{vec}(-2\gamma\mathcal{G}\Gamma_{i-1}\mathcal{H}^T + \gamma^2\mathcal{H}\Pi_{i-1}\mathcal{H}^T)]^T\sigma \tag{7.148}$$

where $\Gamma_i = E\tilde{\boldsymbol{q}}_i^g\boldsymbol{\delta}_i^T$, $\Pi_i = E\boldsymbol{\delta}_i\boldsymbol{\delta}_i^T$ and

$$\mathcal{F}_q = \mathcal{G}^T \otimes \mathcal{G}^T \tag{7.149}$$

Note that we are considering the cross covariance matrix Γ_i because $\tilde{\boldsymbol{q}}_i^g$ and $\boldsymbol{\delta}_i$ are intertwined. We examine the steady-state velocity mean-square-error (MSE) for large time $\hat{i}$. Then it is reasonable to assume that the matrices $\Gamma_{\hat{i}}$ and $\Pi_{\hat{i}}$ converge to constant matrices Γ and Π, respectively, and so does $\mathbb{E}\|\tilde{\boldsymbol{q}}_{\hat{i}}^g\|_\sigma^2$. Moreover, assume the step sizes $\{\nu_k, \beta\}$ are chosen such that $\rho(\mathcal{G}) < 1$ (see (7.139)), then the network MSE for tracking $\boldsymbol{q}_i^g$ can be obtained by setting $\sigma = (I - \mathcal{F}_q)^{-1} \cdot \mathrm{vec}(I_{NM})$

in (7.148), i.e.,

$$\begin{aligned}\text{MSE}_q &\triangleq \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|\boldsymbol{q}_{\hat{i}}^g - \boldsymbol{q}_{k,\hat{i}}^g\|^2 \\ &= \frac{1}{N}[\text{vec}(-2\gamma\mathcal{G}\Gamma\mathcal{H}^T + \gamma^2\mathcal{H}\Pi\mathcal{H}^T)]^T(I-\mathcal{F}_q)^{-1}\cdot\text{vec}(I_{NM})\end{aligned} \tag{7.150}$$

We proceed to evaluate the mean-square performance for $\tilde{\boldsymbol{q}}_i$, which measures the coherence of collective motion of the network. We start from (7.117) and consider the error quantity, $\tilde{\boldsymbol{q}}_{k,i}$, of agent k. The following relation holds under expectation

$$\mathbb{E}\|\tilde{\boldsymbol{q}}_{k,i}\|^2 = \beta^2\mathbb{E}\|\tilde{\boldsymbol{q}}_{k,i}^g\|^2 + \gamma^2\mathbb{E}\|\boldsymbol{\delta}_{k,i}\|^2 - 2\beta\gamma\mathbb{E}[\boldsymbol{\delta}_{k,i}^T\tilde{\boldsymbol{q}}_{k,i}^g] \tag{7.151}$$

We introduce the network mean-square disagreement, D_q, on velocity:

$$\begin{aligned}D_q &\triangleq \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|\boldsymbol{q}_{\hat{i}}^g - \boldsymbol{q}_{k,\hat{i}}\|^2 \\ &= \beta^2\text{MSE}_q + \frac{1}{N}\left[\gamma^2\text{Tr}(\Pi) - 2\beta\gamma\text{Tr}(\Gamma)\right]\end{aligned} \tag{7.152}$$

Expressions (7.150) and (7.152) measure how well the network performs in tracking and moving coherently with the velocity of the center of mass, $\boldsymbol{q}_i^g$. These expressions capture the influence of various components on network performance.

7.B.4 State-Space Model in the Near Field

7.B.4.1 Mean Stability

The state-space model in the near field can be derived in a similar manner. Introduce the state vector:

$$\tilde{\boldsymbol{z}}_i \triangleq \text{col}\{\tilde{\boldsymbol{w}}_i, \tilde{\boldsymbol{q}}_i^g, \tilde{\boldsymbol{\delta}}_i\} \tag{7.153}$$

Using (7.99), (7.105), (7.129), and (7.130), we find that $\tilde{\boldsymbol{z}}_i$ satisfies the state-space model:

$$\tilde{\boldsymbol{z}}_i = \mathcal{A}_z\cdot\tilde{\boldsymbol{z}}_{i-1} + \boldsymbol{v}_{z,i} \tag{7.154}$$

where

$$\mathcal{A}_z = \begin{bmatrix} \mathcal{A}^T(I-\mathcal{M}) & 0 & 0 \\ \lambda\mathcal{H} & \mathcal{G} & -\mathcal{H} \\ \triangle t\cdot\mathcal{D}_\lambda & \triangle t\cdot\beta\mathcal{D}_\lambda & I-\triangle t\cdot\mathcal{D}_\lambda \end{bmatrix} \quad \text{and} \quad \boldsymbol{v}_{z,i} = \begin{bmatrix} -\mathcal{A}^T\mathcal{M}\boldsymbol{s}_i \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{7.155}$$

and $\mathcal{D}_\lambda = \lambda I + \gamma\mathcal{D}$. Using the same argument and for sufficiently small step-sizes, we assume that the matrix $\mathcal{A}_z$ is independent of the error vector $\tilde{\boldsymbol{z}}_{i-1}$. We then take expectations of both sides of (7.154) and get:

$$\mathbb{E}[\tilde{\boldsymbol{z}}_i] = \mathbb{E}[\mathcal{A}_z]\cdot\mathbb{E}[\tilde{\boldsymbol{z}}_{i-1}] \tag{7.156}$$

We conclude that the network is stable in the mean (i.e., $\mathbb{E}\tilde{\boldsymbol{w}}_{k,i}$, $\mathbb{E}\tilde{\boldsymbol{q}}^q_{k,i}$, and $\mathbb{E}\tilde{\boldsymbol{\delta}}_{k,i}$ converge to zero) if the parameters $\{\mu_k, \nu_k, \lambda, \beta, \gamma\}$ satisfy

$$\rho\left(\mathbb{E}[\mathcal{A}_z]\right) < 1 \tag{7.157}$$

If we examine (7.99), we observe that the recursion for $E\tilde{\boldsymbol{w}}_i$ does not depend on the other error quantities that appear in the state-vector $\tilde{\boldsymbol{z}}_i$. Therefore, $E\tilde{\boldsymbol{w}}_i$ converges to zero if

$$\rho\left(\mathcal{A}^T(I-\mathcal{M})\right) < 1 \tag{7.158}$$

which is satisfied by setting

$$0 < \mu_k < 2 \tag{7.159}$$

for all k [34]. Again, to deal with the dependence between the other two state variables in $\tilde{\boldsymbol{z}}_i$, we approximate the matrix $\mathcal{A}_z$ to a block lower triangular matrix since $\mathcal{H}$ in (7.134) can be ignored for small step-sizes. An approximate way to satisfy (7.156) is to choose $\{\nu_k, \lambda, \beta, \gamma\}$ such that

$$\rho\left(\mathcal{G}\right) < 1 \quad \text{and} \quad \rho\left(I - \triangle t(\lambda I + \gamma\mathcal{D})\right) < 1 \tag{7.160}$$

The first condition has been discussed before. For the second condition in (7.160), we use the same technique in the far field. Since the agents are close to the target in the near field, according to the noise model in (7.29), the noise term $\boldsymbol{v}_{z,i}$ in (7.154) will become small. Therefore, the error quantities $\{\tilde{\boldsymbol{w}}_i, \tilde{\boldsymbol{q}}_i^g\}$ will be closer to zero in steady state. From (7.129), expression (7.121) can be approximately written as

$$\tilde{\boldsymbol{x}}_{i+1} \approx \tilde{\boldsymbol{x}}_i - \triangle t \cdot \tilde{\boldsymbol{\delta}}_i \tag{7.161}$$

