Stability and Performance Limits of Adaptive Primal-Dual Networks

Zaid J. Towfic, Student Member, IEEE, and Ali H. Sayed, Fellow, IEEE

Abstract—This work studies distributed primal-dual strategies for adaptation and learning over networks from streaming data. Two first-order methods are considered based on the Arrow-Hurwicz (AH) and augmented Lagrangian (AL) techniques. Several revealing results are discovered in relation to the performance and stability of these strategies when employed over adaptive networks. The conclusions establish that the advantages that these methods have for deterministic optimization problems do not necessarily carry over to stochastic optimization problems. It is found that they have narrower stability ranges and worse steady-state mean-square-error performance than primal methods of the consensus and diffusion type. It is also found that the AH technique can become unstable under a partial observation model, while the other techniques are able to recover the unknown under this scenario. A method to enhance the performance of AL strategies is proposed by tying the selection of the step-size to their regularization parameter. It is shown that this method allows the AL algorithm to approach the performance of consensus and diffusion strategies but that it remains less stable than these other strategies.

Index Terms—Augmented Lagrangian, Arrow-Hurwicz algorithm, dual methods, diffusion strategies, consensus strategies, primal-dual methods, Lagrangian methods.

I. INTRODUCTION

DISTRIBUTED estimation is the task of estimating and tracking slowly drifting parameters by a network of agents, based solely on local interactions. In this work, we focus on distributed strategies that enable continuous adaptation and learning from streaming data by relying on stochastic gradient updates that employ constant step-sizes. The resulting networks become adaptive in nature, which means that the effect of gradient noise never dies out and seeps into the operation of the algorithms. For this reason, the design of such networks requires careful analysis in order to assess performance and provide convergence guarantees.

Many efficient algorithms have already been proposed in the literature for inference over networks (11)–(18) such as consensus strategies (12)–(15) and diffusion strategies (5)–(11). These strategies belong to the class of primal optimization techniques since they rely on estimating and propagating the primal variable. Previous studies have shown that sufficiently small step-sizes enable these strategies to learn well and in a stable manner. Explicit conditions on the step-size parameters for mean-square stability, as well as closed-form expressions for their steady-state mean-square-error performance already exist (see, e.g., (7), (19) and the many references therein).

Besides primal methods, in the broad optimization literature, there is a second formidable class of techniques known as primal-dual methods such as the Arrow-Hurwicz (AH) method (20), (21) and the augmented Lagrangian (AL) method (21), (22). These methods rely on propagating two sets of variables: the primal variable and a dual variable. The main advantage relative to primal methods is their ability to enhance performance for ill-conditioned problems.

In contrast to existing useful studies on primal-dual algorithms (e.g., (23), (24)), we shall examine this class of strategies in the context of adaptive networks, where the optimization problem is not necessarily static anymore (i.e., its minimizer can drift with time) and where the exact form of the cost function is not even known. To do so, we will need to develop distributed variants that can learn directly and continuously from streaming data when the statistical distribution of the data is unknown. It turns out that under these conditions, the dual function cannot be determined explicitly any longer, and, consequently, the conventional computation of the optimal primal and dual variables cannot assume knowledge of the dual function. We will address this difficulty by employing constant step-size adaptation and instantaneous data measurements to approximate the search directions. When this is done, the operation of the resulting algorithm becomes influenced by gradient noise, which measures the difference between the desired search direction and its approximation. This complication alters the dynamics of primal-dual techniques in non-trivial ways and leads to some surprising patterns of behavior in comparison to primal techniques. Before we comment on these findings and their implications, we remark that the stochastic-gradient versions that we develop in this work can be regarded as a first-order variation of the useful algorithm studied in (23, p. 356) with one key difference; this reference assumes that the cost functions are known exactly to the agents and that, therefore, the dual function can be determined explicitly. In contrast, we cannot make this assumption in the adaptive context.

One of the main discoveries in this article is that adaptive primal-dual strategies turn out to have a smaller stability range and degraded performance in comparison to consensus and diffusion strategies. This result implies that AH and AL techniques are not as reliable for adaptation and learning from streaming data as the primal versions that are based on consensus and diffusion constructions. As explained further ahead, one main reason for this anomaly is that the distributed AH and AL strategies enlarge the state dimension of the network in comparison to primal strategies and they exhibit an asymmetry in their update relations; this asymmetry can cause...
an unstable growth in the state of the respective networks. In other words, the advantages of AH and AL techniques that are well-known in the static optimization context do not necessarily carry over to the stochastic context.

A second important conclusion relates to the behavior of AH and AL strategies under the partial observation model. This model refers to the situation in which some agents may not be able to estimate the unknown parameter on their own, whereas the aggregate information from across the entire network is sufficient for the recovery of the unknown vector through local cooperation. We discover that the AH strategy can fail under this condition, i.e., the AH network can fail to recover the unknown and become unstable even though the network has sufficient information to allow the agents to arrive at the unknown. This is a surprising conclusion and we illustrate it analytically by means of an example as well. In comparison, we show that the AL, consensus, and diffusion strategies are able to recover the unknown under the partial observation model.

We further discover that the stability range for the AL strategy depends on two factors: the size of its regularization parameter and the network topology. This means that even if all individual agents are stable and able to solve the inference task on their own, the condition for stability of the AL strategy will still depend on how these agents are connected to each other. A similar behavior is exhibited by consensus networks [25]. This property is a disadvantage in relation to diffusion strategies whose stability ranges have been shown to be independent of the network topology [19], [25].

We also examine the steady-state mean-square-deviation (MSD) of the primal-dual adaptive strategies and discover that the Arrow-Hurwicz method achieves the same MSD performance as non-cooperative processing. This is a disappointing property since the algorithm employs cooperation, and yet the agents are not able to achieve better performance. On the other hand, the augmented Lagrangian algorithm improves on the performance of non-cooperative processing, and can be made to approach the performance that other distributed algorithms can achieve with reasonable parameter values.

**Notation:** Random quantities are denoted in boldface. The notation \( \otimes \) represents the Kronecker product while the notation \( \otimes_b \) denotes the Tracy-Singh block Kronecker product [26]. Throughout the manuscript, all vectors are column vectors with the exception of the regression vector \( u \), which is a row vector. Matrices are denoted in capital letters, while vectors and scalars are denoted in lowercase letters. Network variables that aggregate variables across the network are denoted in calligraphic letters.

### II. Adaptive Primal Strategies

In this section, we describe the problem formulation and review the two main primal techniques: diffusion and consensus for later user. Thus, consider a connected network of \( N \) agents that wish to estimate a real \( M \times 1 \) parameter vector \( w^o \) in a distributed manner. Each agent \( k = 1, 2, \ldots, N \) has access to real scalar observations \( d_k(i) \) and zero-mean real \( 1 \times M \) regression vectors \( u_{k,i} \) that are assumed to be related via the model:

\[
d_k(i) = u_{k,i} w^o + v_k(i)
\]

where \( v_k(i) \) is zero-mean real scalar random noise, and \( i \) is the time index. Models of the form (1) arise in many useful contexts such as in applications involving channel estimation, target tracking, equalization, beamforming, and localization [27]–[29]. We denote the second-order moments by

\[
R_{u,k} = \mathbb{E} u_{k,i}^T u_{k,i} \quad (2)
\]

\[
r_{du,k} = \mathbb{E} u_{k,i} d_k(i) \quad (3)
\]

\[
\sigma^2_{v,k} = \mathbb{E} v_k^2(i) \quad (4)
\]

and assume that the regression and noise processes are each temporally and spatially white. We also assume that \( u_{k,i} \) and \( v_k(j) \) are independent of each other for all \( k, \ell \) and \( i, j \). We allow for the possibility that some individual covariance matrices, \( R_{u,k} \), are singular but assume that the sum of all covariance matrices across the agents is positive-definite:

\[
\sum_{k=1}^{N} R_{u,k} > 0 \quad (5)
\]

This situation corresponds to the partial observation scenario where some of the agents may not be able to solve the estimation problem on their own, and must instead cooperate with other nodes in order to estimate \( w^o \).

To determine \( w^o \), we consider an optimization problem involving an aggregate mean-square-error cost function:

\[
\min_{w} \frac{1}{2} \sum_{k=1}^{N} \mathbb{E} (d_k(i) - u_{k,i} w)^2 \quad (6)
\]

It is straightforward to verify that \( w^o \) from (1) is the unique minimizer of (6). Several useful algorithms have been proposed in the literature to solve (6) in a distributed manner. We are particularly interested in adaptive algorithms that operate on streaming data and do not require knowledge of the underlying signal statistics. Some of the more prominent methods in this regard are consensus-type strategies [12]–[15] and diffusion strategies [5], [6], [19], [28], [29]. The latter class has been shown in [19], [25], [28] to have superior mean-square-error and stability properties when constant step-sizes are used to enable continuous adaptation and learning, which is the main focus of this work. There are several variations of the diffusion strategy. It is sufficient to focus on the adapt-then-combine (ATC) version due to its enhanced performance; its update equations take the following form.

**Algorithm 1 Diffusion Strategy (ATC)**

\[
\psi_{k,i} = \psi_{k,i-1} + \mu u_{k,i}^T (d_k(i) - u_{k,i} \psi_{k,i-1}) \quad (7a)
\]

\[
w_{k,i} = \sum_{\ell \in N_k} \alpha_{\ell,k} \psi_{\ell,i} \quad (7b)
\]
where $\mu > 0$ is a small step-size parameter and $N_k$ denotes the neighborhood of agent $k$. Moreover, the coefficients $\{a_{\ell k}\}$ that comprise the matrix $A$ are non-negative convex combination coefficients that satisfy the conditions:

$$a_{\ell k} \geq 0, \quad \sum_{\ell \in N_k} a_{\ell k} = 1, \quad a_{\ell k} = 0 \quad \text{if} \quad \ell \notin N_k \quad (8)$$

In other words, the matrix $A$ is left-stochastic and satisfies $A^T 1_N = 1_N$. In (7a)–(7b), each agent $k$ first updates its estimate $\tilde{w}_{k,i}^{(t-1)}$ to an intermediate value by using its sensed data $\{d_k(i), u_{k,i}\}$ through (7a), and subsequently aggregates the information from the neighbors through (7b). In comparison, the update equations for the consensus strategy take the following form.

**Algorithm 2 Consensus Strategy**

$$\phi_{k,i}^{(t)} = \sum_{\ell \in N_k} a_{\ell k} w_{\ell,i}^{(t-1)} \quad (9a)$$

$$w_{k,i} = \phi_{k,i}^{(t)} + \mu u_{k,i}^T (d_k(i) - u_{k,i} w_{k,i}^{(t-1)}) \quad (9b)$$

It is important to note the asymmetry in the update (9b) with both $\{\phi_{k,i}^{(t)}, w_{k,i}^{(t-1)}\}$ appearing on the right-hand side of (9b), while the same state variable $w_{k,i}^{(t-1)}$ appears on the right-hand side of the diffusion strategy (7a). This asymmetry has been shown to be a source for instability in consensus-based solutions [19], [25], [28].

A connected network is said to be strongly-connected when at least one $a_{kk}$ is strictly positive; i.e., there exists at least one agent with a self-loop, which is a reasonable condition since it means that at least one agent in the network should have some trust in its own data. For such networks, when mean-square-error stability is ensured by using sufficiently small step-size parameters, the steady-state deviation (MSD) of the consensus and diffusion strategies can be shown to match to first-order in $\mu$ [19]. Specifically, if we let $p$ denote the Perron-vector of $A$ that is associated with the eigenvalue at one, and normalize its entries to add up to one, then the MSD is given by:

$$\text{MSD} = \lim_{i \to \infty} \frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \|\tilde{w}_k(i) - w_k(i)\|^2$$

$$= \frac{\mu}{2} \text{Tr} \left[ \left( \sum_{k=1}^{N} p_k R_{u,k} \right)^{-1} \left( \sum_{k=1}^{N} p_k^2 \sigma_{v,k}^2 R_{u,k} \right) \right] + O(\mu^2)$$

(10a)

(10b)

In particular, when $A$ is doubly-stochastic so that $p = \frac{1}{N} 1_N$, it holds that:

$$\text{MSD} = \frac{\mu}{2N} \text{Tr} \left[ \left( \sum_{k=1}^{N} R_{u,k} \right)^{-1} \left( \sum_{k=1}^{N} \sigma_{v,k}^2 R_{u,k} \right) \right] + O(\mu^2)$$

(11a)

It can be further shown [7], [19] that these strategies are able to equalize the MSD performance across the individual agents. Specifically, if we let $\tilde{w}_{k,i} = w_0 - w_{k,i}$, and define the individual MSD values as

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

(12)

then it holds that

$$\text{MSD}_k \doteq \text{MSD}$$

(13)

where the notation $a \doteq b$ means that the quantities $a$ and $b$ agree to first-order in $\mu$. This is a useful conclusion and it shows that the consensus and diffusion strategies are able to drive the estimates at the individual agents towards agreement within $O(\mu)$ from the desired solution $w_0$ in the mean-square-error sense. Furthermore, it can be shown that the diffusion strategy (7a)–(7b) is guaranteed to converge in the mean for any connected network topology, as long as the agents are individually mean stable, i.e., whenever

$$0 < \mu < \min_{1 \leq k \leq N} \left\{ \frac{2}{\lambda_{\max}(R_{u,k})} \right\}$$

(14)

In contrast, consensus implementations can become unstable for some topologies even if all individual agents are mean stable [19], [25].

