DIFFUSION ADAPTATION OVER NETWORKS*

Ali H. Sayed

Electrical Engineering Department
University of California at Los Angeles

Adaptive networks are well-suited to perform decentralized information processing and optimization tasks and to model various types of self-organized and complex behavior encountered in nature. Adaptive networks consist of a collection of agents with processing and learning abilities. The agents are linked together through a connection topology, and they cooperate with each other through local interactions to solve distributed optimization, estimation, and inference problems in real-time. The continuous diffusion of information across the network enables agents to adapt their performance in relation to streaming data and network conditions; it also results in improved adaptation and learning performance relative to non-cooperative agents. This article provides an overview of diffusion strategies for adaptation and learning over networks. The article is divided into several sections:

*The cite information for this article is as follows: **A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, R. Chellapa and S. Theodoridis, *Eds.*, vol. 3, pp. 323-454, Academic Press, Elsevier, 2014.** The work was supported in part by NSF grants EECS-060126, EECS-0725441, CCF-0942936, and CCF-1011918. The author is with the Electrical Engineering Department, University of California, Los Angeles, CA 90095, USA. Email: sayed@ee.ucla.edu.

1 Motivation

Consider a collection of N agents interested in estimating the same parameter vector, w^o, of size $M \times 1$. The vector is the minimizer of some global cost function, denoted by $J^{\text{glob}}(w)$, which the agents seek to optimize, say,

$$\boxed{w^o = \underset{w}{\text{argmin}} \;\; J^{\text{glob}}(w)} \tag{1}$$

We are interested in situations where the individual agents have access to partial information about the global cost function. In this case, cooperation among the agents becomes beneficial. For example, by cooperating with their neighbors, and by having these neighbors cooperate with their neighbors, procedures can be devised that would enable all agents in the network to converge towards the global optimum w^o through local interactions. The objective of decentralized processing is to allow spatially distributed agents to achieve a global objective by relying solely on local information and on in-network processing. Through a continuous process of cooperation and information sharing with neighbors, agents in a network can be made to approach the global performance level despite the localized nature of their interactions.

1.1 Networks and Neighborhoods

In this article we focus mainly on *connected* networks, although many of the results hold even if the network graph is separated into disjoint subgraphs. In a connected network, if we pick any two arbitrary nodes, then there will exist at least one path connecting them: the nodes may be connected directly by an edge if they are neighbors, or they may be connected by a path that passes through other intermediate nodes. Figure 1 provides a graphical representation of a connected network with $N = 10$ nodes. Nodes that are able to share information with each other are connected by edges. The sharing of information over these edges can be unidirectional or bi-directional. The neighborhood of any particular node is defined as the set of nodes that are connected to it by edges; we include in this set the node itself. The figure illustrates the neighborhood of node 3, which consists of the following subset of nodes: $\mathcal{N}_3 = \{1, 2, 3, 5\}$. For each node, the size of its neighborhood defines its degree. For example, node 3 in the figure has degree $|\mathcal{N}_3| = 4$, while node 8 has degree $|\mathcal{N}_8| = 5$. Nodes that are well connected have higher degrees. Note that we are denoting the neighborhood of an arbitrary node k by $\mathcal{N}_k$ and its size by $|\mathcal{N}_k|$. We shall also use the notation n_k to refer to $|\mathcal{N}_k|$.

The neighborhood of any node k therefore consists of all nodes with which node k can exchange information. We assume a symmetric situation in relation to neighbors so that if node k is a neighbor of node ℓ, then node ℓ is also a neighbor of node k. This does not necessarily mean that the flow of information between these two nodes is symmetrical. For instance, in future sections, we shall assign pairs of nonnegative weights to each edge connecting two neighboring nodes — see Fig. 2. In particular, we will assign the coefficient $a_{\ell k}$ to denote the weight used by node k to scale the data it receives from node ℓ; this scaling can be interpreted as a measure of trustworthiness or reliability that node k assigns to its interaction with node ℓ. Note that we are using two subscripts, ℓk, with the first subscript denoting the source node (where information originates from) and the second subscript denoting the sink node (where information moves to) so that:

$$\boxed{a_{\ell k} \;\; \equiv \;\; a_{\ell \,\to\, k} \quad \text{(information flowing from node } \ell \text{ to node } k\text{)}} \tag{2}$$

In this way, the alternative coefficient $a_{k\ell}$ will denote the weight used to scale the data sent from node k to ℓ:

$$\boxed{a_{k\ell} \;\; \equiv \;\; a_{k \,\to\, \ell} \quad \text{(information flowing from node } k \text{ to node } \ell\text{)}} \tag{3}$$

The weights $\{a_{k\ell}, a_{\ell k}\}$ can be different, and one or both of them can be zero, so that the exchange of information over the edge connecting the neighboring nodes (k, ℓ) need not be symmetric. When one of the

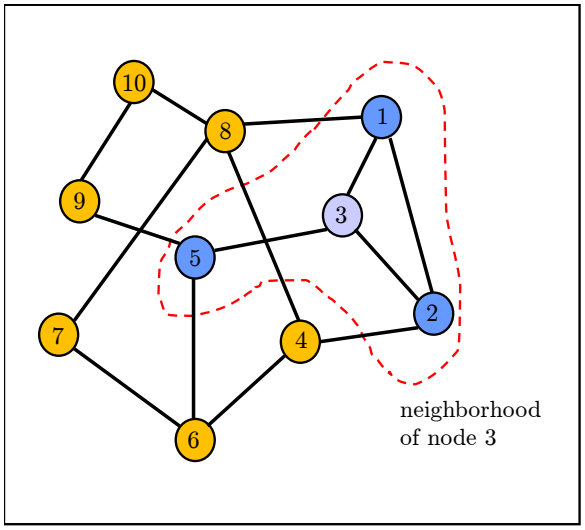

Figure 1: A network consists of a collection of cooperating nodes. Nodes that are linked by edges can share information. The neighborhood of any particular node consists of all nodes that are connected to it by edges (including the node itself). The figure illustrates the neighborhood of node 3, which consists of nodes $\{1, 2, 3, 5\}$. Accordingly, node 3 has degree 4, which is the size of its neighborhood.

weights is zero, say, $a_{k\ell} = 0$, then this situation means that even though nodes (k, ℓ) are neighbors, node ℓ is either not receiving data from node k or the data emanating from node k is being annihilated before reaching node ℓ. Likewise, when $a_{kk} > 0$, then node k scales its own data, whereas $a_{kk} = 0$ corresponds to the situation when node k does not use its own data. Usually, in graphical representations like those in Fig. 1, edges are drawn between neighboring nodes that can share information. And, it is understood that the actual sharing of information is controlled by the values of the scaling weights that are assigned to the edge; these values can turn off communication in one or both directions and they can also scale one direction more heavily than the reverse direction, and so forth.

1.2 Cooperation Among Agents

Now, depending on the application under consideration, the solution vector w^o from (1) may admit different interpretations. For example, the entries of w^o may represent the location coordinates of a nutrition source that the agents are trying to find, or the location of an accident involving a dangerous chemical leak. The nodes may also be interested in locating a predator and tracking its movements over time. In these localization applications, the vector w^o is usually two or three-dimensional. In other applications, the entries of w^o may represent the parameters of some model that the network wishes to learn, such as identifying the model parameters of a biological process or the occupied frequency bands in a shared communications medium. There are also situations where different agents in the network may be interested in estimating different entries of w^o, or even different parameter vectors w^o altogether, say, $\{w_k^o\}$ for node k, albeit with some relation among the different vectors so that cooperation among the nodes can still be rewarding. In this article, however, we focus exclusively on the special (yet frequent and important) case where all agents are interested in estimating the *same* parameter vector w^o.

Since the agents have a common objective, it is natural to expect cooperation among them to be beneficial in general. One important question is therefore how to develop cooperation strategies that can lead to better performance than when each agent attempts to solve the optimization problem individually. Another important question is how to develop strategies that endow networks with the ability to adapt and learn in

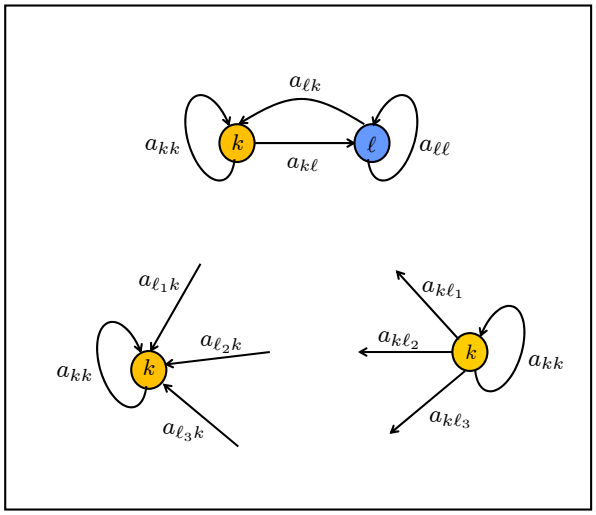

Figure 2: In the top part, and for emphasis purposes, we are representing the edge between nodes k and ℓ by two separate directed links: one moving from k to ℓ and the other moving from ℓ to k. Two nonnegative weights are used to scale the sharing of information over these directed links. The scalar $a_{k\ell}$ denotes the weight used to scale data sent from node k to ℓ, while $a_{\ell k}$ denotes the weight used to scale data sent from node ℓ to k. The weights $\{a_{k,\ell}, a_{\ell k}\}$ can be different, and one or both of them can be zero, so that the exchange of information over the edge connecting any two neighboring nodes need not be symmetric. The bottom part of the figure illustrates directed links arriving to node k from its neighbors $\{\ell_1, \ell_2, \ell_3, \ldots\}$ (left) and leaving from node k towards these same neighbors (right).

real-time in response to changes in the statistical properties of the data. This article provides an overview of results in the area of *diffusion adaptation* with illustrative examples. Diffusion strategies are powerful methods that enable adaptive learning and cooperation over networks. There have been other useful works in the literature on the use of alternative *consensus strategies* to develop distributed optimization solutions over networks. Nevertheless, we explain in App. E why diffusion strategies outperform consensus strategies in terms of their mean-square-error stability and performance. For this reason, we focus in the body of the chapter on presenting the theoretical foundations for diffusion strategies and their performance.

1.3 Notation

In our treatment, we need to distinguish between random variables and deterministic quantities. For this reason, we use **boldface** letters to represent random variables and normal font to represent deterministic (non-random) quantities. For example, the boldface letter $\boldsymbol{d}$ denotes a random quantity, while the normal font letter d denotes an observation or realization for it. We also need to distinguish between matrices and vectors. For this purpose, we use CAPITAL letters to refer to matrices and small letters to refer to both vectors and scalars; Greek letters always refer to scalars. For example, we write R to denote a covariance matrix and w to denote a vector of parameters. We also write σ_v^2 to refer to the variance of a random variable. To distinguish between a vector d (small letter) and a scalar d (also a small letter), we use parentheses to index scalar quantities and subscripts to index vector quantities. Thus, we write $d(i)$ to refer to the value of a scalar quantity d at time i, and d_i to refer to the value of a vector quantity d at time i. Thus, $d(i)$ denotes a scalar while d_i denotes a vector. All vectors in our presentation are column vectors, with the exception of the regression vector (denoted by the letter u), which will be taken to be a row vector for convenience of presentation. The symbol T denotes transposition, and the symbol $*$ denotes complex conjugation for scalars and complex-conjugate transposition for matrices. The notation $\mathrm{col}\{a, b\}$ denotes a column vector

with entries a and b stacked on top of each other, and the notation $\mathrm{diag}\{a, b\}$ denotes a diagonal matrix with entries a and b. Likewise, the notation $\mathrm{vec}(A)$ vectorizes its matrix argument and stacks the columns of A on top of each other. The notation $\|x\|$ denotes the Euclidean norm of its vector argument, while $\|x\|_{b,\infty}$ denotes the block maximum norm of a block vector (defined in App. D). Similarly, the notation $\|x\|_{\Sigma}^2$ denotes the weighted square value, $x^*\Sigma x$. Moreover, $\|A\|_{b,\infty}$ denotes the block maximum norm of a matrix (also defined in App. D), and $\rho(A)$ denotes the spectral radius of the matrix (i.e., the largest absolute magnitude among its eigenvalues). Finally, I_M denotes the identity matrix of size $M \times M$; sometimes, for simplicity of notation, we drop the subscript M from I_M when the size of the identity matrix is obvious from the context. Table 1 provides a summary of the symbols used in the article for ease of reference.

Table 1: Summary of notation conventions used in the article.

$\boldsymbol{d}$	Boldface notation denotes random variables.
d	Normal font denotes realizations of random variables.
A	Capital letters denote matrices.
a	Small letters denote vectors or scalars.
α	Greek letters denote scalars.
$d(i)$	Small letters with parenthesis denote scalars.
d_i	Small letters with subscripts denote vectors.
T	Matrix transposition.
$*$	Complex conjugation for scalars and complex-conjugate transposition for matrices.
$\mathrm{col}\{a, b\}$	Column vector with entries a and b.
$\mathrm{diag}\{a, b\}$	Diagonal matrix with entries a and b.
$\mathrm{vec}(A)$	Vectorizes matrix A and stacks its columns on top of each other.
$\|x\|$	Euclidean norm of its vector argument.
$\|x\|_{\Sigma}^2$	Weighted square value $x^*\Sigma x$.
$\|x\|_{b,\infty}$	Block maximum norm of a block vector – see App. D.
$\|A\|_{b,\infty}$	Block maximum norm of a matrix – see App. D.
$\|A\|$	2−induced norm of matrix A (its largest singular value).
$\rho(A)$	Spectral radius of its matrix argument.
I_M	Identity matrix of size $M \times M$; sometimes, we drop the subscript M.

2 Mean-Square-Error Estimation

Readers interested in the development of the distributed optimization strategies and their adaptive versions can move directly to Sec. 3. The purpose of the current section is to motivate the virtues of distributed in-network processing, and to provide illustrative examples in the context of mean-square-error estimation. Advanced readers can skip this section on a first reading.

We start our development by associating with each agent k an individual cost (or utility) function, $J_k(w)$. Although the algorithms presented in this article apply to more general situations, we nevertheless assume in our presentation that the cost functions $J_k(w)$ are strictly convex so that each one of them has a unique minimizer. We further assume that, for all costs $J_k(w)$, the minimum occurs at the same value w^o. Obviously, the choice of $J_k(w)$ is limitless and is largely dependent on the application. It is sufficient for our purposes to illustrate the main concepts underlying diffusion adaptation by focusing on the case of mean-square-error (MSE) or quadratic cost functions. In the sequel, we provide several examples to illustrate how such quadratic cost functions arise in applications and how cooperative processing over networks can be beneficial. At the same time, we note that most of the arguments in this article can be extended beyond MSE optimization to more general cost functions and to situations where the minimizers of the individual costs $J_k(w)$ need not agree with each other — as already shown in [1–3]; see also Sec. 10.4 for a brief summary.

In non-cooperative solutions, each agent would operate individually on its own cost function $J_k(w)$ and

optimize it to determines w^o, without any interaction with the other nodes. However, the analysis and derivations in future sections will reveal that nodes can benefit from cooperation between them in terms of better performance (such as converging faster to w^o or tracking a changing w^o more effectively) — see, e.g., Theorems 6.3–6.5 and Sec. 7.3.

2.1 Application: Autoregressive Modeling

Our first example relates to identifying the parameters of an auto-regressive (AR) model from noisy data. Thus, consider a situation where agents are spread over some geographical region and each agent k is observing realizations $\{d_k(i)\}$ of an AR zero-mean random process $\{\boldsymbol{d}_k(i)\}$, which satisfies a model of the form:

$$\boldsymbol{d}_k(i) = \sum_{m=1}^{M} \beta_m \boldsymbol{d}_k(i-m) + \boldsymbol{v}_k(i) \tag{4}$$

The scalars $\{\beta_m\}$ represent the model parameters that the agents wish to identify, and $\boldsymbol{v}_k(i)$ represents an additive zero-mean white noise process with power:

$$\sigma_{v,k}^2 \;\triangleq\; \mathbb{E}\,|\boldsymbol{v}_k(i)|^2 \tag{5}$$

It is customary to assume that the noise process is temporally white *and* spatially independent so that noise terms across different nodes are independent of each other and, at the same node, successive noise samples are also independent of each other with a time-independent variance $\sigma_{v,k}^2$:

$$\begin{cases} \mathbb{E}\,\boldsymbol{v}_k(i)\boldsymbol{v}_k^*(j) \;=\; 0, & \text{for all } i \neq j \text{ (temporal whiteness)} \\ \mathbb{E}\,\boldsymbol{v}_k(i)\boldsymbol{v}_m^*(j) \;=\; 0, & \text{for all } i,j \text{ whenever } k \neq m \text{ (spatial whiteness)} \end{cases} \tag{6}$$

The noise process $\boldsymbol{v}_k(i)$ is further assumed to be independent of past signals $\{\boldsymbol{d}_\ell(i-m), m \geq 1\}$ across all nodes ℓ. Observe that we are allowing the noise power profile, $\sigma_{v,k}^2$, to vary with k. In this way, the quality of the measurements is allowed to vary across the network with some nodes collecting noisier data than other nodes. Figure 3 illustrates an example of a network consisting of $N = 10$ nodes spread over a region in space. The figure shows the neighborhood of node 2, which consists of nodes $\{1, 2, 3\}$.

Linear Model

To illustrate the difference between cooperative and non-cooperative estimation strategies, let us first explain that the data can be represented in terms of a linear model. To do so, we collect the model parameters $\{\beta_m\}$ into an $M \times 1$ column vector w^o:

$$w^o \;\triangleq\; \mathrm{col}\{\beta_1, \beta_2, \ldots, \beta_M\} \tag{7}$$

and the past data into a $1 \times M$ row regression vector $\boldsymbol{u}_{k,i}$:

$$\boldsymbol{u}_{k,i} \;\triangleq\; \begin{bmatrix} \boldsymbol{d}_k(i-1) & \boldsymbol{d}_k(i-2) & \ldots & \boldsymbol{d}_k(i-M) \end{bmatrix} \tag{8}$$

Then, we can rewrite the measurement equation (4) at each node k in the equivalent *linear model* form:

$$\boxed{\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o \;+\; \boldsymbol{v}_k(i)} \tag{9}$$

Linear relations of the form (9) are common in applications and arise in many other contexts (as further illustrated by the next three examples in this section).

We assume the stochastic processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ in (9) have zero means and are jointly wide-sense stationary. We denote their second-order moments by:

$$\sigma_{d,k}^2 \;\triangleq\; \mathbb{E}\,|\boldsymbol{d}_k(i)|^2 \qquad \text{(scalar)} \tag{10}$$

$$R_{u,k} \;\triangleq\; \mathbb{E}\,\boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i} \qquad (M \times M) \tag{11}$$

$$r_{du,k} \;\triangleq\; \mathbb{E}\,\boldsymbol{d}_k(i)\boldsymbol{u}_{k,i}^* \qquad (M \times 1) \tag{12}$$

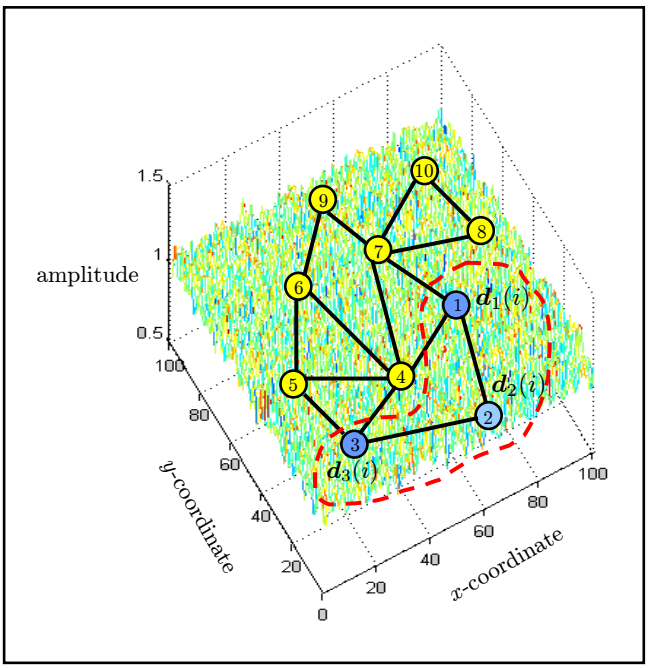

Figure 3: A collection of nodes, spread over a geographic region, observes realizations of an AR random process and cooperates to estimate the underlying model parameters $\{\beta_m\}$ in the presence of measurement noise. The noise power profile can vary over space.

As was the case with the noise power profile, we are allowing the moments $\{\sigma_{d,k}^2, R_{u,k}, r_{du,k}\}$ to depend on the node index k so that these moments can vary with the spatial dimension as well. The covariance matrix $R_{u,k}$ is, by definition, always non-negative definite. However, for convenience of presentation, we shall assume that $R_{u,k}$ is actually positive-definite (and, hence, invertible):

$$R_{u,k} > 0 \tag{13}$$

Non-Cooperative Mean-Square-Error Solution

One immediate result that follows from the linear model (9) is that the unknown parameter vector w^o can be recovered *exactly* by each individual node from knowledge of the local moments $\{r_{du,k}, R_{u,k}\}$ alone. To see this, note that if we multiply both sides of (9) by $\boldsymbol{u}_{k,i}^*$ and take expectations we obtain

$$\underbrace{\mathbb{E}\,\boldsymbol{u}_{k,i}^*\boldsymbol{d}_k(i)}_{r_{du,k}} \;=\; \underbrace{\left(\mathbb{E}\,\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}\right)}_{R_{u,k}} w^o \;+\; \underbrace{\mathbb{E}\,\boldsymbol{u}_{k,i}^*\boldsymbol{v}_k(i)}_{=0} \tag{14}$$

so that

$$\boxed{r_{du,k} = R_{u,k}\,w^o \iff w^o = R_{u,k}^{-1}\,r_{du,k}} \tag{15}$$

It is seen from (15) that w^o is the solution to a linear system of equations and that this solution can be computed by every node directly from its moments $\{R_{u,k}, r_{du,k}\}$. It is useful to re-interpret construction

(15) as the solution to a minimum mean-square-error (MMSE) estimation problem [4, 5]. Indeed, it can be verified that the quantity $R_{u,k}^{-1}\, r_{du,k}$ that appears in (15) is the unique solution to the following MMSE problem:

$$\boxed{\min_{w}\ \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2} \tag{16}$$

To verify this claim, we denote the cost function that appears in (16) by

$$J_k(w) \ \triangleq\ \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w|^2 \tag{17}$$

and expand it to find that

$$J_k(w) \ =\ \sigma_{d,k}^2 - w^*r_{du,k} - r_{du,k}^*w + w^*R_{u,k}w \tag{18}$$

The cost function $J_k(w)$ is quadratic in w and it has a unique minimizer since $R_{u,k} > 0$. Differentiating $J_k(w)$ with respect to w we find its gradient vector:

$$\nabla_w J(w) \ =\ (R_{u,k}w \ -\ r_{du,k})^* \tag{19}$$

It is seen that this gradient vector is annihilated at the same value $w = w^o$ given by (15). Therefore, we can equivalently state that if each node k solves the MMSE problem (16), then the solution vector coincides with the desired parameter vector w^o. This observation explains why it is often justified to consider mean-square-error cost functions when dealing with estimation problems that involve data that satisfy linear models similar to (9).

Besides w^o, the solution of the MMSE problem (16) also conveys information about the noise level in the data. Note that by substituting w^o from (15) into expression (16) for $J_k(w)$, the resulting minimum mean-square-error value that is attained by node k is found to be:

$$\begin{aligned}
\mathrm{MSE}_k \ &\triangleq\ J_k(w^o) \\
&=\ \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}w^o|^2 \\
&\overset{(9)}{=}\ \mathbb{E}\,|\boldsymbol{v}_k(i)|^2 \\
&=\ \sigma_{v,k}^2 \\
&\triangleq\ J_{k,\min}
\end{aligned} \tag{20}$$

We shall use the notation $J_k(w^o)$ and $J_{k,\min}$ interchangeably to denote the minimum cost value of $J_k(w)$. The above result states that, when each agent k recovers w^o from knowledge of its moments $\{R_{u,k}, r_{du,k}\}$ using expression (15), then the agent attains an MSE performance level that is equal to the noise power at its location, $\sigma_{v,k}^2$. An alternative useful expression for the minimum cost can be obtained by substituting expression (15) for w^o into (18) and simplifying the expression to find that

$$\mathrm{MSE}_k \ =\ \sigma_{d,k}^2 - r_{du,k}^*R_{u,k}^{-1}r_{du,k} \tag{21}$$

This second expression is in terms of the data moments $\{\sigma_{d,k}^2, r_{du,k}, R_{u,k}\}$.

Non-Cooperative Adaptive Solution

The optimal MMSE implementation (15) for determining w^o requires knowledge of the statistical information $\{r_{du,k}, R_{u,k}\}$. This information is usually not available beforehand. Instead, the agents are more likely to have access to successive time-indexed observations $\{d_k(i), u_{k,i}\}$ of the random processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ for $i \geq 0$. In this case, it becomes necessary to devise a scheme that would allow each node to use its measurements to approximate w^o. It turns out that an adaptive solution is possible. In this alternative implementation,

each node k feeds its observations $\{d_k(i), u_{k,i}\}$ into an adaptive filter and evaluates successive estimates for w^o. As time passes by, the estimates would get closer to w^o.

The adaptive solution operates as follows. Let $w_{k,i}$ denote an estimate for w^o that is computed by node k at time i based on all the observations $\{d_k(j), u_{k,j},\ j \leq i\}$ it has collected up to that time instant. There are many adaptive algorithms that can be used to compute $w_{k,i}$; some filters are more accurate than others (usually, at the cost of additional complexity) [4–7]. It is sufficient for our purposes to consider one simple (yet effective) filter structure, while noting that most of the discussion in this article can be extended to other structures. One of the simplest choices for an adaptive structure is the least-mean-squares (LMS) filter, where the data are processed by each node k as follows:

$$e_k(i) = d_k(i) - u_{k,i} w_{k,i-1} \tag{22}$$

$$w_{k,i} = w_{k,i-1} + \mu_k u_{k,i}^* e_k(i), \quad i \geq 0 \tag{23}$$

Starting from some initial condition, say, $w_{k,-1} = 0$, the filter iterates over $i \geq 0$. At each time instant, i, the filter uses the local data $\{d_k(i), u_{k,i}\}$ at node k to compute a local estimation error, $e_k(i)$, via (22). The error is then used to update the existing estimate from $w_{k,i-1}$ to $w_{k,i}$ via (23). The factor μ_k that appears in (23) is a constant positive step-size parameter; usually chosen to be sufficiently small to ensure mean-square stability and convergence, as discussed further ahead in the article. The step-size parameter can be selected to vary with time as well; one popular choice is to replace μ_k in (23) with the following construction:

$$\mu_k(i) \stackrel{\Delta}{=} \frac{\mu_k}{\epsilon + \|u_{k,i}\|^2} \tag{24}$$

where $\epsilon > 0$ is a small positive parameter and $\mu_k > 0$. The resulting filter implementation is known as normalized LMS [5] since the step-size is normalized by the squared norm of the regression vector. Other choices include step-size sequences $\{\mu(i) \geq 0\}$ that satisfy both conditions:

$$\sum_{i=0}^{\infty} \mu(i) = \infty, \qquad \sum_{i=0}^{\infty} \mu^2(i) < \infty \tag{25}$$

Such sequences converge slowly towards zero. One example is the choice $\mu_k(i) = \frac{\mu_k}{i+1}$. However, by virtue of the fact that such step-sizes die out as $i \to \infty$, then these choices end up turning off adaptation. As such, step-size sequences satisfying (25) are not generally suitable for applications that require continuous learning, especially under non-stationary environments. For this reason, in this article, we shall focus exclusively on the constant step-size case (23) in order to ensure continuous adaptation and learning.

Equations (22)–(23) are written in terms of the observed quantities $\{d_k(i), u_{k,i}\}$; these are deterministic values since they correspond to observations of the random processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$. Often, when we are interested in highlighting the random nature of the quantities involved in the adaptation step, especially when we study the mean-square performance of adaptive filters, it becomes more useful to rewrite the recursions using the **boldface** notation to highlight the fact that the quantities that appear in (22)–(23) are actually realizations of random variables. Thus, we also write:

$$\boldsymbol{e}_k(i) = \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1} \tag{26}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^* \boldsymbol{e}_k(i), \quad i \geq 0 \tag{27}$$

where $\{\boldsymbol{e}_k(i), \boldsymbol{w}_{k,i}\}$ will be random variables as well.

The performance of adaptive implementations of this kind are well-understood for both cases of stationary w^o and changing w^o [4–7]. For example, in the stationary case, if the adaptive implementation (26)–(27) were to succeed in having its estimator $\boldsymbol{w}_{k,i}$ tend to w^o with probability one as $i \to \infty$, then we would expect the error signal $\boldsymbol{e}_k(i)$ in (26) to tend towards the noise signal $\boldsymbol{v}_k(i)$ (by virtue of the linear model (9)). This means that, under this ideal scenario, the variance of the error signal $\boldsymbol{e}_k(i)$ would be expected to tend towards the noise variance, $\sigma_{v,k}^2$, as $i \to \infty$. Recall from (20) that the noise variance is the least cost that the MSE solution can attain. Therefore, such limiting behavior by the adaptive filter would be desirable.

However, it is well-known that there is always some loss in mean-square-error performance when adaptation is employed due to the effect of gradient noise, which is caused by the algorithm's reliance on observations (or realizations) $\{d_k(i), u_{k,i}\}$ rather than on the actual moments $\{r_{du,k}, R_{u,k}\}$. In particular, it is known that for LMS filters, the variance of $\boldsymbol{e}_k(i)$ in steady-state will be larger than $\sigma_{v,k}^2$ by a small amount, and the size of the offset is proportional to the step-size parameter μ_k (so that smaller step-sizes lead to better mean-square-error (MSE) performance albeit at the expense of slower convergence). It is easy to see why the variance of $\boldsymbol{e}_k(i)$ will be larger than $\sigma_{v,k}^2$ from the definition of the error signal in (26). Introduce the weight-error vector

$$\boxed{\widetilde{\boldsymbol{w}}_{k,i} \;\stackrel{\Delta}{=}\; w^o - \boldsymbol{w}_{k,i}} \tag{28}$$

and the so-called *a-priori* error signal

$$\boxed{\boldsymbol{e}_{a,k}(i) \;\stackrel{\Delta}{=}\; \boldsymbol{u}_{k,i}\widetilde{\boldsymbol{w}}_{k,i-1}} \tag{29}$$

This second error measures the difference between the uncorrupted term $\boldsymbol{u}_{k,i}w^o$ and its estimator prior to adaptation, $\boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}$. It then follows from the data model (9) and from the defining expression (26) for $\boldsymbol{e}_k(i)$ that

$$\begin{aligned}
\boldsymbol{e}_k(i) &= \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1} \\
&= \boldsymbol{u}_{k,i}w^o + \boldsymbol{v}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1} \\
&= \boldsymbol{e}_{a,k}(i) \;+\; \boldsymbol{v}_k(i)
\end{aligned} \tag{30}$$

Since the noise component, $\boldsymbol{v}_k(i)$, is assumed to be zero-mean and independent of all other random variables, we conclude that

$$\boxed{\mathbb{E}\,|\boldsymbol{e}_k(i)|^2 \;=\; \mathbb{E}\,|\boldsymbol{e}_{a,k}(i)|^2 \;+\; \sigma_{v,k}^2} \tag{31}$$

This relation holds for any time instant i; it shows that the variance of the output error, $\boldsymbol{e}_k(i)$, is larger than $\sigma_{v,k}^2$ by an amount that is equal to the variance of the *a-priori* error, $\boldsymbol{e}_{a,k}(i)$. We define the filter mean-square-error (MSE) and excess-mean-square-error (EMSE) as the following steady-state measures:

$$\text{MSE}_k \;\stackrel{\Delta}{=}\; \lim_{i\to\infty} \mathbb{E}\,|\boldsymbol{e}_k(i)|^2 \tag{32}$$

$$\text{EMSE}_k \;\stackrel{\Delta}{=}\; \lim_{i\to\infty} \mathbb{E}\,|\boldsymbol{e}_{a,k}(i)|^2 \tag{33}$$

so that, for the adaptive implementation (compare with (20)):

$$\boxed{\text{MSE}_k \;=\; \text{EMSE}_k \;+\; \sigma_{v,k}^2} \tag{34}$$

Therefore, the EMSE term quantifies the size of the offset in the MSE performance of the adaptive filter. We also define the filter mean-square-deviation (MSD) as the steady-state measure:

$$\text{MSD}_k \;\stackrel{\Delta}{=}\; \lim_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \tag{35}$$

which measures how far $\boldsymbol{w}_{k,i}$ is from w^o in the mean-square-error sense in steady-state. It is known that the MSD and EMSE of LMS filters of the form (26)–(27) can be approximated for sufficiently small-step sizes by the following expressions [4–7]:

$$\text{EMSE}_k \;\approx\; \mu_k\sigma_{v,k}^2\text{Tr}(R_{u,k})/2 \tag{36}$$

$$\text{MSD}_k \;\approx\; \mu_k\sigma_{v,k}^2 M/2 \tag{37}$$

It is seen that the smaller the step-size parameter, the better the performance of the adaptive solution.

Cooperative Adaptation through Diffusion

Observe from (36)–(37) that even if all nodes employ the same step-size, $\mu_k = \mu$, and even if the regression data are spatially uniform so that $R_{u,k} = R_u$ for all k, the mean-square-error performance across the nodes still varies in accordance with the variation of the noise power profile, $\sigma_{v,k}^2$, across the network. Nodes with larger noise power will perform worse than nodes with smaller noise power. However, since all nodes are observing data arising from the *same* underlying model w^o, it is natural to expect cooperation among the nodes to be beneficial. By cooperation we mean that neighboring nodes can share information (such as measurements or estimates) with each other as permitted by the network topology. Starting in the next section, we will motivate and describe algorithms that enable nodes to carry out adaptation and learning in a cooperative manner to enhance performance.

Specifically, we are going to see that one way to achieve cooperation is by developing adaptive algorithms that enable the nodes to optimize the following global cost function in a distributed manner:

$$\boxed{\min_{w} \sum_{k=1}^{N} \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w|^2} \tag{38}$$

where the global cost is the aggregate objective:

$$J^{\text{glob}}(w) \;\triangleq\; \sum_{k=1}^{N} \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w|^2 \;=\; \sum_{k=1}^{N} J_k(w) \tag{39}$$

Comparing (38) with (16), we see that we are now adding the individual costs, $J_k(w)$, from across all nodes. Note that since the desired w^o satisfies (15) at every node k, then it also satisfies

$$\left(\sum_{k=1}^{M} R_{u,k}\right) w^o \;=\; \sum_{k=1}^{N} r_{du,k} \tag{40}$$

But it can be verified that the optimal solution to (38) is given by the same w^o that satisfies (40). Therefore, solving the global optimization problem (38) also leads to the desired w^o. In future sections, we will show how cooperative and distributed adaptive schemes for solving (38), such as (153) or (154) further ahead, lead to improved performance in estimating w^o (in terms of smaller mean-square-deviation and faster convergence rate) than the non-cooperative mode (26)–(27), where each agent runs its own individual adaptive filter — see, e.g., Theorems 6.3–6.5 and Sec. 7.3.

2.2 Application: Tapped-Delay-Line Models

Our second example to motivate MSE cost functions, $J_k(w)$, and linear models relates to identifying the parameters of a moving-average (MA) model from noisy data. MA models are also known as finite-impulse-response (FIR) or tapped-delay-line models. Thus, consider a situation where agents are interested in estimating the parameters of an FIR model, such as the taps of a communications channel or the parameters of some (approximate) model of interest in finance or biology. Assume the agents are able to independently probe the unknown model and observe its response to excitations in the presence of additive noise; this situation is illustrated in Fig. 4, with the probing operation highlighted for one of the nodes (node 4).

The schematics inside the enlarged diagram in Fig. 4 is meant to convey that each node k probes the model with an input sequence $\{\boldsymbol{u}_k(i)\}$ and measures the resulting response sequence, $\{\boldsymbol{d}_k(i)\}$, in the presence

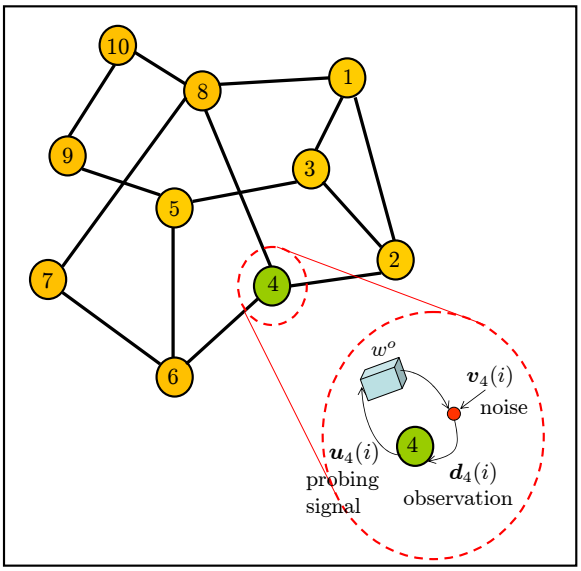

Figure 4: The network is interested in estimating the parameter vector w^o that describes an underlying tapped-delay-line model. The agents are assumed to be able to independently probe the unknown system, and observe its response to excitations, under noise, as indicated in the highlighted diagram for node 4.

of additive noise. The system dynamics for each agent k is assumed to be described by a MA model of the form:

$$\boldsymbol{d}_k(i) = \sum_{m=0}^{M-1} \beta_m \boldsymbol{u}_k(i-m) + \boldsymbol{v}_k(i) \tag{41}$$

In this model, the term $\boldsymbol{v}_k(i)$ again represents an additive zero-mean noise process that is assumed to be temporally white and spatially independent; it is also assumed to be independent of the input process, $\{\boldsymbol{u}_\ell(j)\}$, for all i, j and ℓ. The scalars $\{\beta_m\}$ represent the model parameters that the agents seek to identify. If we again collect the model parameters into an $M \times 1$ column vector w^o:

$$w^o \triangleq \mathrm{col}\,\{\beta_0, \beta_1, \ldots, \beta_{M-1}\} \tag{42}$$

and the input data into a $1 \times M$ row regression vector:

$$\boldsymbol{u}_{k,i} \triangleq \begin{bmatrix} \boldsymbol{u}_k(i) & \boldsymbol{u}_k(i-1) & \ldots & \boldsymbol{u}_k(i-M+1) \end{bmatrix} \tag{43}$$

then, we can again express the measurement equation (41) at each node k in the same linear model as (9), namely,

$$\boxed{\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i)} \tag{44}$$

As was the case with model (9), we can likewise verify that, in view of (44), the desired parameter vector w^o satisfies the same normal equations (15), i.e.,

$$r_{du,k} = R_{u,k}\, w^o \iff w^o = R_{u,k}^{-1}\, r_{du,k} \tag{45}$$

where the moments $\{r_{du,k}, R_{u,k}\}$ continue to be defined by expressions (11)–(12) with $\boldsymbol{u}_{k,i}$ now defined by (43). Therefore, each node k can determine w^o on its own by solving the same MMSE estimation problem

(16). This solution method requires knowledge of the moments $\{r_{du,k}, R_{u,k}\}$ and, according to (20), each agent k would then attain an MSE level that is equal to the noise power level at its location.

Alternatively, when the statistical information $\{r_{du,k}, R_{u,k}\}$ is not available, each agent k can estimate w^o iteratively by feeding data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ into the adaptive implementation (26)–(27). In this way, each agent k will achieve the same performance level shown earlier in (36)–(37), with the limiting performance being again dependent on the local noise power level, $\sigma_{v,k}^2$. Therefore, nodes with larger noise power will perform worse than nodes with smaller noise power. However, since all nodes are observing data arising from the same underlying model w^o, it is natural to expect cooperation among the nodes to be beneficial. As we are going to see, starting from the next section, one way to achieve cooperation and improve performance is by developing algorithms that optimize the same global cost function (38) in an adaptive and distributed manner, such as algorithms (153) and (154) further ahead.

2.3 Application: Target Localization

Our third example relates to the problem of locating a destination of interest (such as the location of a nutrition source or a chemical leak) or locating and tracking an object of interest (such as a predator or a projectile). In several such localization applications, the agents in the network are allowed to move towards the target or away from it, in which case we would end up with a mobile adaptive network [8]. Biological networks behave in this manner such as networks representing fish schools, bird formations, bee swarms, bacteria motility, and diffusing particles [8–12]. The agents may move towards the target (e.g., when it is a nutrition source) or away from the target (e.g., when it is a predator). In other applications, the agents may remain static and are simply interested in locating a target or tracking it (such as tracking a projectile).

To motivate mean-square-error estimation in the context of localization problems, we start with the situation corresponding to a static target and static nodes. Thus, assume that the unknown location of the target in the cartesian plane is represented by the 2×1 vector $w^o = \mathrm{col}\{x^o, y^o\}$. The agents are spread over the same region of space and are interested in locating the target. The location of every agent k is denoted by the 2×1 vector $p_k = \mathrm{col}\{x_k, y_k\}$ in terms of its x and y coordinates – see Fig. 5. We assume the agents are aware of their location vectors. The distance between agent k and the target is denoted by r_k^o and is equal to:

$$r_k^o \;=\; \|w^o - p_k\| \tag{46}$$

The 1×2 unit-norm direction vector pointing from agent k towards the target is denoted by u_k^o and is given by:

$$u_k^o \;=\; \frac{(w^o - p_k)^T}{\|w^o - p_k\|} \tag{47}$$

Observe from (46) and (47) that r_k^o can be expressed in the following useful inner-product form:

$$r_k^o = u_k^o(w^o - p_k) \tag{48}$$

In practice, agents have noisy observations of both their distance and direction vector towards the target. We denote the noisy distance measurement collected by node k at time i by:

$$\boldsymbol{r}_k(i) \;=\; r_k^o \;+\; \boldsymbol{v}_k(i) \tag{49}$$

where $\boldsymbol{v}_k(i)$ denotes noise and is assumed to be zero-mean, and temporally white and spatially independent with variance

$$\sigma_{v,k}^2 \;\triangleq\; \mathbb{E}\,|\boldsymbol{v}_k(i)|^2 \tag{50}$$

We also denote the noisy direction vector that is measured by node k at time i by $\boldsymbol{u}_{k,i}$. This vector is a perturbed version of u_k^o. We assume that $\boldsymbol{u}_{k,i}$ continues to start from the location of the node at p_k, but

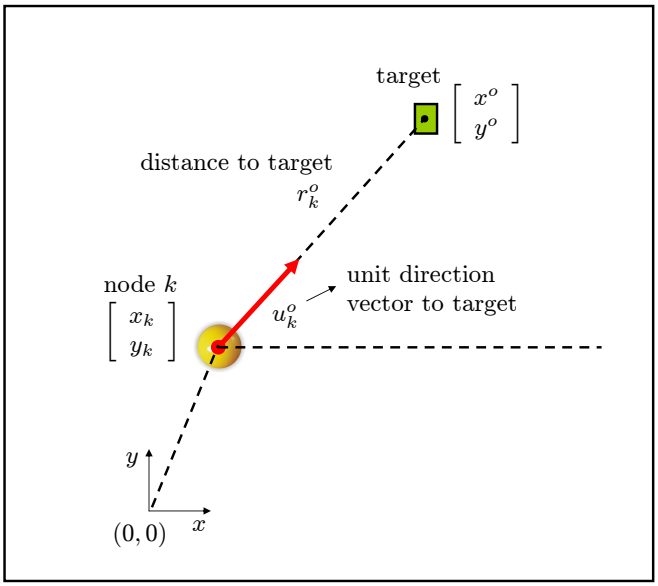

Figure 5: The distance from node k to the target is denoted by r_k^o and the unit-norm direction vector from the same node to the target is denoted by u_k^o. Node k is assumed to have access to noisy measurements of $\{r_k^o, u_k^o\}$.

that its tip is perturbed slightly either to the left or to the right relative to the tip of u_k^o — see Fig. 6. The perturbation to the tip of u_k^o is modeled as being the result of two effects: a small deviation that occurs along the direction that is perpendicular to u_k^o, and a smaller deviation that occurs along the direction of u_k^o. Since we are assuming that the tip of $\boldsymbol{u}_{k,i}$ is only slightly perturbed relative to the tip of u_k^o, then it is reasonable to expect the amount of perturbation along the parallel direction to be small compared to the amount of perturbation along the perpendicular direction.

Thus, we write

$$\boldsymbol{u}_{k,i} \;=\; u_k^o \;+\; \boldsymbol{\alpha}_k(i)\, u_k^{o\perp} \;+\; \boldsymbol{\beta}_k(i)\, u_k^o \qquad (1\times 2) \tag{51}$$

where $u_k^{o\perp}$ denotes a unit-norm row vector that lies in the same plane and whose direction is perpendicular to u_k^o. The variables $\boldsymbol{\alpha}_k(i)$ and $\boldsymbol{\beta}_k(i)$ denote zero-mean independent random noises that are temporally white and spatially independent with variances:

$$\sigma_{\alpha,k}^2 \;\triangleq\; \mathbb{E}\,|\boldsymbol{\alpha}_k(i)|^2, \qquad \sigma_{\beta,k}^2 \;\triangleq\; \mathbb{E}\,|\boldsymbol{\beta}_k(i)|^2 \tag{52}$$

We assume the contribution of $\boldsymbol{\beta}_k(i)$ is small compared to the contributions of the other noise sources, $\boldsymbol{\alpha}_k(i)$ and $\boldsymbol{v}_k(i)$, so that

$$\sigma_{\beta,k}^2 \ll \sigma_{\alpha,k}^2, \qquad \sigma_{\beta,k}^2 \ll \sigma_{v,k}^2 \tag{53}$$

The random noises $\{\boldsymbol{v}_k(i), \boldsymbol{\alpha}_k(i), \boldsymbol{\beta}_k(i)\}$ are further assumed to be independent of each other.

Using (48) we find that the noisy measurements $\{\boldsymbol{r}_k(i), \boldsymbol{u}_{k,i}\}$ are related to the unknown w^o via:

$$\boldsymbol{r}_k(i) = \boldsymbol{u}_{k,i}(w^o - p_k) \;+ \boldsymbol{z}_k(i) \tag{54}$$

where the modified noise term $\boldsymbol{z}_k(i)$ is defined in terms of the noises in $\boldsymbol{r}_k(i)$ and $\boldsymbol{u}_{k,i}$ as follows:

$$\begin{aligned} \boldsymbol{z}_k(i) \;&\triangleq\; \boldsymbol{v}_k(i) \;-\; \boldsymbol{\alpha}_k(i)\, u_k^{o\perp}\cdot(w^o - p_k) \;-\; \boldsymbol{\beta}_k(i)\cdot u_k^o\cdot(w^o - p_k) \\ &=\; \boldsymbol{v}_k(i) \;-\; \boldsymbol{\beta}_k(i)\cdot r_k^o \\ &\approx\; \boldsymbol{v}_k(i) \end{aligned} \tag{55}$$

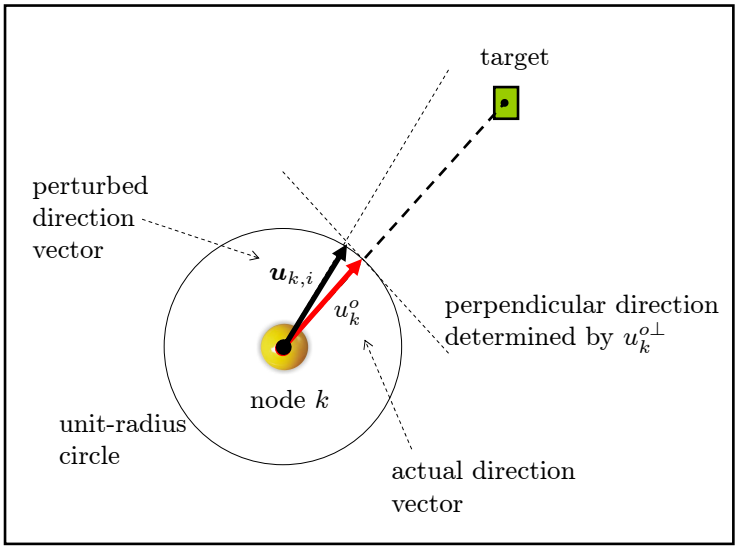

Figure 6: The tip of the noisy direction vector is modeled as being approximately perturbed away from the actual direction by two effects: a larger effect caused by a deviation along the direction that is perpendicular to u_k^o, and a smaller deviation along the direction that is parallel to u_k^o.

since, by construction,

$$u_k^{o\perp} \cdot (w^o - p_k) \;=\; 0 \tag{56}$$

and the contribution by $\boldsymbol{\beta}_k(i)$ is assumed to be sufficiently small. If we now introduce the adjusted signal:

$$\boldsymbol{d}_k(i) \;\triangleq\; \boldsymbol{r}_k(i) \;+\; \boldsymbol{u}_{k,i}\, p_k \tag{57}$$

then we arrive again from (54) and (55) at the following linear model for the available measurement variables $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ in terms of the target location w^o:

$$\boxed{\boldsymbol{d}_k(i) \approx \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i)} \tag{58}$$

There is one important difference in relation to the earlier linear models (9) and (44), namely, the variables $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ in (58) do not have zero means any longer. It is nevertheless straightforward to determine the first and second-order moments of the variables $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$. First, note from (49), (51), and (57) that

$$\mathbb{E}\,\boldsymbol{u}_{k,i} = u_k^o, \qquad \mathbb{E}\,\boldsymbol{d}_k(i) \;=\; r_k^o \;+\; u_k^o p_k \tag{59}$$

Even in this case of non-zero means, and in view of (58), the desired parameter vector w^o can still be shown to satisfy the same normal equations (15), i.e.,

$$r_{du,k} = R_{u,k} w^o \iff w^o = R_{u,k}^{-1}\, r_{du,k} \tag{60}$$

where the moments $\{r_{du,k}, R_{u,k}\}$ continue to be defined as

$$R_{u,k} \;\triangleq\; \mathbb{E}\,\boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i}, \quad r_{du,k} \;\triangleq\; \mathbb{E}\,\boldsymbol{u}_{k,i}^* \boldsymbol{d}_k(i) \tag{61}$$

To verify that (60) holds, we simply multiply both sides of (58) by $\boldsymbol{u}_{k,i}^*$ from the left, compute the expectations of both sides, and use the fact that $\boldsymbol{v}_k(i)$ has zero mean and is assumed to be independent of $\{\boldsymbol{u}_{\ell,j}\}$ for

all times j and nodes ℓ. However, the difference in relation to the earlier normal equations (15) is that the matrix $R_{u,k}$ is not the actual covariance matrix of $\boldsymbol{u}_{k,i}$ any longer. When $\boldsymbol{u}_{k,i}$ is not zero mean, its covariance matrix is instead defined as:

$$\begin{aligned} \mathrm{cov}_{u,k} &\triangleq \mathbb{E}\,(\boldsymbol{u}_{k,i} - u_k^o)^*(\boldsymbol{u}_{k,i} - u_k^o) \\ &= \mathbb{E}\,\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i} \;-\; u_k^{o*}u_k^o \end{aligned} \tag{62}$$

so that

$$R_{u,k} \;=\; \mathrm{cov}_{u,k} \;+\; u_k^{o*}u_k^o \tag{63}$$

We conclude from this relation that $R_{u,k}$ is positive-definite (and, hence, invertible) so that expression (60) is justified. This is because the covariance matrix, $\mathrm{cov}_{u,k}$, is itself positive-definite. Indeed, some algebra applied to the difference $\boldsymbol{u}_{k,i} - u_k^o$ from (51) shows that

$$\mathrm{cov}_{u,k} \;=\; \begin{bmatrix} \big(u_k^{o\perp}\big)^* & (u_k^o)^* \end{bmatrix} \begin{bmatrix} \sigma_{\alpha,k}^2 & \\ & \sigma_{\beta,k}^2 \end{bmatrix} \begin{bmatrix} u_k^{o\perp} \\ u_k^o \end{bmatrix} \tag{64}$$

where the matrix

$$\begin{bmatrix} u_k^{o\perp} \\ u_k^o \end{bmatrix} \tag{65}$$

is full rank since the rows $\{u_k^o,\; u_k^{o\perp}\}$ are linearly independent vectors.