Now we introduce the function

$$h(\tilde{\boldsymbol{x}}_i) \triangleq \frac{\lambda}{2}\|\tilde{\boldsymbol{x}}_i\|^2 + \gamma f(\tilde{\boldsymbol{x}}_i) \tag{7.162}$$

and note that

$$\nabla_{\tilde{\boldsymbol{x}}_i} h(\tilde{\boldsymbol{x}}_i) = \tilde{\boldsymbol{\delta}}_i. \tag{7.163}$$

By Taylor series expansion and the fact that $\mathcal{D} \leq NI$, we obtain

$$h(\tilde{\boldsymbol{x}}_{i+1}) - h(\tilde{\boldsymbol{x}}_i) \leq \left[-1 + \frac{1}{2}(\lambda + \gamma N)\right] \|\tilde{\boldsymbol{\delta}}_i\|^2 \tag{7.164}$$

Then, based on same argument in (7.142) in the far field, $\tilde{\boldsymbol{\delta}}_i$ is close to zero in steady state if the parameters $\{\lambda, \gamma\}$ are chosen such that

$$\triangle t(\lambda + \gamma N) < 2 \tag{7.165}$$

The previous analysis focused on error quantities at each agent. We now investigate how the velocity and location of the center of mass evolve over time. Relation (7.128) shows the evolution of the velocity of the center of mass. This recursion depends on the error vector $\tilde{\boldsymbol{x}}_i^g$, whose recursion can be derived as follows:

$$\begin{aligned} \tilde{\boldsymbol{x}}_{i+1}^g &= \tilde{\boldsymbol{x}}_i^g - \triangle t \cdot \boldsymbol{q}_{i+1}^g \\ &= (1 - \triangle t \cdot \lambda)\tilde{\boldsymbol{x}}_i^g - \triangle t \cdot \beta \boldsymbol{q}_i^g \end{aligned} \tag{7.166}$$

We then can write down a state-space model for $\boldsymbol{q}_i^g$ and $\tilde{\boldsymbol{x}}_i^g$:

$$\begin{bmatrix} \boldsymbol{q}_{i+1}^g \\ \tilde{\boldsymbol{x}}_{i+1}^g \end{bmatrix} = \begin{bmatrix} \beta I & \lambda I \\ -\triangle t \cdot \beta I & (1 - \triangle t \cdot \lambda) I \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i^g \\ \tilde{\boldsymbol{x}}_i^g \end{bmatrix} + \boldsymbol{v}_{g,i} \tag{7.167}$$

where $\boldsymbol{v}_{g,i}$ is an error term to compensate the approximation error and is assumed to be zero mean. Suppose that the parameters $\{\mu_k, \nu_k, \lambda, \beta, \gamma\}$ satisfy condition (7.157) in addition to the following condition:

$$\rho\left(\begin{bmatrix} \beta & \lambda \\ -\triangle t \cdot \beta & 1 - \triangle t \cdot \lambda \end{bmatrix}\right) < 1 \tag{7.168}$$

Then, $\boldsymbol{q}_i^g$ and $\tilde{\boldsymbol{x}}_i^g$ will converge to zero in the mean.

We summary the above results as follows.

Theorem 7.3 (Near-field convergence)**.** *Assume the agents are in the near-field so that*

$$\|w^\circ - \boldsymbol{x}_{k,i}\| < s \tag{7.169}$$

for all k. If the parameters $\{\mu_k, \nu_k, \lambda, \beta, \gamma\}$ are selected to satisfy

1. $0 < \nu_k < 2$ *for all* k
2. $|1 - \nu_{\min}| + |\beta \nu_{\max}| < 1$
3. $0 < \Delta t(\lambda + \gamma N) < 2$
4. $\rho\left(\begin{bmatrix} \beta & \lambda \\ -\triangle t \cdot \beta & 1 - \triangle t \cdot \lambda \end{bmatrix}\right) < 1$

then it holds that

$$\mathbb{E}[\boldsymbol{w}_{k,i}] \to w^\circ, \;\; \mathbb{E}[\boldsymbol{x}_i^g] \to w^\circ, \;\; \mathbb{E}[\boldsymbol{q}_{k,i}^g] \to 0, \;\; \textit{and} \;\; \mathbb{E}[\boldsymbol{q}_{k,i}] \to 0 \tag{7.170}$$

for all k.

Note that the result $\mathbb{E}[\boldsymbol{q}_{k,i}] \to 0$ establishes the second statement of Theorem 7.1 in discrete-time and with stochastic approximations. To obtain the first statement of Theorem 7.1, we further examine the internal disturbance $\tilde{\boldsymbol{\delta}}_{k,i}$ in (7.124). Since $\mathbb{E}[\tilde{\boldsymbol{\delta}}_{k,i}] \to 0$ and $\mathbb{E}[\boldsymbol{x}_i^g] \to w^\circ$, we obtain from (7.124) that

$$\mathbb{E}[\tilde{\boldsymbol{\delta}}_{k,i}] = \lambda \mathbb{E}[w^\circ - \boldsymbol{x}_{k,i}] + \gamma \sum_{l \in \mathcal{N}_k \backslash \{k\}} \mathbb{E}\delta(\boldsymbol{x}_{l,i} - \boldsymbol{x}_{k,i}) \to 0 \tag{7.171}$$

where the function $\delta(\cdot)$ is defined by (7.98). That is, we arrive at the discrete-time and stochastic version of (7.95).

7.B.4.2 Mean-Square Performance for Estimating w°

We can start from (7.99) and obtain the following relation:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|_\sigma^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 + [\mathrm{vec}(\mathcal{A}^T \mathcal{M} \mathcal{S}_i^T \mathcal{M} \mathcal{A})]^T \sigma \tag{7.172}$$

where the matrices $\mathcal{F}$ and $\mathcal{S}_i$ are given by

$$\mathcal{F} = [(I - \mathcal{M})\mathcal{A}] \otimes [(I - \mathcal{M})\mathcal{A}] \tag{7.173}$$

$$\mathcal{S}_i = \mathrm{diag}\,\{R_{\eta,1,i}, R_{\eta,2,i}, \ldots, R_{\eta,N,i}\} \tag{7.174}$$

In general, the noise covariance matrices, $\{R_{\eta,k,i}\}$, vary over time due to the motion of the network. However, since the state vector $\tilde{\boldsymbol{z}}_i$ will be close to zero in steady state, we see that the approximations in (7.112) and (7.127) are reasonably accurate and therefore the noise term $\boldsymbol{v}_{g,i}$ in (7.167) will also be small. Thus, the vectors $\{\boldsymbol{q}_i^g, \tilde{\boldsymbol{x}}_i^g\}$ will be close to zero in steady state. Recall the recursion for $\boldsymbol{q}_{k,i}$ in (7.123). Since each term determining $\boldsymbol{q}_{k,i}$ is close zero in steady state, so will $\boldsymbol{q}_{k,i}$. Therefore, we can assume that all agents almost stop and that the noise variances converge to some constant value, i.e., $R_{\eta,k,i} \to R_{\eta,k}$ as $i \to \infty$. Then we can assume that $\mathcal{S}_i \to \mathcal{S}$, a constant matrix:

$$\mathcal{S} = \mathrm{diag}\,\{R_{\eta,1}, R_{\eta,2}, \ldots, R_{\eta,N}\} \tag{7.175}$$

If condition (7.158) is satisfied (e.g., $0 < \mu_k < 2$ for all k), we obtain that the steady-state network MSD is given by

$$\text{MSD} = \frac{1}{N}[\text{vec}(\mathcal{A}^T\mathcal{M}\mathcal{S}^T\mathcal{M}\mathcal{A})]^T(I-\mathcal{F})^{-1}\cdot\text{vec}(I_{NM}) \tag{7.176}$$

We observe that the noise variances in (7.175) affect the MSD in a linear way. That is, the MSD decreases if the noise variances decrease. In the noise model (7.29), this is equivalent to saying that the MSD decreases if the distances from the agents to the target decrease. Therefore, as we argued before, if the network uses larger λ, the network size becomes smaller and the agents stay closer to the target compared to a network with smaller λ. We can conclude that the network with larger λ will achieve lower MSD.

