Totally Asynchronous Large-Scale Quadratic Programming: Regularization, Convergence Rates, and Parameter Selection

Matthew Ubl* and Matthew T. Hale*

Abstract

Quadratic programs arise in robotics, communications, smart grids, and many other applications. As these problems grow in size, finding solutions becomes more computationally demanding, and new algorithms are needed to efficiently solve them at massive scales. Targeting large-scale problems, we develop a multi-agent quadratic programming framework in which each agent updates only a small number of the total decision variables in a problem. Agents communicate their updated values to each other, though we do not impose any restrictions on the timing with which they do so, nor on the delays in these transmissions. Furthermore, we allow weak parametric coupling among agents, in the sense that they are free to independently choose their stepsizes, subject to mild restrictions. We further provide the means for agents to independently regularize the problems they solve, thereby improving convergence properties while preserving agents' independence in selecting parameters and ensuring a global bound on regularization error is satisfied. Larger regularizations accelerate convergence but increase error in the solution obtained, and we quantify the tradeoff between convergence rates and quality of solutions. Simulation results are presented to illustrate these developments.

I. Introduction

Convex optimization problems arise in a diverse array of engineering applications, including signal processing [1], robotics [2], [3], communications [4], machine learning [5], and many others [6]. In all of these areas, problems can become very large as the number of network members (robots, processors, etc.) becomes large. Accordingly, there has arisen interest in solving large-scale optimization problems. A common feature of large-scale solvers is that they are parallelized or distributed among a collection of agents in some way. As the number of agents grows, it can be difficult or impossible to ensure synchrony among distributed computations and communications, and there has therefore arisen interest in distributed asynchronous optimization algorithms.

One line of research considers asynchronous optimization algorithms in which agents' communication topologies vary in time. A representative sample of this work includes [7]–[12], and these algorithms all rely on an underlying averaging-based update law, i.e., different agents update the same decision variables and then repeatedly average their iterates to mitigate disagreements that stem from asynchrony. These approaches (and others in the literature)

require some form of graph connectivity over intervals of a finite length. In this paper, we are interested in cases in which delay bounds are outside agents' control, e.g., due to environmental hazards and adversarial jamming for a team of mobile autonomous agents. In these settings, verifying graph connectivity can be difficult for single agents to do, and it may not be possible to even check that connectivity assumptions are satisfied over prescribed intervals. Furthermore, even if such checking is possible, it will be difficult to reliably attain connectivity over the required intervals with unreliable and impaired communications. For multi-agent systems with impaired communications, we are interested in developing an algorithmic framework that succeeds without requiring delay bounds.

In this paper, we develop a totally asynchronous quadratic programming (QP) framework for multi-agent optimization. Our interest in quadratic programming is motivated by problems in robotics [3] and data science [13], where some standard problems can be formalized as QPs. The "totally asynchronous" label originates in [14], and it describes a class of algorithms which tolerate arbitrarily long delays, which our framework will do. In addition, our developments will use block-based update laws in which each agent updates only a small subset of the decision variables in a problem, which reduces each agent's computational burden and, as we will show, reduces its onboard storage requirements as well.

Other work on distributed quadratic programming includes [15]–[20]. Our work differs from these existing results because we consider non-separable objective functions, and because we consider unstructured update laws (i.e., we do not require communications and computations to occur in a particular sequence or pattern). Furthermore, we consider only deterministic problems, and our framework converges exactly to a problem's solution, while some existing works consider stochastic problems and converge approximately or in an appropriate statistical sense. This work is also somewhat related to distributed solutions to systems of linear equations, e.g. [21], because the gradient of a quadratic function is a linear function. However, methods for such problems are not readily applicable in this paper due to set constraints.

Asynchrony in agents' communications and computations implies that they will send and receive different information at different times. As a result, they will disagree about the values of decision variables in a problem. Just as it is difficult for agents to agree on this information, it can also be difficult to agree on a stepsize value in their algorithms. One could envision a network of agents solving an agreement problem, e.g., [22], to compute a common stepsize, though we instead allow agents to independently choose stepsizes, subject to mild restrictions, thereby eliminating the need to reach agreement before optimizing.

It has been shown that regularizing problems can endow them with an inherent robustness to asynchrony and improved convergence properties, e.g., [23]–[25]. Although regularizing is not required here, we show, in a precise sense, that regularizing improves convergence rates of our framework as well. It is common for regularization-based approaches to require agents to use the same regularization parameter, though this is undesirable for the same reasons as using a common stepsize. Therefore, we allow agents to independently choose regularization parameters as well.

To the best of our knowledge, few works have considered both independent stepsizes and regularizations. The most relevant is [23], which considers primal-dual algorithms for problems with functional constraints and synchronous

primal updates. This paper is different in that we consider set-constrained problems with totally asynchronous updates, in addition to unconstrained problems. Regularizing introduces errors in a solution, and we bound these errors in terms of agents' regularization parameters.

A preliminary version of this work appeared in [26], however this version further includes distributed regularization selection rules for convergence rate and error bound satisfaction, along with new error bounds and and simulation results.

The rest of the paper is organized as follows. Section II provides background on QPs and formal problem statements. Then, Section III proposes an update law to solve the problems of interest, and Section IV proves its convergence. Next, Section V derives a convergence rate, and Section VI then quantifies the effect of regularizations on the convergence rate. Section VII provides an absolute error bound in terms of agents' regularizations for a set-constrained problem, while Section VIII provides a relative error bound for the unconstrained case. Section IX next illustrates these results in simulation. Finally, Section X concludes the paper.

II. BACKGROUND AND PROBLEM STATEMENT

In this section, we describe the quadratic optimization problems to be solved, as well as the assumptions imposed upon these problems and the agents that solve them. We then describe agents' stepsizes and regularizations and introduce the need to allow agents to choose these parameters independently. We next describe the benefits of independent regularizations, and give two formal problem statements that will be the focus of the remainder of the paper.

A. Quadratic Programming Background

We consider a quadratic optimization problem distributed across a network of N agents, where agents are indexed over $i \in [N] := \{1, ..., N\}$. Agent i has a decision variable $x^{[i]} \in \mathbb{R}^{n_i}, n_i \in \mathbb{N}$, which we refer to as its state, and we allow for $n_i \neq n_j$ if $i \neq j$. The state $x^{[i]}$ is subject to the set constraint $x^{[i]} \in X_i \subset \mathbb{R}^{n_i}$, and we make the following assumption about each X_i.

Assumption 1: For all $i \in [N]$, the set $X_i \subset \mathbb{R}^{n_i}$ is non-empty, compact, and convex. $\triangle$

We define the network-level constraint set $X := X_1 \times \cdots \times X_N$, and Assumption 1 implies that X is non-empty, compact, and convex. We further define the global state as $x := \left(x^{[1]^T}, ..., x^{[N]^T} \right)^T \in X \subset \mathbb{R}^n$, where $n = \sum_{i \in [N]} n_i$. We consider quadratic objectives

$$f(x) := \frac{1}{2} x^T Q x + r^T x,$$

where $Q \in \mathbb{R}^{n \times n}$ and $r \in \mathbb{R}^n$. We then make the following assumption about f.

Assumption 2: In f, Q is symmetric. $\triangle$

Note that Assumption 2 holds without loss of generality because a non-symmetric Q will have only its symmetric part contribute to the value of the quadratic form that defines f. Because f is quadratic, it is twice continuously differentiable, which we indicate by writing that f is C^2. In addition, $\nabla f = Qx + r$, and ∇f is therefore Lipschitz

with constant $\|Q\|_2$. It is common to assume outright that Q is positive definite, though here we are able to dispense with this assumption based on one in terms of the block structure of agents' updates.

In this paper, we divide $n \times n$ matrices into blocks. Given a matrix $B \in \mathbb{R}^{n \times n}$, where $n = \sum_{i=1}^{N} n_i$, the ith block of B, denoted $B^{[i]}$, is the $n_i \times n$ matrix formed by rows of B with indices $\sum_{k=1}^{i-1} n_k + 1$ through $\sum_{k=1}^{i} n_k$. In other words, $B^{[1]}$ is the first n_1 rows of B, $B^{[2]}$ is the next n_2 rows, etc. Similarly, for a vector b, $b^{[1]}$ is the first n_1 entries of b, $b^{[2]}$ is the next n_2 entries, etc. We further define the notation of a sub-block $B_j^{[i]}$, where $B^{[i]} = \begin{bmatrix} B_1^{[i]} & B_2^{[i]} & ... & B_N^{[i]} \end{bmatrix}$. That is, $B_1^{[i]}$ is the first n_1 columns of $B^{[i]}$, $B_2^{[i]}$ is the next n_2 columns, etc. For notational simplicity, $B = \left[B_j^{[i]} \right]_p$ means the matrix B has been partitioned into blocks according to the partition vector $p := [n_1, n_2, \ldots, n_N]^T$. That is,

$$B = \left[B_j^{[i]} \right]_p = \begin{bmatrix} B^{[1]} \\ B^{[2]} \\ \vdots \\ B^{[N]} \end{bmatrix} = \begin{bmatrix} B_1^{[1]} & B_2^{[1]} & \ldots & B_N^{[1]} \\ B_1^{[2]} & B_2^{[2]} & \ldots & B_N^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ B_1^{[N]} & B_2^{[N]} & \ldots & B_N^{[N]} \end{bmatrix},$$

where $B_j^{[i]} \in \mathbb{R}^{n_i \times n_j}$ for all $i, j \in [N]$

Previous work has shown that totally asynchronous algorithms may diverge if Q is not diagonally dominant [14, Example 3.1]. While enforcing this condition is sufficient to ensure a totally asynchronous update scheme will converge, in this paper we will instead require the weaker condition of block diagonal dominance.