Therefore, each node k can determine w^o on its own by solving the same minimum mean-square-error estimation problem (16). This solution method requires knowledge of the moments $\{r_{du,k}, R_{u,k}\}$ and, according to (20), each agent k would then attain an MSE level that is equal to the noise power level, $\sigma_{v,k}^2$, at its location.

Alternatively, when the statistical information $\{r_{du,k}, R_{u,k}\}$ is not available beforehand, each agent k can estimate w^o iteratively by feeding data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ into the adaptive implementation (26)–(27). In this case, each agent k will achieve the performance level shown earlier in (36)–(37), with the limiting performance being again dependent on the local noise power level, $\sigma_{v,k}^2$. Therefore, nodes with larger noise power will perform worse than nodes with smaller noise power. However, since all nodes are observing distances and direction vectors towards the same target location w^o, it is natural to expect cooperation among the nodes to be beneficial. As we are going to see, starting from the next section, one way to achieve cooperation and improve performance is by developing algorithms that solve the same global cost function (38) in an adaptive and distributed manner, by using algorithms such as (153) and (154) further ahead.

Role of Adaptation

The localization application helps highlight one of the main advantages of adaptation, namely, the ability of adaptive implementations to learn and track changing statistical conditions. For example, in the context of mobile networks, where nodes can move closer or further away from a target, the location vector for each agent k becomes time-dependent, say, $p_{k,i} = \mathrm{col}\{x_k(i), y_k(i)\}$. In this case, the actual distance and direction vector between agent k and the target also vary with time and become:

$$r_k^o(i) \;=\; \|w^o - p_{k,i}\|, \qquad u_{k,i}^o \;=\; \frac{(w^o - p_{k,i})^T}{\|w^o - p_{k,i}\|} \tag{66}$$

The noisy distance measurement to the target is then:

$$\boldsymbol{r}_k(i) \;=\; r_k^o(i) \;+\; \boldsymbol{v}_k(i) \tag{67}$$

where the variance of $\boldsymbol{v}_k(i)$ now depends on time as well:

$$\sigma_{v,k}^2(i) \;\triangleq\; \mathbb{E}\,|\boldsymbol{v}_k(i)|^2 \tag{68}$$

In the context of mobile networks, it is reasonable to assume that the variance of $\boldsymbol{v}_k(i)$ varies both with time and with the distance to the target: the closer the node is to the target, the less noisy the measurement of the distance is expected to be. Similar remarks hold for the variances of the noises $\boldsymbol{\alpha}_k(i)$ and $\boldsymbol{\beta}_k(i)$ that perturb the measurement of the direction vector, say,

$$\sigma_{\alpha,k}^2(i) \triangleq \mathbb{E}\,|\boldsymbol{\alpha}_k(i)|^2, \qquad \sigma_{\beta,k}^2(i) \triangleq \mathbb{E}\,|\boldsymbol{\beta}_k(i)|^2 \tag{69}$$

where now

$$\boldsymbol{u}_{k,i} \;=\; u_{k,i}^o \;+\; \boldsymbol{\alpha}_k(i)\, u_{k,i}^{o\perp} \;+\; \boldsymbol{\beta}_k(i)\, u_{k,i}^o \tag{70}$$

The same arguments that led to (58) can be repeated to lead to the same model, except that now the means of the variables $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ become time-dependent as well:

$$\mathbb{E}\,\boldsymbol{u}_{k,i} = u_{k,i}^o, \qquad \mathbb{E}\,\boldsymbol{d}_k(i) \;=\; r_k^o(i) \;+\; u_{k,i}^o\, p_{k,i} \tag{71}$$

Nevertheless, adaptive solutions (whether cooperative or non-cooperative), are able to track such time-variations because these solutions work directly with the observations $\{d_k(i), u_{k,i}\}$ and the successive observations will reflect the changing statistical profile of the data. In general, adaptive solutions are able to track changes in the underlying signal statistics rather well [4, 5], as long as the rate of non-stationarity is slow enough for the filter to be able to follow the changes.

2.4 Application: Collaborative Spectral Sensing

Our fourth and last example to illustrate the role of mean-square-error estimation and cooperation relates to spectrum sensing for cognitive radio applications. Cognitive radio systems involve two types of users: primary users and secondary users. To avoid causing harmful interference to incumbent primary users, unlicensed cognitive radio devices need to detect unused frequency bands even at low signal-to-noise (SNR) conditions [13–16]. One way to carry out spectral sensing is for each secondary user to estimate the aggregated power spectrum that is transmitted by all active primary users, and to locate unused frequency bands within the estimated spectrum. This step can be performed by the secondary users with or without cooperation.

Thus, consider a communications environment consisting of Q primary users and N secondary users. Let $S_q(e^{j\omega})$ denote the power spectrum of the signal transmitted by primary user q. To facilitate estimation of the spectral profile by the secondary users, we assume that each $S_q(e^{j\omega})$ can be represented as a linear combination of basis functions, $\{f_m(e^{j\omega})\}$, say, B of them [17]:

$$S_q(e^{j\omega}) \;=\; \sum_{m=1}^{B} \beta_{qm} f_m(e^{j\omega}), \quad q = 1, 2, \ldots, Q \tag{72}$$

In this representation, the scalars $\{\beta_{qm}\}$ denote the coefficients of the basis expansion for user q. The variable $\omega \in [-\pi, \pi]$ denotes the normalized angular frequency measured in radians/sample. The power spectrum is often symmetric about the vertical axis, $\omega = 0$, and therefore it is sufficient to focus on the interval $\omega \in [0, \pi]$. There are many ways by which the basis functions, $\{f_m(e^{j\omega})\}$, can be selected. The following is one possible construction for illustration purposes. We divide the interval $[0, \pi]$ into B identical intervals and denote their center frequencies by $\{\omega_m\}$. We then place a Gaussian pulse at each location ω_m and control its width through the selection of its standard deviation, σ_m, i.e.,

$$f_m(e^{j\omega}) \triangleq e^{-\frac{(\omega-\omega_m)^2}{\sigma_m^2}} \tag{73}$$

Figure 7 illustrates this construction. The parameters $\{\omega_m, \sigma_m\}$ are selected by the designer and are assumed to be known. For a sufficiently large number, B, of basis functions, the representation (72) can approximate well a large class of power spectra.

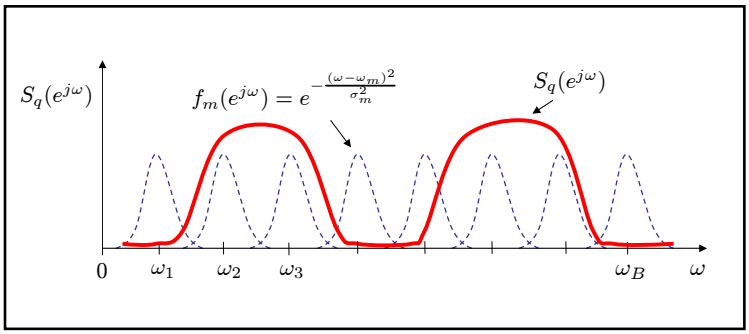

Figure 7: The interval $[0, \pi]$ is divided into B sub-intervals of equal width; the center frequencies of the sub-intervals are denoted by $\{\omega_m\}$. A power spectrum $S_q(e^{j\omega})$ is approximated as a linear combination of Gaussian basis functions centered on the $\{\omega_m\}$.

We collect the combination coefficients $\{\beta_{qm}\}$ for primary user q into a column vector w_q:

$$w_q \triangleq \text{col}\{\beta_{q1}, \beta_{q2}, \beta_{q3}, \ldots, \beta_{qB}\} \qquad (B \times 1) \tag{74}$$

and collect the basis functions into a row vector:

$$f_\omega \triangleq \begin{bmatrix} f_1(e^{j\omega}) & f_2(e^{j\omega}) & \ldots & f_B(e^{j\omega}) \end{bmatrix} \qquad (1 \times B) \tag{75}$$

Then, the power spectrum (72) can be expressed in the alternative inner-product form:

$$S_q(e^{j\omega}) = f_\omega w_q \tag{76}$$

Let p_{qk} denote the path loss coefficient from primary user q to secondary user k. When the transmitted spectrum $S_q(e^{j\omega})$ travels from primary user q to secondary user k, the spectrum that is sensed by node k is $p_{qk}S_q(e^{j\omega})$. We assume in this example that the path loss factors $\{p_{qk}\}$ are known and that they have been determined during a prior training stage involving each of the primary users with each of the secondary users. The training is usually repeated at regular intervals of time to accommodate the fact that the path loss coefficients can vary (albeit slowly) over time. Figure 8 depicts a cognitive radio system with 2 primary users and 10 secondary users. One of the secondary users (user 5) is highlighted and the path loss coefficients from the primary users to its location are indicated; similar path loss coefficients can be assigned to all other combinations involving primary and secondary users.

Each user k senses the *aggregate* effect of the power spectra that are transmitted by all active primary users. Therefore, adding the effect of all primary users, we find that the power spectrum that arrives at secondary user k is given by:

$$\begin{aligned} S_k(e^{j\omega}) &= \sum_{q=1}^{Q} p_{qk} S_q(e^{j\omega}) + \sigma_k^2 \\ &= \sum_{q=1}^{Q} p_{qk} f_\omega w_q + \sigma_k^2 \\ &\triangleq u_{k,\omega} w^o + \sigma_k^2 \end{aligned} \tag{77}$$

where σ_k^2 denotes the receiver noise power at node k, and where we introduced the following vector quantities:

$$w^o \triangleq \text{col}\{w_1, w_2, \ldots, w_Q\} \qquad (BQ \times 1) \tag{78}$$

$$u_{k,\omega} \triangleq \begin{bmatrix} p_{1k} f_\omega & p_{2k} f_\omega & \ldots & p_{Qk} f_\omega \end{bmatrix} \qquad (1 \times BQ) \tag{79}$$

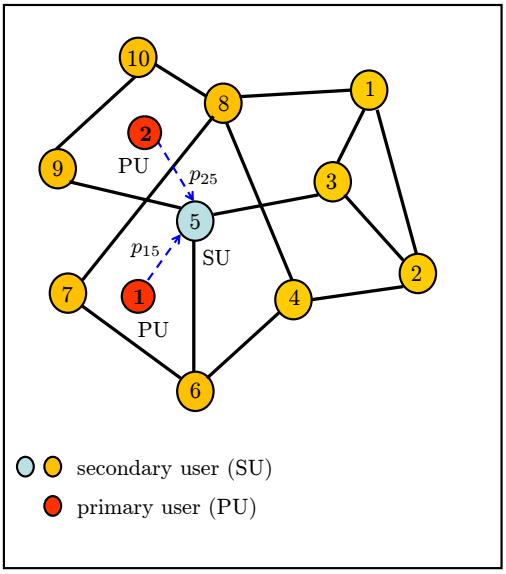

Figure 8: A network of secondary users in the presence of two primary users. One of the secondary users is highlighted and the path loss coefficients from the primary users to its location are indicated as p_{15} and p_{25}.

The vector w^o is the collection of all combination coefficients for all Q primary users. The vector $u_{k,\omega}$ contains the path loss coefficients from all primary users to user k. Now, at every time instant i, user k observers its received power spectrum, $S_k(e^{j\omega})$, over a discrete grid of frequencies, $\{\omega_r\}$, in the interval $[0, \pi]$ in the presence of additive measurement noise. We denote these measurements by:

$$\begin{aligned} \boldsymbol{d}_{k,r}(i) &= \boldsymbol{u}_{k,\omega_r} w^o + \sigma_k^2 + \boldsymbol{v}_{k,r}(i) \\ & \quad r = 1, 2, \ldots, R \end{aligned} \tag{80}$$

The term $\boldsymbol{v}_{k,r}(i)$ denotes sampling noise and is assumed to have zero mean and variance $\sigma_{v,k}^2$; it is also assumed to be temporally white and spatially independent; and is also independent of all other random variables. Since the row vectors $u_{k,\omega}$ in (79) are defined in terms of the path loss coefficients $\{p_{qk}\}$, and since these coefficients are estimated and subject to noisy distortions, we model the $\boldsymbol{u}_{k,\omega_r}$ as zero-mean random variables in (80) and use the boldface notation for them.

Observe that in this application, each node k collects R measurements at every time instant i and not only a single measurement, as was the case with the three examples discussed in the previous sections (AR modeling, MA modeling, and localization). The implication of this fact is that we now deal with an estimation problem that involves vector measurements instead of scalar measurements at each node. The solution structure continues to be the same. We collect the R measurements at node k at time i into vectors and introduce the $R \times 1$ quantities:

$$\boldsymbol{d}_{k,i} \triangleq \begin{bmatrix} \boldsymbol{d}_{k,1}(i) - \sigma_k^2 \\ \boldsymbol{d}_{k,2}(i) - \sigma_k^2 \\ \vdots \\ \boldsymbol{d}_{k,R}(i) - \sigma_k^2 \end{bmatrix}, \qquad \boldsymbol{v}_{k,i} \triangleq \begin{bmatrix} \boldsymbol{v}_{k,1}(i) \\ \boldsymbol{v}_{k,2}(i) \\ \vdots \\ \boldsymbol{v}_{k,R}(i) \end{bmatrix} \tag{81}$$

and the regression matrix:

$$\boldsymbol{U}_{k,i} \triangleq \begin{bmatrix} \boldsymbol{u}_{k,\omega_1} \\ \boldsymbol{u}_{k,\omega_2} \\ \vdots \\ \boldsymbol{u}_{k,\omega_R} \end{bmatrix} \qquad (R \times QB) \tag{82}$$

The time subscript in $\boldsymbol{U}_{k,i}$ is used to model the fact that the path loss coefficients can change over time due to the possibility of node mobility. With the above notation, expression (80) is equivalent to the linear model:

$$\boxed{\boldsymbol{d}_{k,i} = \boldsymbol{U}_{k,i} w^o + \boldsymbol{v}_{k,i}} \tag{83}$$

Compared to the earlier examples (9), (44), and (58), the main difference now is that each agent k collects a *vector* of measurements, $\boldsymbol{d}_{k,i}$, as opposed to the scalar $\boldsymbol{d}_k(i)$, and its regression data are represented by the matrix quantity, $\boldsymbol{U}_{k,i}$, as opposed to the row vector $\boldsymbol{u}_{k,i}$. Nevertheless, the estimation approach will continue to be the same. In cognitive network applications, the secondary users are interested in estimating the aggregate power spectrum of the primary users in order for the secondary users to identify vacant frequency bands that can be used by them. In the context of model (83), this amounts to determining the parameter vector w^o since knowledge of its entries allows each secondary user to reconstruct the aggregate power spectrum defined by:

$$S_A(e^{j\omega}) \triangleq \sum_{q=1}^{Q} S_q(e^{j\omega}) = (\mathbb{1}_Q^T \otimes f_\omega) w^o \tag{84}$$

where the notation $\otimes$ denotes the Kronecker product operation, and $\mathbb{1}_Q$ denotes a $Q \times 1$ vector whose entries are all equal to one.

As before, we can again verify that, in view of (83), the desired parameter vector w^o satisfies the same normal equations:

$$R_{dU,k} = R_{U,k} w^o \iff w^o = R_{U,k}^{-1} R_{dU,k} \tag{85}$$

where the moments $\{R_{dU,k}, R_{U,k}\}$ are now defined by

$$R_{dU,k} \triangleq \mathbb{E}\, \boldsymbol{U}_{k,i}^* \boldsymbol{d}_{k,i} \qquad (QB \times 1) \tag{86}$$

$$R_{U,k} \triangleq \mathbb{E}\, \boldsymbol{U}_{k,i}^* \boldsymbol{U}_{k,i} \qquad (QB \times QB) \tag{87}$$

Therefore, each secondary user k can determine w^o on its own by solving the following minimum mean-square-error estimation problem:

$$\min_{w} \mathbb{E}\, \|\boldsymbol{d}_{k,i} - \boldsymbol{U}_{k,i} w\|^2 \tag{88}$$

This solution method requires knowledge of the moments $\{R_{dU,k}, R_{U,k}\}$ and, in an argument similar to the one that led to (20), it can be verified that each agent k would attain an MSE performance level that is equal to the noise power level, $\sigma_{v,k}^2$, at its location.

Alternatively, when the statistical information $\{R_{dU,k}, R_{U,k}\}$ is not available, each secondary user k can estimate w^o iteratively by feeding data $\{\boldsymbol{d}_{k,i}, \boldsymbol{U}_{k,i}\}$ into an adaptive implementation similar to (26)–(27), such as the following vector LMS recursion:

$$\boldsymbol{e}_{k,i} = \boldsymbol{d}_{k,i} - \boldsymbol{U}_{k,i} \boldsymbol{w}_{k,i-1} \tag{89}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{U}_{k,i}^* \boldsymbol{e}_{k,i} \tag{90}$$

In this case, each secondary user k will achieve the same performance levels shown earlier in (36)–(37) with $R_{u,k}$ replaced by $R_{U,k}$. The performance will again be dependent on the local noise level, $\sigma_{v,k}^2$. As a result, secondary users with larger noise power will perform worse than secondary users with smaller noise power. However, since all secondary users are observing data arising from the same underlying model w^o, it is natural to expect cooperation among the users to be beneficial. As we are going to see, starting from the

next section, one way to achieve cooperation and improve performance is by developing algorithms that solve the following global cost function in an adaptive and distributed manner:

$$\min_{w} \sum_{k=1}^{N} \mathbb{E}\, \|\boldsymbol{d}_{k,i} - \boldsymbol{U}_{k,i} w\|^2 \tag{91}$$

3 Distributed Optimization via Diffusion Strategies

The examples in the previous section were meant to illustrate how MSE cost functions and linear models are useful design tools and how they arise frequently in applications. We now return to problem (1) and study the distributed optimization of global cost functions such as (39), where $J^{\rm glob}(w)$ is assumed to consist of the sum of individual components. Specifically, we are now interested in solving optimization problems of the type:

$$\min_{w} \sum_{k=1}^{N} J_k(w) \tag{92}$$

where each $J_k(w)$ is assumed to be differentiable and convex over w. Although the algorithms presented in this article apply to more general situations, we shall nevertheless focus on mean-square-error cost functions of the form:

$$\boxed{J_k(w) \;\triangleq\; \mathbb{E}\,|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} w|^2} \tag{93}$$

where w is an $M \times 1$ column vector, and the random processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ are assumed to be jointly wide-sense stationary with zero-mean and second-order moments:

$$\sigma_{d,k}^2 \;\triangleq\; \mathbb{E}\,|\boldsymbol{d}_k(i)|^2 \tag{94}$$

$$R_{u,k} \;\triangleq\; \mathbb{E}\,\boldsymbol{u}_{k,i}^* \boldsymbol{u}_{k,i} > 0 \qquad (M \times M) \tag{95}$$

$$r_{du,k} \;\triangleq\; \mathbb{E}\,\boldsymbol{d}_k(i) \boldsymbol{u}_{k,i}^* \qquad (M \times 1) \tag{96}$$

It is clear that each $J_k(w)$ is quadratic in w since, after expansion, we get

$$\boxed{J_k(w) \;=\; \sigma_{d,k}^2 - w^* r_{du,k} - r_{du,k}^*\, w + w^* R_{u,k}\, w} \tag{97}$$

A completion-of-squares argument shows that $J_k(w)$ can be expressed as the sum of two squared terms, i.e.,

$$J_k(w) \;=\; \left(\sigma_{d,k}^2 - r_{du,k}^* R_{u,k}^{-1} r_{du,k}\right) \;+\; (w - w^o)^* R_{u,k} (w - w^o) \tag{98}$$

or, more compactly,

$$\boxed{J_k(w) \;=\; J_{k,\min} \;+\; \|w - w^o\|_{R_{u,k}}^2} \tag{99}$$

where w^o denotes the minimizer of $J_k(w)$ and is given by

$$w^o \;\triangleq\; R_{u,k}^{-1}\, r_{du,k} \tag{100}$$

and $J_{k,\min}$ denotes the minimum value of $J_k(w)$ when evaluated at $w = w^o$:

$$\boxed{J_{k,\min} \;\triangleq\; \sigma_{d,k}^2 - r_{du,k}^* R_{u,k}^{-1} r_{du,k} \;=\; J_k(w^o)} \tag{101}$$

Observe that this value is necessarily non-negative since it can be viewed as the Schur complement of the following covariance matrix:

$$\mathbb{E}\left(\begin{bmatrix} \boldsymbol{d}_k^*(i) \\ \boldsymbol{u}_{k,i}^* \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_k(i) & \boldsymbol{u}_{k,i} \end{bmatrix}\right) = \begin{bmatrix} \sigma_{d,k}^2 & r_{du,k}^* \\ r_{du,k} & R_{u,k} \end{bmatrix} \tag{102}$$

and covariance matrices are nonnegative-definite.

The choice of the quadratic form (93) or (97) for $J_k(w)$ is useful for many applications, as was already illustrated in the previous section for examples involving AR modeling, MA modeling, localization, and spectral sensing. Other choices for $J_k(w)$ are of course possible and these choices can even be different for different nodes. It is sufficient in this article to illustrate the main concepts underlying diffusion adaptation by focusing on the useful case of MSE cost functions of the form (97); still, most of the derivations and arguments in the coming sections can be extended beyond MSE optimization to more general cost functions — as already shown in [1–3]; see also Sec. 10.4.

The positive-definiteness of the covariance matrices $\{R_{u,k}\}$ ensures that each $J_k(w)$ in (97) is strictly convex, as well as $J^{\rm glob}(w)$ from (39). Moreover, all these cost functions have a unique minimum at the same w^o, which satisfies the normal equations:

$$R_{u,k}\, w^o = r_{du,k}, \qquad \text{for every } k = 1, 2, \ldots, N \tag{103}$$

Therefore, given knowledge of $\{r_{du,k}, R_{u,k}\}$, each node can determine w^o on its own by solving (103). One then wonders about the need to seek distributed cooperative and adaptive solutions. There are a couple of reasons:

(a) First, even for MSE cost functions, it is often the case that the required moments $\{r_{du,k}, R_{u,k}\}$ are not known beforehand. In this case, the optimal w^o cannot be determined from the solution of the normal equations (103). The alternative methods that we shall describe will lead to adaptive techniques that enable each node k to estimate w^o directly from data realizations.

(b) Second, since adaptive strategies rely on instantaneous data, these strategies possess powerful tracking abilities. Even when the moments vary with time due to non-stationary behavior (such as w^o changing with time), these changes will be reflected in the observed data and will in turn influence the behavior of the adaptive construction. This is one of the key advantages of adaptive strategies: they enable learning and tracking in real-time.

(c) Third, cooperation among nodes is generally beneficial. When nodes act individually, their performance is limited by the noise power level at their location. In this way, some nodes can perform significantly better than other nodes. On the other hand, when nodes cooperate with their neighbors and share information during the adaptation process, we will see that performance can be improved across the network.

3.1 Relating the Global Cost to Neighborhood Costs

Let us therefore consider the optimization of the following global cost function:

$$\boxed{J^{\rm glob}(w) = \sum_{k=1}^{N} J_k(w)} \tag{104}$$

where $J_k(w)$ is given by (93) or (97). Our strategy to optimize $J^{\rm glob}(w)$ in a distributed manner is based on two steps, following the developments in [1, 2, 18]. First, using a completion-of-squares argument (or, equivalently, a second-order Taylor series expansion), we approximate the global cost function (104) by an alternative local cost that is amenable to distributed optimization. Then, each node will optimize the alternative cost via a steepest-descent method.

To motivate the distributed diffusion-based approach, we start by introducing a set of nonnegative coefficients $\{c_{k\ell}\}$ that satisfy two conditions:

$$\text{for } k = 1, 2, \ldots, N:$$
$$c_{k\ell} \geq 0, \quad \sum_{\ell=1}^{N} c_{k\ell} = 1, \quad \text{and} \quad c_{k\ell} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{105}$$

where $\mathcal{N}_k$ denotes the neighborhood of node k. Condition (105) means that for every node k, the sum of the coefficients $\{c_{k\ell}\}$ that relate it to its neighbors is one. The coefficients $\{c_{k\ell}\}$ are free parameters that are chosen by the designer; obviously, as shown later in Theorem 6.8, their selection will have a bearing on the performance of the resulting algorithms. If we collect the entries $\{c_{k\ell}\}$ into an $N \times N$ matrix C, so that the $k-$th row of C is formed of $\{c_{k\ell},\ \ell = 1, 2, \ldots, N\}$, then condition (105) translates into saying that each of *row* of C adds up to one, i.e.,

$$\boxed{C\mathbb{1} = \mathbb{1}} \tag{106}$$

where the notation $\mathbb{1}$ denotes an $N \times 1$ column vector with all its entries equal to one:

$$\mathbb{1} \triangleq \text{col}\{1, 1, \ldots, 1\} \tag{107}$$

We say that C is a right stochastic matrix. Using the coefficients $\{c_{k\ell}\}$ so defined, we associate with each node ℓ, a local cost function of the following form:

$$J_\ell^{\text{loc}}(w) \triangleq \sum_{k \in \mathcal{N}_\ell} c_{k\ell}\, J_k(w) \tag{108}$$

This cost consists of a weighted combination of the individual costs of the neighbors of node ℓ (including ℓ itself) — see Fig. 9. Since the $\{c_{k\ell}\}$ are all nonnegative and each $J_k(w)$ is strictly convex, then $J_\ell^{\text{loc}}(w)$ is also strictly convex and its minimizer occurs at the same $w = w^o$. Using the alternative representation (99) for the individual $J_k(w)$, we can re-express the local cost $J_\ell^{\text{loc}}(w)$ as

$$J_\ell^{\text{loc}}(w) \;=\; \sum_{k \in \mathcal{N}_\ell} c_{k\ell}\, J_{k,\min} \;+\; \sum_{k \in \mathcal{N}_\ell} c_{k\ell}\, \|w - w^o\|^2_{R_{u,k}} \tag{109}$$

or, equivalently,

$$\boxed{J_\ell^{\text{loc}}(w) \;=\; J_{\ell,\min}^{\text{loc}} \;+\; \|w - w^o\|^2_{R_\ell}} \tag{110}$$

where $J_{\ell,\min}^{\text{loc}}$ corresponds to the minimum value of $J_\ell^{\text{loc}}(w)$ at the minimizer $w = w^o$:

$$J_{\ell,\min}^{\text{loc}} \triangleq \sum_{k \in \mathcal{N}_\ell} c_{k\ell}\, J_{k,\min} \tag{111}$$

and R_ℓ is a positive-definite weighting matrix defined by:

$$R_\ell \triangleq \sum_{k \in \mathcal{N}_\ell} c_{k\ell} R_{u,k} \tag{112}$$

That is, R_ℓ is a weighted combination of the covariance matrices in the neighborhood of node ℓ. Equality (110) amounts to a (second-order) Taylor series expansion of $J_\ell^{\text{loc}}(w)$ around $w = w^o$. Note that the right-hand side consists of two terms: the minimum cost and a weighted quadratic term in the difference $(w - w^o)$.

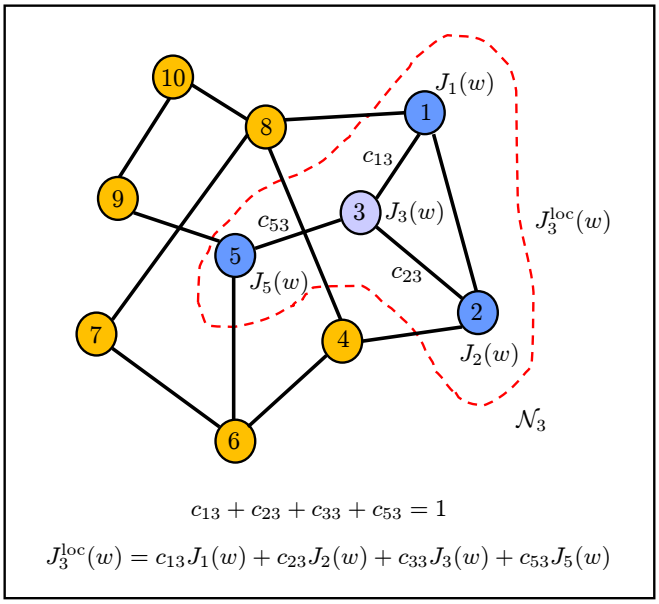

Figure 9: A network with $N = 10$ nodes. The nodes in the neighborhood of node 3 are highlighted with their individual cost functions, and with the combination weights $\{c_{13}, c_{23}, c_{53}\}$ along the connecting edges; there is also a combination weight associated with node 3 and is denoted by c_{33}. The expression for the local cost function, $J_3^{\rm loc}(w)$, is also shown in the figure.

Now note that we can express $J^{\rm glob}(w)$ from (104) as follows:

$$\begin{aligned} J^{\rm glob}(w) &\overset{(105)}{=} \sum_{k=1}^{N}\left(\sum_{\ell=1}^{N} c_{k\ell}\right) J_k(w) \\ &= \sum_{\ell=1}^{N}\left(\sum_{k=1}^{N} c_{k\ell}\, J_k(w)\right) \\ &\overset{(108)}{=} \sum_{\ell=1}^{N} J_\ell^{\rm loc}(w) \\ &= J_k^{\rm loc}(w) + \sum_{\ell\neq k}^{N} J_\ell^{\rm loc}(w) \end{aligned} \tag{113}$$

Substituting (110) into the second term on the right-hand side of the above expression gives:

$$J^{\rm glob}(w) = J_k^{\rm loc}(w) + \sum_{\ell\neq k} \|w - w^o\|_{R_\ell}^2 + \sum_{\ell\neq k} J_{\ell,\min}^{\rm loc} \tag{114}$$

The last term in the above expression does not depend on w. Therefore, minimizing $J^{\rm glob}(w)$ over w is equivalent to minimizing the following alternative global cost:

$$\boxed{J^{\rm glob'}(w) = J_k^{\rm loc}(w) + \sum_{\ell\neq k} \|w - w^o\|_{R_\ell}^2} \tag{115}$$

Expression (115) relates the optimization of the original global cost function, $J^{\rm glob}(w)$ or its equivalent $J^{\rm glob'}(w)$, to the newly-introduced local cost function $J_k^{\rm loc}(w)$. The relation is through the second term on the right-hand side of (115), which corresponds to a sum of quadratic factors involving the minimizer w^o; this term tells us how the local cost $J_k^{\rm loc}(w)$ can be corrected to the global cost $J^{\rm glob'}(w)$. Obviously, the minimizer w^o that appears in the correction term is not known since the nodes wish to determine its value. Likewise, not all the weighting matrices R_ℓ are available to node k; only those matrices that originate from its neighbors can be assumed to be available. Still, expression (115) suggests a useful way to replace $J_k^{\rm loc}$ by another local cost that is closer to $J^{\rm glob'}(w)$. This alternative cost will be shown to lead to a powerful distributed solution to optimize $J^{\rm glob}(w)$ through localized interactions.

Our first step is to limit the summation on the right-hand side of (115) to the neighbors of node k (since every node k can only have access to information from its neighbors). We thus introduce the modified cost function at node k:

$$J_k^{\rm glob'}(w) \;\stackrel{\Delta}{=}\; J_k^{\rm loc}(w) + \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} \|w-w^o\|^2_{R_\ell} \tag{116}$$

The cost functions $J_k^{\rm loc}(w)$ and $J_k^{\rm glob'}(w)$ are both associated with node k; the difference between them is that the expression for the latter is closer to the global cost function (115) that we want to optimize.

The weighting matrices $\{R_\ell\}$ that appear in (116) may or may not be available because the second-order moments $\{R_{u,\ell}\}$ may or may not be known beforehand. If these moments are known, then we can proceed with the analysis by assuming knowledge of the $\{R_\ell\}$. However, the more interesting case is when these moments are not known. This is generally the case in practice, especially in the context of adaptive solutions and problems involving non-stationary data. Often, nodes can only observe realizations $\{u_{\ell,i}\}$ of the regression data $\{\boldsymbol{u}_{\ell,i}\}$ arising from distributions whose covariance matrices are the unknown $\{R_{u,\ell}\}$. One way to address the difficulty is to replace each of the weighted norms $\|w-w^o\|^2_{R_\ell}$ in (116) by a scaled multiple of the un-weighted norm, say,

$$\|w-w^o\|^2_{R_\ell} \;\approx\; b_{\ell k}\cdot\|w-w^o\|^2 \tag{117}$$

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

$$R_\ell \approx b_{\ell k}\, I_M \tag{118}$$

Approximation (117) is reasonable in view of the fact that all vector norms are equivalent [19–21]; this norm property ensures that we can bound the weighted norm $\|w-w^o\|^2_{R_\ell}$ by some constants multiplying the un-weighted norm $\|w-w^o\|^2$, say, as:

$$r_1\|w-w^o\|^2 \;\le\; \|w-w^o\|^2_{R_\ell} \;\le\; r_2\|w-w^o\|^2 \tag{119}$$

for some positive constants (r_1, r_2). Using the fact that the $\{R_\ell\}$ are Hermitian positive-definite matrices, and calling upon the Rayleigh-Ritz characterization of eigenvalues [19, 20], we can be more specific and replace the above inequalities by

$$\lambda_{\min}(R_\ell)\cdot\|w-w^o\|^2 \;\le\; \|w-w^o\|^2_{R_\ell} \;\le\; \lambda_{\max}(R_\ell)\cdot\|w-w^o\|^2 \tag{120}$$

We note that approximations similar to (118) are common in stochastic approximation theory and they mark the difference between using a Newton's iterative method or a stochastic gradient method [5, 22]; the former uses Hessian matrices as approximations for R_ℓ and the latter uses multiples of the identity matrix. Furthermore, as the derivation will reveal, we do not need to worry at this stage about how to select the scalars $\{b_{\ell k}\}$; they will end up being embedded into another set of coefficients $\{a_{\ell k}\}$ that will be set by the designer or adjusted by the algorithm — see (132) further ahead.

Thus, we replace (116) by

$$J_k^{\rm glob''}(w) = J_k^{\rm loc}(w) \;+\; \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} b_{\ell k}\,\|w-w^o\|^2 \tag{121}$$

The argument so far has suggested how to modify $J_k^{\rm loc}(w)$ from (108) and replace it by the cost (121) that is closer in form to the global cost function (115). If we replace $J_k^{\rm loc}(w)$ by its definition (108), we can rewrite (121) as

$$\boxed{J_k^{\rm glob''}(w) \;=\; \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\; J_\ell(w) \;+\; \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} b_{\ell k}\;\|w-w^o\|^2} \tag{122}$$

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

3.2 Steepest-Descent Iterations

Node k can apply a steepest-descent iteration to minimize $J_k^{\rm glob''}(w)$. Let $w_{k,i}$ denote the estimate for the minimizer w^o that is evaluated by node k at time i. Starting from an initial condition $w_{k,-1}$, node k can compute successive estimates iteratively as follows:

$$w_{k,i} \;=\; w_{k,i-1} - \mu_k\;\Big[\nabla_w J_k^{\rm glob''}(w_{k,i-1})\Big]^*, \qquad i\geq 0 \tag{123}$$

where μ_k is a small positive step-size parameter, and the notation $\nabla_w J(a)$ denotes the gradient vector of the function $J(w)$ relative to w and evaluated at $w=a$. The step-size parameter μ_k can be selected to vary with time as well. One choice that is common in the optimization literature [5, 22, 52] is to replace μ_k in (123) by step-size sequences $\{\mu(i)\geq 0\}$ that satisfy the two conditions (25). However, such step-size sequences are not suitable for applications that require continuous learning because they turn off adaptation as $i\to\infty$; the steepest-descent iteration (123) would stop updating since $\mu_k(i)$ would be tending towards zero. For this reason, we shall focus mainly on the constant step-size case described by (123) since we are interested in developing distributed algorithms that will endow networks with continuous adaptation abilities.

Returning to (123) and computing the gradient vector of (122) we get:

$$w_{k,i} = w_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\;[\nabla_w J_\ell(w_{k,i-1})]^* \;-\; \mu_k \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} b_{\ell k}\;(w_{k,i-1}-w^o) \tag{124}$$

Using the expression for $J_\ell(w)$ from (97) we arrive at

$$w_{k,i} \;=\; w_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\;(r_{du,\ell} - R_{u,\ell}\;w_{k,i-1}) \;+\; \mu_k \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} b_{\ell k}\;(w^o - w_{k,i-1}) \tag{125}$$

This iteration indicates that the update from $w_{k,i-1}$ to $w_{k,i}$ involves adding two correction terms to $w_{k,i-1}$. Among many other forms, we can implement the update in two successive steps by adding one correction term at a time, say, as follows:

$$\psi_{k,i} \;=\; w_{k,i-1} \;+\; \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\;(r_{du,\ell} - R_{u,\ell}\;w_{k,i-1}) \tag{126}$$

$$w_{k,i} \;=\; \psi_{k,i} \;+\; \mu_k \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} b_{\ell k}\;(w^o - w_{k,i-1}) \tag{127}$$

Step (126) updates $w_{k,i-1}$ to an intermediate value $\psi_{k,i}$ by using local gradient vectors from the neighborhood of node k. Step (127) further updates $\psi_{k,i}$ to $w_{k,i}$. However, this second step is not realizable since w^o is not known and the nodes are actually trying to estimate it. Two issues stand out from examining (127):

(a) First, iteration (127) requires knowledge of the minimizer w^o. Neither node k nor its neighbors know the value of the minimizer; each of these nodes is actually performing steps similar to (126) and (127)

to estimate the minimizer. However, each node ℓ has a readily available approximation for w^o, which is its local intermediate estimate $\psi_{\ell,i}$. Therefore, we replace w^o in (127) by $\psi_{\ell,i}$. This step helps diffuse information throughout the network. This is because each neighbor of node k determines its estimate $\psi_{\ell,i}$ by processing information from its own neighbors, which process information from their neighbors, and so forth.

(b) Second, the intermediate value $\psi_{k,i}$ at node k is generally a better estimate for w^o than $w_{k,i-1}$ since it is obtained by incorporating information from the neighbors through the first step (126). Therefore, we further replace $w_{k,i-1}$ in (127) by $\psi_{k,i}$. This step is reminiscent of incremental-type approaches to optimization, which have been widely studied in the literature [23–26].

With the substitutions described in items (a) and (b) above, we replace the second step (127) by

$$\begin{aligned} w_{k,i} &= \psi_{k,i} + \mu_k \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell k} \, (\psi_{\ell,i} - \psi_{k,i}) \\ &= \left(1 - \mu_k \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell k}\right) \psi_{k,i} + \mu_k \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell k} \, \psi_{\ell,i} \end{aligned} \tag{128}$$

Introduce the weighting coefficients:

$$a_{kk} \triangleq 1 - \mu_k \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell k} \tag{129}$$

$$a_{\ell k} \triangleq \mu_k b_{\ell k}, \quad \ell \in \mathcal{N}_k \backslash \{k\} \tag{130}$$

$$a_{\ell k} \triangleq 0, \qquad \ell \notin \mathcal{N}_k \tag{131}$$

and observe that, for sufficiently small step-sizes μ_k, these coefficients are nonnegative and, moreover, they satisfy the conditions:

$$\text{for } k = 1, 2, \ldots, N:$$

$$a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad \text{and} \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{132}$$

Condition (132) means that for every node k, the sum of the coefficients $\{a_{\ell k}\}$ that relate it to its neighbors is one. Just like the $\{c_{\ell k}\}$, from now on, we will treat the coefficients $\{a_{\ell k}\}$ as free weighting parameters that are chosen by the designer according to (132); their selection will also have a bearing on the performance of the resulting algorithms — see Theorem 6.8. If we collect the entries $\{a_{\ell k}\}$ into an $N \times N$ matrix A, such that the $k-$th column of A consists of $\{a_{\ell k}, \ \ell = 1, 2, \ldots, N\}$, then condition (132) translates into saying that each *column* of A adds up to one:

$$\boxed{A^T \mathbb{1} = \mathbb{1}} \tag{133}$$

We say that A is a left stochastic matrix.

3.3 Adapt-then-Combine (ATC) Diffusion Strategy

Using the coefficients $\{a_{\ell k}\}$ so defined, we replace (126) and (128) by the following recursions for $i \geq 0$:

$$\text{(ATC strategy)} \quad \boxed{\begin{aligned} \psi_{k,i} &= w_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \, (r_{du,\ell} - R_{u,\ell} \, w_{k,i-1}) \\ w_{k,i} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \, \psi_{\ell,i} \end{aligned}} \tag{134}$$

for some nonnegative coefficients $\{c_{\ell k}, a_{\ell k}\}$ that satisfy conditions (106) and (133), namely,

$$\boxed{C\mathbb{1} = \mathbb{1}, \quad A^T\mathbb{1} = \mathbb{1}} \tag{135}$$

or, equivalently,

$$\begin{aligned} &\text{for } k = 1, 2, \ldots, N: \\ &c_{\ell k} \geq 0, \quad \sum_{k=1}^{N} c_{\ell k} = 1, \quad c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \\ &a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{aligned} \tag{136}$$

To run algorithm (134), we only need to select the coefficients $\{a_{\ell k}, c_{\ell k}\}$ that satisfy (135) or (136); there is no need to worry about the intermediate coefficients $\{b_{\ell k}\}$ any longer since they have been blended into the $\{a_{\ell k}\}$. The scalars $\{c_{\ell k}, a_{\ell k}\}$ that appear in (134) correspond to weighting coefficients over the edge linking node k to its neighbors $\ell \in \mathcal{N}_k$. Note that two sets of coefficients are used to scale the data that are being received by node k: one set of coefficients, $\{c_{\ell k}\}$, is used in the first step of (134) to scale the moment data $\{r_{du,\ell}, R_{u,\ell}\}$, and a second set of coefficients, $\{a_{\ell k}\}$, is used in the second step of (134) to scale the estimates $\{\psi_{\ell,i}\}$. Figure 10 explains what the entries on the columns and rows of the combination matrices $\{A, C\}$ stand for using an example with $N = 6$ and the matrix C for illustration. When the combination matrix is right-stochastic (as is the case with C), each of its rows would add up to one. On the other hand, when the matrix is left-stochastic (as is the case with A), each of its columns would add up to one.

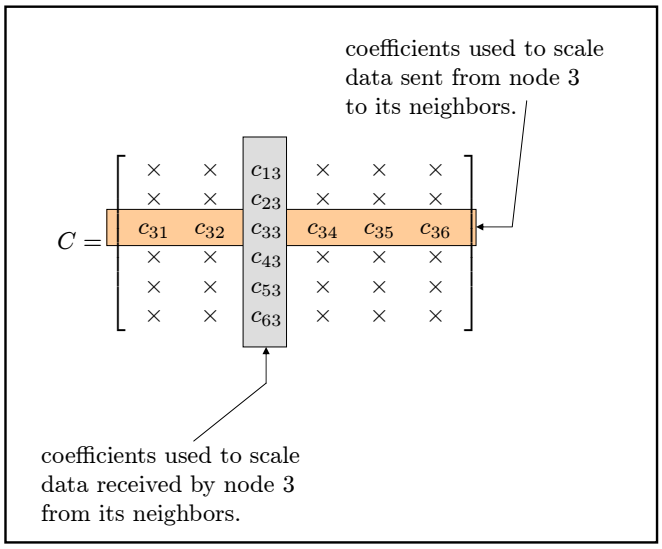

Figure 10: Interpretation of the columns and rows of combination matrices. The pair of entries $\{c_{k\ell}, c_{\ell k}\}$ correspond to weighting coefficients used over the edge connecting nodes k and ℓ. When nodes (k, ℓ) are not neighbors, then these weights are zero.

At every time instant i, the ATC strategy (134) performs two steps. The first step is an *information exchange* step where node k receives from its neighbors their moments $\{R_{u,\ell}, r_{du,\ell}\}$. Node k combines this information and uses it to update its existing estimate $w_{k,i-1}$ to an intermediate value $\psi_{k,i}$. All other nodes in the network are performing a similar step and updating their existing estimates $\{w_{\ell,i-1}\}$ into intermediate estimates $\{\psi_{\ell,i}\}$ by using information from their neighbors. The second step in (134) is an *aggregation* or

consultation step where node k combines the intermediate estimates of its neighbors to obtain its update estimate $w_{k,i}$. Again, all other nodes in the network are simultaneously performing a similar step. The reason for the name Adapt-then-Combine (ATC) strategy is that the first step in (134) will be shown to lead to an adaptive step, while the second step in (134) corresponds to a combination step. Hence, strategy (134) involves adaptation followed by combination or ATC for short. The reason for the qualification "diffusion" is that the combination step in (134) allows information to diffuse through the network in real time. This is because each of the estimates $\psi_{\ell,i}$ is influenced by data beyond the immediate neighborhood of node k.

In the special case when $C = I$, so that no information exchange is performed but only the aggregation step, the ATC strategy (134) reduces to:

$$\text{(ATC strategy without information exchange)} \quad \boxed{\begin{aligned} \psi_{k,i} &= w_{k,i-1} + \mu_k \, (r_{du,k} - R_{u,k} \, w_{k,i-1}) \\ w_{k,i} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \, \psi_{\ell,i} \end{aligned}} \tag{137}$$

where the first step relies solely on the information $\{R_{u,k}, r_{du,k}\}$ that is available locally at node k.

Observe in passing that the term that appears in the information exchange step of (134) is related to the gradient vectors of the local costs $\{J_\ell(w)\}$ evaluated at $w_{k,i-1}$, i.e., it holds that

$$r_{du,\ell} - R_{u,\ell} \, w_{k,i-1} \;=\; -\left[\nabla_w J_\ell(w_{k,i-1})\right]^* \tag{138}$$

so that the ATC strategy (134) can also be written in the following equivalent form:

$$\text{(ATC strategy)} \quad \boxed{\begin{aligned} \psi_{k,i} &= w_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \; \left[\nabla_w J_\ell(w_{k,i-1})\right]^* \\ w_{k,i} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \, \psi_{\ell,i} \end{aligned}} \tag{139}$$

The significance of this general form is that it is applicable to optimization problems involving more general local costs $J_\ell(w)$ that are not necessarily quadratic in w, as detailed in [1–3] — see also Sec. 10.4. The top part of Fig. 11 illustrates the two steps involved in the ATC procedure for a situation where node k has three other neighbors labeled $\{1, 2, \ell\}$. In the first step, node k evaluates the gradient vectors of its neighbors at $w_{k,i-1}$, and subsequently aggregates the estimates $\{\psi_{1,i}, \psi_{2,i}, \psi_{\ell,i}\}$ from its neighbors. The dotted arrows represent flow of information towards node k from its neighbors. The solid arrows represent flow of information from node k to its neighbors. The CTA diffusion strategy is discussed next.

3.4 Combine-then-Adapt (CTA) Diffusion Strategy

Similarly, if we return to (125) and add the second correction term first, then (126)–(127) are replaced by:

$$\psi_{k,i-1} \;=\; w_{k,i-1} \;+\; \mu_k \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell k} \, (w^o - w_{k,i-1}) \tag{140}$$

$$w_{k,i} \;=\; \psi_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \, (r_{du,\ell} - R_{u,\ell} \, w_{k,i-1}) \tag{141}$$

Following similar reasoning to what we did before in the ATC case, we replace w^o in step (140) by $w_{\ell,i-1}$ and replace $w_{k,i-1}$ in (141) by $\psi_{k,i-1}$. We then introduce the same coefficients $\{a_{\ell k}\}$ and arrive at the following

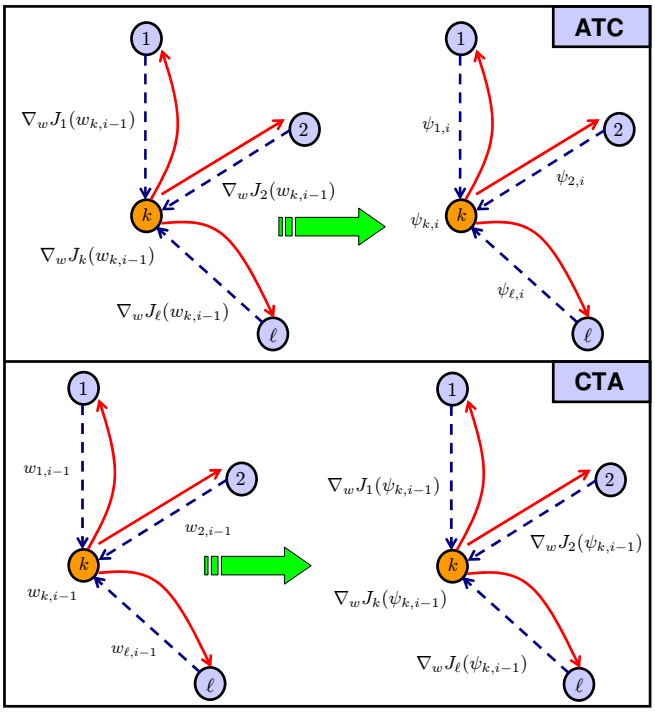

Figure 11: Illustration of the ATC and CTA strategies for a node k with three other neighbors $\{1, 2, \ell\}$. The updates involve two steps: information exchange followed by aggregation in ATC and aggregation followed by information exchange in CTA. The dotted blue arrows represent the data received from the neighbors of node k, and the solid red arrows represent the data sent from node k to its neighbors.

combine-then-adapt (CTA) strategy:

$$\text{(CTA strategy)} \quad \boxed{\begin{aligned} \psi_{k,i-1} &= \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1} \\ w_{k,i} &= \psi_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\,(r_{du,\ell} - R_{u,\ell}\,\psi_{k,i-1}) \end{aligned}} \tag{142}$$

where the nonnegative coefficients $\{c_{\ell k}, a_{\ell k}\}$ satisfy the same conditions (106) and (133), namely,

$$\boxed{C\mathbb{1} = \mathbb{1}, \quad A^T\mathbb{1} = \mathbb{1}} \tag{143}$$

or, equivalently,

$$\begin{aligned} &\text{for } k = 1, 2, \ldots, N: \\ &c_{\ell k} \geq 0, \quad \sum_{k=1}^{N} c_{\ell k} = 1, \quad c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \\ &a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{aligned} \tag{144}$$

At every time instant i, the CTA strategy (142) also consists of two steps. The first step is an aggregation step where node k combines the existing estimates of its neighbors to obtain the intermediate estimate $\psi_{k,i-1}$.

All other nodes in the network are simultaneously performing a similar step and aggregating the estimates of their neighbors. The second step in (142) is an information exchange step where node k receives from its neighbors their moments $\{R_{du,\ell}, r_{du,\ell}\}$ and uses this information to update its intermediate estimate to $w_{k,i}$. Again, all other nodes in the network are simultaneously performing a similar information exchange step. The reason for the name Combine-then-Adapt (CTA) strategy is that the first step in (142) involves a combination step, while the second step will be shown to lead to an adaptive step. Hence, strategy (142) involves combination followed by adaptation or CTA for short. The reason for the qualification "diffusion" is that the combination step of (142) allows information to diffuse through the network in real time.

In the special case when $C = I$, so that no information exchange is performed but only the aggregation step, the CTA strategy (142) reduces to:

$$\text{(CTA strategy without information exchange)} \quad \boxed{\begin{aligned} \psi_{k,i-1} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1} \\ w_{k,i} &= \psi_{k,i-1} + \mu_k\,(r_{du,k} - R_{u,k}\,\psi_{k,i-1}) \end{aligned}} \tag{145}$$

where the second step relies solely on the information $\{R_{u,k}, r_{du,k}\}$ that is available locally at node k. Again, the CTA strategy (142) can be rewritten in terms of the gradient vectors of the local costs $\{J_\ell(w)\}$ as follows:

$$\text{(CTA strategy)} \quad \boxed{\begin{aligned} \psi_{k,i-1} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1} \\ w_{k,i} &= \psi_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k}\; [\nabla_w J_\ell(\psi_{k,i-1})]^* \end{aligned}} \tag{146}$$

The bottom part of Fig. 11 illustrates the two steps involved in the CTA procedure for a situation where node k has three other neighbors labeled $\{1, 2, \ell\}$. In the first step, node k aggregates the estimates $\{w_{1,i-1}, w_{2,i-1}, w_{\ell,i-1}\}$ from its neighbors, and subsequently performs information exchange by evaluating the gradient vectors of its neighbors at $\psi_{k,i-1}$.