7.B.5 Simulation Results

We compare the performance of the ATC different algorithm and the algorithm where the agents estimate w° individually without cooperation among them. The simulation parameters are set according to Section 7.4.1.1. For the no cooperation case we simply set $a_{l,k} = c_{l,k} = \delta_{lk}$ and

$$q_{k,i}^g = \frac{1}{n_k}\sum_{l\in\mathcal{N}_k} q_{l,i}. \tag{7.177}$$

That is, the velocity of the center mass is estimated by the average velocity in the neighborhood, as proposed in [161]. We also evaluate the ATC diffusion algorithm for different values of λ.

We illustrate the transient behavior of estimating q^g. The results are averaged over 100 independent experiments. We ignore the effect of noise variances by setting $\alpha = 0$. The network transient MSE for q^g is shown in Fig. 7.20. As the results show, the ATC network, except for low values of λ, has better MSE

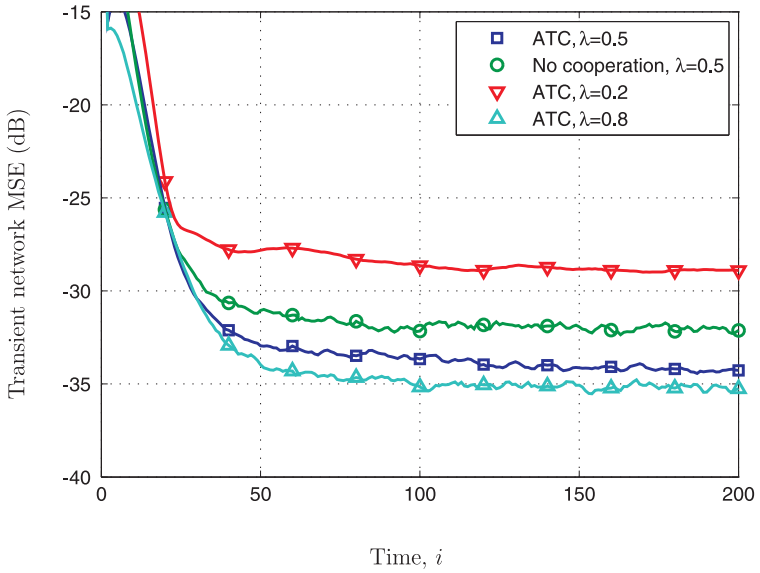


Figure 7.20: Transient network MSE for estimating the velocity of the center of mass v^g in the far field.

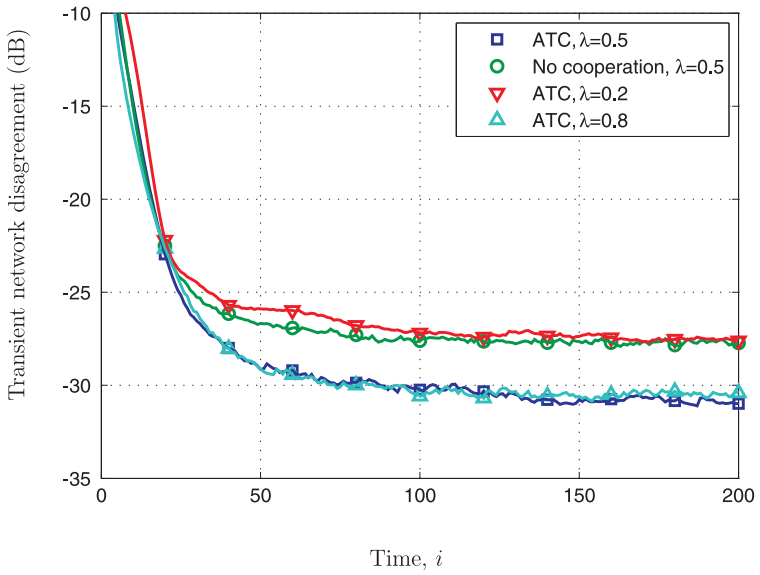


Figure 7.21: Transient network mean-square disagreement of the velocities in the far field.

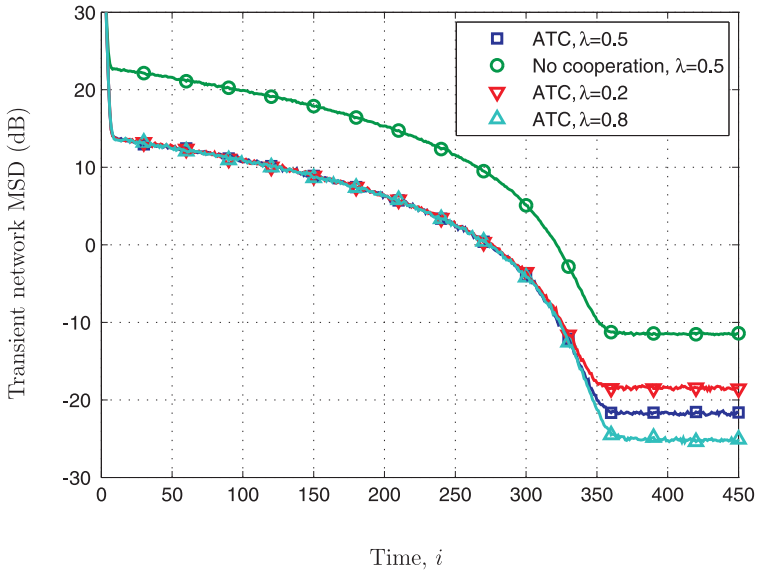


Figure 7.22: Transient network MSD for estimating the target location, w°.

than non-cooperative networks. We also see that for the ATC network, higher values of λ achieve lower MSE because approximation (7.114) is more accurate for higher values of λ. Figure 7.21 shows the network mean-square disagreement, D_v. We observe that the ATC diffusion strategy with high values of λ improves the steady-state performance (achieves better coherence) compared to the average strategy (no cooperation). That is, the ATC diffusion strategy helps the network form coherent movement.

Figure 7.22 shows the network transient MSD for estimating w°. There are three phases. The first phase is transient and the MSD decreases dramatically. In the second phase, the network moves towards the target and the noise variance decreases. Therefore, the MSD decreases accordingly. It is interesting to observe how the motion of the network affects the mean-square performance. In the third phase, the network arrives at the target and achieves steady state. As has been shown in [34], the ATC diffusion scheme has better performance. In steady state,

Table 7.1: Theoretical and simulation results of the network mean-square performance in steady state.

	MSE_q (7.150)	D_q (7.152)	MSD (7.176)
Theory (dB)	-32.94	-29.78	-23.22
Simulation (dB)	-32.86	-30.20	-22.93

when the network arrives at the target, the agents with higher values of λ stay closer due to more weight on approaching the target and the MSD is lower than the other cases as we expect.