Definition 1: Let the matrix $B = \left[B_j^{[i]} \right]_p$, where $p = [n_1, n_2, \ldots, n_N]^T$ is given by the dimensions of agents' states above. If the diagonal sub-blocks $B_i^{[i]}$ are nonsingular and if

$$\left(\left\| B_i^{[i]^{-1}} \right\|_2 \right)^{-1} \geq \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| B_j^{[i]} \right\|_2 \quad \text{for all } i \in [N], \tag{1}$$

then B is said to be *block diagonally dominant* relative to the choice of partitioning p. If strict inequality in Equation (1) is valid for all $i \in [N]$, then B is *strictly block diagonally dominant* relative to the choice of partitioning p. ▲

In later analysis, we will use the gap between the left and right hand side of Equation (1), which we define as

$$\delta_i(B) = \left(\left\| B_i^{[i]^{-1}} \right\|_2 \right)^{-1} - \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| B_j^{[i]} \right\|_2.$$

Note that if $p = [1, 1, \ldots, 1]^T$, Definition 1 reduces to diagonal dominance in the usual sense. We now make the following assumption:

Assumption 3: In f, $Q = \left[Q_j^{[i]} \right]_p$ is strictly block diagonally dominant, where $p = [n_1, n_2, \ldots, n_N]^T$, and n_i is the length of $x^{[i]}$ for all $i \in [N]$. △

Note also that from Theorem 2 in [27], if Assumptions 2 and 3 hold for a matrix B, then B is also positive definite. Therefore Assumptions 2 and 3 imply that $Q \succ 0$, which renders f strongly convex.

B. Problem Statements

Following our goal of reducing parametric coupling between agents, we wish to allow agents to select stepsizes independently. In particular, we wish for the stepsize for block i, denoted γ_i, to be chosen using only knowledge of $Q^{[i]}$. The selection of γ_i should not depend on any other block $Q^{[j]}, j \neq i$, or any stepsize choice, γ_j, for any other block. Allowing independent stepsizes will preclude the need for agents to agree on a single value before optimizing. The following problem will be one focus of the remainder of the paper.

Problem 1: Design a totally asynchronous distributed optimization algorithm to solve

$$\underset{x \in X}{\text{minimize}} \quad \frac{1}{2} x^T Q x + r^T x,$$

where only agent i updates $x^{[i]}$, and where agents choose stepsizes independently. $\diamond$

While an algorithm that satisfies the conditions stated in Problem 1 is sufficient to find a solution, we wish to allow for regularizations as well. Regularizations are commonly used for centralized quadratic programs to improve convergence properties, and we will therefore use them here. However, in keeping with the independence of agents' parameters, we wish to allow agents to choose independent regularization parameters. As with stepsizes, we wish for the regularization for block i, denoted α_i, to be chosen using only knowledge of $Q^{[i]}$. The regularized form of f, denoted f_A, is

$$f_A(x) := f(x) + \frac{1}{2} x^T A x = \frac{1}{2} x^T (Q + A) x + r^T x, \tag{2}$$

where $A = \text{diag}\,(\alpha_1 I_{n_1}, ..., \alpha_N I_{n_N})$, and where I_{n_i} is the $n_i \times n_i$ identity matrix. Note that $\frac{\partial f_A}{\partial x^{[i]}} = Q^{[i]} x + r^{[i]} + \alpha_i x^{[i]}$, where we see that only α_i affects the gradient of f with respect to $x^{[i]}$. With the goal of independent regularizations in mind, we now state the second problem that we will solve.

Problem 2: Design a totally asynchronous distributed optimization algorithm to solve

$$\underset{x \in X}{\text{minimize}} \quad \frac{1}{2} x^T (Q + A) x + r^T x,$$

where only agent i updates $x^{[i]}$, and where agents independently choose their stepsizes and regularizations. $\triangle$

Section III specifies the structure of the asynchronous communications and computations used to solve Problem 1, and we will solve Problem 1 in Section IV. Afterwards, we will solve Problem 2 in Section V.

III. BLOCK-BASED MULTI-AGENT UPDATE LAW

To define the update law for each agent's state, we first describe the information stored onboard each agent and how agents communicate with each other. Each agent will store a vector containing its own state and that of every agent it communicates with. Formally, we will denote agent i's full vector of states by x_i, and this is agent i's local copy of the global state. Agent i's own states in this vector are denoted by $x_i^{[i]}$. The current values stored onboard agent i for agent j's states are denoted by $x_i^{[j]}$. In the forthcoming update law, agent i will only compute updates for $x_i^{[i]}$, and it will share only $x_i^{[i]}$ with other agents when communicating. Agent i will only change the value of $x_i^{[j]}$ when agent j sends its own state to agent i.

At time k, agent i's full state vector is denoted $x_i(k)$, with its own states denoted $x_i^{[i]}(k)$ and those of agent j denoted $x_i^{[j]}(k)$. At any timestep, agent i may or may not update its states due to asynchrony in agents' computations. As a result, we will in general have $x_i(k) \neq x_j(k)$ at all times k. We define the set K^i to contain all times k at which agent i updates $x_i^{[i]}$. In designing an update law, we must provide robustness to asynchrony while allowing computations to be performed in a distributed fashion. First-order gradient descent methods are robust to many disturbances, with the additional benefit of being computationally simple. Using our notation of a matrix block, we define $\nabla^{[i]} f := \frac{\partial f}{\partial x^{[i]}}$, and we see that $\nabla^{[i]} f(x) = Q^{[i]} x + r^{[i]}$, and we propose the following update law:

$$
x_i^{[i]}(k+1) = \begin{cases} \Pi_{X_i}\left[x_i^{[i]}(k) - \gamma_i \left(Q^{[i]} x_i(k) + r^{[i]}\right)\right] & k \in K^i \\ x_i^{[i]}(k) & k \notin K^i \end{cases},
$$

where agent i uses some stepsize $\gamma_i > 0$. The advantage of the block-based update law can be seen above, as agent i only needs to know $Q^{[i]}$ and $r^{[i]}$. Requiring each agent to store the entirety of Q and r would require $O(n^2)$ storage space, while $Q^{[i]}$ and $r^{[i]}$ only require $O(n)$. For large quadratic programs, this block-based update law dramatically reduces each agent's onboard storage requirements, which promotes scalability.

In order to account for communication delays, we use $\tau_i^j(k)$ to denote the time at which the value of $x_i^{[j]}(k)$ was originally computed by agent j. For example, if agent j computes a state update at time k_a and immediately transmits it to agent i, then agent i may receive this state update at time $k_b > k_a$ due to communication delays. Then τ_i^j is defined so that $\tau_i^j(k_b) = k_a$. For K^i and τ_i^j, we assume the following.

Assumption 4: For all $i \in [N]$, the set K^i is infinite. Moreover, for all $i \in [N]$ and $j \in [N] \backslash \{i\}$, if $\{k_d\}_{d \in \mathbb{N}}$ is a sequence in K^i tending to infinity, then

$$
\lim_{d \to \infty} \tau_i^j(k_d) = \infty. \qquad \triangle
$$

Assumption 4 is simply a formalization of the requirement that no agent ever permanently stop updating and sharing its own state with any other agent. For $i \neq j$, the sets K^i and K^j do not need to have any relationship because agents' updates are asynchronous. Our proposed update law for all agents can then be written as follows.

Algorithm 1: For all $i \in [N]$ and $j \in [N] \backslash \{i\}$, execute

$$
x_i^{[i]}(k+1) = \begin{cases} \Pi_{X_i}\left[x_i^{[i]}(k) - \gamma_i \left(Q^{[i]} x_i(k) + r^{[i]}\right)\right] & k \in K^i \\ x_i^{[i]}(k) & k \notin K^i \end{cases}
$$

$$
x_i^{[j]}(k+1) = \begin{cases} x_j^{[j]}\left(\tau_i^j(k+1)\right) & i \text{ receives } j\text{'s state at } k+1 \\ x_i^{[j]}(k) & \text{otherwise} \end{cases} \qquad \diamond
$$

In Algorithm 1 we see that $x_i^{[j]}$ changes only when agent i receives a transmission directly from agent j; otherwise it remains constant. This implies that agent i can update its own state using an old value of agent j's state multiple times and can reuse different agents' states different numbers of times.

IV. CONVERGENCE OF ASYNCHRONOUS OPTIMIZATION

In this section, we prove convergence of Algorithm 1. This will be shown using Lyapunov-like convergence. We will derive stepsize bounds from these concepts that will be used to show asymptotic convergence of all agents.

A. Block-Maximum Norms

The convergence of Algorithm 1 will be measured using a block-maximum norm as in [28], [14], and [25]. Below, we define the block-maximum norm in terms of our partitioning vector p.

Definition 2: Let $x = \left[x^{[i]}\right]_p \in \mathbb{R}^n$, where $p = [n_1, n_2, \ldots, n_N]^T$. The norm of the full vector x is defined as the maximum 2-norm of any single block, i.e.,

$$\|x\|_{2,p} := \max_{i \in [N]} \|x^{[i]}\|_2. \qquad \blacktriangle$$

The following lemma allows us to upper-bound the induced block-maximum matrix norm by the norms of the individual blocks.

Lemma 1: For the matrix $B = \left[B_j^{[i]}\right]_p$ and induced matrix norm $\|B\|_{2,p}$,

$$\|B\|_{2,p} \leq \max_{i \in [N]} \sum_{j=1}^{N} \left\|B_j^{[i]}\right\|_2.$$

Proof: Proof in Appendix A. $\qquad \blacksquare$

B. Convergence Via Lyapunov Sub-Level Sets

We now analyze the convergence of Algorithm 1. We construct a sequence of sets, $\{X(s)\}_{s \in \mathbb{N}}$, based on work in [28] and [14]. These sets behave analogously to sub-level sets of a Lyapunov function, and they will enable an invariance type argument in our convergence proof. Below, we use $\hat{x} := \arg\min_{x \in X} f(x)$ for the minimizer of f. We state the following assumption on these sets, and below we will construct a sequence of sets that satisfies this assumption.