**III. ADAPTIVE PRIMAL–DUAL STRATEGIES**

There is an alternative method to encourage agreement among agents by explicitly introducing equality constraints into the problem formulation (6) — see (15a)–(15b) below. We are going to examine this alternative optimization problem and derive distributed variants for it. Interestingly, it will turn out that while primal-dual techniques generally perform well in the context of deterministic optimization problems, their performance is nevertheless degraded in nontrivial ways when approximation steps become necessary. One notable conclusion that follows from the subsequent analysis is that, strictly speaking, there is no need to incorporate explicit equality constraints into the problem formulation as in (15b). This is because this step ends up limiting the learning ability of the agents in comparison to the primal (consensus and diffusion) strategies. The analysis will clarify these statements.

To motivate the adaptive primal dual strategy, we start by replacing (6) by the following equivalent constrained optimization problem where the variable $w$ is replaced by $w_k$:

$$\min_w \frac{1}{2N} \sum_{k=1}^{N} \mathbb{E} (d_k(i) - u_{k,i} w_k)^2$$

(15a)

s.t. $w_1 = w_2 = \cdots = w_N$ (15b)

The following definitions are useful [29], [30].

**Definition 1** (Incidence matrix of an undirected graph). Given a graph $G$, the incidence matrix $C = [c_{ek}]$ is an $E \times N$ matrix, where $E$ is the total number of edges in the graph and $N$ is the total number of nodes, with entries defined as follows:

$$c_{ek} = \begin{cases} 
+1, & k \text{ is the lower indexed node connected to } e \\
-1, & k \text{ is the higher indexed node connected to } e \\
0, & \text{otherwise} 
\end{cases}$$

(16)
Thus, $C^TN = 0_E$. Self-loops are excluded.

**Definition 2** (Laplacian matrix of a graph). Given a graph $G$, the Laplacian matrix $L = [l_{k\ell}]$ is an $N \times N$ matrix whose entries are defined as follows:

$$l_{k\ell} = \begin{cases} |N_k| - 1, & k = \ell \\ -1, & k \neq \ell, \ell \in N_k \\ 0, & \text{otherwise} \end{cases}$$

(17)

where $|N_k|$ denotes the number of neighbors of agent $k$. It holds that $L = C^TC$, where $C$ is the incidence matrix of the graph.

Since there must exist at least $N - 1$ edges to connect $N$ nodes when the graph is connected, the number of edges in the graph satisfies $E \geq N - 1$. To illustrate the above definitions, consider the network illustrated in Fig. 1.

![Fig. 1. Example of a connected network with $N = 6$ nodes and $E = 8$ edges.](image)

The corresponding incidence and Laplacian matrices are given by

$$C = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

(18)

$$L = \begin{bmatrix} 4 & -1 & -1 & 0 & -1 & -1 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ -1 & 0 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$

(19)

Note that each node $k$ has access to the $k$-th row and column of the Laplacian matrix since they are aware of their local network connections. In addition, each node $k$ has access to row $e$ of the incidence matrix for which $c_{ek} \neq 0$.

Since the network is connected, it is possible to rewrite (15a)-(15b) as

$$\min_w \frac{1}{2} \sum_{k=1}^{N} \mathbb{E}(d_k(i) - u_{k,i}w_k)^2 \quad \text{s.t.} \quad \{w_k = w_\ell, \ell \in N_k, \ell > k\}$$

(20a)

or, in terms of the incidence matrix:

$$\min_w \frac{1}{2} \sum_{k=1}^{N} \mathbb{E}(d_k(i) - u_{k,i}w_k)^2 \quad \text{s.t.} \quad Cw = 0_{MN}$$

(21a)

(21b)

where we introduced the extended quantities:

$$C \triangleq C \otimes I_M$$

(22)

$$w \triangleq \text{col}\{w_1, \ldots, w_N\}$$

(23)

The Lagrangian of the constrained problem (21a)-(21b) is given by [21], [31]:

$$f_1(w, \lambda) = \frac{1}{2} \sum_{k=1}^{N} \mathbb{E}(d_k(i) - u_{k,i}w_k)^2 + \lambda^T Cw$$

(24)

where $\lambda \in \mathbb{R}^{EM \times 1}$ is the Lagrange multiplier vector: it consists of $E$ subvectors, $\lambda = \text{col}\{\lambda_e\}$, each of size $M \times 1$ for $e = 1, 2, \ldots, E$. One subvector $\lambda_e$ is associated with each edge $e$. The dual function $g_1(\lambda)$ associated with (24) is found by minimizing $f_1(w, \lambda)$ over the primal variable $w$:

$$g_1(\lambda) = \min_w f_1(w, \lambda)$$

(25)

where the dual function $g_1(\lambda)$ is always concave regardless of the convexity of the primal problem [31, p. 216] and a dual variable $\lambda^o$ is obtained by maximizing $g_1(\lambda)$ over $\lambda$:

$$\lambda^o \triangleq \arg \max_{\lambda} g_1(\lambda)$$

(26)

where $\lambda^o$ may not be unique unless $g_1(\lambda)$ is strongly concave.

The augmented Lagrangian of the constrained problem (21a)-(21b) is given by [21], [31]:

$$f(w, \lambda) = \frac{1}{2} \sum_{k=1}^{N} \mathbb{E}(d_k(i) - u_{k,i}w_k)^2 + \lambda^T Cw + \frac{\eta}{2} \|Cw\|^2$$

$$= \frac{1}{2} \sum_{k=1}^{N} \mathbb{E}(d_k(i) - u_{k,i}w_k)^2 + \lambda^T Cw + \frac{\eta}{2} \|w\|_2^2$$

(27)

Moreover,

$$L \triangleq L \otimes I_M = C^TC$$

(28)

and $\eta$ is a non-negative regularization parameter. The dual function $g(\lambda)$ associated with (27) is found by minimizing $f(w, \lambda)$ over the primal variable $w$:

$$g(\lambda) = \min_w f(w, \lambda)$$

(29)

The dual function $g(\lambda)$ is known to be always concave regardless of the convexity of the primal problem [31, p. 216]. An optimal dual variable $\lambda^o$ is found by maximizing (29) over $\lambda$:

$$\lambda^o = \arg \max_{\lambda} g(\lambda)$$

(30)

The augmentation of the Lagrangian introduces a penalty term that further encourages the nodes to reach agreement [21, p. 241]. The analysis in the paper will clarify how this
augmentation enhances performance in comparison to the non-augmented version \cite{24}; see Table \ref{table:complexity} further ahead.

It is explained in Appendix \ref{appA} that the unique solution \(w^0\) to (21a) (21b) can be determined by focusing instead on determining the saddle points \(\{w^0, \lambda^0\}\) of the augmented Lagrangian function \cite{27,22,31}. There are various methods to do so, such as relying on a gradient ascent procedure when \(g(\lambda)\) is known. This approach is the one taken in the Alternating Direction Method of Multipliers (ADMM) \cite{24} and particularly in the work \cite{23} p. 356]. However, in the setting under study, the statistical moments of the data are not known and, therefore, the expectation in \(f(w, \lambda)\) in (27) cannot be evaluated beforehand. As a result, the dual function \(g(\lambda)\) defined by (29) cannot be determined either and, consequently, we cannot rely directly on (30) to determine the optimal dual variable. We will need to follow an alternative route that relies on the use of stochastic approximations.

We search for a saddle-point of (27) by employing a stochastic approximation version of the first-order augmented Lagrangian algorithm \cite{21} pp. 240–242] \cite{22} p. 456. The implementation relies on a stochastic gradient descent step with respect to the primal variable, \(w\), and a gradient ascent step with respect to the dual variable, \(\lambda\), as follows:

\[
\begin{align*}
\psi_{k,i-1} &= w_{k,i-1} - \mu \sum_{e=1}^{E} c_{e,k} \lambda_{e,i-1} - \mu \eta \sum_{\ell \in N_k} l_{\ell k} w_{\ell,i-1} \\
\lambda_{e,i} &= \lambda_{e,i-1} + \mu (w_{k,i-1} - w_{\ell,i-1}) & [\ell > k, \ell \in N_k] 
\end{align*}
\]

When \(\eta = 0\) (i.e., when we employ (24) instead of (27)), we obtain the following distributed Arrow-Hurwicz (AH) method.

\begin{algorithm}
\caption{Distributed Arrow-Hurwicz (AH)}
\begin{align*}
\psi_{k,i-1} &= w_{k,i-1} - \mu \sum_{e=1}^{E} c_{e,k} \lambda_{e,i-1} \\
w_{k,i} &= \psi_{k,i-1} + \mu u_{k,i}^T (d_k(i) - u_{k,i} w_{k,i-1}) \\
\lambda_{e,i} &= \lambda_{e,i-1} + \mu (w_{k,i-1} - w_{\ell,i-1}) & [\ell > k, \ell \in N_k]
\end{align*}
\end{algorithm}

Algorithm 4 was also considered in \cite{32,33} for the solution of saddle point problems for other cost functions. Reference \cite{32} considers problems that arise in the context of reinforcement learning in response to target policies, while reference \cite{33} considers regret analysis problems and employs decaying step-sizes rather than continuous adaptation. As noted earlier, constant step-sizes enrich the dynamics in non-trivial ways due to the persistent presence of gradient noise. Note further that by setting \(\eta = 0\), step (34a) will end up relying solely on the dual variables and will not benefit from the neighbors’ iterates. The presence of these iterates in (34a) strongly couples the dynamics of the various steps in the distributed Lagrangian implementation. We also remark that in contrast to the earlier work \cite{15}, we do not require the availability of special bridge nodes in addition to the regular nodes. In our formulation, the focus is on networks that consist of homogeneous nodes with similar capabilities and responsibilities and without imposing constraints on the network topology.

In the above statements, either node connected to edge \(e\) can be responsible for updating \(\lambda_{e,i-1}\). At each node, the steps outlined in Algs. \ref{alg:al}–\ref{alg:ah} are singular, the difficulty is in the implementation of these steps outlined in Algs. \ref{alg:al}–\ref{alg:ah}. In Table \ref{table:complexity} we list the complexity of Algorithms \ref{alg:al}–\ref{alg:ah}. We observe that the primal-dual algorithms require more computation and communication per iterations than the primal methods (generally, by at least a factor of 2; this is because the agents in the AL and AH implementations also need to propagate the additional dual variables, \(\lambda_{e,i-1}\)).

\section{IV. PREVIEW OF RESULTS}

Before proceeding into a close examination of the properties of the adaptive dual-primal networks, we summarize in this section the main highlights to be derived in the sequel.

To begin with, it is known that even when some of the regression covariance matrices, \(R_{u,k}\), are singular, the diffusion strategy (7a–7b) and the consensus strategy (9a–9b) are still able to estimate \(w^0\) through the collaborative process among the agents \cite{19}. We will verify in the sequel that the AL algorithm (34a–34e) can also converge in this case (see Theorems \ref{thm:al} and \ref{thm:ah}), while the AH algorithm (35a–35e) need not converge (see Appendix \ref{appC}). Further differences among the algorithms arise in relation to how stable they are and how close their iterates get to the desired \(w^0\) when they do converge. We will see, for instance, that the AH and AL
Discretizations of the above update equations, i.e., (7a)–(7b), (9a)–(9b), or even (34a)–(34c), (35a)–(35c), or even
AL algorithms, even though they will converge under the non-cooperative solution (see Example 1). A similar
improvement is generally required: the network topology and is inversely proportional to \( \eta \) in the case of the
AL algorithm (see Theorem 3). What these results mean is that connected agents can fail to converge via the AH and
AL algorithms continues to be smaller even in implementations. Diffusion strategies, on the other hand, do not have this asymmetry and will remain stable regardless of
implemented that the AH algorithm achieves the same performance level as the non-cooperative algorithm (see Corollary 3), while the AL algorithm with \( \eta > 0 \) attains a performance level that is worse than that by the diffusion and consensus strategies by a positive additive term that decays as \( \eta \) increases.

With regards to the MSD performance (10a), we will show that the AH algorithm achieves the same performance level as the non-cooperative algorithm (see Corollary 3), while the AL algorithm with \( \eta > 0 \) attains a performance level that is worse than that by the diffusion and consensus strategies by a positive additive term that decays as \( \eta \) increases. The above observations and results are summarized in Table II. The stability ranges and steady-state MSD values are under
observation was proven earlier in [25] for consensus strategies: the network can become unstable even if all individual agents
are stable. The reason for this behavior is the asymmetry noted earlier in the consensus update (9b); this asymmetry can lead to an unstable growth in the state of consensus networks. We observe from (34b) and (35b) that a similar asymmetry exists in the update equations for AL and AH implementations. Diffusion strategies, on the other hand, do not have this asymmetry and will remain stable regardless of
the topology [19].

