Asynchronous Decentralized Optimization with Constraints: Achievable Speeds of Convergence for Directed Graphs

Firooz Shahriari-Mehr and Ashkan Panahi
Chalmers University of Technology and University of Gothenburg

Abstract

We propose a novel decentralized convex optimization algorithm called ASY-DAGP, where each agent has its own distinct objective function and constraint set. Agents compute at different speeds, and their communication is delayed and directed. Employing local buffers, ASY-DAGP enhances asynchronous communication and is robust to challenging scenarios such as message failure. We validate these features by numerical experiments. By analyzing ASY-DAGP, we provide the first sublinear convergence rate for the above setup under mild assumptions. This rate depends on a novel characterization of delay profiles, which we term the delay factor. We calculate the delay factor for the well-known bounded delay profiles, providing new insights for these scenarios. Our analysis is conducted by introducing a novel approach tied to the celebrated PEP framework. Our approach does not require the design of Lyapunov functions and instead provides a novel insight into the optimization algorithms as linear systems.

1 INTRODUCTION

Consider M computational agents exchanging information over a uni-directional communication network, represented by a directed graph $\mathcal{G}$. Agents are nodes in $\mathcal{G}$ and their goal is to minimize a sum of local objective functions $f^v(\mathbf{x})$ under an intersection of local constraints S^v, while each pair (f^v, S^v) is only known to its corresponding node v in $\mathcal{G}$. This scenario describes various problems in machine learning, control engineering and signal processing, and can be written as the following decentralized, constrained optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^m} \ \frac{1}{M}\sum_{v=1}^{M} f^v(\mathbf{x}) \qquad \text{s.t.} \quad \mathbf{x}\in\bigcap_{v=1}^{M} S^v. \tag{P}$$

In the above scenario, (P) can be solved by each node transmitting messages to its out-neighbors and executing a series of local computations based on the received messages from its in-neighbors. In practice, the variation among individual agents makes them compute at different speeds. Due to inherent uncertainties, their communication may also encounter random delays or even message losses. We generally refer to these effects as *temporal distortion.*

The goal of this paper is to investigate how fast a convex instance of (P) can be solved under temporal distortion, and minimal additional assumptions, such as the smoothness of the objective terms (f^v). We also consider the general case of asynchronous algorithms. Unlike synchronous techniques, which require all agents to complete their operations before iterating, in asynchronous techniques, nodes operate uninterruptedly and merely rely on the current information available from their in-neighbors (Figure 1). In this way, they resolve much of the difficulties with the synchronous methods, such as speed limitation by the slowest node and the demand for synchronization (Hannah et al., 2018).

Asynchronous optimization is gaining increasing interest (Assran et al., 2020), but its inherent complexity renders its design and analysis significantly more complex. Little is still known about the general convergence properties of the algorithms for (P) under temporal distortion. As we avoid typical simplifying assumptions, such as strong convexity, common techniques, such as Lyapunov (potential) functions (Polyak, 1987), used in many classical papers (Nesterov, 2012; Defazio et al., 2014; Schmidt et al., 2017), also become highly complicated, in our setting. To address these difficulties, we also propose a novel proof technique.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

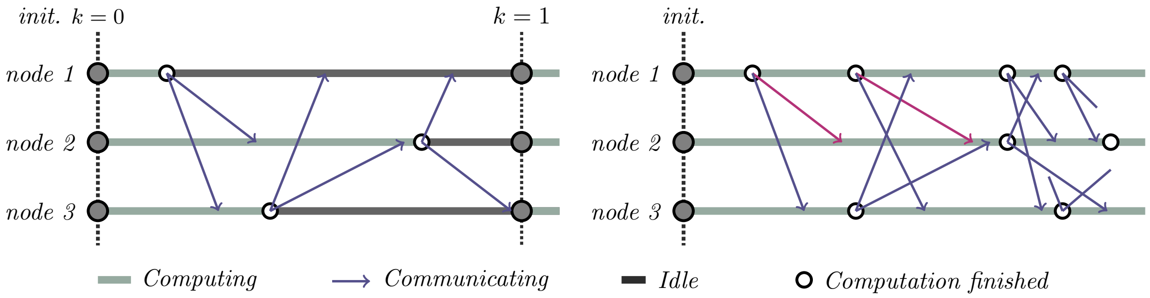


Figure 1: *Synchronous setup (left):* Agents compute and send updates, then wait for all messages delivery before starting the next iteration. *Asynchronous setup (right):* Agents compute, send updates, and immediately begin the next iteration with available information, potentially receiving multiple or no messages from neighbors.

1.1 Contributions

For the decentralized scenario in (P) with temporal distortion, we achieve novel guarantees on the speed of convergence by proposing a new algorithmic solution and a novel analysis methodology. Our algorithm is inspired by DAGP (Double Averaging Gradient projection) proposed by Shahriari-Mehr and Panahi (2022), which is, to our knowledge, the only synchronous algorithm solving (P) over directed graphs. Our analysis methodology is tied to the celebrated performance estimation program (PEP) introduced by Drori and Teboulle (2014) and Taylor (2017), but we present it as an elementary approach.

Algorithm: As we discuss in Section 3, DAGP is intolerant of asynchrony or delay. Hence, we introduce substantial modifications to it and achieve a novel algorithm that we refer to as ASY-DAGP. This is the first asynchronous algorithm handling agent-specific constraints over directed graphs, with a guaranteed convergence. Similar to DAGP, it assumes a fixed (non-vanishing) step size, which is a significant practical advantage. Our studies in Section 5 confirm ASY-DAGP's robustness to temporal distortion, owing to the inclusion of local buffers.

Theory: When (P) is convex and the objective terms are smooth, we establish that ASY-DAGP achieves an ϵ gap in consensus, optimality and feasibility in $O(1/\epsilon^2)$ iterations, under fairly general temporal distortion patterns. We require no restrictive assumption such as strong convexity. This rate is similar to the previous bounds on decentralized optimization with smooth terms, even without temporal distortion (Shahriari-Mehr and Panahi, 2022). In Section 4, we focus on a case where the agents are isolated in the sense that they weakly rely on their received messages (small gossip matrices). More general results are presented in the appendix. Although stronger isolation introduces a larger constant factor to the convergence bound, we show that it also handles larger amounts of temporal distortion, establishing a trade-off between the speed of convergence and robustness to temporal distortion. We quantify temporal distortion in terms of a novel constant called *delay factor*, which generalizes the well-known notion of bounded delays and extends our analysis to settings with unbounded delays. The relationship between the delay factor and conventional delay bounds is further clarified in Section 4.

Method: In our analysis, we avoid the difficult task of designing Lyapunov functions. Instead, we formulate and solve a meta-optimization problem, referred to as Linear-Quadratic PEP (LQ-PEP), searching for the worst bounds on convergence over the feasible trajectories of ASY-DAGP. Our approach can be interpreted as a relaxation of the PEP, which is detailed in Appendix D. Solving the LQ-PEP, we tie the behavior of ASY-DAGP to a linear system characterized by a *forward-backward, matrix-valued transfer function* through which the errors induced by temporal distortion propagates, leading to the concept of the delay factor. This analysis approach can be generalized to various other optimization problems.

2 PROBLEM SETUP

We solve the decentralized constrained optimization problem (P) in the following setting:

1. The local objective functions f^v are convex, differentiable, and $L-$smooth, with $L > 0$.
2. The local constraint sets S^v are closed and convex.
3. The optimization problem is feasible, and there exists an optimal feasible solution $\mathbf{x}^*$ satisfying the sufficient optimality condition:

$$\mathbf{0} \in \sum_{v=1}^{M} \left(\partial I_{S^v}(\mathbf{x}^*) + \nabla f^v(\mathbf{x}^*) \right). \tag{1}$$

where $\partial I_{S^v}(\mathbf{x}^*)$ is the normal cone of S^v at $\mathbf{x}^*$.

4. The communication network of the agents is represented by a fixed, connected, directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of agents, and $\mathcal{E}$ is the set of directed edges. An edge $(v, u) \in \mathcal{E}$ represents a uni-directional communication link from node u to node v. Node u's incoming and outgoing neighbors are denoted by $\mathcal{N}_{\text{in}}^u = \{v | (u, v) \in \mathcal{E}\}$ and $\mathcal{N}_{\text{out}}^u = \{v | (v, u) \in \mathcal{E}\}$, respectively. The graph includes all self-loops; that is, $u \in \mathcal{N}_{\text{in/out}}^u$. We consider two gossip matrices $\mathbf{W}$ and $\mathbf{Q}$, with the same sparsity pattern as the graph adjacency matrix. $\mathbf{W}$ has zero row sums, and $\mathbf{Q}$ has zero column sums, which can be built, for example, by the input and output graph Laplacian matrices.

5. Agents have varying computational and communication power. Their computation may take an arbitrary amount of time, and their communicated messages may experience arbitrary delays or even be lost. This phenomenon is referred to as temporal distortion.

3 ASY-DAGP ALGORITHM

ASY-DAGP is presented in algorithm 1. Similar to DAGP, it attempts to solve (1) by splitting it into

$$\sum_{v=1}^{M} \mathbf{g}^v = \mathbf{0}, \quad \mathbf{g}^v \in \partial I_{S^v}(\mathbf{x}^*) + \nabla f^v(\mathbf{x}^*). \tag{9}$$

The relation $\sum_{v=1}^{M} \mathbf{g}^v = \mathbf{0}$ is referred to as the *null condition.* At each node v, these algorithms update both a local solution $\mathbf{x}^v$ and an auxiliary variable $\mathbf{g}^v$. Both DAGP and ASY-DAGP ensure that, upon convergence, each node satisfies $\mathbf{g}^v \in \partial I_{S^v}(\mathbf{x}^v) + \nabla f^v(\mathbf{x}^v)$, all nodes reach consensus and the null condition holds (i.e., $\mathbf{x}^v = \mathbf{x}^*$ for every v); together, these conditions guarantee optimality. However, both DAGP and ASY-DAGP face a common challenge that the null condition can only be obtained through a distributed mechanism. DAGP resolves this by the *distributed null projection (DNP)* mechanism that uses another set of auxiliary variables $\mathbf{h}^v$, updated by the following rule:

$$\mathbf{h}_{k+1}^v = \mathbf{h}_k^v - \sum_{u \in \mathcal{N}_{\text{in}}^v} q_{vu}(\mathbf{h}_k^u - \mathbf{g}_k^u). \tag{10}$$

This achieves two goals: First, $\mathbf{h}_k^v$ always satisfies the null condition ($\sum_{v=1}^{M} \mathbf{h}_k^v = \mathbf{0}$), due to the properties of the gossip matrix $\mathbf{Q}$. Second, $\mathbf{h}^v$ tracks $\mathbf{g}^v$, that is upon convergence, $\mathbf{g}^v$ becomes equal to $\mathbf{h}^v$, hence similarly satisfying the null condition.

Algorithm 1 ASY-DAGP.

```
Input: Step size $\mu$, scaling parameters $\alpha$, $\rho$, $\eta$, $\gamma$, gossip matrices $\mathbf{W}$ and $\mathbf{Q}$.
Initialize $\mathbf{x}^v$ randomly. Initialize $\mathbf{g}^v$, $\mathbf{h}^v$, and $\mathbf{p}^v$ with zero vectors, $\forall v \in \mathcal{V}$.
Initialize $\mathbf{b}^{vu} = \mathbf{0}$, $\forall v, u \in \mathcal{V}$, $\mathbf{a}^{vv} = \mathbf{x}^v$, $\forall v \in \mathcal{V}$, and $\mathbf{a}^{vu} = \mathbf{0}$, $\forall v, u \in \mathcal{V}, u \neq v$.
Agents continuously receive information from their neighbors and store it in their local buffers $\mathcal{B}^{vu}$.
Agents update their variables in parallel as:
repeat
    Update $\mathbf{z}^v, \mathbf{x}^v, \mathbf{g}^v, \mathbf{p}^v$, and $\mathbf{h}^v$ variables as:
        $\mathbf{z}^v = \mathbf{x}^v - \sum_{u \in \mathcal{N}_{\text{in}}^v} w_{vu}\mathbf{a}^{vu} - \mu\left(\nabla f^v(\mathbf{x}^v) - \mathbf{g}^v\right)$   (2)
        $\mathbf{x}^v = \mathrm{P}_{S^v}\left(\mathbf{z}^v\right)$   (3)
        $\mathbf{p}^v = \mathbf{p}^v - \eta \sum_{u \in \mathcal{N}_{\text{in}}^v} q_{vu}\mathbf{b}^{vu} + \eta(\gamma - 1)\mathbf{g}^v$   (4)
        $\mathbf{g}^v = \mathbf{g}^v + \rho(\nabla f^v(\mathbf{x}^v) - \mathbf{g}^v) + \frac{\rho}{\mu}\left(\mathbf{z}^v - \mathbf{x}^v\right) + \alpha\left(\mathbf{h}^v - \mathbf{g}^v\right)$   (5)
        $\mathbf{h}^v = \gamma\mathbf{h}^v - \sum_{u \in \mathcal{N}_{\text{in}}^v} q_{vu}\mathbf{b}^{vu}$   (6)
    Send the tuple $(\mathbf{x}^v,\ \mathbf{p}^v)$ to all out-neighbors.
    Update $\mathbf{a}^{vu}, \mathbf{b}^{vu}$, $\forall u \in \mathcal{N}_{\text{in}}^v$. If the buffer $\mathcal{B}^{vu}$ is empty, reuse the existing value in memory; otherwise, compute them as:
        $\mathbf{a}^{vu} = \frac{1}{|\mathcal{B}^{vu}|} \sum_{\mathbf{x}^u \in \mathcal{B}^{vu}} \mathbf{x}^u$   (7)
        $\mathbf{b}^{vu} = \frac{1}{|\mathcal{B}^{vu}|} \sum_{\mathbf{p}^u \in \mathcal{B}^{vu}} \mathbf{p}^u$   (8)
    Clear the buffers $\mathcal{B}^{vu}$, for every $u \in \mathcal{N}_{\text{in}}^v$.
until Convergence
```

3.1 Naive idea: imitating DAGP by local buffers

Our initial idea in ASY-DAGP is to closely imitate DAGP by a buffering mechanism, but as shortly explained, it fails at critical aspects, which then leads to more fundamental modifications. In ASY-DAGP, an agent starts its next iteration as soon as it executes an iteration and broadcasts its message to its neighbors. However, the duration of an iteration can be variable, which makes DAGP infeasible. In response, we introduce local buffers $\mathcal{B}^{vu}$, to store all messages node v receives from u. After node v processes its messages, its buffer is cleared.

Although each node broadcasts the same message to all its outgoing neighbours, they receive and process the message at different iterations, due to asynchrony and delay. This leads to a variable number of messages in a buffer. When there are one or more messages in a buffer $\mathcal{B}^{vu}$, two averages, $\mathbf{a}^{vu}$ and $\mathbf{b}^{vu}$, of them are calculated by (7) and (8). If the buffers are empty, the last calculated averages are reused. ASY-DAGP substitutes the terms in DAGP involving the received messages with the buffer averages $(\mathbf{a}^{vu}, \mathbf{b}^{vu})$. This leads to (2) and (6). If the rest of the DAGP algorithm is unchanged, one may hope that, due to temporal correlation in the transmitted messages, the buffer averages $(\mathbf{a}^{vu}, \mathbf{b}^{vu})$ closely track the messages of DAGP and hence, ASY-DAGP behaves similar to DAGP.

3.2 Failure of naive idea and ASY-DAGP

The naive idea presented above fails because the DNP mechanism in (10) is sensitive to errors imposed by the temporal distortion. Since each node calculates a distinct average of its buffer, in general, for two nodes $v \neq w$, we may have $\mathbf{b}^{vu} \neq \mathbf{b}^{wu}$. A consequence of this phenomenon is that the variables $\mathbf{h}^v$ do not satisfy the null condition anymore, and the DNP strategy fails.

In response, we introduce fundamental changes to DAGP. First, we update (10) to (6) and introduce an attenuation factor γ. The effect of this change can be understood by the analysis of the fixed point of ASY-DAGP, presented in Appendix B. In short, it ensures that as ASY-DAGP approaches convergence, $\mathbf{h}^v$ eventually satisfies the null condition again, but it will not track $\mathbf{g}^v$ anymore. This problem is resolved by introducing auxiliary variables $\mathbf{p}^v$. The update rule of $\mathbf{p}^v$ in (4) is designed to ensure that upon convergence $\mathbf{h}^v$ and $\mathbf{g}^v$ become equal again. Although this change requires extra variables, $\mathbf{p}^v$, and additional computation, the communication overhead remains the same, as each node communicates two variables per iteration in both algorithms. Another advantage of ASY-DAGP is that it eliminates the DAGP requirement for $\ker(\mathbf{Q}) = \ker(\mathbf{W}^T)$, which can be difficult to achieve in practice.

4 CONVERGENCE GUARANTEES

Our analysis considers deterministic delay and asynchrony profiles. For the sake of analysis, we index the iterations of each node individually. Hence, different nodes start the k^{th} iteration at different real times. We study the evolution of a state consisting of all nodes at the same iteration index, which allows us to treat the algorithm as a synchronous procedure. Then, we need to consider a corrected delay including both actual delays and asynchrony:

Definition 1. Suppose that the message generated and transmitted at the l^{th} iteration of node u enters the buffer of node v at its k^{th} iteration. We say that this message experiences $d_k^{vu} := k - l$ *corrected delays.* If this message is missed (never arrives), we set $d_k^{vu} = \infty$.

Due to asynchrony, k can be less than l and the corrected delay can be negative (non-causal). This novel notion simplifies the indexing of iterations across asynchronous nodes and does not limit the generality of our analysis. Other indexing approaches addressing temporal distortion have been considered previously; see Section 6 for a detailed comparison. According to the above, we introduce the following definition:

Definition 2. We define the *local index sets* T_k^{vu} as the collection of all iteration numbers l, at which a message is sent from u that is employed by v at its k^{th} iteration. In other words, T_k^{vu} corresponds to the messages from u received by v during the last iteration before k, where the buffer $\mathcal{B}^{vu}$ was nonempty (since the same messages are re-used after this iteration).

Our analysis is also in terms of $\bar{\mathbf{x}}_K^v = \frac{1}{K}\sum_{k=0}^{K-1} \mathbf{x}_k^v$, which is the time-averaged solution at each node and $\bar{\mathbf{x}}_K = \frac{1}{M}\sum_{v=1}^{M} \bar{\mathbf{x}}_K^v$, which is the total average of the solutions up to time K. Throughout our analysis, we use the notation $O(\cdot)$ to denote a bound up to a universal constant. We also make the following assumption:

Assumption 1. The parameters $\mu, \gamma, \rho, \alpha$ and η, and the matrix $\mathbf{Q}$ satisfy $\mu L \in (0\ 1), \gamma \in (0\ 1)$ and $\eta, \rho, \alpha > 0$. Moreover, for eigenvalues λ_i of $\mathbf{Q}$, we take the polynomials $P_i(z) = z^2 + a_i z + b_i$, where $a_i = \eta\lambda_i - 1 + \alpha - \gamma$ and $b_i = \gamma(1-\alpha) + \eta\lambda_i(\alpha - \gamma)$, and assume that they have distinct roots that are strictly inside the unit circle of the complex plane. We refer to these roots and their inverses as *first poles of ASY-DAGP*. Note that since $\lambda = 0$ is an eigenvalue of $\mathbf{Q}$, the values $z = \gamma$ and $z = 1 - \alpha$ are first poles.

Special cases arises when α is nearly zero and $\mathbf{Q}$ has nonnegative eigenvalues, or alternatively, when $\mathbf{Q}$ has negative eigenvalues and α is bounded away from zero. This is formalized in the following proposition:

Proposition 1. Suppose that $\mu L \in (0\ 1), \gamma \in (0\ 1)$, and $\rho > 0$. Moreover, $\mathbf{Q}$ has nonnegative eigenvalues $\lambda_i \geq 0$, and we choose $\eta < \frac{2}{\|\mathbf{Q}\|_2}$, where $\|\mathbf{Q}\|_2$ denotes the induced two-norm of the matrix $\mathbf{Q}$. Then, for sufficiently small α, Assumption 1 is satisfied. In this case, the first poles are arbitrarily close to either γ or $1 - \eta\lambda_i$. Alternatively, if $\mathbf{Q}$ has negative eigenvalues and α is bounded away from zero, Assumption 1 still holds true under the same conditions.