3.5 Useful Properties of Diffusion Strategies

Note that the structure of the ATC and CTA diffusion strategies (134) and (142) are fundamentally the same: the difference between the implementations lies in which variable we choose to correspond to the updated weight estimate $w_{k,i}$. In the ATC case, we choose the result of the *combination* step to be $w_{k,i}$, whereas in the CTA case we choose the result of the *adaptation* step to be $w_{k,i}$.

For ease of reference, Table 2 lists the steepest-descent diffusion algorithms derived in the previous sections. The derivation of the ATC and CTA strategies (134) and (142) followed the approach proposed in [18,27]. CTA estimation schemes were first proposed in the works [35–39], and later extended in [18,27,32,33]. The earlier versions of CTA in [35–37] used the choice $C = I$. This form of the algorithm with $C = I$, and with the additional constraint that the step-sizes μ_k should be time-dependent and decay towards zero as time progresses, was later applied by [40,41] to solve distributed optimization problems that require all nodes to reach consensus or agreement. Likewise, special cases of the ATC estimation scheme (134), involving an information exchange step followed by an aggregation step, first appeared in the work [28] on diffusion least-squares schemes and subsequently in the works [18,29–33] on distributed mean-square-error and state-space estimation methods. A special case of the ATC strategy (134) corresponding to the choice $C = I$ with decaying step-sizes was adopted in [34] to ensure convergence towards a consensus state. Diffusion strategies of the form (134) and (142) (or, equivalently, (139) and (146)) are general in several respects:

(1) These strategies do not only diffuse the local weight estimates, but they can also diffuse the local gradient vectors. In other words, two sets of combination coefficients $\{a_{\ell k}, c_{\ell k}\}$ are used.

Table 2: Summary of steepest-descent diffusion strategies for the distributed optimization of general problems of the form (92), and their specialization to the case of mean-square-error (MSE) individual cost functions given by (93).

Algorithm	Recursions	Reference
ATC strategy (general case)	$\psi_{k,i} = w_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\ [\nabla_w J_\ell(w_{k,i-1})]^*$ $w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\ \psi_{\ell,i}$	(139)
ATC strategy (MSE costs)	$\psi_{k,i} = w_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\ (r_{du,\ell} - R_{u,\ell}\ w_{k,i-1})$ $w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\ \psi_{\ell,i}$	(134)
CTA strategy (general case)	$\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\ w_{\ell,i-1}$ $w_{k,i} = \psi_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\ [\nabla_w J_\ell(\psi_{k,i-1})]^*$	(146)
CTA strategy (MSE costs)	$\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\ w_{\ell,i-1}$ $w_{k,i} = \psi_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\ (r_{du,\ell} - R_{u,\ell}\ \psi_{k,i-1})$	(142)

(2) In the derivation that led to the diffusion strategies, the combination matrices C and A are only required to be right-stochastic (for C) and left-stochastic (for A). In comparison, it is common in consensus-type strategies to require the corresponding combination matrix A to be doubly stochastic (i.e., its rows and columns should add up to one) — see, e.g., App. E and [40, 42–44].

(3) As the analysis in Sec. 6 will reveal, ATC and CTA strategies do *not* force nodes to converge to an agreement about the desired parameter vector w^o, as is common in consensus-type strategies (see App. E and [40, 45–51]). Forcing nodes to reach agreement on w^o ends up limiting the adaptation and learning abilities of these nodes, as well as their ability to react to information in real-time. Nodes in diffusion networks enjoy more flexibility in the learning process, which allows their individual estimates, $\{w_{k,i}\}$, to tend to values that lie within a reasonable mean-square-deviation (MSD) level from the optimal solution, w^o. Multi-agent systems in nature behave in this manner; they do not require exact agreement among their agents (see, e.g., [8–10]).

(4) The step-size parameters $\{\mu_k\}$ are not required to depend on the time index i and are not required to vanish as $i \to \infty$ (as is common in many works on distributed optimization, e.g., [22, 40, 52, 53]). Instead, the step-sizes can assume constant values, which is a critical property to endow networks with continuous adaptation and learning abilities. An important contribution in the study of diffusion strategies is to show that distributed optimization is still possible even for constant step-sizes, in addition to the ability to perform adaptation, learning, and tracking. Sections 5 and 6 highlight the convergence properties of the diffusion strategies — see also [1–3] for results pertaining to more general cost functions.

(5) Even the combination weights $\{a_{\ell k}, c_{\ell k}\}$ can be adapted, as we shall discuss later in Sec. 8.3. In this way, diffusion strategies allow multiple layers of adaptation: the nodes perform adaptive processing, the combination weights can be adapted, and even the topology can be adapted especially for mobile networks [8].

4 Adaptive Diffusion Strategies

The distributed ATC and CTA steepest-descent strategies (134) and (142) for determining the w^o that solves (92)–(93) require knowledge of the statistical information $\{R_{u,k}, r_{du,k}\}$. These moments are needed in order to be able to evaluate the gradient vectors that appear in (134) and (142), namely, the terms:

$$-\left[\nabla_w J_\ell(w_{k,i-1})\right]^* = (r_{du,\ell} - R_{u,\ell}\, w_{k,i-1}) \tag{147}$$

$$-\left[\nabla_w J_\ell(\psi_{k,i-1})\right]^* = (r_{du,\ell} - R_{u,\ell}\, \psi_{k,i-1}) \tag{148}$$

for all $\ell \in \mathcal{N}_k$. However, the moments $\{R_{u,\ell}, r_{du,\ell}\}$ are often not available beforehand, which means that the true gradient vectors are generally not available. Instead, the agents have access to observations $\{d_k(i), u_{k,i}\}$ of the random processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$. There are many ways by which the true gradient vectors can be approximated by using these observations. Recall that, by definition,

$$R_{u,\ell} \triangleq \mathbb{E}\,\boldsymbol{u}_{\ell,i}^* \boldsymbol{u}_{\ell,i}, \qquad r_{du,\ell} \triangleq \mathbb{E}\,\boldsymbol{d}_\ell(i)\boldsymbol{u}_{\ell,i}^* \tag{149}$$

One common stochastic approximation method is to drop the expectation operator from the definitions of $\{R_{u,\ell}, r_{du,\ell}\}$ and to use the following instantaneous approximations instead [4–7]:

$$R_{u,\ell} \approx u_{\ell,i}^* u_{\ell,i}, \qquad r_{du,\ell} \approx d_\ell(i) u_{\ell,i}^* \tag{150}$$

In this case, the approximate gradient vectors become:

$$(r_{du,\ell} - R_{u,\ell}\, w_{k,i-1}) \approx u_{\ell,i}^* \left[d_\ell(i) - u_{\ell,i}\, w_{k,i-1}\right] \tag{151}$$

$$(r_{du,\ell} - R_{u,\ell}\, \psi_{k,i-1}) \approx u_{\ell,i}^* \left[d_\ell(i) - u_{\ell,i}\, \psi_{k,i-1}\right] \tag{152}$$

Substituting into the ATC and CTA steepest-descent strategies (134) and (142), we arrive at the following adaptive implementations of the diffusion strategies for $i \geq 0$:

(adaptive ATC strategy)

$$\boxed{\begin{aligned} \psi_{k,i} &= w_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k}\, u_{\ell,i}^* \left[d_\ell(i) - u_{\ell,i} w_{k,i-1}\right] \\ w_{k,i} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, \psi_{\ell,i} \end{aligned}} \tag{153}$$

and

(adaptive CTA strategy)

$$\boxed{\begin{aligned} \psi_{k,i-1} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1} \\ w_{k,i} &= \psi_{k,i-1} + \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k}\, u_{\ell,i}^* \left[d_\ell(i) - u_{\ell,i}\, \psi_{k,i-1}\right] \end{aligned}} \tag{154}$$

where the coefficients $\{a_{\ell k}, c_{\ell k}\}$ are chosen to satisfy:

$$\begin{aligned} &\text{for } k = 1, 2, \ldots, N: \\ &c_{\ell k} \geq 0, \quad \sum_{k=1}^{N} c_{\ell k} = 1, \quad c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \\ &a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{aligned} \tag{155}$$

The adaptive implementations usually start from the initial conditions $w_{\ell,-1} = 0$ for all ℓ, or from some other convenient initial values. Clearly, in view of the approximations (151)–(152), the successive iterates $\{w_{k,i}, \psi_{k,i}, \psi_{k,i-1}\}$ that are generated by the above adaptive implementations are different from the iterates that result from the steepest-descent implementations (134) and (142). Nevertheless, we shall continue to use the same notation for these variables for ease of reference. One key advantage of the adaptive implementations (153)–(154) is that they enable the agents to react to changes in the underlying statistical information $\{r_{du,\ell}, R_{u,\ell}\}$ and to changes in w^o. This is because these changes end up being reflected in the data realizations $\{d_k(i), u_{k,i}\}$. Therefore, adaptive implementations have an innate tracking and learning ability that is of paramount significance in practice.

We say that the stochastic gradient approximations (151)–(152) introduce gradient noise into each step of the recursive updates (153)–(154). This is because the updates (153)–(154) can be interpreted as corresponding to the following forms:

$$\text{(adaptive ATC strategy)} \quad \boxed{\begin{aligned} \psi_{k,i} &= w_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, \big[\widehat{\nabla_w J_\ell(w_{k,i-1})}\big]^* \\ w_{k,i} &= \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \psi_{\ell,i} \end{aligned}} \tag{156}$$

and

$$\text{(adaptive CTA strategy)} \quad \boxed{\begin{aligned} \psi_{k,i-1} &= \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1} \\ w_{k,i} &= \psi_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, \big[\widehat{\nabla_w J_\ell(\psi_{k,i-1})}\big]^* \end{aligned}} \tag{157}$$

where the true gradient vectors, $\{\nabla_w J_\ell(\cdot)\}$, have been replaced by approximations, $\{\widehat{\nabla_w J_\ell(\cdot)}\}$ — compare with (139) and (146). The significance of the alternative forms (156)–(157) is that they are applicable to optimization problems involving more general local costs $J_\ell(w)$ that are not necessarily quadratic, as detailed in [2, 3]; see also Sec. 10.4. In the next section, we examine how gradient noise affects the performance of the diffusion strategies and how close the successive estimates $\{w_{k,i}\}$ get to the desired optimal solution w^o. Table 3 lists several of the adaptive diffusion algorithms derived in this section.

The operation of the adaptive diffusion strategies is similar to the operation of the steepest-descent diffusion strategies of the previous section. Thus, note that at every time instant i, the ATC strategy (153) performs two steps; as illustrated in Fig. 12. The first step is an *information exchange* step where node k receives from its neighbors their information $\{d_\ell(i), u_{\ell,i}\}$. Node k combines this information and uses it to update its existing estimate $w_{k,i-1}$ to an intermediate value $\psi_{k,i}$. All other nodes in the network are performing a similar step and updating their existing estimates $\{w_{\ell,i-1}\}$ into intermediate estimates $\{\psi_{\ell,i}\}$ by using information from their neighbors. The second step in (153) is an *aggregation* or consultation step where node k combines the intermediate estimates $\{\psi_{\ell,i}\}$ of its neighbors to obtain its update estimate $w_{k,i}$. Again, all other nodes in the network are simultaneously performing a similar step. In the special case when $C = I$, so that no information exchange is performed but only the aggregation step, the ATC strategy (153) reduces to:

$$\begin{array}{l}\text{(adaptive ATC strategy}\\ \text{without information exchange)}\end{array} \quad \boxed{\begin{aligned} \psi_{k,i} &= w_{k,i-1} + \mu_k\, u_{k,i}^*\, [d_k(i) - u_{k,i} w_{k,i-1}] \\ w_{k,i} &= \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \psi_{\ell,i} \end{aligned}} \tag{158}$$

Table 3: Summary of adaptive diffusion strategies for the distributed optimization of general problems of the form (92), and their specialization to the case of mean-square-error (MSE) individual cost functions given by (93). These adaptive solutions rely on stochastic approximations.

ALGORITHM	RECURSIONS	REFERENCE
Adaptive ATC strategy (general case)	$\psi_{k,i} = w_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, [\widehat{\nabla_w J_\ell(w_{k,i-1})}]^*$ $w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \psi_{\ell,i}$	(156)
Adaptive ATC strategy (MSE costs)	$\psi_{k,i} = w_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, u^*_{\ell,i}\, [d_\ell(i) - u_{\ell,i} w_{k,i-1}]$ $w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \psi_{\ell,i}$	(153)
Adaptive ATC strategy (MSE costs) (no information exchange)	$\psi_{k,i} = w_{k,i-1} + \mu_k\, u^*_{k,i}\, [d_k(i) - u_{k,i} w_{k,i-1}]$ $w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \psi_{\ell,i}$	(158)
Adaptive CTA strategy (general case)	$\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1}$ $w_{k,i} = \psi_{k,i-1} - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, [\widehat{\nabla_w J_\ell(\psi_{k,i-1})}]^*$	(157)
Adaptive CTA strategy (MSE costs)	$\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1}$ $w_{k,i} = \psi_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, u^*_{\ell,i}\, [d_\ell(i) - u_{\ell,i}\, \psi_{k,i-1}]$	(154)
Adaptive CTA strategy (MSE costs) (no information exchange)	$\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1}$ $w_{k,i} = \psi_{k,i-1} + \mu_k\, u^*_{k,i}\, [d_k(i) - u_{k,i}\, \psi_{k,i-1}]$	(159)

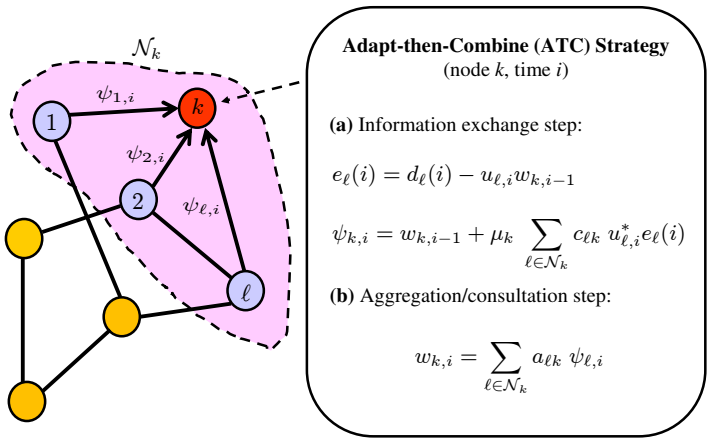

Figure 12: Illustration of the adaptive ATC strategy, which involves two steps: information exchange followed by aggregation.

Likewise, at every time instant i, the CTA strategy (154) also consists of two steps – see Fig. 13. The first step is an aggregation step where node k combines the existing estimates of its neighbors to obtain the intermediate estimate $\psi_{k,i-1}$. All other nodes in the network are simultaneously performing a similar step and aggregating the estimates of their neighbors. The second step in (154) is an information exchange step where node k receives from its neighbors their information $\{d_\ell(i), u_{\ell,i}\}$ and uses this information to update its intermediate estimate to $w_{k,i}$. Again, all other nodes in the network are simultaneously performing a similar information exchange step. In the special case when $C = I$, so that no information exchange is performed but only the aggregation step, the CTA strategy (154) reduces to:

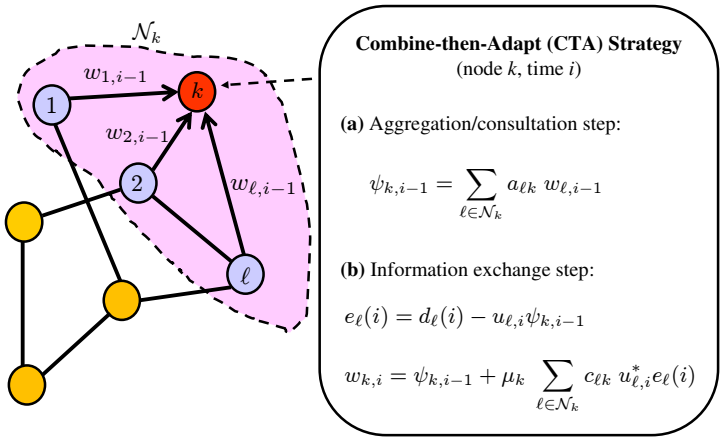

Figure 13: Illustration of the adaptive CTA strategy, which involves two steps: aggregation followed by information exchange.

(adaptive CTA strategy without information exchange)

$$\boxed{\begin{aligned}&\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\; w_{\ell,i-1}\\ &w_{k,i} = \psi_{k,i-1} + \mu_k\; u_{k,i}^{*}\,[d_k(i) - u_{k,i}\;\psi_{k,i-1}]\end{aligned}} \tag{159}$$

We further note that the adaptive ATC and CTA strategies (153)–(154) reduce to the non-cooperative adaptive solution (22)–(23), where each node k runs its own individual LMS filter, when the coefficients $\{a_{\ell k}, c_{\ell k}\}$ are selected as

$$a_{\ell k} = \delta_{\ell k} = c_{\ell k} \qquad \text{(non-cooperative case)} \tag{160}$$

where $\delta_{\ell k}$ denotes the Kronecker delta function:

$$\delta_{\ell k} \;\triangleq\; \left\{\begin{array}{ll} 1, & \ell = k \\ 0, & \text{otherwise}\end{array}\right. \tag{161}$$

In terms of the combination matrices A and C, this situation corresponds to setting

$$\boxed{A = I_N = C} \qquad \text{(non-cooperative case)} \tag{162}$$

5 Performance of Steepest-Descent Diffusion Strategies

Before studying in some detail the mean-square performance of the adaptive diffusion implementations (153)–(154), and the influence of gradient noise, we examine first the convergence behavior of the steepest-descent diffusion strategies (134) and (142), which employ the true gradient vectors. Doing so, will help introduce the necessary notation and highlight some features of the analysis in preparation for the more challenging treatment of the adaptive strategies in Sec. 6.

5.1 General Diffusion Model

Rather than study the performance of the ATC and CTA steepest-descent strategies (134) and (142) separately, it is useful to introduce a more general description that includes the ATC and CTA recursions as special cases. Thus, consider a distributed steepest-descent diffusion implementation of the following general form for $i \geq 0$:

$$\phi_{k,i-1} \;=\; \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\; w_{\ell,i-1} \tag{163}$$

$$\psi_{k,i} \;=\; \phi_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\;\left[r_{du,\ell} - R_{u,\ell}\;\phi_{k,i-1}\right] \tag{164}$$

$$w_{k,i} \;=\; \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\;\psi_{\ell,i} \tag{165}$$

where the scalars $\{a_{1,\ell k}, c_{\ell k}, a_{2,\ell k}\}$ denote three sets of non-negative real coefficients corresponding to the (ℓ, k) entries of $N\times N$ combination matrices $\{A_1, C, A_2\}$, respectively. These matrices are assumed to satisfy the conditions:

$$\boxed{A_1^T\mathbb{1} = \mathbb{1}, \quad C\mathbb{1} = \mathbb{1}, \quad A_2^T\mathbb{1} = \mathbb{1}} \tag{166}$$

so that $\{A_1, A_2\}$ are left stochastic and C is right-stochastic, i.e.,

$$\begin{array}{lll} & & \text{for } k = 1, 2, \ldots, N: \\ c_{\ell k} \geq 0, & \displaystyle\sum_{k=1}^{N} c_{\ell k} = 1, & c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \\ a_{1,\ell k} \geq 0, & \displaystyle\sum_{\ell=1}^{N} a_{1,\ell k} = 1, & a_{1,\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \\ a_{2,\ell k} \geq 0, & \displaystyle\sum_{\ell=1}^{N} a_{2,\ell k} = 1, & a_{2,\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{array} \tag{167}$$

Different choices for $\{A_1, C, A_2\}$ correspond to different cooperation modes. For example, the choice $A_1 = I_N$ and $A_2 = A$ corresponds to the ATC implementation (134), while the choice $A_1 = A$ and $A_2 = I_N$ corresponds to the CTA implementation (142). Likewise, the choice $C = I_N$ corresponds to the case in which the nodes only share weight estimates and the distributed diffusion recursions (163)–(165) become

$$\phi_{k,i-1} = \sum_{\ell \in \mathcal{N}_k} a_{1,\ell k}\, w_{\ell,i-1} \tag{168}$$

$$\psi_{k,i} = \phi_{k,i-1} + \mu_k\, (r_{du,k} - R_{u,k}\, \phi_{k,i-1}) \tag{169}$$

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{2,\ell k}\, \psi_{\ell,i} \tag{170}$$

Furthermore, the choice $A_1 = A_2 = C = I_N$ corresponds to the non-cooperative mode of operation, in which case the recursions reduce to the classical (stand-alone) steepest-descent recursion [4–7], where each node minimizes individually its own quadratic cost $J_k(w)$, defined earlier in (97):

$$w_{k,i} = w_{k,i-1} + \mu_k\, [r_{du,k} - R_{u,k}\, w_{k,i-1}], \quad i \geq 0 \tag{171}$$

Table 4: Different choices for the combination matrices $\{A_1, A_2, C\}$ in (163)–(165) correspond to different cooperation strategies.

A_1	A_2	C	Cooperation Mode
I_N	A	C	ATC strategy (134).
I_N	A	I_N	ATC strategy (137) without information exchange.
A	I_N	C	CTA strategy (142).
A	I_N	I_N	CTA strategy (145) without information exchange.
I_N	I_N	I_N	non-cooperative steepest-descent (171).

5.2 Error Recursions

Our objective is to examine whether, and how fast, the weight estimates $\{w_{k,i}\}$ from the distributed implementation (163)–(165) converge towards the solution w^o of (92)–(93). To do so, we introduce the $M \times 1$ error vectors:

$$\widetilde{\phi}_{k,i} \triangleq w^o - \phi_{k,i} \tag{172}$$

$$\widetilde{\psi}_{k,i} \triangleq w^o - \psi_{k,i} \tag{173}$$

$$\widetilde{w}_{k,i} \triangleq w^o - w_{k,i} \tag{174}$$

Each of these error vectors measures the residual relative to the desired minimizer w^o. Now recall from (100) that

$$r_{du,k} = R_{u,k}\, w^o \tag{175}$$

Then, subtracting w^o from both sides of the relations in (163)–(165) we get

$$\widetilde{\phi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\, \widetilde{w}_{\ell,i-1} \tag{176}$$

$$\widetilde{\psi}_{k,i} = \left(I_M - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, R_{u,\ell}\right)\, \widetilde{\phi}_{k,i-1} \tag{177}$$

$$\widetilde{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\, \widetilde{\psi}_{\ell,i} \tag{178}$$

We can describe these relations more compactly by collecting the information from across the network into block vectors and matrices. We collect the error vectors from across all nodes into the following $N\times 1$ *block* vectors, whose individual entries are of size $M\times 1$ each:

$$\widetilde{\psi}_i \triangleq \begin{bmatrix} \widetilde{\psi}_{1,i} \\ \widetilde{\psi}_{2,i} \\ \vdots \\ \widetilde{\psi}_{N,i} \end{bmatrix}, \qquad \widetilde{\phi}_i \triangleq \begin{bmatrix} \widetilde{\phi}_{1,i} \\ \widetilde{\phi}_{2,i} \\ \vdots \\ \widetilde{\phi}_{N,i} \end{bmatrix}, \qquad \widetilde{w}_i \triangleq \begin{bmatrix} \widetilde{w}_{1,i} \\ \widetilde{w}_{2,i} \\ \vdots \\ \widetilde{w}_{N,i} \end{bmatrix} \tag{179}$$

The block quantities $\{\widetilde{\psi}_i, \widetilde{\phi}_i, \widetilde{w}_i\}$ represent the state of the errors across the network at time i. Likewise, we introduce the following $N\times N$ *block* diagonal matrices, whose individual entries are of size $M\times M$ each:

$$\mathcal{M} \triangleq \mathrm{diag}\{\, \mu_1 I_M,\ \mu_2 I_M,\ \ldots,\ \mu_N I_M \,\} \tag{180}$$

$$\mathcal{R} \triangleq \mathrm{diag}\left\{ \sum_{\ell\in\mathcal{N}_1} c_{\ell 1}\, R_{u,\ell},\ \sum_{\ell\in\mathcal{N}_2} c_{\ell 2}\, R_{u,\ell},\ \ldots,\ \sum_{\ell\in\mathcal{N}_N} c_{\ell N}\, R_{u,\ell} \right\} \tag{181}$$

Each block diagonal entry of $\mathcal{R}$, say, the k-th entry, contains the combination of the covariance matrices in the neighborhood of node k. We can simplify the notation by denoting these neighborhood combinations as follows:

$$\boxed{R_k \triangleq \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, R_{u,\ell}} \tag{182}$$

so that $\mathcal{R}$ becomes

$$\boxed{\mathcal{R} \triangleq \mathrm{diag}\,\{\, R_1,\ R_2,\ \ldots, R_N \,\}} \quad (\text{when } C\neq I) \tag{183}$$

In the special case when $C = I_N$, the matrix $\mathcal{R}$ reduces to

$$\boxed{\mathcal{R}_u = \mathrm{diag}\{R_{u,1}, R_{u,2}, \ldots, R_{u,N}\}} \quad (\text{when } C = I) \tag{184}$$

with the individual covariance matrices appearing on its diagonal; we denote $\mathcal{R}$ by $\mathcal{R}_u$ in this special case. We further introduce the Kronecker products

$$\mathcal{A}_1 \triangleq A_1 \otimes I_M, \qquad \mathcal{A}_2 \triangleq A_2 \otimes I_M \tag{185}$$

The matrix $\mathcal{A}_1$ is an $N \times N$ *block* matrix whose (ℓ, k) block is equal to $a_{1,\ell k} I_M$. Likewise, for $\mathcal{A}_2$. In other words, the Kronecker transformation defined above simply replaces the matrices $\{A_1, A_2\}$ by block matrices $\{\mathcal{A}_1, \mathcal{A}_2\}$ where each entry $\{a_{1,\ell k}, a_{2,\ell k}\}$ in the original matrices is replaced by the diagonal matrices $\{a_{1,\ell k} I_M,\ a_{2,\ell k} I_M\}$. For ease of reference, Table 5 lists the various symbols that have been defined so far, and others that will be defined in the sequel.

Table 5: Definitions of network variables used throughout the analysis.

VARIABLE	EQUATION
$\mathcal{A}_1 = A_1 \otimes I_M$	(185)
$\mathcal{A}_2 = A_2 \otimes I_M$	(185)
$\mathcal{C} = C \otimes I_M$	(245)
$R_k = \sum_{\ell \in \mathcal{N}_k} c_{\ell k}\ R_{u,\ell}$	(182)
$\mathcal{R} = \text{diag}\ \{R_1,\ R_2,\ \ldots, R_N\ \}$	(183)
$\mathcal{R}_u = \text{diag}\{R_{u,1},\ R_{u,2},\ \ldots,\ R_{u,N}\}$	(241)
$R_v = \text{diag}\{\sigma^2_{v,1},\ \sigma^2_{v,2}, \ldots, \sigma^2_{v,N}\}$	(319)
$\mathcal{M} = \text{diag}\{\ \mu_1 I_M,\ \mu_2 I_M,\ \ldots,\ \mu_N I_M\ \}$	(180)
$\mathcal{S} = \text{diag}\{\sigma^2_{v,1} R_{u,1},\ \sigma^2_{v,2} R_{u,2},\ \ldots, \sigma^2_{v,N} R_{u,N}\}$	(241)
$\mathcal{G} = \mathcal{A}_2^T \mathcal{M} \mathcal{C}^T$	(263)
$\mathcal{B} = \mathcal{A}_2^T\ (I_{NM} - \mathcal{M}\mathcal{R})\ \mathcal{A}_1^T$	(264)
$\mathcal{Y} = \mathcal{G}\mathcal{S}\mathcal{G}^T$	(280)
$\mathcal{F} \approx \mathcal{B}^T \otimes \mathcal{B}^*$	(277)
$\mathcal{J}_k = \text{diag}\{\ 0_M, \ldots, 0_M, I_M, 0_M, \ldots, 0_M\ \}$	(294)
$\mathcal{T}_k = \text{diag}\{\ 0_M, \ldots, 0_M, R_{u,k}, 0_M, \ldots, 0_M\ \}$	(298)

Returning to (176)–(178), we conclude that the following relations hold for the block quantities:

$$\widetilde{\phi}_{i-1} = \mathcal{A}_1^T \widetilde{w}_{i-1} \tag{186}$$

$$\widetilde{\psi}_i = (I_{NM} - \mathcal{M}\mathcal{R})\ \widetilde{\phi}_{i-1} \tag{187}$$

$$\widetilde{w}_i = \mathcal{A}_2^T\ \widetilde{\psi}_i \tag{188}$$

so that the network weight error vector, $\widetilde{w}_i$, ends up evolving according to the following dynamics:

$$\boxed{\widetilde{w}_i = \mathcal{A}_2^T\ (I_{NM} - \mathcal{M}\mathcal{R})\ \mathcal{A}_1^T \widetilde{w}_{i-1},\ \ i \geq 0} \quad \text{(diffusion strategy)} \tag{189}$$

For comparison purposes, if each node in the network minimizes its own cost function, $J_k(w)$, separately from the other nodes and uses the non-cooperative steepest-descent strategy (171), then the weight error vector across all N nodes would evolve according to the following alternative dynamics:

$$\boxed{\widetilde{w}_i = \ (I_{NM} - \mathcal{M}\mathcal{R}_u)\ \widetilde{w}_{i-1},\ \ i \geq 0} \quad \text{(non-cooperative strategy)} \tag{190}$$

where the matrices $\mathcal{A}_1$ and $\mathcal{A}_2$ do not appear, and $\mathcal{R}$ is replaced by $\mathcal{R}_u$ from (184). This recursion is a special case of (189) when $A_1 = A_2 = C = I_N$.

5.3 Convergence Behavior

Note from (189) that the evolution of the weight error vector involves block vectors and block matrices; this will be characteristic of the distributed implementations that we consider in this article. To examine the stability and convergence properties of recursions that involve such block quantities, it becomes useful to rely on a certain block vector norm. In App. D, we describe a so-called *block maximum block* and establish some of its useful properties. The results of the appendix will be used extensively in our exposition. It is therefore advisable for the reader to review the properties stated in the appendix at this stage.

Using the result of Lemma D.6, we can establish the following useful statement about the convergence of the steepest-descent diffusion strategy (163)–(165). The result establishes that all nodes end up converging to the optimal solution w^o if the nodes employ positive step-sizes μ_k that are small enough; the lemma provides a sufficient bound on the $\{\mu_k\}$.

Theorem* 5.1.** *(Convergence to Optimal Solution***) Consider the problem of optimizing the global cost (92) with the individual cost functions given by (93). Pick a right stochastic matrix C and left stochastic matrices A_1 and A_2 satisfying (166) or (167); these matrices define the network topology and how information is shared over neighborhoods. Assume each node in the network runs the (distributed) steepest-descent diffusion algorithm (163)–(165). Then, all estimates $\{w_{k,i}\}$ across the network converge to the optimal solution w^o if the positive step-size parameters $\{\mu_k\}$ satisfy*

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_k)}} \tag{191}$$

where the neighborhood covariance matrix R_k is defined by (182).

Proof. The weight error vector $\widetilde{w}_i$ converges to zero if, and only if, the coefficient matrix $\mathcal{A}_2^T\,(I_{NM} - \mathcal{M}\mathcal{R})\,\mathcal{A}_1^T$ in (189) is a stable matrix (meaning that all its eigenvalues lie strictly inside the unit disc). From property (605) established in App. D, we know that $\mathcal{A}_2^T\,(I_{NM} - \mathcal{M}\mathcal{R})\,\mathcal{A}_1^T$ is stable if the block diagonal matrix $(I_{NM} - \mathcal{M}\mathcal{R})$ is stable. It is now straightforward to verify that condition (191) ensures the stability of $(I_{NM} - \mathcal{M}\mathcal{R})$. It follows that

$$\widetilde{w}_i \longrightarrow 0 \quad \text{as} \quad i \longrightarrow \infty \tag{192}$$

□

Observe that the stability condition (191) does not depend on the specific combination matrices A_1 and A_2. Thus, as long as these matrices are chosen to be left-stochastic, the weight-error vectors will converge to zero under condition (191) no matter what $\{A_1, A_2\}$ are. Only the combination matrix C influences the condition on the step-size through the neighborhood covariance matrices $\{R_k\}$. Observe further that the statement of the lemma does not require the network to be connected. Moreover, when $C = I$, in which case the nodes only share weight estimates and do not share the neighborhood moments $\{r_{du,\ell}, R_{u,\ell}\}$, as in (168)–(170), condition (191) becomes

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_{u,k})}} \qquad \text{(cooperation with } C = I\text{)} \tag{193}$$

in terms of the actual covariance matrices $\{R_{u,k}\}$. Results (191) and (193) are reminiscent of a classical result for stand-alone steepest-descent algorithms, as in the non-cooperative case (171), where it is known that the estimate by each individual node in this case will converge to w^o if, and only if, its positive step-size satisfies

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_{u,k})}} \qquad \text{(non-cooperative case (171) with } A_1 = A_2 = C = I_N\text{)} \tag{194}$$

This is the same condition as (193) for the case $C = I$.

The following statement provides a *bi-directional* statement that ensures convergence of the (distributed) steepest-descent diffusion strategy (163)–(165) for *any* choice of left-stochastic combination matrices A_1 and A_2.

Theorem* 5.2.** *(Convergence for Arbitrary Combination Matrices***) Consider the problem of optimizing the global cost (92) with the individual cost functions given by (93). Pick a right stochastic matrix C satisfying (166). Then, the estimates $\{w_{k,i}\}$ generated by (163)–(165) converge to w^o, for all choices of left-stochastic matrices A_1 and A_2 satisfying (166) if, and only if,*

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_k)}} \tag{195}$$

Proof. The result follows from property (b) of Corollary D.1, which is established in App. D. □

More importantly, we can verify that under fairly general conditions, employing the steepest-descent diffusion strategy (163)–(165) enhances the convergence rate of the error vector towards zero relative to the non-cooperative strategy (171). The next three results establish this fact when C is a doubly stochastic matrix, i.e., it has non-negative entries and satisfies

$$C\mathbb{1} = \mathbb{1}, \qquad C^T\mathbb{1} = \mathbb{1} \tag{196}$$

with both its rows and columns adding up to one. Compared to the earlier right-stochastic condition on C in (105), we are now requiring

$$\sum_{\ell\in\mathcal{N}_k} c_{k\ell} = 1, \qquad \sum_{\ell\in\mathcal{N}_k} c_{\ell k} = 1 \tag{197}$$

For example, these conditions are satisfied when C is right stochastic and symmetric. They are also satisfied for $C = I$, when only weight estimates are shared as in (168)–(170); this latter case covers the ATC and CTA diffusion strategies (137) and (145), which do not involve information exchange.

Theorem* 5.3.** *(Convergence Rate is Enhanced: Uniform Step-Sizes***) Consider the problem of optimizing the global cost (92) with the individual cost functions given by (93). Pick a doubly stochastic matrix C satisfying (196) and left stochastic matrices A_1 and A_2 satisfying (166). Consider two modes of operation. In one mode, each node in the network runs the (distributed) steepest-descent diffusion algorithm (163)–(165). In the second mode, each node operates individually and runs the non-cooperative steepest-descent algorithm (171). In both cases, the positive step-sizes used by all nodes are assumed to be the same, say, $\mu_k = \mu$ for all k, and the value of μ is chosen to satisfy the required stability conditions (191) and (194), which are met by selecting*

$$\mu \;<\; \min_{1\le k\le N} \left\{ \frac{2}{\lambda_{\max}(R_{u,k})} \right\} \tag{198}$$

It then holds that the magnitude of the error vector, $\|\widetilde{w}_i\|$, in the diffusion case decays to zero more rapidly than in the non-cooperative case. In other words, diffusion cooperation enhances convergence rate.

Proof. Let us first establish that any positive step-size μ satisfying (198) will satisfy both stability conditions (191) and (194). It is obvious that (194) is satisfied. We verify that (191) is also satisfied when C is doubly stochastic. In this case, each neighborhood covariance matrix, R_k, becomes a convex combination of individual covariance matrices $\{R_{u,\ell}\}$, i.e.,

$$R_k = \sum_{\ell\in\mathcal{N}_k} c_{\ell k} R_{u,\ell}$$

where now

$$\sum_{\ell\in\mathcal{N}_k} c_{\ell k} \;=\; 1 \qquad \text{(when } C \text{ is doubly stochastic)}$$

To proceed, we recall that the spectral norm (maximum singular value) of any matrix X is a convex function of X [56]. Moreover, for Hermitian matrices X, their spectral norms coincide with their spectral radii (largest eigenvalue magnitude). Then, Jensen's inequality [56] states that for any convex function $f(\cdot)$ it holds that

$$f\left(\sum_m \theta_m X_m\right) \;\leq\; \sum_m \theta_m f(X_m)$$

for Hermitian matrices X_m and nonnegative scalars θ_m that satisfy

$$\sum_m \theta_m = 1$$

Choosing $f(\cdot)$ as the spectral radius function, and applying it to the definition of R_k above, we get

$$\begin{aligned}
\rho(R_k) &= \rho\left(\sum_{\ell\in\mathcal{N}_k} c_{\ell k} R_{u,\ell}\right) \\
&\leq \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\cdot\rho(R_{u,\ell}) \\
&\leq \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\cdot\left[\max_{1\leq\ell\leq N}\rho(R_{u,\ell})\right] \\
&= \max_{1\leq\ell\leq N}\rho(R_{u,\ell})
\end{aligned}$$

In other words,

$$\lambda_{\max}(R_k) \;\leq\; \max_{1\leq k\leq N}\;\{\lambda_{\max}(R_{u,k})\}$$

It then follows from (198) that

$$\mu \;<\; \frac{2}{\lambda_{\max}(R_k)}, \quad \text{for all } k=1,2,\ldots,N$$

so that (191) is satisfied as well.

Let us now examine the convergence rate. To begin with, we note that the matrix $(I_{NM}-\mathcal{M}\mathcal{R})$ that appears in the weight-error recursion (189) is block diagonal:

$$(I_{NM}-\mathcal{M}\mathcal{R}) = \mathrm{diag}\{(I_M-\mu R_1),\;(I_M-\mu R_2),\ldots,\;(I_M-\mu R_N)\}$$

and each individual block entry, $(I_M-\mu R_k)$, is a stable matrix since μ satisfies (191). Moreover, each of these entries can be written as

$$I_M-\mu R_k \;=\; \sum_{\ell\in\mathcal{N}_k} c_{\ell k}(I_M-\mu R_{u,\ell})$$

which expresses $(I_M-\mu R_k)$ as a convex combination of stable terms $(I_M-\mu R_{u,\ell})$. Applying Jensen's inequality again we get

$$\rho\left(\sum_{\ell\in\mathcal{N}_k} c_{\ell k}(I_M-\mu R_{u,\ell})\right) \;\leq\; \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\;\rho(I_M-\mu R_{u,\ell})$$

Now, we know from (189) that the rate of decay of $\widetilde{w}_i$ to zero in the diffusion case is determined by the spectral radius of the coefficient matrix $\mathcal{A}_2^T\,(I_{NM}-\mathcal{M}\mathcal{R})\,\mathcal{A}_1^T$. Likewise, we know from (190) that the rate of decay of $\widetilde{w}_i$ to zero in the non-cooperative case is determined by the spectral radius of the coefficient matrix $(I_{NM}-\mathcal{M}\mathcal{R}_u)$. Then, note that

$$
\begin{aligned}
\rho\left(\mathcal{A}_2^T\left(I_{NM}-\mathcal{M}\mathcal{R}\right)\mathcal{A}_1^T\right) &\overset{(605)}{\leq} \rho(I_{NM}-\mathcal{M}\mathcal{R}) \\
&= \max_{1\leq k\leq N}\rho\left(I_M-\mu R_k\right) \\
&= \max_{1\leq k\leq N}\rho\left(\sum_{\ell\in\mathcal{N}_k}c_{\ell k}(I_M-\mu R_{u,\ell})\right) \\
&\leq \max_{1\leq k\leq N}\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\,\rho(I_M-\mu R_{u,\ell}) \\
&\leq \max_{1\leq k\leq N}\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\,\left(\max_{1\leq \ell\leq N}\rho(I_M-\mu R_{u,\ell})\right) \\
&= \max_{1\leq k\leq N}\left\{\left(\max_{1\leq \ell\leq N}\rho(I_M-\mu R_{u,\ell})\right)\cdot\sum_{\ell\in\mathcal{N}_k}c_{\ell k}\right\} \\
&= \max_{1\leq k\leq N}\left(\max_{1\leq \ell\leq N}\rho(I_M-\mu R_{u,\ell})\right) \\
&= \max_{1\leq \ell\leq N}\rho(I_M-\mu R_{u,\ell}) \\
&= \rho(I_{NM}-\mathcal{M}\mathcal{R}_u)
\end{aligned}
$$

Therefore, the spectral radius of $\mathcal{A}_2^T\left(I_{NM}-\mathcal{M}\mathcal{R}\right)\mathcal{A}_1^T$ is at most as large as the largest individual spectral radius in the non-cooperative case.

□

The argument can be modified to handle different step-sizes across the nodes if we assume uniform covariance data across the network, as stated below.

Theorem* 5.4.** *(Convergence Rate is Enhanced: Uniform Covariance Data***) Consider the same setting of Theorem 5.3. Assume the covariance data are uniform across all nodes, say, $R_{u,k}=R_u$ is independent of k. Assume further that the nodes in both modes of operation employ steps-sizes μ_k that are chosen to satisfy the required stability conditions (191) and (194), which in this case are met by:*

$$
\mu_k \;<\; \frac{2}{\lambda_{\max}(R_u)}, \qquad k=1,2,\ldots,N \tag{199}
$$

It then holds that the magnitude of the error vector, $\|\widetilde{w}_i\|$, in the diffusion case decays to zero more rapidly than in the non-cooperative case. In other words, diffusion enhances convergence rate.

Proof. Since $R_{u,\ell}=R_u$ for all ℓ and C is doubly stochastic, we get $R_k=R_u$ and $I_{NM}-\mathcal{M}\mathcal{R}=I_{NM}-\mathcal{M}\mathcal{R}_u$. Then,

$$
\begin{aligned}
\rho\left(\mathcal{A}_2^T\left(I_{NM}-\mathcal{M}\mathcal{R}\right)\mathcal{A}_1^T\right) &\overset{(605)}{\leq} \rho(I_{NM}-\mathcal{M}\mathcal{R}) \\
&= \rho(I_{NM}-\mathcal{M}\mathcal{R}_u)
\end{aligned}
$$

□

The next statement considers the case of ATC and CTA strategies (137) and (145) without information exchange, which correspond to the case $C=I_N$. The result establishes that these strategies always enhance the convergence rate over the non-cooperative case, without the need to assume uniform step-sizes or uniform covariance data.

Theorem* 5.5.** *(Convergence Rate is Enhanced when $\boldsymbol{C=I}$***) Consider the problem of optimizing the global cost (92) with the individual cost functions given by (93). Pick left stochastic matrices A_1 and A_2 satisfying (166) and set $C=I_N$. This situation covers the ATC and CTA strategies (137) and (145), which*

do not involve information exchange. Consider two modes of operation. In one mode, each node in the network runs the (distributed) steepest-descent diffusion algorithm (168)–(170). In the second mode, each node operates individually and runs the non-cooperative steepest-descent algorithm (171). In both cases, the positive step-sizes are chosen to satisfy the required stability conditions (193) and (194), which in this case are met by

$$\mu_k \ < \ \frac{2}{\lambda_{\max}(R_{u,k})}, \qquad k = 1, 2, \ldots, N \tag{200}$$

It then holds that the magnitude of the error vector, $\|\widetilde{w}_i\|$, in the diffusion case decays to zero more rapidly than in the non-cooperative case. In other words, diffusion cooperation enhances convergence rate.

Proof. When $C = I_N$, we get $R_k = R_{u,k}$ and, therefore, $\mathcal{R} = \mathcal{R}_u$ and $I_{NM} - \mathcal{M}\mathcal{R} = I_{NM} - \mathcal{M}\mathcal{R}_u$. Then,

$$\begin{aligned} \rho\left(\mathcal{A}_2^T \left(I_{NM} - \mathcal{M}\mathcal{R}\right) \mathcal{A}_1^T\right) &\overset{(605)}{\leq} \rho(I_{NM} - \mathcal{M}\mathcal{R}) \\ &= \rho(I_{NM} - \mathcal{M}\mathcal{R}_u) \end{aligned}$$

□

The results of the previous theorems highlight the following important facts about the role of the combination matrices $\{A_1, A_2, C\}$ in the convergence behavior of the diffusion strategy (163)–(165):

(a) The matrix C influences the stability of the network through its influence on the bound in (191). This is because the matrices $\{R_k\}$ depend on the entries of C. The matrices $\{A_1, A_2\}$ do not influence network stability.

(b) The matrices $\{A_1, A_2, C\}$ influence the rate of convergence of the network since they influence the spectral radius of the matrix $\mathcal{A}_2^T \left(I_{NM} - \mathcal{M}\mathcal{R}\right) \mathcal{A}_1^T$, which controls the dynamics of the weight error vector in (189).

6 Performance of Adaptive Diffusion Strategies

We now move on to examine the behavior of the *adaptive* diffusion implementations (153)–(154), and the influence of both gradient noise and measurement noise on convergence and steady-state performance. Due to the random nature of the perturbations, it becomes necessary to evaluate the behavior of the algorithms on average, using mean-square convergence analysis. For this reason, we shall study the convergence of the weight estimates both in the mean and mean-square sense. To do so, we will again consider a general diffusion structure that includes the ATC and CTA strategies (153)–(154) as special cases. We shall further resort to the boldface notation to refer to the measurements and weight estimates in order to highlight the fact that they are now being treated as random variables. In this way, the update equations becomes stochastic updates. Thus, consider the following general adaptive diffusion strategy for $i \geq 0$:

$$\boldsymbol{\phi}_{k,i-1} \ = \ \sum_{\ell \in \mathcal{N}_k} a_{1,\ell k} \, \boldsymbol{w}_{\ell,i-1} \tag{201}$$

$$\boldsymbol{\psi}_{k,i} \ = \ \boldsymbol{\phi}_{k,i-1} \ + \ \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \, \boldsymbol{u}_{\ell,i}^* \left[\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i} \, \boldsymbol{\phi}_{k,i-1}\right] \tag{202}$$

$$\boldsymbol{w}_{k,i} \ = \ \sum_{\ell \in \mathcal{N}_k} a_{2,\ell k} \, \boldsymbol{\psi}_{\ell,i} \tag{203}$$

As before, the scalars $\{a_{1,\ell k}, c_{\ell k}, a_{2,\ell k}\}$ are non-negative real coefficients corresponding to the (ℓ, k) entries of $N \times N$ combination matrices $\{A_1, C, A_2\}$, respectively. These matrices are assumed to satisfy the same conditions (166) or (167). Again, different choices for $\{A_1, C, A_2\}$ correspond to different cooperation modes. For example, the choice $A_1 = I_N$ and $A_2 = A$ corresponds to the adaptive ATC implementation (153), while the choice $A_1 = A$ and $A_2 = I_N$ corresponds to the adaptive CTA implementation (154). Likewise, the

choice $C = I_N$ corresponds to the case in which the nodes only share weight estimates and the distributed diffusion recursions (201)–(203) become

$$\boldsymbol{\phi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\, \boldsymbol{w}_{\ell,i-1} \tag{204}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^* \left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\, \boldsymbol{\phi}_{k,i-1}\right] \tag{205}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\, \boldsymbol{\psi}_{\ell,i} \tag{206}$$

Furthermore, the choice $A_1 = A_2 = C = I_N$ corresponds to the non-cooperative mode of operation, where each node runs the classical (stand-alone) least-mean-squares (LMS) filter independently of the other nodes: [4–7]:

$$\boldsymbol{w}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i} \left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\, \boldsymbol{w}_{k,i-1}\right], \quad i \geq 0 \tag{207}$$

6.1 Data Model

When we studied the performance of the steepest-descent diffusion strategy (163)–(165) we exploited result (175), which indicated how the moments $\{r_{du,k}, R_{u,k}\}$ that appeared in the recursions related to the optimal solution w^o. Likewise, in order to be able to analyze the performance of the *adaptive* diffusion strategy (201)–(203), we need to know how the data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ across the network relate to w^o. Motivated by the several examples presented earlier in Sec. 2, we shall assume that the data satisfy a linear model of the form:

$$\boxed{\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o \;+\; \boldsymbol{v}_k(i)} \tag{208}$$

where $\boldsymbol{v}_k(i)$ is measurement noise with variance $\sigma_{v,k}^2$:

$$\sigma_{v,k}^2 \;\triangleq\; \mathbb{E}\,|\boldsymbol{v}_k(i)|^2 \tag{209}$$

and where the stochastic processes $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ are assumed to be jointly wide-sense stationary with moments:

$$\sigma_{d,k}^2 \;\triangleq\; \mathbb{E}\,|\boldsymbol{d}_k(i)|^2 \qquad \text{(scalar)} \tag{210}$$

$$R_{u,k} \;\triangleq\; \mathbb{E}\,\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i} > 0 \qquad (M\times M) \tag{211}$$

$$r_{du,k} \;\triangleq\; \mathbb{E}\,\boldsymbol{d}_k(i)\boldsymbol{u}_{k,i}^* \qquad (M\times 1) \tag{212}$$

All variables are assumed to be zero-mean. Furthermore, the noise process $\{\boldsymbol{v}_k(i)\}$ is assumed to be temporally white and spatially independent, as described earlier by (6), namely,

$$\begin{cases} \mathbb{E}\,\boldsymbol{v}_k(i)\boldsymbol{v}_k^*(j) = 0, & \text{for all } i \neq j \text{ (temporal whiteness)} \\ \mathbb{E}\,\boldsymbol{v}_k(i)\boldsymbol{v}_m^*(j) = 0, & \text{for all } i,j \text{ whenever } k \neq m \text{ (spatial whiteness)} \end{cases} \tag{213}$$

The noise process $\boldsymbol{v}_k(i)$ is further assumed to be independent of the regression data $\boldsymbol{u}_{m,j}$ for all k, m and i, j so that:

$$\mathbb{E}\,\boldsymbol{v}_k(i)\boldsymbol{u}_{m,j}^* = 0, \qquad \text{for all } k,m,i,j \tag{214}$$

We shall also assume that the regression data are temporally white and spatially independent so that:

$$\mathbb{E}\,\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{\ell,j} = R_{u,k}\delta_{k\ell}\delta_{ij} \tag{215}$$

Although we are going to derive performance measures for the network under this independence assumption on the regression data, it turns out that the resulting expressions continue to match well with simulation results for sufficiently small step-sizes, even when the independence assumption does not hold (in a manner similar to the behavior of stand-alone adaptive filters) [4, 5].