V. COUPLED ERROR RECURSION
We move on to establish the above results by carrying out a detailed mean-square-error analysis of the algorithms.
A. Error Quantities

We know that the optimizer of (21a)–(21b) is \( \tilde{w} = w^o \) for all \( k = 1, \ldots, N \), where \( w^o \) was defined in (1) since (21a)–(21b) is equivalent to (6). We introduce the error vector at each agent \( k \):

\[
\tilde{w}_{k,i} = w^o - w_{k,i}
\]

and collect all errors from across the network into the block column vector:

\[
\tilde{\mathbf{w}}_i = \text{col}\{\tilde{w}_{1,i}, \tilde{w}_{2,i}, \ldots, \tilde{w}_{N,i}\}
\]

We next subtract \( w^o \) from both sides of (34a) and use (1) to find that the network error vector evolves according to the following dynamics:

\[
\tilde{\mathbf{w}}_i = \tilde{\mathbf{w}}_{i-1} + \mu \left( -\mathcal{H}_i \tilde{\mathbf{w}}_{i-1} + C^T \lambda_{i-1} + \eta \mathcal{L} \tilde{\mathbf{w}}_{i-1} \right) - \mu z_i
\]

where

\[
z_i \triangleq \text{col}\{u_{1,i}v_1(i), \ldots, u_{N,i}v_N(i)\}
\]

\[
\mathcal{H}_i \triangleq \text{blockdiag}\{u_{1,i}^Tu_{1,i}, \ldots, u_{N,i}^Tu_{N,i}\}
\]

We know from Lemma 4 in the appendix that there exists a vector \( \lambda^o \) (possibly not unique) that satisfies

\[
\nabla \mathcal{W}_f(\mathbb{1}_N \otimes w^o, \lambda^o) = 0_{NM}
\]

i.e.,

\[
\sum_{k=1}^{N} [R_{u,k}w^o - r_{du,k}] + C^T \lambda^o + \eta \mathcal{L} (\mathbb{1}_N \otimes w^o) = 0_{NM}
\]

But since

\[
\mathcal{L} (\mathbb{1}_N \otimes w^o) = (L \otimes I_M)(\mathbb{1} \otimes w^o) = L \mathbb{1} \otimes w^o = 0_{NM}
\]

we can simplify (43) to

\[
\sum_{k=1}^{N} [R_{u,k}w^o - r_{du,k}] + C^T \lambda^o = 0_{NM}
\]

Moreover, since \( w^o \) optimizes (6), we have that \( w^o \) satisfies:

\[
\sum_{k=1}^{N} [R_{u,k}w^o - r_{du,k}] = 0_{NM}
\]

and we conclude from (45) that we must have:

\[
C^T \lambda^o = 0_{NM}
\]

It is known that for a connected network, the rank of \( C \) is \( N - 1 \) [30, p. 12] and, therefore, \( C \) is rank-deficient whenever \( E > N - 1 \) (i.e., when the graph spans all nodes and is not a tree [34, pp. 37–38]). Result (47) confirms that \( \lambda^o \) is not necessarily unique.

B. Useful Eigen-Spaces

The rank-deficiency of \( C \) creates difficulties for the study of the stability and performance of the adaptive primal-dual networks. We will resort to a useful transformation that allows us to identify and ignore redundant dual variables. We start by introducing the singular-value-decomposition of \( C \):

\[
C = USV^T
\]

where \( U \in \mathbb{R}^{E \times E} \) and \( V \in \mathbb{R}^{N \times N} \) are orthogonal matrices and \( S \in \mathbb{R}^{E \times N} \) is partitioned according to

\[
S = \begin{bmatrix}
S_2 & \mathbb{0}_{N-1} \\
\mathbb{0}_{(E-N+1) \times (N-1)} & \mathbb{0}_{E-N+1}
\end{bmatrix}
\]

where the square diagonal matrix \( S_2 \in \mathbb{R}^{(N-1) \times (N-1)} \) contains the nonzero singular values of \( C \) along its main diagonal and is therefore non-singular.

Now, since \( L = C^T C \), it follows that \( S_2^T S_2 \) is a diagonal matrix containing the nonzero eigenvalues of \( L \). More explicitly,

\[
L = VDV^T
\]

where

\[
D \triangleq \begin{bmatrix}
D_1 & \mathbb{0}_{N-1} \\
\mathbb{0}_1 & \mathbb{0}_{N-1}
\end{bmatrix} = \begin{bmatrix}
S_2^T S_2 & \mathbb{0}_{N-1} \\
\mathbb{0}_{(E-N+1) \times (N-1)} & \mathbb{0}_{E-N+1}
\end{bmatrix}
\]

in terms of \( D_1 = S_2^T S_2 \). Furthermore, we know that for a connected network, the Laplacian matrix has a single eigenvalue at zero [35]. Therefore, the nullspace of \( L \) has dimension one. Using the fact that \( L \mathbb{1}_N = 0_N \), we can select the vector \( \mathbb{1}_N/\sqrt{N} \) as a normalized basis vector for this nullspace. In other words, we can partition the eigenbasis \( V \) into:

\[
V = \begin{bmatrix}
V_2 & \frac{1}{\sqrt{N}} \mathbb{1}_N
\end{bmatrix}
\]

In addition, by using the property [36, p. 141]:

\[
\text{SVD}(A \otimes B) = (U_A \otimes U_B)(\Sigma_A \otimes \Sigma_B)(V_A^T \otimes V_B^T)
\]

we can write

\[
C = USV^T
\]

where \( U = U \otimes I_M, V = V \otimes I_M, \) and \( S = S \otimes I_M \) can be partitioned as:

\[
S = \begin{bmatrix}
S_2 & \mathbb{0}_{(N-1)M \times M} \\
\mathbb{0}_{(E-N+1)M \times M} & \mathbb{0}_{(E-N+1)M \times M}
\end{bmatrix}
\]

with a nonsingular diagonal matrix \( S_2 \in \mathbb{R}^{(N-1)M \times (N-1)M} \). Likewise,

\[
V = \begin{bmatrix}
V_2 & \mathbb{0}_0
\end{bmatrix}
\]

where \( V_2 \triangleq V_2 \otimes I_M \) and \( \mathbb{0}_0 \triangleq 1_N/\sqrt{N} \otimes I_M \).
C. Dimensionality Reduction

We recall (51b) and re-write it as

$$\lambda_i = \lambda_{i-1} - \mu C (I_N \otimes u^o - w_{i-1}^o)$$  (57)

Next, we introduce the transformed vectors:

$$\lambda^o = U^T \lambda^o, \quad w^o = V^T (I_N \otimes u^o)$$  (58)

where we are using the prime notation to refer to transformed quantities. Then, relation (47) implies that

$$S^T \lambda^o = 0$$  (59)

We partition $\lambda^o$ as:

$$\lambda^o = \begin{bmatrix} \lambda_{1}^o \\ \lambda_{2}^o \end{bmatrix}$$  (60)

where $\lambda_{1}^o \in \mathbb{R}^{(N-1)M \times 1}$ is the dual variable associated with $N-1$ constraints in the network and $\lambda_{2}^o \in \mathbb{R}^{(E-N+1)M \times 1}$ are the dual variables associated with the remaining constraints. We then conclude from (59) that

$$S^T \lambda_1^o = 0_{(N-1)M}$$  (61)

Observe that while the optimal Lagrange multiplier $\lambda^o$ may not be unique (see Lemma 4), the transformed vector $\lambda_1^o$ is unique since $S_2$ is invertible. We multiply both sides of (57) from the left by $U^T$ to obtain

$$\lambda_i' = \lambda_{i-1}' - \mu S(w^o - w_{i-1}^o)$$  (62)

where $\lambda_i' \triangleq U^T \lambda_i$, $w_i' \triangleq V^T w_i$ (63)

We similarly partition $w^o$, $w_i'$, and $\lambda_i'$:

$$\lambda_i' = \begin{bmatrix} \lambda_{1,i}' \\ \lambda_{2,i}' \end{bmatrix}, \quad w^o = \begin{bmatrix} w_{1}^o \\ w_{2}^o \end{bmatrix}, \quad w_i' = \begin{bmatrix} w_{1,i}' \\ w_{2,i}' \end{bmatrix}$$  (64)

where $w_{1}^o, w_{1,i}' \in \mathbb{R}^{(N-1)M \times 1}$, $w_{2}^o, w_{2,i}' \in \mathbb{R}^{M \times 1}$, $\lambda_{1,i}' \in \mathbb{R}^{(N-1)M \times 1}$, and $\lambda_{2,i}' \in \mathbb{R}^{(E-N+1)M \times 1}$. Rewriting (62) in terms of (64), we obtain

$$\begin{bmatrix} \lambda_{1,i}' \\ \lambda_{2,i}' \end{bmatrix} = \begin{bmatrix} \lambda_{1,i-1}' \\ \lambda_{2,i-1}' \end{bmatrix} - \mu \begin{bmatrix} S_2 (w_{1}^o - w_{1,i-1}^o) \\ 0_{(E-N+1)M \times NM} \end{bmatrix}$$  (65)

We observe from (65) that $\lambda_{2,i}'$ does not change as the algorithm progresses. It is therefore sufficient to study the evolution of $\lambda_{1,i}'$ alone:

$$\lambda_{1,i}' = \lambda_{1,i-1}' - \mu S_2 (w_{1}^o - w_{1,i-1}^o)$$  (66)

We may now subtract (66) from $\lambda_{1}^o$ to obtain the error-recursion:

$$\lambda_{1,i}' = \lambda_{1,i-1}' - \mu S_2 (w_{1}^o - w_{1,i-1}^o)$$  (67)

where

$$\lambda_{1,i}' \triangleq \lambda_{1}^o - \lambda_{1,i}', \quad w_{1,i}' \triangleq w_{1}^o - w_{1,i}$$  (68)

On the other hand, since $L(I_N \otimes u^o) = 0$, we have that (39)

can be re-written as

$$\bar{w}_i = \bar{w}_{i-1} + \mu (-\mathcal{H}_i w_{i-1} + C^T \lambda_{i-1} - \eta L \bar{w}_{i-1}) - \mu z_{i-1}$$  (69)

Multiplying both sides from the left side by $V^T$, we obtain

$$\bar{w}'_i = \bar{w}'_{i-1} - \mu \left( (V^T \bar{H}_i V + \eta D)\bar{w}_{i-1} - V^T C \lambda_{i-1} - \mu z_{i-1} \right)$$  (70)

where

$$D \triangleq D \otimes I_M = \begin{bmatrix} S_2^T S_2 & 0_{(N-1)M \times M} \\ 0_{M \times (N-1)M} & 0_{M \times M} \end{bmatrix}$$  (71)

and

$$z_{i} \triangleq V^T z_{i} = \begin{bmatrix} V_2^T z_i \\ V_0^T z_i \end{bmatrix}$$  (72)

Using (51) and (56) we have:

$$V^T \bar{H}_i V + \eta D = \begin{bmatrix} V_2^T \bar{H}_i V_2 + \eta S_2^2 S_2 & V_2^T \bar{H}_i V_0 \\ V_0^T \bar{H}_i V_2 & V_0^T \bar{H}_i V_0 \end{bmatrix}$$  (73)

and using (55) we also have

$$V^T C \lambda_{i-1} = S^T \lambda_{i-1}'$$  (74)

where step (a) is due to (61).

Collecting (61), (70), and (73)–(74) in matrix form, we arrive at the following theorem for the evolution of the error dynamics of the primal-dual strategy over time.

**Theorem 1** (Error dynamics of primal-dual strategies). Let the network be connected. Then, the error dynamics of the AL (34a–34c) and AH (35a–35c) algorithms evolves over time as follows:

$$\begin{bmatrix} \bar{w}_{1,i}' \\ \bar{w}_{2,i}' \end{bmatrix} = B_i' \begin{bmatrix} \bar{w}_{1,i-1}' \\ \bar{w}_{2,i-1}' \end{bmatrix} - \mu \begin{bmatrix} V_2^T z_i \\ V_0^T z_i \end{bmatrix}$$  (75)

where

$$B_i' \triangleq I_{(2N-1)M} - \mu \mathcal{R}_i'$$  (76)

$$\mathcal{R}_i' \triangleq \begin{bmatrix} V_2^T \bar{H}_i V_2 + \eta S_2^2 S_2 & V_2^T \bar{H}_i V_0 \\ V_0^T \bar{H}_i V_2 & V_0^T \bar{H}_i V_0 \end{bmatrix}$$  (77)

and $\bar{w}_{1,i}'$ and $\bar{w}_{2,i}'$ are defined in (68), $\bar{w}_{2,i}' \triangleq w_{2}^o - w_{2,i}'$, and $\mathcal{H}_i$ is defined in (41).