We also show simulations to illustrate the theoretical results in Figs. 7.23 and 7.24 for the ATC diffusion algorithm with $\lambda = 0.5$. In Fig. 7.23, the error quantities $\tilde{q}^g_{k,i}$ and $\tilde{q}_{k,i}$ evolve according to expressions (7.147) and (7.117), respectively, while $\tilde{w}_{k,i}$ evolves according to (7.99). The theoretical results match simulations rather well. Moreover, we compare these results in steady state in table 7.1. The network mean-square errors MSE_q, D_q, and MSD are evaluated using (7.150), (7.152), and (7.176), respectively. We observe that in steady state theoretical results cohere with simulations.

7.C Bound on $\mathcal{D}$

Introduce the $M \times M$ matrix:

$$[D(y)]_{kl} = \begin{cases} 1 - r\frac{\sum_{l \neq k} y_l^2}{\|y\|^3}, & \text{if } k = l \\ r\frac{y_k y_l}{\|y\|^3}, & \text{otherwise} \end{cases} \tag{7.178}$$

where y is a $M \times 1$ vector. It is easy to verify that $D(y)$ has two distinct eigenvalues at 1 and $1 - r/\|y\|$ with multiplicity 1 and $M - 1$, respectively. Moreover, the

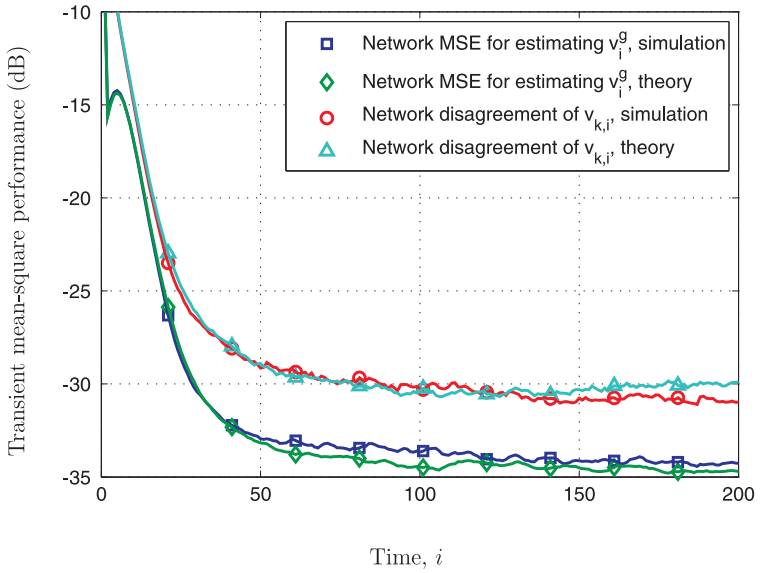


Figure 7.23: Transient network mean-square performance in the far field, comparing simulation and theory.

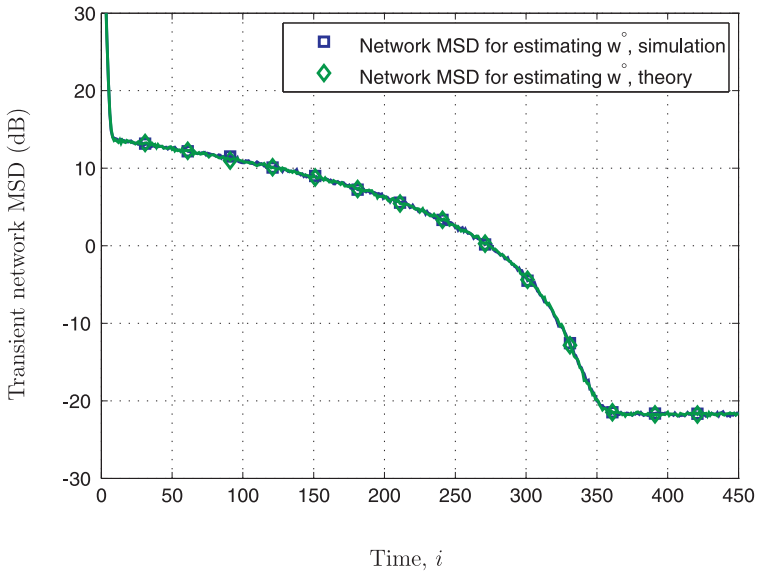


Figure 7.24: Transient network MSD for estimating the target location, w°, comparing simulation and theory.

eigenvector corresponding to eigenvalue 1 is the vector y and the eigenvectors associated with $1 - r/\|y\|$ are $\begin{bmatrix} y_i & 0 & \cdots & 0 & -y_1 & 0 & \cdots & 0 \end{bmatrix}^T$ for $i = 2, \ldots, M$ and $-y_1$ in the ith element.

The (k, l)th block of the Hessian $\mathcal{D}$ can be expressed as follows:

$$\mathcal{D}_{kl} = \begin{cases} -D(\tilde{x}_k - \tilde{x}_l) & \text{if } l \in \mathcal{N}_k \setminus \{k\} \\ \sum_{l \in \mathcal{N}_k \setminus \{k\}} D(\tilde{x}_k - \tilde{x}_l) & \text{if } l = k \\ \mathbf{0}_{M \times M} & \text{otherwise} \end{cases} \tag{7.179}$$

Introduce the Laplacian matrix of a network:

$$L_{kl} = \begin{cases} n_k - 1 & \text{if } l = k \\ -1 & \text{if } l \in \mathcal{N}_k \setminus \{k\} \\ 0 & \text{otherwise} \end{cases} \tag{7.180}$$

It is well-known that $L \leq NI$. In the following we show that $\mathcal{D} \leq L \otimes I_M$, which results in $\mathcal{D} \leq NI$. Let $\pi = \begin{bmatrix} \pi_1 & \cdots & \pi_N \end{bmatrix}^T$ with $\pi_k \in \mathbb{R}^M$. We have

$$\begin{aligned} \pi^T (L \otimes I_M - \mathcal{D}) \pi &= \sum_{k=1}^{N} \left[\pi_k^T \sum_{l \in \mathcal{N}_k \setminus \{k\}} (I + \mathcal{D}_{kl}) \pi_k - \pi_k^T \sum_{l \in \mathcal{N}_k \setminus \{k\}} (I + \mathcal{D}_{kl}) \pi_l \right] \\ &= -\frac{1}{2} \sum_{k=1}^{N} \sum_{l=1}^{N} L_{kl} (\pi_k - \pi_l)^T (I + \mathcal{D}_{kl}) (\pi_k - \pi_l) \end{aligned} \tag{7.181}$$

For $k \neq l$, we have $I + \mathcal{D}_{kl} = I - D(\tilde{x}_k - \tilde{x}_l) \geq \mathbf{0}$ since the eigenvalues of the matrix D are less than or equal to one.

CHAPTER 8

Future Issues

In this dissertation, we addressed several important issues in the design of distributed adaptation strategies. In particular, we verified that more information is not necessarily better and the way by which information is processed and propagated through the network matters: small variations can lead to catastrophic failures. These results suggest that cooperation through diffusion strategies is the proper way to process data. This is an important conclusion, especially in relation to the modeling of biological networks. For example, in fish schools, the network topology and the combination weights evolve over time due to the motion of the fish and the interaction between the fish school and predators. In the following, we list several potential topics for future exploration:

- One of the assumptions throughout this work is that the regression covariance matrices are positive definite so that each individual node is able to solve the problem on its own. However, there are cases, such as [1, 73], where nodes only have access to partial information about the unknown. In this situation, nodes need to cooperate and exchange information to arrive at the full estimate. An important issue would be to study the conditions on network topology and observability so that nodes are able to solve estimation tasks collaboratively under these conditions.