6.2 Performance Measures

Our objective is to analyze whether, and how fast, the weight estimates $\{\boldsymbol{w}_{k,i}\}$ from the adaptive diffusion implementation (201)–(203) converge towards w^o. To do so, we again introduce the $M \times 1$ weight error vectors:

$$\widetilde{\boldsymbol{\phi}}_{k,i} \triangleq w^o - \boldsymbol{\phi}_{k,i} \tag{216}$$

$$\widetilde{\boldsymbol{\psi}}_{k,i} \triangleq w^o - \boldsymbol{\psi}_{k,i} \tag{217}$$

$$\widetilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i} \tag{218}$$

Each of these error vectors measures the residual relative to the desired w^o in (208). We further introduce two scalar error measures:

$$\boldsymbol{e}_k(i) \triangleq \boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1} \quad \text{(output error)} \tag{219}$$

$$\boldsymbol{e}_{a,k}(i) \triangleq \boldsymbol{u}_{k,i}\widetilde{\boldsymbol{w}}_{k,i-1} \quad (\textit{a-priori}\text{ error}) \tag{220}$$

The first error measures how well the term $\boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}$ approximates the measured data, $\boldsymbol{d}_k(i)$; in view of (208), this error can be interpreted as an estimator for the noise term $\boldsymbol{v}_k(i)$. If node k is able to estimate w^o well, then $\boldsymbol{e}_k(i)$ would get close to $\boldsymbol{v}_k(i)$. Therefore, under ideal conditions, we would expect the variance of $\boldsymbol{e}_k(i)$ to tend towards the variance of $\boldsymbol{v}_k(i)$. However, as remarked earlier in (31), there is generally an offset term for adaptive implementations that is captured by the variance of the *a-priori* error, $\boldsymbol{e}_{a,k}(i)$. This second error measures how well $\boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}$ approximates the uncorrupted term $\boldsymbol{u}_{k,i}w^o$. Using the data model (208), we can relate $\{\boldsymbol{e}_k(i), \boldsymbol{e}_{a,k}(i)\}$ as

$$\boldsymbol{e}_k(i) = \boldsymbol{e}_{a,k} + \boldsymbol{v}_k(i) \tag{221}$$

Since the noise component, $\boldsymbol{v}_k(i)$, is assumed to be zero-mean and independent of all other random variables, we recover (31):

$$\boxed{\mathbb{E}\,|\boldsymbol{e}_k(i)|^2 = \mathbb{E}\,|\boldsymbol{e}_{a,k}(i)|^2 + \sigma_{v,k}^2} \tag{222}$$

This relation confirms that the variance of the output error, $\boldsymbol{e}_k(i)$, is always at least as large as $\sigma_{v,k}^2$ and away from it by an amount that is equal to the variance of the *a-priori* error, $\boldsymbol{e}_{a,k}(i)$. Accordingly, in order to quantify the performance of any particular node in the network, we define the mean-square-error (MSE) and excess-mean-square-error (EMSE) for node k as the following steady-state measures:

$$\text{MSE}_k \triangleq \lim_{i\to\infty} \mathbb{E}\,|\boldsymbol{e}_k(i)|^2 \tag{223}$$

$$\text{EMSE}_k \triangleq \lim_{i\to\infty} \mathbb{E}\,|\boldsymbol{e}_{a,k}(i)|^2 \tag{224}$$

Then, it holds that

$$\boxed{\text{MSE}_k = \text{EMSE}_k + \sigma_{v,k}^2} \tag{225}$$

Therefore, the EMSE term quantifies the size of the offset in the MSE performance of each node. We also define the mean-square-deviation (MSD) of each node as the steady-state measure:

$$\text{MSD}_k \triangleq \lim_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \tag{226}$$

which measures how far $\boldsymbol{w}_{k,i}$ is from w^o in the mean-square-error sense.

We indicated earlier in (36)–(37) how the MSD and EMSE of stand-alone LMS filters in the non-cooperative case depend on $\{\mu_k, \sigma_v^2, R_{u,k}\}$. In this section, we examine how cooperation among the nodes

influences their performance. Since cooperation couples the operation of the nodes, with data originating from one node influencing the behavior of its neighbors and their neighbors, the study of the network performance requires more effort than in the non-cooperative case. Nevertheless, when all is said and done, we will arrive at expressions that approximate well the network performance and reveal some interesting conclusions.

6.3 Error Recursions

Using the data model (208) and subtracting w^o from both sides of the relations in (201)–(203) we get

$$\widetilde{\boldsymbol{\phi}}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\ \widetilde{\boldsymbol{w}}_{\ell,i-1} \tag{227}$$

$$\widetilde{\boldsymbol{\psi}}_{k,i} = \left(I_M - \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\ \boldsymbol{u}_{\ell,i}^*\boldsymbol{u}_{\ell,i}\right)\ \widetilde{\boldsymbol{\phi}}_{k,i-1} \ -\ \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\ \boldsymbol{u}_{\ell,i}^*\boldsymbol{v}_\ell(i) \tag{228}$$

$$\widetilde{\boldsymbol{w}}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\ \widetilde{\boldsymbol{\psi}}_{\ell,i} \tag{229}$$

Comparing the second recursion with the corresponding recursion in the steepest-descent case (176)–(178), we see that two new effects arise: the effect of gradient noise, which replaces the covariance matrices $R_{u,\ell}$ by the instantaneous approximation $\boldsymbol{u}_{\ell,i}^*\boldsymbol{u}_{\ell,i}$, and the effect of measurement noise, $\boldsymbol{v}_\ell(i)$.

We again describe the above relations more compactly by collecting the information from across the network in block vectors and matrices. We collect the error vectors from across all nodes into the following $N\times 1$ *block* vectors, whose individual entries are of size $M\times 1$ each:

$$\widetilde{\boldsymbol{\psi}}_i \triangleq \begin{bmatrix} \widetilde{\boldsymbol{\psi}}_{1,i} \\ \widetilde{\boldsymbol{\psi}}_{2,i} \\ \vdots \\ \widetilde{\boldsymbol{\psi}}_{N,i} \end{bmatrix}, \qquad \widetilde{\boldsymbol{\phi}}_i \triangleq \begin{bmatrix} \widetilde{\boldsymbol{\phi}}_{1,i} \\ \widetilde{\boldsymbol{\phi}}_{2,i} \\ \vdots \\ \widetilde{\boldsymbol{\phi}}_{N,i} \end{bmatrix}, \qquad \widetilde{\boldsymbol{w}}_i \triangleq \begin{bmatrix} \widetilde{\boldsymbol{w}}_{1,i} \\ \widetilde{\boldsymbol{w}}_{2,i} \\ \vdots \\ \widetilde{\boldsymbol{w}}_{N,i} \end{bmatrix} \tag{230}$$

The block quantities $\{\widetilde{\boldsymbol{\psi}}_i, \widetilde{\boldsymbol{\phi}}_i, \widetilde{\boldsymbol{w}}_i\}$ represent the state of the errors across the network at time i. Likewise, we introduce the following $N\times N$ *block* diagonal matrices, whose individual entries are of size $M\times M$ each:

$$\mathcal{M} \triangleq \mathrm{diag}\{\ \mu_1 I_M,\ \mu_2 I_M,\ \ldots,\ \mu_N I_M\ \} \tag{231}$$

$$\boldsymbol{\mathcal{R}}_i \triangleq \mathrm{diag}\left\{\ \sum_{\ell\in\mathcal{N}_1} c_{\ell 1}\ \boldsymbol{u}_{\ell,i}^*\boldsymbol{u}_{\ell,i},\ \sum_{\ell\in\mathcal{N}_2} c_{\ell 2}\ \boldsymbol{u}_{\ell,i}^*\boldsymbol{u}_{\ell,i},\ \ldots,\ \sum_{\ell\in\mathcal{N}_N} c_{\ell N}\ \boldsymbol{u}_{\ell,i}^*\boldsymbol{u}_{\ell,i}\ \right\} \tag{232}$$

Each block diagonal entry of $\boldsymbol{\mathcal{R}}_i$, say, the k-th entry, contains a combination of rank-one regression terms collected from the neighborhood of node k. In this way, the matrix $\boldsymbol{\mathcal{R}}_i$ is now stochastic *and* dependent on time, in contrast to the matrix $\mathcal{R}$ in the steepest-descent case in (181), which was a constant matrix. Nevertheless, it holds that

$$\boxed{\mathbb{E}\,\boldsymbol{\mathcal{R}}_i = \mathcal{R}} \tag{233}$$

so that, on average, $\boldsymbol{\mathcal{R}}_i$ agrees with $\mathcal{R}$. We can simplify the notation by denoting the neighborhood combinations as follows:

$$\boldsymbol{R}_{k,i} \triangleq \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\ \boldsymbol{u}_{\ell,i}^*\boldsymbol{u}_{\ell,i} \tag{234}$$

so that $\boldsymbol{\mathcal{R}}_i$ becomes

$$\boldsymbol{\mathcal{R}}_i \triangleq \mathrm{diag}\{\ \boldsymbol{R}_{1,i},\ \boldsymbol{R}_{2,i},\ \ldots,\boldsymbol{R}_{N,i}\ \} \qquad (\text{when } C\neq I) \tag{235}$$

Again, compared with the matrix R_k defined in (182), we find that $\boldsymbol{R}_{k,i}$ is now both stochastic and time-dependent. Nevertheless, it again holds that

$$\boxed{\mathbb{E}\,\boldsymbol{R}_{k,i} \;=\; R_k} \tag{236}$$

In the special case when $C = I$, the matrix $\boldsymbol{\mathcal{R}}_i$ reduces to

$$\boldsymbol{\mathcal{R}}_{u,i} \;\triangleq\; \mathrm{diag}\{\boldsymbol{u}_{1,i}^*\boldsymbol{u}_{1,i},\; \boldsymbol{u}_{2,i}^*\boldsymbol{u}_{2,i},\; \ldots,\; \boldsymbol{u}_{N,i}^*\boldsymbol{u}_{N,i}\} \quad (\text{when } C = I) \tag{237}$$

with

$$\boxed{\mathbb{E}\,\boldsymbol{\mathcal{R}}_{u,i} \;=\; \mathcal{R}_u} \tag{238}$$

where $\mathcal{R}_u$ was defined earlier in (184).

We further introduce the following $N \times 1$ block column vector, whose entries are of size $M \times 1$ each:

$$\boldsymbol{s}_i \;\triangleq\; \mathrm{col}\{\, \boldsymbol{u}_{1,i}^*\boldsymbol{v}_1(i),\; \boldsymbol{u}_{2,i}^*\boldsymbol{v}_2(i), \ldots,\; \boldsymbol{u}_{N,i}^*\boldsymbol{v}_N(i)\,\} \tag{239}$$

Obviously, given that the regression data and measurement noise are zero-mean and independent of each other, we have

$$\boxed{\mathbb{E}\,\boldsymbol{s}_i \;=\; 0} \tag{240}$$

and the covariance matrix of $\boldsymbol{s}_i$ is $N \times N$ block diagonal with blocks of size $M \times M$:

$$\boxed{\mathcal{S} \;\triangleq\; \mathbb{E}\,\boldsymbol{s}_i\boldsymbol{s}_i^* \;=\; \mathrm{diag}\{\sigma_{v,1}^2 R_{u,1},\; \sigma_{v,2}^2 R_{u,2},\; \ldots, \sigma_{v,N}^2 R_{u,N}\}} \tag{241}$$

Returning to (227)–(229), we conclude that the following relations hold for the block quantities:

$$\widetilde{\boldsymbol{\phi}}_{i-1} \;=\; \mathcal{A}_1^T\widetilde{\boldsymbol{w}}_{i-1} \tag{242}$$

$$\widetilde{\boldsymbol{\psi}}_i \;=\; (I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\;\widetilde{\boldsymbol{\phi}}_{i-1} \;-\; \mathcal{M}\mathcal{C}^T\boldsymbol{s}_i \tag{243}$$

$$\widetilde{\boldsymbol{w}}_i \;=\; \mathcal{A}_2^T\widetilde{\boldsymbol{\psi}}_i \tag{244}$$

where

$$\boxed{\mathcal{C} \;\triangleq\; C \otimes I_M} \tag{245}$$

so that the network weight error vector, $\widetilde{\boldsymbol{w}}_i$, ends up evolving according to the following *stochastic* recursion:

$$\boxed{\widetilde{\boldsymbol{w}}_i = \mathcal{A}_2^T\,(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\,\mathcal{A}_1^T\widetilde{\boldsymbol{w}}_{i-1} \;-\; \mathcal{A}_2^T\mathcal{M}\mathcal{C}^T\boldsymbol{s}_i,\;\; i \geq 0} \quad (\text{diffusion strategy}) \tag{246}$$

For comparison purposes, if each node operates individually and uses the non-cooperative LMS recursion (207), then the weight error vector across all N nodes would evolve according to the following stochastic recursion:

$$\boxed{\widetilde{\boldsymbol{w}}_i = \;(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_{u,i})\,\widetilde{\boldsymbol{w}}_{i-1} \;-\; \mathcal{M}\boldsymbol{s}_i,\;\; i \geq 0} \quad (\text{non-cooperative strategy}) \tag{247}$$

where the matrices $\mathcal{A}_1$ and $\mathcal{A}_2$ do not appear, and $\boldsymbol{\mathcal{R}}_i$ is replaced by $\boldsymbol{\mathcal{R}}_{u,i}$ from (237).

6.4 Convergence in the Mean

Taking expectations of both sides of (246) we find that:

$$\boxed{\mathbb{E}\,\widetilde{\boldsymbol{w}}_i = \mathcal{A}_2^T\,(I_{NM} - \mathcal{M}\mathcal{R})\,\mathcal{A}_1^T \cdot \mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1},\;\; i \geq 0} \qquad \text{(diffusion strategy)} \tag{248}$$

where we used the fact that $\widetilde{\boldsymbol{w}}_{i-1}$ and $\boldsymbol{\mathcal{R}}_i$ are independent of each other in view of our earlier assumptions on the regression data and noise in Sec. 6.1. Comparing with the error recursion (189) in the steepest-descent case, we find that both recursions are identical with $\widetilde{w}_i$ replaced by $\mathbb{E}\,\widetilde{\boldsymbol{w}}_i$. Therefore, the convergence statements from the steepest-descent case can be extended to the adaptive case to provide conditions on the step-size to ensure stability in the mean, i.e., to ensure

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_i \longrightarrow 0 \quad \text{as} \quad i \longrightarrow \infty \tag{249}$$

When (249) is guaranteed, we would say that the adaptive diffusion solution is asymptotically unbiased. The following statements restate the results of Theorems 5.1–5.5 in the context of mean error analysis.

***Theorem* 6.1. (Convergence in the Mean)** *Consider the problem of optimizing the global cost (92) with the individual cost functions given by (93). Pick a right stochastic matrix C and left stochastic matrices A_1 and A_2 satisfying (166) or (167). Assume each node in the network measures data that satisfy the conditions described in Sec. 6.1, and runs the adaptive diffusion algorithm (201)–(203). Then, all estimators $\{\boldsymbol{w}_{k,i}\}$ across the network converge in the mean to the optimal solution w^o if the positive step-size parameters $\{\mu_k\}$ satisfy*

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_k)}} \tag{250}$$

where the neighborhood covariance matrix R_k is defined by (182). In other words, $\mathbb{E}\,\boldsymbol{w}_{k,i} \to w^o$ for all nodes k as $i \to \infty$.

□

Observe again that the mean stability condition (250) does not depend on the specific combination matrices A_1 and A_2 that are being used. Only the combination matrix C influences the condition on the step-size through the neighborhood covariance matrices $\{R_k\}$. Observe further that the statement of the lemma does not require the network to be connected. Moreover, when $C = I_N$, in which case the nodes only share weight estimators and do not share neighborhood data $\{\boldsymbol{d}_\ell(i), \boldsymbol{u}_{\ell,i}\}$ as in (204)–(206), condition (250) becomes

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_{u,k})}} \qquad \text{(adaptive cooperation with } C = I_N\text{)} \tag{251}$$

Results (250) and (251) are reminiscent of a classical result for the stand-alone LMS algorithm, as in the non-cooperative case (207), where it is known that the estimator by each individual node in this case would converge in the mean to w^o if, and only if, its step-size satisfies

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_{u,k})}} \qquad \text{(non-cooperative adaptation)} \tag{252}$$

The following statement provides a bi-directional result that ensures the mean convergence of the adaptive diffusion strategy for *any* choice of left-stochastic combination matrices A_1 and A_2.

Theorem* 6.2.** *(Mean Convergence for Arbitrary Combination Matrices***) Consider the problem of optimizing the global cost (92) with the individual cost functions given by (93). Pick a right stochastic matrix C satisfying (166). Assume each node in the network measures data that satisfy the conditions described in Sec. 6.1. Then, the estimators $\{\boldsymbol{w}_{k,i}\}$ generated by the adaptive diffusion strategy (201)–(203), converge in the mean to w^o, for all choices of left stochastic matrices A_1 and A_2 satisfying (166) if, and only if,*

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_k)}} \tag{253}$$

□

As was the case with steepest-descent diffusion strategies, the adaptive diffusion strategy (201)–(203) also enhances the convergence rate of the mean of the error vector towards zero relative to the non-cooperative strategy (207). The next results restate Theorems 5.3–5.5; they assume C is a doubly stochastic matrix.

Theorem* 6.3.** *(Mean Convergence Rate is Enhanced: Uniform Step-Sizes***) Consider the problem of optimizing the global cost (92) with the individual cost functions given by (93). Pick a doubly stochastic matrix C satisfying (196) and left stochastic matrices A_1 and A_2 satisfying (166). Assume each node in the network measures data that satisfy the conditions described in Sec. 6.1. Consider two modes of operation. In one mode, each node in the network runs the adaptive diffusion algorithm (201)–(203). In the second mode, each node operates individually and runs the non-cooperative LMS algorithm (207). In both cases, the positive step-sizes used by all nodes are assumed to be the same, say, $\mu_k = \mu$ for all k, and the value of μ is chosen to satisfy the required mean stability conditions (250) and (252), which are met by selecting*

$$\mu \;<\; \min_{1\leq k\leq N} \left\{ \frac{2}{\lambda_{\max}(R_{u,k})} \right\} \tag{254}$$

It then holds that the magnitude of the mean error vector, $\|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i\|$ in the diffusion case decays to zero more rapidly than in the non-cooperative case. In other words, diffusion enhances convergence rate.

□

Theorem* 6.4.** *(Mean Convergence Rate is Enhanced: Uniform Covariance Data***) Consider the same setting of Theorem 6.3. Assume the covariance data are uniform across all nodes, say, $R_{u,k} = R_u$ is independent of k. Assume further that the nodes in both modes of operation employ steps-sizes μ_k that are chosen to satisfy the required stability conditions (250) and (252), which in this case are met by:*

$$\mu_k \;<\; \frac{2}{\lambda_{\max}(R_u)}, \qquad k = 1, 2, \ldots, N \tag{255}$$

It then holds that the magnitude of the mean error vector, $\|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i\|$, in the diffusion case also decays to zero more rapidly than in the non-cooperative case. In other words, diffusion enhances convergence rate.

□

The next statement considers the case of ATC and CTA strategies (204)–(206) without information exchange, which correspond to the choice $C = I_N$. The result establishes that these strategies always enhance the convergence rate over the non-cooperative case, without the need to assume uniform step-sizes or uniform covariance data.

Theorem* 6.5.** *(Mean Convergence Rate is Enhanced when $\boldsymbol{C = I}$***) Consider the problem of optimizing the global cost (92) with the individual cost functions given by (93). Pick left stochastic matrices A_1 and A_2 satisfying (166) and set $C = I_N$. This situation covers the ATC and CTA strategies (204)–(206) that do not involve information exchange. Assume each node in the network measures data that satisfy the conditions described in Sec. 6.1. Consider two modes of operation. In one mode, each node in the network*

runs the adaptive diffusion algorithm (204)–(206). In the second mode, each node operates individually and runs the non-cooperative LMS algorithm (207). In both cases, the positive step-sizes are chosen to satisfy the required stability conditions (251) and (252), which in this case are met by

$$\mu_k \;<\; \frac{2}{\lambda_{\max}(R_{u,k})}, \qquad k = 1, 2, \ldots, N \tag{256}$$

It then holds that the magnitude of the mean error vector, $\|\mathbb{E}\,\widetilde{\boldsymbol{w}}_i\|$, in the diffusion case decays to zero more rapidly than in the non-cooperative case. In other words, diffusion cooperation enhances convergence rate.

□

The results of the previous theorems again highlight the following important facts about the role of the combination matrices $\{A_1, A_2, C\}$ in the convergence behavior of the adaptive diffusion strategy (201)–(203):

(a) The matrix C influences the mean stability of the network through its influence on the bound in (250). This is because the matrices $\{R_k\}$ depend on the entries of C. The matrices $\{A_1, A_2\}$ do not influence network mean stability.

(b) The matrices $\{A_1, A_2, C\}$ influence the rate of convergence of the mean weight-error vector over the network since they influence the spectral radius of the matrix $\mathcal{A}_2^T\,(I_{NM} - \mathcal{M}\mathcal{R})\,\mathcal{A}_1^T$, which controls the dynamics of the weight error vector in (248).

6.5 Mean-Square Stability

It is not sufficient to ensure the stability of the weight-error vector in the mean sense. The error vectors, $\tilde{\boldsymbol{w}}_{k,i}$, may be converging on average to zero but they may have large fluctuations around the zero value. We therefore need to examine how small the error vectors get. To do so, we perform a mean-square-error analysis. The purpose of the analysis is to evaluate how the variances $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ evolve with time and what their steady-state values are, for each node k.

In this section, we are particularly interested in evaluating the evolution of two mean-square-errors, namely,

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \qquad \text{and} \qquad \mathbb{E}\,|\boldsymbol{e}_{a,k}(i)|^2 \tag{257}$$

The steady-state values of these quantities determine the MSD and EMSE performance levels at node k and, therefore, convey critical information about the performance of the network. Under the independence assumption on the regression data from Sec. 6.1, it can be verified that the EMSE variance can be written as:

$$\begin{aligned}
\mathbb{E}\,|\boldsymbol{e}_{a,k}(i)|^2 &\triangleq \mathbb{E}\,|\boldsymbol{u}_{k,i}\widetilde{\boldsymbol{w}}_{k,i-1}|^2 \\
&= \mathbb{E}\,\widetilde{\boldsymbol{w}}_{k,i-1}^*\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}\widetilde{\boldsymbol{w}}_{k,i-1} \\
&= \mathbb{E}\,\left[\mathbb{E}\,(\widetilde{\boldsymbol{w}}_{k,i-1}^*\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}\widetilde{\boldsymbol{w}}_{k,i-1}|\widetilde{\boldsymbol{w}}_{k,i-1})\right] \\
&= \mathbb{E}\,\widetilde{\boldsymbol{w}}_{k,i-1}^*\left[\mathbb{E}\,\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i}\right]\widetilde{\boldsymbol{w}}_{k,i-1} \\
&= \mathbb{E}\,\widetilde{\boldsymbol{w}}_{k,i-1}^* R_{u,k}\widetilde{\boldsymbol{w}}_{k,i-1} \\
&= \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i-1}\|_{R_{u,k}}^2
\end{aligned} \tag{258}$$

in terms of a weighted square measure with weighting matrix $R_{u,k}$. Here we are using the notation $\|x\|_\Sigma^2$ to denote the weighted square quantity $x^*\Sigma x$, for any column vector x and matrix Σ. Thus, we can evaluate mean-square-errors of the form (257) by evaluating the means of weighted square quantities of the following form:

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|_{\Sigma_k}^2 \tag{259}$$

for an arbitrary Hermitian nonnegative-definite weighting matrix Σ_k that we are free to choose. By setting Σ_k to different values (say, $\Sigma_k = I$ or $\Sigma_k = R_{u,k}$), we can extract various types of information about

the nodes and the network, as the discussion will reveal. The approach we follow is based on the energy conservation framework of [4, 5, 57].

So, let Σ denote an arbitrary $N \times N$ block Hermitian nonnegative-definite matrix that we are free to choose, with $M \times M$ block entries $\{\Sigma_{\ell k}\}$. Let σ denote the $(NM)^2 \times 1$ vector that is obtained by stacking the columns of Σ on top of each other, written as

$$\sigma \triangleq \mathrm{vec}(\Sigma) \tag{260}$$

In the sequel, it will become more convenient to work with the vector representation σ than with the matrix Σ itself.

We start from the weight-error vector recursion (246) and re-write it more compactly as:

$$\boxed{\widetilde{\boldsymbol{w}}_i \;=\; \boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1} \;-\; \mathcal{G}\boldsymbol{s}_i, \;\; i \geq 0} \tag{261}$$

where the coefficient matrices $\boldsymbol{\mathcal{B}}_i$ and $\mathcal{G}$ are short-hand representations for

$$\boxed{\boldsymbol{\mathcal{B}}_i \;\triangleq\; \mathcal{A}_2^T\,(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i)\,\mathcal{A}_1^T} \tag{262}$$

and

$$\boxed{\mathcal{G} \;\triangleq\; \mathcal{A}_2^T\mathcal{M}\mathcal{C}^T} \tag{263}$$

Note that $\boldsymbol{\mathcal{B}}_i$ is stochastic and time-variant, while $\mathcal{G}$ is constant. We denote the mean of $\boldsymbol{\mathcal{B}}_i$ by

$$\boxed{\mathcal{B} \;\triangleq\; \mathbb{E}\,\boldsymbol{\mathcal{B}}_i \;=\; \mathcal{A}_2^T\,(I_{NM} - \mathcal{M}\mathcal{R})\,\mathcal{A}_1^T} \tag{264}$$

where $\mathcal{R}$ is defined by (181). Now equating weighted square measures on both sides of (261) we get

$$\|\widetilde{\boldsymbol{w}}_i\|_\Sigma^2 \;=\; \|\boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1} \;-\; \mathcal{G}\boldsymbol{s}_i\|_\Sigma^2 \tag{265}$$

Expanding the right-hand side we find that

$$\begin{aligned}\|\widetilde{\boldsymbol{w}}_i\|_\Sigma^2 \;&=\; \widetilde{\boldsymbol{w}}_{i-1}^*\boldsymbol{\mathcal{B}}_i^*\Sigma\boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1} \;+\; \boldsymbol{s}_i^*\mathcal{G}^T\Sigma\mathcal{G}\boldsymbol{s}_i \;-\\ &\quad\; \widetilde{\boldsymbol{w}}_{i-1}^*\boldsymbol{\mathcal{B}}_i^*\Sigma\mathcal{G}\boldsymbol{s}_i \;-\; \boldsymbol{s}_i^*\mathcal{G}^T\Sigma\boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1}\end{aligned} \tag{266}$$

Under expectation, the last two terms on the right-hand side evaluate to zero so that

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_\Sigma^2 \;=\; \mathbb{E}\,\left(\widetilde{\boldsymbol{w}}_{i-1}^*\boldsymbol{\mathcal{B}}_i^*\Sigma\boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1}\right) \;+\; \mathbb{E}\,\left(\boldsymbol{s}_i^*\mathcal{G}^T\Sigma\mathcal{G}\boldsymbol{s}_i\right) \tag{267}$$

Let us evaluate each of the expectations on the right-hand side. The last expectation is given by

$$\begin{aligned}\mathbb{E}\,\left(\boldsymbol{s}_i^*\mathcal{G}^T\Sigma\mathcal{G}\boldsymbol{s}_i\right) \;&=\; \mathrm{Tr}\left(\mathcal{G}^T\Sigma\mathcal{G}\;\mathbb{E}\,\boldsymbol{s}_i\boldsymbol{s}_i^*\right)\\ &\overset{(241)}{=}\; \mathrm{Tr}\left(\mathcal{G}^T\Sigma\mathcal{G}\mathcal{S}\right)\\ &=\; \mathrm{Tr}\left(\Sigma\mathcal{G}\mathcal{S}\mathcal{G}^T\right)\end{aligned} \tag{268}$$

where $\mathcal{S}$ is defined by (241) and where we used the fact that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ for any two matrices A and B of compatible dimensions. With regards to the first expectation on the right-hand side of (267), we have

$$\begin{aligned}\mathbb{E}\,\left(\widetilde{\boldsymbol{w}}_{i-1}^*\boldsymbol{\mathcal{B}}_i^*\Sigma\boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1}\right) \;&=\; \mathbb{E}\,\left[\mathbb{E}\,\left(\widetilde{\boldsymbol{w}}_{i-1}^*\boldsymbol{\mathcal{B}}_i^*\Sigma\boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{w}}_{i-1}|\widetilde{\boldsymbol{w}}_{i-1}\right)\right]\\ &=\; \mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}^*\left[\mathbb{E}\,\left(\boldsymbol{\mathcal{B}}_i^*\Sigma\boldsymbol{\mathcal{B}}_i\right)\right]\widetilde{\boldsymbol{w}}_{i-1}\\ &\triangleq\; \mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1}^*\Sigma'\widetilde{\boldsymbol{w}}_{i-1}\\ &=\; \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2\end{aligned} \tag{269}$$

where we introduced the nonnegative-definite weighting matrix

$$\begin{aligned}\Sigma' &\triangleq \mathbb{E}\,\boldsymbol{\mathcal{B}}_i^*\Sigma\boldsymbol{\mathcal{B}}_i \\ &\overset{(262)}{=} \mathbb{E}\,\mathcal{A}_1\left(I_{NM}-\boldsymbol{\mathcal{R}}_i\mathcal{M}\right)\mathcal{A}_2\Sigma\mathcal{A}_2^T\left(I_{NM}-\mathcal{M}\boldsymbol{\mathcal{R}}_i\right)\mathcal{A}_1^T \\ &= \mathcal{A}_1\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{A}_1^T-\mathcal{A}_1\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\mathcal{R}\mathcal{A}_1^T-\mathcal{A}_1\mathcal{R}\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{A}_1^T\;+\;O(\mathcal{M}^2)\end{aligned} \tag{270}$$

where $\mathcal{R}$ is defined by (181) and the term $O(\mathcal{M}^2)$ denotes the following factor, which depends on the square of the step-sizes, $\{\mu_k^2\}$:

$$O(\mathcal{M}^2)\;=\;\mathbb{E}\left(\mathcal{A}_1\boldsymbol{\mathcal{R}}_i\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\boldsymbol{\mathcal{R}}_i\mathcal{A}_1^T\right) \tag{271}$$

The evaluation of the above expectation depends on higher-order moments of the regression data. While we can continue with the analysis by taking this factor into account, as was done in [4, 5, 18, 57], it is sufficient for the exposition in this article to focus on the case of sufficiently small step-sizes where terms involving higher powers of the step-sizes can be ignored. Therefore, we continue our discussion by letting

$$\boxed{\Sigma'\;\triangleq\;\mathcal{A}_1\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{A}_1^T-\mathcal{A}_1\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{M}\mathcal{R}\mathcal{A}_1^T-\mathcal{A}_1\mathcal{R}\mathcal{M}\mathcal{A}_2\Sigma\mathcal{A}_2^T\mathcal{A}_1^T} \tag{272}$$

The weighting matrix Σ' is fully defined in terms of the step-size matrix, $\mathcal{M}$, the network topology through the matrices $\{\mathcal{A}_1,\mathcal{A}_2,\mathcal{C}\}$, and the regression statistical profile through $\mathcal{R}$. Expression (272) tells us how to construct Σ' from Σ. The expression can be transformed into a more compact and revealing form if we instead relate the vector forms $\sigma'=\mathrm{vec}(\Sigma')$ and $\sigma=\mathrm{vec}(\Sigma)$. Using the following equalities for arbitrary matrices $\{U,W,\Sigma\}$ of compatible dimensions [5]:

$$\mathrm{vec}(U\Sigma W)\;=\;(W^T\otimes U)\sigma \tag{273}$$

$$\mathrm{Tr}(\Sigma W)\;=\;\left[\mathrm{vec}(W^T)\right]^T\sigma \tag{274}$$

and applying the vec operation to both sides of (272) we get

$$\sigma'\;=\;(\mathcal{A}_1\mathcal{A}_2\otimes\mathcal{A}_1\mathcal{A}_2)\,\sigma\;-\;\left(\mathcal{A}_1\mathcal{R}^T\mathcal{M}\mathcal{A}_2\otimes\mathcal{A}_1\mathcal{A}_2\right)\sigma\;-\;(\mathcal{A}_1\mathcal{A}_2\otimes\mathcal{A}_1\mathcal{R}\mathcal{M}\mathcal{A}_2)\,\sigma$$

That is,

$$\boxed{\sigma'\;\triangleq\;\mathcal{F}\sigma} \tag{275}$$

where we are introducing the coefficient matrix of size $(NM)^2\times(NM)^2$:

$$\boxed{\mathcal{F}\;\triangleq\;(\mathcal{A}_1\mathcal{A}_2\otimes\mathcal{A}_1\mathcal{A}_2)\;-\;\left(\mathcal{A}_1\mathcal{R}^T\mathcal{M}\mathcal{A}_2\otimes\mathcal{A}_1\mathcal{A}_2\right)\;-\;(\mathcal{A}_1\mathcal{A}_2\otimes\mathcal{A}_1\mathcal{R}\mathcal{M}\mathcal{A}_2)} \tag{276}$$

A reasonable approximate expression for $\mathcal{F}$ for sufficiently small step-sizes is

$$\boxed{\mathcal{F}\approx\mathcal{B}^T\otimes\mathcal{B}^*} \tag{277}$$

Indeed, if we replace $\mathcal{B}$ from (264) into (277) and expand terms, we obtain the same factors that appear in (276) plus an additional term that depends on the square of the step-sizes, $\{\mu_k^2\}$, whose effect can be ignored for sufficiently small step-sizes.

In this way, using in addition property (274), we find that relation (267) becomes:

$$\boxed{\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_{\Sigma}^2\;=\;\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|_{\Sigma'}^2\;+\;\left[\mathrm{vec}\left(\mathcal{G}\mathcal{S}^T\mathcal{G}^T\right)\right]^T\sigma} \tag{278}$$

The last term is dependent on the network topology through the matrix $\mathcal{G}$, which is defined in terms of $\{\mathcal{A}_2, \mathcal{C}, \mathcal{M}\}$, and the noise and regression data statistical profile through $\mathcal{S}$. It is convenient to introduce the alternative notation $\|x\|_\sigma^2$ to refer to the weighted square quantity $\|x\|_\Sigma^2$, where $\sigma = \mathrm{vec}(\Sigma)$. We shall use these two notations interchangeably. The convenience of the vector notation is that it allows us to exploit the simpler linear relation (275) between σ' and σ to rewrite (278) as shown in (279) below, with the *same* weight vector σ appearing on both sides.

Theorem* 6.6.** *(Variance Relation***) Consider the data model of Sec. 6.1 and the independence statistical conditions imposed on the noise and regression data, including (208)–(215). Assume further sufficiently small step-sizes are used so that terms that depend on higher-powers of the step-sizes can be ignored. Pick left stochastic matrices A_1 and A_2 and a right stochastic matrix C satisfying (166). Under these conditions, the weight-error vector $\widetilde{\boldsymbol{w}}_i = \mathrm{col}\{\widetilde{\boldsymbol{w}}_{k,i}\}_{k=1}^N$ associated with a network running the adaptive diffusion strategy (201)–(203) satisfies the following variance relation*

$$\boxed{\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_\sigma^2 \;=\; \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 \;+\; \left[\mathrm{vec}\left(\mathcal{Y}^T\right)\right]^T \sigma} \tag{279}$$

for any Hermitian nonnegative-definite matrix Σ with $\sigma = \mathrm{vec}(\Sigma)$, and where $\{\mathcal{S}, \mathcal{G}, \mathcal{F}\}$ are defined by (241), (263), and (277), and

$$\boxed{\mathcal{Y} \;\triangleq\; \mathcal{G}\mathcal{S}\mathcal{G}^T} \tag{280}$$

□

Note that relation (279) 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 [4, 5, 18, 57] and is not necessary for the exposition here, except to say that stability of the matrix $\mathcal{F}$ ensures the mean-square stability of the filter — this fact is also established further ahead through relation (327). By mean-square stability we mean that each term $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2$ remains bounded over time and converges to a steady-state MSD_k value. Moreover, the spectral radius of $\mathcal{F}$ controls the rate of convergence of $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2$ towards its steady-state value.

Theorem* 6.7.** *(Mean-Square Stability***) Consider the same setting of Theorem 6.6. The adaptive diffusion strategy (201)–(203) is mean-square stable if, and only if, the matrix $\mathcal{F}$ defined by (276), or its approximation (277), is stable (i.e., all its eigenvalues lie strictly inside the unit disc). This condition is satisfied by sufficiently small positive step-sizes $\{\mu_k\}$ that also satisfy:*

$$\mu_k < \frac{2}{\lambda_{\max}(R_k)} \tag{281}$$

where the neighborhood covariance matrix R_k is defined by (182). Moreover, the convergence rate of the algorithm is determined by the value $[\rho(\mathcal{B})]^2$ (the square of the spectral radius of $\mathcal{B}$).

Proof. Recall that, for two arbitrary matrices A and B of compatible dimensions, the eigenvalues of the Kronecker product $A \otimes B$ is formed of all product combinations $\lambda_i(A)\lambda_j(B)$ of the eigenvalues of A and B [19]. Therefore, using expression (277), we have that $\rho(\mathcal{F}) = [\rho(\mathcal{B})]^2$. It follows that $\mathcal{F}$ is stable if, and only if, $\mathcal{B}$ is stable. We already noted earlier in Theorem 6.1 that condition (281) ensures the stability of $\mathcal{B}$. Therefore, step-sizes that ensure stability in the mean and are sufficiently small will also ensure mean-square stability.

□

Remark. More generally, had we not ignored the second-order term (271), the expression for $\mathcal{F}$ would have been the following. Starting from the definition $\Sigma' = \mathbb{E}\,\boldsymbol{\mathcal{B}}_i^* \Sigma \boldsymbol{\mathcal{B}}_i$, we would get

$$\sigma' \;=\; \left(\mathbb{E}\,\boldsymbol{\mathcal{B}}_i^T \otimes \boldsymbol{\mathcal{B}}_i^*\right)\sigma$$

so that

$$\begin{aligned}\mathcal{F} &\triangleq \mathbb{E}\left(\boldsymbol{\mathcal{B}}_i^T \otimes \boldsymbol{\mathcal{B}}_i^*\right) \quad \text{(for general step-sizes)} \\ &= (\mathcal{A}_1 \otimes \mathcal{A}_1)\cdot\Big\{I \;-\; \left(\mathcal{R}^T\mathcal{M}\otimes I\right) \;-\; (I\otimes \mathcal{R}\mathcal{M}) \;+\; \mathbb{E}\left(\boldsymbol{\mathcal{R}}_i^T\mathcal{M}\otimes \boldsymbol{\mathcal{R}}_i\mathcal{M}\right)\Big\}\cdot(\mathcal{A}_2\otimes\mathcal{A}_2)\end{aligned} \tag{282}$$

Mean-square stability of the filter would then require the step-sizes $\{\mu_k\}$ to be chosen such that they ensure the stability of this matrix $\mathcal{F}$ (in addition to condition (281) to ensure mean stability).

□

6.6 Network Mean-Square Performance

We can now use the variance relation (279) to evaluate the network performance, as well as the performance of the individual nodes, in steady-state. Since the dynamics is mean-square stable for sufficiently small step-sizes, we take the limit of (279) as $i \to \infty$ and write:

$$\lim_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_\sigma^2 \;=\; \lim_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 \;+\; \left[\mathrm{vec}\left(\mathcal{Y}^T\right)\right]^T\sigma \tag{283}$$

Grouping terms leads to the following result.

Corollary* 6.1.** *(Steady-State Variance Relation***) Consider the same setting of Theorem 6.6. The weight-error vector, $\widetilde{\boldsymbol{w}}_i = \mathrm{col}\{\widetilde{\boldsymbol{w}}_{k,i}\}_{k=1}^N$, of the adaptive diffusion strategy (201)–(203) satisfies the following relation in steady-state:*

$$\boxed{\lim_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_{(I-\mathcal{F})\sigma}^2 \;=\; \left[\mathrm{vec}\left(\mathcal{Y}^T\right)\right]^T\sigma} \tag{284}$$

for any Hermitian nonnegative-definite matrix Σ with $\sigma = \mathrm{vec}(\Sigma)$, and where $\{\mathcal{F},\mathcal{Y}\}$ are defined by (277) and (280).

□

Expression (284) is a very useful relation; it allows us to evaluate the network MSD and EMSE through proper selection of the weighting vector σ (or, equivalently, the weighting matrix Σ). For example, the network MSD is defined as the average value:

$$\mathrm{MSD}^{\mathrm{network}} \;\triangleq\; \lim_{i\to\infty} \frac{1}{N}\sum_{k=1}^N \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \tag{285}$$

which amounts to averaging the MSDs of the individual nodes. Therefore,

$$\mathrm{MSD}^{\mathrm{network}} \;=\; \lim_{i\to\infty} \frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2 \;=\; \lim_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_{1/N}^2 \tag{286}$$

This means that in order to recover the network MSD from relation (284), we should select the weighting vector σ such that

$$(I-\mathcal{F})\sigma \;=\; \frac{1}{N}\mathrm{vec}\left(I_{NM}\right)$$

Solving for σ and substituting back into (284) we arrive at the following expression for the network MSD:

$$\boxed{\mathrm{MSD}^{\mathrm{network}} \;=\; \frac{1}{N}\cdot\left[\mathrm{vec}\left(\mathcal{Y}^T\right)\right]^T\cdot(I-\mathcal{F})^{-1}\cdot\mathrm{vec}\left(I_{NM}\right)} \tag{287}$$

Likewise, the network EMSE is defined as the average value

$$\begin{aligned}\text{EMSE}^{\text{network}} &\triangleq \lim_{i\to\infty}\ \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\,|\boldsymbol{e}_{a,k}(i)|^2 \\ &= \lim_{i\to\infty}\ \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2_{R_{u,k}}\end{aligned} \tag{288}$$

which amounts to averaging the EMSEs of the individual nodes. Therefore,

$$\text{EMSE}^{\text{network}} \;=\; \lim_{i\to\infty}\ \frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2_{\text{diag}\{R_{u,1},R_{u,2},\ldots,R_{u,N}\}} \;=\; \lim_{i\to\infty}\ \frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2_{\mathcal{R}_u} \tag{289}$$

where $\mathcal{R}_u$ is the matrix defined earlier by (184), and which we repeat below for ease of reference:

$$\mathcal{R}_u = \text{diag}\{R_{u,1}, R_{u,2}, \ldots, R_{u,N}\} \tag{290}$$

This means that in order to recover the network EMSE from relation (284), we should select the weighting vector σ such that

$$(I-\mathcal{F})\sigma \;=\; \frac{1}{N}\text{vec}\,(\mathcal{R}_u) \tag{291}$$

Solving for σ and substituting into (284) we arrive at the following expression for the network EMSE:

$$\boxed{\text{EMSE}^{\text{network}} \;=\; \frac{1}{N}\cdot\left[\text{vec}\,\left(\mathcal{Y}^T\right)\right]^T\cdot(I-\mathcal{F})^{-1}\cdot\text{vec}\,(\mathcal{R}_u)} \tag{292}$$

6.7 Mean-Square Performance of Individual Nodes

We can also assess the mean-square performance of the individual nodes in the network from (284). For instance, the MSD of any particular node k is defined by

$$\text{MSD}_k \;\triangleq\; \lim_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2 \tag{293}$$

Introduce the $N\times N$ block diagonal matrix with blocks of size $M\times M$, where all blocks on the diagonal are zero except for an identity matrix on the diagonal block of index k, i.e.,

$$\mathcal{J}_k \;\triangleq\; \text{diag}\{\,0_M,\ldots,0_M,I_M,0_M,\ldots,0_M\,\} \tag{294}$$

Then, we can express the node MSD as follows:

$$\text{MSD}_k \;\triangleq\; \lim_{i\to\infty}\ \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2_{\mathcal{J}_k} \tag{295}$$

The same argument that was used to obtain the network MSD then leads to

$$\boxed{\text{MSD}_k \;=\; \left[\text{vec}\,\left(\mathcal{Y}^T\right)\right]^T\cdot(I-\mathcal{F})^{-1}\cdot\text{vec}\,(\mathcal{J}_k)} \tag{296}$$

Likewise, the EMSE of node k is defined by

$$\begin{aligned}\text{EMSE}_k &\triangleq \lim_{i\to\infty}\mathbb{E}\,|\boldsymbol{e}_{a,k}(i)|^2 \\ &= \lim_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2_{R_{u,k}}\end{aligned} \tag{297}$$

Introduce the $N \times N$ block diagonal matrix with blocks of size $M \times M$, where all blocks on the diagonal are zero except for the diagonal block of index k whose value is $R_{u,k}$, i.e.,

$$\mathcal{T}_k \triangleq \text{diag}\{ 0_M, \ldots, 0_M, R_{u,k}, 0_M, \ldots, 0_M \} \tag{298}$$

Then, we can express the node EMSE as follows:

$$\text{EMSE}_k \triangleq \lim_{i \to \infty} \mathbb{E} \|\widetilde{\boldsymbol{w}}_i\|^2_{\mathcal{T}_k} \tag{299}$$

The same argument that was used to obtain the network EMSE then leads to

$$\boxed{\text{EMSE}_k = \left[\text{vec}\left(\mathcal{Y}^T\right)\right]^T \cdot (I - \mathcal{F})^{-1} \cdot \text{vec}\left(\mathcal{T}_k\right)} \tag{300}$$

We summarize the results in the following statement.

Theorem* 6.8.** *(Network Mean-Square Performance***) Consider the same setting of Theorem 6.6. Introduce the* $1 \times (NM)^2$ *row vector* h^T *defined by*

$$h^T \triangleq \left[\text{vec}\left(\mathcal{Y}^T\right)\right]^T \cdot (I - \mathcal{F})^{-1} \tag{301}$$

where $\{\mathcal{F}, \mathcal{Y}\}$ *are defined by (277) and (280). Then the network MSD and EMSE and the individual node performance measures are given by*

$$\begin{aligned}
\text{MSD}^{\text{network}} &= h^T \cdot \text{vec}\left(I_{NM}\right)/N && (302)\\
\text{EMSE}^{\text{network}} &= h^T \cdot \text{vec}\left(\mathcal{R}_u\right)/N && (303)\\
\text{MSD}_k &= h^T \cdot \text{vec}\left(\mathcal{J}_k\right) && (304)\\
\text{EMSE}_k &= h^T \cdot \text{vec}\left(\mathcal{T}_k\right) && (305)
\end{aligned}$$

where $\{\mathcal{J}_k, \mathcal{T}_k\}$ *are defined by (294) and (298).*

□

We can obviously recover from the above expressions the performance of the nodes in the non-cooperative implementation (207), where each node performs its adaptation individually, by setting $A_1 = A_2 = C = I_N$.

We can express the network MSD, and its EMSE if desired, in an alternative useful form involving a series representation.

Corollary* 6.2.** *(Series Representation for Network MSD***) Consider the same setting of Theorem 6.6. The network MSD can be expressed in the following alternative series expansion form:*

$$\boxed{\text{MSD}^{\text{network}} = \frac{1}{N} \sum_{j=0}^{\infty} \text{Tr}\left(\mathcal{B}^j \mathcal{Y} \mathcal{B}^{*j}\right)} \tag{306}$$

where

$$\begin{aligned}
\mathcal{Y} &= \mathcal{G}\mathcal{S}\mathcal{G}^T && (307)\\
\mathcal{G} &= \mathcal{A}_2^T \mathcal{M} \mathcal{C}^T && (308)\\
\mathcal{B} &= \mathcal{A}_2^T (I - \mathcal{M}\mathcal{R}) \mathcal{A}_1^T && (309)
\end{aligned}$$

Proof. Since $\mathcal{F}$ is stable when the filter is mean-square stable, we can expand $(I - \mathcal{F})^{-1}$ as

$$\begin{aligned}
(I - \mathcal{F})^{-1} &= I + \mathcal{F} + \mathcal{F}^2 + \ldots \\
&\overset{(277)}{=} I + \left(\mathcal{B}^T \otimes \mathcal{B}^*\right) + \left(\mathcal{B}^T \otimes \mathcal{B}^*\right)^2 + \ldots
\end{aligned}$$

Substituting into (287) and using property (274), we obtain the desired result. □

6.8 Uniform Data Profile

We can simplify expressions (307)–(309) for $\{\mathcal{Y}, \mathcal{G}, \mathcal{B}\}$ in the case when the regression covariance matrices are uniform across the network and all nodes employ the same step-size, i.e., when

$$R_{u,k} = R_u, \quad \text{for all } k \quad \text{(uniform covariance profile)} \tag{310}$$

$$\mu_k = \mu, \quad \text{for all } k \quad \text{(uniform step-sizes)} \tag{311}$$

and when the combination matrix C is doubly stochastic, so that

$$\boxed{C\mathbb{1} = \mathbb{1}, \quad C^T\mathbb{1} = \mathbb{1}} \tag{312}$$

We refer to conditions (310)–(312) as corresponding to a *uniform data profile* environment. The noise variances, $\{\sigma_{v,k}^2\}$, do not need to be uniform so that the signal-to-noise ratio (SNR) across the network can still vary from node to node. The simplified expressions derived in the sequel will be useful in Sec. 7 when we compare the performance of various cooperation strategies.

Thus, under conditions (310)–(312), expressions (180), (181), and (263) for $\{\mathcal{M}, \mathcal{R}, \mathcal{G}\}$ simplify to

$$\mathcal{M} = \mu I_{NM} \tag{313}$$

$$\mathcal{R} = I_N \otimes R_u \tag{314}$$

$$\mathcal{G} = \mu \mathcal{A}_2^T \mathcal{C}^T \tag{315}$$

Substituting these values into expression (309) for $\mathcal{B}$ we get

$$\begin{aligned}
\mathcal{B} &= \mathcal{A}_2^T(I - \mathcal{M}\mathcal{R})\mathcal{A}_1^T \\
&= (A_2^T \otimes I) \cdot (I - \mu(I \otimes R_u)) \cdot (A_1^T \otimes I) \\
&= (A_2^T \otimes I)(A_1^T \otimes I) \;-\; \mu(A_2^T \otimes I)(I \otimes R_u)(A_1^T \otimes I) \\
&= (A_2^T A_1^T \otimes I) \;-\; \mu(A_2^T A_1^T \otimes R_u) \\
&= A_2^T A_1^T \otimes (I - \mu R_u)
\end{aligned} \tag{316}$$

where we used the useful Kronecker product identities:

$$(X + Y) \otimes Z = (X \otimes Z) \;+\; (Y \otimes Z) \tag{317}$$

$$(X \otimes Y)(W \otimes Z) = (XW \otimes YZ) \tag{318}$$

for any matrices $\{X, Y, Z, W\}$ of compatible dimensions. Likewise, introduce the $N \times N$ diagonal matrix with noise variances:

$$\boxed{R_v \;\triangleq\; \text{diag}\{\sigma_{v,1}^2,\; \sigma_{v,2}^2, \ldots, \sigma_{v,N}^2\}} \tag{319}$$

Then, expression (241) for $\mathcal{S}$ becomes

$$\begin{aligned}
\mathcal{S} &= \text{diag}\{\sigma_{v,1}^2 R_u,\; \sigma_{v,2}^2 R_u, \ldots,\; \sigma_{v,N}^2 R_u\} \\
&= R_v \otimes R_u
\end{aligned} \tag{320}$$

It then follows that we can simplify expression (307) for $\mathcal{Y}$ as:

$$\begin{aligned}
\mathcal{Y} &= \mu^2 \mathcal{A}_2^T \mathcal{C}^T \mathcal{S} \mathcal{C} \mathcal{A}_2 \\
&= \mu^2 \cdot (A_2^T \otimes I) \cdot (C^T \otimes I) \otimes (R_v \otimes R_u) \cdot (C \otimes I) \cdot (A_2 \otimes I) \\
&= \mu^2 (A_2^T C^T R_v C A_2 \otimes R_u)
\end{aligned} \tag{321}$$

Corollary* 6.3.** *(Network MSD for Uniform Data Profile***) Consider the same setting of Theorem 6.6 with the additional requirement that conditions (310)–(312) for a uniform data profile hold. The network MSD is still given by the same series representation (306) where now*

$$\mathcal{Y} = \mu^2(A_2^T C^T R_v C A_2 \otimes R_u) \tag{322}$$

$$\mathcal{B} = A_2^T A_1^T \otimes (I - \mu R_u) \tag{323}$$

Using these expressions, we can decouple the network MSD expression (306) into two separate factors: one is dependent on the step-size and data covariance $\{\mu, R_u\}$, and the other is dependent on the combination matrices and noise profile $\{A_1, A_2, C, R_v\}$:

$$\boxed{\mathrm{MSD}^{\mathrm{network}} = \frac{\mu^2}{N}\sum_{j=0}^{\infty} \mathrm{Tr}\left(\left[\left(A_2^T A_1^T\right)^j \left(A_2^T C^T R_v C A_2\right)(A_1 A_2)^j\right] \otimes \left[(I-\mu R_u)^j R_u (I-\mu R_u)^j\right]\right)} \tag{324}$$

Proof. Using (306) and the given expressions (322)–(323) for $\{\mathcal{Y}, \mathcal{B}\}$, we get

$$\mathrm{MSD}^{\mathrm{network}} = \frac{\mu^2}{N}\sum_{j=0}^{\infty} \mathrm{Tr}\left(\left[\left(A_2^T A_1^T\right)^j \otimes (I-\mu R_u)^j\right](A_2^T C^T R_v C A_2 \otimes R_u)\left[(A_1 A_2)^j \otimes (I-\mu R_u)^j\right]\right)$$

Result (324) follows from property (317).