Assumption 5: There exists a collection of sets $\{X(s)\}_{s \in \mathbb{N}}$ that satisfies:

1) $\cdots \subset X(s+1) \subset X(s) \subset \cdots \subset X$

2) $\lim_{s \to \infty} X(s) = \{\hat{x}\}$

3) There exists $X_i(s) \subset X_i$ for all $i \in [N]$ and $s \in \mathbb{N}$ such that $X(s) = X_1(s) \times \ldots \times X_N(s)$

4) $\theta_i(y) \in X_i(s+1)$, where $\theta_i(y) := \Pi_{X_i}\left[y^{[i]} - \gamma_i \nabla^{[i]} f(y)\right]$ for all $y \in X(s)$ and $i \in [N]$. $\qquad \triangle$

Assumptions 5.1 and 5.2 jointly guarantee that the collection $\{X(s)\}_{s \in \mathbb{N}}$ is nested and that the sets converge to a singleton containing $\hat{x}$. Assumption 5.3 allows for the blocks of x to be updated independently by the agents, which allows for decoupled update laws. Assumption 5.4 ensures that state updates make only forward progress toward $\hat{x}$, which ensures that each set is forward-invariant in time. It is shown in [28] and [14] that the existence of such a sequence of sets implies asymptotic convergence of the asynchronous update law in Algorithm 1. We therefore use this strategy to show asymptotic convergence in this paper. We propose to use the construction

$$X(s) = \left\{y \in X : \|y - \hat{x}\|_{2,p} \leq q^s D_o\right\},$$

where we define $D_o := \max_{i \in [N]} \left\| x^i(0) - \hat{x} \right\|_{2,p}$, which is the block furthest from $\hat{x}$ onboard any agent at timestep zero, and where we define the constant

$$q = \max_{i \in [N]} \left\| I - \gamma_i Q_i^{[i]} \right\|_2 + \gamma_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2.$$

To show convergence, we will use the fact that each update contracts towards $\hat{x}$ by a factor of q, and will state a lemma that establishes bounds on every γ_i that imply $q \in (0, 1)$. Additionally, we will see that a proof of convergence using this method requires a block diagonal dominance condition on Q. This result will be used to show convergence of Algorithm 1 through satisfaction of Assumption 5.

If we wish for $q \in (0, 1)$, this condition can be restated as

$$\left\| I - \gamma_i Q_i^{[i]} \right\|_2 + \gamma_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2 < 1 \text{ for all } i \in [N]. \tag{3}$$

Note that because $Q = Q^T \succ 0$ and $Q_i^{[i]}$ is a diagonal submatrix of Q, we have $Q_i^{[i]} = Q_i^{[i]^T} \succ 0$. From this fact, we see $\left(\left\| Q_i^{[i]^{-1}} \right\|_2 \right)^{-1} = \lambda_{min} \left(Q_i^{[i]} \right)$, meaning that Assumption 3 holds. Then, in particular,

$$\lambda_{min} \left(Q_i^{[i]} \right) > \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2 \text{ for all } i \in [N].$$

The following lemma states an equivalent condition for Equation (3), which demonstrates the necessity and sufficiency of strict block diagonal dominance.

Lemma 2: Let $Q = Q^T = \left[Q_j^{[i]} \right]_p$, where $p = [n_1, n_2, \ldots, n_N]^T$. Additionally, let the $n \times n$ matrix $\Gamma = \text{diag}(\gamma_1 I_{n_1}, \gamma_2 I_{n_2}, ..., \gamma_N I_{n_N})$, where I_{n_i} is the identity matrix of size n_i and $\gamma_i > 0$. Then

$$\left\| I - \gamma_i Q_i^{[i]} \right\|_2 + \gamma_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2 < 1 \text{ for all } i \in [N]$$

if and only if

$$\lambda_{min} \left(Q_i^{[i]} \right) > \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2 \text{ and } \gamma_i < \frac{2}{\sum_{j=1}^{N} \left\| Q_j^{[i]} \right\|_2}$$

for all $i \in [N]$.

Proof: Proof in Appendix B. ∎

Note that γ_i only depends on $Q^{[i]}$. This lemma implies that γ_i can be chosen according to the conditions of Problem 1 such that $q \in (0, 1)$, given that Assumption 3 holds for Q. Choosing appropriate stepsizes for all $i \in [N]$ and recalling our construction of sets $\{X(s)\}_{s \in \mathbb{N}}$ as

$$X(s) = \left\{ y \in X : \| y - \hat{x} \|_{2,p} \leq q^s D_o \right\}, \tag{4}$$

we next show that Assumption 5 is satisfied, thereby ensuring convergence of Algorithm 1.

Theorem 1: If Assumptions 1-4 hold and $\Gamma = \text{diag}(\gamma_1 I_{n_1}, \gamma_2 I_{n_2}, ..., \gamma_N I_{n_N})$ satisfies the conditions in Lemma 2, then the collection of sets $\{X(s)\}_{s \in \mathbb{N}}$ as defined in Equation (4) satisfies Assumption 5.

Proof: Proof in Appendix C. ■

Regarding Problem 1, we therefore state the following:

Theorem 2: Algorithm 1 solves Problem 1 and asymptotically converges to $\hat{x}$.

Proof: Proof in Appendix D. ■

From these requirements, we see that agent i only needs to be initialized with $Q^{[i]}$ and $r^{[i]}$. Agents are then free to choose stepsizes independently, provided stepsizes obey the bounds established in Lemma 2.

V. CONVERGENCE RATE

Beyond asymptotic convergence, the structure of the sets $\{X(s)\}_{s\in\mathbb{N}}$ allows us to determine a convergence rate. To do so, we first define the notion of a *communication cycle*.

Definition 3: One *communication cycle* occurs when every agent has calculated a state update and this updated state has been sent to and received by every other agent. ▲

Once the last updated state has been received by the last agent, a communication cycle ends and another begins. It is only at the conclusion of the first communication cycle that each agents' copy of the ensemble state is moved from $X(0)$ to $X(1)$. Once another cycle is completed every agent's copy of the ensemble state is moved from $X(1)$ to $X(2)$. This process repeats indefinitely, and coupled with Assumption 5, means the convergence rate is geometric in the number of cycles completed, which we now show.

Theorem 3: Let Assumptions 1-5 hold and let $\gamma_i \in \left(0, \frac{2}{\sum_{j=1}^{N}\left\|Q_j^{[i]}\right\|_2}\right)$ for all $i \in [N]$. At time k, if $c(k)$ cycles have been completed, then $\|x_i(k) - \hat{x}\|_{2,p} \leq q^{c(k)} D_o$ for all $i \in [N]$.

Proof: Proof in Appendix E. ■

From the definition of q, we may write $q = \max_{i\in[N]} q_i$, where

$$q_i = \left\|I - \gamma_i Q_i^{[i]}\right\|_2 + \gamma_i \sum_{\substack{j=1\\j\neq i}}^{N} \left\|Q_j^{[i]}\right\|_2, \tag{5}$$

which illustrates the dependence of each q_i upon γ_i. As in all forms of gradient descent optimization, the choice of stepsizes has a significant impact on the convergence rate, which can be expressed through its effect on q. Therefore, we would like to determine the optimal stepsizes for each block in order to minimize q, which will accelerate convergence to a solution. Due to the structure of q, minimizing q_i for each $i \in [N]$ will minimize q. This fact leads to the following theorem:

Theorem 4: q is minimized when, for every $i \in [N]$,

$$\gamma_i = \frac{2}{\lambda_{max}\left(Q_i^{[i]}\right) + \lambda_{min}\left(Q_i^{[i]}\right)}.$$

Proof: Proof in Appendix F. ■

VI. REGULARIZATION AND CONVERGENCE RATE

In centralized optimization, regularization can be used to accelerate convergence by reducing the condition number of Q. It is well known that the condition number of Q, denoted k_Q, plays a significant role in the convergence

rate, with large condition numbers correlating to slow convergence rates. However in a decentralized setting it is difficult for agents to independently select regularizations such that k_Q is reduced, and harder still to know the magnitude of the reduction. In [26] it is shown that if the ratio of the largest to smallest regularization used in the network is less than k_Q, then the condition number of the regularized problem is guaranteed to be smaller. However, this requires global knowledge of k_Q, requires an upper bound on regularizations to somehow be agreed on, and institutes a lower bound on agents' choice of regularizations, all of which lead to the type of parametric coupling that we wish to avoid.

As stated in Problem 2, we want to allow agents to choose regularization parameters independently. Here, we therefore only require that agent i use a positive regularization parameter $\alpha_i > 0$. In Algorithm 1, this changes only agent i's updates to $x_i^{[i]}$, which now take the form

$$x_i^{[i]} = \Pi_{X_i} \left[x_i^{[i]}(k) - \gamma_i \left(Q^{[i]} x_i(k) + r^{[i]} + \alpha_i x_i^{[i]}(k) \right) \right].$$

Before we analyze the effects of independently chosen regularizations on convergence, we must first show that an algorithm that utilizes them will preserve the convergence properties of Algorithm 1. As shown in Equation (2), a regularized cost function takes the form

$$f_A(x) := \frac{1}{2} x^T (Q + A)x + r^T x,$$

where $Q + A$ is symmetric and positive definite because $Q = Q^T \succ 0$. We now state the following theorem that confirms that minimizing f_A succeeds.

Theorem 5: Suppose that $A = \text{diag}\left(\alpha_1 I_{n_1}, ..., \alpha_N I_{n_N} \right) \succ 0$, where agent i chooses α_i independently of all other agents. Then Algorithm 1 satisfies the conditions stated in Problem 2 when f_A is minimized.