Before introducing our general result in Section 4.2, we present a special case that illustrate the underlying intuitions. This case is related to bounded temporal distortion, for which we introduce the following intuitive assumption that simplifies the analysis but is not necessary for the general theory.

Assumption 2 (Agility)**.** For every u, v and k, the set T_k^{vu} is a singleton ($|T_k^{vu}| = 1$). A typical example is when each node processes the received message immediately and the process time is significantly smaller than the message travel time.

4.1 Results for bounded temporal distortion

We start by the case of bounded errors:

Assumption 3. For every u, v and k, we have $|d_k^{vu}| \leq d$ for some constant bound $d = 0, 1, 2, \ldots$.

We take two gossip matrices $\bar{\mathbf{W}}, \bar{\mathbf{Q}}$ such that $\|\bar{\mathbf{W}}\|_\infty = \|\bar{\mathbf{Q}}\|_\infty = 1$. Recall the induced infinity norm $\|\mathbf{A}\|_\infty$ of a matrix $\mathbf{A} = (a_{ij})$ is given by $\|\mathbf{A}\|_\infty = \max_i \sum_j |a_{ij}|$.

Our goal is to study ASY-DAGP in a limit where d grows to infinity and the gossip matrices are simultaneously re-scaled to mitigate the growth of the delay. For this, we introduce a positive, real factor ι called *isolation factor* and employ matrices $\mathbf{W} = \iota\bar{\mathbf{W}}$ and $\mathbf{Q} = \iota\bar{\mathbf{Q}}$, in ASY-DAGP. Note that reducing ι makes each node rely less on the incoming messages, thus making it more isolated. As such, this scenario naturally leads to slower convergence, but we show that it also makes the algorithm more robust against temporal distortion. In particular, we show that as the bound d grows, we may choose $\iota = O(1/d)$ and achieve convergence bounds, albeit being proportional to d.

Remark 1. Our scaling approach by ι is related to the existing works that employ delay-dependent step sizes to handle asynchrony (Koloskova et al., 2022; Mishchenko et al., 2022; Even et al., 2024; Bornstein et al., 2022). Our strategy does not require dynamically adjusted step sizes, making it more practical for decentralized implementations. This strategy is conceptually similar to the gossip step size used in the CHOCO-SGD algorithm (Koloskova et al., 2019).

According to proposition 1, we identify two distinct cases in this regime:

Case 1: In this case, $\bar{\mathbf{Q}}$ has negative eigenvalues. Then, we introduce the following assumption:

Assumption 4. We assume the hyperparameters α, ρ, and μ satisfy $0 < \rho < (1 - L\mu)(2 - \alpha)$. Moreover, there exists a strictly positive constant $\bar{\zeta}$, called *consensus factor*, satisfying the following relation:

$$\frac{\eta\rho}{\alpha}\left(\bar{\mathbf{Q}} + \bar{\mathbf{Q}}^T\right) + \left(1 + \frac{\eta\rho}{\alpha}\right)\left(\bar{\mathbf{W}} + \bar{\mathbf{W}}^T\right) \succeq \bar{\zeta}\mathbf{P}_{\mathbf{1}}^{\perp}, \quad (11)$$

where $\mathbf{P}_{\mathbf{1}}^{\perp} = \mathbf{I} - \frac{1}{M}\mathbf{1}\mathbf{1}^T$ is the orthogonal projection matrix onto the complement of the span of the all-one vector $\mathbf{1}$.

Though not apparent above, the consensus factor controls the speed at which the consensus is achieved. Note that the span of the all-one vector represents the *consensus subspace* that we aim to achieve, as it makes the solution of all nodes equal. Then, one can interpret (11) as a requirement that the left hand side matrix has large eigenvalues, except in the consensus subspace. More discussion can be found in the Appendix. Additionally, we note that in this case, Assumption 1 bounds α from below (Proposition 1) and η from above, which implies that $\bar{\zeta}$ is bounded.

Case 2: In this case, all eigenvalues of $\bar{\mathbf{Q}}$ are non-negative and we select α sufficiently small together with the following assumption:

Assumption 5. We take $\eta < \frac{2}{\|\mathbf{Q}\|_2}$, $\rho < 2(1 - L\mu)$. Moreover, we assume:

$$\eta\rho\left(\bar{\mathbf{W}} + \bar{\mathbf{W}}^T\right) \succeq \bar{\zeta}\mathbf{P}_{\mathbf{1}}^{\perp}. \quad (12)$$

For both these two cases, we have the following single theorem:

Theorem 1. *Suppose that Assumption 1 holds for $\bar{\mathbf{W}}$ and $\bar{\mathbf{Q}}$, and Assumptions 2 and 3 also hold. Moreover, either Case 1 holds together with Assumption 4, or Case 2 holds with Assumption 5. Then, for every value of the bound d, there exists a value of $\iota = O(1/d)$ such that utilizing $\mathbf{W} = \iota\bar{\mathbf{W}}$ and $\mathbf{Q} = \iota\bar{\mathbf{Q}}$ in ASY-DAGP leads to the following convergence bounds:*

$$\sum_v \mathrm{dist}^2(\bar{\mathbf{x}}_K, S_v) \leq \sum_\nu \|\bar{\mathbf{x}}_K^v - \bar{\mathbf{x}}_K\|_2^2 = O\left(\frac{dC_0}{\bar{\zeta}K}\right), \quad (13)$$

$$\sum_v f^v(\bar{\mathbf{x}}_K^v) - \sum_v f^v(\mathbf{x}^*) = O\left(\frac{C_0}{\mu K} + \sqrt{\frac{dC_0C_1}{\bar{\zeta}K}}\right), \quad (14)$$

where K indicates that all nodes have performed K iterations, C_0 is the distance of the initial state from an optimal state[1]*, and $C_1 = \sqrt{\sum_v \|\mathbf{n}^v + \nabla f^v(\mathbf{x}^*)\|^2}$ quantifies the heterogeneity of local functions, with $\nabla f^v(\mathbf{x}^*)$, and $\mathbf{n}^v \in \partial I_{S^v}(\mathbf{x}^*)$ satisfying (1).*

Theorem 1, proved in Appendix F, reflects the main ideas of our analysis. As the number of iterations K increases, the optimality and feasibility gaps diminish and the solutions at all nodes become equal, all with rate $O(1/\sqrt{K})$. Moreover, the speed explicitly depends on the delay bound d, the step size μ, consensus factor $\bar{\zeta}$, and the constants C_0, C_1, which aligns with our theoretical expectations and intuition.

Remark 2. The constant C_1 quantifies the heterogeneity of the local functions. Unlike some existing studies, we do not assume bounded heterogeneity explicitly. This is because the variables $\mathbf{g}^v$ in ASY-DAGP

[1] For exact definition, see Appendix D.

inherently implement a variance-reduction (gradient tracking) mechanism. Such an assumption-free treatment of heterogeneity is also seen in other works; e.g., (Nguyen et al., 2023; Xin et al., 2022).

4.2 General results

Our first result in Theorem 1 is limited to Assumptions 2,3, and 4. Our general result relaxes these assumptions and is instead based on the notion of delay factor, for which we first need to define the following matrices:

Definition 3. For any $z \in \mathbb{C}$, take $\boldsymbol{\Omega}(z) = \mathbf{S}(z) + \mathbf{G}\left(z^{-1}\right) + \mathbf{G}^T(z)$, where

$$\mathbf{S}(z) = (1 - L\mu)(2 - z - z^{-1})\mathbf{I} + z\mathbf{W} + z^{-1}\mathbf{W}^T \tag{15}$$

$$\mathbf{E}(z) = \left[(z - 1 + \alpha)\mathbf{I} + \eta\left(1 + \frac{\alpha}{z - \gamma}\right)\mathbf{Q}\right]^{-1} \tag{16}$$

$$\mathbf{G}(z) = \rho\Big[(z-1)\mathbf{I} + \eta\mathbf{Q}\Big]\frac{\mathbf{E}(z)}{z(z-1)}\Big[(z-1)\mathbf{I} + \eta\mathbf{W}\Big]. \tag{17}$$

Next, we replace Assumptions 4, 5 with a milder one:

Assumption 6. For every complex number $z \neq 1$ on the unit circle, it holds that $\Omega(z) \succeq \zeta\mathbf{P}_{\mathbf{1}}^{\perp}$, where the constant ζ is again referred to as the consensus factor. We refer to the roots of $\det\left(\Omega - \zeta\mathbf{P}_{\mathbf{1}}^{\perp}\right)$ as *second poles of ASY-DAGP* and define the *poles of ASY-DAGP* as the union of the first and the second poles.

Assumptions 4 and 5 are special cases of Assumption 6, obtained by substituting $(\mathbf{W}, \mathbf{Q})$ with their scaled versions $(\iota\bar{\mathbf{W}}, \iota\bar{\mathbf{Q}})$ and letting the isolation factor ι approach zero. Hence, if either Assumption 4 or 5 holds, Assumption 6 is satisfied for sufficiently small ι. A detailed proof is provided in Appendix F.

Now, we may define the delay factor as follows:

Definition 4. We define the p^{th} *forward-backward* impulse response as:

$$\psi_p(k) := \begin{cases} z_p^k & \text{if } |z_p| < 1, k \geq 0 \\ -z_p^k & \text{if } |z_p| > 1, k < 0 \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where z_p is one of the poles of ASY-DAGP. We also define the p^{th} delay response of link (v, u) as

$$\tau_p^{vu}(m) = \sum_l \max\left\{\tau_{p,l,1}(m), \tau_{p,l,2}(m)\right\} \tag{19}$$

where:

$$\tau_{p,l,1}(m) = \left|\psi_p(l-m) - \frac{1}{|T_l^{vu}|}\sum_{k \in T_l^{vu}} \psi_p(k-m)\right|$$

$$\tau_{p,l,2}(m) = \left|\psi_p(l-m) - \sum_{k \in S_l^{vu}} \frac{1}{|T_k^{vu}|}\psi_p(k-m)\right|$$

The set S_l^{vu} denotes all k such that $l \in T_k^{vu}$. Finally, we define the delay factor κ as the supremum of $\tau_p^{vu}(m)$ over all links, all poles, and all $m = 0, 1, \ldots$.

Note that in the absence of temporal distortion, $T_k^{vu} = \{k\}$ and $S_l^{vu} = \{l\}$, making the delay factor zero. We now present our general result based on the definition of the delay factor.

Theorem 2. *Suppose that Assumption 1 holds true for a pair* $(\bar{\mathbf{W}}, \bar{\mathbf{Q}})$*. For every finite* κ*, there exists* $\iota = O(1/\kappa)$ *such that taking* $\mathbf{W} = \iota\bar{\mathbf{W}}, \mathbf{Q} = \iota\bar{\mathbf{Q}}$*, leads to the following rates whenever Assumption 6 also holds for* $(\mathbf{W}, \mathbf{Q})$*:*

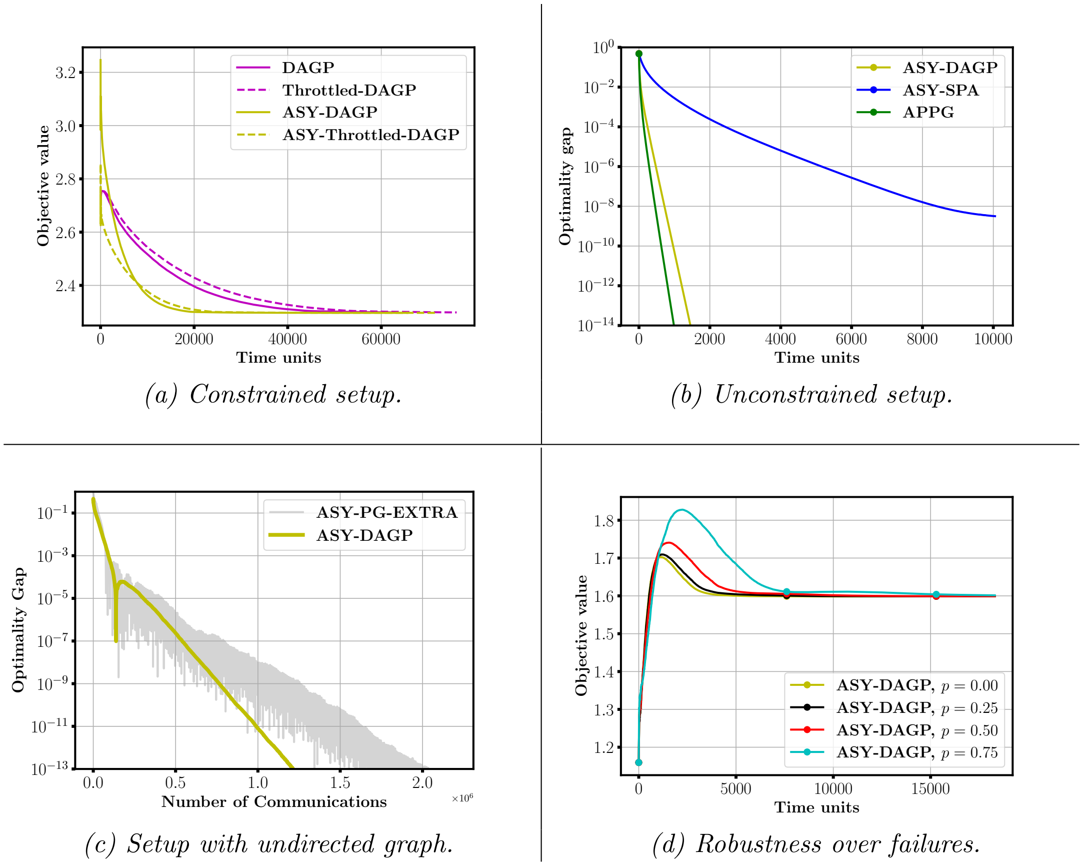
$$\sum_v \operatorname{dist}^2(\bar{\mathbf{x}}_K, S_v) \leq \sum_\nu \|\bar{\mathbf{x}}_K^v - \bar{\mathbf{x}}_K\|_2^2 = O\left(\frac{\kappa C_0}{\zeta K}\right), \tag{20}$$

$$\sum_v f^v(\bar{\mathbf{x}}_K^v) - \sum_v f^v(\mathbf{x}^*) = O\left(\frac{C_0}{\mu K} + \sqrt{\frac{\kappa C_0 C_1}{\zeta K}}\right), \tag{21}$$

where K*,* $\bar{\mathbf{x}}_K^v$*,* $\bar{\mathbf{x}}_K$ *and the constants* C_0, C_1 *are as defined in Theorem 1.*

Compared to Theorem 1, the term d is replaced, in Theorem 2, with κ, which is a more general concept. Let us show how Theorem 1 is obtained from Theorem 2. Under Assumptions 2 and 3, the delay factor can be proved to be bounded by $\kappa \leq 2\max\frac{1-|z_p|^d}{1-|z_p|} \leq 2d$, where max is taken over all poles inside the unit circle. Then, Theorem 1 is obtained by taking $\mathbf{W} = \iota\bar{\mathbf{W}}, \mathbf{Q} = \iota\bar{\mathbf{Q}}$ which bounds the term $\iota\kappa$ by a constant. When d grows large, we verify Assumption 6, which leads to the two cases both with $\zeta = \iota\bar{\zeta}$ and their corresponding Assumptions 4 and 5. Replacing ζ proves Theorem 1. More details and the proof of Theorem 2 is presented in the appendix. Theorem 2 discovers regimes, where Assumption 4 does not hold but the poles still remain distant from the unit circle.

Remark 3. Our theorems are not for particular $\mathbf{W}$ and $\mathbf{Q}$ matrices. $\mathbf{W}, \mathbf{Q}$ are restricted in two ways: First, we require Assumptions 1 and 6. This is natural and aligned with other studies. Even for a simple gossip protocol, certain assumptions on the gossip matrix are required for convergence (Koloskova et al., 2019). Note that we do not restrict the underlying network, because $\mathbf{W}, \mathbf{Q}$ can always be chosen to be proportional to its in/out Laplacian. Second, our analysis requires suitably scaled gossip matrices by ι. This scaling allows our algorithm to be valid for arbitrary delays, and it does not necessarily limit our theorem and the underlying network.



(a) Constrained setup.

(b) Unconstrained setup.

(c) Setup with undirected graph.

(d) Robustness over failures.

Figure 2: *(a)* Solving a constrained problem, and drawing a comparison to DAGP and its throttled version. *(b)* Solving unconstrained logistic regression problem, and comparing to APPG and ASY-SPA. *(c)* Solving a constrained problem over undirected graphs, and comparing to ASY-PG-EXTRA. *(d)* Robustness to message losses with the communication failure probability of p.

5 EXPERIMENTAL RESULTS

In our experiments, we utilized directed random networks, with edge probability p_e. We assumed that the delays follow an exponential distribution with mean τ^{vu}_{comm}. In the first experiment, the computation time for each node followed a uniform distribution within $[1, \bar{\tau}^{v}_{\text{comp}}]$. In the second one, they were exponentially distributed with mean τ^{v}_{comp}. We used scaled Graph Laplacian matrices as gossip matrices. We report the objective values computed at the average solution across all nodes.

In every experiment, we simulated a distributed setup on a single machine by sequentially performing the actions of different nodes. This approach gives better control over parameters and facilitates the implementation of asynchrony, delays, and message failures. As our experiments have an illustrative nature, we postpone more realistic implementations to a future study.

To the best of our knowledge, there is no other algorithm like ASY-DAGP that handles a constrained optimization with asynchrony over directed graphs. Hence, we compare ASY-DAGP to other existing algorithms in simplified setups. Specifically, in the first experiment, we use a directed and constrained setup, but it is synchronous. In the second one, our setup is asynchronous and directed, but without constraints. The third experiment features an asynchronous setup that is constrained, but undirected.

First experiment: In this experiment, we compare ASY-DAGP with DAGP (Shahriari-Mehr and Panahi, 2022). We consider the following local objective functions and constraints

$$f^{v}(\mathbf{x}) = \log\big(\cosh(\mathbf{a}_v^T\mathbf{x} - b_v)\big), \tag{22}$$

$$S^{v} = \{\mathbf{x} \mid \mathbf{c}_v^T\mathbf{x} - d_v \leq 0\}, \tag{23}$$

where constants and coefficients are generated from normal distributions. We initialize both algorithms randomly and assume $M = 10, \mathbf{x} \in \mathbb{R}^5, p_e = 0.8, \tau^{vu}_{\text{comm}} = 10, \bar{\tau}^{v}_{\text{comp}} = 5v$, where v indicates the node number. We set the design parameters to $\rho = 0.01, \alpha = 0.1, \gamma = 0.5, \eta = 1.0$. Figure 2a presents the resulting objective values. We observe both algorithms converge to the same point, demonstrating that ASY-DAGP is capable of reaching the optimal solution of (P). ASY-DAGP achieves significantly faster wall-clock convergence. It is worth noting that an increase in heterogeneity of the local computation time (called throttling) – for instance, by deliberately slowing down several nodes (two nodes in this experiment) by a factor of two – may result in a slower convergence for the synchronous setup. For ASY-DAGP, this situation is not harmful.

To further illustrate the merits of asynchronous algorithms over synchronous algorithms, extended simulations are provided in Appendix H.

Second experiment: We compare ASY-DAGP to the APPG algorithm (Zhang and You, 2019b) in an asynchronous setup. We consider an unconstrained logistic regression problem with ℓ_2 regularization, similar to the one presented by Shahriari-Mehr and Panahi (2022). This problem is applied to two digits of the MNIST dataset (LeCun et al., 2010). A random graph with $p_e = 0.6$ and $M = 20$ nodes is considered. We use $N_s = 1000$ samples from the dataset and a regularization factor of $1/N_s$. The optimal solution is computed by running the centralized gradient descent. The optimality gap, i.e., the error in the objective function, is reported in Figure 2b. The design parameters are set to $\tau^{vu}_{\text{comm}} = 50, \tau^{v}_{\text{comp}} = 5v, \gamma = 0.5, \eta = 1.0, \rho = 0.1, \alpha = 0.7$. For both APPG and ASY-DAGP, we selected similar step sizes slightly smaller than $2/L$. Although ASY-SPA requires a diminishing step size, we chose a small fixed one. With this choice, we may only arrive at a neighborhood of the optimal solution. ASY-DAGP performs on par with APPG but is slightly slower, yet it is capable of solving constrained problems.

Third experiment: In this experiment, we use the same setup as in the first experiment, with the key difference of utilizing an undirected graph. We compare ASY-DAGP with the ASY-PG-EXTRA algorithm by Wu et al. (2017). We employ a near-optimal step size for both algorithms. For the ASY-PG-EXTRA, we set the relaxing parameter to 0.8. The results are shown in Figure 2c. We plot the optimality gap with respect to the number of completed communications. ASY-DAGP converges to the optimal solution with only half the communications needed by ASY-PG-EXTRA to achieve the same result.