It is clear from (65) that if $E[\bar{w}_{i}'] \triangleq E[\bar{w}_{1}^o - \bar{w}_{1,i}']$ converges to zero, then $E[\bar{w}_i'] \triangleq E[\bar{w}^o - \bar{w}_i]$ also converges to zero since $\bar{w}_i' = V^T \bar{w}_i$. Furthermore, it holds that, for any $NM \times NM$ positive-semidefinite real matrix $\Sigma$,

$$E\|\bar{w}_i\|_{\Sigma}^2 \leq E\|\bar{w}_i'\|_{\Sigma}^2$$  (78)

where

$$\Sigma' \triangleq V^T \Sigma V$$  (79)

Therefore, for mean-square-error analysis, it is sufficient to examine the behavior of $E\|\bar{w}_i\|_{\Sigma}^2$, knowing that we can relate it back to $E\|\bar{w}_i\|_{\Sigma}^2$ via (78).
D. Mean Error Recursion

To obtain the mean error recursion, we compute expecta-
tions of both sides of (75) and use the fact that $Ez_i = 0_{NM}$
to get

$$
\begin{bmatrix}
E\tilde{w}_{j,i}^r \\
E\tilde{w}_{2,i} \\
E\tilde{\lambda}_{1,i}
\end{bmatrix} = B' \\
\begin{bmatrix}
E\tilde{w}_{j,i-1}^r \\
E\tilde{w}_{2,i-1} \\
E\tilde{\lambda}_{1,i-1}
\end{bmatrix}
$$

(80)

where

$$
B' \triangleq E\tilde{B}' = I_{(2(N-1)M) - \mu R'}
$$

(81)

$$
R' \triangleq \begin{bmatrix}
\frac{\lambda}{N} \sum_{i=1}^{N} R_{u,k} & 0_{Mx(N-1)M} \\
0_{N-1} & 0_{N-1}
\end{bmatrix}
$$

(82)

$$
H \triangleq \text{blockdiag}\{R_{u,1}, \ldots, R_{u,N}\}
$$

(83)

We would like to determine conditions on the step-size and
regularization parameters $\{\mu, \eta\}$ to ensure asymptotic
convergence of the mean quantity $E\tilde{w}_i'$ to zero. We will examine this
question later in Sec. [VI] when we study the stability of $B'$. In the next section, we derive the mean-square-error recursion.

E. Mean-Square-Error Recursion

Different choices for $\Sigma$ in (78) allow us to evaluate different
performance metrics. For example, when the network mean-
square-deviation (MSD) is desired, we set $\Sigma = I_{MN}$. Using
(78), we shall instead examine $E$$\|\tilde{w}_i'\|_2^2$. We first lift $\Sigma'$ to

$$
\Gamma' = \begin{bmatrix}
\Sigma' & 0_{N\times(N-1)M} \\
0_{(N-1)\times N} & 0_{(N-1)\times (N-1)M}
\end{bmatrix}
$$

(84)

so that the extended model (75) can be used to evaluate
$E$$\|\tilde{w}_i'\|_2^2$, as follows:

$$
E\left[\begin{bmatrix}
\tilde{w}_{j,i} \\
\tilde{w}_{2,i} \\
\tilde{\lambda}_{1,i}
\end{bmatrix}^T \Gamma' \begin{bmatrix}
\tilde{w}_{j,i} \\
\tilde{w}_{2,i} \\
\tilde{\lambda}_{1,i}
\end{bmatrix}\right] = E\left[\begin{bmatrix}
\tilde{w}_{j,i-1} \\
\tilde{w}_{2,i-1} \\
\tilde{\lambda}_{1,i-1}
\end{bmatrix}^T \Gamma' \begin{bmatrix}
\tilde{w}_{j,i-1} \\
\tilde{w}_{2,i-1} \\
\tilde{\lambda}_{1,i-1}
\end{bmatrix}\right] + \mu^2 E\left[\begin{bmatrix}
\tilde{w}_{j,i} \\
\tilde{w}_{2,i} \\
\tilde{\lambda}_{1,i}
\end{bmatrix}^T \Gamma' \begin{bmatrix}
\tilde{w}_{j,i} \\
\tilde{w}_{2,i} \\
\tilde{\lambda}_{1,i}
\end{bmatrix}\right]
$$

(85)

Following arguments similar to [29] pp. 381–383, it can be verified that

$$
E\{B'^T \Gamma' B'\} = B'^T \Gamma' B' + O(\mu^2)
$$

(86)

We will be assuming sufficiently small step-sizes (which
correspond to a slow adaptation regime) so that we can ignore
terms that depend on higher-order powers of $\mu$. Arguments in [19] show that conclusions obtained under this approximation lead to performance results that are accurate to first-order in the step-size parameters and match well with actual performance for small step-sizes. Therefore, we approximate

$$
E\{B'^T \Gamma' B'\} \approx B'^T \Gamma' B' + O(\mu^2)
$$

(87)

which, by using properties of the Kronecker product operation,
can be rewritten in the equivalent form:

$$
E\left[\begin{bmatrix}
\tilde{w}_{j,i} \\
\tilde{w}_{2,i} \\
\tilde{\lambda}_{1,i}
\end{bmatrix}^T \gamma' \begin{bmatrix}
\tilde{w}_{j,i} \\
\tilde{w}_{2,i} \\
\tilde{\lambda}_{1,i}
\end{bmatrix}\right] = E\left[\begin{bmatrix}
\tilde{w}_{j,i-1} \\
\tilde{w}_{2,i-1} \\
\tilde{\lambda}_{1,i-1}
\end{bmatrix}^T \gamma' \begin{bmatrix}
\tilde{w}_{j,i-1} \\
\tilde{w}_{2,i-1} \\
\tilde{\lambda}_{1,i-1}
\end{bmatrix}\right] + \mu^2 (\text{bvec}(R_k^T))^T \gamma'
$$

(88)

In this second form, we are using the notation $\|x\|_2^2$ and $\|x\|^2_2$ interchangeably in terms of the weighting matrix $\Sigma$ or its
erectorized form, $\sigma = \text{bvec}(\Sigma)$, where $\text{bvec}(X)$ denotes
block vectorization [5]. In block vectorization, each $M \times M$ submatrix of $X$ is vectorized, and the vectors are stacked on
top of each other. Moreover, we are introducing

$$
\gamma' \triangleq \text{bvec}(\Gamma')
$$

(89)

$$
F' \triangleq B'^T \otimes_b B'^T
$$

(90)

$$
R_b \triangleq E\left[\begin{bmatrix}
\frac{\lambda}{N} \sum_{i=1}^{N} R_{u,k} & 0_{Mx(N-1)M} \\
0_{N-1} & 0_{N-1}
\end{bmatrix}^T \begin{bmatrix}
\frac{\lambda}{N} \sum_{i=1}^{N} R_{u,k} & 0_{Mx(N-1)M} \\
0_{N-1} & 0_{N-1}
\end{bmatrix}\right]
$$

(91)

$$
= \begin{bmatrix}
\frac{\lambda}{N} \sum_{i=1}^{N} R_{u,k} & 0_{Mx(N-1)M} \\
0_{N-1} & 0_{N-1}
\end{bmatrix} \otimes_{b} \begin{bmatrix}
\frac{\lambda}{N} \sum_{i=1}^{N} R_{u,k} & 0_{Mx(N-1)M} \\
0_{N-1} & 0_{N-1}
\end{bmatrix} \otimes_{b} \begin{bmatrix}
\lambda & 0_{Mx(N-1)M} \\
0_{N-1} & 0_{N-1}
\end{bmatrix}
$$

(92)

where $\otimes_b$ denotes the block Kronecker product [26] and

$$
R_2 \triangleq E[z_i z_i^T] = \begin{bmatrix}
\sigma_{v,1}^2 R_{u,1} & \cdots \\
\cdots & \sigma_{v,N}^2 R_{u,N}
\end{bmatrix}
$$

(93)

VI. STABILITY ANALYSIS

Using the just derived mean and mean-square update relations, we start by analyzing the stability of the algorithm [34a]–[34c]. We first review the following concepts.

**Definition 3 (Hurwitz Matrix).** A real square matrix is called
Hurwitz if all its eigenvalues possess negative real parts [37].

**Definition 4 (Stable Matrix).** A real square matrix is called
(Schur) stable if all its eigenvalues lie inside the unit circle
[37].

The following lemma relates the two concepts [21] p. 39.

**Lemma 1 (Hurwitz and stable matrices).** Let $A$ be a Hurwitz
matrix. Then, the matrix $B = I + \mu A$ is stable if, and only if,

$$
0 < \mu < \min_j \left\{ -\frac{2 \Re \lambda_j(A)}{\lambda_j(A)^2} \right\}
$$

(94)
where \( \lambda_j(A) \) is the \( j \)-th eigenvalue of the matrix \( A \).

A. Mean Stability

In order to show that \( \mathbb{E} \bar{w}_i' \rightarrow 0_{NM} \), we need to show that the matrix \( B' \) in (81) is stable. To establish this fact, we rely on Lemma [2] and on the following two auxiliary results.

**Lemma 2** (Hurwitz stability of a matrix). Let a block square matrix \( G \) have the following form:

\[
G = \begin{bmatrix} X & Y^T \\ -Y & 0_{Q \times Q} \end{bmatrix}
\]

and let the matrix \( X \in \mathbb{R}^{P \times P} \) be positive-definite and \( Y \in \mathbb{R}^{Q \times P} \) possess full row rank. Then, the matrix \( G \) is Hurwitz.

**Proof.** The argument follows a similar procedure to the proof of Proposition 4.4.2 in [22] p. 449. Let \( \lambda_j \) denote the \( j \)-th eigenvalue of \( G \) and let the corresponding eigenvector be \( g_j = [g_{1,j}, g_{2,j}]^T \neq 0_{Q+P} \) so that \( Gg_j = \lambda_j g_j \), where \( g_{1,j} \in \mathbb{C}^{P \times 1} \) and \( g_{2,j} \in \mathbb{C}^{Q \times 1} \). Then, we have that

\[
\Re \{ g_j^* Gg_j \} = \Re \{ \lambda_j g_j \cdot g_j \} = \Re \{ \lambda_j \} (\|g_{1,j}\|^2 + \|g_{2,j}\|^2)
\]

Similarly, using (95), we have that the same quantity is given by

\[
\Re \{ g_j^* Gg_j \} = \Re \{ g_j^* X g_{1,j} - g_j^* Y g_{2,j} \} = \Re \{ g_j^* X g_{1,j} \}
\]

since

\[
\Re \{ g_j^* Y g_{1,j} \} = \Re \{ g_j^* Y g_{1,j}^* \} = \Re \{ g_j^* Y g_{2,j} \}
\]

Now, combining (96)–(97), we have that

\[
\Re \{ \lambda_j \} (\|g_{1,j}\|^2 + \|g_{2,j}\|^2) = -\Re \{ g_j^* X g_{1,j} \}
\]

Since the matrix \( X \) is positive-definite, then either 1) \( \Re \{ \lambda_j \} < 0 \) and \( g_{1,j} \neq 0_P \) or 2) \( \Re \{ \lambda_j \} = 0 \) and \( g_{1,j} = 0_P \). Suppose now that \( g_{1,j} = 0_P \), then

\[
Gg_j = \lambda_j g_j \Rightarrow -Y^T g_{2,j} = g_{1,j} = 0_P
\]

but since we assumed that \( g_{1,j} \neq 0_Q + P \) while \( g_{1,j} = 0_P \), then we have that \( g_{2,j} \neq 0_Q \). This implies that \( g_{2,j} \neq 0_Q \) is in the nullspace of \( Y^T \), which is not possible since \( Y \) has full row rank. Therefore \( g_{1,j} \neq 0_P \) and we conclude that \( \Re \{ \lambda_j \} < 0 \) and thus the matrix \( G \) is Hurwitz.

We can also establish the following result regarding the positive-definiteness of \( V^T H V + \eta D \).

**Lemma 3** (Positive-definiteness of \( V^T H V + \eta D \)). Let the sum of regressor covariance matrices satisfy (5), and let the network be connected. Then, there exists \( \eta \geq 0 \) such that for all \( \eta > \bar{\eta} \), we have that the matrix \( V^T H V + \eta D \) is positive-definite.

**Proof.** First, note that

\[
V^T H V + \eta D = \begin{bmatrix} V_2^T H V_2 + \eta S_2^T S_2 & V_2^T H V_0 \\ V_0^T H V_2 & \frac{1}{N} \sum_{k=1}^{N} R_{u,k} \end{bmatrix}
\]

To show that (101) is positive-definite, it is sufficient to show that the Schur complement relative to the lower-right block is positive definite. This Schur complement is given by:

\[
\chi \triangleq \eta S_2^T S_2 + Z
\]

where we defined

\[
Z \triangleq V_2^T H V_2 - V_2^T H V_0 \left( \frac{1}{N} \sum_{k=1}^{N} R_{u,k} \right)^{-1} V_0^T H V_2
\]

Then, by Weyl’s inequality [39, p. 181], we have that the Schur complement is positive-definite when

\[
\lambda_{\min}(\eta S_2^T S_2 + Z) \geq \eta \lambda_{\min}(S_2^T S_2) + \lambda_{\min}(Z) > 0
\]

which is guaranteed when

\[
\eta > -\frac{\lambda_{\min}(Z)}{\lambda_{\min}(S_2^T S_2)}
\]

where \( \lambda_{\min}(S_2^T S_2) > 0 \) is the second-smallest eigenvalue of the Laplacian matrix (algebraic connectivity of the topology graph or Fiedler value [40]–[43]) and is positive when the network is connected.