Proof: Replacing Q with $Q + A$, all assumptions and conditions used to prove Theorem 2 hold, with the only modifications being the network will converge to $\hat{x}_A := \arg\min_{x \in X} f_A(x)$. These steps are similar to those used to prove Theorem 2 and are therefore omitted. ∎

Theorem 5 establishes that regularizing preserves asymptotic convergence, and we next turn to analyzing convergence rates. Because the condition number k_Q is a parameter that depends on the entirety of Q, and each agent only has access to a portion of Q, it is impossible for agents to know how their independent choices of regularizations affect k_Q. However, we can instead use q, which provides our convergence rate and can be directly manipulated by agents' choice of regularizations. Assume the optimal stepsize for block i is chosen as given in Equation (12). We then have

$$q_i = \frac{2 \sum_{j \neq i}^{N} \left\| Q_j^{[i]} \right\|_2 + \lambda_{max}\left(Q_i^{[i]} \right) - \lambda_{min}\left(Q_i^{[i]} \right)}{\lambda_{max}\left(Q_i^{[i]} \right) + \lambda_{min}\left(Q_i^{[i]} \right)}.$$

When we regularize the problem with A, the convergence parameter becomes $q_A = \max_i q_{\alpha_i}$, where

$$q_{\alpha_i} = \frac{2 \sum_{j \neq i}^{N} \left\| Q_j^{[i]} \right\|_2 + \lambda_{max}\left(Q_i^{[i]} \right) - \lambda_{min}\left(Q_i^{[i]} \right)}{\lambda_{max}\left(Q_i^{[i]} \right) + \lambda_{min}\left(Q_i^{[i]} \right) + 2\alpha_i}.$$

The only effect regularization has on q_i is adding $2\alpha_i$ to the denominator, meaning that *any* choice of positive regularization will result in $q_{\alpha_i} < q_i$, and thus all regularizations accelerate convergence. Using this fact, we can tailor parameter selections to attain a desired convergence rate. Assume we have a desired convergence rate for our system, corresponding to q^*. If we want to set $q_A \leq q^*$, we need $q_{\alpha_i} \leq q^*$ for all $i \in [N]$. Some algebraic manipulation of the above equation shows we therefore need to choose α_i such that

$$\alpha_i \geq \left(\frac{q_i}{q^*} - 1\right)\left(\frac{\lambda_{max}\left(Q_i^{[i]}\right) + \lambda_{min}\left(Q_i^{[i]}\right)}{2}\right).$$

Note that this term will be negative if $q_i < q^*$. That is, if the dynamics of block i are such that it will already converge faster than required by q^*, then there is no need to regularize that block. We now state the following theorem:

Theorem 6: Given $q^* \in (0,1)$, if for all $i \in [N]$ agent i chooses

$$\gamma_i = \frac{2}{\lambda_{max}\left(Q_i^{[i]}\right) + \lambda_{min}\left(Q_i^{[i]}\right) + 2\alpha_i}, \tag{6}$$

where

$$\alpha_i = \max\left\{\left(\frac{q_i}{q^*} - 1\right)\left(\frac{\lambda_{max}\left(Q_i^{[i]}\right) + \lambda_{min}\left(Q_i^{[i]}\right)}{2}\right), 0\right\},$$

then $q_A \leq q^*$.

Proof: Substitute Equation (6) into Equation (5). ∎

VII. REGULARIZATION ABSOLUTE ERROR BOUND: SET CONSTRAINED CASE

Regularization inherently results in a suboptimal solution because the system converges to $\Pi_X\left[\hat{x}_A\right]$ rather than $\Pi_X\left[\hat{x}\right]$. We therefore wish to bound this error by a function of the regularization matrix A. We define this error in two ways, $\left\|\Pi_X\left[\hat{x}\right] - \Pi_X\left[\hat{x}_A\right]\right\|_{2,p} = \max_i\left\|\Pi_{X_i}\left[\hat{x}^{[i]}\right] - \Pi_{X_i}\left[\hat{x}_A^{[i]}\right]\right\|_2$, which is the largest error of any one block in the network, and $\left|f\left(\Pi_X\left[\hat{x}\right]\right) - f\left(\Pi_X\left[\hat{x}_A\right]\right)\right|$, which is the difference in cost for the system between the regularized and unregularized cases. Note that in this section we are deriving descriptive error bounds in the sense that a network operator with access to each agent's local information can bound the error for the entire system, but no individual agent is expected to have access to this information.

Looking at the first definition of error, we find

$$\left\|\Pi_X\left[\hat{x}\right] - \Pi_X\left[\hat{x}_A\right]\right\|_{2,p} \leq \left\|\hat{x} - \hat{x}_A\right\|_{2,p}$$

which follows from the non-expansive property of the projection operator. Because of the fact that $\hat{x} = -Q^{-1}r$ and $\hat{x}_A = -(Q+A)^{-1}r$, we see

$$\left\|\hat{x} - \hat{x}_A\right\|_{2,p} = \left\|(Q^{-1} - (Q+A)^{-1})r\right\|_{2,p}.$$

Through use of the Woodbury matrix identity, one can see $Q^{-1} - (Q+A)^{-1} = (I + A^{-1}Q)^{-1}Q^{-1}$, because A is invertible. This gives

$$\left\|\hat{x} - \hat{x}_A\right\|_{2,p} \leq \left\|(I + A^{-1}Q)^{-1}\right\|_{2,p}\left\|Q^{-1}\right\|_{2,p}\left\|r\right\|_{2,p}. \tag{7}$$

Here $\|r\|_{2,p} = \max_i \left\|r^{[i]}\right\|_2$ is the largest norm of any individual block of r, which a network operator can gather from agents. However, the two other terms are $2, p$-norms of inverse matrices, which we do not assume the network operator has the ability to calculate. However, these terms can be bounded above using local information from agents according to the following lemma.

Lemma 3: If there is a block strictly diagonally dominant matrix $B = \left[B_j^{[i]}\right]_p$, where $p = [n_1, n_2, \ldots, n_N]^T$, and $\beta_p(B) = \min_i \left(\left\|B_i^{[i]^{-1}}\right\|_2^{-1} - \sum_{\substack{j=1 \\ j \neq i}}^N \left\|B_j^{[i]}\right\|_2 \right)$, then

$$\left\|B^{-1}\right\|_{2,p} \leq \beta_p^{-1}(B).$$

Proof: Theorem 2 in [29] establishes the above result for $\|\cdot\|_\infty$, and the proof for $\|\cdot\|_{2,p}$ follows identical steps. ∎

We note also that $I + A^{-1}Q$ is strictly block diagonally dominant, as $(A^{-1}Q)^{[i]} = \alpha_i^{-1} Q^{[i]}$. That is, each block of Q is multiplied by a positive scalar, which preserves the strict diagonal dominance of each block, as does the addition of I. Therefore, using Lemma 3 and $Q_i^{[i]} = Q_i^{[i]^T} \succ 0$ for all $i \in [N]$ we see $\left\|(I + A^{-1}Q)^{-1}\right\|_{2,p} \leq \beta_p^{-1}(I + A^{-1}Q)$ and $\left\|Q^{-1}\right\|_{2,p} \leq \beta_p^{-1}(Q)$, where $\beta_p(I + A^{-1}Q) = \min_i \left(1 + \alpha_i^{-1}\delta_i(Q)\right)$ and $\beta_p(Q) = \min_i \delta_i(Q)$. Finally,

$$\|\Pi_X[\hat{x}] - \Pi_X[\hat{x}_A]\|_{2,p} \leq \frac{\max_i \|r^{[i]}\|_2}{\beta_p(I + A^{-1}Q)\beta_p(Q)}. \tag{8}$$

The significance of this error bound is that if a network operator has access to $\|r^{[i]}\|_2$, α_i, and $\delta_i(Q)$ for all $i \in [N]$, which are locally known to every agent, the network operator can compute these bounds.

Defining $\Delta_{X_A} = \Pi_X[\hat{x}] - \Pi_X[\hat{x}_A]$, we find that $f(\Pi_X[\hat{x}]) - f(\Pi_X[\hat{x}_A]) = \frac{1}{2}(\Pi_X[\hat{x}] + \Pi_X[\hat{x}_A])^T Q(\Delta_{X_A}) + r^T(\Delta_{X_A})$, which gives

$$
\begin{aligned}
&|f(\Pi_X[\hat{x}]) - f(\Pi_X[\hat{x_A}])| \\
&= \left|\frac{1}{2}(\Pi_X[\hat{x}] + \Pi_X[\hat{x}_A])^T Q(\Delta_{X_A}) + r^T(\Delta_{X_A})\right| \\
&\leq \|\frac{1}{2}(\Pi_X[\hat{x}] + \Pi_X[\hat{x}_A])^T Q + r^T\|_{2,p}\|\Delta_{X_A}\|_{2,p} \\
&\leq (\|\frac{1}{2}(\Pi_X[\hat{x}] + \Pi_X[\hat{x}_A])^T Q\|_{2,p} + \|r^T\|_{2,p})\|\Delta_{X_A}\|_{2,p} \\
&\leq (\|\frac{1}{2}(\Pi_X[\hat{x}] + \Pi_X[\hat{x}_A])^T\|_{2,p}\|Q\|_{2,p} + \|r^T\|_{2,p})\|\Delta_{X_A}\|_{2,p}.
\end{aligned}
$$

Note that by definition, $\|x^T\|_{2,p} = \sum_{i=1}^N \|x^{[i]}\|_2$, and by Lemma 1 $\|B\|_{2,p} \leq \max_i \sum_{j=1}^N \left\|B_j^{[i]}\right\|_2$. Combining this with the non-expansive property of the projection operator gives

$$
\begin{aligned}
&|f(\Pi_X[\hat{x}]) - f(\Pi_X[\hat{x_A}])| \\
&\leq \left(\max_i \left\| \frac{1}{2}\left(\Pi_{X_i}\left[\hat{x}^{[i]}\right] + \Pi_{X_i}\left[\hat{x}_A^{[i]}\right]\right)^T\right\|_2 \max_i \sum_{j=1}^N \left\|Q_j^{[i]}\right\|_2 \right. \\
&\qquad \left. + \sum_{i=1}^N \left\|r^{[i]}\right\|_2 \right) \|\Delta_{X_A}\|_{2,p}.
\end{aligned}
$$