- Cooperation among nodes relies on the fact that the nodes are willing to

cooperate. In many situations, this is not the case. For example, when nodes are selfish or competing with each other, they need to be enticed to cooperate for the common good. It is important to develop strategies that allow selfish agents to work together for the social benefit for the network.

- In this work, we mainly considered mean-square error cost functions. However, the cost of information sharing and the cost for data access may be be demanding in some cases involving sensor nodes with limited capabilities. Therefore, it is desired for the nodes to equipped with energy-aware adaptation strategies. In this situation, the nodes may only share partial information or only maintain a fraction of connections that can provide reasonable performance.

- There have been extensive works on social networks [27,169] and epidemiological networks [163]. These networks involve distributed signal processing as well. Examples include how a crowd of people forms opinion and how disease spreads over networks. It will be useful to apply adaptation strategies to model these networks and to study their behavior.

- Adaptation of the network topology is a critical element in mobile networks. It adds one more degree of freedom. It is important to develop techniques that examine the co-evolution of the learning process and the changing topology. Examples from biological networks abound.

References

[1] R. Abdolee, B. Champagne, and A. H. Sayed. Diffusion LMS for source and process estimation in sensor networks. *Proc. IEEE Workshop on Statistical Signal Processing (SSP)*, pages 1–4, Ann Arbor, MI, Aug. 2012.

[2] T. Y. Al-Naffouri and A. H. Sayed. Transient analysis of data-normalized adaptive filters. *IEEE Trans. on Signal Processing*, 51(3):639–652, Mar. 2003.

[3] R. Albert and A.-L. Barabasi. Statistical mechanics of complex networks. *Rev. Modern Phys.*, 74:47–97, Jan. 2002.

[4] M. Andersson and J. Wallander. Kin selection and reciprocity in flight formation? *Behavioral Ecology*, pages 158–162, 2004.

[5] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed. Mean-square performance of a convex combination of two adaptive filters. *IEEE Trans. on Signal Processing*, 54(3):1078–1090, Mar. 2006.

[6] J. Arenas-Garcia, V. Gomez-Verdejo, and A. R. Figueiras-Vidal. New algorithms for improved adaptive convex combination of LMS transversal filters. *IEEE Trans. on Instrumentation and Measurement*, 54(6):2239–2249, Dec. 2005.

[7] A. Avitabile, R. A. Morse, and R. Boch. Swarming honey bees guided by pheromones. *Ann. Entomol. Soc. Am.*, 68:1079–1082, 1975.

[8] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Trans. on Signal Processing*, 57(7):2748–2761, July 2009.

[9] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[10] A. L. Barabasi and E. Bonabeau. Scale-free networks. *Scientific American*, 288:50–59, 2003.

[11] S. Barbarossa and G. Scutari. Bio-inspired sensor network design: Distributed decisions through self-synchronization. *IEEE Signal Processing Magazine*, 24:26–35, May 2007.

[12] M. Beekman, R. L. Fathke, and T. D. Seeley. How does an informed minority of scouts guide a honey bee swarm as it flies to its new home? *Animal Behavior*, 71:161–171, doi:10.1016/j.anbehav.2005.04.009, 2006.

[13] K. E. Bekris, K. Tsianos, and L. E. Kavraki. Safe and distributed kinodynamic replanning for vehicular networks. *Mobile Networks and Applications (MONET)*, 14:292–308, Feb. 2009.

[14] K. J. Benoit-Bird and W. W. L. Au. Cooperative prey herding by the pelagic dolphin, stenella longirostris. *Journal of Acoustical Society of America*, 125:125–137, Jan. 2009.

[15] A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations*. Springer-Verlag, NY, 1987.

[16] R. L. Berger. A necessary and sufficient condition for reaching a consensus using degroots method. *Journal of the American Statistical Association*, 76(374):118–121, 1981.

[17] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. 1994.

[18] D. P. Bertsekas. A new class of incremental gradient methods for least squares problems. *SIAM J. Optim.*, 7(4):913–926, 1997.

[19] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Singapore, 1997.

[20] B. Bollobas. *Random Graphs*. Cambridge University Press, 2001.

[21] A. Bovier and P. Picco. A law of the iterated logarithm for random geometric series. *The Annals of Probability*, 21(1):168–184, 1993.

[22] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. on Information Theory*, 52(6):2508–2530, Jun. 2006.

[23] N. F. Britton, N. R. Franks, S. C. Pratt, and T. D. Seeley. Deciding on a new home: How do honeybees agree? *Proc. R. Soc. Lond. B*, 269:1383–1388, May 2002.

[24] F. Bullo, J. Cortes, and S. Martinez. *Distributed Control of Robotic Networks: A Mathematial Approach to Motion Coordination Algorithms*. Princeton University Press, 2009.

[25] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organization in Biological Systems*. Princeton University Press, 2003.

[26] R. Candido, M. T. M. Silva, and V. H. Nascimento. Transient and steady-state analysis of the affine combination of two adaptive filters. *IEEE Trans. on Signal Processing*, 58(8):4064–4078, Aug. 2010.

[27] C. Castellano, S. Fortunato, and V. Loreto. Statistical physics of social dynamics. *Rev. Modern Phys.*, 81:591–646, Apr.-Jun. 2009.

[28] F. Cattivelli and A. H. Sayed. Distributed detection over adaptive networks using diffusion adaptation. *IEEE Trans. on Signal Processing*, 59(5):1917–1932, May 2011.

[29] F. Cattivelli and A. H. Sayed. Modeling bird flight formations using diffusion adaptation. *IEEE Trans. on Signal Processing*, 59(5):2038–2051, May 2011.

[30] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed. Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Trans. on Signal Processing*, 56(5):1865–1877, May 2008.

[31] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed. Diffusion strategies for distributed kalman filtering: Formulation and performance analysis. *Proc. 2008 IAPR Workshop on Cognitive Information Processing*, pages 36–41, Santorini, Greece, Jun. 2008.

[32] F. S. Cattivelli and A. H. Sayed. Diffusion LMS algorithms with information exchange. *Proc. Asilomar Conference on Signals, Systems and Computers*, pages 251–255, Pacific Grove, CA, Oct. 2008.

[33] F. S. Cattivelli and A. H. Sayed. Diffusion distributed Kalman filtering with adaptive weights. *Proc. Asilomar Conference on Signals, Systems and Computers*, pages 908–912, Pacific Grove, CA, Nov. 2009.

[34] F. S. Cattivelli and A. H. Sayed. Diffusion LMS strategies for distributed estimation. *IEEE Trans. on Signal Processing*, 58(3):1035–1048, Mar. 2010.

[35] F. S. Cattivelli and A. H. Sayed. Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Trans. on Automatic Control*, 55(9):2069–2084, Sep. 2010.

[36] J. F. Chamberland and V. V. Veeravalli. Wireless sensors in distributed detection applications. *IEEE Signal Processing Magazine*, 24:16–25, May 2007.

[37] J. Chen and A. H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. on Signal Processing*, 60(8):4289–4305, Aug. 2012.

[38] J. Chen and A. H. Sayed. On the limiting behavior of distributed optimization strategies. *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, Oct. 2012.

[39] J. Chen and A. H. Sayed. Distributed Pareto optimization via diffusion strategies. *IEEE J. Selected Topics in Signal Processing*, 7(2):205–220, Apr. 2013.