Using the preceding lemmas, we are now ready to prove the mean-stability of algorithm (34a)–(34c) for large \( \eta > 0 \).

**Theorem 2** (Mean stability of the AL algorithm). Under (5) and over connected networks, there exists some \( \bar{\eta} \) such that for all \( \eta > \bar{\eta} \), the matrix \( B' \) is stable, i.e., \( \rho(B') < 1 \) for small enough step-sizes.

**Proof.** From Lemma 3, we have that \( V^T H V + \eta D \) is positive-definite for large \( \eta \). Using (82) and (106) we write

\[
-\mathcal{R}' = \begin{bmatrix} V^T H V + \eta D & S_2^T \\ -S_2 & 0_{(N-1)M \times M} \end{bmatrix}
\]

The matrix \( -\mathcal{R}' \) so defined is in the same form required by Lemma 2 where the bottom-left block has full-row-rank since \( S_2 \) is invertible. We conclude from Lemma 2 that \( -\mathcal{R}' \) is Hurwitz. Then, by Lemma 1, there exists some range for \( 0 < \mu < \bar{\mu} \) for which the matrix \( B' = I - \mu \mathcal{R}' \) is stable, where

\[
\bar{\mu} = \min_j \left\{ 2 \frac{\Re \{ \lambda_j (\mathcal{R}') \}^2}{|\lambda_j (\mathcal{R}')|^2} \right\}
\]

We conclude that the AL algorithm (34a)–(34c) is mean stable under partial observation conditions (i.e., when some of the \( R_{u,k} \) may be singular but the aggregate sum (5) is still positive-definite). Observe though that the result of Theorem 2 does not necessarily apply to the AH algorithm (35a)–(35c) since for that algorithm, \( \eta = 0 \), while the bound on the right-hand side of (105) can be positive (in fact, there are cases in which the matrix \( -\mathcal{R}' \) will not be Hurwitz when \( \eta = 0 \) –see Appendix C). It is nevertheless still possible to prove the stability of the AH algorithm under the more restrictive assumption of a positive-definite \( H \) (which requires all individual \( R_{u,k} \) to be positive-definite rather than only their sum as we have assumed so far in (5)).
Corollary 1 (Mean stability of the AL and AH algorithms). Let the matrix $H$ be positive-definite. Furthermore, let the network be connected. Then, the matrix $B'$ is stable, i.e., $\rho(B') < 1$ for small enough step-sizes.

Proof. Since $H$ is now assumed positive-definite, we have that $V^T H V + \eta D$ in (106) is positive-definite for any $\eta \geq 0$. We may then appeal to Lemma 2 to conclude that the matrix $-R'$ is Hurwitz, and by Lemma 1 there exists some range for $0 < \mu < \bar{\mu}$ so that $B'$ is stable, where $\bar{\mu}$ is given by (107). □

The assumption that the matrix $H$ is positive-definite is only satisfied when all regressor covariance matrices $R_{u, k}$ are positive-definite. We observe, therefore, that the AH algorithm cannot generally solve the partial observation problem in which only the sum of the covariance matrices is positive-definite and not each one individually. Furthermore, we observe that the AL algorithm may not be able to solve this problem either unless the regularizer $\eta$ is large enough.

B. Mean-Square Stability

Theorem 3 (Mean-square stability). Under the same conditions of Theorem 2 there exists some $\bar{\eta}$ such that for all $\eta > \bar{\eta}$, the matrix $F'$ is stable, i.e., $\rho(F') < 1$ for small enough step-sizes.

Proof. We know that $\rho(F') = \rho(B'^T \otimes_b B'^T) = \rho(B')^2$. Now, by Theorem 2, $\rho(B') < 1$, and we have that $\rho(F') < 1$ for small $\mu$. □

Therefore, when the step-size is sufficiently small, the AL algorithm can be guaranteed to converge in the mean and mean-square senses under partial observation for large enough $\eta$. We can similarly establish an analogous result to Corollary 1 for the AH algorithm under the more restrictive assumption of positive-definite $H$.

Corollary 2 (Mean-square stability of the AL and AH algorithms). Under the same conditions of Corollary 1 then the matrix $F'$ is stable, i.e., $\rho(F') < 1$ for small enough step-sizes.

Proof. The argument is similar to Theorem 3 by noting that $B'$ is also stable. □

Next, we will examine the step-size range (107) required for convergence, and we will see that the AL and AH algorithms are not as stable as non-cooperative and diffusion strategies.

C. Examination of Stability Range

In order to gain insight into the step-size range defined by (107), we will analyze the eigenvalues of the matrix $R'$ in the case when $H = I_N \otimes R_u$, where $R_u > 0$.

Theorem 4 (Eigenvalues of $R'$). Assuming $H = I_N \otimes R_u$ where $R_u$ is positive-definite, the $(2N - 1)M$ eigenvalues of the matrix $R'$ are given by:

$$\{\lambda_{k, \ell}(R')\} = \sigma \cup \tau$$  \hspace{1cm} (108)

where

$$\sigma \triangleq \{\lambda_k(R_u) : 1 \leq \ell \leq M\}$$  \hspace{1cm} (109)

$$\tau \triangleq \left\{\frac{1}{2}(\lambda_k(R_u) + \eta \lambda_k(L)) \pm \frac{1}{2}\sqrt{(\lambda_k(R_u) + \eta \lambda_k(L))^2 - 4\lambda_k(L)} : 1 \leq k \leq N - 1, 1 \leq \ell \leq M\right\}$$  \hspace{1cm} (110)

where $L$ is the Laplacian matrix of the network topology.

Proof. To obtain the eigenvalues of $R'$, we will solve for $\lambda_{k, \ell}$ using

$$\det(R' - \lambda_{k, \ell}I) = 0$$  \hspace{1cm} (111)

To achieve this, we call upon Schur’s determinantal formula for a block matrix [36, p. 5]:

$$\det \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \det(D)\det(A - BD^{-1}C)$$  \hspace{1cm} (112)

to note, using (106), that $\det(R' - \lambda_{k, \ell}I)$ is given by

$$\det(-\lambda_{k, \ell}I_{(N - 1)M}) \det \left((I_N \otimes R_u) + \eta D - \lambda_{k, \ell}I_{MN} \right) = \det(D)$$  \hspace{1cm} (113)

where we used (71) the fact that

$$\gamma(V^T \otimes M)(I_N \otimes R_u)(V \otimes I_M) = I_N \otimes R_u$$  \hspace{1cm} (114)

We already demonstrated in Corollary 1 that the matrix $-R'$ is Hurwitz and, therefore, $\lambda_{k, \ell} \neq 0$. We conclude that $\det(-\lambda_{k, \ell}I_{(N - 1)M}) \neq 0$ and we focus on the last term in (113). Observe that the matrix $(I_N \otimes R_u) + \eta D - \lambda_{k, \ell}I_{MN} - \frac{D}{\lambda_{k, \ell}}$ is block-diagonal, with the $k$-th block given by $R_u + \left(-\lambda_{k, \ell} + \left(\eta - \frac{1}{\lambda_{k, \ell}}\right) \lambda_k(L)\right) I_M$ since the matrix $D$ is a diagonal matrix with diagonal blocks $\lambda_k(L) \otimes I_M$, where $L$ is the Laplacian matrix of the network (see (71)). Therefore, since the smallest eigenvalue, $\lambda(N, L)$, of $L$ is zero, we obtain

$$\det \left\{(I_N \otimes R_u) - \lambda_{k, \ell}I_{MN} + \left(\eta - \frac{1}{\lambda_{k, \ell}}\right) D\right\} = \det\{R_u - \lambda_{k, \ell}I_{MN}\} \times \prod_{k = 1}^{N - 1} \det \left\{R_u + \left(-\lambda_{k, \ell} + \left(\eta - \frac{1}{\lambda_{k, \ell}}\right) \lambda_k(L)\right) I_M\right\}$$  \hspace{1cm} (115)

It follows, by setting the above to zero, that $M$ of the eigenvalues of $R'$, $\{\lambda_{1,1}, \ldots, \lambda_{1,M}\}$, are given by:

$$\lambda_{1, \ell} = \lambda_{k, \ell}(R_u)$$  \hspace{1cm} (116)

The remaining $2(N - 1)M$ eigenvalues are obtained by solving algebraic equations of the form:

$$\det \left\{\Delta + \left(-\lambda_{k, \ell} + \left(\eta - \frac{1}{\lambda_{k, \ell}}\right) \lambda_k(L)\right) I_M\right\} = 0$$  \hspace{1cm} (117)
where $\Delta$ is a diagonal matrix with the eigenvalues of $R_u$ along its diagonal, since
\[
\det \left\{ R_u + -\frac{1}{\lambda_{k,t}} \lambda_k(L) I_M \right\} = \det \left\{ \Phi \Delta \Phi^T + \left( -\lambda_{k,t} + \left( \eta - \frac{1}{\lambda_{k,t}} \right) \lambda_k(L) \right) I_M \right\} = \det \left\{ \Delta + \left( -\lambda_{k,t} + \eta \lambda_k(L) \right) \Phi^T \Phi \right\}
\]
(a) or, equivalently,
\[
\Delta \lambda_{k,t} - \lambda_{k,t} + \eta \lambda_k(L) \lambda_{k,t} + \lambda_k(L) = 0 \tag{120}
\]
Solving the above quadratic equation, we obtain the remaining $2(N - 1)M$ eigenvalues by expression $\{110\}$. 

Since we have shown that $\lambda(t(R_u))$ are always eigenvalues of $\mathcal{R}'$, we have that the step-size range required for convergence of the augmented Lagrangian algorithm is always bounded above (and, hence, smaller) than the stability bound for the non-cooperative and diffusion algorithms when $\mathcal{H} = I_N \otimes R_u$ (see [14]).

Another aspect we need to consider is how the stability range \cite{107} depends on the regularization parameter $\eta$ and the network topology. It is already known that the mean stability range for diffusion strategies \cite{14} is independent of the network topology \cite{28}. In contrast, it is also known that the stability range for consensus strategies \cite{51, 52, 53} is dependent on the network topology \cite{19, 25}. We are going to see that the stability of the AL algorithm for large $\eta$ is also dependent on the network topology. Indeed, as $\eta \to \infty$, we have from \cite{110} that some of the eigenvalues of $\mathcal{R}'$, besides the ones fixed at $\lambda(t(R_u))$, will approach $\eta \lambda_k(L)$. This means that the step-size range \cite{107} required for convergence will need to satisfy:
\[
\mu < \min_{1 \leq k \leq N-1} \left\{ \frac{2}{\eta \lambda_k(L)} \right\} = \frac{2}{\eta \lambda_1(L)} \tag{121}
\]
where $\lambda_1(L)$ denotes the largest eigenvalue of $L$. Clearly, as $\eta \to \infty$, the upper-bound on the step-size approaches zero. This means that the algorithm is sensitive to both the regularization parameter $\eta$ and the topology (through $\lambda_1(L)$). Lower and upper bounds for $\lambda_1(L)$ can be derived \cite{44}. For example, when the network is fully-connected, it is known that $\lambda_1(L) = N$ and, hence, the bound \cite{121} on the step-size become:
\[
\mu < \frac{2}{\eta \cdot N} \tag{122}
\]

On the other hand, for a network of $N$ agents with maximum degree $\delta$, a lower-bound for the largest eigenvalue of $L$ is \cite{40}
\[
\lambda_1(L) \geq \frac{N}{N - 1} \delta \tag{123}
\]
and thus the step-size range becomes
\[
\mu < \frac{2}{\eta \cdot N} \cdot \delta \tag{124}
\]
This result implies that as the network size increases (in the case of a fully-connected network) or the connectivity in the network improves (\delta increases), the algorithm becomes less stable (smaller stability range for the step-size is necessary), unlike other distributed algorithms such as consensus or diffusion (Algs. \cite{112, 122}). We conclude from \cite{121}, therefore, that in this regime, the stability condition on the step-size is dependent on the network topology.

In the following example, we demonstrate a case where the convergence step-size ranges for the AL and AH algorithms are strictly smaller than \cite{14} so that the stability range for these algorithms can be smaller than non-cooperative agents.