From Assumption 1, the set constraint for each block is compact, meaning agents can find the vector $\bar{x}^{[i]} = \arg\max_{x^{[i]} \in X_i} \|x^{[i]}\|_2$. Setting $M_{X_i} = \|\bar{x}^{[i]}\|_2$ and combining this with Equation (8) gives

$$|f(\Pi_X[\hat{x}]) - f(\Pi_X[\hat{x_A}])|$$

$$\leq \frac{(\max_{i \in [N]} M_{X_i} \max_{i \in [N]} \sum_{j=1}^{N} \left\|Q_j^{[i]}\right\|_2 + \sum_{i=1}^{N} \|r^{[i]}\|_2)}{\beta_p(I + A^{-1}Q)\beta_p(Q)}$$

$$+ \frac{\max_{i \in [N]} \|r^{[i]}\|_2}{\beta_p(I + A^{-1}Q)\beta_p(Q)}.$$

VIII. REGULARIZATION RELATIVE ERROR BOUND: UNCONSTRAINED CASE

In the previous section we derived a descriptive bound for the absolute error in both the states of the system and the cost due to regularizing. This bound is descriptive in the sense that given the agents' regularization choices, one can derive a bound describing error for the system. However given a desired error bound, agents cannot use the above rules to independently select regularizations due to the need for global information. Eliminating this dependence upon global information appears to be difficult because of the wide range of possibilities for the set constraints X_i. However, in the case where our problem does not have set constraints, i.e. Assumption 1 no longer holds and $X = \mathbb{R}^n$, we find that we can develop an entirely independent regularization selection rule to bound relative error. In particular, given some $\epsilon > 0$, we wish to bound the relative cost error via

$$\frac{|f(\hat{x}) - f(\hat{x}_A)|}{|f(\hat{x})|} \leq \epsilon.$$

If agents independently select regularizations, then α_i is selected using only knowledge of $Q^{[i]}$. Because we do not want to require agents to coordinate their regularizations to ensure the error bound is satisfied, we must develop independent regularization selection guidelines that depend only on $Q^{[i]}$.

Problem 3: Given the restriction that α_i can be chosen using only knowledge of $Q^{[i]}$ and ϵ, where $\epsilon \in (0, 1)$, develop independent regularization selection guidelines that guarantee

$$\frac{|f(\hat{x}) - f(\hat{x}_A)|}{|f(\hat{x})|} \leq \epsilon. \hspace{2cm} \triangle$$

For the unregularized problem, the solution is $\hat{x} = -Q^{-1}r$ and the optimal cost is $f(\hat{x}) = -\frac{1}{2}r^T Q^{-1} r$. For the regularized problem, the regularized solution is $\hat{x}_A = -P^{-1}r$, where $P = Q + A$, and the suboptimal cost is $f(\hat{x}_A) = \frac{1}{2}r^T P^{-1} Q P^{-1} r - r^T P^{-1} r$. Note that $f(\hat{x}) \leq f(\hat{x}_A) \leq 0$. That is, the cost can be upper-bounded by zero trivially for both the regularized and unregularized cases using $x = 0$. Therefore the optimal cost in both cases will be negative, with $f(\hat{x}) \leq f(\hat{x}_A)$. In particular, we know $f(\hat{x}) - f(\hat{x}_A) \leq 0$ and $f(\hat{x}) \leq 0$. Assuming $f(\hat{x}) \neq 0$, we can say

$$\frac{f(\hat{x}) - f(\hat{x}_A)}{f(\hat{x})} \geq 0.$$

That is,

$$\frac{|f(\hat{x}) - f(\hat{x}_A)|}{|f(\hat{x})|} \leq \epsilon \text{ if and only if } \frac{f(\hat{x}) - f(\hat{x}_A)}{f(\hat{x})} \leq \epsilon.$$

The solution to Problem 3 will be developed in two parts. First, it will be shown that the block diagonal dominance condition of Q allows A to be chosen under the restrictions of Problem 3 such that a certain eigenvalue condition of the matrix $A^{-1}Q$ is satisfied. Afterward, it will be shown that this condition on $A^{-1}Q$ is sufficient to guarantee the error bound given by ϵ is satisfied.

A. Block Gershgorin Circle Theorem

The Gershgorin Circle Theorem tells us that for any eigenvalue of a symmetric $n \times n$ matrix B, we have $\lambda_k(B) \in \bigcup_{k=1}^{n}[b_{k,k} - \sum_{j \neq k}^{n} |b_{k,j}|, b_{k,k} + \sum_{j \neq k}^{n} |b_{k,j}|]$ for all $k = 1, ..., n$. That is, every eigenvalue of B is contained within a union of intervals dependent on the rows of B. This implies that we can lower bound the minimum eigenvalue of B by $\lambda_{min}(B) \geq \min_k(b_{k,k} - \sum_{j \neq k}^{n} |b_{k,j}|)$. In the event that B is a strictly diagonally dominant matrix in the usual sense, i.e., $n_i = 1$ for all $i \in [N]$, this implies that every eigenvalue of B is positive, because $\lambda_{min}(B) \geq \min_k b_{k,k} - \sum_{j \neq k}^{n} |b_{k,j}| > 0$ for all $k = 1, ..., n$. Note further that if we let C be an $n \times n$ positive definite diagonal matrix, then $\lambda_{min}(CB) \geq \min_k c_{k,k}(b_{k,k} - \sum_{j \neq k}^{n} |b_{k,j}|) > 0$. That is, if B is a strictly diagonally dominant matrix and C is a positive definite diagonal matrix, then CB is strictly diagonally dominant.

Let B and C meet the criteria above, and now let us treat C as a design choice. Suppose we wish for the smallest eigenvalue of CB to be greater than or equal to a particular constant l, i.e., we want $\lambda_{min}(CB) \geq l$. From the Gershgorin Circle Theorem, we see this is true if $c_{k,k}(b_{k,k} - \sum_{j \neq k}^{n} |b_{k,j}|) \geq l$ for all $k = 1, ..., n$. This condition can be restated as

$$\text{if } c_{k,k} \geq \frac{l}{b_{k,k} - \sum_{j \neq k}^{n} |b_{k,j}|} \text{ for all } k = 1, ..., n,$$

then $\lambda_{min}(CB) \geq l$.

That is, given a strictly diagonally dominant matrix B and a positive constant l, the kth diagonal element of C can be chosen using only knowledge of the kth row of B and l such that $\lambda_{min}(CB) \geq l$. This intuition can be extended to a strictly block diagonally dominant matrix B using a block analogue of the Gershgorin Circle Theorem, as described below.

Lemma 4: For the matrix $B = \left[B_j^{[i]}\right]_p$, where $p = [n_1, n_2, \ldots, n_N]^T$, each eigenvalue $\lambda(B)$ satisfies

$$\left(\left\|\left(B_i^{[i]} - \lambda(B)I\right)^{-1}\right\|_2\right)^{-1} \leq \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\|B_j^{[i]}\right\|_2$$

for at least one $i \in [N]$.

Proof: See Theorem 2 in [27]. ∎

Note that

$$\left(\left\|\left(B_i^{[i]} - \lambda_{min}(B)I\right)^{-1}\right\|_2\right)^{-1} = \min_i \left|\lambda_{min}(B) - \lambda_i\left(B_i^{[i]}\right)\right|.$$

Additionally, let

$$\mu\left(B_i^{[i]}\right) = \arg\min_{\lambda_i} \left|\lambda_{min}(B) - \lambda_i\left(B_i^{[i]}\right)\right|,$$

which is the eigenvalue of $B_i^{[i]}$ closest to the minimum eigenvalue of B. Then,

$$\left(\left\| \left(B_i^{[i]} - \lambda_{min}(B)I \right)^{-1} \right\|_2 \right)^{-1} = \left| \lambda_{min}(B) - \mu \left(B_i^{[i]} \right) \right|.$$

From the block Gershgorin Circle Theorem, we then have

$$\lambda_{min}(B) \geq \mu \left(B_i^{[i]} \right) - \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| B_j^{[i]} \right\|_2 \text{ for at least one } i \in [N].$$

Because $\mu \left(B_i^{[i]} \right) \geq \lambda_{min} \left(B_i^{[i]} \right)$, we can say $\lambda_{min}(B) \geq \delta_i(B)$ for at least one $i \in [N]$.

Just as before, if B is strictly block diagonally dominant, then every eigenvalue of B is positive. Now let $C = \left[C_j^{[i]} \right]_p$, with $C_i^{[i]} = c_i I$ for every $i \in [N]$ and $C_j^{[i]} = 0$ when $j \neq i$. In the same manner as above, we find

$$\text{if } c_i \geq \frac{l}{\delta_i(B)} \text{ for all } i \in [N], \tag{9}$$

then $\lambda_{min}(CB) \geq l$.

That is, c_i can be chosen using only knowledge of $B^{[i]}$ and l. This brings us back to the restrictions imposed in Problem 3. For reasons that will be shown in the following subsection, choose $B = Q$, $C = A^{-1}$, and $l = \frac{1-\sqrt{\epsilon}}{\sqrt{\epsilon}}$. Assuming each block uses a scalar regularization, i.e. $c_i = \frac{1}{\alpha_i}$ where $\alpha_i > 0$, we have the following lemma

Lemma 5: Let Assumptions 2 and 3 hold for the matrix Q with respect to the partitioning vector $p = [n_1, n_2, \ldots, n_N]^T$. Let $A = \left[A_j^{[i]} \right]_p$, with $A_i^{[i]} = \alpha_i I$ for every $i \in [N]$ and $A_j^{[i]} = 0$ when $j \neq i$. If we have $\alpha_i \leq \frac{\sqrt{\epsilon}}{1-\sqrt{\epsilon}} \delta_i(Q)$ for all $i \in [N]$, then $\lambda_{min} \left(A^{-1}Q \right) \geq \frac{1-\sqrt{\epsilon}}{\sqrt{\epsilon}}$.

Proof: Use Equation (9) and substitute $C = A^{-1}$, $B = Q$, and $l = \frac{1-\sqrt{\epsilon}}{\sqrt{\epsilon}}$. ∎

We have shown this eigenvalue condition can be satisfied according to the conditions in Problem 3, i.e. $A^{[i]}$ is chosen using only knowledge of $Q^{[i]}$ and ϵ. The following subsection will show this condition is sufficient to satisfy the error bound in Problem 3.