□

6.9 Transient Mean-Square Performance

Before comparing the mean-square performance of various cooperation strategies, we pause to comment that the variance relation (279) can also be used to characterize the transient behavior of the network, and not just its steady-state performance. To see this, iterating (279) starting from $i = 0$, we find that

$$\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_\sigma^2 = \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{-1}\|_{\mathcal{F}^{i+1}\sigma}^2 + \left[\mathrm{vec}\left(\mathcal{Y}^T\right)\right]^T \cdot \left(\sum_{j=0}^{i} \mathcal{F}^j \sigma\right) \tag{325}$$

where

$$\widetilde{\boldsymbol{w}}_{-1} \triangleq w^o - \boldsymbol{w}_{-1} \tag{326}$$

in terms of the initial condition, $\boldsymbol{w}_{-1}$. If this initial condition happens to be $\boldsymbol{w}_{-1} = 0$, then $\widetilde{\boldsymbol{w}}_{-1} = w^o$. Comparing expression (325) at time instants i and $i-1$ we can relate $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_\sigma^2$ and $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|_\sigma^2$ as follows:

$$\boxed{\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_\sigma^2 = \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|_\sigma^2 + \left[\mathrm{vec}\left(\mathcal{Y}^T\right)\right]^T \cdot \mathcal{F}^i\sigma - \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{-1}\|_{(I-\mathcal{F})\mathcal{F}^i\sigma}^2} \tag{327}$$

This recursion relates the same weighted square measures of the error vectors $\{\widetilde{\boldsymbol{w}}_i, \widetilde{\boldsymbol{w}}_{i-1}\}$. It therefore describes how these weighted square measures evolve over time. It is clear from this relation that, for mean-square stability, the matrix $\mathcal{F}$ needs to be stable so that the terms involving $\mathcal{F}^i$ do not grow unbounded.

The learning curve of the network is the curve that describes the evolution of the network EMSE over time. At any time i, the network EMSE is denoted by $\zeta(i)$ and measured as:

$$\zeta(i) \triangleq \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\,|\boldsymbol{e}_{a,k}(i)|^2 \tag{328}$$

$$= \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|_{R_{u,k}}^2$$

The above expression indicates that $\zeta(i)$ is obtained by averaging the EMSE of the individual nodes at time i. Therefore,

$$\zeta(i) \;=\; \frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2_{\mathrm{diag}\{R_{u,1},R_{u,2},\ldots,R_{u,N}\}} \;=\; \frac{1}{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|^2_{\mathcal{R}_u} \tag{329}$$

where $\mathcal{R}_u$ is the matrix defined by (290). This means that in order to evaluate the evolution of the network EMSE from relation (327), we simply select the weighting vector σ such that

$$\sigma \;=\; \frac{1}{N}\mathrm{vec}\,(\mathcal{R}_u) \tag{330}$$

Substituting into (327) we arrive at the learning curve for the network.

Corollary* 6.4.** *(Network Learning Curve***) Consider the same setting of Theorem 6.6. Let $\zeta(i)$ denote the network EMSE at time i, as defined by (328). Then, the learning curve of the network corresponds to the evolution of $\zeta(i)$ with time and is described by the following recursion over $i \geq 0$:*

$$\boxed{\zeta(i) \;=\; \zeta(i-1) \;+\; \frac{1}{N}\left[\mathrm{vec}\left(\mathcal{Y}^T\right)\right]^T \cdot \mathcal{F}^i \cdot \mathrm{vec}\,(\mathcal{R}_u) \;-\; \frac{1}{N}\mathbb{E}\,\|\tilde{\boldsymbol{w}}_{-1}\|^2_{(I-\mathcal{F})\mathcal{F}^i\mathrm{vec}(\mathcal{R}_u)}} \tag{331}$$

where $\{\mathcal{F},\mathcal{Y},\mathcal{R}_u\}$ are defined by (277), (280), and (290).

□

7 Comparing the Performance of Cooperative Strategies

Using the expressions just derived for the MSD of the network, we can compare the performance of various cooperative and non-cooperative strategies. Table 6 further ahead summarizes the results derived in this section and the conditions under which they hold.

7.1 Comparing ATC and CTA Strategies

We first compare the performance of the adaptive ATC and CTA diffusion strategies (153) and (154) when they employ a *doubly* stochastic combination matrix A. That is, let us consider the two scenarios:

$$C,\;\; A_1 = A,\;\; A_2 = I_N \qquad \text{(adaptive CTA strategy)} \tag{332}$$

$$C,\;\; A_1 = I_N,\;\; A_2 = A \qquad \text{(adaptive ATC strategy)} \tag{333}$$

where A is now assumed to be doubly stochastic, i.e.,

$$A\mathbb{1} = \mathbb{1}, \qquad A^T\mathbb{1} = \mathbb{1} \tag{334}$$

with its rows and columns adding up to one. For example, these conditions are satisfied when A is left stochastic and symmetric. Then, expressions (307) and (309) give:

$$\mathcal{B}_{\mathrm{cta}} \;=\; (I-\mathcal{M}\mathcal{R})\mathcal{A}^T, \qquad \mathcal{Y}_{\mathrm{cta}} \;=\; \mathcal{M}\mathcal{C}^T\mathcal{S}\mathcal{C}\mathcal{M} \tag{335}$$

$$\mathcal{B}_{\mathrm{atc}} \;=\; \mathcal{A}^T(I-\mathcal{M}\mathcal{R}), \qquad \mathcal{Y}_{\mathrm{atc}} \;=\; \mathcal{A}^T\mathcal{M}\mathcal{C}^T\mathcal{S}\mathcal{C}\mathcal{M}\mathcal{A} \tag{336}$$

where

$$\mathcal{A} = A \otimes I_M \tag{337}$$

Following [18], introduce the auxiliary nonnegative-definite matrix

$$\mathcal{H}_j \;\triangleq\; \left[(I-\mathcal{M}\mathcal{R})\mathcal{A}^T\right]^j \cdot \mathcal{M}\mathcal{C}^T\mathcal{S}\mathcal{C}\mathcal{M} \cdot \left[(I-\mathcal{M}\mathcal{R})\mathcal{A}^T\right]^{*j} \tag{338}$$

Then, it is immediate to verify from (306) that

$$\mathrm{MSD}_{\mathrm{cta}}^{\mathrm{network}} = \frac{1}{N}\sum_{j=0}^{\infty}\mathrm{Tr}(\mathcal{H}_j) \tag{339}$$

$$\mathrm{MSD}_{\mathrm{atc}}^{\mathrm{network}} = \frac{1}{N}\sum_{j=0}^{\infty}\mathrm{Tr}(\mathcal{A}^T\,\mathcal{H}_j\,\mathcal{A}) \tag{340}$$

so that

$$\mathrm{MSD}_{\mathrm{cta}}^{\mathrm{network}} - \mathrm{MSD}_{\mathrm{atc}}^{\mathrm{network}} = \frac{1}{N}\sum_{j=0}^{\infty}\mathrm{Tr}\left(\mathcal{H}_j - \mathcal{A}^T\mathcal{H}_j\mathcal{A}\right) \tag{341}$$

Now, since A is doubly stochastic, it also holds that the enlarged matrix $\mathcal{A}$ is doubly stochastic. Moreover, for any doubly stochastic matrix $\mathcal{A}$ and any nonnegative-definite matrix $\mathcal{H}$ of compatible dimensions, it holds that (see part (f) of Theorem C.3):

$$\mathrm{Tr}(\mathcal{A}^T\mathcal{H}\mathcal{A}) \le \mathrm{Tr}(\mathcal{H}) \tag{342}$$

Applying result (342) to (341) we conclude that

$$\boxed{\mathrm{MSD}_{\mathrm{atc}}^{\mathrm{network}} \le \mathrm{MSD}_{\mathrm{cta}}^{\mathrm{network}}} \quad \text{(doubly stochastic } A\text{)} \tag{343}$$

so that the adaptive ATC strategy (153) outperforms the adaptive CTA strategy (154) for doubly stochastic combination matrices A.

7.2 Comparing Strategies with and without Information Exchange

We now examine the effect of information exchange ($C \neq I$) on the performance of the adaptive ATC and CTA diffusion strategies (153)–(154) under conditions (310)–(312) for *uniform data profile*.

CTA Strategies

We start with the adaptive CTA strategy (154), and consider two scenarios with and without information exchange. These scenarios correspond to the following selections in the general description (201)–(203):

$$C \neq I,\ \ A_1 = A,\ \ A_2 = I_N \qquad \text{(adaptive CTA with information exchange)} \tag{344}$$

$$C = I,\ \ A_1 = A,\ \ A_2 = I_N \qquad \text{(adaptive CTA without information exchange)} \tag{345}$$

Then, expressions (322) and (323) give:

$$\mathcal{B}_{\mathrm{cta,C\neq I}} = A^T \otimes (I - \mu R_u),\quad \mathcal{Y}_{\mathrm{cta,C\neq I}} = \mu^2(C^T R_v C \otimes R_u) \tag{346}$$

$$\mathcal{B}_{\mathrm{cta,C=I}} = A^T \otimes (I - \mu R_u),\quad \mathcal{Y}_{\mathrm{cta,C=I}} = \mu^2(R_v \otimes R_u) \tag{347}$$

where the matrix R_v is defined by (319). Note that $\mathcal{B}_{\mathrm{cta,C\neq I}} = \mathcal{B}_{\mathrm{cta,C=I}}$, so we denote them simply by $\mathcal{B}$ in the derivation that follows. Then, from expression (306) for the network MSD we get:

$$\mathrm{MSD}_{\mathrm{cta},C=I}^{\mathrm{network}} - \mathrm{MSD}_{\mathrm{cta},C\neq I}^{\mathrm{network}} = \frac{\mu^2}{N}\sum_{j=0}^{\infty}\mathrm{Tr}\left(\mathcal{B}^j\left[(R_v - C^T R_v C)\otimes R_u\right]\mathcal{B}^{*j}\right) \tag{348}$$

It follows that the difference in performance between both CTA implementations depends on how the matrices R_v and $C^T R_v C$ compare to each other:

(1) When $R_v - C^T R_v C \geq 0$, we obtain

$$\boxed{\text{MSD}^{\text{network}}_{\text{cta},C=I} \;\geq\; \text{MSD}^{\text{network}}_{\text{cta},C\neq I}} \qquad (\text{when } C^T R_v C \leq R_v) \tag{349}$$

so that a CTA implementation with information exchange performs better than a CTA implementation without information exchange. Note that the condition on $\{R_v, C\}$ corresponds to requiring

$$C^T R_v C \leq R_v \tag{350}$$

which can be interpreted to mean that the cooperation matrix C should be such that it does not amplify the effect of measurement noise. For example, this situation occurs when the noise profile is uniform across the network, in which case $R_v = \sigma_v^2 I_M$. This is because it would then hold that

$$R_v - C^T R_v C \;\;=\;\; \sigma_v^2 (I - C^T C) \geq 0 \tag{351}$$

in view of the fact that $(I - C^T C) \geq 0$ since C is doubly stochastic (cf. property (e) in Lemma C.3).

(2) When $R_v - C^T R_v C \leq 0$, we obtain

$$\boxed{\text{MSD}^{\text{network}}_{\text{cta},C=I} \;\leq\; \text{MSD}^{\text{network}}_{\text{cta},C\neq I}} \qquad (\text{when } C^T R_v C \geq R_v) \tag{352}$$

so that a CTA implementation without information exchange performs better than a CTA implementation with information exchange. In this case, the condition on $\{R_v, C\}$ indicates that the combination matrix C ends up amplifying the effect of noise.

ATC Strategies

We can repeat the argument for the adaptive ATC strategy (153), and consider two scenarios with and without information exchange. These scenarios correspond to the following selections in the general description (201)–(203):

$$C \neq I, \;\; A_1 = I_N, \;\; A_2 = A \qquad (\text{adaptive ATC with information exchange }) \tag{353}$$

$$C = I, \;\; A_1 = I_N, \;\; A_2 = A \qquad (\text{adaptive ATC without information exchange}) \tag{354}$$

Then, expressions (322) and (323) give:

$$\mathcal{B}_{\text{atc,C}\neq\text{I}} \;\;=\;\; A^T \otimes (I - \mu R_u), \quad \mathcal{Y}_{\text{atc,C}\neq\text{I}} \;=\; \mu^2 (A^T C^T R_v C A \otimes R_u) \tag{355}$$

$$\mathcal{B}_{\text{atc,C=I}} \;\;=\;\; A^T \otimes (I - \mu R_u), \quad \mathcal{Y}_{\text{atc,C=I}} \;=\; \mu^2 (A^T R_v A \otimes R_u) \tag{356}$$

Note again that $\mathcal{B}_{\text{atc,C}\neq\text{I}} = \mathcal{B}_{\text{atc,C=I}}$, so we denote them simply by $\mathcal{B}$. Then,

$$\text{MSD}^{\text{network}}_{\text{atc},C=I} - \text{MSD}^{\text{network}}_{\text{atc},C\neq I} \;\;=\;\; \frac{\mu^2}{N} \sum_{j=0}^{\infty} \text{Tr}\left(\mathcal{B}^j \left[A^T (R_v - C^T R_v C) A \otimes R_u\right] \mathcal{B}^{*j}\right) \tag{357}$$

It again follows that the difference in performance between both ATC implementations depends on how the matrices R_v and $C^T R_v C$ compare to each other and we obtain:

$$\boxed{\text{MSD}^{\text{network}}_{\text{atc},C=I} \;\geq\; \text{MSD}^{\text{network}}_{\text{atc},C\neq I}} \qquad (\text{when } C^T R_v C \leq R_v) \tag{358}$$

and

$$\boxed{\text{MSD}^{\text{network}}_{\text{atc},C=I} \;\leq\; \text{MSD}^{\text{network}}_{\text{atc},C\neq I}} \qquad (\text{when } C^T R_v C \geq R_v) \tag{359}$$

Table 6: Comparison of the MSD performance of various cooperative strategies.

Comparison	Conditions
$\mathrm{MSD}_{\mathrm{atc}}^{\mathrm{network}} \leq \mathrm{MSD}_{\mathrm{cta}}^{\mathrm{network}}$	A doubly stochastic, C right stochastic.
$\mathrm{MSD}_{\mathrm{cta},C\neq I}^{\mathrm{network}} \leq \mathrm{MSD}_{\mathrm{cta},C=I}^{\mathrm{network}}$	$C^T R_v C \leq R_v$, C doubly stochastic, $R_{u,k} = R_u$, $\mu_k = \mu$.
$\mathrm{MSD}_{\mathrm{cta},C=I}^{\mathrm{network}} \leq \mathrm{MSD}_{\mathrm{cta},C\neq I}^{\mathrm{network}}$	$C^T R_v C \geq R_v$, C doubly stochastic, $R_{u,k} = R_u$, $\mu_k = \mu$.
$\mathrm{MSD}_{\mathrm{atc},C\neq I}^{\mathrm{network}} \leq \mathrm{MSD}_{\mathrm{atc},C=I}^{\mathrm{network}}$	$C^T R_v C \leq R_v$, C doubly stochastic, $R_{u,k} = R_u$, $\mu_k = \mu$.
$\mathrm{MSD}_{\mathrm{atc},C=I}^{\mathrm{network}} \leq \mathrm{MSD}_{\mathrm{atc},C\neq I}^{\mathrm{network}}$	$C^T R_v C \geq R_v$, C doubly stochastic, $R_{u,k} = R_u$, $\mu_k = \mu$.
$\mathrm{MSD}_{\mathrm{atc}}^{\mathrm{network}} \leq \mathrm{MSD}_{\mathrm{cta}}^{\mathrm{network}} \leq \mathrm{MSD}_{\mathrm{lms}}^{\mathrm{network}}$	$\{A, C\}$ doubly stochastic, $R_{u,k} = R_u$, $\mu_k = \mu$.

7.3 Comparing Diffusion Strategies with the Non-Cooperative Strategy

We now compare the performance of the adaptive CTA strategy (154) to the non-cooperative LMS strategy (207) assuming conditions (310)–(312) for uniform data profile. These scenarios correspond to the following selections in the general description (201)–(203):

$$C, \;\; A_1 = A, \;\; A_2 = I \qquad \text{(adaptive CTA)} \tag{360}$$

$$C = I, \;\; A_1 = I, \;\; A_2 = I \qquad \text{(non-cooperative LMS)} \tag{361}$$

where A is further assumed to be doubly stochastic (along with C) so that

$$A\mathbb{1} = \mathbb{1}, \qquad A^T\mathbb{1} = \mathbb{1} \tag{362}$$

Then, expressions (322) and (323) give:

$$\mathcal{B}_{\mathrm{cta}} = A^T \otimes (I - \mu R_u), \qquad \mathcal{Y}_{\mathrm{cta}} = \mu^2 (C^T R_v C \otimes R_u) \tag{363}$$

$$\mathcal{B}_{\mathrm{lms}} = I \otimes (I - \mu R_u), \qquad \mathcal{Y}_{\mathrm{lms}} = \mu^2 (R_v \otimes R_u) \tag{364}$$

Now recall that

$$\mathcal{C} = C \otimes I_M \tag{365}$$

so that, using the Kronecker product property (317),

$$\begin{aligned} \mathcal{Y}_{\mathrm{cta}} &= \mu^2 (C^T R_v C \otimes R_u) \\ &= \mu^2 (C^T \otimes I_M)(R_v \otimes R_u)(C \otimes I_M) \\ &= \mu^2 \mathcal{C}^T (R_v \otimes R_u) \mathcal{C} \\ &= \mathcal{C}^T \mathcal{Y}_{\mathrm{lms}} \, \mathcal{C} \end{aligned} \tag{366}$$

Then,

$$\begin{aligned} \mathrm{MSD}_{\mathrm{lms}}^{\mathrm{network}} - \mathrm{MSD}_{\mathrm{cta}}^{\mathrm{network}} &= \frac{1}{N} \sum_{j=0}^{\infty} \mathrm{Tr}\left(\mathcal{B}_{\mathrm{lms}}^j \mathcal{Y}_{\mathrm{lms}} \mathcal{B}_{\mathrm{lms}}^{*j} \right) - \frac{1}{N} \sum_{j=0}^{\infty} \mathrm{Tr}\left(\mathcal{B}_{\mathrm{cta}}^j \mathcal{C}^T \mathcal{Y}_{\mathrm{lms}} \mathcal{C} \mathcal{B}_{\mathrm{cta}}^{*j} \right) \\ &= \frac{1}{N} \sum_{j=0}^{\infty} \mathrm{Tr}\left(\mathcal{B}_{\mathrm{lms}}^{*j} \mathcal{B}_{\mathrm{lms}}^j \mathcal{Y}_{\mathrm{lms}} \right) - \frac{1}{N} \sum_{j=0}^{\infty} \mathrm{Tr}\left(\mathcal{C} \mathcal{B}_{\mathrm{cta}}^{*j} \mathcal{B}_{\mathrm{cta}}^j \mathcal{C}^T \mathcal{Y}_{\mathrm{lms}} \right) \\ &= \frac{1}{N} \sum_{j=0}^{\infty} \mathrm{Tr}\left[\left(\mathcal{B}_{\mathrm{lms}}^{*j} \mathcal{B}_{\mathrm{lms}}^j - \mathcal{C} \mathcal{B}_{\mathrm{cta}}^{*j} \mathcal{B}_{\mathrm{cta}}^j \mathcal{C}^T \right) \mathcal{Y}_{\mathrm{lms}} \right] \end{aligned} \tag{367}$$

Let us examine the difference:

$$\begin{aligned}\mathcal{B}_{\rm lms}^{*j}\mathcal{B}_{\rm lms}^{j} \;-\; \mathcal{C}\mathcal{B}_{\rm cta}^{*j}\mathcal{B}_{\rm cta}^{j}\mathcal{C}^T \;&=\; \big(I\otimes(I-\mu R_u)^{2j}\big) \;-\; \big(CA^j\otimes(I-\mu R_u)^{j}\big)\big(A^{jT}C^T\otimes(I-\mu R_u)^{j}\big)\\ &\overset{(317)}{=}\; \big(I\otimes(I-\mu R_u)^{2j}\big) \;-\; \big(CA^jA^{jT}C^T\otimes(I-\mu R_u)^{2j}\big)\\ &=\; (I-\;CA^jA^{jT}C^T)\otimes(I-\mu R_u)^{2j} \end{aligned} \tag{368}$$

Now, due to the even power, it always holds that $(I-\mu R_u)^{2j}\geq 0$. Moreover, since A^j and C are doubly stochastic, it follows that $CA^jA^{jT}C^T$ is also doubly stochastic. Therefore, the matrix $(I-CA^jA^{jT}C^T)$ is nonnegative-definite as well (cf. property (e) of Lemma C.3). It follows that

$$\mathcal{B}_{\rm lms}^{*j}\mathcal{B}_{\rm lms}^{j} \;-\; \mathcal{C}\mathcal{B}_{\rm cta}^{*j}\mathcal{B}_{\rm cta}^{j}\mathcal{C}^T \;\geq\; 0 \tag{369}$$

But since $\mathcal{Y}_{\rm lms}\geq 0$, we conclude from (367) that

$$\boxed{\mathrm{MSD}_{\rm lms}^{\rm network}\geq \mathrm{MSD}_{\rm cta}^{\rm network}} \tag{370}$$

This is because for any two Hermitian nonnegative-definite matrices A and B of compatible dimensions, it holds that $\mathrm{Tr}(AB)\geq 0$; indeed if we factor $B=XX^*$ with X full rank, then $\mathrm{Tr}(AB)=\mathrm{Tr}(X^*AX)\geq 0$. We conclude from this analysis that adaptive CTA diffusion performs better than non-cooperative LMS under uniform data profile conditions *and* doubly stochastic A. If we refer to the earlier result (343), we conclude that the following relation holds:

$$\boxed{\mathrm{MSD}_{\rm atc}^{\rm network}\;\leq\;\mathrm{MSD}_{\rm cta}^{\rm network}\;\leq\;\mathrm{MSD}_{\rm lms}^{\rm network}} \tag{371}$$

Table 6 lists the comparison results derived in this section and lists the conditions under which the conclusions hold.

8 Selecting the Combination Weights

The adaptive diffusion strategy (201)–(203) employs combination weights $\{a_{1,\ell k},a_{2,\ell k},c_{\ell k}\}$ or, equivalently, combination matrices $\{A_1,A_2,C\}$, where A_1 and A_2 are left-stochastic matrices and C is a right-stochastic matrix. There are several ways by which these matrices can be selected. In this section, we describe constructions that result in left-stochastic or doubly-stochastic combination matrices, A. When a right-stochastic combination matrix is needed, such as C, then it can be obtained by transposition of the left-stochastic constructions shown below.

8.1 Constant Combination Weights

Table 7 lists a couple of common choices for selecting constant combination weights for a network with N nodes. Several of these constructions appeared originally in the literature on graph theory. In the table, the symbol n_k denotes the degree of node k, which refers to the size of its neighborhood. Likewise, the symbol $n_{\max}$ refers to the maximum degree across the network, i.e.,

$$n_{\max}\;=\;\max_{1\leq k\leq N}\{n_k\} \tag{372}$$

The Laplacian rule, which appears in the second line of the table, relies on the use of the Laplacian matrix $\mathcal{L}$ of the network and a positive scalar γ. The Laplacian matrix is defined by (574) in App. B, namely, it is a symmetric matrix whose entries are constructed as follows [64–66]:

$$[\mathcal{L}]_{k\ell}=\begin{cases} n_k-1, & \text{if } k=\ell\\ -1, & \text{if } k\neq\ell \text{ and nodes } k \text{ and } \ell \text{ are neighbors}\\ 0, & \text{otherwise}\end{cases} \tag{373}$$

The Laplacian rule can be reduced to other forms through the selection of the positive parameter γ. One choice is $\gamma = 1/n_{\max}$, while another choice is $\gamma = 1/N$ and leads to the maximum-degree rule. Obviously, it always holds that $n_{\max} \leq N$ so that $1/n_{\max} \geq 1/N$. Therefore, the choice $\gamma = 1/n_{\max}$ ends up assigning larger weights to neighbors than the choice $\gamma = 1/N$. The averaging rule in the first row of the table is one of the simplest combination rules whereby nodes simply average data from their neighbors.

Table 7: Selections for combination matrices $A = [a_{\ell k}]$.

Entries of Combination Matrix A	Type of A
1. Averaging rule [68]: $a_{\ell k} = \begin{cases} 1/n_k, & \text{if } k \neq \ell \text{ are neighbors or } k = \ell \\ 0, & \text{otherwise} \end{cases}$	left-stochastic
2. Laplacian rule [49, 69]: $A = I_N - \gamma\mathcal{L},\ \gamma > 0$	symmetric and doubly-stochastic
3. Laplacian rule using $\gamma = 1/n_{\max}$: $a_{\ell k} = \begin{cases} 1/n_{\max}, & \text{if } k \neq \ell \text{ are neighbors} \\ 1 - (n_k - 1)/n_{\max}, & k = \ell \\ 0, & \text{otherwise} \end{cases}$	symmetric and doubly-stochastic
4. Laplacian rule using $\gamma = 1/N$(maximum-degree rule [50]) : $a_{\ell k} = \begin{cases} 1/N, & \text{if } k \neq \ell \text{ are neighbors} \\ 1 - (n_k - 1)/N, & k = \ell \\ 0, & \text{otherwise} \end{cases}$	symmetric and doubly-stochastic
5. Metropolis rule [49, 70, 71]: $a_{\ell k} = \begin{cases} 1/\max\{n_k, n_\ell\}, & \text{if } k \neq \ell \text{ are neighbors} \\ 1 - \sum_{\ell \in \mathcal{N}_k \setminus \{k\}} a_{\ell k}, & k = \ell \\ 0, & \text{otherwise} \end{cases}$	symmetric and doubly-stochastic
6. Relative-degree rule [29]: $a_{\ell k} = \begin{cases} n_\ell / \left(\sum_{m \in \mathcal{N}_k} n_m \right), & \text{if } k \text{ and } \ell \text{ are neighbors or } k = \ell \\ 0, & \text{otherwise} \end{cases}$	left-stochastic

In the constructions in Table 7, the values of the weights $\{a_{\ell k}\}$ are largely dependent on the degree of the nodes. In this way, the number of connections that each node has influences the combination weights with its neighbors. While such selections may be appropriate in some applications, they can nevertheless degrade the performance of adaptation over networks [54]. This is because such weighting schemes ignore the noise profile across the network. And since some nodes can be noisier than others, it is not sufficient to rely solely on the amount of connectivity that nodes have to determine the combination weights to their neighbors. It is important to take into account the amount of noise that is present at the nodes as well. Therefore, designing combination rules that are aware of the variation in noise profile across the network is an important task.

It is also important to devise strategies that are able to *adapt* these combination weights in response to variations in network topology and data statistical profile. For this reason, following [58, 62], we describe in the next subsection one adaptive procedure for adjusting the combination weights. This procedure allows the network to assign more or less relevance to nodes according to the quality of their data.

8.2 Optimizing the Combination Weights

Ideally, we would like to select $N \times N$ combination matrices $\{A_1, A_2, C\}$ in order to minimize the network MSD given by (302) or (306). In [18], the selection of the combination weights was formulated as the following optimization problem:

$$\boxed{\begin{array}{l} \min\limits_{\{A_1,A_2,C\}} \quad \text{MSD}^{\text{network}} \ \text{ given by (302) or (306)} \\ \text{over left and right-stochastic matrices with nonnegative entries:} \\ \quad A_1^T\mathbb{1} = \mathbb{1}, \quad a_{1,\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \\ \quad A_2^T\mathbb{1} = \mathbb{1}, \quad a_{2,\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \\ \quad C\mathbb{1} = \mathbb{1}, \qquad c_{\ell k} = 0 \ \text{ if } \ell \notin \mathcal{N}_k \end{array}} \tag{374}$$

We can pursue a numerical solution to (374) in order to search for optimal combination matrices, as was done in [18]. Here, however, we are interested in an adaptive solution that becomes part of the learning process so that the network can adapt the weights on the fly in response to network conditions. We illustrate an approximate approach from [58, 62] that leads to one adaptive solution that performs reasonably well in practice.

We illustrate the construction by considering the ATC strategy (158) without information exchange where $A_1 = I_N$, $A_2 = A$, and $C = I$. In this case, recursions (204)–(206) take the form:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^* \left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}\right] \tag{375}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell,i} \tag{376}$$

and, from (306), the corresponding network MSD performance is:

$$\text{MSD}_{\text{atc}}^{\text{network}} = \frac{1}{N}\sum_{j=0}^{\infty} \text{Tr}\left(\mathcal{B}_{\text{atc}}^j \mathcal{Y}_{\text{atc}} \mathcal{B}_{\text{atc}}^{*j}\right) \tag{377}$$

where

$$\mathcal{B}_{\text{atc}} = \mathcal{A}^T(I - \mathcal{M}\mathcal{R}_u) \tag{378}$$

$$\mathcal{Y}_{\text{atc}} = \mathcal{A}^T\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A} \tag{379}$$

$$\mathcal{R}_u = \text{diag}\{R_{u,1}, R_{u,2}, \ldots, R_{u,N}\} \tag{380}$$

$$\mathcal{S} = \text{diag}\{\sigma_{v,1}^2 R_{u,1},\ \sigma_{v,2}^2 R_{u,2}, \ldots, \sigma_{v,N}^2 R_{u,N}\} \tag{381}$$

$$\mathcal{M} = \text{diag}\{\mu_1 I_M,\ \mu_2 I_M, \ldots, \mu_N I_M\} \tag{382}$$

$$\mathcal{A} = A \otimes I_M \tag{383}$$

Minimizing the MSD expression (377) over left-stochastic matrices A is generally non-trivial. We pursue an approximate solution.

To begin with, for compactness of notation, let r denote the spectral radius of the $N \times N$ block matrix $I - \mathcal{M}\mathcal{R}_u$:

$$r \triangleq \rho(I - \mathcal{M}\mathcal{R}_u) \tag{384}$$

We already know, in view of the mean and mean-square stability of the network, that $|r| < 1$. Now, consider the series that appears in (377) and whose trace we wish to minimize over A. Note that its block maximum norm can be bounded as follows:

$$\begin{aligned}
\left\|\sum_{j=0}^{\infty}\mathcal{B}_{\rm atc}^{j}\mathcal{Y}_{\rm atc}\mathcal{B}_{\rm atc}^{*j}\right\|_{b,\infty} &\leq \sum_{j=0}^{\infty}\left\|\mathcal{B}_{\rm atc}^{j}\right\|_{b,\infty}\cdot\|\mathcal{Y}_{\rm atc}\|_{b,\infty}\cdot\left\|\mathcal{B}_{\rm atc}^{*j}\right\|_{b,\infty} \\
&\overset{(a)}{\leq} N\cdot\left(\sum_{j=0}^{\infty}\left\|\mathcal{B}_{\rm atc}^{j}\right\|_{b,\infty}^{2}\cdot\|\mathcal{Y}_{\rm atc}\|_{b,\infty}\right) \\
&\leq N\cdot\left(\sum_{j=0}^{\infty}\|\mathcal{B}_{\rm atc}\|_{b,\infty}^{2j}\cdot\|\mathcal{Y}_{\rm atc}\|_{b,\infty}\right) \\
&\overset{(b)}{\leq} N\cdot\left(\sum_{j=0}^{\infty}r^{2j}\cdot\|\mathcal{Y}_{\rm atc}\|_{b,\infty}\right) \\
&= \frac{N}{1-r^{2}}\cdot\|\mathcal{Y}_{\rm atc}\|_{b,\infty} \qquad (385)
\end{aligned}$$

where for step (b) we use result (602) to conclude that

$$\begin{aligned}
\|\mathcal{B}_{\rm atc}\|_{b,\infty} &= \|\mathcal{A}^{T}(I-\mathcal{M}\mathcal{R}_{u})\|_{b,\infty} \\
&\leq \|\mathcal{A}^{T}\|_{b,\infty}\cdot\|I-\mathcal{M}\mathcal{R}_{u}\|_{b,\infty} \\
&= \|I-\mathcal{M}\mathcal{R}_{u}\|_{b,\infty} \\
&\overset{(602)}{=} r \qquad (386)
\end{aligned}$$

To justify step (a), we use result (584) to relate the norms of $\mathcal{B}_{\rm atc}^{j}$ and its complex conjugate, $\left[\mathcal{B}_{\rm atc}^{j}\right]^{*}$, as

$$\|\mathcal{B}_{\rm atc}^{*j}\|_{b,\infty} \;\leq\; N\cdot\|\mathcal{B}_{\rm atc}^{j}\|_{b,\infty} \qquad (387)$$

Expression (385) then shows that the norm of the series appearing in (377) is bounded by a scaled multiple of the norm of $\mathcal{Y}_{\rm atc}$, and the scaling constant is independent of A. Using property (586) we conclude that there exists a positive constant c, also independent of A, such that

$$\mathrm{Tr}\left(\sum_{j=0}^{\infty}\mathcal{B}_{\rm atc}^{j}\mathcal{Y}_{\rm atc}\mathcal{B}_{\rm atc}^{*j}\right) \;\leq\; c\cdot\mathrm{Tr}(\mathcal{Y}_{\rm atc}) \qquad (388)$$

Therefore, instead of attempting to minimize the trace of the series, the above result motivates us to minimize an upper bound to the trace. Thus, we consider the alternative problem of minimizing the first term of the series (377), namely,

$$\begin{aligned}
&\min_{A} \quad \mathrm{Tr}(\mathcal{Y}_{\rm atc}) \\
&\text{subject to } \; A^{T}\mathbb{1}=\mathbb{1}, \;\; a_{\ell k}\geq 0, \;\; a_{\ell k}=0 \text{ if } \ell\notin\mathcal{N}_{k}
\end{aligned} \qquad (389)$$

Using (379), the trace of $\mathcal{Y}_{\rm atc}$ can be expressed in terms of the combination coefficients as follows:

$$\mathrm{Tr}(\mathcal{Y}_{\rm atc}) \;=\; \sum_{k=1}^{N}\sum_{\ell=1}^{N}\mu_{\ell}^{2}\;a_{\ell k}^{2}\;\sigma_{v,\ell}^{2}\;\mathrm{Tr}(R_{u,\ell}) \qquad (390)$$

so that problem (389) can be decoupled into N separate optimization problems of the form:

$$\boxed{\begin{aligned}&\min_{\{a_{\ell k}\}_{\ell=1}^{N}} \quad \sum_{\ell=1}^{N} \mu_\ell^2 \, a_{\ell k}^2 \, \sigma_{v,\ell}^2 \, \mathrm{Tr}(R_{u,\ell}), \quad k = 1, \ldots, N \\ &\text{subject to} \\ &\qquad a_{\ell k} \geq 0, \qquad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k\end{aligned}} \tag{391}$$

With each node ℓ, we associate the following nonnegative noise-data-dependent measure:

$$\boxed{\gamma_\ell^2 \;\triangleq\; \mu_\ell^2 \cdot \sigma_{v,\ell}^2 \cdot \mathrm{Tr}(R_{u,\ell})} \tag{392}$$

This measure amounts to scaling the noise variance at node ℓ by μ_ℓ^2 and by the power of the regression data (measured through the trace of its covariance matrix). We shall refer to γ_ℓ^2 as the noise-data variance product (or *variance product*, for simplicity) at node ℓ. Then, the solution of (391) is given by:

$$\boxed{a_{\ell k} = \begin{cases} \dfrac{\gamma_\ell^{-2}}{\sum_{m \in \mathcal{N}_k} \gamma_m^{-2}}, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}} \quad \text{(relative-variance rule)} \tag{393}$$

We refer to this combination rule as the *relative-variance combination rule* [58]; it leads to a left-stochastic matrix A. In this construction, node k combines the intermediate estimates $\{\boldsymbol{\psi}_{\ell,i}\}$ from its neighbors in (376) in proportion to the inverses of their variance products, $\{\gamma_m^{-2}\}$. The result is physically meaningful. Nodes with smaller variance products will generally be given larger weights. In comparison, the following *relative-degree-variance rule* was proposed in [18] (a typo appears in Table III in [18], where the noise variances appear written in the table instead of their inverses):

$$\boxed{a_{\ell k} = \begin{cases} \dfrac{n_\ell \sigma_{v,\ell}^{-2}}{\sum_{m \in \mathcal{N}_k} n_m \sigma_{v,m}^{-2}}, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}} \quad \text{(relative degree-variance rule)} \tag{394}$$

This second form also leads to a left-stochastic combination matrix A. However, rule (394) does not take into account the covariance matrices of the regression data across the network. Observe that in the special case when step-sizes, the regression covariance matrices, and the noise variances are uniform across the network, i.e., $\mu_k = \mu$, $R_{u,k} = R_u$, and $\sigma_{v,k}^2 = \sigma_v^2$ for all k, expression (393) reduces to the simple averaging rule (first line of Table 7). In contrast, expression (394) reduces the relative degree rule (last line of Table 7).

8.3 Adaptive Combination Weights

To evaluate the combination weights (393), the nodes need to know the variance products, $\{\gamma_m^2\}$, of their neighbors. According to (392), the factors $\{\gamma_m^2\}$ are defined in terms of the noise variances, $\{\sigma_{v,m}^2\}$, and the regression covariance matrices, $\{\mathrm{Tr}(R_{u,m})\}$, and these quantities are not known beforehand. The nodes only have access to realizations of $\{\boldsymbol{d}_m(i), \boldsymbol{u}_{m,i}\}$. We now describe one procedure that allows every node k to learn the variance products of its neighbors in an adaptive manner. Note that if a particular node ℓ happens to belong to two neighborhoods, say, the neighborhood of node k_1 and the neighborhood of node k_2, then

each of k_1 and k_2 need to evaluate the variance product, γ_ℓ^2, of node ℓ. The procedure described below allows each node in the network to estimate the variance products of its neighbors in a recursive manner.

To motivate the algorithm, we refer to the ATC recursion (375)–(376) and use the data model (208) to write for node ℓ:

$$\boldsymbol{\psi}_{\ell,i} = \boldsymbol{w}_{\ell,i-1} \;+\; \mu_\ell \boldsymbol{u}_{\ell,i}^* \left[\boldsymbol{u}_{\ell,i}\widetilde{\boldsymbol{w}}_{\ell,i-1} + \boldsymbol{v}_\ell(i)\right] \tag{395}$$

so that, in view of our earlier assumptions on the regression data and noise in Sec. 6.1, we obtain in the limit as $i \to \infty$:

$$\lim_{i\to\infty} \;\mathbb{E}\,\left\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{\ell,i-1}\right\|^2 \;=\; \mu_\ell^2 \cdot \left(\lim_{i\to\infty} \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}\,(\boldsymbol{u}_{\ell,i}^*\|\boldsymbol{u}_{\ell,i}\|^2\boldsymbol{u}_{\ell,i})}\right) \;+\; \mu_\ell^2 \cdot \sigma_{v,\ell}^2 \cdot \mathrm{Tr}(R_{u,\ell}) \tag{396}$$

We can evaluate the limit on the right-hand side by using the steady-state result (284). Indeed, we select the vector σ in (284) to satisfy

$$(I - \mathcal{F})\sigma \;=\; \mathrm{vec}\left[\mathbb{E}\,\left(\boldsymbol{u}_{\ell,i}^*\|\boldsymbol{u}_{\ell,i}\|^2\boldsymbol{u}_{\ell,i}\right)\right] \tag{397}$$

Then, from (284),

$$\lim_{i\to\infty} \;\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}\,(\boldsymbol{u}_{\ell,i}^*\|\boldsymbol{u}_{\ell,i}\|^2\boldsymbol{u}_{\ell,i})} \;=\; \left[\mathrm{vec}\left(\mathcal{Y}^T\right)\right]^T \cdot (I-\mathcal{F})^{-1} \cdot \mathrm{vec}\left[\mathbb{E}\,\left(\boldsymbol{u}_{\ell,i}^*\|\boldsymbol{u}_{\ell,i}\|^2\boldsymbol{u}_{\ell,i}\right)\right] \tag{398}$$

Now recall from expression (379) for $\mathcal{Y}$ that for the ATC algorithm under consideration we have

$$\mathcal{Y} = \mathcal{A}^T\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A} \tag{399}$$

so that the entries of $\mathcal{Y}$ depend on combinations of the squared step-sizes, $\{\mu_m^2,\; m = 1, 2, \ldots, N\}$. This fact implies that the first term on the right-hand side of (396) depends on products of the form $\{\mu_\ell^2\mu_m^2\}$; these fourth-order factors can be ignored in comparison to the second-order factor μ_ℓ^2 for small step-sizes so that

$$\begin{aligned}\lim_{i\to\infty} \;\mathbb{E}\,\left\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{\ell,i-1}\right\|^2 \;&\approx\; \mu_\ell^2 \cdot \sigma_{v,\ell}^2 \cdot \mathrm{Tr}(R_{u,\ell}) \\ &=\; \gamma_\ell^2\end{aligned} \tag{400}$$

in terms of the desired variance product, γ_ℓ^2. Using the following instantaneous approximation at node k (where $w_{\ell,i-1}$ is replaced by $w_{k,i-1}$):

$$\mathbb{E}\,\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{\ell,i-1}\|^2 \;\approx\; \|\psi_{\ell,i} - w_{k,i-1}\|^2 \tag{401}$$

we can motivate an algorithm that enables node k to estimate the variance product, γ_ℓ^2, of its neighbor ℓ. Thus, let $\gamma_{\ell k}^2(i)$ denote an estimate for γ_ℓ^2 that is computed by node k at time i. Then, one way to evaluate $\gamma_{\ell k}^2(i)$ is through the recursion:

$$\boxed{\gamma_{\ell k}^2(i) = (1-\nu_k)\cdot\gamma_{\ell k}^2(i-1) \;+\; \nu_k \cdot \|\psi_{\ell,i} - w_{k,i-1}\|^2} \tag{402}$$

where $0 < \nu_k \ll 1$ is a positive coefficient smaller than one. Note that under expectation, expression (402) becomes

$$\mathbb{E}\boldsymbol{\gamma}_{\ell k}^2(i) = (1-\nu_k)\cdot\mathbb{E}\boldsymbol{\gamma}_{\ell k}^2(i-1) \;+\; \nu_k \cdot \mathbb{E}\,\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2 \tag{403}$$

so that in steady-state, as $i \to \infty$,

$$\lim_{i\to\infty} \;\mathbb{E}\boldsymbol{\gamma}_{\ell k}^2(i) \;\approx\; (1-\nu_k)\cdot\lim_{i\to\infty} \;\mathbb{E}\boldsymbol{\gamma}_{\ell k}^2(i-1) \;+\; \nu_k\cdot\gamma_\ell^2 \tag{404}$$

Hence, we obtain

$$\lim_{i\to\infty} \;\mathbb{E}\boldsymbol{\gamma}_{\ell k}^2(i) \approx \gamma_\ell^2 \tag{405}$$

That is, the estimator $\boldsymbol{\gamma}_{\ell k}^2(i)$ converges on average close to the desired variance product γ_ℓ^2. In this way, we can replace the optimal weights (393) by the adaptive construction:

$$\boxed{a_{\ell k}(i) = \begin{cases} \frac{\gamma_{\ell k}^{-2}(i)}{\sum_{m \in \mathcal{N}_k} \gamma_{mk}^{-2}(i)}, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}} \tag{406}$$

Equations (402) and (406) provide one adaptive construction for the combination weights $\{a_{\ell k}\}$.

9 Diffusion with Noisy Information Exchanges

The adaptive diffusion strategy (201)–(203) relies on the fusion of local information collected from neighborhoods through the use of combination matrices $\{A_1, A_2, C\}$. In the previous section, we described several constructions for selecting such combination matrices. We also motivated and developed an adaptive scheme for the ATC mode of operation (375)–(376) that computes combination weights in a manner that is aware of the variation of the variance-product profile across the network. Nevertheless, in addition to the measurement noises $\{\boldsymbol{v}_k(i)\}$ at the individual nodes, we also need to consider the effect of perturbations that are introduced during the exchange of information among neighboring nodes. Noise over the communication links can be due to various factors including thermal noise and imperfect channel information. Studying the degradation in mean-square performance that results from these noisy exchanges can be pursued by straightforward extension of the mean-square analysis of Sec. 6, as we proceed to illustrate. Subsequently, we shall use the results to show how the combination weights can also be adapted in the presence of noisy exchange links.

9.1 Noise Sources over Exchange Links

To model noisy links, we introduce an additive noise component into each of the steps of the diffusion strategy (201)–(203) during the operations of information exchange among the nodes. The notation becomes a bit cumbersome because we need to account for both the source and destination of the information that is being exchanged. For example, the same signal $\boldsymbol{d}_\ell(i)$ that is generated by node ℓ will be broadcast to all the neighbors of node ℓ. When this is done, a different noise will interfere with the exchange of $\boldsymbol{d}_\ell(i)$ over each of the edges that link node ℓ to its neighbors. Thus, we will need to use a notation of the form $\boldsymbol{d}_{\ell k}(i)$, with two subscripts ℓ and k, to indicate that this is the noisy version of $\boldsymbol{d}_\ell(i)$ that is received by node k from node ℓ. The subscript ℓk indicates that ℓ is the source and k is the sink, i.e., information is moving from ℓ to k. For the reverse situation where information flows from node k to ℓ, we would use instead the subscript $k\ell$.

With this notation in mind, we model the noisy data received by node k from its neighbor ℓ as follows:

$$\boldsymbol{w}_{\ell k,i-1} = \boldsymbol{w}_{\ell,i-1} + \boldsymbol{v}_{\ell k,i-1}^{(w)} \tag{407}$$

$$\boldsymbol{\psi}_{\ell k,i} = \boldsymbol{\psi}_{\ell,i} + \boldsymbol{v}_{\ell k,i}^{(\psi)} \tag{408}$$

$$\boldsymbol{u}_{\ell k,i} = \boldsymbol{u}_{\ell,i} + \boldsymbol{v}_{\ell k,i}^{(u)} \tag{409}$$

$$\boldsymbol{d}_{\ell k}(i) = \boldsymbol{d}_\ell(i) + \boldsymbol{v}_{\ell k}^{(d)}(i) \tag{410}$$

where $\boldsymbol{v}_{\ell k,i-1}^{(w)}$ $(M \times 1)$, $\boldsymbol{v}_{\ell k,i}^{(\psi)}$ $(M \times 1)$, and $\boldsymbol{v}_{\ell k,i}^{(u)}$ $(1 \times M)$ are vector noise signals, and $\boldsymbol{v}_{\ell k}^{(d)}(i)$ is a scalar noise signal. These are the noise signals that perturb exchanges over the edge linking source ℓ to sink k (i.e., for data sent from node ℓ to node k). The superscripts $\{(w),\ (\psi),\ (u),\ (d)\}$ in each case refer to the variable that these noises perturb. Figure 14 illustrates the various noise sources that perturb the exchange of information from node ℓ to node k. The figure also shows the measurement noises $\{\boldsymbol{v}_\ell(i), \boldsymbol{v}_k(i)\}$ that exist locally at the nodes in view of the data model (208).

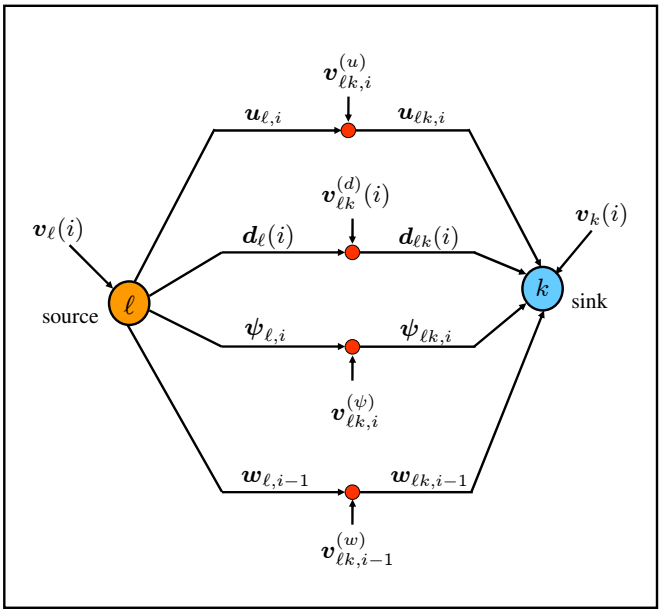

Figure 14: Additive noise sources perturb the exchange of information from node ℓ to node k. The subscript ℓk in this illustration indicates that ℓ is the source node and k is the sink node so that information is flowing from ℓ to k.

We assume that the following noise signals, which influence the data received by node k,

$$\left\{\boldsymbol{v}_k(i),\;\; \boldsymbol{v}_{\ell k}^{(d)}(i),\;\; \boldsymbol{v}_{\ell k,i-1}^{(w)},\;\; \boldsymbol{v}_{\ell k,i}^{(\psi)},\;\; \boldsymbol{v}_{\ell k,i}^{(u)}\right\} \tag{411}$$

are temporally white and spatially independent random processes with zero mean and variances or covariances given by

$$\boxed{\left\{\sigma_{v,k}^2,\;\; \sigma_{v,\ell k}^2,\;\; R_{v,\ell k}^{(w)},\;\; R_{v,\ell k}^{(\psi)},\;\; R_{v,\ell k}^{(u)}\right\}} \tag{412}$$

Obviously, the quantities

$$\left\{\sigma_{v,\ell k}^2,\;\; R_{v,\ell k}^{(w)},\;\; R_{v,\ell k}^{(\psi)},\;\; R_{v,\ell k}^{(u)}\right\} \tag{413}$$

are all zero if $\ell \notin \mathcal{N}_k$ or when $\ell = k$. We further assume that the noise processes (411) are independent of each other and of the regression data $\boldsymbol{u}_{m,j}$ for all k, ℓ, m and i, j.

9.2 Error Recursion

Using the perturbed data (407)–(410), the adaptive diffusion strategy (201)–(203) becomes

$$\boldsymbol{\phi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{1,\ell k}\, \boldsymbol{w}_{\ell k,i-1} \tag{414}$$

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\phi}_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, \boldsymbol{u}_{\ell k,i}^{*}\, [\boldsymbol{d}_{\ell k}(i) - \boldsymbol{u}_{\ell k,i}\boldsymbol{\phi}_{k,i-1}] \tag{415}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{2,\ell k}\, \boldsymbol{\psi}_{\ell k,i} \tag{416}$$

Observe that the perturbed quantities $\{\boldsymbol{w}_{\ell k,i-1}, \boldsymbol{u}_{\ell k,i}, \boldsymbol{d}_{\ell k}(i), \boldsymbol{\psi}_{\ell k,i}\}$, with subscripts ℓk, appear in (414)–(416) in place of the original quantities $\{\boldsymbol{w}_{\ell,i-1}, \boldsymbol{u}_{\ell,i}, \boldsymbol{d}_{\ell}(i), \boldsymbol{\psi}_{\ell,i}\}$ that appear in (201)–(203). This is because these quantities are now subject to exchange noises. As before, we are still interested in examining the evolution of the weight-error vectors:

$$\widetilde{\boldsymbol{w}}_{k,i} \triangleq w^o - \boldsymbol{w}_{k,i}, \quad k = 1, 2, \ldots, N \tag{417}$$

For this purpose, we again introduce the following $N \times 1$ block vector, whose entries are of size $M \times 1$ each:

$$\widetilde{\boldsymbol{w}}_i \triangleq \begin{bmatrix} \widetilde{\boldsymbol{w}}_{1,i} \\ \widetilde{\boldsymbol{w}}_{2,i} \\ \vdots \\ \widetilde{\boldsymbol{w}}_{N,i} \end{bmatrix} \tag{418}$$

and proceed to determine a recursion for its evolution over time. The arguments are largely similar to what we already did before in Sec. 6.3 and, therefore, we shall emphasize the differences that arise. The main deviation is that we now need to account for the presence of the new noise signals; they will contribute additional terms to the recursion for $\widetilde{\boldsymbol{w}}_i$ — see (442) further ahead. We may note that some studies on the effect of imperfect data exchanges on the performance of adaptive diffusion algorithms are considered in [59–61]. However, these earlier investigations were limited to particular cases in which only noise in the exchange of $\boldsymbol{w}_{\ell,i-1}$ was considered (as in (407)), in addition to setting $C = I$ (in which case there is *no* exchange of $\{\boldsymbol{d}_{\ell}(i), \boldsymbol{u}_{\ell,i}\}$), and by focusing on the CTA case for which $A_2 = I$. Here, we consider instead the general case that accounts for the additional sources of imperfections shown in (408)–(410), in addition to the general diffusion strategy (201)–(203) with combination matrices $\{A_1, A_2, C\}$.