Communication failures: The goal of this experiment is to demonstrate the effectiveness of ASY-DAGP in handling dropped messages. We consider the same problem and parameters as in the first experiment. The messages can be lost with a failure probability of p. Figure 2d investigates different values of p. We observe that increasing the probability of failure generally decrease the speed of convergence. We performed similar experiments on APPG, but it failed to converge.

The conducted experiments provide an evidence of ASY-DAGP's efficiency under diverse conditions. ASY-DAGP performs robustly even under extensive communication failures, tolerating a loss of more than 75% of messages. Additionally, ASY-DAGP surpasses its synchronous counterpart in wall-clock convergence speed. Despite being originally designed for constrained setups with directed communication networks, it can be successfully used for either unconstrained problems or setups with undirected networks.

6 RELATED WORKS

In this section, we provide a review of the relevant literature on decentralized optimization algorithms, focusing primarily on directed communication networks due to their real-world relevance and inherent complexity. We then briefly discuss algorithms designed for undirected and centralized networks, highlighting their differences with respect to our proposed method. Finally, we clarify the distinctions between our novel indexing approach for managing temporal distortion and the approaches found in previous studies.

Decentralized algorithms: Earlier papers consider the unconstrained version of (P). For example, Nedić and Olshevsky (2014, 2016) propose the *subgradient-push* algorithm based on the push-sum protocol. Zhang and You (2019a) and Assran and Rabbat (2020) propose the asynchronous version of the subgradient-push algorithm. These algorithms require a vanishing step size. Taking a fixed step size leads to an error proportional to the average dissimilarity between local objective functions, a quantity related to *distribution shift* in federated learning literature (Reisizadeh et al., 2020; Fallah et al., 2020).

In response to vanishing step size, a family of algorithms based on the so-called gradient tracking technique is introduced (Xi and Khan, 2017; Nedic et al., 2017). In particular, the SONATA algorithm by Scutari and Sun (2019) combines the push-sum protocol with a gradient tracking technique. The Push-Pull algorithm by Xin and Khan (2018) and Pu et al. (2020) forgoes the push-sum protocol, making it a simpler algorithm to analyse and implement. The asynchronous version of the SONATA algorithm, ASY-SONATA, is proposed by Tian et al. (2020). They introduce the perturbed sum-push protocol with a gradient tracking technique. Zhang and You (2019b) propose the fully asynchronous push-pull gradient (APPG) algorithm. Both ASY-SONATA and APPG achieve a linear rate of convergence using a constant step size: ASY-SONATA for strongly convex and smooth problems, while APPG for smooth problems satisfying the Polyak-Lojasiewicz condition.

None of the above studies explicitly consider constraints; Xi and Khan (2016) study constraints by proposing the DDPS algorithm, which handles only identical constraints with diminishing step sizes. The DAGP algorithm by Shahriari-Mehr and Panahi (2022) is the state-of-the-art, ensuring provable convergence for the generic case of (P) with a fixed step size and directed graphs. This work proposes the asynchronous version of DAGP under temporal distortion.

Several papers address asynchrony and constraints by assuming composite objective functions with nonsmooth terms, notably the works by Wu et al. (2017) and Latafat and Patrinos (2022). These studies are based on the dual formulation of the problem, leading to increased communication and computations due to the incorporation of dual variables. The algorithm by Latafat and Patrinos (2022) is partially asynchronous; it only permits agents to handle delayed information. Wu et al. (2017) consider undirected graphs, and their algorithm requires step size adjustments, which demands knowledge of the agents' update rates—a stipulation not practical for real-world applications. Moreover, notable papers (Lian et al., 2018; Peng et al., 2016; Wu et al., 2023; Even et al., 2021b) have explored undirected graphs in asynchronous setups. However, their approach cannot be generalized to directed networks.

The works by Even et al. (2024); Tyurin and Richtárik (2025) are applicable to general graphs but with smooth terms (no constraints). Moreover, the heterogeneous setup in Even et al. (2024) considers a stochastic model of computation, while ours is deterministic. Moreover, their algorithm includes steps that might be difficult to implement in practice.

Centralized algorithms: There are also many papers that have considered centralized federated learning setups (star topology) and have analyzed various variants of asynchronous (S)GD; e.g., papers by (Koloskova et al., 2022; Mishchenko et al., 2022; Islamov et al., 2024; Tyurin and Richtárik, 2023; Even et al., 2024; Tyurin and Richtárik, 2025). These works are outside the scope of this paper; however, Mishchenko et al. (2022) presents a similar attempt to ours in going beyond bounded delays. However, it also does not study constraints and considers a stochastic model, differing significantly from our deterministic setting.

Iteration indexing in the presence of temporal distortion: Defining iteration indices in asynchronous setups with temporal distortion is not straightforward. We introduced a novel indexing method in Section 4 to handle these complexities. Related but different approaches have been explored in the literature, often considering stochastic models for computation and communication delays (Even et al., 2021a, 2024). These stochastic frameworks assign varying computation rates (activation probabilities) to nodes, requiring the amount of update at each iteration to scale inversely with these probabilities. Thus, faster nodes update with smaller steps compared to slower nodes. In contrast, ASY-DAGP assumes equal rates across all nodes, justifying the choice of a uniform iteration index across nodes.

7 CONCLUSION

We proposed an asynchronous decentralized algorithm that accommodates delays and missing messages, and is capable of solving smooth optimization problems with local constraints over directed graphs. The asynchronous updates eliminate idle time, thus leading to faster wall-clock convergence. We presented the convergence analysis of our algorithm based on a novel generalized relaxation of the PEP framework. Our analysis results in summarizing the effect of the asynchrony and delay in a parameter that we referred to as delay factor. Lastly, our experimental results substantiate the resilience against dropped messages, and show ASY-DAGP surpasses existing algorithms, even in their specialized, restricted setups.

Acknowledgements

We express our gratitude to the anonymous reviewers for their insightful feedback. This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

Mahmoud Assran, Arda Aytekin, Hamid Reza Feyzmahdavian, Mikael Johansson, and Michael G Rabbat. Advances in asynchronous parallel and distributed optimization. *Proceedings of the IEEE*, 108 (11):2013–2031, 2020.

Mahmoud S Assran and Michael G Rabbat. Asynchronous gradient push. *IEEE Transactions on Automatic Control*, 66(1):168–183, 2020.

Marco Bornstein, Tahseen Rabbani, Evan Wang, Amrit Singh Bedi, and Furong Huang. Swift: Rapid decentralized federated learning via wait-free model communication. *arXiv preprint arXiv:2210.14026*, 2022.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien.

Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145 (1-2):451–482, 2014.

Mathieu Even, Raphaël Berthier, Francis Bach, Nicolas Flammarion, Hadrien Hendrikx, Pierre Gaillard, Laurent Massoulié, and Adrien Taylor. Continuized accelerations of deterministic and stochastic gradient descents, and of gossip algorithms. *Advances in Neural Information Processing Systems*, 34:28054–28066, 2021a.

Mathieu Even, Hadrien Hendrikx, and Laurent Massoulié. Asynchronous speedup in decentralized optimization. *arXiv preprint arXiv:2106.03585*, 2021b.

Mathieu Even, Anastasia Koloskova, and Laurent Massoulié. Asynchronous sgd on graphs: a unified framework for asynchronous decentralized and federated optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 64–72. PMLR, 2024.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

Robert Hannah, Fei Feng, and Wotao Yin. A2bcd: Asynchronous acceleration with optimal complexity. In *International Conference on Learning Representations*, 2018.

Rustem Islamov, Mher Safaryan, and Dan Alistarh. Asgrad: A sharp unified analysis of asynchronous-sgd algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 649–657. PMLR, 2024.

Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International conference on machine learning*, pages 3478–3487. PMLR, 2019.

Anastasiia Koloskova, Sebastian U Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous sgd for distributed and federated learning. *Advances in Neural Information Processing Systems*, 35:17202–17215, 2022.

Puya Latafat and Panagiotis Patrinos. Primal-dual algorithms for multi-agent structured optimization over message-passing architectures with bounded communication delays. *Optimization Methods and Software*, 37(6):2052–2079, 2022.

Yann LeCun, Corinna Cortes, Chris Burges, et al. Mnist handwritten digit database, 2010.

Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, pages 3043–3052. PMLR, 2018.

Konstantin Mishchenko, Francis Bach, Mathieu Even, and Blake E Woodworth. Asynchronous sgd beats minibatch sgd under arbitrary delays. *Advances in Neural Information Processing Systems*, 35:420–433, 2022.

Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2014.

Angelia Nedić and Alex Olshevsky. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *IEEE Transactions on Automatic Control*, 61(12):3936–3947, 2016.

Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Duong Thuy Anh Nguyen, Duong Tung Nguyen, and Angelia Nedić. Accelerated ab/push–pull methods for distributed optimization over time-varying directed networks. *IEEE Transactions on Control of Network Systems*, 11(3):1395–1407, 2023.

Zhimin Peng, Yangyang Xu, Ming Yan, and Wotao Yin. Arock: an algorithmic framework for asynchronous parallel coordinate updates. *SIAM Journal on Scientific Computing*, 38(5):A2851–A2879, 2016.

Boris T Polyak. Introduction to optimization. optimization software. *Inc., Publications Division, New York*, 1:32, 1987.

Shi Pu, Wei Shi, Jinming Xu, and Angelia Nedić. Push–pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, 66(1):1–16, 2020.

Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: The case of affine distribution shifts. *Advances in Neural Information Processing Systems*, 33: 21554–21565, 2020.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162:83–112, 2017.

Gesualdo Scutari and Ying Sun. Distributed nonconvex constrained optimization over time-varying digraphs. *Mathematical Programming*, 176:497–544, 2019.

Firooz Shahriari-Mehr and Ashkan Panahi. Double averaging and gradient projection: Convergence guarantees for decentralized constrained optimization, 2022. URL https://arxiv.org/abs/2210.03232.

Adrien B Taylor. *Convex interpolation and performance estimation of first-order methods for convex optimization.* PhD thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2017.

Ye Tian, Ying Sun, and Gesualdo Scutari. Achieving linear convergence in distributed asynchronous multiagent optimization. *IEEE Transactions on Automatic Control*, 65(12):5264–5279, 2020.

Alexander Tyurin and Peter Richtárik. Optimal time complexities of parallel stochastic optimization methods under a fixed computation model. *Advances in Neural Information Processing Systems*, 36:16515–16577, 2023.

Alexander Tyurin and Peter Richtárik. On the optimal time complexities in decentralized stochastic asynchronous optimization. *Advances in Neural Information Processing Systems*, 37:122652–122705, 2025.

Tianyu Wu, Kun Yuan, Qing Ling, Wotao Yin, and Ali H Sayed. Decentralized consensus optimization with asynchrony and delays. *IEEE Transactions on Signal and Information Processing over Networks*, 4 (2):293–307, 2017.

Xuyang Wu, Changxin Liu, Sindri Magnusson, and Mikael Johansson. Delay-agnostic asynchronous distributed optimization. *arXiv preprint arXiv:2303.18034*, 2023.

Chenguang Xi and Usman A Khan. Distributed subgradient projection algorithm over directed graphs. *IEEE Transactions on Automatic Control*, 62(8): 3986–3992, 2016.

Chenguang Xi and Usman A Khan. Dextra: A fast algorithm for optimization over directed graphs. *IEEE Transactions on Automatic Control*, 62(10):4980–4993, 2017.

Ran Xin and Usman A Khan. A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Systems Letters*, 2(3): 315–320, 2018.

Ran Xin, Usman A Khan, and Soummya Kar. Fast decentralized nonconvex finite-sum optimization with recursive variance reduction. *SIAM Journal on Optimization*, 32(1):1–28, 2022.

Jiaqi Zhang and Keyou You. Asyspa: An exact asynchronous algorithm for convex optimization over digraphs. *IEEE Transactions on Automatic Control*, 65(6):2494–2509, 2019a.

Jiaqi Zhang and Keyou You. Fully asynchronous distributed optimization with linear convergence in directed networks. *arXiv preprint arXiv:1901.08215*, 2019b.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [**Yes**/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [**Yes**/No/Not Applicable]
   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [**Yes**/No/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [**Yes**/No/Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [**Yes**/No/Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [**Yes**/No/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [**Yes**/No/Not Applicable]
   (b) The license information of the assets, if applicable. [Yes/No/**Not Applicable**]

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/**Not Applicable**]

(d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

The equation numbers and references are in agreement with the main body of the paper.

A Mathematical notation

We denote vectors using bold lowercase letters and matrices with bold uppercase letters. The element at the v^{th} row and u^{th} column of matrix $\mathbf{W}$ is denoted by w_{vu}. The transpose of $\mathbf{W}$ is $\mathbf{W}^T$, and its right null space is shown by $\ker(\mathbf{W})$. Therefore, $\mathbf{x}$ belongs to $\ker(\mathbf{W})$ if and only if $\mathbf{Wx} = \mathbf{0}$. We use $\mathbf{1}_n$ and $\mathbf{0}_n$ to respectively denote $n-$dimensional all-one and all-zero vectors. Furthermore, an $m \times n$ matrix containing only zero elements is shown by $\mathbf{O}_{m\times n}$. The indices m and n may be omitted if there is no risk of confusion. The Euclidean inner product of vectors is denoted by $\langle .,. \rangle$, and the matrix inner product is denoted by $\langle \mathbf{A}, \mathbf{C} \rangle = \mathrm{Tr}(\mathbf{AC}^T)$, where $\mathrm{Tr}(.)$ denotes the matrix trace operator. The Kronecker delta function is shown by $\delta_{k,l}$. ∂I_S indicates the normal cone of S. $|T|$ indicates the cardinality of set T.

We generally use subscripts to define the iteration number, and superscripts to denote the node number. For example, $\nabla f^v(\mathbf{x}_k^v)$ represents the gradient of the local function of node v at its local variable at the k^{th} iteration. Additionally, to provide matrix representations of equations, we arrange vector variables as the rows of a matrix. For instance, the matrix $\mathbf{G} \in \mathbb{R}^{M\times m}$ contains all $\mathbf{g}^v \in \mathbb{R}^m$ vectors, with $v \in \{1, \dots, M\}$, as its rows. For simplicity, we write $\mathbf{G} \in \ker(\mathbf{W})$ to imply that each column of $\mathbf{G}$ is an element in the null space of $\mathbf{W}$. For simplicity, we call the cone of the normal directions of a convex set S at a point $\mathbf{x}$ its normal cone.

B Analysis of fixed points

We start by analyzing the fixed points of ASY-DAGP. This provides a simple explanation of the main intuitions behind the design and convergence of ASY-DAGP. In this analysis, we verify that any fixed point of ASY-DAGP is inevitably an optimal solution of (P). If this does not hold true, our analysis identifies undesired local minima, and disproves global convergence. However, the mere existence of global minima (i.e. showing that the fixed points are optimal) does not imply convergence to them. This requires a separate *convergence analysis*, which we carry out afterwards.

A fixed point of an algorithm is any set of state vectors and variables (simply called a point) which has a potential of being a convergence point. In algorithms with no temporal distortion, a fixed point is simply a point that remains unchanged once attained, but this definition does not work for time-varying systems, such as ASY-DAGP. Instead, we define a fixed point as the one which remains unchanged after being maintained for a sufficiently long number of iterations and achieves consensus among the agents. Such a point naturally satisfies $\mathbf{x}_{k+1}^v = \mathbf{x}_k^v = \mathbf{x}^v$, and a similar relation must hold true for $\mathbf{g}^v, \mathbf{p}^v$, and $\mathbf{h}^v$ variables. Since, the solution is attained for a long time, the buffers may only store the values of this point (forget old solutions) and hence we have $\mathbf{a}_{k+1}^{vu} = \mathbf{a}_k^{vu} = \mathbf{x}^u$, and $\mathbf{b}_{k+1}^{vu} = \mathbf{b}_k^{vu} = \mathbf{p}^u$. Note that these relations will hold whether or not a buffer is empty. Now, let us investigate what happens if the solution is maintained in the next iteration. This so called fixed point iteration of ASY-DAGP can be written in the following matrix form:

$$\mathbf{Z} = \mathbf{X} - \mathbf{WX} - \mu(\boldsymbol{\nabla}\mathbf{F} - \mathbf{G}), \quad \mathbf{X} = \mathbf{P}(\mathbf{Z}) \tag{24}$$

$$\alpha(\mathbf{H} - \mathbf{G}) + \rho\left(\boldsymbol{\nabla}\mathbf{F} - \mathbf{G} + \frac{1}{\mu}(\mathbf{Z} - \mathbf{X})\right) = \mathbf{O}, \tag{25}$$

$$\mathbf{QP} = (\gamma - 1)\mathbf{G} = (\gamma - 1)\mathbf{H}. \tag{26}$$

Accordingly, the ASY-DAGP iterations are designed to establish the following relations: From the last equation, $\mathbf{H} = \mathbf{G}$. Then, from (25), we have:

$$\mathbf{G} = \boldsymbol{\nabla}\mathbf{F} + \frac{1}{\mu}(\mathbf{Z} - \mathbf{X}). \tag{27}$$

This relation proves that the fixed point is the consensus and optimal solution of (P). To see this, note that by (27) and (24), we have $\mathbf{WX} = \mathbf{O}$, which corroborate a consensus solution, i.e. $\mathbf{x}^v = \mathbf{x}^u = \mathbf{x}$ for all $v, u \in \mathcal{V}$. To show $\mathbf{x}$ is an optimal solution, first note that $\mathbf{1}^T\mathbf{G} = \mathbf{0}^T$ from (26). Moreover, consider that $\mathbf{z}^v - \mathbf{x}^v \in \partial I_{S^v}$, for all $v \in \mathcal{V}$. Then, by left multiplying (27) with $\mathbf{1}^T$, considering the conic property of normal cone, we have the optimality condition satisfied at $\mathbf{x}$. We conclude that any fixed point is a consensus, feasible and optimal solution of the underlying problem.

Note that in the similar analysis of DAGP, the extra assumption $\ker(\mathbf{Q}) = \ker(\mathbf{W}^T)$ is required to show any fixed point is a consensus and optimal solution of problem (P). However, In ASY-DAGP, by defining new $\mathbf{p}^v$ variables and designing their update equation, defined in (4), based on the satisfaction of the optimality condition for the fixed point, there is no need for this extra assumption on gossip matrices anymore.

In Section D, we show that the algorithm is bound to converge and also there exist an upper bound on the convergence time.

C Simplifications of algorithm dynamics

Before proceeding to the proof of the main results in the paper, we introduce a series of transformations to the variables of the ASY-DAGP. These transformations remarkably simplify our analysis and our expressions.

Changing the origin to the optimal solution: First, we re-represent the variables in a shifted coordinated system that is centered on a suitable optimal solution of (P). To this end, we take an optimal solution of (P), assumed in Item 3 of section (2). This point is naturally the fixed point of the algorithm and satisfies the set of relations in Section B. We define the shifted variables as

$$\tilde{\mathbf{x}}_k^v := \mathbf{x}_k^v - \mathbf{x}^*, \quad \tilde{\mathbf{a}}_k^{vu} := \mathbf{a}_k^{vu} - \mathbf{x}^*, \quad \tilde{\mathbf{b}}_k^{vu} := \mathbf{b}_k^{vu} - \mathbf{x}^*, \tag{28}$$

$$\tilde{\mathbf{g}}_k^v := \mathbf{g}_k^v - (\nabla f^v(\mathbf{x}^*) + \mathbf{n}^v), \quad \tilde{\mathbf{h}}_k^v := \mathbf{h}_k^v - (\nabla f^v(\mathbf{x}^*) + \mathbf{n}^v), \tag{29}$$

$$\tilde{\mathbf{p}}_k^v := \mathbf{p}_k^v - \frac{\gamma - 1}{\sum_u q_{vu}}(\nabla f^v(\mathbf{x}^*) + \mathbf{n}^v), \tag{30}$$

where $\mathbf{n}^v$ is an element in the normal cone of S^v at $\mathbf{x}^*$, i.e., $\partial I_{S^v}(\mathbf{x}^*)$, satisfying the optimality condition in (1).