**Example 1.** Let $M = 1$ and consider the simple 2-node network illustrated in Fig. 3

---

**Fig. 3. Network topology for Example 1.**

The Laplacian matrix is given by
\[
L = \begin{bmatrix}
+1 & -1 \\
-1 & +1
\end{bmatrix} \tag{125}
\]
The eigenvalues of $L$ are $\{0, 2\}$ and $S_2 = 1$. Let $R_u,1 = R_u,2 = 1$. We consider first the AH algorithm. It can be verified that when $\eta = 0$,
\[
\mathcal{R}' = \begin{bmatrix}
1 & 0 & -1 \\
0 & 1 & 0 \\
1 & 0 & 0
\end{bmatrix} \tag{126}
\]
and its eigenvalues are given by
\[
\lambda(\mathcal{R}') = \left\{ 1 - \frac{1}{2} + \frac{\sqrt{3}}{2} j, 1 - \frac{1}{2} - \frac{\sqrt{3}}{2} j \right\} \tag{127}
\]
so that
\[
\lambda(B') = \left\{ 1 - \mu, 1 - \frac{\mu - \sqrt{3}}{2} j, 1 - \frac{\mu + \sqrt{3}}{2} j \right\} \tag{128}
\]
Now assume every agent $k$ runs a non-cooperative algorithm of the following form independently of the other agents
\[
w_{k,i} = w_{k,i} + \mu w_{k,i}^T (d_k(i) - u_{k,i} w_{k,i-1}) \tag{129}
\]
Then, from \cite{14}, we know that a sufficient condition on the step-size in order for this non-cooperative solution and for the

---

1For a fully connected network, $L = N I_N - I_N \otimes 1_N$. Take any vector $v_i$ that is orthogonal to $1_N$, then $Lv_i = N v_i$, and $v_i$ is an eigenvector of $L$ with eigenvalue $N$. There are $N - 1$ linearly independent such vectors, and one vector in the nullspace of $L$ ($v_0 = 1_N$).
diffusion strategy to be mean stable is $0 < \mu < 2$. If we select \( \mu = 3/2 \), then
\[
\rho(B') = \max \left\{ \frac{1}{4} \sqrt{7} \frac{1}{2} \right\} = \frac{\sqrt{7}}{2} > 1 \tag{130}
\]
which implies that the AH algorithm will diverge in the mean error sense. Indeed, it can be verified from (107) that the AH algorithm is mean stable when \( \mu < 1 \). As for the AL algorithm, we may use (122) for the large \( \eta \) regime to conclude that the AL algorithm will converge when \( \mu < 1/\eta \). If we let \( \eta = 20 \), then we see that the step-size needs to satisfy \( \mu < 1/20 = 0.05 \). In Figure 2 we simulate this example for \( \eta = 20 \), and \( \mu = 1.1 \), \( \mu = 0.75 \), and \( \mu = 3/64 \approx 0.0469 \), and set the noise variance for both nodes at \( \sigma^2 = 0.1 \). The consensus algorithm is outperformed by the non-cooperative algorithm when the step-size is large, as predicted by the analysis in [25]. We observe that, even in this simple fully-connected setting, the AH and AL algorithms are less stable than the diffusion strategies. In fact, the AH and AL algorithms are less stable than the non-cooperative strategy as well. We also observe that the AH and AL algorithms do not match the steady-state MSD performance of the other distributed algorithms. We examine this issue next.

\section{Mean-Square-Error Performance}

From (88), we have in the limit that:
\[
\lim_{i \to \infty} \mathbb{E} \left[ \left\| \tilde{w}_1^T \right\|_0 (I-F') \gamma' \right] = \mu^2 (bvec(R_h))^T (I-F') \gamma' \tag{131}
\]
Now, since we are interested in the network MSD defined by (10a), we may introduce the auxiliary matrix:
\[
\Phi = \begin{bmatrix}
\frac{1}{\sqrt{N}M \times N M} & \frac{1}{(N-1)M \times N M} \\
\frac{1}{(N-1)M \times N M} & \frac{1}{(N-1)M \times N M}
\end{bmatrix}
\tag{132}
\]
and select \( \gamma' = (I-F')^{-1} bvec(\Phi) \) in (131) to obtain the following expression for the network MSD:
\[
\lim_{i \to \infty} \mathbb{E} \left[ \left\| \tilde{w}_i \right\|_0^2 I_{NM} \right] = \frac{\mu^2}{N} (bvec(R_h))^T (I-F')^{-1} bvec(\Phi) \tag{133}
\]
\[\textbf{Theorem 5} (MSD approximation for AL and AH algorithms).
Given that the matrix \( \mathcal{H} + \eta \mathcal{L} \) is positive-definite from Lemma 5 then under sufficiently small step-sizes, the network MSD (133) simplifies to:
\[
\text{MSD} = \frac{\mu}{2N} \text{Tr} \left( \mathcal{R}_z (\mathcal{H} + \eta \mathcal{L})^{-1} \right) + O(\mu^2) \tag{134}
\]
where \( \mathcal{R}_z \) is given by (93).

\textbf{Proof.} See Appendix D.

\]
Observe that the positive-definiteness of the matrix \( \mathcal{H} + \eta \mathcal{L} \) is guaranteed for the AH algorithm by assuming that \( R_{u,k} > 0 \) for all \( k \). The following special cases follow from (134).

\[\textbf{Corollary 3} (MSD performance of AL algorithm). Assume each \( R_{u,k} \) is positive-definite and the network is connected, the network MSD (134) for the AH algorithm is given by:
\[
\text{MSD} = \frac{M}{2N} \sum_{k=1}^{N} \sigma^2_{u,k} + O(\mu^2) \tag{135}
\]

\textbf{Proof.} The result follows by setting \( \eta = 0 \) in (134).

Expression (135) is actually equal to the average performance across a collection of \( N \) non-cooperative agents (see, e.g., [19], [28]). In this way, Corollary 3 is a surprising result for the AH algorithm since even with cooperation, the AH network is not able to improve over the non-cooperative mode of operation where each agent acts independently of the other agents. This conclusion does not carry over to the AL algorithm, although the following result is still not particularly encouraging for AL strategies.

\[\textbf{Corollary 4} (MSD performance of AL algorithm). Assume that the matrix \( \mathcal{H} + \eta \mathcal{L} \) is positive-definite. Then, for sufficiently small step-sizes, the network MSD (134) for the AL algorithm for large \( \eta \) simplifies to
\]
\[
\text{MSD} = \frac{\mu}{2N} \text{Tr} \left( \sum_{k=1}^{N} R_{u,k} \right)^{-1} \left( \sum_{k=1}^{N} \sigma^2_{u,k} R_{u,k} \right) + \frac{\mu}{2N \eta} \text{Tr} \left( \mathcal{R}_z \mathcal{L}^1 \right) + O \left( \frac{\mu}{N^3 \eta} + O(\mu^2) \right) \tag{136}
\]
where $L^\dagger$ denotes the pseudoinverse of $L$.

**Proof.** See Appendix E. 

By examining (136), we learn that the performance of the AL algorithm for large $\eta$ approaches the performance of the diffusion strategy given by (11a). However, recalling the fact that the step-size range required for convergence, under the large $\eta$ regime and the assumption that $H = I_N \otimes R_u$ in (121), is inversely proportional to $\eta$, we conclude that the AL algorithm only can approach the performance of the diffusion strategy as $\eta \to 0$ and $\eta \to \infty$. In addition, the performance of the AL algorithm depends explicitly on the network topology through the Laplacian matrix $L$. Observe that this is not the case in (11a) for the consensus and diffusion strategies. For this reason, even in the large $\eta$ regime, the AL algorithm is not robust to the topology. We specialize Corollary 4 for the special case were $H = I_N \otimes R_u$.

**Corollary 5** (MSD performance of AL algorithm for uniform regression vectors). Let $H = I \otimes R_u$ and $R_u > 0$. Then, $R_z = R_u \otimes R_u$, where $R_u = \text{diag}\{\sigma_{v,1}^2, \ldots, \sigma_{v,N}^2\}$, and the network MSD (134) for the AL algorithm for large $\eta$ becomes:

$$\text{MSD} \approx \frac{\mu}{2N} \left[ \frac{M}{N} \sum_{k=1}^{N} \sigma_{v,k}^2 + \frac{\text{Tr}(R_u)}{\eta} \text{Tr}(R_u L^\dagger) \right]$$

**Proof.** By making the appropriate substitutions into (178) and noting that $V_0^T H L^\dagger \approx 0$ as well as $L^\dagger H V_0 = 0$ when $H = I_N \otimes R_u$, we arrive at the result. 

Finally, we can obtain the following approximation for the completely homogeneous network.

**Corollary 6** (MSD performance of AL algorithm for homogeneous networks). Let the network be completely homogeneous, i.e., $R_u = \sigma_u^2 I_N$ and $H = I \otimes R_u$, where $R_u$ is positive-definite. Then, the network MSD (134) for the AL algorithm for large $\eta$ becomes:

$$\text{MSD} \approx \frac{\mu \sigma_u^2}{2N} \left[ M + \frac{1}{\eta} \text{Tr}(R_u) \cdot \text{Tr}(L^\dagger) \right]$$

We observe that the AL algorithm can obtain in this case an $N$-fold improvement in MSD as $\eta \to \infty$ in comparison to the performance of the AH algorithm (cf. (135)). However, for every fixed $\mu$, we see that the performance of the AL algorithm will be above that of the diffusion and consensus algorithms (even for very small $\mu$—see (11a) and Table I).

In order to illustrate these results, we consider a connected network with $N = 100$ agents and set $\mu = 1 \times 10^{-4}$ and $M = 5$. First, we illustrate the network MSD (134) as a function of $\eta$ when $H = I_N \otimes R_u$ and $R_u$ is positive-definite. We observe from Fig. 4(a) that the MSD performance of the AH algorithm is identical to that of the non-cooperative solution. In addition, the AL algorithm only approaches the same steady-state performance as the diffusion strategy asymptotically as $\eta \to \infty$. Furthermore, we graph the characteristic curves for various strategies in Fig. 4(b). In this figure, we plot the convergence rate (measured by the spectral radius of $B'$ for the AL algorithm) against the MSD (134). For the other algorithms, we formulate their error recursions as:

$$\mathbb{E} \tilde{w}_{i,\text{cons}} = \mathbb{E} w_{i-1,\text{cons}}$$
$$\mathbb{E} \tilde{w}_{i,\text{diffusion}} = \mathbb{E} w_{i-1,\text{diff}}$$
$$\mathbb{E} \tilde{w}_{i,\text{non-coop}} = \mathbb{E} w_{i-1,\text{non-coop}}$$

and use the spectral radii of $B_{\text{cons}}$, $B_{\text{diffusion}}$, and $B_{\text{non-coop}}$ to measure the convergence rate and their theoretical MSD values from prior literature [7], [19], [28]. Clearly, curves closer to the bottom-left corner indicate better performance since in that case an algorithm converges quicker and has better steady-state performance. We observe that the AH algorithm is outperformed by non-cooperation. In addition, as $\eta$ increases,
we see that the AL algorithm can approach the performance of the diffusion strategy for very small \(\mu\) (slow convergence rate—bottom-right part of the plot), but not anywhere else. We also observe the effect of (121) where the AL algorithm with \(\eta = 10\) is less stable than the other distributed algorithms.

VIII. IMPROVING THE MSD PERFORMANCE

In this section, we set the step-size, \(\mu\), as a function of the regularization parameter \(\eta\). More specifically, we choose:

\[ \eta = \mu^{-1/\theta}, \quad [\theta > 1] \]  

(142)

The AL algorithm \((34a)-(34c)\) then leads to:

\[
\begin{align*}
\mathbf{w}_i &= (I - \mu^{1-\theta} \mathcal{L}) \mathbf{w}_{i-1} - \mu \mathbf{h}_i - \mu \mathbf{c}^T \lambda_{i-1} \\
\lambda_i &= \lambda_{i-1} + \mu \mathbf{c} \mathbf{w}_{i-1}
\end{align*}
\]  

(143a)-(143b)

In the special case when \(\theta = 1\) (which we exclude), we obtain

\[
\begin{align*}
\mathbf{w}_i &= (I - \mathcal{L}) \mathbf{w}_{i-1} - \mu \mathbf{h}_i - \mu \mathbf{c}^T \lambda_{i-1} \\
\lambda_i &= \lambda_{i-1} + \mu \mathbf{c} \mathbf{w}_{i-1}
\end{align*}
\]  

(144a)-(144b)

If it were not for the dual variable, the above algorithm is reminiscent of consensus-type strategies (9a)-(9b). In that case, the combination matrix would be given by \(A^1 = I - \mathcal{L}\).

Now, the main difficulty in the analysis for the case when \(\mu = \eta^{-\theta}\) is that the step-size moves simultaneously with \(\eta\). Observe that in our previous results, we found that there exists some step-size range for which the algorithm is stable when \(\eta\) is fixed at a large value, and that this step-size bound depends on \(\eta\). Unfortunately, we do not know in general how the upper-bound (107) depends on \(\eta\), and so it is not clear if the algorithm can still be guaranteed to converge when \(\mu = \eta^{-\theta}\).

If, however, we assume that \(\mathcal{H} = I_N \otimes R_u\), then we can obtain the eigenvalues of the matrix \(\mathbf{R}'\) using Theorem 4 which then allows us to continue to guarantee convergence for large enough \(\eta\).

A. Convergence Condition

**Theorem 6.** Let \(\mu = \eta^{-\theta}\) and let \(\theta > 1\). Furthermore, let \(\mathcal{H} = I_N \otimes R_u\), with positive-definite \(R_u\). Then, there exists some positive \(\eta\) such that for all \(\eta > \eta\), the matrix \(\mathbf{R}'\) is stable; i.e., \(\rho(\mathbf{R}') < 1\).

**Proof.** See Appendix E. \(\blacksquare\)

Therefore, the adjusted AL algorithm can still be made to converge, at least when \(\mathcal{H} = I_N \otimes R_u\). We do not concentrate on proving it for the more general cases since we are only interested in demonstrating that in order for the AL algorithm to achieve the same steady-state performance as other distributed algorithms such as diffusion strategies, it is helpful to choose \(\eta\) in terms of \(\mu\). We will see how this change will improve the steady-state performance under small step-sizes (large \(\eta\)).