To begin with, we introduce the aggregate $M \times 1$ zero-mean noise signals:

$$\boldsymbol{v}_{k,i-1}^{(w)} \triangleq \sum_{\ell \in \mathcal{N}_k} a_{1,\ell k}\, \boldsymbol{v}_{\ell k,i-1}^{(w)}, \qquad \boldsymbol{v}_{k,i}^{(\psi)} \triangleq \sum_{\ell \in \mathcal{N}_k} a_{2,\ell k}\, \boldsymbol{v}_{\ell k,i}^{(\psi)} \tag{419}$$

These noises represent the aggregate effect on node k of all exchange noises from the neighbors of node k while exchanging the estimates $\{\boldsymbol{w}_{\ell,i-1}, \boldsymbol{\psi}_{\ell,i}\}$ during the two combination steps (201) and (203). The $M \times M$ covariance matrices of these noises are given by

$$\boxed{R_{v,k}^{(w)} \triangleq \sum_{\ell \in \mathcal{N}_k} a_{1,\ell k}^2\, R_{v,\ell k}^{(w)}, \qquad R_{v,k}^{(\psi)} \triangleq \sum_{\ell \in \mathcal{N}_k} a_{2,\ell k}^2\, R_{v,\ell k}^{(\psi)}} \tag{420}$$

These expressions aggregate the exchange noise covariances in the neighborhood of node k; the covariances are scaled by the squared coefficients $\{a_{1,\ell k}^2,\ a_{2,\ell k}^2\}$. We collect these noise signals, and their covariances, from across the network into $N \times 1$ block vectors and $N \times N$ block diagonal matrices as follows:

$$\boldsymbol{v}_{i-1}^{(w)} \triangleq \operatorname{col}\left\{\boldsymbol{v}_{1,i-1}^{(w)},\ \boldsymbol{v}_{2,i-1}^{(w)}, \ldots,\ \boldsymbol{v}_{N,i-1}^{(w)}\right\} \tag{421}$$

$$\boldsymbol{v}_{i}^{(\psi)} \triangleq \operatorname{col}\left\{\boldsymbol{v}_{1,i}^{(\psi)},\ \boldsymbol{v}_{2,i}^{(\psi)}, \ldots,\ \boldsymbol{v}_{N,i}^{(\psi)}\right\} \tag{422}$$

$$R_{v}^{(w)} \triangleq \operatorname{diag}\left\{R_{v,1}^{(w)},\ R_{v,2}^{(w)},\ \ldots,\ R_{v,N}^{(w)}\right\} \tag{423}$$

$$R_{v}^{(\psi)} \triangleq \operatorname{diag}\left\{R_{v,1}^{(\psi)},\ R_{v,2}^{(\psi)},\ \ldots,\ R_{v,N}^{(\psi)}\right\} \tag{424}$$

We further introduce the following scalar zero-mean noise signal:

$$\boldsymbol{v}_{\ell k}(i) \triangleq \boldsymbol{v}_{\ell}(i) + \boldsymbol{v}_{\ell k}^{(d)}(i) - \boldsymbol{v}_{\ell k,i}^{(u)}\, w^o \tag{425}$$

whose variance is

$$\boxed{\sigma^2_{\ell k} = \sigma^2_{v,\ell} \;+\; \sigma^2_{v,\ell k} \;+\; w^{o*} R^{(u)}_{v,\ell k} w^o} \tag{426}$$

In the absence of exchange noises for the data $\{\boldsymbol{d}_\ell(i), \boldsymbol{u}_{\ell,i}\}$, the signal $\boldsymbol{v}_{\ell k}(i)$ would coincide with the measurement noise $\boldsymbol{v}_\ell(i)$. Expression (425) is simply a reflection of the aggregate effect of the noises in exchanging $\{\boldsymbol{d}_\ell(i), \boldsymbol{u}_{\ell,i}\}$ on node k. Indeed, starting from the data model (208) and using (409)–(410), we can easily verify that the noisy data $\{\boldsymbol{d}_{\ell k}(i), \boldsymbol{u}_{\ell k,i}\}$ are related via:

$$\boldsymbol{d}_{\ell k}(i) \;=\; \boldsymbol{u}_{\ell k,i} w^o \;+\; \boldsymbol{v}_{\ell k}(i) \tag{427}$$

We also define (compare with (234)–(235) and note that we are now using the perturbed regression vectors $\{\boldsymbol{u}_{\ell k,i}\}$):

$$\boldsymbol{R}'_{k,i} \;\triangleq\; \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{u}^*_{\ell k,i} \boldsymbol{u}_{\ell k,i} \tag{428}$$

$$\boldsymbol{\mathcal{R}}'_i \;\triangleq\; \mathrm{diag}\left\{\boldsymbol{R}'_{1,i},\; \boldsymbol{R}'_{2,i},\; \ldots,\; \boldsymbol{R}'_{N,i}\right\} \tag{429}$$

It holds that

$$\boxed{\mathbb{E}\, \boldsymbol{R}'_{k,i} = R'_k} \tag{430}$$

where

$$\boxed{R'_k \;\triangleq\; \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \left[R_{u,\ell} \;+\; R^{(u)}_{v,\ell k}\right]} \tag{431}$$

When there is no noise during the exchange of the regression data, i.e., when $R^{(u)}_{v,\ell k} = 0$, the expressions for $\{\boldsymbol{R}'_{k,i},\; \boldsymbol{\mathcal{R}}'_i,\; R'_k\}$ reduce to expressions (234)–(235) and (182) for $\{\boldsymbol{R}_{k,i},\; \boldsymbol{\mathcal{R}}_i,\; R_k\}$.

Likewise, we introduce (compare with (239)):

$$\boldsymbol{z}_{k,i} \;\triangleq\; \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \boldsymbol{u}^*_{\ell k,i} \boldsymbol{v}_{\ell k}(i) \tag{432}$$

$$\boldsymbol{z}_i \;\triangleq\; \mathrm{col}\left\{\boldsymbol{z}_{1,i},\; \boldsymbol{z}_{2,i}, \ldots,\; \boldsymbol{z}_{N,i}\right\} \tag{433}$$

Compared with the earlier definition for $\boldsymbol{s}_i$ in (239) when there is no noise over the exchange links, we see that we now need to account for the various noisy versions of the same regression vector $\boldsymbol{u}_{\ell,i}$ and the same signal $\boldsymbol{d}_\ell(i)$. For instance, the vectors $\boldsymbol{u}_{\ell k,i}$ and $\boldsymbol{u}_{\ell m,i}$ would denote two noisy versions received by nodes k and m for the *same* regression vector $\boldsymbol{u}_{\ell,i}$ transmitted from node ℓ. Likewise, the scalars $\boldsymbol{d}_{\ell k}(i)$ and $\boldsymbol{d}_{\ell m}(i)$ would denote two noisy versions received by nodes k and m for the *same* scalar $\boldsymbol{d}_\ell(i)$ transmitted from node ℓ. As a result, the quantity $\boldsymbol{z}_i$ is not zero mean any longer (in contrast to $\boldsymbol{s}_i$, which had zero mean). Indeed, note that

$$\begin{aligned}
\mathbb{E}\, \boldsymbol{z}_{k,i} &= \sum_{\ell \in \mathcal{N}_k} c_{\ell k}\, \mathbb{E}\, \boldsymbol{u}^*_{\ell k,i} \boldsymbol{v}_{\ell k}(i) \\
&= \sum_{\ell \in \mathcal{N}_k} c_{\ell k}\, \mathbb{E}\, \left(\left[\boldsymbol{u}_{\ell,i} + \boldsymbol{v}^{(u)}_{\ell k,i}\right]^* \cdot \left[\boldsymbol{v}_\ell(i) \;+\; \boldsymbol{v}^{(d)}_{\ell k}(i) \;-\; \boldsymbol{v}^{(u)}_{\ell k,i}\, w^o\right]\right) \\
&= -\left(\sum_{\ell \in \mathcal{N}_k} c_{\ell k}\, R^{(u)}_{v,\ell k}\right) w^o
\end{aligned} \tag{434}$$

It follows that

$$\mathbb{E}\,\boldsymbol{z}_i \;=\; -\begin{bmatrix} \displaystyle\sum_{\ell\in\mathcal{N}_1} c_{\ell 1}\, R_{v,\ell 1}^{(u)} \\ \displaystyle\sum_{\ell\in\mathcal{N}_2} c_{\ell 2}\, R_{v,\ell 2}^{(u)} \\ \vdots \\ \displaystyle\sum_{\ell\in\mathcal{N}_N} c_{\ell N}\, R_{v,\ell N}^{(u)} \end{bmatrix} w^o \tag{435}$$

Although we can continue our analysis by studying this general case in which the vectors $\boldsymbol{z}_i$ do not have zero-mean (see [62, 63]), we shall nevertheless limit our discussion in the sequel to the case in which there is no noise during the exchange of the regression data, i.e., we henceforth assume that:

$$\boxed{\boldsymbol{v}_{\ell k,i}^{(u)} = 0, \quad R_{v,\ell k}^{(u)} = 0, \quad \boldsymbol{u}_{\ell k,i} = \boldsymbol{u}_{\ell,i}} \quad \text{(assumption from this point onwards)} \tag{436}$$

We maintain all other noise sources, which occur during the exchange of the weight estimates $\{\boldsymbol{w}_{\ell,i-1}, \boldsymbol{\psi}_{\ell,i}\}$ and the data $\{\boldsymbol{d}_\ell(i)\}$. Under condition (436), we obtain

$$\mathbb{E}\,\boldsymbol{z}_i \;=\; 0 \tag{437}$$

$$\sigma_{\ell k}^2 \;=\; \sigma_{v,\ell}^2 + \sigma_{v,\ell k}^2 \tag{438}$$

$$R_k' \;=\; \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\, R_{u,\ell} \;\overset{(182)}{=}\; R_k \tag{439}$$

Then, the covariance matrix of each term $\boldsymbol{z}_{k,i}$ is given by

$$\boxed{R_{z,k} \;\triangleq\; \sum_{\ell\in\mathcal{N}_k} c_{\ell k}^2\, \sigma_{\ell k}^2\, R_{u,\ell}} \tag{440}$$

and the covariance matrix of $\boldsymbol{z}_i$ is $N \times N$ block diagonal with blocks of size $M \times M$:

$$\boxed{\mathcal{Z} \;\triangleq\; \mathbb{E}\,\boldsymbol{z}_i\boldsymbol{z}_i^* \;=\; \mathrm{diag}\{R_{z,1},\, R_{z,2}, \ldots,\, R_{z,N}\}} \tag{441}$$

Now repeating the argument that led to (246) we arrive at the following recursion for the weight-error vector:

$$\boxed{\widetilde{\boldsymbol{w}}_i = \mathcal{A}_2^T \left(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i'\right) \mathcal{A}_1^T \widetilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^T \mathcal{M}\boldsymbol{z}_i - \mathcal{A}_2^T \left(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i'\right) \boldsymbol{v}_{i-1}^{(w)} - \boldsymbol{v}_i^{(\psi)}} \quad \text{(noisy links)} \tag{442}$$

For comparison purposes, we repeat recursion (246) here (recall that this recursion corresponds to the case when the exchanges over the links are not subject to noise):

$$\widetilde{\boldsymbol{w}}_i = \mathcal{A}_2^T \left(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right) \mathcal{A}_1^T\, \widetilde{\boldsymbol{w}}_{i-1} \;-\; \mathcal{A}_2^T \mathcal{M}\mathcal{C}^T \boldsymbol{s}_i \quad \text{(perfect links)} \tag{443}$$

Comparing (442) and (443) we find that:

(1) The covariance matrix $\boldsymbol{\mathcal{R}}_i$ in (443) is replaced by $\boldsymbol{\mathcal{R}}_i'$. Recall from (429) that $\boldsymbol{\mathcal{R}}_i'$ contains the influence of the noises that arise during the exchange of the regression data, i.e., the $\{R_{v,\ell k}^{(u)}\}$. But since we are now assuming that $R_{v,\ell k}^{(u)} = 0$, then $\boldsymbol{\mathcal{R}}_i' = \boldsymbol{\mathcal{R}}_i$.

(2) The term $\mathcal{C}^T \boldsymbol{s}_i$ in (443) is replaced by $\boldsymbol{z}_i$. Recall from (432) that $\boldsymbol{z}_i$ contains the influence of the noises that arise during the exchange of the measurement data and the regression data, i.e., the $\{\sigma^2_{v,\ell k},\ R^{(u)}_{v,\ell k}\}$.

(3) Two new driving terms appear involving $\boldsymbol{v}^{(w)}_{i-1}$ and $\boldsymbol{v}^{(\psi)}_i$. These terms reflect the influence of the noises during the exchange of the weight estimates $\{\boldsymbol{w}_{\ell,i-1}, \boldsymbol{\psi}_{\ell,i}\}$.

(4) Observe further that:

(4a) The term involving $\boldsymbol{v}^{(w)}_{i-1}$ accounts for noise introduced at the information-exchange step (414) *before* adaptation.

(4b) The term involving $\boldsymbol{z}_i$ accounts for noise introduced during the adaptation step (415).

(4c) The term involving $\boldsymbol{v}^{(\psi)}_i$ accounts for noise introduced at the information-exchange step (416) *after* adaptation.

Therefore, since we are not considering noise during the exchange of the regression data, the weight-error recursion (442) simplifies to:

$$\boxed{\widetilde{\boldsymbol{w}}_i = \mathcal{A}_2^T \left(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right) \mathcal{A}_1^T \widetilde{\boldsymbol{w}}_{i-1} - \mathcal{A}_2^T \mathcal{M} \boldsymbol{z}_i - \mathcal{A}_2^T \left(I_{NM} - \mathcal{M}\boldsymbol{\mathcal{R}}_i\right) \boldsymbol{v}^{(w)}_{i-1} - \boldsymbol{v}^{(\psi)}_i} \quad \text{(noisy links)} \tag{444}$$

where we used the fact that $\boldsymbol{\mathcal{R}}'_i = \boldsymbol{\mathcal{R}}_i$ under these conditions.

9.3 Convergence in the Mean

Taking expectations of both sides of (444) we find that the mean error vector evolves according to the following recursion:

$$\boxed{\mathbb{E}\,\widetilde{\boldsymbol{w}}_i = \mathcal{A}_2^T \left(I_{NM} - \mathcal{M}\mathcal{R}\right) \mathcal{A}_1^T \cdot \mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1},\ \ i \geq 0} \tag{445}$$

with $\mathcal{R}$ defined by (181). This is the same recursion encountered earlier in (248) during perfect data exchanges. Note that had we considered noises during the exchange of the regression data, then the vector $\boldsymbol{z}_i$ in (444) *would not* be zero mean and the matrix $\boldsymbol{\mathcal{R}}'_i$ will have to be used instead of $\boldsymbol{\mathcal{R}}_i$. In that case, the recursion for $\mathbb{E}\,\widetilde{\boldsymbol{w}}_i$ will be different from (445); i.e., the presence of noise during the exchange of regression data alters the dynamics of the mean error vector in an important way — see [62, 63] for details on how to extend the arguments to this general case with a driving non-zero bias term. We can now extend Theorem 6.1 to the current scenario.

Theorem* 9.1.** *(Convergence in the Mean***) Consider the problem of optimizing the global cost (92) with the individual cost functions given by (93). Pick a right stochastic matrix C and left stochastic matrices A_1 and A_2 satisfying (166). Assume each node in the network runs the perturbed adaptive diffusion algorithm (414)–(416). Assume further that the exchange of the variables $\{\boldsymbol{w}_{\ell,i-1}, \boldsymbol{\psi}_{\ell,i}, \boldsymbol{d}_\ell(i)\}$ is subject to additive noises as in (407), (408), and (410). We assume that the regressors are exchanged unperturbed. Then, all estimators $\{\boldsymbol{w}_{k,i}\}$ across the network will still converge in the mean to the optimal solution w^o if the step-size parameters $\{\mu_k\}$ satisfy*

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_k)}} \tag{446}$$

where the neighborhood covariance matrix R_k is defined by (182). That is, $\mathbb{E}\,\boldsymbol{w}_{k,i} \to w^o$ as $i \to \infty$.

□

9.4 Mean-Square Convergence

Recall from (264) that we introduced the matrix:

$$\mathcal{B} \triangleq \mathcal{A}_2^T \left(I_{NM} - \mathcal{M}\mathcal{R}\right) \mathcal{A}_1^T \tag{447}$$

We further introduce the $N \times N$ block matrix with blocks of size $M \times M$ each:

$$\mathcal{H} \triangleq \mathcal{A}_2^T (I_{NM} - \mathcal{M}\mathcal{R}) \tag{448}$$

Then, starting from (444) and repeating the argument that led to (279) we can establish the validity of the following variance relation:

$$\boxed{\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_i\|_\sigma^2 = \mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{i-1}\|_{\mathcal{F}\sigma}^2 + \left[\mathrm{vec}\left(\mathcal{A}_2^T \mathcal{M}\mathcal{Z}^T \mathcal{M}\mathcal{A}_2\right) + \mathrm{vec}\left(\left(\mathcal{H}\mathcal{R}_v^{(w)T}\mathcal{H}^*\right)^T\right) + \mathrm{vec}\left(\mathcal{R}_v^{(\psi)T}\right)\right]^T \sigma} \tag{449}$$

for an arbitrary nonnegative-definite weighting matrix Σ with $\sigma = \mathrm{vec}(\Sigma)$, and where $\mathcal{F}$ is the same matrix defined earlier either by (276) or (277). We can therefore extend the statement of Theorem 6.7 to the present scenario.

Theorem* 9.2.** *(Mean-Square Stability***) Consider the same setting of Theorem 9.1. Assume sufficiently small step-sizes to justify ignoring terms that depend on higher powers of the step-sizes. The perturbed adaptive diffusion algorithm (414)–(416) is mean-square stable if, and only if, the matrix $\mathcal{F}$ defined by (276), or its approximation (277), is stable (i.e., all its eigenvalues are strictly inside the unit disc). This condition is satisfied by sufficiently small step-sizes $\{\mu_k\}$ that are also smaller than:*

$$\boxed{\mu_k < \frac{2}{\lambda_{\max}(R_k)}} \tag{450}$$

where the neighborhood covariance matrix R_k is defined by (182). Moreover, the convergence rate of the algorithm is determined by $[\rho(\mathcal{B})]^2$.

□

We conclude from the previous two theorems that the conditions for the mean and mean-square convergence of the adaptive diffusion strategy are not affected by the presence of noises over the exchange links (under the assumption that the regression data are exchanged without perturbation; otherwise, the convergence conditions would be affected). The mean-square performance, on the other hand, is affected as follows. Introduce the $N \times N$ block matrix:

$$\boxed{\mathcal{Y}_{\text{imperfect}} \triangleq \mathcal{A}_2^T \mathcal{M}\mathcal{Z}\mathcal{M}\mathcal{A}_2 + \mathcal{H}\mathcal{R}_v^{(w)}\mathcal{H}^* + \mathcal{R}_v^{(\psi)}} \quad \text{(imperfect exchanges)} \tag{451}$$

which should be compared with the corresponding quantity defined by (280) for the perfect exchanges case, namely,

$$\mathcal{Y}_{\text{perfect}} = \mathcal{A}_2^T \mathcal{M}\mathcal{C}^T \mathcal{S}\mathcal{C}\mathcal{M}\mathcal{A}_2 \quad \text{(perfect exchanges)} \tag{452}$$

When perfect exchanges occur, the matrix $\mathcal{Z}$ reduces to $\mathcal{C}^T\mathcal{S}\mathcal{C}$. We can relate $\mathcal{Y}_{\text{imperfect}}$ and $\mathcal{Y}_{\text{perfect}}$ as follows. Let

$$\mathcal{R}^{(du)} \triangleq \mathrm{diag}\left\{\sum_{\ell\in\mathcal{N}_1} c_{\ell 1}^2 \sigma_{v,\ell 1}^2 R_{u,\ell},\ \sum_{\ell\in\mathcal{N}_2} c_{\ell 2}^2 \sigma_{v,\ell 2}^2 R_{u,\ell},\ \ldots \sum_{\ell\in\mathcal{N}_N} c_{\ell N}^2 \sigma_{v,\ell N}^2 R_{u,\ell}\right\} \tag{453}$$

Then, using (438) and (441), it is straightforward to verify that

$$\mathcal{Z} = \mathcal{C}^T \mathcal{S} \mathcal{C} \; + \; \mathcal{R}^{(du)} \tag{454}$$

and it follows that:

$$\begin{aligned} \mathcal{Y}_{\text{imperfect}} &= \mathcal{Y}_{\text{perfect}} \; + \; \mathcal{A}_2^T \mathcal{M} \mathcal{R}^{(du)} \mathcal{M} \mathcal{A}_2 \; + \; \mathcal{H} \mathcal{R}_v^{(w)} \mathcal{H}^* \; + \; \mathcal{R}_v^{(\psi)} \\ &\triangleq \mathcal{Y}_{\text{perfect}} \; + \; \Delta \mathcal{Y} \end{aligned} \tag{455}$$

Expression (455) reflects the influence of the noises $\{R_v^{(w)}, \; R_v^{(\psi)}, \; \sigma_{v,\ell k}^2\}$. Substituting the definition (451) into (449), and taking the limit as $i \to \infty$, we obtain from the latter expression that:

$$\boxed{\lim_{i \to \infty} \; \mathbb{E} \, \|\widetilde{\boldsymbol{w}}_i\|^2_{(I-\mathcal{F})\sigma} \; = \; \left[\text{vec}\left(\mathcal{Y}_{\text{imperfect}}^T\right)\right]^T \sigma} \tag{456}$$

which has the same form as (284); therefore, we can proceed analogously to obtain:

$$\boxed{\text{MSD}_{\text{imperfect}}^{\text{network}} \; = \; \frac{1}{N} \cdot \left[\text{vec}\left(\mathcal{Y}_{\text{imperfect}}^T\right)\right]^T \cdot (I - \mathcal{F})^{-1} \cdot \text{vec}\left(I_{NM}\right)} \tag{457}$$

and

$$\boxed{\text{EMSE}_{\text{imperfect}}^{\text{network}} \; = \; \frac{1}{N} \cdot \left[\text{vec}\left(\mathcal{Y}_{\text{imperfect}}^T\right)\right]^T \cdot (I - \mathcal{F})^{-1} \cdot \text{vec}\left(\mathcal{R}_u\right)} \tag{458}$$

Using (455), we see that the network MSD and EMSE deteriorate as follows:

$$\text{MSD}_{\text{imperfect}}^{\text{network}} \; = \; \text{MSD}_{\text{perfect}}^{\text{network}} \; + \; \frac{1}{N} \cdot \left[\text{vec}\left(\Delta \mathcal{Y}^T\right)\right]^T \cdot (I - \mathcal{F})^{-1} \cdot \text{vec}\left(I_{NM}\right) \tag{459}$$

$$\text{EMSE}_{\text{imperfect}}^{\text{network}} \; = \; \text{EMSE}_{\text{perfect}}^{\text{network}} \; + \; \frac{1}{N} \cdot \left[\text{vec}\left(\Delta \mathcal{Y}^T\right)\right]^T \cdot (I - \mathcal{F})^{-1} \cdot \text{vec}\left(\mathcal{R}_u\right) \tag{460}$$

9.5 Adaptive Combination Weights

We can repeat the discussion from Secs. 8.2 and 8.3 to devise one adaptive scheme to adjust the combination coefficients in the noisy exchange case. We illustrate the construction by considering the ATC strategy corresponding to $A_1 = I_N, A_2 = A, C = I_N$, so that only weight estimates are exchanged and the update recursions are of the form:

$$\boldsymbol{\psi}_{k,i} \; = \; \boldsymbol{w}_{k,i-1} \; + \; \mu_k \boldsymbol{u}_{k,i}^* [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i} \boldsymbol{w}_{k,i-1}] \tag{461}$$

$$\boldsymbol{w}_{k,i} \; = \; \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \; \boldsymbol{\psi}_{\ell k,i} \tag{462}$$

where from (408):

$$\boldsymbol{\psi}_{\ell k,i} \; = \; \boldsymbol{\psi}_{\ell,i} \; + \; \boldsymbol{v}_{\ell k,i}^{(\psi)} \tag{463}$$

In this case, the network MSD performance (457) becomes

$$\text{MSD}_{\text{atc,C=I,imperfect}}^{\text{network}} \; = \; \frac{1}{N} \sum_{j=0}^{\infty} \; \text{Tr}\left(\mathcal{B}_{\text{atc,C=I}}^j \mathcal{Y}_{\text{atc,imperfect}} \mathcal{B}_{\text{atc,C=I}}^{*j}\right) \tag{464}$$

where, since now $\mathcal{Z} = \mathcal{S}$ and $\mathcal{R}_v^{(w)} = 0$, we have

$$\begin{aligned}
\mathcal{B}_{\text{atc,C=I}} &= \mathcal{A}^T(I - \mathcal{M}\mathcal{R}_u) && (465)\\
\mathcal{Y}_{\text{atc,imperfect}} &= \mathcal{A}^T\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A} \;+\; \mathcal{R}_v^{(\psi)} && (466)\\
R_v^{(\psi)} &= \text{diag}\left\{R_{v,1}^{(\psi)},\; R_{v,2}^{(\psi)},\; \ldots,\; R_{v,N}^{(\psi)}\right\} && (467)\\
R_{v,k}^{(\psi)} &= \sum_{\ell\in\mathcal{N}_k} a_{\ell k}^2\; R_{v,\ell k}^{(\psi)} && (468)\\
\mathcal{R}_u &= \text{diag}\{R_{u,1}, R_{u,2}, \ldots, R_{u,N}\} && (469)\\
\mathcal{S} &= \text{diag}\{\sigma_{v,1}^2 R_{u,1},\; \sigma_{v,2}^2 R_{u,2}, \ldots, \sigma_{v,N}^2 R_{u,N}\} && (470)\\
\mathcal{M} &= \text{diag}\{\mu_1 I_M,\; \mu_2 I_M, \ldots, \mu_N I_M\} && (471)\\
\mathcal{A} &= A \otimes I_M && (472)
\end{aligned}$$

To proceed, as was the case with (389), we consider the following simplified optimization problem:

$$\begin{aligned}
&\min_{A} \quad \text{Tr}(\mathcal{Y}_{\text{atc,imperfect}})\\
&\text{subject to}\;\; A^T\mathbb{1} = \mathbb{1},\;\; a_{\ell k} \geq 0,\;\; a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k
\end{aligned} \tag{473}$$

Using (466), the trace of $\mathcal{Y}_{\text{atc,imperfect}}$ can be expressed in terms of the combination coefficients as follows:

$$\text{Tr}(\mathcal{Y}_{\text{atc,imperfect}}) \;=\; \sum_{k=1}^{N}\sum_{\ell=1}^{N} a_{\ell k}^2 \;\left[\mu_\ell^2\sigma_{v,\ell}^2\;\text{Tr}(R_{u,\ell}) \;+\; \text{Tr}\left(R_{v,\ell k}^{(\psi)}\right)\right] \tag{474}$$

so that problem (473) can be decoupled into N separate optimization problems of the form:

$$\boxed{\begin{aligned}
&\min_{\{a_{\ell k}\}_{\ell=1}^N} \quad \sum_{\ell=1}^{N} a_{\ell k}^2 \left[\mu_\ell^2\sigma_{v,\ell}^2\;\text{Tr}(R_{u,\ell}) \;+\; \text{Tr}\left(R_{v,\ell k}^{(\psi)}\right)\right], \quad k = 1, \ldots, N\\
&\text{subject to}\\
&\qquad a_{\ell k} \geq 0, \qquad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k
\end{aligned}} \tag{475}$$

With each node ℓ, we associate the following nonnegative variance products:

$$\boxed{\gamma_{\ell k}^2 \;\triangleq\; \mu_\ell^2 \cdot \sigma_{v,\ell}^2 \cdot \text{Tr}(R_{u,\ell}) \;+\; \text{Tr}\left(R_{v,\ell k}^{(\psi)}\right), \;\; k \in \mathcal{N}_\ell} \tag{476}$$

This measure now incorporates information about the exchange noise covariances $\{R_{v,\ell k}^{(\psi)}\}$. Then, the solution of (475) is given by:

$$\boxed{a_{\ell k} = \begin{cases} \dfrac{\gamma_{\ell k}^{-2}}{\sum_{m\in\mathcal{N}_k}\gamma_{mk}^{-2}}, & \text{if } \ell \in \mathcal{N}_k\\ 0, & \text{otherwise}\end{cases}} \quad \text{(relative-variance rule)} \tag{477}$$

We continue to refer to this combination rule as the *relative-variance combination rule* [58]; it leads to a left-stochastic matrix A. To evaluate the combination weights (477), the nodes need to know the variance products, $\{\gamma_{mk}^2\}$, of their neighbors. As before, we can motivate one adaptive construction as follows.

We refer to the ATC recursion (461)–(463) and use the data model (208) to write for node ℓ:

$$\boldsymbol{\psi}_{\ell k,i} = \boldsymbol{w}_{\ell,i-1} \;+\; \mu_\ell \boldsymbol{u}_{\ell,i}^* \left[\boldsymbol{u}_{\ell,i}\widetilde{\boldsymbol{w}}_{\ell,i-1} + \boldsymbol{v}_\ell(i)\right] \;+\; \boldsymbol{v}_{\ell k,i}^{(\psi)} \tag{478}$$

so that, in view of our earlier assumptions on the regression data and noise in Secs. 6.1 and 9.1, we obtain in the limit as $i \to \infty$:

$$\lim_{i\to\infty} \mathbb{E}\left\|\boldsymbol{\psi}_{\ell k,i} - \boldsymbol{w}_{\ell,i-1}\right\|^2 \;=\; \mu_\ell^2 \cdot \left(\lim_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}\left(\boldsymbol{u}_{\ell,i}^*\|\boldsymbol{u}_{\ell,i}\|^2\boldsymbol{u}_{\ell,i}\right)}\right) + \mu_\ell^2 \cdot \sigma_{v,\ell}^2 \cdot \mathrm{Tr}(R_{u,\ell}) \;+\; \mathrm{Tr}\left(R_{v,\ell k}^{(\psi)}\right) \tag{479}$$

In a manner similar to what was done before for (396), we can evaluate the limit on the right-hand side by using the corresponding steady-state result (456). We select the vector σ in (456) to satisfy:

$$(I - \mathcal{F})\sigma \;=\; \mathrm{vec}\left[\mathbb{E}\left(\boldsymbol{u}_{\ell,i}^*\|\boldsymbol{u}_{\ell,i}\|^2\boldsymbol{u}_{\ell,i}\right)\right] \tag{480}$$

Then, from (456),

$$\lim_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{w}}_{i-1}\|^2_{\mathbb{E}\left(\boldsymbol{u}_{\ell,i}^*\|\boldsymbol{u}_{\ell,i}\|^2\boldsymbol{u}_{\ell,i}\right)} \;=\; \left[\mathrm{vec}\left(\mathcal{Y}_{\mathrm{atc,imperfect}}^T\right)\right]^T \cdot (I-\mathcal{F})^{-1} \cdot \mathrm{vec}\left[\mathbb{E}\left(\boldsymbol{u}_{\ell,i}^*\|\boldsymbol{u}_{\ell,i}\|^2\boldsymbol{u}_{\ell,i}\right)\right] \tag{481}$$

Now recall from expression (466) that the entries of $\mathcal{Y}_{\mathrm{atc,imperfect}}$ depend on combinations of the squared step-sizes, $\{\mu_m^2,\ m = 1, 2, \ldots, N\}$, and on terms involving $\left\{\mathrm{Tr}\left(R_{v,m}^{(\psi)}\right)\right\}$. This fact implies that the first term on the right-hand side of (479) depends on products of the form $\{\mu_\ell^2\mu_m^2\}$; these fourth-order factors can be ignored in comparison to the second-order factor μ_ℓ^2 for small step-sizes. Moreover, the same first term on the right-hand side of (479) depends on products of the form $\left\{\mu_\ell^2 \mathrm{Tr}\left(R_{v,m}^{(\psi)}\right)\right\}$, which can be ignored in comparison to the last term, $\mathrm{Tr}\left(R_{v,\ell k}^{(\psi)}\right)$, in (479), which does not appear multiplied by a squared step-size. Therefore, we can approximate:

$$\begin{aligned}\lim_{i\to\infty} \mathbb{E}\left\|\boldsymbol{\psi}_{\ell k,i} - \boldsymbol{w}_{\ell,i-1}\right\|^2 \;&\approx\; \mu_\ell^2 \cdot \sigma_{v,\ell}^2 \cdot \mathrm{Tr}(R_{u,\ell}) \;+\; \mathrm{Tr}\left(R_{v,\ell k}^{(\psi)}\right) \\ &=\; \gamma_{\ell k}^2\end{aligned} \tag{482}$$

in terms of the desired variance product, $\gamma_{\ell k}^2$. Using the following instantaneous approximation at node k (where $w_{\ell,i-1}$ is replaced by $w_{k,i-1}$):

$$\mathbb{E}\|\boldsymbol{\psi}_{\ell k,i} - \boldsymbol{w}_{\ell,i-1}\|^2 \;\approx\; \|\psi_{\ell k,i} - w_{k,i-1}\|^2 \tag{483}$$

we can motivate an algorithm that enables node k to estimate the variance products $\gamma_{\ell k}^2$. Thus, let $\widehat{\gamma}_{\ell k}^2(i)$ denote an estimate for $\gamma_{\ell k}^2$ that is computed by node k at time i. Then, one way to evaluate $\widehat{\gamma}_{\ell k}^2(i)$ is through the recursion:

$$\boxed{\widehat{\gamma}_{\ell k}^2(i) = (1-\nu_k)\cdot\widehat{\gamma}_{\ell k}^2(i-1) \;+\; \nu_k \cdot \|\psi_{\ell k,i} - w_{k,i-1}\|^2} \tag{484}$$

where ν_k is a positive coefficient smaller than one. Indeed, it can be verified that

$$\lim_{i\to\infty} \mathbb{E}\widehat{\gamma}_{\ell k}^2(i) \approx \gamma_{\ell k}^2 \tag{485}$$

so that the estimator $\widehat{\gamma}_{\ell k}^2(i)$ converges on average close to the desired variance product $\gamma_{\ell k}^2$. In this way, we can replace the weights (477) by the adaptive construction:

$$\boxed{a_{\ell k}(i) = \begin{cases} \dfrac{\widehat{\gamma}_{\ell k}^{-2}(i)}{\sum_{m\in\mathcal{N}_k}\widehat{\gamma}_{mk}^{-2}(i)}, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases}} \tag{486}$$

Equations (484) and (486) provide one adaptive construction for the combination weights $\{a_{\ell k}\}$.

10 Extensions and Further Considerations

Several extensions and variations of diffusion strategies are possible. Among those variations we mention strategies that endow nodes with temporal processing abilities, in addition to their spatial cooperation abilities. We can also apply diffusion strategies to solve recursive least-squares and state-space estimation problems in a distributed manner. In this section, we highlight select contributions in these and related areas.

10.1 Adaptive Diffusion Strategies with Smoothing Mechanisms

In the ATC and CTA adaptive diffusion strategies (153)–(154), each node in the network shares information locally with its neighbors through a process of spatial cooperation or combination. In this section, we describe briefly an extension that adds a temporal dimension to the processing at the nodes. For example, in the ATC implementation (153), rather than have each node k rely solely on current data, $\{d_\ell(i), u_{\ell,i},\ \ell \in \mathcal{N}_k\}$, and on current weight estimates received from the neighbors, $\{\psi_{\ell,i},\ \ell \in \mathcal{N}_k\}$, node k can be allowed to store and process present and past weight estimates, say, P of them as in $\{\psi_{\ell,j},\ j = i, i-1, \ldots, i-P+1\}$. In this way, previous weight estimates can be smoothed and used more effectively to help enhance the mean-square-deviation performance especially in the presence of noise over the communication links.

To motivate diffusion strategies with smoothing mechanisms, we continue to assume that the random data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ satisfy the modeling assumptions of Sec. 6.1. The global cost (92) continues to be the same but the individual cost functions (93) are now replaced by:

$$J_k(w) \;=\; \sum_{j=0}^{P-1} q_{kj}\; \mathbb{E}\,|\boldsymbol{d}_k(i-j) - \boldsymbol{u}_{k,i-j}\; w|^2 \tag{487}$$

so that

$$J^{\rm glob}(w) = \sum_{k=1}^{N} \left(\sum_{j=0}^{P-1} q_{kj}\; \mathbb{E}\,|\boldsymbol{d}_k(i-j) - \boldsymbol{u}_{k,i-j}\; w|^2 \right) \tag{488}$$

where each coefficient q_{kj} is a non-negative scalar representing the weight that node k assigns to data from time instant $i-j$. The coefficients $\{q_{kj}\}$ are assumed to satisfy the normalization condition:

$$q_{ko} > 0, \qquad \sum_{j=0}^{P-1} q_{kj} \;=\; 1, \quad k = 1, 2, \ldots, N \tag{489}$$

When the random processes $\boldsymbol{d}_k(i)$ and $\boldsymbol{u}_{k,i}$ are jointly wide-sense stationary, as was assumed in Sec. 6.1, the optimal solution w^o that minimizes (488) is still given by the same normal equations (40). We can extend the arguments of Secs. 3 and 4 to (488) and arrive at the following version of a diffusion strategy incorporating temporal processing (or smoothing) of the intermediate weight estimates [73, 74]:

$$\phi_{k,i} \;=\; w_{k,i-1} \;+\; \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k}\; q_{\ell o}\; u_{\ell,i}^*\, [d_\ell(i) - u_{\ell,i} w_{k,i-1}] \qquad \text{(adaptation)} \tag{490}$$

$$\psi_{k,i} \;=\; \sum_{j=0}^{P-1} f_{kj}\; \phi_{k,i-j} \qquad \text{(temporal processing or smoothing)} \tag{491}$$

$$w_{k,i} \;=\; \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\; \psi_{\ell,i} \qquad \text{(spatial processing)} \tag{492}$$

where the nonnegative coefficients $\{c_{\ell k}, a_{\ell k}, f_{kj}, q_{lo}\}$ satisfy:

$$\text{for } k = 1, 2, \ldots, N:$$

$$c_{\ell k} \geq 0, \quad \sum_{k=1}^{N} c_{\ell k} = 1, \quad c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{493}$$

$$a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{494}$$

$$f_{kj} \geq 0, \quad \sum_{j=0}^{P-1} f_{kj} = 1 \tag{495}$$

$$0 < q_{\ell o} \leq 1 \tag{496}$$

Since only the coefficients $\{q_{\ell o}\}$ are needed, we alternatively denote them by the simpler notation $\{q_\ell\}$ in the listing in Table 8. These are simply chosen as nonnegative coefficients:

$$0 < q_\ell \leq 1, \quad \ell = 1, 2, \ldots, N \tag{497}$$

Note that algorithm (490)-(492) involves three steps: (a) an adaptation step (A) represented by (490); (b) a temporal filtering or smoothing step (T) represented by (491), and a spatial cooperation step (S) represented by (492). These steps are illustrated in Fig. 15. We use the letters (A,T,S) to label these steps; and we use the sequence of letters (A,T,S) to designate the order of the steps. According to this convention, algorithm (490)-(492) is referred to as the ATS diffusion strategy since adaptation is followed by temporal processing, which is followed by spatial processing. In total, we can obtain six different combinations of diffusion algorithms by changing the order by which the temporal and spatial combination steps are performed in relation to the adaptation step. The resulting variations are summarized in Table 8. When we use only the most recent weight vector in the temporal filtering step (i.e., set $\psi_{k,i} = \phi_{k,i}$), which corresponds to the case $P = 1$, the algorithms of Table 8 reduce to the ATC and CTA diffusion algorithms (153) and (154). Specifically, the variants TSA, STA, and SAT (where spatial processing S precedes adaptation A) reduce to CTA, while the variants TAS, ATS, and AST (where adaptation A precedes spatial processing S) reduce to ATC.

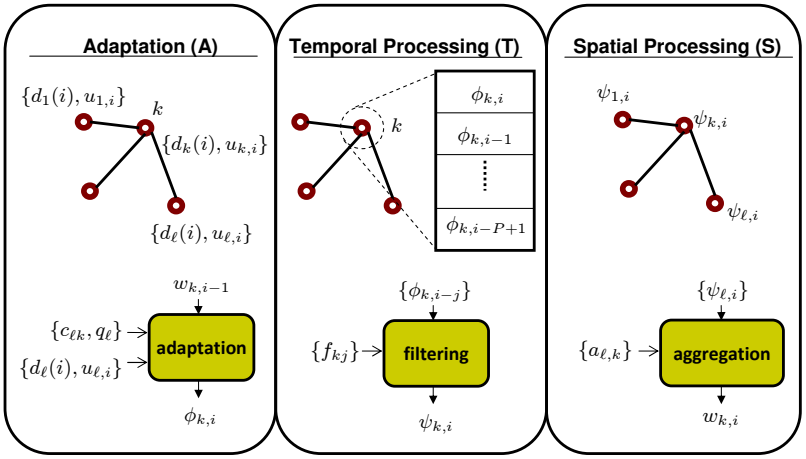

Figure 15: Illustration of the three steps involved in an ATS diffusion strategy: adaptation, followed by temporal processing or smoothing, followed by spatial processing.

The mean-square performance analysis of the smoothed diffusion strategies can be pursued by extending the arguments of Sec. 6. This step is carried out in [73, 74] for doubly stochastic combination matrices A

when the filtering coefficients $\{f_{kj}\}$ do not change with k. For instance, it is shown in [74] that whether temporal processing is performed before or after adaptation, the strategy that performs adaptation before spatial cooperation is always better. Specifically, the six diffusion variants can be divided into two groups with the respective network MSDs satisfying the following relations:

$$\text{Group \#1}: \qquad \text{MSD}_{\text{TSA}}^{\text{network}} \;=\; \text{MSD}_{\text{STA}}^{\text{network}} \;\geq\; \text{MSD}_{\text{TAS}}^{\text{network}} \tag{498}$$

$$\text{Group \#2}: \qquad \text{MSD}_{\text{SAT}}^{\text{network}} \;\geq\; \text{MSD}_{\text{ATS}}^{\text{network}} \;=\; \text{MSD}_{\text{AST}}^{\text{network}} \tag{499}$$

Note that within groups 1 and 2, the order of the A and T operations is the same: in group 1, T precedes A and in group 2, A precedes T. Moreover, within each group, the order of the A and S operations determines performance; the strategy that performs A before S is better. Note further that when $P = 1$, so that temporal processing is not performed, then TAS reduces to ATC and TSA reduces to CTA. This conclusion is consistent with our earlier result (343) that ATC outperforms CTA.

Table 8: Six diffusion strategies with temporal smoothing steps.

TSA diffusion: $\phi_{k,i-1} = \sum_{j=0}^{P-1} f_{kj}\; w_{k,i-j-1}$ $\psi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\; \phi_{\ell,i-1}$ $w_{k,i} = \psi_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} q_\ell\; c_{\ell k}\; u_{\ell,i}^*\,[d_\ell(i) - u_{\ell,i}\;\psi_{k,i-1}]$	**TAS diffusion:** $\phi_{k,i-1} = \sum_{j=0}^{P-1} f_{kj}\; w_{k,i-j-1}$ $\psi_{k,i} = \phi_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} q_\ell\; c_{\ell k}\; u_{\ell,i}^*\,[d_\ell(i) - u_{\ell,i}\phi_{k,i-1}]$ $w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell,k}\psi_{\ell,i}$
STA diffusion: $\phi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\; w_{\ell,i-1}$ $\psi_{k,i-1} = \sum_{j=0}^{P-1} f_{kj}\; \phi_{k,i-j-1}$ $w_{k,i} = \psi_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} q_\ell\; c_{\ell k}\; u_{\ell,i}^*\,[d_\ell(i) - u_{\ell,i}\psi_{k,i-1}]$	**ATS diffusion:** $\phi_{k,i} = w_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} q_\ell\; c_{\ell k}\; u_{\ell,i}^*\,[d_\ell(i) - u_{\ell,i}w_{k,i-1}]$ $\psi_{k,i} = \sum_{j=0}^{P-1} f_{kj}\phi_{k,i-j}$ $w_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\psi_{\ell,i}$
SAT diffusion: $\phi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k} w_{\ell,i-1}$ $\psi_{k,i} = \phi_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} q_\ell\; c_{\ell k}\; u_{\ell,i}^*\,[d_\ell(i) - u_{\ell,i}\phi_{k,i-1}]$ $w_{k,i} = \sum_{j=0}^{P-1} f_{kj}\psi_{k,i-j}$	**AST diffusion:** $\phi_{k,i} = w_{k,i-1} + \mu_k \sum_{\ell\in\mathcal{N}_k} q_\ell\; c_{\ell k}\; u_{\ell,i}^*\,[d_\ell(i) - u_{\ell,i}w_{k,i-1}]$ $\psi_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\phi_{\ell,i}$ $w_{k,i} = \sum_{j=0}^{P-1} f_{kj}\psi_{k,i-j}$

In related work, reference [75] started from the CTA algorithm (159) without information exchange and added a useful projection step to it between the combination step and the adaptation step; i.e., the work considered an algorithm with an STA structure (with spatial combination occurring first, followed by a projection step, and then adaptation). The projection step uses the current weight estimate, $\phi_{k,i}$, at node k and projects it onto hyperslabs defined by the current and past raw data. Specifically, the algorithm

from [75] has the following form:

$$\phi_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, w_{\ell,i-1} \tag{500}$$

$$\psi_{k,i-1} = \mathcal{P}_{k,i}^{'}[\phi_{k,i-1}] \tag{501}$$

$$w_{k,i} = \psi_{k,i-1} - \mu_k \left\{ \psi_{k,i-1} - \sum_{j=0}^{P-1} f_{kj}\cdot \mathcal{P}_{k,i-j}[\phi_{k,i-1}] \right\} \tag{502}$$

where the notation $\psi = \mathcal{P}_{k,i}[\phi]$ refers to the act of projecting the vector ϕ onto the hyperslab $P_{k,i}$ that consists of all $M\times 1$ vectors z satisfying (similarly for the projection $\mathcal{P}_{k,i}^{'}$):

$$P_{k,i} \triangleq \{\ z \text{ such that } |d_k(i) - u_{k,i}z| \le \epsilon_k\ \} \tag{503}$$

$$P_{k,i}^{'} \triangleq \{\ z \text{ such that } |d_k(i) - u_{k,i}z| \le \epsilon_k'\ \} \tag{504}$$

where $\{\epsilon_k, \epsilon_k'\}$ are positive (tolerance) parameters chosen by the designer to satisfy $\epsilon_k' > \epsilon_k$. For generic values $\{d, u, \epsilon\}$, where d is a scalar and u is a row vector, the projection operator is described analytically by the following expression [76]:

$$\mathcal{P}[\phi] = \phi + \begin{cases} \frac{u^*}{\|u\|^2}\,[d-\epsilon-u\phi], & \text{if } d-\epsilon > u\phi \\ 0, & \text{if } |d-u\phi| \le \epsilon \\ \frac{u^*}{\|u\|^2}\,[d+\epsilon-u\phi], & \text{if } d+\epsilon < u\phi \end{cases} \tag{505}$$

The projections that appear in (501)–(502) can be regarded as another example of a temporal processing step. Observe from the middle plot in Fig. 15 that the temporal step that we are considering in the algorithms listed in Table 8 is based on each node k using its current and past weight estimates, such as $\{\phi_{k,i}, \phi_{k,i-1}, \ldots, \phi_{k,i-P+1}\}$, rather than only $\phi_{k,i}$ and current and past *raw* data $\{d_k(i), d_k(i-1), \ldots, d_k(i-P+1), u_{k,i}, u_{k,i-1}, \ldots, u_{k,i-P+1}\}$. For this reason, the temporal processing steps in Table 8 tend to exploit information from across the network more broadly and the resulting mean-square error performance is generally improved relative to (500)–(502).

10.2 Diffusion Recursive Least-Squares

Diffusion strategies can also be applied to recursive least-squares problems to enable distributed solutions of least-squares designs [28,29]; see also [72]. Thus, consider again a set of N nodes that are spatially distributed over some domain. The objective of the network is to collectively estimate some unknown column vector of length M, denoted by w^o, using a least-squares criterion. At every time instant i, each node k collects a scalar measurement, $d_k(i)$, which is assumed to be related to the unknown vector w^o via the linear model:

$$d_k(i) = u_{k,i}w^o + v_k(i) \tag{506}$$

In the above relation, the vector $u_{k,i}$ denotes a row regression vector of length M, and $v_k(i)$ denotes measurement noise. A snapshot of the data in the network at time i can be captured by collecting the measurements and noise samples, $\{d_k(i), v_k(i)\}$, from across all nodes into column vectors y_i and v_i of sizes $N\times 1$ each, and the regressors $\{u_{k,i}\}$ into a matrix H_i of size $N\times M$:

$$y_i = \begin{bmatrix} d_1(i) \\ d_2(i) \\ \vdots \\ d_N(i) \end{bmatrix} (N\times 1), \quad v_i = \begin{bmatrix} v_1(i) \\ v_2(i) \\ \vdots \\ v_N(i) \end{bmatrix} (N\times 1), \quad H_i = \begin{bmatrix} u_{1,i} \\ u_{2,i} \\ \vdots \\ u_{N,i} \end{bmatrix} (N\times M) \tag{507}$$

Likewise, the history of the data across the network up to time i can be collected into vector quantities as follows:

$$\mathcal{Y}_i \;=\; \begin{bmatrix} y_i \\ y_{i-1} \\ \vdots \\ y_0 \end{bmatrix}, \qquad \mathcal{V}_i \;=\; \begin{bmatrix} v_i \\ v_{i-1} \\ \vdots \\ v_0 \end{bmatrix}, \qquad \mathcal{H}_i \;=\; \begin{bmatrix} H_i \\ H_{i-1} \\ \vdots \\ H_0 \end{bmatrix} \tag{508}$$

Then, one way to estimate w^o is by formulating a global least-squares optimization problem of the form:

$$\boxed{\min_{w} \;\; \|w\|^2_{\Pi_i} \;+\; \|\mathcal{Y}_i \;-\; \mathcal{H}_i w\|^2_{\mathcal{W}_i}} \tag{509}$$

where $\Pi_i > 0$ represents a Hermitian regularization matrix and $\mathcal{W}_i \geq 0$ represents a Hermitian weighting matrix. Common choices for Π_i and $\mathcal{W}_i$ are

$$\mathcal{W}_i \;=\; \text{diag}\{I_N,\; \lambda I_N,\; \ldots,\; \lambda^i I_N\} \tag{510}$$

$$\Pi_i \;=\; \lambda^{i+1}\delta^{-1} \tag{511}$$

where $\delta > 0$ is usually a large number and $0 \ll \lambda \leq 1$ is a forgetting factor whose value is generally very close to one. In this case, the global cost function (509) can be written in the equivalent form:

$$\boxed{\min_{w} \;\; \lambda^{i+1}\|w\|^2 \;+\; \sum_{j=0}^{i} \lambda^{i-j} \left(\sum_{k=1}^{N} |d_k(j) - u_{k,j} w|^2 \right)} \tag{512}$$

which is an exponentially weighted least-squares problem. We see that, for every time instant j, the squared errors, $|d_k(j) - u_{k,j}w|^2$, are summed across the network and scaled by the exponential weighting factor λ^{i-j}. The index i denotes current time and the index j denotes a time instant in the past. In this way, data occurring in the remote past are scaled more heavily than data occurring closer to present time. The global solution of (509) is given by [5]:

$$w_i \;=\; \left[\Pi_i + \mathcal{H}_i \mathcal{W}_i \mathcal{H}_i\right]^{-1} \mathcal{H}_i^* \mathcal{W}_i \mathcal{Y}_i \tag{513}$$

and the notation w_i, with a subscript i, is meant to indicate that the estimate w_i is based on all data collected from across the network up to time i. Therefore, the w_i that is computed via (513) amounts to a global construction.