B. Error Bound Satisfaction

Proof of error bound satisfaction will be done using the following lemma.

Lemma 6: Let $f(x) = \frac{1}{2}x^T Q x + r^T x$, where $Q = Q^T \succ 0$, $Q \in \mathbb{R}^{n \times n}$, and $r, x \in \mathbb{R}^n$. Let $\hat{x} = \arg\min_{x \in \mathbb{R}^n} f(x)$ and $\hat{x}_A = \arg\min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2}x^T A x$, where $A \succ 0$ and diagonal. Additionally, let $\epsilon \in [0, 1]$. If

$$\frac{1 - \sqrt{\epsilon}}{\sqrt{\epsilon}} \leq \lambda_{min}(A^{-1}Q), \text{ then } \frac{|f(\hat{x}) - f(\hat{x}_A)|}{|f(\hat{x})|} \leq \epsilon.$$

Proof: Proof in Appendix G. ∎

With these lemmas, we now present the following theorem.

Theorem 7: Let Assumptions 2 and 3 hold for the matrix Q with respect to the partitioning vector $p = [n_1, n_2, \ldots, n_N]^T$. Let $A = \left[A_j^{[i]} \right]_p$, with $A_i^{[i]} = \alpha_i I$ for every $i \in [N]$ and $A_j^{[i]} = 0$ when $j \neq i$. Let

$f(x) = \frac{1}{2}x^T Q x + r^T x$, where $r, x \in \mathbb{R}^n$. Let $\hat{x} = \arg\min_{x \in \mathbb{R}^n} f(x) = -Q^{-1}r$ and $\hat{x}_A = \arg\min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2}x^T A x = -P^{-1}r$, where $P = Q + A$. Additionally, let $\epsilon \in [0,1]$. If

$$\alpha_i \le \frac{\sqrt{\epsilon}}{1 - \sqrt{\epsilon}} \delta_i(Q) \text{ for all } i \in [N],$$

then,

$$\frac{|f(\hat{x}) - f(\hat{x}_A)|}{|f(\hat{x})|} \le \epsilon$$

Proof: Lemma 5 shows that the regularization selection rules presented above, along with Assumption 3, imply that $\frac{1 - \sqrt{\epsilon}}{\sqrt{\epsilon}} \le \lambda_{min}(A^{-1}Q)$. Lemma 6 shows that $\frac{1 - \sqrt{\epsilon}}{\sqrt{\epsilon}} \le \lambda_{min}(A^{-1}Q)$ implies that $\frac{|f(\hat{x}) - f(\hat{x}_A)|}{|f(\hat{x})|} \le \epsilon$. ∎

Additionally, we can derive a similar bound for relative error in the solution itself. Defining this error as $\frac{\|\hat{x} - \hat{x}_A\|_{2,p}}{\|\hat{x}\|_{2,p}}$ and using Equation (7) we see

$$\frac{\|\hat{x} - \hat{x}_A\|_{2,p}}{\|\hat{x}\|_{2,p}} = \frac{\|(I + A^{-1}Q)^{-1}Q^{-1}r\|_{2,p}}{\|Q^{-1}r\|_{2,p}}$$
$$\le \frac{\|(I + A^{-1}Q)^{-1}\|_{2,p}\|Q^{-1}r\|_{2,p}}{\|Q^{-1}r\|_{2,p}} = \|(I + A^{-1}Q)^{-1}\|_{2,p}$$
$$\le \frac{1}{\min_{i \in [N]}\left[1 + \alpha_i^{-1}\delta_i(Q)\right]}.$$

If we wish for agents to select regularizations such that the above error is less than a given constant η, we see this is accomplished if

$$\frac{1}{\eta} \le \min_{i \in [N]} 1 + \alpha_i^{-1}\delta_i(Q)$$
$$\alpha_i \le \frac{\eta}{1 - \eta}\delta_i(Q) \text{ for all } i \in [N].$$

This rule has the same structure as the one in Theorem 7, with the only difference being there is no square root taken of η.

Note that throughout this section it was assumed that A is invertible, which is true if $\alpha_i > 0$ for all $i \in [N]$. However in scenarios where there is no need for a particular agent to regularize, e.g. $q_i < q^*$, that agent can choose $\alpha_i = 0$ for all practical applications. This is because all of the above analysis holds if α_i is chosen to be a small positive value, which can be set arbitrarily close to zero.

C. Trade-Off Analysis

There is an inherent trade-off between the speed at which we reach a solution and the quality of that solution. Theorem 6 provides a lower bound on α_i that allows us to converge at any speed we wish, while Theorem 7 provides an upper bound on α_i that allows us to bound the cost error between the solution we find and the optimal solution. However, in general, there is no reason to expect these two bounds to be compatible in the sense that α_i can be chosen such that both are satisfied for all $i \in [N]$. Therefore, when implemented, it is likely that the network operator will be able to choose whether speed or accuracy is more critical for the specific problem. If
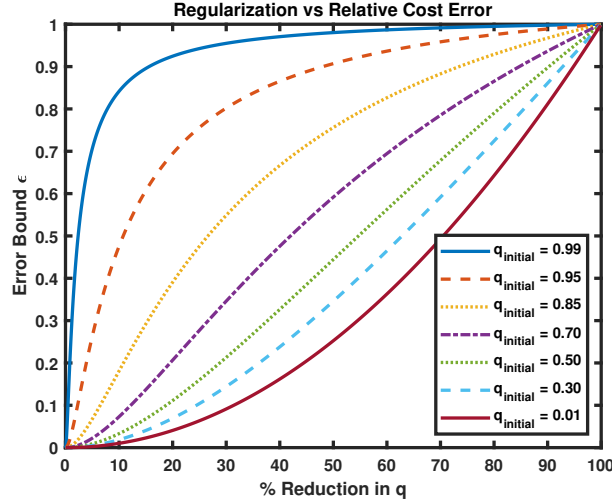
Fig. 1. The percent reduction in q due to regularization plotted vs the relative cost error bound that regularization induces, with different lines plotting this relationship for QPs with different initial values for q.

speed is mission-critical, then agents may select the smallest regularizations required to match that speed, and if accuracy is mission-critical, agents may select the largest regularizations that obey the specific error bound.

IX. SIMULATION

To visualize the trade-off between speed and error when regularizing, we generate seven QPs, each with 100 diagonally dominant blocks. The QPs are generated to have initial convergence parameters of $q_{initial} = 0.99$, 0.95, 0.85, 0.70, 0.50, 0.30, and 0.01. For each QP, A is independently chosen according to Theorem 6 such that q is reduced by percentages ranging from 0% to 100%, and this percentage reduction is plotted against the corresponding error bound given by Theorem 7 in Figure 1. For example, the data for the QP with $q_{initial} = 0.85$ is plotted by the yellow dotted line in Figure 1, and one can see that if this QP is regularized to reduce q by 10% (i.e., a reduction from 0.85 to 0.765), the relative error in cost can be upper bounded by approximately $\epsilon = 18\%$.

There are two main takeaways from Figure 1. The first is that, as expected, larger regularizations result in a larger relative error bound, which is upper bounded by 1. This is because $q \to 0$ as $A \to \infty$, $f(\hat{x}_A) \to 0$ as $A \to 0$, and $\epsilon \to 1$ as $f(\hat{x}_A) \to 0$. The second is that the larger $q_{initial}$ is, the more sensitive the error bound for the QP is to regularizing. That is, if $q_{initial}$ is thought of as a condition number, then "poorly conditioned" QPs will have larger errors due to regularizing.

A second simulation was run to demonstrate the convergence properties due to regularizing. One QP was generated with 100 blocks and $q_{initial} = 0.85$. Three different regularization matrices were chosen according to Theorem 6, called A_5, A_{15}, and A_{45}, such that q is reduced by 5%, 15%, and 45%, respectively. The blocks are then distributed among 100 agents, who have a 10% chance of computing an update and a 1% chance of transmitting a state to each other agent at each timestep. Four simulations were run, one solving the unregularized QP, and three others using each regularization matrix. The 2-norm of the system error to the unregularized solution, $\|x(k) - \hat{x}\|_2$, is
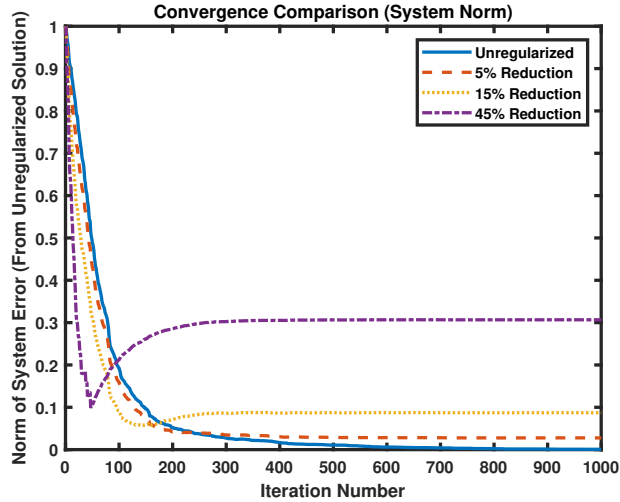
Fig. 2. Network error convergence of Algorithm 1 when unregularized vs regularizing such that q is reduced by 5%, 15%, and 45%.

plotted for each simulation against iteration number in Figure 2.

As expected, only the unregularized case converges to the unregularized solution, while the other cases converge to other solutions whose distances to the unregularized solution grow with larger regularizations. However, the cases with larger regularizations initially converge to $\hat{x}$ faster, up to a point. That is, larger regularizations mean the system will initially move toward $\hat{x}$ faster, but will reach the turn-off point, where the system error grows again, earlier and further away from $\hat{x}$. This behavior suggests a vanishing regularization scheme, where A shrinks to zero with time, may lead to accelerated convergence to the exact solution $\hat{x}$. Note also that convergence even in the unregularized case is non-monotone, and at times the norm of the system error may even grow due to the asynchronous nature of of communications, but Theorem 2 guarantees these growths are bounded and error will converge to zero.