B. Steady-State Approximation

**Corollary 7.** Let \(\mu = \eta^{-\theta}\), where \(\theta > 1\). Assuming the algorithm converges (guaranteed by Theorem 6), then the network MSD (137) of the modified AL algorithm (143a)-(143b) is approximated by:

\[
\text{MSD} = \frac{\mu}{2N} \text{Tr}\left(\sum_{k=1}^{N} R_{u,k}^{-1} \sum_{k=1}^{N} \sigma_{v,k}^2 R_{u,k}\right) + O\left(\mu^{1+1/\theta}\right)
\]

(145)

for large \(\eta\), or small \(\mu\).

**Proof.** Substitute \(\eta = \mu^{-1/\theta}\) into the results of Corollary 3. \(\blacksquare\)

The drawback remains that the AL algorithm is less stable than primal-optimization techniques, such as the diffusion strategy.

IX. NUMERICAL RESULTS

Consider a network of \(N = 20\) agents and \(M = 5\). We generate a positive-definite matrix \(R_u > 0\) with eigenvalues \(1 + x_m\), where \(x_m\) is a uniform random variable. We let \(\mathcal{H} = I_N \otimes R_u\) with \(\mu = 0.01\). This value allows all algorithms to converge. The diffusion and consensus strategies utilize a doubly-stochastic matrix generated through the Metropolis rule (29). We note that the diffusion and consensus algorithms can improve their MSD performance by designing the combination matrix based on the Hastings rule (45), (46), but we assume that the nodes are noise-variance agnostic. In Fig. 5 we simulate Algorithms (14a)-(14b) and for Algorithm 3 we simulate two values of \(\eta\): 0.2 and 2. This will allow us to validate our analysis results where an increase in \(\eta\) yields improvement in the MSD (see Theorem 5). The theoretical curves are generated using (134). We observe that as \(\eta\) is increased, the performance of the AL algorithm improves, but is still worse than that of the consensus algorithm (9a)-(9b) and the diffusion strategy (73)-(76). Furthermore, as indicated by Fig. 4(b), the convergence rate of the AH algorithm is worse than that of non-cooperation, even though both algorithms...
achieve the same MSD performance. This was observed earlier in Fig. 2. It is possible to further increase $\eta$ in order to make the consensus and diffusion strategies up to first order in the step-size. Unfortunately, as $\eta$ is increased, we saw that the permissible step-size range for the algorithm to converge shrinks. We provided a “fix” for the augmented Lagrangian algorithm where we link the step-size to the regularization parameter $\eta$. With this modification, we show that the performance of the augmented Lagrangian method matches that of the diffusion and consensus strategies up to first order in the step-size. Unfortunately, this change does not remedy the fact that the step-size range for stability is still more restrictive than that of other distributed methods.

**APPENDIX A**

**NECESSARY AND SUFFICIENT CONDITIONS FOR FINDING THE OPTIMIZER OF (21a)–(21b)**

Primal-dual algorithms that solve (21a)–(21b) generally seek to find saddle-points of either the Lagrangian (24) or the augmented Lagrangian (27). To explain why finding such points yields a solution to the original optimization problem, we first recall the Karush-Kuhn-Tucker (KKT) conditions. The statement clarifies why finding the saddle-point of the Lagrangian is unique. Further, as $\eta$ is increased, the performance of the augmented Lagrangian algorithm matches that of the diffusion and consensus strategies. However, it is important to note that if $\eta$ is increased too much, the algorithm will diverge. For this reason, it is necessary to find a compromise between finding a large enough $\eta$ that the network MSD would be small, and a small enough $\eta$ to enhance the stability range.

**X. Conclusion**

In this work, we examined the performance of primal-dual methods in a stochastic setting. In particular, we analyzed the performance of a first-order Arrow-Hurwicz algorithm and a first-order augmented Lagrangian algorithm. We discovered that the performance of the Arrow-Hurwicz algorithm matches that of a non-cooperative solution and has stability limitations. We also showed that the augmented Lagrangian algorithm can asymptotically match the MSD performance of the diffusion algorithm when the regularization parameter is increased. Unfortunately, as $\eta$ is increased, we saw that the permissible step-size range for the algorithm to converge shrinks. We provided a “fix” for the augmented Lagrangian algorithm where we link the step-size to the regularization parameter $\eta$. With this modification, we show that the performance of the augmented Lagrangian method matches that of the diffusion and consensus strategies up to first order in the step-size. Unfortunately, this change does not remedy the fact that the step-size range for stability is still more restrictive than that of other distributed methods.

**Lemma 4** (KKT Optimality Conditions). Consider the following linear equality-constrained convex optimization problem

$$\min_{\mathbf{w}} \ J(\mathbf{w}) \tag{146a}$$

subject to $A\mathbf{w} = 0 \tag{146b}$

with Lagrangian function

$$f_1(\mathbf{w}, \lambda) = J(\mathbf{w}) + \lambda^T A\mathbf{w} \tag{147}$$

and dual function

$$g_1(\lambda) = \arg\min_{\mathbf{w}} f_1(\mathbf{w}, \lambda) \tag{148}$$

where $J(\mathbf{w})$ is strongly-convex and differentiable. Then, the following statements hold [47 p. 217] [22 p. 328] [37 Ch. 5]:

1) There exists a vector $\mathbf{w}^o$ that optimizes (146a)–(146b).
2) The vector $\mathbf{w}^o$ that optimizes (146a)–(146b) is unique.
3) $g_1(\lambda) \leq J(\mathbf{w}^o)$ for any vector $\lambda$.
4) There exists a vector $\lambda^o$ that maximizes $g_1(\lambda)$.
5) Strong duality holds for (146a)–(146b); i.e., if $\mathbf{w}^o$ optimizes (146a)–(146b) and $\lambda^o$ maximizes $g_1(\lambda)$, then $J(\mathbf{w}^o) = g_1(\lambda^o)$.
6) The pair $(\mathbf{w}^o, \lambda^o)$ satisfies

$$\nabla_\mathbf{w} J(\mathbf{w}^o) + A^T \lambda^o = 0 \tag{149a}$$

$$A\mathbf{w}^o = 0 \tag{149b}$$

i.e., $(\mathbf{w}^o, \lambda^o)$ is a saddle-point of the Lagrangian [22 p. 328] [37 pp. 238–243].

Conversely, suppose there exist variables $\mathbf{w}^o$ and $\lambda^o$ that simultaneously satisfy (149a)–(149b). Then $\mathbf{w}^o$ and $\lambda^o$ are primal and dual optimal; i.e., $\mathbf{w}^o$ optimizes (146a)–(146b) and $\lambda^o$ maximizes $g_1(\lambda)$.

**APPENDIX B**

**EXPLOITING PARTIAL ADJACENCY MATRIX**

The augmented Lagrangian defined by (27) is similar to, albeit different from, a form used in [23 p. 356]; the definition there relied on the use of the adjacency matrix instead of the incidence and Laplacian matrices. To clarify the connection (and differences), we start by expressing the Laplacian matrix as

$$L = D_d - G \tag{152}$$

where $D_d$ is a diagonal matrix containing the degrees of the nodes (the number of their neighbors), and $G$ is the adjacency matrix of the graph: its $(k, \ell)$–th entry is equal to one when nodes $k$ and $\ell$ are neighbors and all diagonal entries of $G$ are...
equal to one. Let \( \mathcal{G} = G \otimes I_M \) and \( \mathcal{D}_d = D_d \otimes I_M \). Then, it holds that:

\[
\mathbf{w}^T \mathbf{L} \mathbf{w} = \mathbf{w}^T \mathbf{D}_d \mathbf{w} - \mathbf{w}^T \mathbf{G} \mathbf{w} = \sum_{k=1}^{N} \sum_{\ell=1}^{N} g_{k\ell} \|w_k\|^2 - \sum_{k=1}^{N} \sum_{\ell=1}^{N} g_{k\ell} w_k w_{\ell}
\]

\[
= \frac{1}{2} \sum_{k=1}^{N} \sum_{\ell=1}^{N} g_{k\ell} \|w_k - w_{\ell}\|^2
\]

\[
= \sum_{k=1}^{N} \sum_{\ell=k+1}^{N} g_{k\ell} \|w_k - w_{\ell}\|^2
\]

(153)

and

\[
\lambda^T \mathbf{C} \mathbf{w} = \sum_{k=1}^{N} \sum_{\ell=k+1}^{N} g_{k\ell} \lambda_{\ell k}^T (w_k - w_{\ell})
\]

(154)

where \( \lambda_{\ell k} \) denotes the subvector in \( \lambda \) that is associated with the edge connecting nodes \( k \) and \( \ell \). Therefore, we can re-write (27) as

\[
f(w, \lambda) = \frac{1}{2} \sum_{\ell=1}^{N} \mathbb{E}((d_{\ell}(i) - u_{k,i} w_k)^2 + \sum_{k=1}^{N} \sum_{\ell=k+1}^{N} g_{k\ell} \lambda_{\ell k}^T (w_k - w_{\ell}) + \frac{\eta}{2} \sum_{k=1}^{N} \sum_{\ell=k+1}^{N} g_{k\ell} \|w_k - w_{\ell}\|^2
\]

One difference in the formulation (155) in relation to the form adopted in [23, p. 356] is that instead of using the entire adjacency matrix, we are only using its upper-triangular part since we assume that the nodes can order themselves relative to their neighbors (such as by using their MAC addresses). This step allows us to reduce the number of dual variables \( \{ \lambda_{\ell k} \} \). Our analysis and results will still apply if one prefers to employ the entire adjacency matrix.

**APPENDIX C**

**THE AH STRATEGY UNDER PARTIAL OBSERVATION**

In this appendix we provide an example to illustrate that the AH strategy (35a)–(35c) can become unstable under the partial observation setting when some of the individual covariance matrices are singular. Thus, consider the fully-connected three node network illustrated in Fig. 7 with the incidence matrix

![Fig. 7. Fully-connected three-node network used in App. C](image)

\[
C = \begin{bmatrix}
1 & -1 & 0 \\
1 & 0 & -1 \\
0 & 1 & -1
\end{bmatrix}
\]

(156)

and the edge connecting nodes \( k \) and \( \ell \). Therefore, we can re-write (27) as

\[
f(w, \lambda) = \frac{1}{2} \sum_{\ell=1}^{N} \mathbb{E}((d_{\ell}(i) - u_{k,i} w_k)^2 + \sum_{k=1}^{N} \sum_{\ell=k+1}^{N} g_{k\ell} \lambda_{\ell k}^T (w_k - w_{\ell}) + \frac{\eta}{2} \sum_{k=1}^{N} \sum_{\ell=k+1}^{N} g_{k\ell} \|w_k - w_{\ell}\|^2
\]

One difference in the formulation (155) in relation to the form adopted in [23, p. 356] is that instead of using the entire adjacency matrix, we are only using its upper-triangular part since we assume that the nodes can order themselves relative to their neighbors (such as by using their MAC addresses). This step allows us to reduce the number of dual variables \( \{ \lambda_{\ell k} \} \). Our analysis and results will still apply if one prefers to employ the entire adjacency matrix.

Furthermore, let

\[
R_{u,1} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad R_{u,2} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad R_{u,3} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\]

(157)

Observe that (157) satisfies the condition \( R_{u,1} + R_{u,2} + R_{u,3} > 0 \), even though each \( R_{u,k} \) is singular. Now, computing the SVD of the incidence matrix \( C \), we obtain

\[
S_2 = \begin{bmatrix}
\sqrt{3} & 0 \\
0 & \sqrt{3}
\end{bmatrix}
\]

(158)

\[
V = \begin{bmatrix}
\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\
\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}}
\end{bmatrix}
\]

(159)

Then,

\[
\mathcal{R}' = -\mathcal{V}^T \mathcal{H} \mathcal{V} = \begin{bmatrix}
\mathcal{V}^T_2 \mathcal{H} \mathcal{V}_2 & \mathcal{V}^T_2 \mathcal{H} \mathcal{V}_0 & S_2^T \\
\mathcal{V}^T_0 \mathcal{H} \mathcal{V}_2 & \frac{1}{3} \sum_{k=1}^{N} R_{u,k} 0_{3 \times 6} & 0_{3 \times 6} \\
-S_2 & 0_{6 \times 3} & 0_6
\end{bmatrix}
\]

(160)

where

\[
\mathcal{H} = \begin{bmatrix}
R_{u,1} & R_{u,2} & R_{u,3}
\end{bmatrix}
\]

(161)

It is straightforward to verify that the spectrum of \( \mathcal{R}' \) contains a purely imaginary eigenvalue at \( j \sqrt{3} \). For example, let \( v = \frac{1}{\sqrt{2}} \begin{bmatrix} 0, j, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0 \end{bmatrix}^T \), then

\[
\mathcal{R}'v = j\sqrt{3}v
\]

(162)

which implies that \( \mathcal{R}' \) is not Hurwitz. Therefore, we cannot find a positive range for the step-size \( \mu \) to guarantee convergence of the AH algorithm (35a)–(35c) according to Lemma 1.