Eliminating $\mathbf{z}_k^v$ in the dynamics: Our next step is to plug the definition of $\mathbf{z}_{k+1}^v$ into the update rule or the dynamics of $\mathbf{g}_k^v$ in (5). This gives us

$$\mathbf{g}_{k+1}^v = \mathbf{g}_k^v + \frac{\rho}{\mu}\left(\mathbf{x}_k^v - \sum_u w_{vu}\mathbf{a}_k^{vu} - \mathbf{x}_{k+1}^v\right) + \alpha(\mathbf{h}_k^v - \mathbf{g}_k^v). \tag{31}$$

Reformulating equations (7) and (8) and eliminating the buffer variables: Our final step is to reduce the number of variables involved in the analysis by eliminating the buffer vectors. Take $\mathbb{1}_{\text{condition}}$ as the indicator function, which returns 1 if the specified condition is satisfied, and 0 otherwise. Then, we may write (7) and (8) as

$$\mathbf{a}_{k+1}^{vu} = \mathbf{x}_{k+1}^u + \sum_{l=0}^{K-1} c_{k,k-l}^{vu}\mathbf{x}_l^u, \quad \mathbf{b}_{k+1}^{vu} = \mathbf{p}_{k+1}^u + \sum_{l=0}^{K-1} c_{k,k-l}^{vu}\mathbf{p}_l^u, \tag{32}$$

where K is the minimum number of iterations that all nodes have performed and

$$c_{k,k-l}^{vu} = -\mathbb{1}_{l=k} + \frac{1}{|T_k^{vu}|}\mathbb{1}_{l\in T_k^{vu}}. \tag{33}$$

Note that the expressions in (32) are in the form of ideal messages $\mathbf{x}_k^u$ and $\mathbf{p}_k^u$ with additional temporal distortion terms.

Resulting algorithm dynamics: Combining all the above simplifications results in the following algorithm dynamics, where, for simplicity, we denote $\sum_u \sum_{l=0}^{K-1}$ as $\sum_{u,l}$.

$$\tilde{\mathbf{g}}_{k+1}^v = \tilde{\mathbf{g}}_k^v + \frac{\rho}{\mu}\left(\tilde{\mathbf{x}}_k^v - \sum_u w_{vu}\tilde{\mathbf{x}}_k^u - \sum_{u,l} w_{vu}c_{k,k-l}^{vu}\tilde{\mathbf{x}}_l^u - \tilde{\mathbf{x}}_{k+1}^v\right) + \alpha(\tilde{\mathbf{h}}_k^v - \tilde{\mathbf{g}}_k^v), \tag{34}$$

$$\tilde{\mathbf{p}}_{k+1}^v = \tilde{\mathbf{p}}_k^v - \eta\sum_u q_{vu}\tilde{\mathbf{p}}_k^u - \eta\sum_{u,l} q_{vu}c_{k,k-l}^{vu}\tilde{\mathbf{p}}_l^u + \eta(\gamma-1)\tilde{\mathbf{g}}_k^v, \tag{35}$$

$$\tilde{\mathbf{h}}_{k+1}^v = \gamma\tilde{\mathbf{h}}_k^v - \sum_u q_{vu}\tilde{\mathbf{p}}_k^u - \sum_{u,l} q_{vu}c_{k,k-l}^{vu}\tilde{\mathbf{p}}_l^u. \tag{36}$$

D Proof of Theorem 2

D.1 Proof overview

Our proof is based on three common and elementary inequalities:

1. Convexity of each term f^v:

$$\forall \mathbf{x}, \mathbf{y}; \quad f^v(\mathbf{y}) \geq f^v(\mathbf{x}) + \langle \mathbf{y} - \mathbf{x}, \nabla f^v(\mathbf{x}) \rangle \tag{37}$$

2. Smoothness of each term f^v:

$$\forall \mathbf{x}, \mathbf{y}; \quad f^v(\mathbf{y}) \leq f^v(\mathbf{x}) + \langle \mathbf{y} - \mathbf{x}, \nabla f^v(\mathbf{x}) \rangle + \frac{L}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 \tag{38}$$

3. Variational inequality of projection operators:

$$\forall \mathbf{x} \in S^v, \mathbf{y}; \quad 0 \geq \langle \mathbf{y} - P_S(\mathbf{y}), \mathbf{x} - P_S(\mathbf{y}) \rangle \tag{39}$$

We also define the following positive sub-optimality metric $\mathcal{P}_k$ of the variables at iterations k:

$$\mathcal{P}_k = \sum_{v=1}^{M} \mu \left(F_{k+1}^v + T_{k+1}^v \right) + \frac{\zeta}{4M} \sum_{u,v=1}^{M} \|\mathbf{x}_{k+1}^u - \mathbf{x}_{k+1}^v\|^2, \tag{40}$$

which consists of:

1. Bregman divergence of individual terms:

$$F_k^v := F^v(\mathbf{x}_k^v); \quad F^v(\mathbf{x}) := f^v(\mathbf{x}) - f^v(\mathbf{x}^*) - \langle \nabla f^v(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle, \tag{41}$$

   where $\mathbf{x}^*$ is a regular optimal feasible solution of (P), satisfying item 3 in Section 2.

2. Variational bound of individual constraints:

$$T_k^v := T^v(\mathbf{x}_k^v); \quad T^v(\mathbf{x}) := -\langle \mathbf{n}^v, \mathbf{x} - \mathbf{x}^* \rangle, \tag{42}$$

   where $\mathbf{n}^v \in \partial I_{S^v}(\mathbf{x}^*)$ (i.e. $\mathbf{n}^v$ is an element in the normal cone of S^v at $\mathbf{x}^*$), satisfying the optimality condition in (1).

3. The variation around consensus:

$$\mathcal{C}_k := \frac{\zeta}{4M} \sum_{u,v=1}^{M} \|\mathbf{x}_k^u - \mathbf{x}_k^v\|^2 = \frac{\zeta}{4M} \sum_{u,v=1}^{M} \|\tilde{\mathbf{x}}_k^u - \tilde{\mathbf{x}}_k^v\|^2. \tag{43}$$

Note that each of these terms is non-negative: $F^v(\mathbf{x})$ is always positive by convexity. $T^v(\mathbf{x}_k^v)$ is positive for $k = 1, 2 \ldots$ because $\mathbf{x}_k^v$ is in S^v by design. Finally, $\mathcal{C}_k$ is a sum of positive terms. We utilize the above elementary inequalities at various points and make a linear combination of them to show that $\mathcal{P}_k$ vanishes such that $\sum_k \mathcal{P}_k$ remains finite.

D.1.1 From $\mathcal{P}_k$ to the claims of Theorem 2

We shortly discuss how to show that $\sum_k \mathcal{P}_k$ remains finite, but first show how this bound implies the claims of our theorem. This is explained in the following proposition:

Proposition 2. Suppose that there exists a constant $B > 0$ such that $\sum\limits_{k=0}^{K-1} \mathcal{P}_k \leq B$ for a given $K \geq 1$. For every $K = 1, 2, \ldots$ define the time-averaged local solutions $\bar{\mathbf{x}}_K^v = \frac{1}{K} \sum\limits_{k=0}^{K-1} \mathbf{x}_k^v$ and the consensus solution $\bar{\mathbf{x}}_K = \frac{1}{M} \sum\limits_{v=1}^{M} \bar{\mathbf{x}}_K^v$.

Then, the following statements hold true:

Consensus: The time-averaged local solutions converge to the consensus solution by the following rate:

$$\sum_{v\in\mathcal{V}} \|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_K^v\|_2^2 \le \frac{2B}{\zeta K}. \tag{44}$$

Feasibility gap: The distance of the consensus solution to every constraint set vanishes with:

$$\mathrm{dist}^2(\bar{\mathbf{x}}_K, S^v) \le \frac{2B}{\zeta K}. \qquad \forall v \in \mathcal{V} \tag{45}$$

Optimality gap: The objective value converges to the optimal value by:

$$\sum_v f^v(\bar{\mathbf{x}}_K^v) \le \sum_v f^v(\mathbf{x}^*) + \frac{B}{\mu K} + \sqrt{\frac{2BC_1}{\zeta K}}, \tag{46}$$

where $C_1 = \sum\limits_v \|\mathbf{n}^v + \nabla f^v(\mathbf{x}^*)\|^2$.

Proof. For the first part, we observe, by the Jensen's inequality, that

$$\sum_{v\in\mathcal{V}} \|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_K^v\|_2^2 = \frac{1}{2M}\sum_{u,v\in\mathcal{V}} \|\bar{\mathbf{x}}_K^u - \bar{\mathbf{x}}_K^v\|_2^2 \le \frac{1}{2MK}\sum_{k=0}^{K-1}\sum_{u,v\in\mathcal{V}} \|\mathbf{x}_k^u - \mathbf{x}_k^v\|_2^2 \le \frac{2}{K\zeta}\sum_{k=0}^{K-1}\mathcal{P}_k \le \frac{2B}{K\zeta} \tag{47}$$

For the second part, we note that $\bar{\mathbf{x}}_K^v \in S_v$ and hence

$$\mathrm{dist}^2(\bar{\mathbf{x}}_K, S^v) \le \|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_K^v\|_2^2 \le \frac{2B}{\zeta K} \qquad \forall v \in \mathcal{V} \tag{48}$$

For the third part, we note that

$$\begin{aligned}
\frac{B}{\mu K} &\ge \frac{1}{\mu K}\sum_{k=0}^{K-1}\mathcal{P}_k \\
&\ge \frac{1}{K}\sum_{k=0}^{K-1}\sum_v F_k^v + T_k^v \\
&= \frac{1}{K}\sum_{k=0}^{K-1}\sum_v f^v(\mathbf{x}_k^v) - f^v(\mathbf{x}^*) - \langle \nabla f^v(\mathbf{x}^*) + \mathbf{n}^v, \mathbf{x}_k^v - \mathbf{x}^*\rangle \\
&\ge \sum_v f^v(\bar{\mathbf{x}}_K^v) - f^v(\mathbf{x}^*) - \langle \nabla f^v(\mathbf{x}^*) + \mathbf{n}^v, \bar{\mathbf{x}}_K^v - \mathbf{x}^*\rangle \\
&= \sum_v f^v(\bar{\mathbf{x}}_K^v) - f^v(\mathbf{x}^*) - \langle \nabla f^v(\mathbf{x}^*) + \mathbf{n}^v, \bar{\mathbf{x}}_K^v - \bar{\mathbf{x}}_K\rangle + \sum_v \langle \nabla f^v(\mathbf{x}^*) + \mathbf{n}^v, \bar{\mathbf{x}}_K - \mathbf{x}^*\rangle \\
&= \sum_v f^v(\bar{\mathbf{x}}_K^v) - f^v(\mathbf{x}^*) - \langle \nabla f^v(\mathbf{x}^*) + \mathbf{n}^v, \bar{\mathbf{x}}_K^v - \bar{\mathbf{x}}_K\rangle \\
&\ge \sum_v f^v(\bar{\mathbf{x}}_K^v) - \sum_v f^v(\mathbf{x}^*) - \sqrt{C_1 \sum_v \|\bar{\mathbf{x}}_K^v - \bar{\mathbf{x}}_K\|_2^2}
\end{aligned} \tag{49}$$

where the last inequality is by the Cauchy-Schwarz inequality. From part 1, we conclude the desired result. ■

Since the squared Euclidean distance in (44) and (45) decays at the rate of $O(1/K)$, the distance itself decays at $O(1/\sqrt{K})$. Consequently, the consensus, feasibility gap, and optimality gap all decay at the order of $O(1/\sqrt{K})$.

D.1.2 Bounding $\sum_k \mathcal{P}_k$

As already stated, we show $\sum_{k=0}^{K-1} \mathcal{P}_k \leq B$ by appealing to the aforementioned elementary inequalities, namely convexity, smoothness and variational inequalities of the projection operators. Due to the delay, this is not a straightforward procedure and consists of multiple steps:

1. In the first step, we employ the elementary inequalities and show that

$$\sum_{k=0}^{K-1} \mathcal{P}_k + \sum_{k=0}^{K-1} \mathcal{A}_k \leq 0, \tag{50}$$

where

$$-\mathcal{A}_k = \tfrac{L\mu}{2} \sum_{v=1}^{M} \left\| \tilde{\mathbf{x}}_{k+1}^v - \tilde{\mathbf{x}}_k^v \right\|^2 +$$
$$\sum_{v=1}^{M} \left\langle \tilde{\mathbf{x}}_{k+1}^v, \tilde{\mathbf{x}}_k^v - \tilde{\mathbf{x}}_{k+1}^v - \sum_u w_{vu} \tilde{\mathbf{x}}_k^u - \sum_{u,l} w_{vu} c_{k,k-l}^{vu} \mathbf{x}_l^u + \mu \tilde{\mathbf{g}}_k^v \right\rangle + \tfrac{\zeta}{4M} \sum_{u,v=1}^{M} \| \tilde{\mathbf{x}}_{k+1}^u - \tilde{\mathbf{x}}_{k+1}^v \|^2. \tag{51}$$

We refer to $\sum_k \mathcal{A}_k$ as the aggregate term. This step is established in Section D.2. Note that unlike $\mathcal{P}_k$, the aggregate term purely depends on the trajectory, not the problem, making it more suitable for the analysis.

2. Next, we observe that due to (50), any lower bound on the aggregate term $\sum_k \mathcal{A}_k$ implies an upper bound on $\sum_k \mathcal{P}_k$. Hence, we formulate a meta-optimization problem, seeking the minimum of the aggregate term over the trajectories satisfying the dynamics (34), (35) and (36). Note that the objective of this meta-optimization problem (the aggregate term with some minor modifications) is quadratic, while its constraint (the dynamics) is linear. Hence, due to its similarity to the celebrated performance estimation program (PEP), we refer to it as *linear-quadratic PEP (LQ-PEP)*. The LQ-PEP is give in (61) in Section D.3.

3. Our task is now to solve the LQ-PEP by the Lagrangian method of multipliers. This is presented in Section E in two steps.

   (a) By investigating the local optimality conditions of LQ-PEP, we establish a homogeneous linear system of equations and show that the aggregate term has a lower bound $-B$ if this system does not have any nonzero solution. This system is given in (67), (68) and (69).

   (b) We notice that this system of equations is in the form of a perturbed, linear, time-invariant (LTI) system with two distinct forward ($\tilde{\mathbf{\Psi}}$) and backward ($\mathbf{\Lambda}$) components and boundary conditions. Hence, we solve it by means of a finite time $z-$ transform and show that it has no nonzero solution. The $z-$transform shows that the solution (as a time series) is expanded in the basis of pole exponents z_p^k. This also bounds the perturbation by the delay factor. Finally, the boundary conditions show that for a small delay factor, there is no nonzero solution.

   For convenience, we present step 3 with an abstract notation and then explicitly verify it in Section E.3.

D.2 Step one: computing the aggregate term

Employing the elementary inequalities: Recall the following definitions:

$$F^v(\mathbf{x}) \coloneqq f^v(\mathbf{x}) - f^v(\mathbf{x}^*) - \langle \nabla f^v(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle, \qquad F_{k+1}^v = F^v(\mathbf{x}_{k+1}^v), \tag{52}$$

$$T^v(\mathbf{x}) \coloneqq -\langle \mathbf{n}^v, \mathbf{x} - \mathbf{x}^* \rangle, \qquad T_{k+1}^v = T^v(\mathbf{x}_{k+1}^v). \tag{53}$$

From convexity of f^v, we have

$$F_k^v + \langle \nabla f^v(\mathbf{x}^*) - \nabla f^v(\mathbf{x}_k^v), \mathbf{x}_k^v - \mathbf{x}^* \rangle \leq 0. \tag{54}$$

From the $L-$smoothness property of f^v, we have

$$F_{k+1}^v - F_k^v - \langle \nabla f^v(\mathbf{x}_k^v) - \nabla f^v(\mathbf{x}^*), \mathbf{x}_{k+1}^v - \mathbf{x}_k^v \rangle - \frac{L}{2} \left\| \mathbf{x}_{k+1}^v - \mathbf{x}_k^v \right\| \leq 0. \tag{55}$$

From the projection step in (2), which results in $\mathbf{x}_{k+1}^v \in S^v$, we have

$$\mu T_{k+1}^v + \langle \mathbf{x}^* - \mathbf{x}_{k+1}^v, \mathbf{z}_{k+1}^v - \mathbf{x}_{k+1}^v - \mu \mathbf{n}^v \rangle \leq 0. \tag{56}$$

By summing (54) and (55), multiplying the resulting expression by μ, and then adding it to (56), we obtain

$$\mu(F_{k+1}^v + T_{k+1}^v) - \frac{L\mu}{2} \left\| \mathbf{x}_{k+1}^v - \mathbf{x}_k^v \right\| + \Big\langle \mathbf{x}^* - \mathbf{x}_{k+1}^v, \mathbf{z}_{k+1}^v - \mathbf{x}_{k+1}^v + \mu\big(\nabla f^v(\mathbf{x}_k^v) - \nabla f^v(\mathbf{x}^*) - \mathbf{n}^v\big) \Big\rangle \leq 0. \tag{57}$$

Plugging algorithm's dynamics: The objective of this step is to eliminate the gradient term from (57). By substituting the definition of $\mathbf{z}_{k+1}^v$ from (2) into (57), we obtain

$$\mu(F_{k+1}^v + T_{k+1}^v) - \frac{L\mu}{2} \left\| \mathbf{x}_{k+1}^v - \mathbf{x}_k^v \right\|^2 + \Big\langle \mathbf{x}^* - \mathbf{x}_{k+1}^v, \mathbf{x}_k^v - \sum_u w_{vu} \mathbf{a}_k^{vu} - \mathbf{x}_{k+1}^v + \mu(\mathbf{g}_k^v - \nabla f^v(\mathbf{x}^*) - \mathbf{n}^v) \Big\rangle \leq 0. \tag{58}$$

By applying the same simplifications as those in Section C, we obtain

$$\mu(F_{k+1}^v + T_{k+1}^v) - \frac{L\mu}{2} \left\| \tilde{\mathbf{x}}_{k+1}^v - \tilde{\mathbf{x}}_k^v \right\|^2 - \Big\langle \tilde{\mathbf{x}}_{k+1}^v, \tilde{\mathbf{x}}_k^v - \tilde{\mathbf{x}}_{k+1}^v - \sum_u w_{vu} \tilde{\mathbf{x}}_k^u + \mu \tilde{\mathbf{g}}_k^v \Big\rangle + \Big\langle \tilde{\mathbf{x}}_{k+1}^v, \sum_{u,l} w_{vu} c_{k,k-l}^{vu} \mathbf{x}_l^u \Big\rangle \leq 0. \tag{59}$$

Define the positive sub-optimality metric: To create the positive sub-optimality metric $\mathcal{P}_k$, as in (39), we add and remove $\frac{\zeta}{2} \sum_{u,v} \|\tilde{\mathbf{x}}_{k+1}^u - \tilde{\mathbf{x}}_{k+1}^v\|^2$ to (59). Then, by summing over all $v = 1, \ldots, M$ and $k = 0, \ldots, K-1$, we have

$$\begin{aligned}
&\sum_{k=0}^{K-1} \left(\sum_{v=1}^{M} \mu \left(F_{k+1}^v + T_{k+1}^v \right) + \frac{\zeta}{4M} \sum_{u,v=1}^{M} \|\tilde{\mathbf{x}}_{k+1}^u - \tilde{\mathbf{x}}_{k+1}^v\|^2 \right) \\
&- \frac{L\mu}{2} \sum_{k=0}^{K-1} \sum_{v=1}^{M} \left\| \tilde{\mathbf{x}}_{k+1}^v - \tilde{\mathbf{x}}_k^v \right\|^2 \\
&- \sum_{k=0}^{K-1} \sum_{v=1}^{M} \Big\langle \tilde{\mathbf{x}}_{k+1}^v, \tilde{\mathbf{x}}_k^v - \tilde{\mathbf{x}}_{k+1}^v - \sum_u w_{vu} \tilde{\mathbf{x}}_k^u - \sum_{u,l} w_{vu} c_{k,k-l}^{vu} \mathbf{x}_l^u + \mu \tilde{\mathbf{g}}_k^v \Big\rangle \\
&- \frac{\zeta}{4M} \sum_{k=0}^{K-1} \sum_{u,v=1}^{M} \|\tilde{\mathbf{x}}_{k+1}^u - \tilde{\mathbf{x}}_{k+1}^v\|^2 \leq 0.
\end{aligned} \tag{60}$$

Note that (60) is the explicit expression of (50). The first line in (60) represents $\sum_k \mathcal{P}_k$ and all the terms in the last three summations represent the aggregate term $\sum_k \mathcal{A}_k$.