X. CONCLUSIONS

We have developed a distributed quadratic programming framework that converges under totally asynchronous conditions. This framework allows agents to select stepsizes and regularizations independently of one another, using only knowledge of their block of the QP, that guarantee a specified global convergence rate and cost error bound. Future work will apply these developments to quadratic programs with functional constraints.

REFERENCES

[1] Z.-Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE Journal on selected areas in communications*, vol. 24, no. 8, pp. 1426–1438, 2006.

[2] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[3] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2520–2525.

[4] M. Chiang *et al.*, "Geometric programming for communication systems," *Foundations and Trends® in Communications and Information Theory*, vol. 2, no. 1–2, pp. 1–154, 2005.

[5] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[6] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[7] A. I. Chen and A. Ozdaglar, "A fast distributed proximal-gradient method," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 601–608.

[8] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.

[9] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[10] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, p. 48, 2009.

[11] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of opt. theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.

[12] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1291–1306, 2011.

[13] I. Rodriguez-Lujan, R. Huerta, C. Elkan, and C. S. Cruz, "Quadratic programming feature selection," *Journal of Machine Learning Research*, vol. 11, no. Apr, pp. 1491–1516, 2010.

[14] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall, 1989, vol. 23.

[15] R. Carli, G. Notarstefano, L. Schenato, and D. Varagnolo, "Distributed quadratic programming under asynchronous and lossy communications via newton-raphson consensus," in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 2514–2520.

[16] A. Teixeira, E. Ghadimi, I. Shames, H. Sandberg, and M. Johansson, "Optimal scaling of the admm algorithm for distributed quadratic programming," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 6868–6873.

[17] C.-P. Lee and D. Roth, "Distributed box-constrained quadratic optimization for dual linear svm," in *International Conference on Machine Learning*, 2015, pp. 987–996.

[18] K. Lee and R. Bhattacharya, "On the convergence analysis of asynchronous distributed quadratic programming via dual decomposition," *arXiv preprint arXiv:1506.05485*, 2015.

[19] A. Kozma, J. V. Frasch, and M. Diehl, "A distributed method for convex quadratic programming problems arising in optimal control of distributed systems," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 1526–1531.

[20] M. Todescato, G. Cavraro, R. Carli, and L. Schenato, "A robust block-jacobi algorithm for quadratic programming under lossy communications," *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 126–131, 2015.

[21] P. Wang, S. Mou, J. Lian, and W. Ren, "Solving a system of linear equations: From centralized to distributed algorithms," *Annual Reviews in Control*, 2019.

[22] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 1859–1864.

[23] J. Koshal, A. Nedić, and U. V. Shanbhag, "Multiuser optimization: Distributed algorithms and error analysis," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 1046–1081, 2011.

[24] M. T. Hale, A. Nedić, and M. Egerstedt, "Cloud-based centralized/decentralized multi-agent optimization with communication delays," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 700–705.

[25] M. T. Hale, A. Nedić, and M. Egerstedt, "Asynchronous multiagent primal-dual optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4421–4435, 2017.

[26] M. Ubl and M. Hale, "Totally asynchronous distributed quadratic programming with independent stepsizes and regularizations," *arXiv preprint arXiv:1903.08618*, 2019.

[27] D. G. Feingold, R. S. Varga *et al.*, "Block diagonally dominant matrices and generalizations of the gerschgorin circle theorem." *Pacific Journal of Mathematics*, vol. 12, no. 4, pp. 1241–1250, 1962.

[28] D. P. Bertsekas and J. N. Tsitsiklis, "Convergence rate and termination of asynchronous iterative algorithms," in *Proceedings of the 3rd International Conference on Supercomputing*, 1989, pp. 461–470.

[29] J. M. Varah, "A lower bound for the smallest singular value of a matrix," *Linear Algebra and its Applications*, vol. 11, no. 1, pp. 3–5, 1975.

[30] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas-Second Edition*. Princeton university press, 2009.

Matthew Ubl is a PhD student at the University of Florida, where he is an Institute for Networked Autonomous Systems Fellow and a recipient of the Graduate Student Preeminence Award. He received his Bachelor's degree in Aerospace Engineering from the University of Central Florida in 2018. His current research interests are in asynchronous multi-agent coordination, with particular focus upon multi-agent optimization with impaired and unreliable communications.

Matthew Hale is an Assistant Professor of Mechanical and Aerospace Engineering at the University of Florida. He received his BSE in Electrical Engineering summa cum laude from the University of Pennsylvania in 2012, and his MS and PhD in Electrical and Computer Engineering from the Georgia Institute of Technology in 2015 and 2017, respectively. He directs the Control, Optimization, and Robotics Engineering (CORE) Lab at the University of Florida, and his research interests include multi-agent systems, mobile robotics, privacy in control, distributed optimization, and graph theory. He was the Teacher of the Year in the UF Department of Mechanical and Aerospace Engineering for the 2018-2019 school year, and he received an NSF CAREER Award in 2020 for his work on privacy in control systems.

Appendix A

Proof of Lemma 1: By definition of the maximum norm,

$$\|B\|_{2,p} = \sup_{\|x\|_{2,p}=1} \|Bx\|_{2,p} = \sup_{\|x\|_{2,p}=1} \max_{i \in [N]} \left\| B^{[i]}x \right\|_2 .$$

Since $B^{[i]} = \left[B_1^{[i]} \ B_2^{[i]} \ ... \ B_N^{[i]} \right]$, we can now write $B^{[i]}x = B_1^{[i]}x^{[1]} + B_2^{[i]}x^{[2]} + ... + B_N^{[i]}x^{[N]}$. Therefore,

$$\|B\|_{2,p} = \sup_{\|x\|_{2,p}=1} \max_{i \in [N]} \left\| B_1^{[i]}x^{[1]} + ... + B_N^{[i]}x^{[N]} \right\|_2 .$$

By the triangle inequality, we have

$$\|B\|_{2,p} \leq \sup_{\|x\|_{2,p}=1} \max_{i \in [N]} \sum_{j=1}^{N} \left\| B_j^{[i]}x^{[j]} \right\|_2 .$$

The condition $\|x\|_{2,p} = 1$ implies $\left\| x^{[i]} \right\|_2 \leq 1$ for all $i \in [N]$. Therefore, for each element in the sum above, we can write $\left\| B_j^{[i]}x^{[j]} \right\|_2 \leq \sup_{\|x^{[j]}\|_2=1} \left\| B_j^{[i]}x^{[j]} \right\|_2 = \left\| B_j^{[i]} \right\|_2$. Substituting this above completes the proof. ∎

APPENDIX B

Proof of Lemma 2: Because $Q_i^{[i]} = Q_i^{[i]^T} \succ 0$, we see that

$$
\left\| I - \gamma_i Q_i^{[i]} \right\|_2
$$
$$
= \max \left\{ \left| \lambda_{min} \left(I - \gamma_i Q_i^{[i]} \right) \right|, \left| \lambda_{max} \left(I - \gamma_i Q_i^{[i]} \right) \right| \right\}
$$
$$
= \max \left\{ \left| 1 - \gamma_i \lambda_{min} \left(Q_i^{[i]} \right) \right|, \left| 1 - \gamma_i \lambda_{max} \left(Q_i^{[i]} \right) \right| \right\},
$$

which allows us to write

$$
\left\| I - \gamma_i Q_i^{[i]} \right\|_2 + \gamma_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2 < 1
$$

if and only if both

$$
\left| 1 - \gamma_i \lambda_{min} \left(Q_i^{[i]} \right) \right| < 1 - \gamma_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2
$$

and

$$
\left| 1 - \gamma_i \lambda_{max} \left(Q_i^{[i]} \right) \right| < 1 - \gamma_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2.
$$

The first inequality will be true if and only if both

$$
\lambda_{min}(Q_i^{[i]}) > \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2 \tag{10}
$$

and

$$
\gamma_i < \frac{2}{\lambda_{min}(Q_i^{[i]}) + \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2},
$$

and the second will be true if and only if both

$$
\lambda_{max}(Q_i^{[i]}) > \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2
$$

and

$$
\gamma_i < \frac{2}{\lambda_{max}(Q_i^{[i]}) + \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2}. \tag{11}
$$

Taking the most restrictive of these conditions, we can write

$$
\left\| I - \gamma_i Q_i^{[i]} \right\|_2 + \gamma_i \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2 < 1
$$

if and only if Equations (10) and (11) hold. ∎

APPENDIX C

Proof of Theorem 1: For Assumption 5.1, by definition we have

$$X(s + 1) = \left\{ y \in X : \|y - \hat{x}\|_{2,p} \leq q^{s+1} D_o \right\}.$$

Since $q \in (0, 1)$, we have $q^{s+1} < q^s$, which results in $\|y - \hat{x}\|_{2,p} \leq q^{s+1} D_o < q^s D_o$. Then $y \in X(s + 1)$ implies $y \in X(s)$ and $X(s + 1) \subset X(s) \subset X$, as desired.

For Assumption 5.2 we find

$$\lim_{s \to \infty} X(s) = \lim_{s \to \infty} \{ y \in X : \|y - \hat{x}\|_{2,p} \leq q^s D_o \} = \{\hat{x}\}.$$

The structure of the weighted block-maximum norm then allows us to see that $\|y - \hat{x}\|_{2,p} \leq q^s D_o$ if and only if $\|y^{[i]} - \hat{x}^{[i]}\|_2 \leq q^s D_o$ for all $i \in [N]$. It then follows that

$$X_i(s) = \left\{ y^{[i]} \in X_i : \|y^{[i]} - \hat{x}^{[i]}\|_2 \leq q^s D_o \right\},$$

which gives $X(s) = X_1(s) \times ... \times X_N(s)$, thus satisfying Assumption 5.3.