![Fig. 6. Simulation result for Algs. 1-4 for the partial observation scenario described in App. C. The curves are averaged over 1000 experiments.](image)
even though the AL algorithm (34a)-(34c) can be guaranteed to converge for large \( \eta \) (see Theorem 2) and the diffusion and consensus strategies can also be shown to converge in this setting [7], [19]. In Figure 6 we simulate the described scenario with \( \mu = 0.02 \) and \( \sigma_{v,k} = 0.01 \) for all \( k = 1, 2, 3 \). We see that while the AH algorithm oscillates, the other algorithms converge. This result is surprising since the network is fully connected and yet the AH algorithm cannot estimate the desired parameter vector. Indeed, for this example, AH algorithm will not converge no matter how small \( \mu \) is. The curves are averaged over 1000 experiments.

**Appendix D**

**Proof of Theorem 3**

We refer to expression (133). We know that the matrix \( I - \mathcal{F}' \) is stable when the step-size is small by Theorem 3. Hence, we can write:

\[
(I - \mathcal{F}')^{-1} = I + \mathcal{F}' + (\mathcal{F}')^2 + \ldots = \sum_{n=0}^{\infty} \mathcal{B}^n \mathcal{I} = 0 \mathcal{B}^n \mathcal{T}
\]

Then, using properties of block Kronecker products [7], we have:

\[
(\text{bvec}(R_h))^T (I - \mathcal{F}')^{-1} \text{bvec}(\Phi) = \sum_{n=0}^{\infty} \text{Tr} \left( R_h \mathcal{B}^n \otimes \Phi \mathcal{B}^n \right)
\]

Now, observe that for small step-sizes we have, using (106), that

\[
\mathcal{B}^n = (I - \mu \mathcal{R}^T)^n
\]

Collecting these results we get:

\[
(\text{bvec}(R_h))^T (I - \mathcal{F}')^{-1} \text{bvec}(\Gamma) \approx \sum_{n=0}^{\infty} \text{Tr} \left( \mathcal{V}^T \mathcal{R} \mathcal{V} \mathcal{D} (I - \mathcal{F})^{-2n} \mathcal{D}^T \mathcal{V}^T \mathcal{R} \mathcal{V} \mathcal{D} \right)
\]

which leads to (134).

**Appendix E**

**Proof of Corollary 4**

Using (101) we have

\[
\mathcal{V}^T (\mathcal{H} + \eta \mathcal{L}) \mathcal{V} = \begin{bmatrix}
\mathcal{V}^T \mathcal{H} \mathcal{V} + \eta \mathcal{D}_1 & \mathcal{V}^T \mathcal{H} \mathcal{V} 1 \\
\mathcal{V}^T \mathcal{H} \mathcal{V} & \frac{1}{N} \sum_{k=1}^{N} \mathcal{R}_{u,k}
\end{bmatrix}
\]

where \( \mathcal{D}_1 \triangleq \mathcal{S}^T \mathcal{S} = \mathcal{D}_1 \otimes \mathcal{I}_M \). The invertibility of the above matrix is guaranteed by a large enough \( \eta \) (see Lemma 3). We may now use the block matrix inversion formula:

\[
\begin{bmatrix}
\mathcal{A} & \mathcal{B} \\
\mathcal{C} & \mathcal{D}
\end{bmatrix}^{-1} = \begin{bmatrix}
\mathcal{E} & -\mathcal{E} \mathcal{D}^{-1} \mathcal{C} \\
-\mathcal{D}^{-1} \mathcal{C} \mathcal{D}^{-1} & \mathcal{D}^{-1} \mathcal{C} \mathcal{D}^{-1}
\end{bmatrix}
\]

where \( \mathcal{E} = (A - \mathcal{B} \mathcal{D}^{-1} \mathcal{C})^{-1} \approx \frac{1}{\eta} \mathcal{D}^{-1} \) for large \( \eta \). Defining

\[
\mathcal{R}_u \triangleq \frac{1}{N} \sum_{k=1}^{N} \mathcal{R}_{u,k}, \quad \mathcal{R}_z \triangleq \frac{1}{N} \sum_{k=1}^{N} \mathcal{R}_{z,k}
\]

Applying (169) to (168), we obtain

\[
\mathcal{V}^T (\mathcal{H} + \eta \mathcal{L})^{-1} \mathcal{V} \approx \begin{bmatrix}
\frac{1}{\eta} \mathcal{D}_1^{-1} & -\frac{1}{\eta} \mathcal{D}_1^{-1} \mathcal{V}^T \mathcal{H} \mathcal{V} \mathcal{R}_u^{-1} \\
-\frac{1}{\eta} \mathcal{R}_u^{-1} \mathcal{V}^T \mathcal{H} \mathcal{V} \mathcal{D}_1^{-1} \mathcal{R}_u^{-1} & \frac{1}{\eta} \mathcal{D}_1^{-1} \mathcal{V}^T \mathcal{H} \mathcal{V} \mathcal{R}_u^{-1}
\end{bmatrix}
\]

We also have that:

\[
\mathcal{V}^T \mathcal{R}_z \mathcal{V} = \begin{bmatrix}
\mathcal{V}_z^T & \mathcal{V}_z^T \mathcal{R}_z \mathcal{V}_0 \end{bmatrix}
\]

\[
\mathcal{V}_z^T \mathcal{R}_z \mathcal{V}_0
\]

\[
\mathcal{R}_z \mathcal{V}_0
\]
Substituting (171)–(172) into (134), we obtain that the performance of the AL algorithm, in the large \( \eta \) regime, is given by

\[
\text{MSD} \approx \frac{\mu}{2N} \text{Tr} \left( R_z (H + \eta L)^{-1} \right) = \frac{\mu}{2N} \left[ \text{Tr} \left( R_z \bar{R}_u \right) + \frac{1}{\eta} \text{Tr} \left( R_z L^\dagger \right) + \frac{1}{\eta} \text{Tr} \left( R_z \bar{R}_u \bar{R}_u \right) V_0^\dagger L^\dagger V_0 \left( \eta L \right)^{-1} \right] - \frac{1}{\eta} \text{Tr} \left( R_z \bar{R}_u \bar{R}_u \right) V_0^\dagger \text{Tr} \left( \eta L \right) + \frac{1}{\eta} \text{Tr} \left( R_z L^\dagger V_0 \left( \eta L \right)^{-1} \right) \]

(173)

**APPENDIX F**

**PROOF OF THEOREM 6**

Observe that through Lemma [1], we have that we have a sufficient condition on \( \mu \) for stability to be satisfied:

\[
\mu = \eta^{-\theta} < \min \left\{ 2 \Re \left\{ \lambda_j \left( R' \right) \right\}, \frac{2 \Re \left\{ \tau_{\ell, k} \right\}}{\lambda_j \left( R_u \right)} \right\} \]

(174)

Now, under the assumption that \( H = I_N \otimes R_u \), the eigenvalues of the matrix \( -R' \) are given in Theorem [4]. From this result, we know that we have a sufficient condition on the step-size (or \( \eta \)) becomes:

\[
\eta^{-\theta} \leq \min \left\{ \frac{2}{\max_{1 \leq j \leq M} \lambda_j \left( R_u \right)}, \frac{2 \Re \left\{ \tau_{\ell, k} \right\}}{\lambda_j \left( R_u \right)} \right\} \]

(175)

where \( \tau_{\ell, k} \) is given by (110). Now, we verify that for large enough \( \eta \), \( \tau_{\ell, k} \) is real. To see this, observe that a sufficient condition for \( \tau_{\ell, k} \) to be real is that the term under the radical sign in (110) must be non-negative. In order to guarantee this condition, we need:

\[
(\lambda_\ell (R_u) + \eta \lambda_k (L))^2 \geq 4 \lambda_k (L) \]

(176)

for \( 1 \leq k \leq N - 1 \), which is satisfied by the sufficient condition:

\[
\eta > \tilde{\eta}_2 \triangleq \frac{2}{\min_{1 \leq k \leq N - 1} \sqrt{\lambda_k (L)}} \]

(177)

When \( \eta > \tilde{\eta}_2 \), the second condition in (175) becomes:

\[
\eta^{-\theta} < \frac{2 \tau_{\ell, k}}{(\tau_{\ell, k})^2} = \frac{2}{\tau_{\ell, k}} \quad [\eta > \tilde{\eta}_2] \]

(178)

We can now divide the set \( \tau \) into two sets \( \phi \) and \( \psi \):

\[
\tau = \phi \cup \psi
\]

\[
\phi \triangleq \left\{ \frac{1}{2} (\lambda_\ell (R_u) + \eta \lambda_k (L)), \frac{1}{2} \sqrt{(\lambda_\ell (R_u) + \eta \lambda_k (L))^2 - 4 \lambda_k (L)} : 1 \leq k < N, 1 \leq \ell \leq M \right\}
\]

(179)

\[
\psi \triangleq \left\{ \frac{1}{2} (\lambda_\ell (R_u) + \eta \lambda_k (L)), \frac{1}{2} \sqrt{(\lambda_\ell (R_u) + \eta \lambda_k (L))^2 - 4 \lambda_k (L)} : 1 \leq k < N, 1 \leq \ell \leq M \right\}
\]

(180)

We will now upper-bound the values in \( \phi \) and \( \psi \). First,

\[
\frac{1}{2} (\lambda_\ell (R_u) + \eta \lambda_k (L)) + \frac{1}{2} \sqrt{(\lambda_\ell (R_u) + \eta \lambda_k (L))^2 - 4 \lambda_k (L)} \leq \frac{1}{2} (\lambda_\ell (R_u) + \eta \lambda_k (L)) + \frac{1}{2} (\lambda_\ell (R_u) + \eta \lambda_k (L))
\]

(181)

so that

\[
\phi_{\ell, k} \leq \lambda_\ell (R_u) + \eta \lambda_k (L) \]

(182)

Next, for \( \psi \) we use the inequality \( \sqrt{x} - y \geq \sqrt{\frac{x}{2}} - \sqrt{y} \) for \( x \geq y \geq 0 \), under \( \eta \geq \eta_2 \), to obtain:

\[
\frac{1}{2} (\lambda_\ell (R_u) + \eta \lambda_k (L)) - \frac{1}{2} \sqrt{(\lambda_\ell (R_u) + \eta \lambda_k (L))^2 - 4 \lambda_k (L)} \leq \frac{1}{2} (\lambda_\ell (R_u) + \eta \lambda_k (L)) - \frac{\sqrt{7}}{4} (\lambda_\ell (R_u) + \eta \lambda_k (L)) + \frac{\sqrt{2} \lambda_k (L)}{2}
\]

\[
= 2 - \frac{\sqrt{7}}{4} (\lambda_\ell (R_u) + \eta \lambda_k (L)) + \frac{\sqrt{2}}{2} \sqrt{\lambda_k (L)} \]

(183)

so that

\[
\psi_{\ell, k} \leq 2 - \frac{\sqrt{7}}{4} (\lambda_\ell (R_u) + \eta \lambda_k (L)) + \frac{\sqrt{2}}{2} \sqrt{\lambda_k (L)} \]

(184)

Then, we obtain the sufficient condition for convergence:

\[
\eta^{-\theta} \leq \min \left\{ \frac{2}{\max_{1 \leq j \leq M} \lambda_j \left( R_u \right)}, \frac{2 \Re \left\{ \tau_{\ell, k} \right\}}{\lambda_j \left( R_u \right)} \right\} \min_{\ell, k} \left\{ \frac{1}{2 - \frac{\sqrt{7}}{4} (\lambda_\ell (R_u) + \eta \lambda_k (L)) + \frac{\sqrt{2}}{2} \sqrt{\lambda_k (L)}} \right\}
\]

(185)

Now, when \( \theta > 1 \), there exists some \( \eta > \eta_2 \) such that for all \( \eta > \eta_2 \), (185) is satisfied since the left-hand-side decreases more rapidly than the right-hand-side with \( \eta \).

**REFERENCES**

[1] L. Li and J. A. Chambers, “A new incremental affine projection-based adaptive algorithm for distributed networks,” Signal Processing, vol. 88, no. 10, pp. 2599–2603, Oct. 2008.

[2] O. N. Gharahshiran, V. Krishnamurthy, and G. Yiu, “Distributed energy-aware diffusion least mean squares: Game-theoretic learning,” IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 5, pp. 821–836, Oct. 2013.

[3] 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. Approximation Theory, vol. 153, no. 2, pp. 139–160, Aug. 2008.

[4] K. I. Tsianos and M. G. Rabbat, “Distributed strongly convex optimization,” in Proc. Allerton Conf., Allerton, IL, Oct., 2012, pp. 593–600.

[5] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” IEEE Transactions on Signal Processing, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[6] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” IEEE Trans. Signal Process., vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[7] A. H. Sayed, “Adaptation, learning, and optimization over networks,” Foundations and Trends in Machine Learning, vol. 7, no. 4-5, pp. 311–801, Jul. 2014.

[8] 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.

[9] J. Chen and A. H. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” IEEE Trans. Signal Process., vol. 60, no. 8, pp. 4289–4305, Aug. 2012.