[40] J. Chen, X. Zhao, and A. H. Sayed. Bacterial motility via diffusion adaptation. *Proc. 44th Asilomar Conference on Signals, Systems and Computers*, pages 1930–1934, Pacific Grove, CA, Nov. 2010.

[41] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis. Sparsity-promoting adaptive algorithm for distributed learning in diffusion networks. *Proc. EUSIPCO*, pages 1084–1088, Bucharest, Romania, Aug. 2012.

[42] S. Chouvardas, K. Slavakis, and S. Theodoridis. Adaptive robust distributed learning in diffusion sensor networks. *IEEE Trans. on Signal Processing*, 59(10):4692–4707, Oct. 2011.

[43] F. Chung. *Spectral Graph Theory*. American Math. Soc., Providence, 1997.

[44] F. Chung, L. Li, and V. Vu. Spectra of random graphs with given expected degrees. *Proc. National Academy of Sciences*, 100(11):6313–6318, 2003.

[45] I. D. Couzin. Collective cognition in animal groups. *Trends in Cognitive Sciences*, 13:36–43, Jan. 2009.

[46] I. D. Couzin, C. C. Ioannou, G. Demirel, T. Gross, C. J. Torney, A. Hartnett, L. Conradt, S. A. Levin, and N. E. Leonard. Uninformed individuals promote democratic consensus in animal groups. *Science*, 332:1578–1580, 2011.

[47] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision making in animal groups on the move. *Nature*, 433:513–516, 2005.

[48] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218:1–11, 2002.

[49] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69:118–121, 1974.

[50] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, Nov. 2010.

[51] S. N. Dorogovtsev, A. V. Goltsev, J. F. F. Mendes, and A. N. Samukhin. Random networks: Eigenvalue spectra. *Physica A*, 338:76–83, 2004.

[52] P. Erdos and A. Renyi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutato Int. Kozl.*, 5:17–61, 1960.

[53] Jr. F. S. Hill and M. A. Blanco. Random geometric series and intersymbol interference. *IEEE Trans. on Information Theory*, 19(3):326–335, May 1973.

[54] I. J. Farkas, I. Derenyi, A. L. Barabasi, and T. Vicsek. Spectra of real-world graphs: Beyond the semicircle law. *Phys. Rev. E*, 64(026704), 2001.

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

[56] V. Gazi and K. M. Passino. Stability analysis of social foraging swarms. *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, 34:539–557, Feb. 2004.

[57] J. C. Gittins. *Multi-Armed Bandit Allocation Indices*. John Wiley and Sons, New York, NY, 1989.

[58] K. I. Goh, B. Kahng, and D. Kim. Spectra and eigenvectors of scale-free networks. *Phys. Rev. E*, 64(051903), 2001.

[59] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[60] W. D. Hamilton. Geometry for the selfish herd. *Journal of Theoretical Biology*, 31:295–311, 1971.

[61] Y. Hatano, A. K. Das, and M. Mesbahi. Agreement in presence of noise: Pseudogradients on random geometric networks. *Proc. of Conf. on Dec. and Contr.*, pages 6382–6387, Seville, Spain, Dec. 2005.

[62] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 2002.

[63] Y. W. Hong and A. Scaglione. A scalable synchronization protocol for large scale sensor networks and its applications. *IEEE J. Sel. Areas Commun.*, 23(5):1085–1099, May 2005.

[64] R. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[65] Y. Hu and A. Ribeiro. Adaptive distributed algorithms for optimal random access channels. *IEEE Trans. on Wireless Commun.*, 10(8):2703–2715, Aug. 2011.

[66] S. Hubbard, P. Babak, S. T. Sigurdsson, and K. G. Magnusson. A model of the formation of fish schools and migrations of fish. *Ecological Modeling*, 174:359–374, Jun. 2004.

[67] A. Huth and C. Wissel. The simulation of the movement of fish schools. *Journal of Theoretical Biology*, 156:365–385, Jun. 1992.

[68] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups for mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on Automatic Control*, 48:988–1001, Jun. 2003.

[69] D. Jakovetic, J. Xavier, and J. M. F Moura. Weight optimization for consenus algorithms with correlated switching topology. *IEEE Trans. on Signal Processing*, pages 3788–3801, Jul. 2010.

[70] S. Janson, M. Middendorf, and M. Beekman. Honeybee swarms: How do scouts guide a swarm of uninformed bees? *Anim. Behav*, 70:349–358, 2005.

[71] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. *Proc. IEEE CDC*, pages 4185–4190, Cancun, Mexico, Dec. 2008.

[72] S. Kar and J. M. F. Moura. Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise. *IEEE Trans. on Signal Processing*, 57(1):355–369, Jan. 2009.

[73] S. Kar and J. M. F. Moura. Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs. *IEEE J. Selected Topics in Signal Processing*, 5(5):674–690, Aug. 2011.

[74] S. Kar and J. M. F. Moura. Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication. *IEEE Trans. on Information Theory*, 58(6):3575–3605, Jun. 2012.

[75] R. M. Karp. Reducibility among combinational problems. *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher (Eds.), New York: Plenum, pages 85–104, 1972.

[76] Y. Katz, C. C. Ioannou, K. Tunstrom, C. Huepe, and I. D. Couzin. Inferring the structure and dynamics of interactions in schooling fish. *PNAS*, 108(46):18720–18725, Nov. 2011.

[77] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers. Steady-state analysis of diffusion LMS adaptive networks with noisy links. *IEEE Trans. on signal Processing*, 60(2):974–979, Feb. 2012.

[78] U. A. Khan, S. Kar, and J. M. F. Moura. Higher dimensional consensus: Learning in large-scale networks. *IEEE Trans. on Signal Processing*, 58(5):2836–2849, May 2010.

[79] J. H. Kotecha, V. Ramachandran, and A. M. Sayeed. Distributed multitarget classification in wireless sensor networks. *IEEE J. Select. Areas Commun.*, 23(4):703–713, Apr. 2005.

[80] S. S. Kozat and A. T. Erdogan. Competitive linear estimation under model uncertainties. *IEEE Trans. on Signal Processing*, 58(4):2388–2393, Apr. 2010.

[81] S. S. Kozat, A. C. Singer, A. T. Erdogan, and A. H. Sayed. Unbiased model combinations for adaptive filtering. *IEEE Trans. on Signal Processing*, 58(8):4421–4427, Aug. 2010.

[82] H. J. Kushner. *Approximation and Weak Convergence Methods for Random Processes with Applications to Stochastic System Theory*. MIT Press, Cambridge, MA, 1984.

[83] H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer, NY, 1997.

[84] A. J. Laub. *Matrix Analysis for Scientists and Engineers*. SIAM, PA, 2005.

[85] G. F. Lawler. *Introduction to Stochastic Processes*. Chapman & Hall/CRC, 2006.

[86] J. Li and A. H. Sayed. Modeling bee swarming behavior through diffusion adaptation with asymmetric information sharing. *EURASIP Journal on Advances in Signal Processing*, pages 1–14, doi:10.1186/1687-6180-2012-18, Jan. 2012.

[87] L. Li and J. A. Chambers. Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology. *Proc. IEEE Statist. Signal Process. Workshop*, pages 757–760, Cardiff, Wales, Sep. 2009.

[88] L. Li, C. G. Lopes, J. Chambers, and A. H. Sayed. Distributed estimation over an adaptive incremental network based on the affine projection algorithm. *IEEE Trans. Signal Processing*, 58(1):151–164, Jan. 2010.

[89] K. Liu and Q. Zhao. Distributed learning in multi-armed bandit with multiple players. *IEEE Trans. on Signal Processing*, 58(11):5667–5681, Nov. 2010.