In [28, 29] a diffusion strategy was developed that allows nodes to approach the global solution w_i by relying solely on local interactions. Let $w_{k,i}$ denote a local estimate for w^o that is computed by node k at time i. The diffusion recursive-least-squares (RLS) algorithm takes the following form. For every node k, we start with the initial conditions $w_{k,-1} = 0$ and $P_{k,-1} = \delta I_M$, where $P_{k,-1}$ is an $M \times M$ matrix. Then, for every time instant i, each node k performs an incremental step followed by a diffusion step as follows:

Diffusion RLS.

Step 1 (incremental update)

$$\begin{aligned}
&\psi_{k,i} \leftarrow w_{k,i-1} \\
&P_{k,i} \leftarrow \lambda^{-1} P_{k,i-1} \\
&\text{for every neighboring node } \ell \in \mathcal{N}_k, \text{ update :} \\
&\quad \psi_{k,i} \leftarrow \psi_{k,i} \; + \; \frac{c_{\ell k} P_{k,i} u_{\ell,i}^*}{1 + c_{\ell k} u_{\ell,i} P_{k,i} u_{\ell,i}^*} \left[d_{\ell,i} - u_{\ell,i} \psi_{k,i} \right] \\
&\quad P_{k,i} \leftarrow P_{k,i} - \frac{c_{\ell k} P_{k,i} u_{\ell,i}^* u_{\ell,i} P_{k,i}}{1 + c_{\ell k} u_{\ell,i} P_{k,i} u_{\ell,i}^*} \\
&\text{end}
\end{aligned} \tag{514}$$

Step 2 (diffusion update)

$$w_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi_{\ell,i}$$

where the symbol $\leftarrow$ denotes a sequential assignment, and where the scalars $\{a_{\ell k}, c_{\ell k}\}$ are nonnegative coefficients satisfying:

$$\text{for } k = 1, 2, \ldots, N :$$

$$c_{\ell k} \geq 0, \quad \sum_{k=1}^{N} c_{\ell k} = 1, \quad c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{515}$$

$$a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{516}$$

The above algorithm requires that at every instant i, nodes communicate to their neighbors their measurements $\{d_\ell(i), u_{\ell,i}\}$ for the incremental update, and the intermediate estimates $\{\psi_{\ell,i}\}$ for the diffusion update. During the incremental update, node k cycles through its neighbors and incorporates their data contributions represented by $\{d_\ell(i), u_{\ell,i}\}$ into $\{\psi_{k,i}, P_{k,i}\}$. Every other node in the network is performing similar steps. At the end of the incremental step, neighboring nodes share their intermediate estimates $\{\psi_{\ell,i}\}$ to undergo diffusion. Thus, at the end of both steps, each node k would have updated the quantities $\{w_{k,i-1}, P_{k,i-1}\}$ to $\{w_{k,i}, P_{k,i}\}$. The quantities $P_{k,i}$ are matrices of size $M \times M$ each. Observe that the diffusion RLS implementation (514) does not require the nodes to share their matrices $\{P_{\ell,i}\}$, which would amount to a substantial burden in terms of communications resources since each of these matrices has M^2 entries. Only the quantities $\{d_\ell(i), u_{\ell,i}, \psi_{\ell,i}\}$ are shared. The mean-square performance and convergence of the diffusion RLS strategy are studied in some detail in [29].

The incremental step of the diffusion RLS strategy (514) corresponds to performing a number of $|\mathcal{N}_k|$ successive least-squares updates starting from the initial conditions $\{w_{k,i-1}, P_{k,i-1}\}$ and ending with the values $\{\psi_{k,i}, P_{k,i}\}$ that move on to the diffusion update step. It can be verified from the properties of recursive least-squares solutions [4, 5] that these variables satisfy the following equations at the *end* of the incremental stage (step 1):

$$P_{k,i}^{-1} = \lambda P_{k,i-1}^{-1} + \sum_{\ell \in \mathcal{N}_k} c_{\ell k} u_{\ell,i}^* u_{\ell,i} \tag{517}$$

$$P_{k,i}^{-1} \psi_{k,i} = \lambda P_{k,i-1}^{-1} w_{k,i-1} + \sum_{\ell \in \mathcal{N}_k} c_{\ell k} u_{\ell,i}^* d_\ell(i) \tag{518}$$

Introduce the auxiliary $M \times 1$ variable:

$$q_{k,i} \triangleq P_{k,i}^{-1}\psi_{k,i} \tag{519}$$

Then, the above expressions lead to the following alternative form of the diffusion RLS strategy (514).

Alternative form of diffusion RLS.

$$\begin{aligned} w_{k,i-1} &= \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\psi_{\ell,i-1} \\ P_{k,i}^{-1} &= \lambda P_{k,i-1}^{-1} + \sum_{\ell\in\mathcal{N}_k} c_{\ell k}u_{\ell,i}^{*}u_{\ell,i} \\ q_{k,i} &= \lambda P_{k,i-1}^{-1}w_{k,i-1} + \sum_{\ell\in\mathcal{N}_k} c_{\ell k}u_{\ell,i}^{*}d_{\ell}(i) \\ \psi_{k,i} &= P_{k,i}q_{k,i} \end{aligned} \tag{520}$$

Under some approximations, and for the special choices $A = C$ and $\lambda = 1$, the diffusion RLS strategy (520) can be reduced to a form given in [79] and which is described by the following equations:

$$P_{k,i}^{-1} = \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\left[P_{\ell,i-1}^{-1} + u_{\ell,i}^{*}u_{\ell,i}\right] \tag{521}$$

$$q_{k,i} = \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\left[q_{\ell,i-1} + u_{\ell,i}^{*}d_{\ell}(i)\right] \tag{522}$$

$$\psi_{k,i} = P_{k,i}q_{k,i} \tag{523}$$

Algorithm (521)–(523) is motivated in [79] by using consensus-type arguments. Observe that the algorithm requires the nodes to share the variables $\{d_\ell(i), u_{\ell,i}, q_{\ell,i-1}, P_{\ell,i-1}\}$, which corresponds to more communications overburden than required by diffusion RLS; the latter only requires that nodes share $\{d_\ell(i), u_{\ell,i}, \psi_{\ell,i-1}\}$. In order to illustrate how a special case of diffusion RLS (520) can be related to this scheme, let us set

$$A = C \quad \text{and} \quad \lambda = 1 \tag{524}$$

Then, equations (520) give:

Special form of diffusion RLS when $A = C$ and $\lambda = 1$.

$$\begin{aligned} w_{k,i-1} &= \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\psi_{\ell,i-1} \\ P_{k,i}^{-1} &= P_{k,i-1}^{-1} + \sum_{\ell\in\mathcal{N}_k} c_{\ell k}u_{\ell,i}^{*}u_{\ell,i} \\ q_{k,i} &= P_{k,i-1}^{-1}w_{k,i-1} + \sum_{\ell\in\mathcal{N}_k} c_{\ell k}u_{\ell,i}^{*}d_{\ell}(i) \\ \psi_{k,i} &= P_{k,i}q_{k,i} \end{aligned} \tag{525}$$

Comparing these equations with (521)–(523), we find that algorithm (521)–(523) of [79] would relate to the diffusion RLS algorithm (520) when the following approximations are justified:

$$\begin{aligned}
\sum_{\ell\in\mathcal{N}_k} c_{\ell k} P_{\ell,i-1}^{-1} &\approx P_{k,i-1}^{-1} && (526)\\
\sum_{\ell\in\mathcal{N}_k} c_{\ell k} q_{\ell,i-1} &= \sum_{\ell\in\mathcal{N}_k} c_{\ell k} P_{\ell,i-1}^{-1}\psi_{\ell,i-1} \\
&\approx \sum_{\ell\in\mathcal{N}_k} c_{\ell k} P_{k,i-1}^{-1}\psi_{\ell,i-1} \\
&= P_{k,i-1}^{-1}\sum_{\ell\in\mathcal{N}_k} c_{\ell k}\psi_{\ell,i-1} && (527)\\
&= P_{k,i-1}^{-1} w_{k,i-1} && (528)
\end{aligned}$$

It was indicated in [29] that the diffusion RLS implementation (514) or (520) leads to enhanced performance in comparison to the consensus-based update (521)–(523).

10.3 Diffusion Kalman Filtering

Diffusion strategies can also be applied to the solution of distributed state-space filtering and smoothing problems [30, 31, 33]. Here, we describe briefly the diffusion version of the Kalman filter; other variants and smoothing filters can be found in [33]. We assume that some system of interest is evolving according to linear state-space dynamics, and that every node in the network collects measurements that are linearly related to the unobserved state vector. The objective is for every node to track the state of the system over time based solely on local observations and on neighborhood interactions.

Thus, consider a network consisting of N nodes observing the state vector, $\boldsymbol{x}_i$, of size $n\times 1$ of a linear state-space model. At every time i, every node k collects a measurement vector $\boldsymbol{y}_{k,i}$ of size $p\times 1$, which is related to the state vector as follows:

$$\begin{aligned}
\boldsymbol{x}_{i+1} &= F_i\boldsymbol{x}_i + G_i\boldsymbol{n}_i && (529)\\
\boldsymbol{y}_{k,i} &= H_{k,i}\boldsymbol{x}_i + \boldsymbol{v}_{k,i}, \quad k=1,2,\ldots,N && (530)
\end{aligned}$$

The signals $\boldsymbol{n}_i$ and $\boldsymbol{v}_{k,i}$ denote state and measurement noises of sizes $n\times 1$ and $p\times 1$, respectively, and they are assumed to be zero-mean, uncorrelated and white, with covariance matrices denoted by

$$\mathbb{E}\begin{bmatrix}\boldsymbol{n}_i\\ \boldsymbol{v}_{k,i}\end{bmatrix}\begin{bmatrix}\boldsymbol{n}_j\\ \boldsymbol{v}_{k,j}\end{bmatrix}^* \triangleq \begin{bmatrix}Q_i & 0\\ 0 & R_{k,i}\end{bmatrix}\delta_{ij} \tag{531}$$

The initial state vector, $\boldsymbol{x}_o$, is assumed to be zero-mean with covariance matrix

$$\mathbb{E}\,\boldsymbol{x}_o\boldsymbol{x}_o^* = \Pi_o > 0 \tag{532}$$

and is uncorrelated with $\boldsymbol{n}_i$ and $\boldsymbol{v}_{k,i}$, for all i and k. We further assume that $R_{k,i} > 0$. The parameter matrices $\{F_i, G_i, H_{k,i}, Q_i, R_{k,i}, \Pi_o\}$ are assumed to be known by node k.

Let $\widehat{\boldsymbol{x}}_{k,i|j}$ denote a local estimator for $\boldsymbol{x}_i$ that is computed by node k at time i based solely on local observations and on neighborhood data up to time j. The following diffusion strategy was proposed in [30, 31, 33] to evaluate approximate predicted and filtered versions of these local estimators in a distributed manner for data satisfying model (529)–(532). For every node k, we start with $\widehat{\boldsymbol{x}}_{k,0|-1} = 0$ and $P_{k,0|-1} = \Pi_o$, where $P_{k,0|-1}$ is an $M\times M$ matrix. At every time instant i, every node k performs an incremental step

followed by a diffusion step:

Time and measurement-form of the diffusion Kalman filter.

Step 1 (incremental update)

$$
\begin{aligned}
&\boldsymbol{\psi}_{k,i} \leftarrow \widehat{\boldsymbol{x}}_{k,i|i-1} \\
&P_{k,i} \leftarrow P_{k,i|i-1} \\
&\text{for every neighboring node } \ell \in \mathcal{N}_k, \text{ update :} \\
&\quad R_e \leftarrow R_{\ell,i} + H_{\ell,i} P_{k,i} H_{\ell,i}^* \\
&\quad \boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k,i} + P_{k,i} H_{\ell,i}^* R_e^{-1} \left[\boldsymbol{y}_{\ell,i} - H_{\ell,i} \boldsymbol{\psi}_{k,i}\right] \\
&\quad P_{k,i} \leftarrow P_{k,i} - P_{k,i} H_{\ell,i}^* R_e^{-1} H_{\ell,i} P_{k,i} \\
&\text{end}
\end{aligned}
\tag{533}
$$

Step 2 (diffusion update)

$$
\begin{aligned}
&\widehat{\boldsymbol{x}}_{k,i|i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i} \\
&P_{k,i|i} = P_{k,i} \\
&\widehat{\boldsymbol{x}}_{k,i+1|i} = F_i \widehat{\boldsymbol{x}}_{k,i|i} \\
&P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*.
\end{aligned}
$$

where the symbol $\leftarrow$ denotes a sequential assignment, and where the scalars $\{a_{\ell k}\}$ are nonnegative coefficients satisfying:

$$
\begin{aligned}
&\text{for } k = 1, 2, \ldots, N: \\
&a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k
\end{aligned}
\tag{534}
$$

The above algorithm requires that at every instant i, nodes communicate to their neighbors their measurement matrices $H_{\ell,i}$, the noise covariance matrices $R_{\ell,i}$, and the measurements $\boldsymbol{y}_{\ell,i}$ for the incremental update, and the intermediate estimators $\boldsymbol{\psi}_{\ell,i}$ for the diffusion update. During the incremental update, node k cycles through its neighbors and incorporates their data contributions represented by $\{\boldsymbol{y}_{\ell,i}, H_{\ell,i}, R_{\ell,i}\}$ into $\{\boldsymbol{\psi}_{k,i}, P_{k,i}\}$. Every other node in the network is performing similar steps. At the end of the incremental step, neighboring nodes share their updated intermediate estimators $\{\boldsymbol{\psi}_{\ell,i}\}$ to undergo diffusion. Thus, at the end of both steps, each node k would have updated the quantities $\{\widehat{\boldsymbol{x}}_{k,i|i-1}, P_{k,i|i-1}\}$ to $\{\widehat{\boldsymbol{x}}_{k,i+1|i}, P_{k,i+1|i}\}$. The quantities $P_{k,i|i-1}$ are $n \times n$ matrices. It is important to note that even though the notation $P_{k,i|i}$ and $P_{k,i|i-1}$ has been retained for these variables, as in the standard Kalman filtering notation [5, 77], these matrices do *not* represent any longer the covariances of the state estimation errors, $\tilde{\boldsymbol{x}}_{k,i|i-1} = \boldsymbol{x}_i - \widehat{\boldsymbol{x}}_{k,i|i-1}$, but can be related to them [33].

An alternative representation of the diffusion Kalman filter may be obtained in information form by further assuming that $P_{k,i|i-1} > 0$ for all k and i; a sufficient condition for this fact to hold is to requires the matrices $\{F_i\}$ to be invertible [77]. Thus, consider again data satisfying model (529)–(532). For every node k, we start with $\widehat{\boldsymbol{x}}_{k,0|-1} = 0$ and $P_{k,0|-1}^{-1} = \Pi_o^{-1}$. At every time instant i, every node k performs an incremental step followed by a diffusion step:

Information form of the diffusion Kalman filter.

Step 1 (incremental update)

$$
\begin{aligned}
&S_{k,i} = \sum_{\ell\in\mathcal{N}_k} H_{\ell,i}^* R_{\ell,i}^{-1} H_{\ell,i} \\
&\boldsymbol{q}_{k,i} = \sum_{\ell\in\mathcal{N}_k} H_{\ell,i}^* R_{\ell,i}^{-1} \boldsymbol{y}_{\ell,i} \\
&P_{k,i|i}^{-1} = P_{k,i|i-1}^{-1} + S_{k,i} \\
&\boldsymbol{\psi}_{k,i} = \widehat{\boldsymbol{x}}_{k,i|i-1} + P_{k,i|i}\left[\boldsymbol{q}_{k,i} - S_{k,i}\widehat{\boldsymbol{x}}_{k,i|i-1}\right]
\end{aligned}
\tag{535}
$$

Step 2: (diffusion update)

$$
\begin{aligned}
&\widehat{\boldsymbol{x}}_{k,i|i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{\psi}_{\ell,i} \\
&\widehat{\boldsymbol{x}}_{k,i+1|i} = F_i\widehat{\boldsymbol{x}}_{k,i|i} \\
&P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*
\end{aligned}
$$

The incremental update in (535) is similar to the update used in the distributed Kalman filter derived in [48]. An important difference in the algorithms is in the diffusion step. Reference [48] starts from a continuous-time consensus implementation and discretizes it to arrive at the following update relation:

$$
\widehat{\boldsymbol{x}}_{k,i|i} \;=\; \boldsymbol{\psi}_{k,i} + \epsilon \sum_{\ell\in\mathcal{N}_k} (\boldsymbol{\psi}_{\ell,i} - \boldsymbol{\psi}_{k,i}) \tag{536}
$$

which, in order to facilitate comparison with (535), can be equivalently rewritten as:

$$
\widehat{\boldsymbol{x}}_{k,i|i} \;=\; (1+\epsilon-n_k\epsilon)\cdot\boldsymbol{\psi}_{k,i} \;+\; \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} \epsilon\cdot\boldsymbol{\psi}_{\ell,i} \tag{537}
$$

where n_k denotes the degree of node k (i.e., the size of its neighborhood, $\mathcal{N}_k$). In comparison, the diffusion step in (535) can be written as:

$$
\widehat{\boldsymbol{x}}_{k,i|i} \;=\; a_{kk}\cdot\boldsymbol{\psi}_{k,i} \;+\; \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} a_{\ell k}\cdot\boldsymbol{\psi}_{\ell,i} \tag{538}
$$

Observe that the weights used in (537) are $(1+\epsilon-n_k\epsilon)$ for the node's estimator, $\boldsymbol{\psi}_{k,i}$, and ϵ for all other estimators, $\{\boldsymbol{\psi}_{\ell,i}\}$, arriving from the neighbors of node k. In contrast, the diffusion step (538) employs a convex combination of the estimators $\{\boldsymbol{\psi}_{\ell,i}\}$ with generally different weights $\{a_{\ell k}\}$ for different neighbors; this choice is motivated by the desire to employ combination coefficients that enhance the fusion of information at node k, as suggested by the discussion in App. D of [33]. It was verified in [33] that the diffusion implementation (538) leads to enhanced performance in comparison to the consensus-based update (537). Moreover, the weights $\{a_{\ell k}\}$ in (538) can also be adjusted over time in order to further enhance performance, as discussed in [78]. The mean-square performance and convergence of the diffusion Kalman filtering implementations are studied in some detail in [33], along with other diffusion strategies for smoothing problems including fixed-point and fixed-lag smoothing.

10.4 Diffusion Distributed Optimization

The ATC and CTA steepest-descent diffusion strategies (134) and (142) derived earlier in Sec. 3 provide distributed mechanisms for the solution of global optimization problems of the form:

$$\min_{w} \sum_{k=1}^{N} J_k(w) \tag{539}$$

where the individual costs, $J_k(w)$, were assumed to be quadratic in w, namely,

$$J_k(w) \;=\; \sigma_{d,k}^2 - w^* r_{du,k} - r_{du,k}^* \, w + w^* R_{u,k} \, w \tag{540}$$

for given parameters $\{\sigma_{d,k}^2, r_{du,k}, R_{u,k}\}$. Nevertheless, we remarked in that section that similar diffusion strategies can be applied to more general cases involving individual cost functions, $J_k(w)$, that are not necessarily quadratic in w [1–3]. We restate below, for ease of reference, the general ATC and CTA diffusion strategies (139) and (146) that can be used for the distributed solution of global optimization problems of the form (539) for more general convex functions $J_k(w)$:

$$\text{(ATC strategy)} \quad \boxed{\begin{aligned} \psi_{k,i} &= w_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \; [\nabla_w J_\ell(w_{k,i-1})]^* \\ w_{k,i} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \; \psi_{\ell,i} \end{aligned}} \tag{541}$$

and

$$\text{(CTA strategy)} \quad \boxed{\begin{aligned} \psi_{k,i-1} &= \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \; w_{\ell,i-1} \\ w_{k,i} &= \psi_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \; [\nabla_w J_\ell(\psi_{k,i-1})]^* \end{aligned}} \tag{542}$$

for positive step-sizes $\{\mu_k\}$ and for nonnegative coefficients $\{c_{\ell k}, a_{\ell k}\}$ that satisfy:

$$\begin{aligned} &\text{for } k = 1, 2, \ldots, N: \\ c_{\ell k} \geq 0, \quad &\sum_{k=1}^{N} c_{\ell k} = 1, \quad c_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \\ a_{\ell k} \geq 0, \quad &\sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{aligned} \tag{543}$$

That is, the matrix $A = [a_{\ell k}]$ is left-stochastic while the matrix $C = [c_{\ell k}]$ is right-stochastic:

$$C\mathbb{1} = \mathbb{1}, \qquad A^T \mathbb{1} = \mathbb{1} \tag{544}$$

We can again regard the above ATC and CTA strategies as special cases of the following general diffusion scheme:

$$\phi_{k,i-1} \;=\; \sum_{\ell \in \mathcal{N}_k} a_{1,\ell k} \; w_{\ell,i-1} \tag{545}$$

$$\psi_{k,i} \;=\; \phi_{k,i-1} - \mu_k \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \, [\nabla_w J_\ell(\phi_{k,i-1})]^* \tag{546}$$

$$w_{k,i} \;=\; \sum_{\ell \in \mathcal{N}_k} a_{2,\ell k} \; \psi_{\ell,i} \tag{547}$$

where the coefficients $\{a_{1,\ell k}, a_{2,\ell k}, c_{\ell k}\}$ are nonnegative coefficients corresponding to the (l,k)-th entries of combination matrices $\{A_1, A_2, C\}$ that satisfy:

$$A_1^T \mathbb{1} = \mathbb{1}, \quad A_2^T \mathbb{1} = \mathbb{1}, \quad C\mathbb{1} = \mathbb{1} \tag{548}$$

The convergence behavior of these diffusion strategies can be examined under both conditions of noiseless updates (when the gradient vectors are available) and noisy updates (when the gradient vectors are subject to gradient noise). The following properties can be proven for the diffusion strategies (545)–(547) [1–3]. The statements that follow assume, for convenience of presentation, that all data are *real-valued*; the conditions would need to be adjusted for complex-valued data.

Noiseless Updates
Let

$$J^{\rm glob}(w) \;=\; \sum_{k=1}^{N} J_k(w) \tag{549}$$

denote the global cost function that we wish to minimize. Assume $J^{\rm glob}(w)$ is strictly convex so that its minimizer w^o is unique. Assume further that each individual cost function $J_k(w)$ is convex and has a minimizer at the *same* w^o. This case is common in practice; situations abound where nodes in a network need to work cooperatively to attain a common objective (such as tracking a target, locating the source of chemical leak, estimating a physical model, or identifying a statistical distribution). The case where the $\{J_k(w)\}$ have different individual minimizers is studied in [1,3], where it is shown that the same diffusion strategies of this section are still applicable and nodes would converge instead to a Pareto-optimal solution.

Theorem* 10.1.** *(Convergence to Optimal Solution: Noise-Free Case***) Consider the problem of minimizing the strictly convex global cost (549), with the individual cost functions $\{J_k(w)\}$ assumed to be convex with each having a minimizer at the same w^o. Assume that all data are real-valued and suppose the Hessian matrices of the individual costs are bounded from below and from above as follows:*

$$\lambda_{\ell,\min} I_M \;\leq\; \nabla_w^2 J_\ell(w) \;\leq\; \lambda_{\ell,\max} I_M, \quad \ell = 1, 2, \ldots, N \tag{550}$$

for some positive constants $\{\lambda_{\ell,\min}, \lambda_{\ell,\max}\}$. Let

$$\sigma_{k,\min} \;\triangleq\; \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \lambda_{\ell,\min}, \qquad \sigma_{k,\max} \;\triangleq\; \sum_{\ell \in \mathcal{N}_k} c_{\ell k} \lambda_{\ell,\max} \tag{551}$$

Assume further that $\sigma_{k,\min} > 0$ and that the positive step-sizes are chosen such that:

$$\mu_k \leq \frac{2}{\sigma_{k,\max}}, \qquad k = 1, \ldots, N \tag{552}$$

Then, it holds that $w_{k,i} \to w^o$ as $i \to \infty$. That is, the weight estimates generated by (545)–(547) at all nodes will tend towards the desired global minimizer.

□

We note that in works on distributed sub-gradient methods (e.g., [40, 80]), the norms of the sub-gradients are usually required to be uniformly bounded. Such a requirement is restrictive in the unconstrained optimization of differentiable functions. Condition (550) is more relaxed since it allows the gradient vector $\nabla_w J_\ell(w)$ to have *unbounded* norm. This extension is important because requiring bounded gradient norms, as opposed to bounded Hessian matrices, would exclude the possibility of using quadratic costs for the $J_\ell(w)$ (since the gradient vectors would then be unbounded). And, as we saw in the body of the chapter, quadratic costs play a critical role in adaptation and learning over networks.

Updates with Gradient Noise

It is often the case that we do not have access to the exact gradient vectors to use in (546), but to noisy versions of them, say,

$$\widehat{\nabla_w J_\ell(\boldsymbol{\phi}_{k,i-1})} \;\triangleq\; \nabla_w J_\ell(\boldsymbol{\phi}_{k,i-1}) \;+\; \boldsymbol{v}_\ell(\widetilde{\boldsymbol{\phi}}_{k,i-1}) \tag{553}$$

where the random vector variable $\boldsymbol{v}_\ell(\cdot)$ refers to gradient noise; its value is generally dependent on the weight-error vector realization,

$$\widetilde{\boldsymbol{\phi}}_{k,i-1} \;\triangleq\; w^o - \boldsymbol{\phi}_{k,i-1} \tag{554}$$

at which the gradient vector is being evaluated. In the presence of gradient noise, the weight estimates at the various nodes become random quantities and we denote them by the boldface notation $\{\boldsymbol{w}_{k,i}\}$. We assume that, conditioned on the past history of the weight estimators at all nodes, namely,

$$\mathcal{F}_{i-1} \;\triangleq\; \{\boldsymbol{w}_{m,j},\; m=1,2,\ldots,N,\; j<i\} \tag{555}$$

the gradient noise has zero mean and its variance is upper bounded as follows:

$$\mathbb{E}\left\{\boldsymbol{v}_\ell(\widetilde{\boldsymbol{\phi}}_{k,i-1}) \mid \mathcal{F}_{i-1}\right\} \;=\; 0 \tag{556}$$

$$\mathbb{E}\left\{\|\boldsymbol{v}_\ell(\widetilde{\boldsymbol{\phi}}_{k,i-1})\|^2 \mid \mathcal{F}_{i-1}\right\} \;\leq\; \alpha\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^2 \;+\; \sigma_v^2 \tag{557}$$

for some $\alpha > 0$ and $\sigma_v^2 \geq 0$. Condition (557) allows the variance of the gradient noise to be time-variant, so long as it does not grow faster than $\mathbb{E}\,\|\widetilde{\boldsymbol{\phi}}_{k,i-1}\|^2$. This condition on the noise is more general than the "uniform-bounded assumption" that appears in [40], which required instead:

$$\mathbb{E}\left\{\|\boldsymbol{v}_\ell(\widetilde{\boldsymbol{\phi}}_{k,i-1})\|^2\right\} \leq \sigma_v^2, \qquad \mathbb{E}\left\{\|\boldsymbol{v}_\ell(\widetilde{\boldsymbol{\phi}}_{k,i-1})\|^2 \mid \mathcal{F}_{i-1}\right\} \leq \sigma_v^2 \tag{558}$$

These two requirements are special cases of (557) for $\alpha = 0$. Furthermore, condition (557) is similar to condition (4.3) in [81], which requires the noise variance to satisfy:

$$\mathbb{E}\left\{\|\boldsymbol{v}_\ell(\widetilde{\boldsymbol{\phi}}_{k,i-1})\|^2 \mid \mathcal{F}_{i-1}\right\} \leq \alpha\left[\|\nabla_w J_\ell(\boldsymbol{\phi}_{k,i-1})\|^2 \;+\; 1\right] \tag{559}$$

This requirement can be verified to be a combination of the "relative random noise" and the "absolute random noise" conditions defined in [22] — see [2].

Now, introduce the column vector:

$$\boldsymbol{z}_i \;\triangleq\; \sum_{\ell=1}^{N} \mathrm{col}\left\{c_{\ell 1}\boldsymbol{v}_\ell(w^o),\; c_{\ell 2}\boldsymbol{v}_\ell(w^o),\; \ldots,\; c_{\ell N}\boldsymbol{v}_\ell(w^o)\right\} \tag{560}$$

and let

$$\mathcal{Z} \;\triangleq\; \mathbb{E}\,\boldsymbol{z}_i\boldsymbol{z}_i^* \tag{561}$$

Let further

$$\widetilde{\boldsymbol{w}}_i \;\triangleq\; \mathrm{col}\left\{\widetilde{\boldsymbol{w}}_{i,1},\; \widetilde{\boldsymbol{w}}_{i,2},\ldots,\widetilde{\boldsymbol{w}}_{i,N}\right\} \tag{562}$$

where

$$\widetilde{\boldsymbol{w}}_{k,i} \;\triangleq\; w^o - \boldsymbol{w}_{k,i} \tag{563}$$

Then, the following result can be established [2]; it characterizes the network mean-square deviation in steady-state, which is defined as

$$\mathrm{MSD}^{\mathrm{network}} \;\triangleq\; \lim_{i\to\infty}\left(\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_{k,i}\|^2\right) \tag{564}$$

Theorem* 10.2.** *(Mean-Square Stability: Noisy Case***) Consider the problem of minimizing the strictly convex global cost (549), with the individual cost functions $\{J_k(w)\}$ assumed to be convex with each having a minimizer at the same w^o. Assume all data are real-valued and suppose the Hessian matrices of the individual costs are bounded from below and from above as stated in (550). Assume further that the diffusion strategy (545)–(547) employs noisy gradient vectors, where the noise terms are zero mean and satisfy conditions (557) and (561). We select the positive step-sizes to be sufficiently small and to satisfy:*

$$\mu_k \;<\; \min\left\{\frac{2\sigma_{k,\max}}{\sigma_{k,\max}^2 \;+\; \alpha\|C\|_1^2}, \frac{2\sigma_{k,\min}}{\sigma_{k,\min}^2 \;+\; \alpha\|C\|_1^2} \;\right\} \tag{565}$$

for $k = 1, 2, \ldots, N$. Then, the diffusion strategy (545)–(547) is mean-square stable and the mean-square-deviation of the network is given by:

$$\mathrm{MSD}^{\mathrm{network}} \;\approx\; \frac{1}{N}\left[\mathrm{vec}\left(\mathcal{A}_2^T\mathcal{M}\mathcal{Z}^T\mathcal{M}\mathcal{A}_2\right)\right]^T\cdot(I-\mathcal{F})^{-1}\cdot\mathrm{vec}(I_{NM}) \tag{566}$$

where

$$\begin{aligned}
\mathcal{A}_2 &= A_2\otimes I_M && (567)\\
\mathcal{M} &= \mathrm{diag}\{\mu_1 I_M,\; \mu_2 I_M,\; \ldots, \mu_N I_M\} && (568)\\
\mathcal{F} &\approx \mathcal{B}^T\otimes\mathcal{B}^* && (569)\\
\mathcal{B} &= \mathcal{A}_2^T\,(I-\mathcal{M}\mathcal{R})\,\mathcal{A}_1^T && (570)\\
\mathcal{R} &= \sum_{\ell=1}^{N}\mathrm{diag}\left\{c_{\ell 1}\nabla_w^2 J_\ell(w^o),\; c_{\ell 2}\nabla_w^2 J_\ell(w^o),\; \ldots,\; c_{\ell N}\nabla_w^2 J_\ell(w^o)\right\} && (571)
\end{aligned}$$

□

Acknowledgements

The development of the theory and applications of diffusion adaptation over networks has benefited greatly from the insights and contributions of several UCLA PhD students, and several visiting graduate students to the UCLA Adaptive Systems Laboratory (http://www.ee.ucla.edu/asl). The assistance and contributions of all students are hereby gratefully acknowledged, including Cassio G. Lopes, Federico S. Cattivelli, Sheng-Yuan Tu, Jianshu Chen, Xiaochuan Zhao, Zaid Towfic, Chung-Kai Yu, Noriyuki Takahashi, Jae-Woo Lee, Alexander Bertrand, and Paolo Di Lorenzo. The author is also particularly thankful to S.-Y. Tu, J. Chen, X. Zhao, Z. Towfic, and C.-K. Yu for their assistance in reviewing an earlier draft of this article.

A Appendix: Properties of Kronecker Products

For ease of reference, we collect in this appendix some useful properties of Kronecker products. All matrices are assumed to be of compatible dimensions; all inverses are assumed to exist whenever necessary. Let $E = [e_{ij}]_{i,j=1}^{n}$ and $B = [b_{ij}]_{i,j=1}^{m}$ be $n \times n$ and $m \times m$ matrices, respectively. Their Kronecker product is denoted by $E \otimes B$ and is defined as the $nm \times nm$ matrix whose entries are given by [20]:

$$E \otimes B = \begin{bmatrix} e_{11}B & e_{12}B & \dots & e_{1n}B \\ e_{21}B & e_{22}B & \dots & e_{2n}B \\ \vdots & & \vdots & \\ e_{n1}B & e_{n2}B & \dots & e_{nn}B \end{bmatrix} \tag{572}$$

In other words, each entry of E is replaced by a scaled multiple of B. Let $\{\lambda_i(E), i = 1, \dots, n\}$ and $\{\lambda_j(B), j = 1, \dots, m\}$ denote the eigenvalues of E and B, respectively. Then, the eigenvalues of $E \otimes B$ will consist of all nm product combinations $\{\lambda_i(E)\lambda_j(B)\}$. Table 9 lists some well-known properties of Kronecker products.

Table 9: Properties of Kronecker products.

$$(E + B) \otimes C = (E \otimes C) \; + \; (B \otimes C)$$
$$(E \otimes B)(C \otimes D) = (EC \otimes BD)$$

$$(E \otimes B)^T = E^T \otimes B^T$$
$$(E \otimes B)^* = E^* \otimes B^*$$
$$(E \otimes B)^{-1} = E^{-1} \otimes B^{-1}$$
$$(E \otimes B)^{\ell} = E^{\ell} \otimes B^{\ell}$$

$$\{\lambda(E \otimes B)\} = \{\lambda_i(E)\lambda_j(B)\}_{i=1,j=1}^{n,m}$$
$$\det(E \otimes B) = (\det E)^m (\det B)^n$$

$$\mathrm{Tr}(E \otimes B) = \mathrm{Tr}(E)\mathrm{Tr}(B)$$
$$\mathrm{Tr}(EB) = \big[\mathrm{vec}(B^T)\big]^T \mathrm{vec}(E)$$
$$\mathrm{vec}(ECB) = (B^T \otimes E)\mathrm{vec}(C)$$

B Appendix: Graph Laplacian and Network Connectivity

Consider a network consisting of N nodes and L edges connecting the nodes to each other. In the constructions below, we only need to consider the edges that connect distinct nodes to each other; these edges do not contain any self-loops that may exist in the graph and which connect nodes to themselves directly. In other words, when we refer to the L edges of a graph, we are excluding self-loops from this set; but we are still allowing loops of at least length 2 (i.e., loops generated by paths covering at least 2 edges).

The neighborhood of any node k is denoted by $\mathcal{N}_k$ and it consists of all nodes that node k can share information with; these are the nodes that are connected to k through edges, in addition to node k itself. The degree of node k, which we denote by n_k, is defined as the positive integer that is equal to the size of its neighborhood:

$$n_k \; \triangleq \; |\mathcal{N}_k| \tag{573}$$

Since $k \in \mathcal{N}_k$, we always have $n_k \geq 1$. We further associate with the network an $N \times N$ Laplacian matrix,

denoted by $\mathcal{L}$. The matrix $\mathcal{L}$ is symmetric and its entries are defined as follows [64–66]:

$$[\mathcal{L}]_{k\ell} = \begin{cases} n_k - 1, & \text{if } k = \ell \\ -1, & \text{if } k \neq \ell \text{ and nodes } k \text{ and } \ell \text{ are neighbors} \\ 0, & \text{otherwise} \end{cases} \tag{574}$$

Note that the term $n_k - 1$ measures the number of edges that are incident on node k, and the locations of the $-1's$ on row k indicate the nodes that are connected to node k. We also associate with the graph an $N \times L$ incidence matrix, denoted by $\mathcal{I}$. The entries of $\mathcal{I}$ are defined as follows. Every column of $\mathcal{I}$ represents one edge in the graph. Each edge connects two nodes and its column will display two nonzero entries at the rows corresponding to these nodes: one entry will be $+1$ and the other entry will be -1. For directed graphs, the choice of which entry is positive or negative can be used to identify the nodes from which edges emanate (source nodes) and the nodes at which edges arrive (sink nodes). Since we are dealing with undirected graphs, we shall simply assign positive values to lower indexed nodes and negative values to higher indexed nodes:

$$[\mathcal{I}]_{ke} = \begin{cases} +1, & \text{if node } k \text{ is the lower-indexed node connected to edge } e \\ -1, & \text{if node } k \text{ is the higher-indexed node connected to edge } e \\ 0, & \text{otherwise} \end{cases} \tag{575}$$

Figure 16 shows the example of a network with $N = 6$ nodes and $L = 8$ edges. Its Laplacian and incidence matrices are also shown and these have sizes 6×6 and 6×8, respectively. Consider, for example, column 6 in the incidence matrix. This column corresponds to edge 6, which links nodes 3 and 5. Therefore, at location $\mathcal{I}_{36}$ we have a $+1$ and at location $\mathcal{I}_{56}$ we have -1. The other columns of $\mathcal{I}$ are constructed in a similar manner.

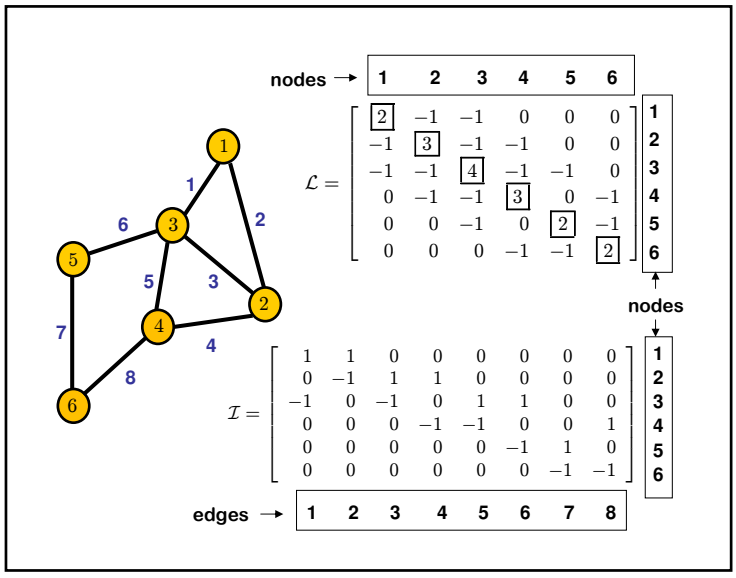

Figure 16: A network with $N = 6$ nodes and $L = 8$ edges. The nodes are marked 1 through 6 and the edges are marked 1 through 8. The corresponding Laplacian and incidence matrices $\mathcal{L}$ and $\mathcal{I}$ are 6×6 and 6×8.

Observe that the Laplacian and incidence matrices of a graph are related as follows:

$$\mathcal{L} = \mathcal{I}\,\mathcal{I}^T \tag{576}$$

The Laplacian matrix conveys useful information about the topology of the graph. The following is a classical result from graph theory [64–67].

***Lemma* B.1. (Laplacian and Network Connectivity)** *Let*

$$\theta_1 \geq \theta_2 \geq \ldots \geq \theta_N \tag{577}$$

denote the ordered eigenvalues of $\mathcal{L}$. Then the following properties hold:

(a) $\mathcal{L}$ is symmetric nonnegative-definite so that $\theta_i \geq 0$.

(b) The rows of $\mathcal{L}$ add up to zero so that $\mathcal{L}\mathbb{1} = 0$. This means that $\mathbb{1}$ is a right eigenvector of $\mathcal{L}$ corresponding to the eigenvalue zero.

(c) The smallest eigenvalue is always zero, $\theta_N = 0$. The second smallest eigenvalue, θ_{N-1}, is called the algebraic connectivity of the graph.

(d) The number of times that zero is an eigenvalue of $\mathcal{L}$ (i.e., its multiplicity) is equal to the number of connected subgraphs.

(e) The algebraic connectivity of a connected graph is nonzero, i.e., $\theta_{N-1} \neq 0$. In other words, a graph is connected if, and only if, its algebraic connectivity is nonzero.

Proof. Property (a) follows from the identity $\mathcal{L} = \mathcal{I}\,\mathcal{I}^T$. Property (b) follows from the definition of $\mathcal{L}$. Note that for each row of $\mathcal{L}$, the entries on the row add up to zero. Property (c) follows from properties (a) and (b) since $\mathcal{L}\mathbb{1} = 0$ implies that zero is an eigenvalue of $\mathcal{L}$. For part (d), assume the network consists of two separate connected subgraphs. Then, the Laplacian matrix would have a block diagonal structure, say, of the form $\mathcal{L} = \text{diag}\{\mathcal{L}_1, \mathcal{L}_2\}$, where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the Laplacian matrices of the smaller subgraphs. The smallest eigenvalue of each of these Laplacian matrices would in turn be zero and unique by property (e). More generally, if the graph consists of m connected subgraphs, then the multiplicity of zero as an eigenvalue of $\mathcal{L}$ must be m. To establish property (e), first observe that if the algebraic connectivity is nonzero then it is obvious that the graph must be connected. Otherwise, if the graph were disconnected, then its Laplacian matrix would be block diagonal and the algebraic multiplicity of zero as an eigenvalue of $\mathcal{L}$ would be larger than one so that θ_{N-1} would be zero, which is a contradiction. For the converse statement, assume the graph is connected and let x denote an arbitrary eigenvector of $\mathcal{L}$ corresponding to the eigenvalue at zero, i.e., $\mathcal{L}x = 0$. We already know that $\mathcal{L}\mathbb{1} = \mathbb{1}$ from property (b). Let us verify that x must be proportional to the vector $\mathbb{1}$ so that the algebraic multiplicity of the eigenvalue at zero is one. Thus note that $x^T\mathcal{L}x = 0$. If we denote the individual entries of x by x_k, then this identity implies that for each node k:

$$\sum_{\ell \in \mathcal{N}_k} (x_k - x_\ell)^2 \;=\; 0$$

It follows that the entries of x within each neighborhood have equal values. But since the graph is connected, we conclude that all entries of x must be equal. It follows that the eigenvector x is proportional to the vector $\mathbb{1}$, as desired. □

C Appendix: Stochastic Matrices

Consider $N \times N$ matrices A with nonnegative entries, $\{a_{\ell k} \geq 0\}$. The matrix $A = [a_{\ell k}]$ is said to be right-stochastic if it satisfies

$$A\mathbb{1} = \mathbb{1} \qquad \text{(right-stochastic)} \tag{578}$$

in which case each row of A adds up to one. The matrix A is said to be left-stochastic if it satisfies

$$A^T\mathbb{1} = \mathbb{1} \qquad \text{(left-stochastic)} \tag{579}$$

in which case each column of A adds up to one. And the matrix is said to be doubly stochastic if both conditions hold so that both its columns and rows add up to one:

$$A\mathbb{1} = \mathbb{1}, \quad A^T\mathbb{1} = \mathbb{1} \qquad \text{(doubly-stochastic)} \tag{580}$$

Stochastic matrices arise frequently in the study of adaptation over networks. This appendix lists some of their properties.

***Lemma* C.1.** *(**Spectral Norm of Stochastic Matrices**) Let A be an $N \times N$ right or left or doubly stochastic matrix. Then, $\rho(A) = 1$ and, therefore, all eigenvalues of A lie inside the unit disc, i.e., $|\lambda(A)| \leq 1$.*

Proof. We prove the result for right stochastic matrices; a similar argument applies to left or doubly stochastic matrices. Let A be a right-stochastic matrix. Then, $A\mathbb{1} = \mathbb{1}$, so that $\lambda = 1$ is one of the eigenvalues of A. Moreover, for any matrix A, it holds that $\rho(A) \leq \|A\|_\infty$, where $\|\cdot\|_\infty$ denotes the maximum absolute row sum of its matrix argument. But since all rows of A add up to one, we have $\|A\|_\infty = 1$. Therefore, $\rho(A) \leq 1$. And since we already know that A has an eigenvalue at $\lambda = 1$, we conclude that $\rho(A) = 1$.

□

The above result asserts that the spectral radius of a stochastic matrix is unity and that A has an eigenvalue at $\lambda = 1$. The result, however, does not rule out the possibility of multiple eigenvalues at $\lambda = 1$, or even other eigenvalues with magnitude equal to one. Assume, in addition, that the stochastic matrix A is *regular*. This means that there exists an integer power j_o such that all entries of A^{j_o} are *strictly* positive, i.e.,

$$\text{for all } (\ell, k), \text{ it holds that } \left[A^{j_o}\right]_{\ell k} > 0, \text{ for some } j_o > 0 \tag{581}$$

Then a result in matrix theory known as the Perron-Frobenius Theorem [20] leads to the following stronger characterization of the eigen-structure of A.

***Lemma* C.2.** *(**Spectral Norm of Regular Stochastic Matrices**) Let A be an $N \times N$ right stochastic and regular matrix. Then:*

(a) $\rho(A) = 1$.

(b) All other eigenvalues of A are strictly inside the unit circle (and, hence, have magnitude strictly less than one).

(c) The eigenvalue at $\lambda = 1$ is simple, i.e., it has multiplicity one. Moreover, with proper sign scaling, all entries of the corresponding eigenvector are positive. For a right-stochastic A, this eigenvector is the vector $\mathbb{1}$ since $A\mathbb{1} = \mathbb{1}$.

(d) All other eigenvectors associated with the other eigenvalues will have at least one negative or complex entry.

Proof. Part (a) follows from Lemma C.1. Parts (b)-(d) follow from the Perron-Frobenius Theorem when A is regular [20].

□

***Lemma* C.3.** *(**Useful Properties of Doubly Stochastic Matrices**) Let A be an $N \times N$ doubly stochastic matrix. Then the following properties hold:*

(a) $\rho(A) = 1$.

(b) AA^T and A^TA are doubly stochastic as well.

(c) $\rho(AA^T) = \rho(A^TA) = 1$.

(d) The eigenvalues of AA^T or A^TA are real and lie inside the interval $[0, 1]$.

(e) $I - AA^T \geq 0$ *and* $I - A^TA \geq 0$.

(f) $\mathrm{Tr}(A^THA) \leq \mathrm{Tr}(H)$, *for any* $N \times N$ *nonnegative-definite Hermitian matrix* H.

Proof. Part (a) follows from Lemma C.1. For part (b), note that AA^T is symmetric and $AA^T\mathbb{1} = A\mathbb{1} = \mathbb{1}$. Therefore, AA^T is doubly stochastic. Likewise for A^TA. Part (c) follows from part (a) since AA^T and A^TA are themselves doubly stochastic matrices. For part (d), note that AA^T is symmetric and nonnegative-definite. Therefore, its eigenvalues are real and nonnegative. But since $\rho(AA^T) = 1$, we must have $\lambda(AA^T) \in [0, 1]$. Likewise for the matrix A^TA. Part (e) follows from part (d). For part (f), since $AA^T \geq 0$ and its eigenvalues lie within $[0, 1]$, the matrix AA^T admits an eigen-decomposition of the form:

$$AA^T = U\Lambda U^T$$

where U is orthogonal (i.e., $U^{-1} = U^T$) and Λ is diagonal with entries in the range $[0, 1]$. It then follows that

$$\begin{aligned}
\mathrm{Tr}(A^THA) &= \mathrm{Tr}(AA^TH) \\
&= \mathrm{Tr}(U\Lambda U^TH) \\
&= \mathrm{Tr}(\Lambda U^THU) \\
&\overset{(*)}{\leq} \mathrm{Tr}(U^THU) \\
&= \mathrm{Tr}(UU^TH) \\
&= \mathrm{Tr}(H)
\end{aligned}$$

where step $(*)$ is because $U^THU = U^{-1}HU$ and, by similarity, the matrix $U^{-1}HU$ has the same eigenvalues as H. Therefore, $U^THU \geq 0$. This means that the diagonal entries of U^THU are nonnegative. Multiplying U^THU by Λ ends up scaling the nonnegative diagonal entries to smaller values so that $(*)$ is justified.

□

D Appendix: Block Maximum Norm

Let $x = \mathrm{col}\{x_1, x_2, \ldots, x_N\}$ denote an $N \times 1$ block column vector whose individual entries are of size $M \times 1$ each. Following [52, 54, 55], the block maximum norm of x is denoted by $\|x\|_{b,\infty}$ and is defined as

$$\boxed{\|x\|_{b,\infty} \triangleq \max_{1 \leq k \leq N} \|x_k\|} \tag{582}$$

where $\|\cdot\|$ denotes the Euclidean norm of its vector argument. Correspondingly, the induced block maximum norm of an arbitrary $N \times N$ block matrix $\mathcal{A}$, whose individual block entries are of size $M \times M$ each, is defined as

$$\boxed{\|\mathcal{A}\|_{b,\infty} \triangleq \max_{x \neq 0} \frac{\|\mathcal{A}x\|_{b,\infty}}{\|x\|_{b,\infty}}} \tag{583}$$

The block maximum norm inherits the unitary invariance property of the Euclidean norm, as the following result indicates [54].

***Lemma* D.1. (Unitary Invariance)** *Let* $\mathcal{U} = \mathrm{diag}\{U_1, U_2, \ldots, U_N\}$ *be an* $N \times N$ *block diagonal matrix with* $M \times M$ *unitary blocks* $\{U_k\}$. *Then, the following properties hold:*

(a) $\|\mathcal{U}x\|_{b,\infty} = \|x\|_{b,\infty}$

(b) $\|\mathcal{U}\mathcal{A}\mathcal{U}^*\|_{b,\infty} = \|\mathcal{A}\|_{b,\infty}$

for all block vectors x *and block matrices* $\mathcal{A}$ *of appropriate dimensions.* □

The next result provides useful bounds for the block maximum norm of a block matrix.

***Lemma* D.2. (Useful Bounds)** *Let $\mathcal{A}$ be an arbitrary $N \times N$ block matrix with blocks $A_{\ell k}$ of size $M \times M$ each. Then, the following results hold:*

(a) The norms of $\mathcal{A}$ and its complex conjugate are related as follows:

$$\|\mathcal{A}^*\|_{b,\infty} \;\leq\; N \cdot \|\mathcal{A}\|_{b,\infty} \tag{584}$$

(b) The norm of $\mathcal{A}$ is bounded as follows:

$$\max_{1\leq \ell,k\leq N} \|A_{\ell k}\| \;\leq\; \|\mathcal{A}\|_{b,\infty} \;\leq\; N \cdot \left(\max_{1\leq \ell,k\leq N} \|A_{\ell k}\|\right) \tag{585}$$

where $\|\cdot\|$ denotes the $2-$induced norm (or maximum singular value) of its matrix argument.

(c) If $\mathcal{A}$ is Hermitian and nonnegative-definite ($\mathcal{A} \geq 0$), then there exist finite positive constants c_1 and c_2 such that

$$c_1 \cdot \mathrm{Tr}(\mathcal{A}) \;\leq\; \|\mathcal{A}\|_{b,\infty} \;\leq\; c_2 \cdot \mathrm{Tr}(\mathcal{A}) \tag{586}$$

Proof. Part (a) follows directly from part (b) by noting that

$$\begin{aligned}
\|\mathcal{A}^*\|_{b,\infty} &\leq N \cdot \left(\max_{1\leq \ell,k\leq N} \|A_{\ell k}^*\|\right) \\
&= N \cdot \left(\max_{1\leq \ell,k\leq N} \|A_{\ell k}\|\right) \\
&\leq N \cdot \|\mathcal{A}\|_{b,\infty}
\end{aligned}$$

where the equality in the second step is because $\|A_{\ell k}^*\| = \|A_{\ell k}\|$; i.e., complex conjugation does not alter the $2-$induced norm of a matrix.