We then see that, for $y \in X(s)$,

$$\left\| \theta_i(y) - \hat{x}^{[i]} \right\|_2 = \left\| \Pi_{X_i} \left[y^{[i]} - \gamma_i \left(Q^{[i]} y + r^{[i]} \right) \right] \right.$$
$$\left. - \Pi_{X_i} \left[\hat{x}^{[i]} - \gamma_i \left(Q^{[i]} \hat{x} + r^{[i]} \right) \right] \right\|_2,$$

which follows from the definition of $\theta_i(y)$ and the fact that $\hat{x}^{[i]} = \Pi_{X_i} [\theta_i(\hat{x})]$. Using the non-expansive property of the projection operator $\Pi_{X_i} [\cdot]$, we find

$$\left\| \theta_i(y) - \hat{x}^{[i]} \right\|_2 \leq \left\| y^{[i]} - \hat{x}^{[i]} - \gamma_i Q^{[i]} (y - \hat{x}) \right\|_2$$
$$= \left\| \left(I^{[i]} - \gamma_i Q^{[i]} \right) (y - \hat{x}) \right\|_2$$
$$\leq \max_{i \in [N]} \left\| \left(I^{[i]} - \gamma_i Q^{[i]} \right) (y - \hat{x}) \right\|_2$$
$$= \| (I - \Gamma Q) (y - \hat{x}) \|_{2,p}$$
$$\leq \| I - \Gamma Q \|_{2,p} \| y - \hat{x} \|_{2,p},$$

which follows from our definition of the block-maximum norm. From Lemmas 1 and 2 we know $\| I - \Gamma Q \|_{2,p} \leq q < 1$, and using the hypothesis that $y \in X(s)$, we find

$$\left\| \theta_i(y) - \hat{x}^{[i]} \right\|_2 \leq q \| y - \hat{x} \|_{2,p} \leq q^{s+1} D_o,$$

which shows $\theta_i(y) \in X_i(s + 1)$ and Assumption 5.4 is satisfied. ∎

APPENDIX D

Proof of Theorem 2: Theorem 1 shows the construction of the sets $\{X(s)\}_{s \in \mathbb{N}}$ satisfies Assumption 5, and from [28] and [14] we see this implies asymptotic convergence of Algorithm 1 for all $i \in [N]$. The total asynchrony

required by Problem 1 is incorporated by not requiring delay bounds, and agents do not require any coordination in selecting stepsizes because the bound on γ_i depends only upon $Q^{[i]}$, which means that all of the criteria of Problem 1 are satisfied. ∎

APPENDIX E

Proof of Theorem 3: From the definition of D_o, for all $i \in [N]$ we have $x_i(0) \in X(0)$. If agent i computes a state update, then $\theta_i(x_i(0)) \in X_i(1)$ and after one cycle is completed, say at time k, we have $x_i(k) \in X(1)$ for all i. Iterating this process, after $c(k)$ cycles have been completed by some time k, $x_i(k) \in X(c(k))$. The result follows by expanding the definition of $X(c(k))$. ∎

APPENDIX F

Proof of Theorem 4: If $\gamma_i \leq \frac{2}{\lambda_{max}\left(Q_i^{[i]}\right) + \lambda_{min}\left(Q_i^{[i]}\right)}$, then

$$q_i = 1 - \gamma_i \left(\lambda_{min}\left(Q_i^{[i]}\right) - \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2 \right),$$

and if $\gamma_i \geq \frac{2}{\lambda_{max}\left(Q_i^{[i]}\right) + \lambda_{min}\left(Q_i^{[i]}\right)}$, then

$$q_i = -1 + \gamma_i \left(\lambda_{max}\left(Q_i^{[i]}\right) + \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| Q_j^{[i]} \right\|_2 \right).$$

That is, when $\gamma_i \leq \frac{2}{\lambda_{max}\left(Q_i^{[i]}\right) + \lambda_{min}\left(Q_i^{[i]}\right)}$, the relationship between q_i and γ_i is a line with negative slope, and when $\gamma_i \geq \frac{2}{\lambda_{max}\left(Q_i^{[i]}\right) + \lambda_{min}\left(Q_i^{[i]}\right)}$ the relationship is a line with positive slope. Then q_i is minimized at the point where the slope changes sign, which occurs when

$$\gamma_i = \frac{2}{\lambda_{max}\left(Q_i^{[i]}\right) + \lambda_{min}\left(Q_i^{[i]}\right)}. \tag{12}$$

If every q_i has been minimized, then by definition q has been minimized. ∎

APPENDIX G

Proof of Lemma 6: To facilitate this proof, we first present the following facts to which we will repeatedly refer:

Fact 1: If B is a square matrix such that $0 < \lambda_{min}(B) \leq \lambda_{max}(B)$, then $\lambda_{max}(B^{-1}) = \lambda_{min}^{-1}(B)$.

Fact 2: If B is a square matrix such that $0 < \lambda_{min}(B) \leq \lambda_{max}(B)$, then $\lambda_{min}(B^2) = \lambda_{min}^2(B)$.

Fact 3: If B is a square matrix, then $-\lambda_{max}(B) = \lambda_{min}(-B)$.

Fact 4: If B is a square matrix and C is an invertible matrix of the same dimension, then $\lambda_i(C^{-1}BC) = \lambda_i(B)$ for all i.

Fact 5: If $B = B^T \preceq 0$ and C is an invertible matrix of the same dimension, then $\lambda_i(C^TBC) \leq 0$ for all i.

Facts 1-3 can be easily shown, Fact 4 simply states eigenvalues are invariant under a similarity transform, and Fact 5 is a corollary of Sylvester's Law of Inertia [30, Fact 5.8.17].

Bearing these facts in mind, we first rearrange the condition in the lemma statement to find

$$\frac{1}{\sqrt{\epsilon}} - 1 \leq \lambda_{min}(A^{-1}Q)$$

$$\frac{1}{\sqrt{\epsilon}} \leq 1 + \lambda_{min}(A^{-1}Q) = \lambda_{min}(I + A^{-1}Q)$$

$$= \lambda_{min}(A^{-1}(A + Q)) = \lambda_{min}(A^{-1}P)$$

$$\lambda_{min}^{-1}(A^{-1}P) \leq \sqrt{\epsilon}.$$

From Fact 1, it follows that $\lambda_{max}(P^{-1}A) \leq \sqrt{\epsilon}$ and $\lambda_{max}^2(P^{-1}A) \leq \epsilon$. From Fact 2, $\lambda_{max}((P^{-1}A)^2) \leq \epsilon$, which implies $-\epsilon \leq -\lambda_{max}((P^{-1}A)^2)$. From Fact 3,

$$-\epsilon \leq \lambda_{min}(-(P^{-1}A)^2)$$

$$1 - \epsilon \leq 1 + \lambda_{min}(-(P^{-1}A)^2) = \lambda_{min}(I - (P^{-1}A)^2)$$

$$= \lambda_{min}((I + P^{-1}A)(I - P^{-1}A)).$$

Note that $I - P^{-1}A = P^{-1}(P - A) = P^{-1}Q$, therefore

$$1 - \epsilon \leq \lambda_{min}((I + P^{-1}A)P^{-1}Q)$$

$$1 - \epsilon \leq \lambda_{min}((P^{-1} + P^{-1}AP^{-1})Q).$$

Note that $P^{-1} + P^{-1}AP^{-1} = P^{-1} + P^{-1}(P - Q)P^{-1} = 2P^{-1} - P^{-1}QP^{-1}$, therefore $1 - \epsilon \leq \lambda_{min}((2P^{-1} - P^{-1}QP^{-1})Q)$, which implies

$$0 \leq -(1 - \epsilon) + \lambda_{min}((2P^{-1} - P^{-1}QP^{-1})Q)$$

$$0 \leq \lambda_{min}(-(1 - \epsilon)I + (2P^{-1} - P^{-1}QP^{-1})Q).$$

From Fact 3, $0 \leq -\lambda_{max}((1 - \epsilon)I - (2P^{-1} - P^{-1}QP^{-1})Q)$ and $\lambda_{max}((1 - \epsilon)I - (2P^{-1} - P^{-1}QP^{-1})Q) \leq 0$. From Fact 4, taking $C = Q^{-\frac{1}{2}}$

$$\lambda_{max}((1 - \epsilon)I - Q^{\frac{1}{2}}(2P^{-1} - P^{-1}QP^{-1})Q^{\frac{1}{2}}) \leq 0.$$

Note that the matrix above is symmetric. Therefore, from Fact 5, taking $C = Q^{-\frac{1}{2}}$, we have

$$\lambda_{max}((1 - \epsilon)Q^{-1} - 2P^{-1} + P^{-1}QP^{-1}) \leq 0.$$

Note that the matrix above is still symmetric. Therefore, we can write $(1 - \epsilon)Q^{-1} - 2P^{-1} + P^{-1}QP^{-1} \preceq 0$, which implies $Q^{-1} - 2P^{-1} + P^{-1}QP^{-1} \preceq \epsilon Q^{-1}$.

This means that for any arbitrary vector x of dimension n, $x^T(Q^{-1} - 2P^{-1} + P^{-1}QP^{-1})x \leq x^T(\epsilon Q^{-1})x$, and $x^T Q^{-1}x - 2x^T P^{-1}x + x^T P^{-1}QP^{-1}x \leq \epsilon x^T Q^{-1}x$.

Assuming $x \neq 0$, $x^T Q^{-1} x$ is a positive scalar. Dividing both sides by this term gives

$$\frac{x^T Q^{-1} x - 2 x^T P^{-1} x + x^T P^{-1} Q P^{-1} x}{x^T Q^{-1} x} \leq \epsilon.$$

Because this relation is true for any arbitrary vector, we can choose $x = r$ and multiply by $\frac{-\frac{1}{2}}{-\frac{1}{2}}$ to find

$$\frac{-\frac{1}{2} r^T Q^{-1} r - (\frac{1}{2} r^T P^{-1} Q P^{-1} r - r^T P^{-1} r)}{-\frac{1}{2} r^T Q^{-1} r} \leq \epsilon,$$

and substituting returns the desired result. ∎