D.3 Step two: the meta-optimization problem

In this section, we formulate the following meta-optimization problem, seeking the minimum of the aggregate term $\sum_k \mathcal{A}_k$ over the trajectories satisfying the dynamics (34), (35) and (36).

$$\begin{aligned}
\min_{\{\tilde{\mathbf{x}}_k^v, \tilde{\mathbf{g}}_k^v, \tilde{\mathbf{h}}_k^v, \tilde{\mathbf{p}}_k^v\},\ \forall k,v} \quad & -\frac{L\mu}{2}\sum_{k=0}^{K-1}\sum_{v=1}^{M}\left\|\tilde{\mathbf{x}}_{k+1}^v - \tilde{\mathbf{x}}_k^v\right\|^2 - \frac{\zeta}{4M}\sum_{k=0}^{K-1}\sum_{u,v=1}^{M}\|\tilde{\mathbf{x}}_{k+1}^u - \tilde{\mathbf{x}}_{k+1}^v\|^2 \qquad (61)\\
& -\sum_{k=0}^{K-1}\sum_{v=1}^{M}\left\langle \tilde{\mathbf{x}}_{k+1}^v, \tilde{\mathbf{x}}_k^v - \tilde{\mathbf{x}}_{k+1}^v - \sum_u w_{vu}\tilde{\mathbf{x}}_k^u - \sum_{u,l} w_{vu} c_{k,k-l}^{vu}\tilde{\mathbf{x}}_l^u + \mu\tilde{\mathbf{g}}_k^v \right\rangle \\
\text{subject to} \quad & \tilde{\mathbf{g}}_{k+1}^v - \tilde{\mathbf{g}}_k^v - \frac{\rho}{\mu}\left(\tilde{\mathbf{x}}_k^v - \sum_u w_{vu}\tilde{\mathbf{x}}_k^u - \tilde{\mathbf{x}}_{k+1}^v\right) - \alpha(\tilde{\mathbf{h}}_k^v - \tilde{\mathbf{g}}_k^v) = -\frac{\rho}{\mu}\sum_{u,l} w_{vu} c_{k,k-l}^{vu}\tilde{\mathbf{x}}_l^u, \qquad \forall k,v\\
& \tilde{\mathbf{p}}_{k+1}^v - \tilde{\mathbf{p}}_k^v + \eta\sum_u q_{vu}\tilde{\mathbf{p}}_k^u - \eta(\gamma-1)\tilde{\mathbf{g}}_k^v = -\eta\sum_{u,l} q_{vu} c_{k,k-l}^{vu}\tilde{\mathbf{p}}_l^u, \qquad \forall k,v\\
& \tilde{\mathbf{h}}_{k+1}^v - \gamma\tilde{\mathbf{h}}_k^v + \sum_u q_{vu}\tilde{\mathbf{p}}_k^u = -\sum_{u,l} q_{vu} c_{k,k-l}^{vu}\tilde{\mathbf{p}}_l^u, \qquad \forall k,v\\
& \tilde{\mathbf{x}}_0^v,\ \tilde{\mathbf{g}}_0^v,\ \tilde{\mathbf{h}}_0^v,\ \tilde{\mathbf{p}}_0^v \text{ are bounded.} \qquad \forall v
\end{aligned}$$

Note that a sufficiently bad initialization leads to an arbitrarily large optimality gap in any given iteration. Hence, the last constraint is necessary, but at this point, we do not need its explicit form. We present a complete analysis of this optimization problem in Section E. However, we first note that the optimization problem in (61) is related to the PEP literature, on which we elaborate in Section D.4 for interested readers. This discussion is not essential for the proof and can be skipped.

D.4 Alternative Perspective by PEP

The goal of this section is to clearly demonstrate the advancements of our analysis over existing works related to the PEP. In this section, first, we review the performance estimation problem (PEP). Then, we show that the optimization problem in (61) is a relaxation of a generalized PEP, tailored to decentralized constrained optimization algorithms capable of handling temporal distortion. This relaxation involves a quadratic objective function and linear constraints; hence, we term it LQ-PEP. To the best of our knowledge, this is the first time such a generalized and complex PEP has been introduced and solved, marking a substantial contribution to the literature. In this regard, Section E, where we solve this LQ-PEP, holds independent value for researchers working on PEP problems.

D.4.1 Performance estimation problem

PEP evaluates the worst-case performance of an optimization method $\mathcal{M}$ after K iterations, which is formulated as the following optimization problem (Taylor, 2017):

$$\begin{aligned}
\max_{f, \mathbf{x}_*, \mathbf{x}_0, \ldots, \mathbf{x}_K} \quad & \mathcal{P}(\mathcal{O}_f,\ \mathbf{x}_*,\ \mathbf{x}_0,\ \ldots,\ \mathbf{x}_K) \qquad (62)\\
\text{s.t.} \quad & \mathbf{x}_1,\ \ldots,\ \mathbf{x}_K, \text{ are generated by method } \mathcal{M} \text{ from } \mathbf{x}_0,\\
& \mathbf{x}_* \text{ is the minimizer of } f,\\
& f \text{ is a function in family } \mathcal{F},\\
& \|\mathbf{x}_0 - \mathbf{x}_*\|^2 \le R.
\end{aligned}$$

Here, $\mathcal{P}$ is a positive definite function quantifying the performance of $\mathcal{M}$. Assuming $\mathcal{M}$ belongs to the class of first-order optimization methods, $\mathcal{O}_f$ indicates the first-order information computed at the algorithm's iterates $\mathbf{x}_0, \ldots, \mathbf{x}_K$, and $\mathcal{F}$ represents a family of functions to which f belongs. R limits the starting point to not be far away from the optimal point.

D.4.2 Optimization problem in (62) versus LQ-PEP (61)

Although the PEP has been extensively studied for analyzing the worst case performance of many optimization algorithms, it has not been introduced for decentralized algorithms capable of handling local constraints and temporal distortion. If we look at the optimization problem in (61) more closely, we observe that it is closely related to a relaxation of (62) for ASY-DAGP by taking into account that:

1. PEP in (62) assumes that $\mathcal{M}$ iteratively updates only one optimization variable $\mathbf{x}$, which is typical in centralized optimization algorithms. Moreover, it assumes that there are no additional variables being updated besides $\mathbf{x}$. However, when solving decentralized constrained optimization problems over directed graphs, there are M nodes, and each node may update a set of variables, eg. $\{\mathbf{x}^v, \mathbf{g}^v, \mathbf{h}^v, \mathbf{p}^v\}$ for ASY-DAGP. Therefore, one can develop a *generalized PEP* to compute the worst-case performance of decentralized methods by replacing $\mathbf{x}$ with the set of all variables being updated by the algorithm.

2. Since we have simplified the algorithm dynamics by shifting the origin to the optimal solution, the second constraint in (62) can be removed, and all expressions in the objective and constraints must be rewritten using the shifted variables. Moreover, the optimal solution is not the optimization variable anymore.

3. Considering Item 1, the first constraint of (62) is equivalent to requiring that the variables $\{\tilde{\mathbf{g}}^v, \tilde{\mathbf{h}}^v, \tilde{\mathbf{p}}^v\}$ satisfy the dynamics of ASY-DAGP given in Section C, which are the first three constraints in (61).

4. Taking Items 1 and 2 into account, the last constraint of (62) is equivalent to requiring that the variables $\{\tilde{\mathbf{x}}^v, \tilde{\mathbf{g}}^v, \tilde{\mathbf{h}}^v, \tilde{\mathbf{p}}^v\}$ are bounded, ie. the last constraint in (61).

5. To finalize the connection between (62) and (61), we need to clarify the relationship between their objectives, as well as the connection with the third constraint in (62).

   Taylor (2017) introduced a set of interpolation inequalities that can replace the third constraint in (62) for different classes of functions. In the derivation of (60), which is in the abstract form of (50), we used a subset of these interpolation inequalities (what we referred to as elementary inequalities) for the class of convex smooth functions with closed convex constraints.

   To relate (60), or equivalently (50), to PEP in (62), we observe that $\sum_k \mathcal{P}_k$ is a positive definite function measuring the performance of ASY-DAGP. Therefore, it can serve as the objective function in (62). From (50), we notice that any lower bound on the aggregate term $\sum_k \mathcal{A}_k$ implies an upper bound on $\sum_k \mathcal{P}_k$. Hence, the minimization of the aggregate term over all possible dynamics becomes a relaxation of (62). As mentioned earlier, several elementary inequalities were considered in deriving (60), so the third constraint of (62) has already been addressed in (61).

D.4.3 Advantages of LQ-PEP

In this section, we clarify the advantages of LQ-PEP compared to PEP-based and Lyapunov-type analyses:

- First, our approach formulates a search problem for the worst instance of the problem ASY-DAGP solves. This is similar to PEP, but unlike PEP, which is completely automatic, we still require hand-crafted steps described in Section D.2. In other words, we consider a hand-crafted relaxation of PEP, called LQ-PEP. The main benefit of this approach is that unlike PEP, which is generally applicable to a limited number of iterations, our method provides convergence bounds for large and infinite number of iterations (asymptotic analysis). Despite the hand-crafted steps, we believe that our analysis approach presented in summary in Section D.1 can serve as a blueprint for convergence analysis in many other optimization algorithms.

- The second major benefit of LQ-PEP is in problems suffering from non-stationary perturbation, such as delays and asynchrony in our case. In these scenarios, finding a Lyapunov function can be extremely difficult, if not impossible. LQ-PEP gives a systematic way to calculate a metric for the amount of perturbation (like delay factor in our case) and provides conditions where a limited amount of perturbation provides similar speeds of convergence to an unperturbed system, though with different constants.

- Third, LQ-PEP gives a new perspective of first-order methods through the lens of linear systems. Analysis by LQ-PEP always leads to a characterization of a given algorithm by a matrix-valued function on the complex plane, called forward-backward transfer function. The poles (zeros of the determinate) of this function can be interpreted as the *"excitation modes"* of the algorithm. These poles characterize how the error introduced in an iteration is propagated throughout the algorithm. This will be clarified in Section E.

- Our derivation follows the methodology of Shahriari-Mehr and Panahi (2022), however, our work is the first to study an algorithm under non-stationary perturbations (temporal distortion). In our work, we have a crucial additional innovation. In Shahriari-Mehr and Panahi (2022), the right hand sides of equations (67), (68), and (69) will be zero, making it straightforward—via the $z-$transform—to show that the system has no solution. This is not the case for us. To resolve this issue, we propose the novel intrinsic basis in Section E.2.1 and the representation of this system of equations in Section E.2.2 to arrive at a novel contraction argument in Section E.2.3.

- Finally, there are close ties to Lyapunov-type analysis. As explained in Section D.1, we need to design an optimality metric $\mathcal{P}_k$ which resembles the process of designing a Lyapunov function, but $\mathcal{P}_k$ is not associated with a Lyapunov function. In a Lyapunov analysis, we need to find a positive definite Lyapunov function L satisfying: $L_{k+1} - L_k + \mathcal{P}_k \leq 0$, but we do not find such a function and we are not sure that under delays and asynchrony, such a function can be easily found by elementary operations. However, LQ-PEP establishes that $\sum_k \mathcal{P}_k$ is bounded, which is also a step in a Lyapunov-type analysis. As a result, LQ-PEP essentially bypasses the design of a Lyapunov function in a Lyapunov-type analysis.

E Solving LQ-PEP

E.1 Deriving optimality conditions of (61)

To avoid tedious calculations, we begin by introducing an abstract notation and solving (61) accordingly. This approach results in abstract assumptions. In Section E.3, we present and partially verify these assumptions in an explicit form.

The LQ-PEP optimization problem in (61) can be reformulated as an abstract optimization problem by defining a state vector $\mathbf{\Psi}_k$ containing all the variables being updated throughout the algorithm. The ASY-DAGP state vectors $\mathbf{\Psi}_k \in \mathbb{R}^{5M\times m}$ are defined as follows

$$\mathbf{\Psi}_k = \left[\mathbf{X}_{k+1}^T \;\; \mathbf{X}_k^T \;\; \mathbf{G}_k^T \;\; \mathbf{H}_k^T \;\; \mathbf{P}_k^T\right]^T. \tag{63}$$

In this notation, each block (e.g. $\mathbf{G}_k$) contains all the vectors of different nodes (e.g. $\mathbf{g}^v$) as its rows, with $v \in 1,\ldots,M$. We also introduce the shifted states $\tilde{\mathbf{\Psi}}_k = \mathbf{\Psi}_k - \mathbf{\Psi}_*$, where $\mathbf{\Psi}_*$ contains the optimal states. By this convention, we can describe the evolution of ASY-DAGP's variables, which are the constrains in (61), using the following linear dynamical system

$$\tilde{\mathbf{\Psi}}_{k+1} - \bar{\mathbf{R}}\tilde{\mathbf{\Psi}}_k - \mathbf{P}\tilde{\mathbf{U}}_k = \sum_{l=0}^{K-1} \tilde{\mathbf{R}}_{k,k-l}\tilde{\mathbf{\Psi}}_l, \qquad k = 0,\ldots,K-1 \tag{64}$$

where $\bar{\mathbf{R}}$, $\tilde{\mathbf{R}}_{k,k-l}$, and $\mathbf{P}$ matrices are defined in (161), (162) and (163), respectively, and $\tilde{\mathbf{U}}_k := \tilde{\mathbf{X}}_{k+2}$ represents the input of the system. In developing the dynamical system in (64), specifically through the creation of $\bar{\mathbf{R}}$ and $\tilde{\mathbf{R}}_{k,k-l}$ matrices, we divide the system into two components. The first one contains the ideal scenario, which means nodes compute at the same rate with no communication delays. The second component compensates for errors stemming from our imperfect setup, notably temporal distortion, represented by the term on the right hand side of (64). So, by setting the right-hand side of (64) to zero, the system represents the dynamics of the synchronous non-delayed setup. Although our analysis primarily addresses temporal distortion, the analysis is adaptable to various irregularities, e.g. quantization, thereby broadening the applicability of our analysis.

By this convention, we can also replace the objective terms in (61), ie. the aggregate terms $\sum_k \mathcal{A}_k$, by the following quadratic term

$$\sum_k \mathcal{A}_k = \sum_{k=0}^{K-1} \langle \tilde{\mathbf{\Psi}}_k, \bar{\mathbf{S}}\tilde{\mathbf{\Psi}}_k\rangle + \sum_{k,l=0}^{K-1} \langle \tilde{\mathbf{\Psi}}_k, \tilde{\mathbf{S}}_{k,k-l}\tilde{\mathbf{\Psi}}_l\rangle, \tag{65}$$

where $\bar{\mathbf{S}}$ and $\tilde{\mathbf{S}}_{k,k-l}$ are defined in (164) and (165), respectively. Considering the abstract linear dynamical system as the constraint and the abstract quadratic form of the aggregate term, the abstract LQ-PEP optimization can be written as follows:

$$\begin{aligned}
\min_{\{\tilde{\mathbf{\Psi}}_k\}_{k=0}^{K-1}, \{\tilde{\mathbf{U}}_k\}_{k=0}^{K-2}} \quad & \frac{1}{2}\sum_{k=0}^{K-1} \langle \tilde{\mathbf{\Psi}}_k, \mathbf{S}\tilde{\mathbf{\Psi}}_k\rangle + \frac{1}{2}\sum_{k,l=0}^{K-1} \langle \tilde{\mathbf{\Psi}}_k, \tilde{\mathbf{S}}_{k,k-l}\tilde{\mathbf{\Psi}}_l\rangle + \frac{C}{2}\|\tilde{\mathbf{\Psi}}_0\|_{\mathrm{F}}^2 \\
\text{subject to} \quad & \tilde{\mathbf{\Psi}}_{k+1} - \bar{\mathbf{R}}\tilde{\mathbf{\Psi}}_k - \mathbf{P}\tilde{\mathbf{U}}_k = \sum_{l=0}^{K-1} \tilde{\mathbf{R}}_{k,k-l}\tilde{\mathbf{\Psi}}_l, \qquad k = 0,\ldots,K-1 \\
& \frac{1}{2}\|\tilde{\mathbf{\Psi}}_0\|_{\mathrm{F}}^2 + \frac{1}{2}\sum_{k=0}^{K-2} \|\tilde{\mathbf{U}}_k\|_{\mathrm{F}}^2 \le \frac{1}{2},
\end{aligned} \tag{66}$$

where we introduced two modifications: first, we wrote the last constraint of (61) in a Lagrangian dual form with the additional term $\frac{C}{2}\|\tilde{\mathbf{\Psi}}_0\|_{\mathrm{F}}^2$. Second, we restricted, the problem to a sphere given by $\frac{1}{2}\sum_{k=0}^{K-2}\|\tilde{\mathbf{U}}_k\|_{\mathrm{F}}^2 \le \frac{1}{2}$. Now, we note that two different cases may occur: either the optimal value is 0, in which case $B = \frac{C}{2}\|\tilde{\mathbf{\Psi}}_0\|_{\mathrm{F}}^2$ is an upper bound for $-\sum_k \mathcal{A}_k$ or there exists an optimal point on the sphere with a negative objective value. We simply need to verify that the second case does not occur. To demonstrate this, we introduce the dual Lagrangian multipliers $\mathbf{\Lambda}_k$ and $\beta \geq 0$ and write the local optimality condition as below:

$$\tilde{\mathbf{\Psi}}_{k+1} - \bar{\mathbf{R}}\tilde{\mathbf{\Psi}}_k - \mathbf{P}\tilde{\mathbf{U}}_k = \sum_{l=0}^{K-1} \tilde{\mathbf{R}}_{k,k-l}\tilde{\mathbf{\Psi}}_l \qquad k = 0, \dots, K-1 \tag{67}$$

$$\mathbf{\Lambda}_{k-1} - \bar{\mathbf{R}}^T\mathbf{\Lambda}_k + \bar{\mathbf{S}}\tilde{\mathbf{\Psi}}_k = \sum_{l=0}^{K-2} \tilde{\mathbf{R}}^T_{l,l-k}\mathbf{\Lambda}_l + \sum_{l=0}^{K-2} \left(\tilde{\mathbf{S}}_{k,k-l} + \tilde{\mathbf{S}}^T_{l,l-k}\right)\mathbf{\Psi}_l \qquad k = 0, \dots, K-1 \tag{68}$$

$$\beta\tilde{\mathbf{U}}_k - \mathbf{P}^T\mathbf{\Lambda}_k = \mathbf{O} \qquad k = 0, \dots, K-1 \tag{69}$$

with boundary conditions

$$\mathbf{\Lambda}_{-1} = (\beta + C)\tilde{\mathbf{\Psi}}_0, \tag{70}$$

$$\mathbf{\Lambda}_{K-1} = \mathbf{O}. \tag{71}$$

After some simplifications and substitutions, we notice that the optimal value of the optimization problem in (66) is $-\beta\left(\|\mathbf{\Psi}_0\|_\text{F}^2 + \sum_{k=0}^{K-2}\|\tilde{\mathbf{U}}_k\|_\text{F}^2\right)$. Since β is positive, the optimal solution is negative unless the system of linear recurrences in (67), (68), and (69) has no non-zero solution.

E.2 Solution by $z-$transform

In summary, if we show the system of linear recurrences in (67), (68), and (69) with boundary conditions in (70) and (71) has no non-zero solution, zero will be the optimal value of the optimization problem in (66), consequently, $\sum_k \mathcal{A}_k$ has a lower bound $-C\|\tilde{\mathbf{\Psi}}_0\|_\text{F}^2$, and hence, the statements of Proposition 2 hold true with $B = C\|\tilde{\mathbf{\Psi}}_0\|_\text{F}^2 := CC_0$.