[90] L. Ljung. Analysis of recursive stochastic algorithms. *IEEE Trans. on Autom. Control*, 22(4):551–575, Aug. 1977.

[91] L. Ljung. On positive real transfer functions and the convergence of some recursive schemes. *IEEE Trans. on Autom. Control*, 22(4):539–551, Aug. 1977.

[92] C. G. Lopes and A. H. Sayed. Distributed processing over adaptive networks. *Proc. Adaptive Sensor Array Processing Workshop*, pages 1–5, MIT Lincoln Laboratory, MA, Jun. 2006.

[93] C. G. Lopes and A. H. Sayed. Incremental adaptive strategies over distributed networks. *IEEE Trans. on Signal Processing*, 55(8):4064–4077, Aug. 2007.

[94] C. G. Lopes and A. H. Sayed. Diffusion adaptive networks with changing topologies. *Proc. IEEE ICASSP*, pages 3285–3288, Las Vegas, NV, Apr. 2008.

[95] C. G. Lopes and A. H. Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Trans. on Signal Processing*, 56(7):3122–3136, Jul. 2008.

[96] P. Di Lorenzo and S. Barbarossa. A bio-inspired swarming algorithm for decentralized access in cognitive radio. *IEEE Trans. on Signal Processing*, 59(12):6160–6174, Dec. 2011.

[97] P. Di Lorenzo and S. Barbarossa. Swarming algorithms for distributed radio resource allocation. *IEEE Signal Processing Magazine*, 30(3):144–154, May 2013.

[98] P. Di Lorenzo and A. H. Sayed. Sparse distributed learning based on diffusion adaptation. *IEEE Trans. on Signal Processing*, 61(6):1419–1433, Mar. 2013.

[99] H. Milinski and R. Heller. Influence of a predator on the optimal foraging behavior of sticklebacks. *Nature*, 275:642–644, 1978.

[100] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. on Automatic Control*, 50(2):169–182, Feb. 2005.

[101] V. H. Nascimento, M. Silva, R. Candido, and J. Arenas-Garcia. A transient analysis for the convex combination of adaptive filters. *Proc. IEEE Statist. Signal Process. Workshop*, pages 53–56, Cardiff, Wales, Sep. 2009.

[102] A. Nedic and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM J. Optim.*, 12(1):109–138, 2001.

[103] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multiagent optimization. *IEEE Trans. on Autom. Control*, 54(1):48–61, Jan. 2009.

[104] A. Nedic and A. Ozdaglar. Cooperative distributed multi-agent optimization. *Convex Optimization in Signal Processing and Communications*, Y. Eldar and D. Palomar (Eds.), Cambridge University Press, pages 340–386, 2009.

[105] M. E. J. Newman, D. J. Watts, and A. L. Barabasi. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[106] P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Trans. on Automatic Control*, 49(8):1292–1302, Aug. 2004.

[107] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. on Automatic Control*, 51(3):401–420, Mar. 2006.

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

[109] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control*, 49:1520–1533, Sep. 2004.

[110] A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 2002.

[111] B. L. Partridge. The structure and function of fish schools. *Scientific American*, 246:114–123, 1982.

[112] S. U. Pillai, T. Suel, and S. Cha. The Perro-Frobenius theorem: Some of its applications. *IEEE Signal Process. Mag.*, 22(2):62–75, Mar. 2005.

[113] S. C. Pratt, E. B. Mallon, D. J. T. Sumpter, and N. R. Franks. Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant Leptothorax albipennis. *Behav. Ecol. Sociobiol.*, 52:117–127, May 2002.

[114] J. B. Predd, S. B. Kulkarni, and H. V. Poor. Distributed learning in wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):56–69, Jul. 2006.

[115] M. G. Rabbat and R. D. Nowak. Quantized incremental algorithms for distributed optimization. *IEEE J. Sel. Areas Commun.*, 23(4):798–808, 2005.

[116] S.S. Ram, A. Nedic, and V. V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of Optim. Theory and Applic.*, 147(3):516–545, 2010.

[117] S. G. Reebs. Can a minority of informed leaders determine the foraging movements of a fish shoal? *Anim. Behav.*, 59:403–409, 2000.

[118] W. Ren and R. W. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Trans. on Automatic Control*, 50:655–661, May 2005.

[119] W. Ren, R. W. Beard, and E. M. Atkins. Information consensus in multivehicle cooperative control. *IEEE Control System Magazine*, 27(2):71–82, Apr. 2007.

[120] C. W. Reynolds. Flocks, herds, and schools: A distributed behavior model. *ACM Proc. Computer Graphs and Interactive Techniques*, pages 25–34, 1987.

[121] G. Roberts. Why individual vigilance increases as group size increases. *Animal Behavior*, 51:1077–1086, 1996.

[122] S. Sardellitti, M. Giona, and S. Barbarossa. Fast distributed average consensus algorithms based on advection-diffusion processes. *IEEE Trans. on Signal Processing*, pages 826–842, Feb. 2010.

[123] A. H. Sayed. *Adaptive Filters*. NJ. Wiley, 2008.

[124] A. H. Sayed. Diffusion adaptation over networks. *E-Reference Signal Processing*, R. Chellapa and S. Theodoridis, editors, Elsevier, 2013. Also available online at http://arxiv.org/abs/1205.4220, May 2012.

[125] A. H. Sayed and C. G. Lopes. Distributed recursive least-squares strategies over adaptive networks. *Proc. 40th Asilomar Conference on Signals, Systems and Computers*, pages 233–237, Pacific Grove, CA, Oct.-Nov. 2006.

[126] A. H. Sayed and C. G. Lopes. Adaptive processing over distributed networks. *IEICE Trans. on Fundamentals of Electronics,Communications and Computer Sciences*, E90-A(8):1504–1510, 2007.

[127] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, and Z. Towfic. Diffusion strategies for adaptation and learning over networks. *IEEE Signal Processing Magazine*, 30(3):155–171, May 2013.

[128] I. Schizas, A. Ribeiro, and G. Giannakis. Consensus in ad hoc WSNs with noisy links - part I: Distributed estimation of deterministic signals. *IEEE Trans. on Signal Processing*, 56(1):350–364, Jan. 2008.

[129] I. D. Schizas, G. Mateos, and G. B. Giannakis. Distributed LMS for consensus-based in-network adaptive processing. *IEEE Trans. on Signal Processing*, 57(6):2365–2381, June 2009.

[130] T. D. Seeley, R. A. Morse, and P. K. Visscher. The natural history of the flight of honey bee swarms. *Psyche*, 86:103–114, 1979.

[131] T. Simila and F. Ugarte. Surface and underwater observations of cooperatively feeding killer whales in northern Norway. *Canadian Journal of Zoology*, 71:1494–1499, 1993.

[132] P. J. Smith. The distribution functions of certain random geometric series concerning intersymbol interference. *IEEE Trans. on Information Theory*, 37(6):1657–1662, Nov. 1991.

[133] K. Srivastava and A. Nedic. Distributed asynchronous constrained stochastic optimization. *IEEE J. Selected Topics on Signal Processing*, 5(4):772–790, Aug. 2011.

[134] S. S. Stankovic, M. S. Stankovic, and D. M Stipanovic. Decentralized parameter estimation by consensus based stochastic approximation. *IEEE Trans. on Autom. Control*, 56(3):531–543, Mar. 2011.

[135] D. J. T. Sumpter. *Collective Animal Behavior*. Princeton University Press, 2010.

[136] D. J. T. Sumpter, J. Buhl, D. Biro, and I. D. Couzin. Information transfer in moving animal groups. *Theory in Biosciences*, 127(2):177–186, 2008.