To establish part (b), we consider arbitrary $N \times 1$ block vectors x with entries $x = \mathrm{col}\{x_1, x_2, \ldots, x_N\}$ and where each x_k is $M \times 1$. Then, note that

$$\begin{aligned}
\|\mathcal{A}x\|_{b,\infty} &= \max_{1\leq \ell\leq N} \left\| \sum_{k=1}^{N} A_{\ell k} x_k \right\| \\
&\leq \max_{1\leq \ell\leq N} \left(\sum_{k=1}^{N} \|A_{\ell k}\| \cdot \|x_k\| \right) \\
&\leq \left(\max_{1\leq \ell\leq N} \sum_{k=1}^{N} \|A_{\ell k}\| \right) \cdot \max_{1\leq k\leq N} \|x_k\| \\
&\leq \left(\max_{1\leq \ell\leq N} \sum_{k=1}^{N} \max_{1\leq k\leq N} \|A_{\ell k}\| \right) \cdot \|x\|_{b,\infty} \\
&\leq N \cdot \left(\max_{1\leq \ell,k\leq N} \|A_{\ell k}\| \right) \cdot \|x\|_{b,\infty}
\end{aligned}$$

so that

$$\|\mathcal{A}\|_{b,\infty} \;\triangleq\; \max_{x\neq 0} \frac{\|\mathcal{A}x\|_{b,\infty}}{\|x\|_{b,\infty}} \;\leq\; N \cdot \left(\max_{1\leq \ell,k\leq N} \|A_{\ell k}\|\right)$$

which establishes the upper bound in (585).

To establish the lower bound, we assume without loss of generality that $\max_{1\leq \ell,k\leq N} \|A_{\ell k}\|$ is attained at $\ell = 1$ and $k = 1$. Let σ_1 denote the largest singular value of A_{11} and let $\{v_1, u_1\}$ denote the corresponding $M \times 1$ right and left singular vectors. That is,

$$\|A_{11}\| = \sigma_1, \qquad A_{11} v_1 = \sigma_1 u_1 \tag{587}$$

where v_1 and u_1 have unit norms. We now construct an $N \times 1$ block vector x^o as follows:

$$x^o \triangleq \mathrm{col}\{v_1,\ 0_M,\ 0_M,\ \ldots,\ 0_M\} \tag{588}$$

Then, obviously,

$$\|x^o\|_{b,\infty} = 1 \tag{589}$$

and

$$\mathcal{A}x^o \ = \ \mathrm{col}\{A_{11}v_1,\ A_{21}v_1,\ \ldots,\ A_{N1}v_1\} \tag{590}$$

It follows that

$$\begin{aligned}
\|\mathcal{A}x^o\|_{b,\infty} &= \max\ \{\ \|A_{11}v_1\|,\ \|A_{21}v_1\|,\ \ldots,\ \|A_{N1}v_1\|\ \} \\
&\geq \|A_{11}v_1\| \\
&= \|\sigma_1 u_1\| \\
&= \sigma_1 \\
&= \|A_{11}\| \\
&= \max_{1\leq \ell,k\leq N} \|A_{\ell k}\|
\end{aligned} \tag{591}$$

Therefore, by the definition of the block maximum norm,

$$\begin{aligned}
\|\mathcal{A}\|_{b,\infty} &\triangleq \max_{x\neq 0}\left(\frac{\|\mathcal{A}x\|_{b,\infty}}{\|x\|_{b,\infty}}\right) \\
&\geq \frac{\|\mathcal{A}x^o\|_{b,\infty}}{\|x^o\|_{b,\infty}} \\
&= \|\mathcal{A}x^o\|_{b,\infty} \\
&\geq \max_{1\leq \ell,k\leq N} \|A_{\ell k}\|
\end{aligned} \tag{592}$$

which establishes the lower bound in (585).

To establish part (c), we start by recalling that all norms on finite-dimensional vector spaces are equivalent [20,21]. This means that if $\|\cdot\|_a$ and $\|\cdot\|_d$ denote two different matrix norms, then there exist positive constants c_1 and c_2 such that for any matrix X,

$$c_1 \cdot \|X\|_a \ \leq \ \|X\|_d \ \leq \ c_2 \cdot \|X\|_a \tag{593}$$

Now, let $\|\mathcal{A}\|_*$ denote the nuclear norm of the square matrix $\mathcal{A}$. It is defined as the sum of its singular values:

$$\|\mathcal{A}\|_* \ \triangleq \ \sum_m \sigma_m(\mathcal{A}) \tag{594}$$

Since $\mathcal{A}$ is Hermitian and nonnegative-definite, its eigenvalues coincide with its singular values and, therefore,

$$\|\mathcal{A}\|_* \ = \ \sum_m \lambda_m(\mathcal{A}) \ = \ \mathrm{Tr}(\mathcal{A})$$

Now applying result (593) to the two norms $\|\mathcal{A}\|_{b,\infty}$ and $\|\mathcal{A}\|_*$ we conclude that

$$c_1 \cdot \mathrm{Tr}(\mathcal{A}) \ \leq \ \|\mathcal{A}\|_{b,\infty} \ \leq \ c_2 \cdot \mathrm{Tr}(\mathcal{A}) \tag{595}$$

as desired. □

The next result relates the block maximum norm of an extended matrix to the $\infty-$norm (i.e., maximum absolute row sum) of the originating matrix. Specifically, let A be an $N \times N$ matrix with bounded entries and introduce the block matrix

$$\mathcal{A} \ \triangleq \ A \otimes I_M \tag{596}$$

The extended matrix $\mathcal{A}$ has blocks of size $M \times M$ each.

***Lemma* D.3. (Relation to Maximum Absolute Row Sum)** *Let $\mathcal{A}$ and A be related as in (596). Then, the following properties hold:*

(a) $\|\mathcal{A}\|_{b,\infty} = \|A\|_\infty$*, where the notation* $\|\cdot\|_\infty$ *denotes the maximum absolute row sum of its argument.*

(b) $\|\mathcal{A}^*\|_{b,\infty} \leq N \cdot \|\mathcal{A}\|_{b,\infty}$.

Proof. The results are obvious for a zero matrix A. So assume A is nonzero. Let $x = \mathrm{col}\{x_1, x_2, \ldots, x_N\}$ denote an arbitrary $N \times 1$ block vector whose individual entries $\{x_k\}$ are vectors of size $M \times 1$ each. Then,

$$\begin{aligned}
\|\mathcal{A}x\|_{b,\infty} &= \max_{1\leq k\leq N} \left\| \sum_{\ell=1}^{N} a_{k\ell} x_\ell \right\| \\
&\leq \max_{1\leq k\leq N} \left(\sum_{\ell=1}^{N} |a_{k\ell}| \cdot \|x_\ell\| \right) \\
&\leq \left(\max_{1\leq k\leq N} \sum_{\ell=1}^{N} |a_{k\ell}| \right) \cdot \max_{1\leq \ell\leq N} \|x_\ell\| \\
&= \|A\|_\infty \cdot \|x\|_{b,\infty} \qquad (597)
\end{aligned}$$

so that

$$\|\mathcal{A}\|_{b,\infty} \triangleq \max_{x\neq 0} \frac{\|\mathcal{A}x\|_{b,\infty}}{\|x\|_{b,\infty}} \leq \|A\|_\infty \qquad (598)$$

The argument so far establishes that $\|\mathcal{A}\|_{b,\infty} \leq \|A\|_\infty$. Now, let k_o denote the row index that corresponds to the maximum absolute row sum of A, i.e.,

$$\|A\|_\infty = \sum_{\ell=1}^{N} |a_{k_o\ell}|$$

We construct an $N \times 1$ block vector $z = \mathrm{col}\{z_1, z_2, \ldots, z_N\}$, whose $M \times 1$ entries $\{z_\ell\}$ are chosen as follows:

$$z_\ell = \mathrm{sign}(a_{k_o\ell}) \cdot e_1$$

where e_1 is the $M \times 1$ basis vector:

$$e_1 = \mathrm{col}\{1, 0, 0, \ldots, 0\}$$

and the sign function is defined as

$$\mathrm{sign}(a) = \begin{cases} 1, & \text{if } a \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

Then, note that $z \neq 0$ for any nonzero matrix A, and

$$\|z\|_{b,\infty} = \max_{1\leq \ell\leq N} \|z_\ell\| = 1$$

Moreover,

$$\begin{aligned}
\|\mathcal{A}\|_{b,\infty} &\triangleq \max_{x\neq 0}\frac{\|\mathcal{A}x\|_{b,\infty}}{\|x\|_{b,\infty}} \\
&\geq \frac{\|\mathcal{A}z\|_{b,\infty}}{\|z\|_{b,\infty}} \\
&= \|\mathcal{A}z\|_{b,\infty} \\
&= \max_{1\leq k\leq N}\left\|\sum_{\ell=1}^{N}a_{k\ell}z_\ell\right\| \\
&\geq \left\|\sum_{\ell=1}^{N}a_{k_o\ell}z_\ell\right\| \\
&= \left\|\sum_{\ell=1}^{N}a_{k_o\ell}\cdot\mathrm{sign}(a_{k_o\ell})e_1\right\| \\
&= \sum_{\ell=1}^{N}|a_{k_o\ell}|\cdot\|e_1\| \\
&= \sum_{\ell=1}^{N}|a_{k_o\ell}| \\
&= \|A\|_\infty
\end{aligned} \tag{599}$$

Combining this result with (598) we conclude that $\|\mathcal{A}\|_{b,\infty} = \|A\|_\infty$, which establishes part (a). Part (b) follows from the statement of part (a) in Lemma D.2.

□

The next result establishes a useful property for the block maximum norm of right or left stochastic matrices; such matrices arise as combination matrices for distributed processing over networks as in (166) and (185).

***Lemma* D.4. (Right and Left Stochastic Matrices)** *Let C be an $N\times N$ right stochastic matrix, i.e., its entries are nonnegative and it satisfies $C\mathbb{1} = \mathbb{1}$. Let A be an $N\times N$ left stochastic matrix, i.e., its entries are nonnegative and it satisfies $A^T\mathbb{1} = \mathbb{1}$. Introduce the block matrices*

$$\mathcal{A}^T \triangleq A^T\otimes I_M, \qquad \mathcal{C} \triangleq C\otimes I_M \tag{600}$$

The matrices $\mathcal{A}$ and $\mathcal{C}$ have blocks of size $M\times M$ each. It holds that

$$\boxed{\|\mathcal{A}^T\|_{b,\infty} = 1, \qquad \|\mathcal{C}\|_{b,\infty} = 1} \tag{601}$$

Proof. Since A^T and C are right stochastic matrices, it holds that $\|A^T\|_\infty = 1$ and $\|C\|_\infty = 1$. The desired result then follows from part (a) of Lemma D.3.

□

The next two results establish useful properties for the block maximum norm of a block diagonal matrix transformed by stochastic matrices; such transformations arise as coefficient matrices that control the evolution of weight error vectors over networks, as in (189).

***Lemma* D.5. (Block Diagonal Hermitian Matrices)** *Consider an $N\times N$ block diagonal Hermitian matrix $\mathcal{D} = \mathrm{diag}\{D_1,\ D_2,\ \ldots, D_N\}$, where each D_k is $M\times M$ Hermitian. It holds that*

$$\boxed{\rho(\mathcal{D}) = \max_{1\leq k\leq N}\rho(D_k) = \|\mathcal{D}\|_{b,\infty}} \tag{602}$$

where $\rho(\cdot)$ denotes the spectral radius (largest eigenvalue magnitude) of its argument. That is, the spectral radius of $\mathcal{D}$ agrees with the block maximum norm of $\mathcal{D}$, which in turn agrees with the largest spectral radius of its block components.

Proof. We already know that the spectral radius of any matrix $\mathcal{X}$ satisfies $\rho(\mathcal{X}) \leq \|\mathcal{X}\|$, for any induced matrix norm [19, 20]. Applying this result to $\mathcal{D}$ we readily get that $\rho(\mathcal{D}) \leq \|\mathcal{D}\|_{b,\infty}$. We now establish the reverse inequality, namely, $\|\mathcal{D}\|_{b,\infty} \leq \rho(\mathcal{D})$. Thus, pick an arbitrary $N \times 1$ block vector x with entries $\{x_1, x_2, \ldots, x_N\}$, where each x_k is $M \times 1$. From definition (583) we have

$$\begin{aligned}
\|\mathcal{D}\|_{b,\infty} &\triangleq \max_{x\neq 0} \frac{\|\mathcal{D}x\|_{b,\infty}}{\|x\|_{b,\infty}} \\
&= \max_{x\neq 0} \left(\frac{1}{\|x\|_{b,\infty}} \cdot \max_{1\leq k\leq N} \|D_k x_k\| \right) \\
&\leq \max_{x\neq 0} \left(\frac{1}{\|x\|_{b,\infty}} \cdot \max_{1\leq k\leq N} (\|D_k\| \cdot \|x_k\|) \right) \\
&= \max_{x\neq 0} \max_{1\leq k\leq N} \left(\|D_k\| \cdot \frac{\|x_k\|}{\|x\|_{b,\infty}} \right) \\
&\leq \max_{1\leq k\leq N} \|D_k\| \\
&= \max_{1\leq k\leq N} \rho(D_k)
\end{aligned} \tag{603}$$

where the notation $\|D_k\|$ denotes the 2−induced norm of D_k (i.e., its largest singular value). But since D_k is assumed to be Hermitian, its 2−induced norm agrees with its spectral radius, which explains the last equality.

□

***Lemma* D.6. (Block Diagonal Matrix Transformed by Left Stochastic Matrices)** *Consider an $N \times N$ block diagonal Hermitian matrix $\mathcal{D} = \mathrm{diag}\{D_1,\ D_2,\ \ldots, D_N\}$, where each D_k is $M \times M$ Hermitian. Let A_1 and A_2 be $N \times N$ left stochastic matrices, i.e., their entries are nonnegative and they satisfy $A_1^T \mathbb{1} = \mathbb{1}$ and $A_2^T \mathbb{1} = \mathbb{1}$. Introduce the block matrices*

$$\mathcal{A}_1^T = A_1^T \otimes I_M, \qquad \mathcal{A}_2^T \triangleq A_2^T \otimes I_M \tag{604}$$

The matrices $\mathcal{A}_1$ and $\mathcal{A}_2$ have blocks of size $M \times M$ each. Then it holds that

$$\boxed{\rho\left(\mathcal{A}_2^T \cdot \mathcal{D} \cdot \mathcal{A}_1^T\right) \leq \rho(\mathcal{D})} \tag{605}$$

Proof. Since the spectral radius of any matrix never exceeds any induced norm of the same matrix, we have that

$$\begin{aligned}
\rho\left(\mathcal{A}_2^T \cdot \mathcal{D} \cdot \mathcal{A}_1^T\right) &\leq \left\| \mathcal{A}_2^T \cdot \mathcal{D} \cdot \mathcal{A}_1^T \right\|_{b,\infty} \\
&\leq \left\|\mathcal{A}_2^T\right\|_{b,\infty} \cdot \|\mathcal{D}\|_{b,\infty} \cdot \left\|\mathcal{A}_1^T\right\|_{b,\infty} \\
&\overset{(601)}{=} \|\mathcal{D}\|_{b,\infty} \\
&\overset{(602)}{=} \rho(\mathcal{D})
\end{aligned} \tag{606}$$

□

In view of the result of Lemma D.5, we also conclude from (605) that

$$\boxed{\rho\left(\mathcal{A}_2^T \cdot \mathcal{D} \cdot \mathcal{A}_1^T\right) \leq \max_{1\leq k\leq N} \rho(D_k)} \tag{607}$$

It is worth noting that there are choices for the matrices $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{D}\}$ that would result in strict inequality in (605). Indeed, consider the special case:

$$\mathcal{D} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathcal{A}_1^T = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}, \quad \mathcal{A}_2^T = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

This case corresponds to $N = 2$ and $M = 1$ (scalar blocks). Then,

$$\mathcal{A}_2^T \mathcal{D} \mathcal{A}_1^T = \begin{bmatrix} \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & 1 \end{bmatrix}$$

and it is easy to verify that

$$\rho(\mathcal{D}) \;=\; 2, \qquad \rho(\mathcal{A}_2^T \mathcal{D} \mathcal{A}_1^T) \;\approx\; 1.52$$

The following conclusions follow as corollaries to the statement of Lemma D.6, where by a stable matrix X we mean one whose eigenvalues lie strictly inside the unit circle.

***Corollary* D.1. (Stability Properties)** *Under the same setting of Lemma D.6, the following conclusions hold:*

(a) The matrix $\mathcal{A}_2^T \mathcal{D} \mathcal{A}_1^T$ is stable whenever $\mathcal{D}$ is stable.

(b) The matrix $\mathcal{A}_2^T \mathcal{D} \mathcal{A}_1^T$ is stable for all possible choices of left stochastic matrices $\mathcal{A}_1$ and $\mathcal{A}_2$ if, and only if, $\mathcal{D}$ is stable.

Proof. Since $\mathcal{D}$ is block diagonal, part (a) follows immediately from (605) by noting that $\rho(\mathcal{D}) < 1$ whenever $\mathcal{D}$ is stable. [This statement fixes the argument that appeared in App. I of [18] and Lemma 2 of [33]. Since the matrix X in App. I of [18] and the matrix $\mathcal{M}$ in Lemma 2 of [33] are block diagonal, the $\|\cdot\|_{b,\infty}$ norm should replace the $\|\cdot\|_\rho$ norm used there, as in the proof that led to (606) and as already done in [54].] For part (b), assume first that $\mathcal{D}$ is stable, then $\mathcal{A}_2^T \mathcal{D} \mathcal{A}_1^T$ will also be stable by part (a) for any left-stochastic matrices $\mathcal{A}_1$ and $\mathcal{A}_2$. To prove the converse, assume that $\mathcal{A}_2^T \mathcal{D} \mathcal{A}_1^T$ is stable for any choice of left stochastic matrices $\mathcal{A}_1$ and $\mathcal{A}_2$. Then, $\mathcal{A}_2^T \mathcal{D} \mathcal{A}_1^T$ is stable for the particular choice $\mathcal{A}_1 = I = \mathcal{A}_2$ and it follows that $\mathcal{D}$ must be stable.

□

E Appendix: Comparison with Consensus Strategies

Consider a connected network consisting of N nodes. Each node has a state or measurement value x_k, possibly a vector of size $M \times 1$. All nodes in the network are interested in evaluating the average value of their states, which we denote by

$$w^o \;\triangleq\; \frac{1}{N} \sum_{k=1}^{N} x_k \tag{608}$$

A centralized solution to this problem would require each node to transmit its measurement x_k to a fusion center. The central processor would then compute w^o using (608) and transmit it back to all nodes. This centralized mode of operation suffers from at least two limitations. First, it requires communications and power resources to transmit the data back and forth between the nodes and the central processor; this problem is compounded if the fusion center is stationed at a remote location. Second, the architecture has a critical point of failure represented by the central processor; if it fails, then operations would need to be halted.

Consensus Recursion

The consensus strategy provides an elegant distributed solution to the same problem, whereby nodes interact locally with their neighbors and are able to converge to w^o through these interactions. Thus, consider an

arbitrary node k and assign nonnegative weights $\{a_{\ell k}\}$ to the edges linking k to its neighbors $\ell \in \mathcal{N}_k$. For each node k, the weights $\{a_{\ell k}\}$ are assumed to add up to one so that

$$\text{for } k = 1, 2, \ldots, N:$$
$$a_{\ell k} \geq 0, \quad \sum_{\ell=1}^{N} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{609}$$

The resulting combination matrix is denoted by A and its $k-$th column consists of the entries $\{a_{\ell k}, \ell = 1, 2, \ldots, N\}$. In view of (609), the combination matrix A is seen to satisfy $A^T \mathbb{1} = \mathbb{1}$. That is, A is left-stochastic. The consensus strategy can be described as follows. Each node k operates repeatedly on the data from its neighbors and updates its state iteratively according to the rule:

$$\boxed{w_{k,n} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}\, w_{\ell,n-1}, \quad n > 0} \tag{610}$$

where $w_{\ell,n-1}$ denotes the state of node ℓ at iteration $n-1$, and $w_{k,n}$ denotes the updated state of node k after iteration n. The initial conditions are

$$w_{k,o} = x_k, \quad k = 1, 2, \ldots, N \tag{611}$$

If we collect the states of all nodes at iteration n into a column vector, say,

$$z_n \triangleq \text{col}\{w_{1,n},\, w_{2,n},\, \ldots, w_{N,n}\} \tag{612}$$

Then, the consensus iteration (610) can be equivalently rewritten in vector form as follows:

$$\boxed{z_n = \mathcal{A}^T z_{n-1}, \quad n > 0} \tag{613}$$

where

$$\mathcal{A}^T = A^T \otimes I_M \tag{614}$$

The initial condition is

$$z_o \triangleq \text{col}\{x_1,\, x_2,\, \ldots, x_N\} \tag{615}$$

Error Recursion

Note that we can express the average value, w^o, from (608) in the form

$$\boxed{w^o = \frac{1}{N} \cdot (\mathbb{1}^T \otimes I_M) \cdot z_o} \tag{616}$$

where $\mathbb{1}$ is the vector of size $M \times 1$ and whose entries are all equal to one. Let

$$\widetilde{w}_{k,n} = w^o - w_{k,n} \tag{617}$$

denote the weight error vector for node k at iteration n; it measures how far the iterated state is from the desired average value w^o. We collect all error vectors across the network into an $N \times 1$ block column vector whose entries are of size $M \times 1$ each:

$$\widetilde{w}_n \triangleq \begin{bmatrix} \widetilde{w}_{1,n} \\ \widetilde{w}_{2,n} \\ \vdots \\ \widetilde{w}_{N,n} \end{bmatrix} \tag{618}$$

Then,

$$\boxed{\widetilde{w}_n \;=\; (\mathbb{1}\otimes I_M)w^o \;-\; z_n} \tag{619}$$

Convergence Conditions

The following result is a classical result on consensus strategies [42–44]. It provides conditions under which the state of all nodes will converge to the desired average, w^o, so that $\widetilde{w}_n$ will tend to zero.

Theorem* E.1.** *(Convergence to Consensus***) For any initial states $\{x_k\}$, the successive iterates $w_{k,n}$ generated by the consensus iteration (610) converge to the network average value w^o as $n\to\infty$ if, and only if, the following three conditions are met:*

$$A^T\mathbb{1} \;=\; \mathbb{1} \tag{620}$$

$$A\mathbb{1} \;=\; \mathbb{1} \tag{621}$$

$$\rho\left(A^T-\frac{1}{N}\mathbb{1}\mathbb{1}^T\right) \;<\; 1 \tag{622}$$

That is, the combination matrix A needs to be doubly stochastic, and the matrix $A^T-\frac{1}{N}\mathbb{1}\mathbb{1}^T$ needs to be stable.

Proof. (*Sufficiency*). Assume first that the three conditions stated in the theorem hold. Since A is doubly stochastic, then so is any power of A, say, A^n for any $n\geq 0$, so that

$$[A^n]^T\,\mathbb{1}=\mathbb{1}, \qquad A^n\mathbb{1}=\mathbb{1} \tag{623}$$

Using this fact, it is straightforward to verify by induction the validity of the following equality:

$$\left(A^T-\frac{1}{N}\mathbb{1}\mathbb{1}^T\right)^n \;=\; [A^n]^T-\frac{1}{N}\mathbb{1}\mathbb{1}^T \tag{624}$$

Likewise, using the Kronecker product identities

$$(E+B)\otimes C \;=\; (E\otimes C)\;+\;(B\otimes C) \tag{625}$$

$$(E\otimes B)(C\otimes D) \;=\; (EC\otimes BD) \tag{626}$$

$$(E\otimes B)^n \;=\; E^n\otimes B^n \tag{627}$$

for matrices $\{E,B,C,D\}$ of compatible dimensions, we observe that

$$\begin{aligned}
(\mathcal{A}^n)^T \;-\; \frac{1}{N}\cdot(\mathbb{1}\otimes I_M)\cdot(\mathbb{1}^T\otimes I_M) \;&=\; \left[(A^n)^T\otimes I_M\right] \;-\; \frac{1}{N}\cdot(\mathbb{1}\mathbb{1}^T\otimes I_M)\\
&=\; \left[(A^n)^T \;-\; \frac{1}{N}\cdot\mathbb{1}\mathbb{1}^T\right]\otimes I_M\\
&\overset{(624)}{=}\; \left(A^T-\frac{1}{N}\mathbb{1}\mathbb{1}^T\right)^n\otimes I_M\\
&=\; \left[\left(A^T-\frac{1}{N}\mathbb{1}\mathbb{1}^T\right)\otimes I_M\right]^n
\end{aligned} \tag{628}$$

Iterating (613) we find that

$$z_n=[\mathcal{A}^n]^T\,z_o \tag{629}$$

and, hence, from (616) and (619),

$$\begin{aligned}
\widetilde{w}_n \;&=\; -\left[(\mathcal{A}^n)^T \;-\; \frac{1}{N}\cdot(\mathbb{1}\otimes I_M)\cdot(\mathbb{1}^T\otimes I_M)\right]\cdot z_o\\
&\overset{(628)}{=}\; -\left[\left(A^T-\frac{1}{N}\mathbb{1}\mathbb{1}^T\right)\otimes I_M\right]^n\cdot z_o
\end{aligned} \tag{630}$$

Now recall that, for two arbitrary matrices C and D of compatible dimensions, the eigenvalues of the Kronecker product $C \otimes D$ is formed of all product combinations $\lambda_i(C)\lambda_j(D)$ of the eigenvalues of C and D [19]. We conclude from this property, and from the fact that $A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T$ is stable, that the coefficient matrix

$$\left(A^T \; - \; \frac{1}{N} \cdot \mathbb{1}\mathbb{1}^T\right) \otimes I_M$$

is also stable. Therefore,

$$\widetilde{w}_n \;\; \rightarrow \; 0 \quad \text{as } n \rightarrow \infty \tag{631}$$

(*Necessity*). In order for z_n in (629) to converge to $(\mathbb{1} \otimes I_M)w^o$, for any initial state z_o, it must hold that

$$\lim_{n\rightarrow\infty} (\mathcal{A}^n)^T \cdot z_o \; = \; \frac{1}{N} \cdot (\mathbb{1} \otimes I_M) \cdot (\mathbb{1}^T \otimes I_M) \cdot z_o \tag{632}$$

for any z_o. This implies that we must have

$$\lim_{n\rightarrow\infty} (\mathcal{A}^n)^T \; = \; \frac{1}{N} \cdot (\mathbb{1}\mathbb{1}^T \otimes I_M) \tag{633}$$

or, equivalently,

$$\lim_{n\rightarrow\infty} (A^n)^T \; = \; \frac{1}{N}\mathbb{1}\mathbb{1}^T \tag{634}$$

This in turn implies that we must have

$$\lim_{n\rightarrow\infty} A^T \cdot (A^n)^T \; = \; A^T \cdot \frac{1}{N}\mathbb{1}\mathbb{1}^T \tag{635}$$

But since

$$\lim_{n\rightarrow\infty} A^T \cdot (A^n)^T \; = \; \lim_{n\rightarrow\infty} \left(A^{n+1}\right)^T \; = \; \lim_{n\rightarrow\infty} (A^n)^T \tag{636}$$

we conclude from (634) and (635) that it must hold that

$$\frac{1}{N}\mathbb{1}\mathbb{1}^T \; = \; \frac{1}{N}A^T \cdot \mathbb{1}\mathbb{1}^T \tag{637}$$

That is,

$$\frac{1}{N}\left(A^T\mathbb{1} \; - \; \mathbb{1}\right) \cdot \mathbb{1}^T = 0 \tag{638}$$

from which we conclude that we must have $A^T\mathbb{1} = \mathbb{1}$. Similarly, we can show that $A\mathbb{1} = \mathbb{1}$ by studying the limit of $(A^n)^T A^T$. Therefore, A must be a doubly stochastic matrix. Now using the fact that A is doubly stochastic, we know that (624) holds. It follows that in order for condition (634) to be satisfied, we must have

$$\rho\left(A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T\right) < 1 \tag{639}$$

□

Rate of Convergence

From (630) we conclude that the rate of convergence of the error vectors $\{\widetilde{w}_{k,n}\}$ to zero is determined by the spectrum of the matrix

$$A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T \tag{640}$$

Now since A is a doubly stochastic matrix, we know that it has an eigenvalue at $\lambda = 1$. Let us denote the eigenvalues of A by $\lambda_k(A)$ and let us order them in terms of their magnitudes as follows:

$$0 \leq \; |\lambda_M(A)| \; \leq \; \ldots \; \leq \; |\lambda_3(A)| \; \leq \; |\lambda_2(A)| \; \leq \; 1 \tag{641}$$

where $\lambda_1(A) = 1$. Then, the eigenvalues of the coefficient matrix $(A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T)$ are equal to

$$\{ \; \lambda_M(A), , \ldots, \; \lambda_3(A), \; \lambda_2(A), 0 \; \} \tag{642}$$

It follows that the magnitude of $\lambda_2(A)$ becomes the spectral radius of $A^T - \frac{1}{N}\mathbb{1}\mathbb{1}^T$. Then condition (639) ensures that $|\lambda_2(A)| < 1$. We therefore arrive at the following conclusion.

Corollary* E.1.** *(Rate of Convergence of Consensus***) Under conditions (620)–(622), the rate of convergence of the successive iterates $\{w_{k,n}\}$ towards the network average w^o in the consensus strategy (610) is determined by the second largest eigenvalue magnitude of A, i.e., by $|\lambda_2(A)|$ as defined in (641).*

□

It is worth noting that doubly stochastic matrices A that are also *regular* satisfy conditions (620)–(622). This is because, as we already know from Lemma C.2, the eigenvalues of such matrices satisfy $|\lambda_m(A)| < 1$, for $m = 2, 3, \ldots, N$, so that condition (622) is automatically satisfied.

Corollary* E.2.** *(Convergence for Regular Combination Matrices***) Any doubly-stochastic and regular matrix A satisfies the three conditions (620)–(622) and, therefore, ensures the convergence of the consensus iterates $\{w_{k,n}\}$ generated by (610) towards w^o as $n \to \infty$.*

□

A regular combination matrix A would result when the two conditions listed below are satisfied by the graph connecting the nodes over which the consensus iteration is applied.

Corollary* E.3.** *(Sufficient Condition for Regularity***) Assume the combination matrix A is doubly stochastic and that the graph over which the consensus iteration (610) is applied satisfies the following two conditions:*

(a) The graph is connected. This means that there exists a path connecting any two arbitrary nodes in the network. In terms of the Laplacian matrix that is associated with the graph (see Lemma B.1), this means that the second smallest eigenvalue of the Laplacian is nonzero.

(b) $a_{\ell k} = 0$ if, and only if, $\ell \notin \mathcal{N}_k$. That is, the combination weights are strictly positive between any two neighbors, including $a_{kk} > 0$.

Then, the corresponding matrix A will be regular and, therefore, the consensus iterates $\{w_{k,n}\}$ generated by (610) will converge towards w^o as $n \to \infty$.

Proof. We first establish that conditions (a) and (b) imply that A is a regular matrix, namely, that there should exist an integer $j_o > 0$ such that

$$\left[A^{j_o}\right]_{\ell k} > 0 \tag{643}$$

for all (ℓ, k). To begin with, by the rules of matrix multiplication, the (ℓ, k) entry of the $i-$th power of A is given by:

$$\left[A^{i}\right]_{\ell k} = \sum_{m_1=1}^{N} \sum_{m_2=1}^{N} \cdots \sum_{m_{i-1}=1}^{N} a_{\ell m_1} a_{m_1 m_2} \ldots a_{m_{i-1} k} \tag{644}$$

The summand in (644) is nonzero if, and only if, there is some sequence of indices $(\ell, m_1, \ldots, m_{i-1}, k)$ that forms a path from node ℓ to node k. Since the network is assumed to be connected, there exists a minimum (and finite) integer value $i_{\ell k}$ such that a path exists from node ℓ to node k using $i_{\ell k}$ edges and that

$$\left[A^{i_{\ell k}}\right]_{\ell k} > 0$$

In addition, by induction, if $\left[A^{i_{\ell k}}\right]_{\ell k} > 0$, then

$$\begin{aligned} \left[A^{i_{\ell k}+1}\right]_{\ell k} &= \sum_{m=1}^{N} \left[A^{i_{\ell k}}\right]_{\ell m} a_{mk} \\ &\geq \left[A^{i_{\ell k}}\right]_{\ell k} a_{kk} \\ &> 0 \end{aligned}$$

Let

$$j_o = \max_{1 \leq k, \ell \leq N} \{i_{\ell k}\}$$

Then, property (643) holds for all (ℓ, k). And we conclude from (581) that A is a regular matrix. It then follows from Corollary E.2 that the consensus iterates $\{w_{k,n}\}$ converge to the average network value w^o.

□

Comparison with Diffusion Strategies

Observe that in comparison to diffusion strategies, such as the ATC strategy (153), the consensus iteration (610) employs the same quantities $w_{k,\cdot}$ on both sides of the iteration. In other words, the consensus construction keeps iterating on the same set of vectors until they converge to the average value w^o. Moreover, the index n in the consensus algorithm is an iteration index. In contrast, diffusion strategies employ different quantities on both sides of the combination step in (153), namely, $w_{k,i}$ and $\{\psi_{\ell,i}\}$; the latter variables have been processed through an information exchange step and are updated (or filtered) versions of the $w_{\ell,i-1}$. In addition, each step of the diffusion strategy (153) can incorporate new data, $\{d_\ell(i), u_{\ell,i}\}$, that are collected by the nodes at every time instant. Moreover, the index i in the diffusion implementation is a time index (and not an iteration index); this is because diffusion strategies are inherently adaptive and perform online learning. Data keeps streaming in and diffusion incorporates the new data into the update equations at every time instant. As a result, diffusion strategies are able to respond to data in an adaptive manner, and they are also able to solve general optimization problems: the vector w^o in adaptive diffusion iterations is the minimizer of a global cost function (cf. (92)), while the vector w^o in consensus iterations is the average value of the initial states of the nodes (cf. (608)).

Moreover, it turns out that diffusion strategies influence the evolution of the network dynamics in an interesting and advantageous manner in comparison to consensus strategies. We illustrate this point by means of an example. Consider initially the ATC strategy (158) without information exchange, whose update equation we repeat below for ease of reference:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^* \left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}\right] \tag{645}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{\psi}_{\ell,i} \qquad \text{(ATC diffusion)} \tag{646}$$

These recursions were derived in the body of the article as an effective distributed solution for optimizing (92)–(93). Note that they involve two steps, where the weight estimator $\boldsymbol{w}_{k,i-1}$ is first updated to the intermediate estimator $\boldsymbol{\psi}_{k,i}$, before the intermediate estimators from across the neighborhood are combined to obtain $\boldsymbol{w}_{k,i}$. Both steps of ATC diffusion (645)–(646) can be combined into a single update as follows:

$$\boxed{\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\left[\boldsymbol{w}_{\ell,i-1} + \mu_\ell \boldsymbol{u}_{\ell,i}^*\left(\boldsymbol{d}_\ell(i) - \boldsymbol{u}_{\ell,i}\boldsymbol{w}_{\ell,i-1}\right)\right]} \qquad \text{(ATC diffusion)} \tag{647}$$

Likewise, consider the CTA strategy (159) without information exchange, whose update equation we also repeat below:

$$\boldsymbol{\psi}_{k,i-1} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \qquad \text{(CTA diffusion)} \tag{648}$$

$$\boldsymbol{w}_{k,i} = \boldsymbol{\psi}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^* \left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k,i-1}\right] \tag{649}$$

Again, the CTA strategy involves two steps: the weight estimators $\{\boldsymbol{w}_{\ell,i-1}\}$ from the neighborhood of node k are first combined to yield the intermediate estimator $\boldsymbol{\psi}_{k,i-1}$, which is subsequently updated to $\boldsymbol{w}_{k,i}$. Both steps of CTA diffusion can also be combined into a single update as follows:

$$\boxed{\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} + \mu_k \boldsymbol{u}_{k,i}^* \left[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\sum_{\ell\in\mathcal{N}_k} a_{\ell k}\boldsymbol{w}_{\ell,i-1}\right]} \qquad \text{(CTA diffusion)} \tag{650}$$

Now, motivated by the consensus iteration (610), and based on a procedure for distributed optimization suggested in [52] (see expression (7.1) in that reference), some works in the literature (e.g., [45, 53, 82–88]) considered distributed strategies that correspond to the following form for the optimization problem under consideration (see, e.g., expression (1.20) in [53] and expression (9) in [87]):

$$\boxed{\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_{\ell,i-1} \;+\; \mu_k \boldsymbol{u}_{k,i}^{*}\,[\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}]} \qquad \text{(consensus strategy)} \tag{651}$$

This strategy can be derived by following the same argument we employed earlier in Secs. 3.2 and 4 to arrive at the diffusion strategies, namely, we replace w^o in (127) by $w_{\ell,i-1}$ and then apply the instantaneous approximations (150). Note that the *same* variable $\boldsymbol{w}_{k,\cdot}$ appears on both sides of the equality in (651). Thus, compared with the ATC diffusion strategy (647), the update from $\boldsymbol{w}_{k,i-1}$ to $\boldsymbol{w}_{k,i}$ in the consensus implementation (651) is only influenced by data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ from node k. In contrast, the ATC diffusion structure (645)–(646) helps incorporate the influence of the data $\{\boldsymbol{d}_\ell(i), \boldsymbol{u}_{\ell,i}\}$ from across the neighborhood of node k into the update of $\boldsymbol{w}_{k,i}$, since these data are reflected in the intermediate estimators $\{\boldsymbol{\psi}_{\ell,i}\}$. Likewise, the contrast with the CTA diffusion strategy (650) is clear, where the right-most term in (650) relies on a combination of all estimators from across the neighborhood of node k, and not only on $\boldsymbol{w}_{k,i-1}$ as in the consensus strategy (651). These facts have desirable implications on the evolution of the weight-error vectors across diffusion networks. Some simple algebra, similar to what we did in Sec. 6, will show that the mean of the extended error vector for the consensus strategy (651) evolves according to the recursion:

$$\boxed{\mathbb{E}\,\widetilde{\boldsymbol{w}}_i = \left(\mathcal{A}^T - \mathcal{M}\mathcal{R}_u\right)\cdot \mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1},\;\; i\geq 0} \qquad \text{(consensus strategy)} \tag{652}$$

where $\mathcal{R}_u$ is the block diagonal covariance matrix defined by (184) and $\widetilde{\boldsymbol{w}}_i$ is the aggregate error vector defined by (230). We can compare the above mean error dynamics with the ones that correspond to the ATC and CTA diffusion strategies (645)–(646) and (648)–(650); their error dynamics follow as special cases from (248) by setting $A_1 = I = C$ and $A_2 = A$ for ATC and $A_2 = I = C$ and $A_1 = A$ for CTA:

$$\boxed{\mathbb{E}\,\widetilde{\boldsymbol{w}}_i = \mathcal{A}^T\,(I_{NM} - \mathcal{M}\mathcal{R}_u)\cdot \mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1},\;\; i\geq 0} \qquad \text{(ATC diffusion)} \tag{653}$$

and

$$\boxed{\mathbb{E}\,\widetilde{\boldsymbol{w}}_i = (I_{NM} - \mathcal{M}\mathcal{R}_u)\,\mathcal{A}^T\cdot \mathbb{E}\,\widetilde{\boldsymbol{w}}_{i-1},\;\; i\geq 0} \qquad \text{(CTA diffusion)} \tag{654}$$

We observe that the coefficient matrices that control the evolution of $\mathbb{E}\,\widetilde{\boldsymbol{w}}_i$ are different in all three cases. In particular,

$$\text{consensus strategy (652) is stable in the mean} \iff \rho\left(\mathcal{A}^T - \mathcal{M}\mathcal{R}_u\right) < 1 \tag{655}$$

$$\text{ATC diffusion (653) is stable in the mean} \iff \rho\left[\mathcal{A}^T\,(I_{NM} - \mathcal{M}\mathcal{R}_u)\right] < 1 \tag{656}$$

$$\text{CTA diffusion (654) is stable in the mean} \iff \rho\left[(I_{NM} - \mathcal{M}\mathcal{R}_u)\,\mathcal{A}^T\right] < 1 \tag{657}$$

It follows that the mean stability of the consensus network is sensitive to the choice of the combination matrix A. This is not the case for the diffusion strategies. This is because from property (605) established in App. D, we know that the matrices $\mathcal{A}^T\,(I_{NM} - \mathcal{M}\mathcal{R}_u)$ and $(I_{NM} - \mathcal{M}\mathcal{R}_u)\,\mathcal{A}^T$ are stable if $(I_{NM} - \mathcal{M}\mathcal{R}_u)$ is stable. Therefore, we can select the step-sizes to satisfy $\mu_k < 2/\lambda_{\max}(R_{u,k})$ for the ATC or CTA diffusion strategies and ensure their mean stability regardless of the combination matrix A. This also means that the diffusion networks will be mean stable whenever the individual nodes are mean stable, regardless of the topology defined by A. In contrast, for consensus networks, the network can exhibit unstable mean behavior

even if all its individual nodes are stable in the mean. For further details and other results on the mean-square performance of diffusion networks in relation to consensus networks, the reader is referred to [89, 90].

Acknowledgement. The development of the theory and applications of diffusion adaptation over networks has benefited greatly from the insights and contributions of several UCLA Ph.D. students, and several visiting graduate students to the UCLA Adaptive Systems Laboratory (http://www.ee.ucla.edu/asl). The assistance and contributions of all students are hereby gratefully acknowledged, including Cassio G. Lopes, Federico S. Cattivelli, Sheng-Yuan Tu, Jianshu Chen, Xiaochuan Zhao, Zaid Towfic, Chung-Kai Yu, Noriyuki Takahashi, Jae-Woo Lee, Alexander Bertrand, and Paolo Di Lorenzo. The author is also particularly thankful to S.-Y. Tu, J. Chen, X. Zhao, Z. Towfic, and C.-K. Yu for their assistance in reviewing an earlier draft of this chapter.

References

[1] J. Chen and A. H. Sayed, "On the limiting behavior of distributed optimization strategies," *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1–8, Allerton, IL, October 2012.

[2] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Processing*, vol. 60, no. 8, pp. 4289–4305, August 2012.

[3] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE J. Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, April 2013.

[4] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley, NJ, 2003.

[5] A. H. Sayed, *Adaptive Filters*, Wiley, NJ, 2008.

[6] S. Haykin, *Adaptive Filter Theory*, 4th edition, Prentice Hall, NJ, 2002.

[7] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, NJ, 1985.

[8] S.-Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Sel. Topics. Signal Process.*, vol. 5, no. 4, pp. 649–664, Aug. 2011.

[9] F. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2038–2051, May 2011.

[10] J. Li and A. H. Sayed, "Modeling bee swarming behavior through diffusion adaptation with asymmetric information sharing," *EURASIP Journal on Advances in Signal Processing*, 2012:18, doi:10.1186/1687-6180-2012-18, 2012.

[11] J. Chen and A. H. Sayed, "Bio-inspired cooperative optimization with application to bacteria motility," *Proc. ICASSP*, Prague, Czech Republic, pp. 5788–5791, May 2011.

[12] A. H. Sayed and F. A. Sayed, "Diffusion adaptation over networks of particles subject to Brownian fluctuations," *Proc. Asilomar Conference on Signals, Systems, and Computers*, pp. 685–690, Pacific Grove, CA, November 2011.

[13] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Commun.*, vol. 6, pp. 13-18, 1999.

[14] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201-220, Feb. 2005.

[15] Z. Quan, W. Zhang, S. J. Shellhammer, and A. H. Sayed, "Optimal spectral feature detection for spectrum sensing at very low SNR," *IEEE Transactions on Communications*, vol. 59, no. 1, pp. 201-212, January 2011.

[16] Q. Zou, S. Zheng, and A. H. Sayed, "Cooperative sensing via sequential detection," *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6266-6283, December 2010.

[17] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Bio-inspired swarming for dynamic radio access based on diffusion adaptation," *Proc. EUSIPCO*, pp. 402-406, Barcelona, Spain, August 2011.

[18] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, March 2010.

[19] Golub, G. H. and C. F. Van Loan (1996), *Matrix Computations*, 3rd edition, The John Hopkins University Press, Baltimore.

[20] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2003.

[21] E. Kreyszig, *Introductory Functional Analysis with Applications*, Wiley, NY, 1989.

[22] B. Poljak, *Introduction to Optimization*, Optimization Software, NY, 1987.

[23] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim.*, vol. 7, no. 4, pp. 913–926, 1997.

[24] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM J. Optim.,* vol. 12, no. 1, pp. 109–138, 2001.

[25] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, 2005.

[26] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.,* vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[27] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS algorithms with information exchange," *Proc. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, pp. 251–255, Nov. 2008.

[28] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "A diffusion RLS scheme for distributed estimation over adaptive networks," *Proc. IEEE Workshop on Signal Process. Advances Wireless Comm. (SPAWC)*, Helsinki, Finland, pp. 1–5, June 2007.

[29] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[30] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering: Formulation and performance analysis," *Proc. IAPR Workshop on Cognitive Inf. Process.*(CIP), Santorini, Greece, pp. 36–41, June 2008.

[31] F. S. Cattivelli and A. H. Sayed, "Diffusion mechanisms for fixed-point distributed Kalman smoothing," *Proc. EUSIPCO*, Lausanne, Switzerland, pp. 1–4, Aug. 2008.

[32] A. H. Sayed and F. Cattivelli, "Distributed adaptive learning mechanisms," *Handbook on Array Processing and Sensor Networks,* S. Haykin and K. J. Ray Liu, *Eds.*, pp. 695–722, Wiley, NJ, 2009.

[33] F. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[34] S. S. Stankovic, M. S. Stankovic, and D. S. Stipanovic, "Decentralized parameter estimation by consensus based stochastic approximation," *IEEE Trans. on Autom. Control*, vol. 56, no. 3, pp. 531–543, Mar. 2011.

[35] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," in *Proc. Adaptive Sensor Array Processing Workshop*, MIT Lincoln Laboratory, MA, pp.1–5, June 2006.

[36] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fund. of Electron., Commun. and Comput. Sci.*, vol. E90-A, no. 8, pp. 1504–1510, 2007.

[37] C. G. Lopes and A. H. Sayed, "Diffusion least-mean-squares over adaptive networks," *Proc. IEEE ICASSP*, Honolulu, Hawaii, vol. 3, pp. 917-920, April 2007.

[38] C. G. Lopes and A. H. Sayed, "Steady-state performance of adaptive diffusion least-mean squares," *Proc. IEEE Workshop on Statistical Signal Processing* (SSP), pp. 136-140, Madison, WI, August 2007.

[39] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[40] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

[41] P. Bianchi, G. Fort, W. Hachem, and J. Jakubowicz, "Convergence of a distributed parameter estimator for sensor networks with local averaging of the estimates," *Proc. IEEE ICASSP,* Prague, Czech, pp. 3764–3767, May 2011.

[42] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association,* vol. 69, no. 345, pp. 118–121, 1974.

[43] R. L. Berger, "A necessary and sufficient condition for reaching a consensus using DeGroot's method," *Journal of the American Statistical Association*, vol. 76, no. 374, pp. 415-418, Jun. 1981.

[44] J. Tsitsiklis and M. Athans, "Convergence and asymptotic agreement in distributed decision problems," *IEEE Trans. Autom. Control*, vol. 29, no. 1, pp. 42–50, Jan. 1984.

[45] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[46] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, pp. 1520–1533, Sep. 2004.

[47] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," *Proc. 44th IEEE Conf. Decision Control*, pp. 8179–8184, Sevilla, Spain, Dec. 2005.

[48] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," *Proc. 46th IEEE Conf. Decision Control*, pp. 5492–5498, New Orleans, LA, Dec. 2007.

[49] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[50] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *Proc. IPSN*, 2005, pp. 63–70, Los Angeles, CA, April 2005.

[51] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Processing*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

[52] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 1st edition, Athena Scientific, Singapore, 1997.

[53] A. Nedic and A. Ozdaglar, "Cooperative distributed multi-agent optimization," in *Convex Optimization in Signal Processing and Communications*, Y. Eldar and D. Palomar (Eds.), Cambridge University Press, pp. 340-386, 2010.

[54] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean-squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. on Signal Processing*, vol. 9, pp. 4795-4810, Sep. 2010.

[55] N. Takahashi and I. Yamada, "Parallel algorithms for variational inequalities over the cartesian product of the intersections of the fixed point sets of nonexpansive mappings," *J. Approx. Theory,* vol. 153, no. 2, pp. 139–160, Aug. 2008.

[56] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[57] T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 639–652, March 2003.

[58] 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), San Juan, Puerto Rico, pp. 317–320, December 2011.

[59] R. Abdolee and B. Champagne, "Diffusion LMS algorithms for sensor networks over non-ideal inter-sensor wireless channels," *Proc. IEEE Int. Conf. Dist. Comput. Sensor Systems* (DCOSS), pp. 1–6, Barcelona, Spain, June 2011.

[60] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers, "Steady state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Trans. Signal Processing*, vol. 60, no. 2, pp. 974–979, Feb. 2012.

[61] S-Y. Tu and A. H. Sayed, "Adaptive networks with noisy links," *Proc. IEEE Globecom*, pp. 1–5, Houston, TX, December 2011.

[62] X. Zhao, S-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, July 2012.

[63] X. Zhao and A. H. Sayed, "Combination weights for diffusion strategies with imperfect information exchange," *Proc. IEEE ICC*, pp. 1–5, Ottawa, Canada, June 2012.

[64] D. M. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Applications*, Wiley, NY, 1998.

[65] B. Bollobas, *Modern Graph Theory*, Springer, 1998.

[66] W. Kocay and D. L. Kreher, *Graphs, Algorithms and Optimization*, Chapman & Hall/CRC Press, Boca Raton, 2005.

[67] M. Fiedler, "Algebraic connectivity of graphs," *Czech. Math. J.*, vol. 23, pp. 298–305, 1973.

[68] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," *Proc. Joint 44th IEEE Conf. on Decision and Control and European Control Conf.* (CDC-ECC), pp. 2996-3000, Seville, Spain, Dec. 2005.

[69] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," *Proc. Information Processing in Sensor Networks (IPSN)*, pp. 11-19, Berkeley, CA, April 2004.

[70] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087-1092, 1953.

[71] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97-109, 1970.

[72] A. H. Sayed and C. Lopes, "Distributed recursive least-squares strategies over adaptive networks," *Proc. 40th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, pp. 233-237, October-November, 2006.

[73] J-W. Lee, S-E. Kim, W-J. Song, and A. H. Sayed, "Spatio-temporal diffusion mechanisms for adaptation over networks," *Proc. EUSIPCO*, pp. 1040-1044, Barcelona, Spain, August-September 2011.

[74] J-W. Lee, S-E. Kim, W-J. Song, and A. H. Sayed, "Spatio-temporal diffusion strategies for estimation and detection over networks," *IEEE Trans. Signal Processing*, vol. 60, no. 8, pp. 4017–4034, August 2012.

[75] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[76] K. Slavakis, Y. Kopsinis, and S. Theodoridis, "Adaptive algorithm for sparse system identification using projections onto weighted ℓ_1 balls," *Proc. IEEE ICASSP*, pp. 3742–3745, Dallas, TX, March 2010.

[77] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, NJ, 2000.

[78] F. Cattivelli and A. H. Sayed, "Diffusion distributed Kalman filtering with adaptive weights," *Proc. Asilomar Conference on Signals, Systems and Computers*, pp. 908–912, Pacific Grove, CA, November 2009.

[79] 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)*, pp. 168–176, Nashville, TN, April 2006.

[80] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[81] D. P. Bertsekas and J. N. Tsitsiklis, "Gradient convergence in gradient methods with errors," *SIAM J. Optim.*, vol. 10, no. 3, pp. 627–642, 2000.

[82] S. Barbarossa, and G. Scutari, "Bio-inspired sensor network design," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 26–35, May 2007.

[83] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," *Proc. IEEE CDC*, pp. 7036–7042, Shangai, China, 2009.

[84] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, June 2009.

[85] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Performance analysis of the consensus-based distributed LMS algorithm," *EURASIP J. Adv. Signal Process.*, pp. 1–19, 2009.

[86] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[87] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE Journal on Selected Topics on Signal Processing*, vol. 5, no. 4, pp. 674–690, August 2011.

[88] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, November 2010.

[89] S-Y. Tu and A. H. Sayed, "Diffusion networks outperform consensus networks," *Proc. IEEE Statistical Signal Processing Workshop*, pp. 313–316, Ann Arbor, Michigan, August 2012.

[90] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Processing*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.