To show this, first, we remove the dependence on $\tilde{\mathbf{U}}_k$. Then, we have the new system of linear recurrences

$$\tilde{\mathbf{\Psi}}_{k+1} - \bar{\mathbf{R}}\tilde{\mathbf{\Psi}}_k - \frac{1}{\beta}\mathbf{P}\mathbf{P}^T\mathbf{\Lambda}_k = \sum_{l=0}^{K-1} \tilde{\mathbf{R}}_{k,k-l}\tilde{\mathbf{\Psi}}_l \qquad k = 0, \dots, K-1 \tag{72}$$

$$\mathbf{\Lambda}_{k-1} - \bar{\mathbf{R}}^T\mathbf{\Lambda}_k + \bar{\mathbf{S}}\tilde{\mathbf{\Psi}}_k = \sum_{l=0}^{K-1} \tilde{\mathbf{R}}^T_{l,l-k}\mathbf{\Lambda}_l + \sum_{l=0}^{K-1} \left(\tilde{\mathbf{S}}_{k,k-l} + \tilde{\mathbf{S}}^T_{l,l-k}\right)\tilde{\mathbf{\Psi}}_l \qquad k = 0, \dots, K-1 \tag{73}$$

By defining the finite duration $z-$transforms of $\tilde{\mathbf{\Psi}}_k$, and $\mathbf{\Lambda}_k$ as

$$\tilde{\mathbf{\Psi}}(z) = \sum_{k=0}^{K} \tilde{\mathbf{\Psi}}_k z^k, \qquad \mathbf{\Lambda}(z) = \sum_{k=-1}^{K-1} \mathbf{\Lambda}_k z^{k+1}, \tag{74}$$

and taking the $z-$transform of (72) and (73), we have

$$\begin{bmatrix} \tilde{\mathbf{\Psi}}(z) \\ \mathbf{\Lambda}(z) \end{bmatrix} = \mathbf{F}_\beta^{-1}(z) \begin{bmatrix} \mathbf{S}\tilde{\mathbf{\Psi}}_K z^K + \mathbf{\Lambda}_{K-1}z^K - z^{-1}\bar{\mathbf{R}}^T\mathbf{\Lambda}_{-1} \\ \tilde{\mathbf{\Psi}}_0 - \bar{\mathbf{R}}\tilde{\mathbf{\Psi}}_K z^{K+1} - \frac{1}{\beta}\mathbf{P}\mathbf{P}^T\mathbf{\Lambda}_{-1} \end{bmatrix}$$
$$+ \mathbf{F}_\beta^{-1}(z) \begin{bmatrix} \sum_{k,l=0}^{K-1} \tilde{\mathbf{R}}^T_{l,l-k}\mathbf{\Lambda}_l z^k + \sum_{k,l=0}^{K-1}\left(\tilde{\mathbf{S}}_{k,k-l} + \tilde{\mathbf{S}}^T_{l,l-k}\right)\tilde{\mathbf{\Psi}}_l z^k \\ \sum_{k,l=0}^{K-1} \tilde{\mathbf{R}}_{k,k-l}\tilde{\mathbf{\Psi}}_l z^{k+1} \end{bmatrix} \tag{75}$$

where

$$\mathbf{F}_\beta(z) = \begin{bmatrix} \bar{\mathbf{S}} & \mathbf{I} - z^{-1}\bar{\mathbf{R}}^T \\ \mathbf{I} - z\bar{\mathbf{R}} & -\frac{1}{\beta}\mathbf{P}\mathbf{P}^T \end{bmatrix}. \tag{76}$$

Using (70) and (71), and considering $\mathbf{F}_\beta(z)\mathbf{F}_\beta^{-1}(z) = \mathbf{I}$ and $\mathbf{F}_\beta^{-1}(z)\mathbf{F}_\beta(z) = \mathbf{I}$, we obtain

$$\mathbf{F}_\beta^{-1}(z)\begin{bmatrix}\mathbf{S}\tilde{\boldsymbol{\Psi}}_K z^K + \boldsymbol{\Lambda}_{K-1}z^K - z^{-1}\bar{\mathbf{R}}^T\boldsymbol{\Lambda}_{-1}\\ \tilde{\boldsymbol{\Psi}}_0 - \bar{\mathbf{R}}\tilde{\boldsymbol{\Psi}}_K z^{K+1} - \frac{1}{\beta}\mathbf{P}\mathbf{P}^T\boldsymbol{\Lambda}_{-1}\end{bmatrix} = \tag{77}$$

$$z^K\left(\begin{bmatrix}\mathbf{I}\\ \mathbf{O}\end{bmatrix} - \mathbf{F}_\beta^{-1}(z)\begin{bmatrix}\mathbf{O}\\ \mathbf{I}\end{bmatrix}\right)\tilde{\boldsymbol{\Psi}}_K + \left(\begin{bmatrix}\mathbf{O}\\ \mathbf{I}\end{bmatrix} - \mathbf{F}_\beta^{-1}(z)\begin{bmatrix}\mathbf{I}\\ (\beta+C)^{-1}\mathbf{I}\end{bmatrix}\right)\boldsymbol{\Lambda}_{-1} \tag{78}$$

From the definition of inverse $z-$transform, we have

$$\tilde{\boldsymbol{\Psi}}_m = \frac{1}{2\pi j}\oint z^{-(m+1)}\tilde{\boldsymbol{\Psi}}(z)\mathrm{d}z, \tag{79}$$

$$\boldsymbol{\Lambda}_{m-1} = \frac{1}{2\pi j}\oint z^{-(m+1)}\boldsymbol{\Lambda}(z)\mathrm{d}z. \tag{80}$$

Hence, we have

$$\begin{aligned}\begin{bmatrix}\tilde{\boldsymbol{\Psi}}_m\\ \boldsymbol{\Lambda}_{m-1}\end{bmatrix} &= \begin{bmatrix}\tilde{\boldsymbol{\Psi}}_K\delta_{m-K}\\ \boldsymbol{\Lambda}_{-1}\delta_m\end{bmatrix} - \frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1}(z)z^{-(m+1)}\mathrm{d}z\begin{bmatrix}\mathbf{I}\\ (\beta+C)^{-1}\mathbf{I}\end{bmatrix}\boldsymbol{\Lambda}_{-1}\\ &\quad - \frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1}(z)z^{K-(m+1)}\mathrm{d}z\begin{bmatrix}\mathbf{O}\\ \mathbf{I}\end{bmatrix}\tilde{\boldsymbol{\Psi}}_K + \begin{bmatrix}\bar{\boldsymbol{\Psi}}_m\\ \bar{\boldsymbol{\Lambda}}_{m-1}\end{bmatrix},\end{aligned} \tag{81}$$

where

$$\begin{bmatrix}\bar{\boldsymbol{\Psi}}_m\\ \bar{\boldsymbol{\Lambda}}_{m-1}\end{bmatrix} = \frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1}(z)z^{-(m+1)}\begin{bmatrix}\sum_{k,l=0}^{K-1}\tilde{\mathbf{R}}^T_{l,l-k}\boldsymbol{\Lambda}_l z^k + \sum_{k,l=0}^{K-1}\left(\tilde{\mathbf{S}}_{k,k-l} + \tilde{\mathbf{S}}^T_{l,l-k}\right)\tilde{\boldsymbol{\Psi}}_l z^k\\ \sum_{k,l=0}^{K-1}\tilde{\mathbf{R}}_{k,k-l}\tilde{\boldsymbol{\Psi}}_l z^{k+1}\end{bmatrix}\mathrm{d}z \tag{82}$$

Boundary conditions: Setting, $m = 0, K$ gives

$$\begin{aligned}\begin{bmatrix}\tilde{\boldsymbol{\Psi}}_0\\ \mathbf{O}\end{bmatrix} &= -\frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1}(z)z^{-1}\mathrm{d}z\begin{bmatrix}\mathbf{I}\\ (\beta+C)^{-1}\mathbf{I}\end{bmatrix}\boldsymbol{\Lambda}_{-1}\\ &\quad - \frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1}(z)z^{K-1}\mathrm{d}z\begin{bmatrix}\mathbf{O}\\ \mathbf{I}\end{bmatrix}\tilde{\boldsymbol{\Psi}}_K + \begin{bmatrix}\bar{\boldsymbol{\Psi}}_0\\ \bar{\boldsymbol{\Lambda}}_{-1}\end{bmatrix},\end{aligned} \tag{83}$$

and

$$\begin{aligned}\begin{bmatrix}\mathbf{O}\\ \boldsymbol{\Lambda}_{K-1}\end{bmatrix} &= -\frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1)}(z)z^{-(K+1)}\mathrm{d}z\begin{bmatrix}\mathbf{I}\\ (\beta+C)^{-1}\mathbf{I}\end{bmatrix}\boldsymbol{\Lambda}_{-1}\\ &\quad - \frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1}(z)z^{-1}\mathrm{d}z\begin{bmatrix}\mathbf{O}\\ \mathbf{I}\end{bmatrix}\tilde{\boldsymbol{\Psi}}_K + \begin{bmatrix}\bar{\boldsymbol{\Psi}}_K\\ \bar{\boldsymbol{\Lambda}}_{K-1}\end{bmatrix},\end{aligned} \tag{84}$$

Next, according to the properties of the forward-backward matrix presented in Section E.2.1, together with the result of Lemma 1, and under the assumption that no pole is on the unit circle, we may neglect the terms $\frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1}(z)z^{K-1}\mathrm{d}z$ and $\frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1}(z)z^{-(K+1)}\mathrm{d}z$, for sufficiently large K. Hence, by defining

$$\boldsymbol{\Phi}_\beta := \frac{1}{2\pi j}\oint \mathbf{F}_\beta^{-1}(z)z^{-1}\mathrm{d}z, \tag{85}$$

we have

$$\begin{bmatrix} \tilde{\mathbf{\Psi}}_0 \\ \mathbf{O} \end{bmatrix} = -\mathbf{\Phi}_\beta \begin{bmatrix} \mathbf{I} \\ (\beta + C)^{-1}\mathbf{I} \end{bmatrix} \mathbf{\Lambda}_{-1} + \begin{bmatrix} \bar{\mathbf{\Psi}}_0 \\ \bar{\mathbf{\Lambda}}_{-1} \end{bmatrix}, \tag{86}$$

and

$$\begin{bmatrix} \mathbf{O} \\ \mathbf{\Lambda}_{K-1} \end{bmatrix} = -\mathbf{\Phi}_\beta \begin{bmatrix} \mathbf{O} \\ \mathbf{I} \end{bmatrix} \tilde{\mathbf{\Psi}}_K + \begin{bmatrix} \bar{\mathbf{\Psi}}_K \\ \bar{\mathbf{\Lambda}}_{K-1} \end{bmatrix}. \tag{87}$$

Next, we invoke the boundary conditions in (70) and (71) and replace $\mathbf{\Lambda}_{K-1} = \mathbf{O}$ and $\mathbf{\Lambda}_{-1} = (\beta + C)\tilde{\mathbf{\Psi}}_0$, which leads to

$$\left(\begin{bmatrix} \frac{1}{\beta+C}\mathbf{I} \\ \mathbf{O} \end{bmatrix} + \mathbf{\Phi}_\beta \begin{bmatrix} \mathbf{I} \\ \frac{1}{\beta+C}\mathbf{I} \end{bmatrix} \right) \mathbf{\Lambda}_{-1} = \begin{bmatrix} \bar{\mathbf{\Psi}}_0 \\ \bar{\mathbf{\Lambda}}_{-1} \end{bmatrix}, \tag{88}$$

and

$$\mathbf{\Phi}_\beta \begin{bmatrix} \mathbf{O} \\ \mathbf{I} \end{bmatrix} \tilde{\mathbf{\Psi}}_K = \begin{bmatrix} \bar{\mathbf{\Psi}}_K \\ \bar{\mathbf{\Lambda}}_{K-1} \end{bmatrix}. \tag{89}$$

E.2.1 Properties of the Forward-Backward transfer matrix

Before we proceed with the proof, we need to establish some key properties of the forward-backward transfer matrix. Remember that it is defined as

$$\mathbf{F}_\beta(z) = \begin{bmatrix} \mathbf{S} & \mathbf{I} - z^{-1}\bar{\mathbf{R}}^T \\ \mathbf{I} - z\bar{\mathbf{R}} & -\frac{1}{\beta}\mathbf{P}\mathbf{P}^T \end{bmatrix}. \tag{90}$$

We henceforth drop the index β for simplicity. Moreover, for convenience we re-scale this matrix and define

$$\mathbf{H}(z) := \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & z^{-1}\mathbf{I} \end{bmatrix} \mathbf{F}(z) \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & z\mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{S} & z\mathbf{I} - \bar{\mathbf{R}}^T \\ z^{-1}\mathbf{I} - \bar{\mathbf{R}} & -\frac{1}{\beta}\mathbf{P}\mathbf{P}^T \end{bmatrix}. \tag{91}$$

Then, we have

$$\mathbf{F}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & z\mathbf{I} \end{bmatrix} \mathbf{H}(z) \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & z^{-1}\mathbf{I} \end{bmatrix} \tag{92}$$

and

$$\mathbf{F}^{-1}(z) = \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & z\mathbf{I} \end{bmatrix} \mathbf{H}^{-1}(z) \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & z^{-1}\mathbf{I} \end{bmatrix}. \tag{93}$$

By investigating the highest and lowest powers of z in the determinant and cofactors of $\mathbf{H}(z)$, it is simple to see that as long as the roots of $\det(\mathbf{H}) = \det(\mathbf{F})$ are simple, it has the following expansion:

$$\mathbf{H}^{-1}(z) = \mathbf{H}_\infty + \sum_p \frac{\mathbf{H}_p}{z - z_p}, \tag{94}$$

where z_p is the root of $\det(\mathbf{H}) = \det(\mathbf{F})$, and $\mathbf{H}_\infty, \{\mathbf{H}_p\}$ are fixed matrices. Now, note that $\mathbf{F}^T(z) = \mathbf{F}(z^{-1})$ and $\mathbf{H}^T(z) = \mathbf{H}(z^{-1})$. We call this property *fundamental symmetry.* Hence, z_p^{-1} is also a root of the determinant. We index the roots such that $z_{-p} = z_p^{-1}$ for $p = 1, 2, \dots, P$ (where p cannot be 0) and $|z_p| \leq 1$ for $p > 0$.

Let us review some basic properties of the above expansion. According to the fundamental symmetry, we have

$$\mathbf{H}_\infty^T + \sum_p \frac{\mathbf{H}_p^T}{z - z_p} = \mathbf{H}_\infty + \sum_p \frac{\mathbf{H}_p}{z^{-1} - z_p} = \mathbf{H}_\infty + \sum_p \frac{\mathbf{H}_{-p}}{z^{-1} - z_p^{-1}}, \tag{95}$$

which yields

$$\mathbf{H}_\infty - \mathbf{H}_\infty^T = \sum_p z_p^{-1}\mathbf{H}_p, \quad \mathbf{H}_p^T = -z_p^2\mathbf{H}_{-p}. \tag{96}$$

Next, we delve deeper into the properties of the coefficient matrices. Note that

$$\mathbf{H}_p = \lim_{z\to z_p} \mathbf{H}^{-1}(z)(z-z_p). \tag{97}$$

We conclude that

$$\mathbf{H}_p\mathbf{H}(z_p) = \mathbf{H}(z_p)\mathbf{H}_p = \lim_{z\to z_p}(z-z_p)\mathbf{I} = \mathbf{O}. \tag{98}$$

Note that $\mathbf{H}(z_p)$ is rank-deficient, and when the roots z_p of $\det(\mathbf{H}(z))$ are simple, $\mathbf{H}(z_p)$ does not have multiple zero singular values. This is seen, for example, by noting that the derivative of $\det(\mathbf{H}(z))$ at z_p is nonzero. Hence, $\ker(\mathbf{H}(z_p))$ and $\ker(\mathbf{H}^T(z_p))$ are single-dimensional. We take arbitrary normalized nonzero vectors $\mathbf{a}_p, \mathbf{b}_p$ in these kernels, respectively. Now, according to (98), we have

$$\mathbf{H}_p = \alpha_p\mathbf{a}_p\mathbf{b}_p^T, \tag{99}$$

where α_p is a scalar. According to (96), we further choose these vectors such that $(\mathbf{a}_{-p}, \mathbf{b}_{-p}) = (\mathbf{b}_p, \mathbf{a}_p)$.

Orthogonality: Now, we make the following observation:

$$\mathbf{H}(z) = \bar{\mathbf{H}} + \begin{bmatrix} \mathbf{O} & z\mathbf{I} \\ z^{-1}\mathbf{I} & \mathbf{O} \end{bmatrix}, \quad \bar{\mathbf{H}} := \begin{bmatrix} \mathbf{S} & -\bar{\mathbf{R}}^T \\ -\bar{\mathbf{R}} & -\frac{1}{\beta}\mathbf{P}\mathbf{P}^T \end{bmatrix} \tag{100}$$

Hence, we have

$$\mathbf{H}(z)\mathbf{H}_p = (\mathbf{H}(z) - \mathbf{H}(z_p))\mathbf{H}_p = (z-z_p)\begin{bmatrix} \mathbf{O} & \mathbf{I} \\ -z^{-1}z_p^{-1}\mathbf{I} & \mathbf{O} \end{bmatrix}\mathbf{H}_p. \tag{101}$$

Multiplying by $\mathbf{H}^{-1}(z)$ and taking the limit $z \to z_p$, we have

$$\mathbf{H}_p = \mathbf{H}_p\begin{bmatrix} \mathbf{O} & \mathbf{I} \\ -z_p^{-2}\mathbf{I} & \mathbf{O} \end{bmatrix}\mathbf{H}_p. \tag{102}$$

In particular, we get

$$\alpha_p^{-1} = \mathbf{b}_p^T\begin{bmatrix} \mathbf{O} & \mathbf{I} \\ -z_p^{-2}\mathbf{I} & \mathbf{O} \end{bmatrix}\mathbf{a}_p. \tag{103}$$

Moreover, we have

$$\mathbf{O} = \mathbf{H}_p(\mathbf{H}(z_p) - \mathbf{H}(z_q))\mathbf{H}_q = \mathbf{H}_p\begin{bmatrix} \mathbf{O} & (z_p - z_q)\mathbf{I} \\ (z_p^{-1} - z_q^{-1})\mathbf{I} & \mathbf{O} \end{bmatrix}\mathbf{H}_q, \tag{104}$$

and for $p \neq q$ we have

$$\mathbf{O} = \mathbf{H}_p\begin{bmatrix} \mathbf{O} & -z_pz_q\mathbf{I} \\ \mathbf{I} & \mathbf{O} \end{bmatrix}\mathbf{H}_q = \mathbf{H}_p\begin{bmatrix} \mathbf{O} & -z_p\mathbf{I} \\ \mathbf{I} & \mathbf{O} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \mathbf{O} & z_q\mathbf{I} \end{bmatrix}\mathbf{H}_q. \tag{105}$$

The above discussion can be summarized as the following orthogonality result:

$$\mathbf{b}_p^T\begin{bmatrix} \mathbf{O} & \mathbf{I} \\ -z_p^{-1}\mathbf{I} & \mathbf{O} \end{bmatrix}\begin{bmatrix} z_q^{-1}\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix}\mathbf{a}_q = \delta_{p,q}\alpha_q^{-1} \tag{106}$$

Accordingly, we also define

$$\mathbf{A}_q := \alpha_q \begin{bmatrix} z_q^{-1}\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \mathbf{a}_q, \quad \mathbf{B}_p := \begin{bmatrix} \mathbf{O} & -z_p^{-1}\mathbf{I} \\ \mathbf{I} & \mathbf{O} \end{bmatrix} \mathbf{b}_p, \tag{107}$$

and the matrices $\mathbf{A}, \mathbf{B}$ with $\mathbf{A}_q, \mathbf{B}_p^T$ as its q^{th} column and p^{th} row, respectively. Then, we see that $\mathbf{BA} = \mathbf{I}$, and hence $\mathbf{B} = \mathbf{A}^{-1}$.

Eliminating $\mathbf{H}_\infty$ and intrinsic basis: Similarly to the above discussion, by letting $z \to \infty$, we observe that

$$\begin{bmatrix} \mathbf{O} & \mathbf{I} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \mathbf{H}_\infty = \mathbf{H}_\infty \begin{bmatrix} \mathbf{O} & \mathbf{I} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} = \mathbf{O} \tag{108}$$

Replacing this result in (96) yields

$$\mathbf{H}_\infty = \sum_p \begin{bmatrix} z_p^{-1}\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \mathbf{H}_p. \tag{109}$$

Hence, we have

$$\mathbf{H}^{-1}(z) = \sum_p \begin{bmatrix} z_p^{-1}z\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \frac{\mathbf{H}_p}{z - z_p}, \tag{110}$$

and

$$\mathbf{F}^{-1}(z) = \sum_p \begin{bmatrix} z_p^{-1}\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \frac{\mathbf{H}_p}{z - z_p} \begin{bmatrix} z\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{bmatrix}. \tag{111}$$

This gives an expansion of the inverse forward-backward matrix in a particular basis, which we call intrinsic basis, consisting of the vectors $\mathbf{A}_p$. Note that, based on the above, we conclude that $\mathbf{A}_p$ is complete and independent, and hence forms a basis set. We conclude this part with a key result on contour integrals:

Lemma 1 (Contour integral over the unit circle)**.** *Let $f(z) = \frac{z^k}{z - z_p}$. The contour integral of $f(z)$ over the unit circle Γ is given by*

$$\psi_p(k) := \frac{1}{2\pi j} \oint_\Gamma \frac{z^k}{z - z_p} \mathrm{d}z = \begin{cases} z_p^k & \textit{if } |z_p| < 1, k \geq 0 \\ -z_p^k & \textit{if } |z_p| > 1, k < 0 \\ 0 & \textit{otherwise.} \end{cases} \tag{112}$$

Proof. To compute the integral, we consider the following four cases:

- $|z_p| > 1, k \geq 0$: The function is analytic inside the contour. Based on the Cauchy integral theorem, the integral equals 0.
- $|z_p| < 1, k \geq 0$: $z = z_p$ is the only singular point of $f(z)$ inside the contour. Based on the residue theorem, the integral equals z_p^k.
- $|z_p| > 1, k < 0$: $z = 0$ is the only singularity of order k of $f(z)$. The residue at $z = 0$ is computed as

  $$\operatorname{Res}[f(z), 0] = \lim_{z \to 0} \frac{1}{(-k-1)!} \frac{d^{-k-1}}{dz^{-k-1}} \left[\frac{1}{z - z_p}\right] = -z_p^k$$

  Based on the Residue theorem, the solution of the integral is $-z_p^k$.
- $|z_p| < 1, k < 0$: The function $f(z)$ has two singularities at $z = 0$ and $z = z_p$. Based on the Residue theorem, considering that the residues cancel each other out, the integral equals 0.