[137] D. J. T. Sumpter and S. C. Pratt. A modelling framework for understanding social insect foraging. *Behav. Ecol. Sociobiol.*, pages 131–144, 2003.

[138] D. J. T. Sumpter and S. C. Pratt. Quorum responses and consensus decision making. *Phil. Trans. R. Soc. B*, 364:743–753, Dec. 2009.

[139] J. Surowiecki. *The Wisdom of Crowds*. Anchor Books, 2005.

[140] N. Takahashi and I. Yamada. Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks. *Proc. IEEE ICASSP*, pages 3518–3521, Dallas, TX, Mar. 2010.

[141] N. Takahashi, I. Yamada, and A. H. Sayed. Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis. *IEEE Trans. on Signal Processing*, 8(9):4795–4810, Sep. 2010.

[142] Z. Towfic, J. Chen, and A. H. Sayed. On the generalization ability of online learners. *Proc. IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Santander, Spain, Sep. 2012.

[143] Z. Towfic, J. Chen, and A. H. Sayed. On distributed online classification in the midst of concept drifts. *Neurocomputing*, 112:139–152, 2013.

[144] J. N. Tsitsiklis, J. N. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimzation algorithms. *IEEE Trans. on Autom. Control*, 31(9):803–812, Sep. 1986.

[145] S. Y. Tu and A. H. Sayed. Foraging behavior of fish schools via diffustion adaptation. *Proc. International Workshop on Cognitive Information Processing*, pages 63–68, Elba Island, Italy, Aug. 2010.

[146] S. Y. Tu and A. H. Sayed. Mobile adaptive networks with self-organization abilities. *Proc. International Symposium on Wireless Communication Systems*, pages 379–383, York, United Kingdom, Sep. 2010.

[147] S. Y. Tu and A. H. Sayed. Tracking behavior of mobile adaptive networks. *Proc. Asilomar Conference on Signals, Systems and Computers*, pages 698–702, Pacific Grove, CA, Nov. 2010.

[148] S. Y. Tu and A. H. Sayed. Adaptive networks with noisy links. *Proc. IEEE Globecom*, pages 1–5, Houston, TX, Dec. 2011.

[149] S. Y. Tu and A. H. Sayed. Cooperative prey herding based on diffusion adaptation. *Proc. IEEE ICASSP*, pages 3752–3755, Prague, Czech Republic, May 2011.

[150] S. Y. Tu and A. H. Sayed. Mobile adaptive networks. *IEEE J. Selected Topics on Signal Processing*, 5(4):649–664, Aug. 2011.

[151] S. Y. Tu and A. H. Sayed. On the effects of topology and node distribution on learning over complex adaptive networks. *Proc. Asilomar Conference on Signals, Systems, and Computers*, pages 1166–1171, Pacific Grove, CA, Nov. 2011.

[152] S. Y. Tu and A. H. Sayed. Optimal combination rules for adaptation and learning over networks. *Proc. IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 317–320, San Juan, Puerto Rico, Dec. 2011.

[153] S. Y. Tu and A. H. Sayed. Adaptive decision making over complex networks. *Proc. Asilomar Conference on Signals, Systems, and Computers*, pages 525–530, Pacific Grove, CA, Nov. 2012.

[154] S. Y. Tu and A. H. Sayed. Diffusion networks outperform consensus networks. *Proc. IEEE Statistical Signal Processing Workshop*, pages 313–316, Ann Arbor, MI, Aug. 2012.

[155] S. Y. Tu and A. H. Sayed. Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Trans. on Signal Processing*, 60(12):6217–6234, Dec. 2012.

[156] S. Y. Tu and A. H. Sayed. Effective information flow over mobile adaptive networks. *Proc. International Workshop on Cognitive Information Processing (CIP)*, pages 1–6, Parador de Baiona, Spain, May 2012.

[157] S. Y. Tu and A. H. Sayed. On the influence of informed agents on learning and adaptation over networks. *IEEE Trans. on Signal Processing*, 61(6):1339–1356, Mar. 2013.

[158] G. Turner and T. Pitcher. Attack abatement: A model for group protection by combined avoidance and dilution. *American Naturalist*, 128:228–240, 1986.

[159] R. Vabo and L. Nottestad. An individual based model of fish school reaction: Predicting antipredator behavior as observed in nature. *Fisheries Oceanography*, 6:155–171, 1997.

[160] N. P. van der Aa, H. G. ter Morsche, and R. R. M. Mattheij. Computation of eigenvalue and eigenvector derivatives for a general complex-valued eigensystem. *Electronic Journal of Linear Algebra*, 16(1):300–314, Oct. 2007.

[161] T. Vicsek, A. Czirook, E. Ben-Jacob, O. Cohen, and I. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75:1226–1229, Aug. 1995.

[162] R. Vishwanathan and P. K. Varshney. Distributed detection with multiple sensors: Part i - fundamentals. *Proc. IEEE*, 85(1):54–63, Jan. 1997.

[163] N. K. Vitanov and M. R. Ausloos. Knowledge epidemics and population dynamics models for decribing idea diffusion. *Models of Science Dynamics - Encouters between Complexity Theory and Information Sciences*, A. Scharnhorst, K. Borner, and P. van den Besselaar (Eds.), Springer-Verlag, pages 69–126, 2012.

[164] Y. Xia, L. Li, J. Cao, M. Golz, and D. P. Mandic. A collaborative filtering approach for quasi-brain-death EEG analysis. *Proc. IEEE ICASSP*, pages 645–648, Prague, Czech Republic, May 2011.

[165] Y. Xia, D. Mandic, and A. H. Sayed. An adaptive diffusion augmented CLMS algorithm for distributed filtering of noncircular complex signals. *IEEE Signal Processing Letters*, to appear, 2011.

[166] L. Xiao and S. Boyd. Fast linear iteration for distributed averaging. *Systems and Control Letters*, 53:65–78, 2004.

[167] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. *Proc. Int. Symp. Information Processing Sensor Networks (IPSN)*, pages 63–70, Los Angeles, CA, Apr. 2005.

[168] L. Xiao, S. Boyd, and S. Lall. A space-time diffusion scheme peer-to-peer least-squares-estimation. *Proc. Information Processing in Sensor Networks (IPSN)*, pages 168–176, Nashville, TN, Apr. 2006.

[169] V. M. Yakovenko and Jr. J. B. Rosserm. Colloquium: Statistical mechanics of money, wealth, and income. *Rev. Modern Phys.*, 81:1703–1725, Oct.-Dec. 2009.

[170] M. E. Yildiz, A. Scaglione, and A. Ozdaglar. Asymmetric information diffusion via gossiping on static and dynamic networks. *IEEE Proc. of CDC*, pages 7467–7472, Atlanta, GA, Dec. 2010.

[171] M. M. Zavlanos, A. Ribeiro, and G. J. Pappas. Distributed control of mobility & routing in networks of robots. *Proc. IEEE SPAWC*, pages 236–240, Jun. 2011.

[172] Q. Zhao and B. Sadler. A survey of dynamic spectrum access. *IEEE Signal Process. Mag.*, 24:79–89, May 2007.

[173] X. Zhao and A. H. Sayed. Performance limits for distributed estimation over LMS adaptive networks. *IEEE Trans. on Signal Processing*, 60(10):5107–5124, Oct. 2012.

[174] X. Zhao, S. Y. Tu, and A. H. Sayed. Diffusion adaptation over networks under imperfect information exchange and non-stationary data. *IEEE Trans. on Signal Processing*, 60(7):3460–3475, Jul. 2012.