Combining the above cases, we obtain the solution to the contour integral as given in (112). ■

E.2.2 Representation in the intrinsic basis

Now, we use the properties presented in Section E.2.1 to simplify the relations in (81), (82) and the boundary conditions in (88), (89). We use the notations $\mathbf{H}_p, \mathbf{a}_p, \mathbf{b}_p, \alpha_p, \ldots$ introduced in Section E.2.1, and the results in (111) and (106). Let us start by (82). To simplify the notation, we introduce

$$\mathbf{e}_k := \begin{bmatrix} \sum_{l=0}^{K-1} \tilde{\mathbf{R}}_{l,l-k}^T \mathbf{\Lambda}_l + \sum_{l=0}^{K-1} \left(\tilde{\mathbf{S}}_{k,k-l} + \tilde{\mathbf{S}}_{l,l-k}^T \right) \tilde{\mathbf{\Psi}}_l \\ \\ \sum_{l=0}^{K-1} \tilde{\mathbf{R}}_{k,k-l} \tilde{\mathbf{\Psi}}_l \end{bmatrix}, \tag{113}$$

$$\mathbf{e}_p(m) := \begin{bmatrix} \sum_{k,l=0}^{K-1} \tilde{\mathbf{R}}_{l,l-k}^T \psi_p(k-m) \mathbf{\Lambda}_l + \sum_{k,l=0}^{K-1} \left(\tilde{\mathbf{S}}_{k,k-l} + \tilde{\mathbf{S}}_{l,l-k}^T \right) \psi_p(k-m) \tilde{\mathbf{\Psi}}_l \\ \\ \sum_{k,l=0}^{K-1} \tilde{\mathbf{R}}_{k,k-l} \psi_p(k-m) \tilde{\mathbf{\Psi}}_l \end{bmatrix}, \tag{114}$$

where the definition of the function $\psi_p(k)$ is given in (112). Then, we have

$$\begin{bmatrix} \bar{\mathbf{\Psi}}_m \\ \\ \bar{\mathbf{\Lambda}}_{m-1} \end{bmatrix} = \sum_{k,p} \begin{bmatrix} z_p^{-1}\mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \mathbf{H}_p \mathbf{e}_{k,p} \psi_p(k-m) = \sum_p \mathbf{A}_p \mathbf{b}_p^T \mathbf{e}_p(m). \tag{115}$$

Now, we consider (81). Note that for $m \neq 0, K$,

$$\begin{aligned} \frac{1}{2\pi j} \oint \mathbf{F}_\beta^{-1}(z) z^{-(m+1)} \mathrm{d}z &= \sum_p \begin{bmatrix} z_p^{-1}\mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \mathbf{H}_p \begin{bmatrix} \psi_p(-m)\mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & \psi_p(-m-1)\mathbf{I} \end{bmatrix} \\ &= -\sum_{p<0} \begin{bmatrix} z_p^{-1}\mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \mathbf{H}_p z_p^{-m} \begin{bmatrix} \mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & z_p^{-1}\mathbf{I} \end{bmatrix} \end{aligned} \tag{116}$$

and similarly

$$\begin{aligned} \frac{1}{2\pi j} \oint \mathbf{F}_\beta^{-1}(z) z^{K-(m+1)} \mathrm{d}z &= \sum_p \begin{bmatrix} z_p^{-1}\mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \mathbf{H}_p \begin{bmatrix} \psi_p(K-m)\mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & \psi_p(K-m-1)\mathbf{I} \end{bmatrix} \\ &= \sum_{p>0} \begin{bmatrix} z_p^{-1}\mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \mathbf{H}_p z_p^{K-m-1} \begin{bmatrix} z_p\mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \end{aligned} \tag{117}$$

Hence, for $m \neq 0, K$, we have

$$\begin{bmatrix} \tilde{\mathbf{\Psi}}_m \\ \\ \mathbf{\Lambda}_{m-1} \end{bmatrix} = \sum_p \begin{bmatrix} z_p^{-1}\mathbf{I} & \mathbf{O} \\ \\ \mathbf{O} & \mathbf{I} \end{bmatrix} \mathbf{H}_p \mathbf{e}'_p(m) + O\left(\frac{\|\mathbf{\Lambda}_{-1}\|}{C} \right) \tag{118}$$

where

$$\mathbf{e}'_p(m) = \mathbf{e}_p(m) - \boldsymbol{\delta}_p(m), \qquad \boldsymbol{\delta}_p(m) := \begin{cases} -z_p^{-m} \begin{bmatrix} \mathbf{\Lambda}_{-1} \\ \\ \mathbf{O} \end{bmatrix} & p < 0 \\ \\ z_p^{K-m-1} \begin{bmatrix} \mathbf{O} \\ \\ \tilde{\mathbf{\Psi}}_K \end{bmatrix} & p > 0 \end{cases} \tag{119}$$

The idea with the last term $O\left(\frac{\|\mathbf{\Lambda}_{-1}\|}{C}\right)$ is to choose C large enough to make the contribution of this term negligible. Note that $O(.)$ means a bound up to a *universal constant*. Now, we observe that by defining

$$\mathbf{R}_p(m,l) = \left[\begin{array}{cc} \sum\limits_{k=0}^{K-1} \left(\tilde{\mathbf{S}}_{k,k-l} + \tilde{\mathbf{S}}_{l,l-k}^T\right)\psi_p(k-m) & \sum\limits_{k=0}^{K-1} \tilde{\mathbf{R}}_{l,l-k}^T \psi_p(k-m) \\ \sum\limits_{k=0}^{K-1} \tilde{\mathbf{R}}_{k,k-l}\psi_p(k-m) & \mathbf{O} \end{array}\right], \tag{120}$$

we have

$$\mathbf{e}_p(m) = \sum_{l=0}^{K-1} \mathbf{R}_p(m,l) \left[\begin{array}{c} \tilde{\mathbf{\Psi}}_l \\ \mathbf{\Lambda}_l \end{array}\right] = \sum_{l,q} \mathbf{R}_p(m,l)\mathbf{A}_q\mathbf{b}_q^T\mathbf{e}'_q(l) + O\left(\frac{R\|\mathbf{\Lambda}_{-1}\|}{C}\right), \tag{121}$$

where R is any norm of $\mathbf{R}_p(m,l)$ (as an operator). We define the *distortion-dispersion* coefficients as

$$\tau_{p,q}(m,l) := \mathbf{b}_p^T\mathbf{R}_p(m,l)\mathbf{A}_q \tag{122}$$

and conclude that

$$\epsilon_p(m) := \mathbf{b}_p^T\mathbf{e}_p(m) = \sum_{q,l} \tau_{p,q}(m,l)\left(\epsilon_q(l) + \delta_q(l)\right) + O\left(\frac{R\|\mathbf{\Lambda}_{-1}\|}{C}\right), \tag{123}$$

where

$$\delta_p(m) := \mathbf{b}_p^T\boldsymbol{\delta}_p(m) \tag{124}$$

Now, let us check the boundary conditions. Note that from the definition of $\mathbf{\Phi}_\beta$ in (85), we have

$$\mathbf{\Phi}_\beta = \sum_p \left[\begin{array}{cc} z_p^{-1}\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} \end{array}\right] \mathbf{H}_p \left[\begin{array}{cc} \psi_p(0)\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \psi_p(-1)\mathbf{I} \end{array}\right] = \sum_p \mathbf{A}_p\mathbf{b}_p^T \left[\begin{array}{cc} \psi_p(0)\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \psi_p(-1)\mathbf{I} \end{array}\right] \tag{125}$$

Hence, from (88) and (89),

$$\begin{aligned} \mathbf{\Phi}_\beta \left[\begin{array}{c} \mathbf{I} \\ \mathbf{O} \end{array}\right] \mathbf{\Lambda}_{-1} &= \sum_{p>0} \mathbf{A}_p\mathbf{b}_p^T \left[\begin{array}{c} \mathbf{I} \\ \mathbf{O} \end{array}\right] \mathbf{\Lambda}_{-1} = \left[\begin{array}{c} \bar{\mathbf{\Psi}}_0 \\ \bar{\mathbf{\Lambda}}_{-1} \end{array}\right] + O\left(\frac{\|\mathbf{\Lambda}_{-1}\|}{C}\right) & (126) \\ &= \sum_p \mathbf{A}_p\mathbf{b}_p^T\mathbf{e}_p(0) + O\left(\frac{\|\mathbf{\Lambda}_{-1}\|}{C}\right) = \sum_p \mathbf{A}_p\epsilon_p(0) + O\left(\frac{\|\mathbf{\Lambda}_{-1}\|}{C}\right), & (127) \end{aligned}$$

and

$$\mathbf{\Phi}_\beta \left[\begin{array}{c} \mathbf{O} \\ \mathbf{I} \end{array}\right] \tilde{\mathbf{\Psi}}_K = \sum_{p<0} \mathbf{A}_p\mathbf{b}_p^T \left[\begin{array}{c} \mathbf{O} \\ z_p^{-1}\mathbf{I} \end{array}\right] \tilde{\mathbf{\Psi}}_K = \left[\begin{array}{c} \bar{\mathbf{\Psi}}_K \\ \bar{\mathbf{\Lambda}}_{K-1} \end{array}\right] = \sum_p \mathbf{A}_p\epsilon_p(K). \tag{128}$$

We conclude that

$$\mathbf{b}_p^T \left[\begin{array}{c} \mathbf{I} \\ \mathbf{O} \end{array}\right] \mathbf{\Lambda}_{-1} = \epsilon_p(0), \quad p > 0 \tag{129}$$

and

$$\mathbf{b}_p^T \left[\begin{array}{c} \mathbf{O} \\ z_p^{-1}\mathbf{I} \end{array}\right] \tilde{\mathbf{\Psi}}_K = \epsilon_p(K). \quad p < 0 \tag{130}$$

Hence,

$$\delta_p(m) = \begin{cases} -z_p^{-m}\epsilon_p(0) & p > 0 \\ z_p^{K-m}\epsilon_p(K) & p < 0 \end{cases} \tag{131}$$

E.2.3 Final step

Now, we define λ as the largest value of $|\epsilon_p(m)|$ and take the vector $\boldsymbol{\epsilon}(m) := (\epsilon_p(m))$. The notation $\|.\|$ refers to the infinity norm for vectors and the corresponding operator norm for matrices. We note that, by (131), we have $|\delta_p(m)| \leq \lambda$. Then, by (123), we conclude that

$$\lambda \leq 2 \max_{m,p} \sum_l \|\mathbf{R}_p(m,l)\| \|\mathbf{A}\| \lambda + O\left(\frac{R\lambda}{C}\right), \tag{132}$$

where we note that $\|\mathbf{\Lambda}_{-1}\| = O(\lambda)$.

Now, by considering the definition of $\tilde{\mathbf{R}}_{k,k-l}$ and $\tilde{\mathbf{S}}_{k,k-l}$, or specifically their non-zero elements, we define the p^{th} delay response of link (v,u) as

$$\tau_p^{vu}(m) = \sum_l \max\left\{ \left| \psi_p(l-m) - \frac{1}{|T_l^{vu}|} \sum_{k \in T_l^{vu}} \psi_p(k-m) \right| , \left| \psi_p(l-m) - \sum_{k \in S_l^{vu}} \frac{1}{|T_k^{vu}|} \psi_p(k-m) \right| \right\} \tag{133}$$

and note that

$$\sum_l \|\mathbf{R}_p(m,l)\| \leq \iota \tau_p^{vu}(m). \tag{134}$$

As a result, we have $R = O(\iota\tau)$, where

$$\tau = \max_{u,v,p,m} \tau_p^{vu}(m). \tag{135}$$

Hence, we conclude that

$$\lambda \leq \iota\tau\lambda\left(2\|\mathbf{A}\| + O\left(\frac{1}{C}\right)\right), \tag{136}$$

The proof is completed by noting that for small ι , $\|\mathbf{A}\|$ remains bounded (see next section). As such, we can choose $\iota = O(1/\tau)$ to guarantee that $\lambda = 0$ and hence the claims hold true. As a final note, we require all poles to be away from unit circle for all $\beta > 0$. In the next section, we show that this is implied by Assumption 2.

E.3 Explicit LQ-PEP

In the previous section, we computed the transfer matrix $\mathbf{F}_\beta(z)$ in an abstract form. Our theorems are based on the poles of this transfer matrix and its intrinsic basis. In this section, we explicitly calculate these quantities. For this, we solve the equation

$$\mathbf{H}_\beta(z)\mathbf{a} = \mathbf{O} \tag{137}$$

for a pole z and a basis $\mathbf{a}$. First, we note that the state $\mathbf{\Psi}$ consists of five block, which we divide into two part as $\tilde{\mathbf{\Psi}} = [\boldsymbol{\xi}^T \ \boldsymbol{\theta}^T]^T$ where $\boldsymbol{\xi}$ includes the two top blocks and $\boldsymbol{\theta}$ the remaining three. We also denote $\boldsymbol{\xi} = [\boldsymbol{\xi}_1^T \ \boldsymbol{\xi}_2^T]^T$ to refer to the two blocks in $\boldsymbol{\xi}$. Similarly, we write the dual vector $\mathbf{\Lambda} = [\boldsymbol{\lambda}_\xi^T \ \boldsymbol{\lambda}_\theta^T]^T$ and also take $\boldsymbol{\lambda}_\xi = [\boldsymbol{\lambda}_{\xi,1}^T \ \boldsymbol{\lambda}_{\xi,2}^T]$. Finally, we take $\bar{\mathbf{a}} = \left[\bar{\mathbf{a}}_1^T := \frac{z^2 \boldsymbol{\lambda}_{\xi,1}^T}{\beta} \ \boldsymbol{\theta}^T \ \boldsymbol{\lambda}_\theta^T\right]^T$. By the explicit computation of $\mathbf{F}_\beta(z)$, we see that

$$\begin{gathered}\bar{\mathbf{H}}_\beta(z)\bar{\mathbf{a}} = \mathbf{O}, \quad \boldsymbol{\xi} = \begin{bmatrix} \begin{bmatrix} z^{-1}\mathbf{I} \\ \mathbf{I} \end{bmatrix} & \mathbf{O} & \mathbf{O} \end{bmatrix} \bar{\mathbf{a}}, \\ \boldsymbol{\lambda}_{\xi,2} = z^{-1} \begin{bmatrix} (z^{-1}(1-L\mu)\mathbf{I} - \mathbf{W}^T) - L\mu\mathbf{I} & \begin{bmatrix} \frac{\rho}{\mu}(\mathbf{I}-\mathbf{W}^T) & \mathbf{O} & \mathbf{O} \end{bmatrix} & \mathbf{O} \end{bmatrix} \bar{\mathbf{a}} \end{gathered} \tag{138}$$

with the following block matrix structure:

$$\bar{\mathbf{H}}_\beta(z) = \begin{bmatrix} \mathbf{S}_\beta(z) & -\mathbf{a}_\pi^T(z) & -\mathbf{a}_\delta^T(z) \\ -\mathbf{a}_\pi(z^{-1}) & \mathbf{O} & \mathbf{B}^T(z) \\ -\mathbf{a}_\delta(z^{-1}) & \mathbf{B}(z^{-1}) & \mathbf{O} \end{bmatrix} \tag{139}$$

where

$$\mathbf{S}_\beta(z) = (1-L\mu)(2-z-z^{-1})\mathbf{I} + z\mathbf{W} + z^{-1}\mathbf{W}^T + \beta\mathbf{I} - \zeta\mathbf{P}_{\mathbf{1}}^{\perp}, \tag{140}$$

$$\mathbf{a}_\pi^T(z) = \begin{bmatrix} \mu z\mathbf{I} & \mathbf{O} & \mathbf{O} \end{bmatrix}, \tag{141}$$

$$\mathbf{a}_\delta^T(z) = \begin{bmatrix} -\frac{\rho}{\mu}\left((z-1)\mathbf{I} + \mathbf{W}^T\right) & \mathbf{O} & \mathbf{O} \end{bmatrix}, \tag{142}$$

and

$$\mathbf{B}(z) = \begin{bmatrix} (z-1+\alpha)\mathbf{I} & -\alpha\mathbf{I} & \mathbf{O} \\ \mathbf{O} & (z-\gamma)\mathbf{I} & \mathbf{Q} \\ \eta(1-\gamma)\mathbf{I} & \mathbf{O} & (z-1)\mathbf{I} + \eta\mathbf{Q} \end{bmatrix}. \tag{143}$$

We observe that the poles and the intrinsic basis are calculated by solving $\bar{\mathbf{H}}_\beta(z)\bar{\mathbf{a}} = \mathbf{O}$. We can directly verify this by defining

$$\mathbf{\Omega}_\beta(z) = \mathbf{S}_\beta(z) + \mathbf{G}\left(z^{-1}\right) + \mathbf{G}^T\left(z\right) \tag{144}$$

with $\mathbf{G}(z)$ defined as:

$$\mathbf{G}(z) = \frac{\rho}{z(z-1)}\left[(z-1)\mathbf{I} + \eta\mathbf{Q}\right]\left[(z-1+\alpha)\mathbf{I} + \eta\left(1 + \frac{\alpha}{z-\gamma}\right)\mathbf{Q}\right]^{-1}\left[(z-1)\mathbf{I} + \eta\mathbf{W}\right], \tag{145}$$

we have

$$\bar{\mathbf{H}}_\beta^{-1}(z) = \begin{bmatrix} \mathbf{I} \\ \mathbf{B}^{-1}(z^{-1})\mathbf{a}_\delta(z^{-1}) \\ \mathbf{B}^{-T}(z)\mathbf{a}_\pi(z^{-1}) \end{bmatrix} \mathbf{\Omega}^{-1}(z^{-1}) \begin{bmatrix} \mathbf{I} & \mathbf{a}_\delta^T(z)\mathbf{B}^{-T}(z) & \mathbf{a}_\pi^T(z)\mathbf{B}^{-1}(z^{-1}) \end{bmatrix}$$

$$+ \begin{bmatrix} \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{B}^{-1}(z^{-1}) \\ \mathbf{O} & \mathbf{B}^{-T}(z) & \mathbf{O} \end{bmatrix} \tag{146}$$

$$\mathbf{B}^{-1}(z) = \begin{bmatrix} (\mathbf{I} + \frac{\eta}{z-1}\mathbf{Q}) \\ -\frac{\eta(\gamma-1)}{(z-\gamma)(z-1)}\mathbf{Q} \\ \frac{\eta(\gamma-1)}{z-1}\mathbf{I} \end{bmatrix} \left[(z-1+\alpha)\mathbf{I} + \eta\left(1 + \frac{\alpha}{z-\gamma}\right)\mathbf{Q}\right]^{-1} \begin{bmatrix} \mathbf{I} & \frac{\alpha}{z-\gamma}\mathbf{I} & \frac{z-1+\alpha}{\eta(\gamma-1)}\mathbf{I} \end{bmatrix}$$

$$+ \begin{bmatrix} \mathbf{O} & \mathbf{O} & -\frac{1}{\eta(\gamma-1)}\mathbf{I} \\ \mathbf{O} & \frac{1}{z-\gamma}\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \end{bmatrix}. \tag{147}$$

Hence, the poles exist in two different cases:

case 1: $\mathbf{B}(z)$ or $\mathbf{B}(z^{-1})$ loses rank. In this case, $z = z_p$ is one of the first poles. We first discuss the poles inside the unit circle. According to Assumption 1, this corresponds to $\mathbf{B}(z)$ losing rank and $\mathbf{B}(z^{-1})$ being invertible. Note that such a first pole is a root of the polynomial:

$$z^2 + (\eta\lambda_i - 1 + \alpha - \gamma)z + [\gamma(1-\alpha) + \eta\lambda_i(\alpha-\gamma)] \tag{148}$$

where λ_i is an eigenvalue of $\mathbf{Q}$. We directly solve $\bar{\mathbf{a}}$, which gives $\bar{\mathbf{a}} = [\mathbf{O} \quad \bar{\mathbf{a}}_0^T \quad \mathbf{O}]^T$ with

$$\bar{\mathbf{a}}_0 = \begin{bmatrix} (1 + \frac{\eta}{z-1}\lambda_i)(z-\gamma)\mathbf{I} \\ -\frac{\eta(\gamma-1)}{z-1}\lambda_i\mathbf{I} \\ \frac{\eta(\gamma-1)(z-\gamma)}{z-1}\mathbf{I} \end{bmatrix} \mathbf{u}_i, \tag{149}$$

where $\mathbf{u}_i$ is the right eigenvector of $\mathbf{Q}$ corresponding to λ_i. Now, we consider the case that $\mathbf{B}(z^{-1})$ loses rank and $\mathbf{B}(z)$ is bijective. This corresponds to the inverse of the poles in the first case with a similar notion of λ_i. Then, we have $\bar{\mathbf{a}} = [\mathbf{O} \quad \mathbf{O} \quad \bar{\mathbf{a}}_1^T]^T$, where

$$\bar{\mathbf{a}}_1 = \begin{bmatrix} (z-\gamma)\mathbf{I} \\ \alpha\mathbf{I} \\ \frac{z-1+\alpha}{\eta(\gamma-1)}(z-\gamma)\mathbf{I} \end{bmatrix} \mathbf{v}_i, \tag{150}$$

where $\mathbf{v}_i$ is the left eigenvalue of $\mathbf{Q}$ corresponding to λ_i.

case 2: The matrices $\mathbf{B}(z)$ and $\mathbf{B}(z^{-1})$ are invertible. Then, we see that

$$\bar{\mathbf{a}} = \begin{bmatrix} \mathbf{I} \\ \mathbf{B}^{-1}(z^{-1})\mathbf{a}_\delta(z^{-1}) \\ \mathbf{B}^{-T}(z)\mathbf{a}_\pi(z^{-1}) \end{bmatrix} \bar{\mathbf{a}}_2, \tag{151}$$

and we have $\mathbf{\Omega}_\beta(z)\bar{\mathbf{a}}_2 = \mathbf{O}$. This results in the second poles.

As a final note, the first poles are independent of β. The condition that no pole is on the unit circle for all $\beta > 0$ means that $\mathbf{\Omega}_\beta(z)$ remains full rank for all β and for any z on the unit circle. This is equivalent to $\mathbf{\Omega} \succeq \zeta \mathbf{P}_\mathbf{1}^\perp$, where

$$\mathbf{\Omega}(z) = \mathbf{S}(z) + \mathbf{G}\left(z^{-1}\right) + \mathbf{G}^T\left(z\right), \quad \mathbf{S}(z) := (1-L\mu)(2-z-z^{-1})\mathbf{I} + z\mathbf{W} + z^{-1}\mathbf{W}^T \tag{152}$$

F Proof of Theorem 1

In this part, we present the proof of Theorem 1 by studying nearly isolated nodes with bounded delay and assuming agility (Assumption 2). To proceed, we consider a case where $\mathbf{W}, \mathbf{Q}$ and ξ are infinitesimal. Note that by agility, all terms in the delay factor are in the form of

$$|\psi_p(m-l) - \psi_p(m-k)| \tag{153}$$

if the message at time l (in a given node) is used at time k. Fixing m, we can bound the sum over (l,k) of these terms by dividing them into two groups: the first group consists of cases where m is between l and k, and the second group contains all other possibilities. The sum over the first group is bounded by $1 + |z| + |z|^2 + \ldots + |z|^{d-1} \leq d$ and the sum over the second group is bounded by $(1-|z|^d)(1+|z|+|z|^2+\ldots) \leq d$. Hence, $\tau \leq 2d$.

Now we calculate the poles. We first look at the case $\mathbf{W} = \mathbf{Q} = \mathbf{O}$ and $\xi = 0$. In this case, the poles of type 1 are $z = \gamma, 1-\alpha$ and their inverses. If $L\mu < 1$, the poles of the second type are well-separated from the unit circle for all β except the dominant pole which approaches $z = 1$ as $\beta \to 0$. Hence, we only consider the Taylor expansion of $\mathbf{\Omega}_\beta(z)$ up to the first order of $(z-1)$ and $\mathbf{W}, \mathbf{Q}$, which provides us with the two cases in Theorem 1. If α is large, we have:

$$\mathbf{\Omega}_\beta(z) = \left[(z-1)\frac{\rho}{\alpha} + \beta\right]\mathbf{I} + \mathbf{W} + \mathbf{W}^T + \frac{\rho\eta}{\alpha}[\mathbf{Q} + \mathbf{W} + \mathbf{Q}^T + \mathbf{W}^T] - \zeta\mathbf{P}_\mathbf{1}^\perp + \text{h.o.t} \tag{154}$$

We observe that if

$$\frac{\eta\rho}{\alpha}\left(\mathbf{Q} + \mathbf{Q}^T\right) + \left(1 + \frac{\eta\rho}{\alpha}\right)\left(\mathbf{W} + \mathbf{W}^T\right) \succeq \zeta\mathbf{P}_\mathbf{1}^\perp, \tag{155}$$

then Ω_β will not have any zero for purely imaginary $z-1$ (which is the unit circle in this vicinity), and hence, choosing $\xi = O(\iota)$ will lead to the dominant pole being separated from the unit circle with $O(\iota)$ for all $\beta > 0$. Finally, we observe that by choosing $\iota = O(\frac{1}{d})$, the conditions of Theorem 2 are satisfied, which proves the first case of Theorem 1.

If α is nearly zero, $\mathbf{G}(z)$ reduces to

$$\mathbf{G}(z) = \frac{\rho}{z(z-1)}[(z-1)\mathbf{I} + \eta\mathbf{W}] \tag{156}$$

Repeating the above argument for this function, we require

$$\eta\rho(\mathbf{W} + \mathbf{W}^T) \succeq \zeta\mathbf{P}_\mathbf{1}^\perp. \tag{157}$$

G Behaviour of the dominant pole

As our assumptions require the computation of the system's poles, in this section, we focus on a special communication network, which we refer to as the *divided network*, and demonstrate how the dominant pole changes under different scenarios. The dominant pole is the closest pole to the unit circle from inside the circle. Since this pole has the greatest impact on the delay factor and significantly influences the system's behaviour, we concentrate only on the behaviour of this pole in this section.

G.1 Divided network setup

This network consists of two clusters, each containing N nodes that are fully connected within the cluster. Among these N nodes in each cluster, there are n nodes that can communicate with all the nodes in the other cluster. These nodes are known as communicators. The ratio for this setup is defined as n/N. By increasing this ratio, the network approaches a fully connected structure, whereas decreasing it results in two separate fully connected sub-networks. An example of this setup with $N = 4$ and $n = 2$ is depicted in Figure 3a.

Adjacency matrix A, zero row-sum matrix W, and zero column-sum matrix Q: The adjacency and gossip matrices for this setup can be easily computed, as they have a block structure where each block is in the form of identity and all-ones matrices. Here is their exact formulation:

$$\mathbf{A} = \begin{bmatrix} \mathbf{1}_{N-n}\mathbf{1}_{N-n}^T & \mathbf{1}_{N-n}\mathbf{1}_{n}^T & \mathbf{1}_{N-n}\mathbf{1}_{n}^T & \mathbf{O} \\ \mathbf{1}_{n}\mathbf{1}_{N-n}^T & \mathbf{1}_{n}\mathbf{1}_{n}^T & \mathbf{1}_{n}\mathbf{1}_{n}^T & \mathbf{O} \\ \mathbf{O} & \mathbf{1}_{n}\mathbf{1}_{n}^T & \mathbf{1}_{n}\mathbf{1}_{n}^T & \mathbf{1}_{n}\mathbf{1}_{N-n}^T \\ \mathbf{O} & \mathbf{1}_{N-n}\mathbf{1}_{N-n}^T & \mathbf{1}_{N-n}\mathbf{1}_{n}^T & \mathbf{1}_{N-n}\mathbf{1}_{N-n}^T \end{bmatrix} \tag{158}$$

$$\mathbf{D}_{\text{out}} = \begin{bmatrix} N\mathbf{I}_{(N-n)\times(N-n)} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & 2N\mathbf{I}_{n\times n} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & 2N\mathbf{I}_{n\times n} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & N\mathbf{I}_{(N-n)\times(N-n)} \end{bmatrix} \tag{159}$$

$$\mathbf{W} = (N+n)\mathbf{I}_{2N\times 2N} - \mathbf{A} \qquad \mathbf{Q} = \mathbf{D}_{\text{out}} - \mathbf{A} \tag{160}$$

In Figure 3, the factor named *Power of the weights* is used to scale $\mathbf{W}$ and $\mathbf{Q}$. This factor is equivalent to the isolation factor ι, presented in the main paper.

Simplification of the calculations The structure of this graph and its associated gossip matrices offers an opportunity to reduce the computational complexity of calculating the roots of $\det \mathbf{\Omega}(z)$. This reduction simplifies the computation from determining the determinant of an $M \times M$ matrix to working with 4×4 matrices, whose elements depend only on the network's ratio. The code is provided in Section H.2.

G.2 Numerical results

Figure 3 demonstrates the behaviour of the dominant pole in this setup. In Figure 3b, we observe that the magnitude of the dominant pole decreases as the ratio increases. This is rationale since adding more links allows some nodes to compensate for delayed messages by passing the required information for convergence in the network. The edgy behavior observed is due to the fixed parameters used during the simulation and the presence of multiple poles in the system. At these transition points, a different pole may become dominant.

Figure 3 also shows how the dominant pole behaves when the gossip matrices are scaled by the factor 'Power of the weights.' The results support our theory, as these gossip matrices do not satisfy assumption 6. By either decreasing or increasing this power, the system fails, causing the dominant pole to move to the unit circle and preventing the algorithm from converging theoretically.

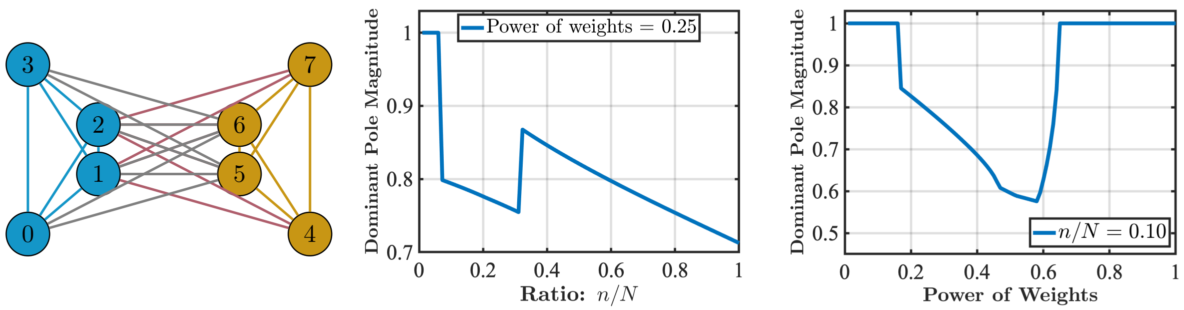


Figure 3: *(a) Divided network setup. (b) Dominant pole behavior with respect to the ratio in divided networks. (c) Dominant pole behavior with respect to the isolation factor, in a divided network with a ratio of* 0.1.

H Extended experiments

H.1 Merits of asynchronous algorithms over synchronous algorithms

In the first experiment of the paper, we considered an asynchronous setup with varying communication delays on network edges and different processing speeds across nodes, measuring the wall-clock time to convergence. In both DAGP and ASY-DAGP, the delays and computation speeds follow the same distribution. However, in DAGP, all nodes must wait for the slowest node in each iteration before proceeding (synchronization), which significantly slows down the overall convergence time.

The selected setup in the first experiment represents an extremely challenging scenario, as some nodes can be up to 50 times faster than others. In this section, we have conducted experiments with varying levels of asynchrony, ranging from one extreme (an ideal setup with fixed computation time per iteration per node and no communication delay) to the other extremes (imperfect setups with computational or communication heterogeneity, or both). These experiments demonstrate how asynchronous algorithms outperform synchronous ones in convergence time as the scenario becomes more challenging. The results are presented in Figure H.1. They show that as computation or communication heterogeneity increases, both ASY-DAGP and DAGP require more time to converge, but ASY-DAGP performs several times faster than DAGP. Please note that the time scale in each plot is different from the others.

Computational Heterogeneity: This refers to the variation in processing speeds across different nodes in the network. We assume each node's computation time follows a uniform distribution within the interval $[1, d + h \times v]$, where d and h increase computational heterogeneity, and v is the node number. Specifically, d increases network heterogeneity, while h increases device heterogeneity. We considered four different levels of computational heterogeneity:

- $d = 1$, $h = 0$: All nodes perform at the same speed, representing an ideal computation setup.
- $d = 10$, $h = 0$: All nodes have the same computational power, but computation speeds vary at each iteration.
- $d = 10$, $h = 2$: Nodes have different computational powers, and computation speeds vary at each iteration.
- $d = 10$, $h = 5$: Similar to the above setup but with greater heterogeneity among devices.

Communication Heterogeneity: This refers to the variation in communication times between different nodes in the network. We assumed that communication times follow a uniform distribution $[0, t_{comm}]$, and we tested five different setups with $t_{comm} = 0, 1, 5, 10, 30$.

Please Note that the setups could be further challenged by introducing more imperfections, such as nodes with minimum computation times other than 1, or delays with minimum other than 0.

H.2 Code repository

The code for all experiments is provided in the following anonymous link: https://github.com/Firooz-shahriari/Asynchronous-Decentralized-Constrained-Optimization

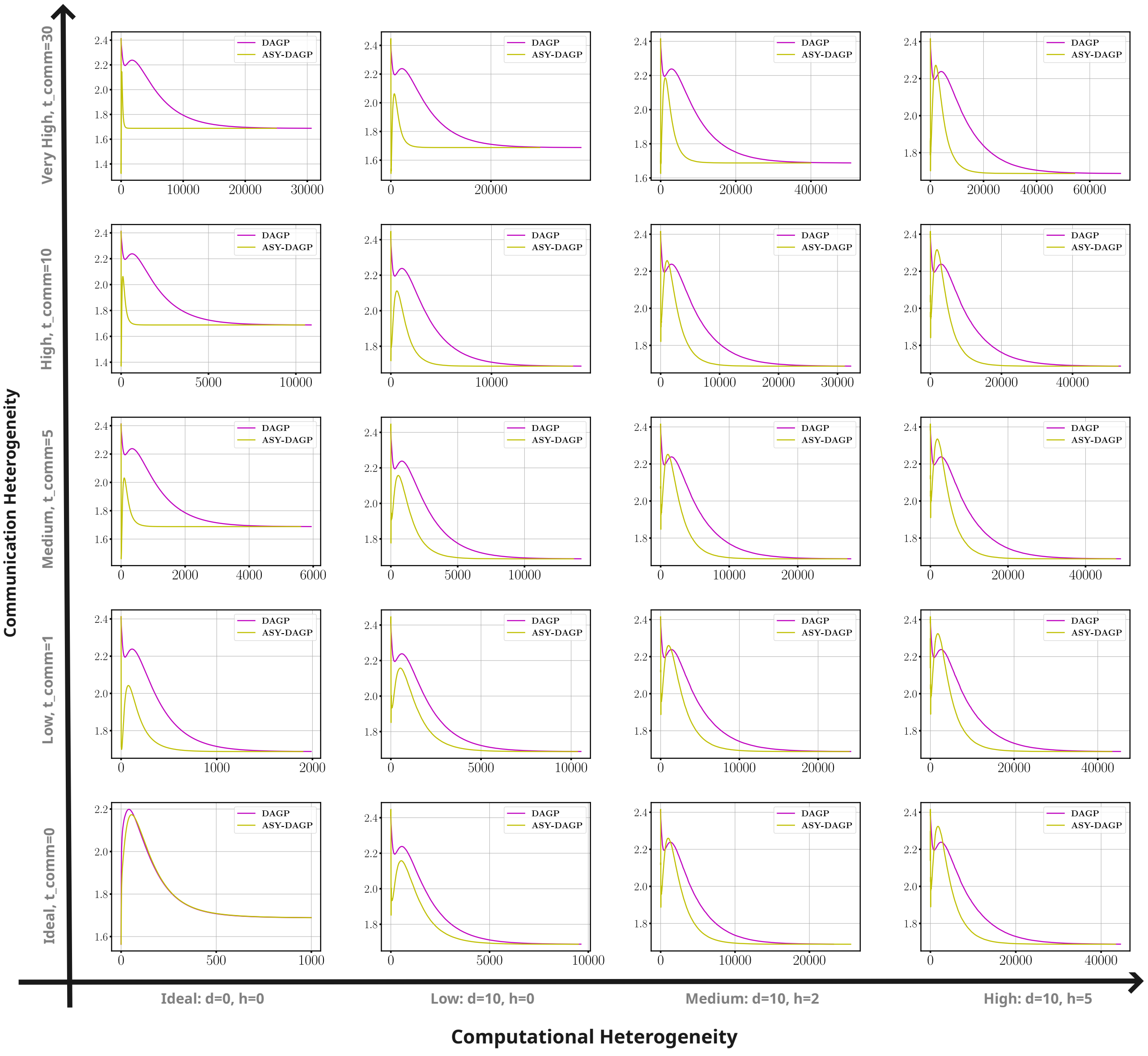


Figure 4: Each plot demonstrates the value of the objective function versus time. The agents' computation times follow a uniform distribution $[1, d + h \times v]$, and the communication times follow a uniform distribution $[0, t_{\mathrm{comm}}]$.

I Matrix definitions

In the following definitions, $\odot$ represents the Hadamard product. The matrices $\mathbf{C}_{k,k-l}$ are constructed by defining their elements as $c^{vu}_{k,k-l}$, as given in (33), which depend on the asynchrony and delay profiles. We assume deterministic asynchrony and delays, making these matrices known.

$$\bar{\mathbf{R}} = \begin{bmatrix} \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{I} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ -\frac{\rho}{\mu}\mathbf{I} & \frac{\rho}{\mu}(\mathbf{I}-\mathbf{W}) & (1-\alpha)\mathbf{I} & \alpha\mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \gamma\mathbf{I} & -\mathbf{Q} \\ \mathbf{O} & \mathbf{O} & \eta(\gamma-1)\mathbf{I} & \mathbf{O} & \mathbf{I}-\eta\mathbf{Q} \end{bmatrix} \tag{161}$$

$$\tilde{\mathbf{R}}_{k,k-l} = \begin{bmatrix} \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & -\frac{\rho}{\mu}\mathbf{W}\odot\mathbf{C}_{k,k-l} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & -\mathbf{Q}\odot\mathbf{C}_{k,k-l} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & -\eta\mathbf{Q}\odot\mathbf{C}_{k,k-l} \end{bmatrix} \tag{162}$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{I}_{M\times M} \\ \mathbf{O}_{4M\times M} \end{bmatrix} \tag{163}$$

$$\bar{\mathbf{S}} = \begin{bmatrix} (2-L\mu)\,\mathbf{I}-\zeta\mathbf{P}_{\mathbf{1}}^{\perp} & (L\mu-1)\mathbf{I}+\mathbf{W} & -\mu\mathbf{I} & \mathbf{O} & \mathbf{O} \\ (L\mu-1)\mathbf{I}+\mathbf{W}^T & -L\mu\mathbf{I} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ -\mu\mathbf{I} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \end{bmatrix} \tag{164}$$

$$\tilde{\mathbf{S}}_{k,k-l} = \begin{bmatrix} \mathbf{O} & \mathbf{W}\odot\mathbf{C}_{k,k-l} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{W}^T\odot\mathbf{C}^T_{k,k-l} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{O} \end{bmatrix} \tag{165}$$