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

Matthew Ubl and Matthew Hale, Member, IEEE

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 multiagent 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 agents 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. Larger regularizations accelerate convergence but increase the error in the solution obtained, and we quantify the tradeoff between these transmissions. Furthermore, we allow agents 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. Larger regularizations accelerate convergence but increase the error in the solution obtained, and we quantify the tradeoff between

Index Terms—Autonomous agents, distributed algorithms, optimization, network control systems, quadratic programming.

I. INTRODUCTION

ONVEX 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 team of agents. As the number of agents grows, it can be difficult or impossible to ensure synchrony in their computations and communications, and there has 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. Furthermore, even if such checking is possible, it will be difficult to actually attain connectivity over the required intervals with unreliable communications. Because we are addressing cases when communications are beyond our control, we are interested in developing an algorithmic framework that succeeds without making any assumptions regarding the system’s communications, such as requiring delay bounds.

In this paper, we develop a totally asynchronous quadratic programming (QP) framework for multiagent 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.

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. This work is also somewhat related to distributed solutions to systems of linear equations, e.g., [21].
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 in their algorithms. One could envision first 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 regularize.

To the best of our knowledge, few works have considered both independent stepsizes and regularizations. The most relevant is [23], which allows agents to independently select stepsizes and regularizations from a globally broadcast range without affecting convergence. This paper is different in that we do not require global bounds, which must be calculated and broadcast by an entity with global knowledge. Instead, we allow agents to select stepsizes and regularizations based only on their local knowledge of the problem. Additionally, we present rules describing the precise effect of agents’ parameter selections on global convergence rate and error, which allows agents to select their parameters based on such tradeoffs. Furthermore, we differ by allowing unbounded delays in communications and computations.

A preliminary version of this work appeared in [26]. This version adds new 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 problem statements, and Section III proposes an update law. Section IV proves its convergence. Next, Section V derives a convergence rate, and Section VI quantifies the effect of regularizing on convergence. Section VII provides error bounds for set-constrained problems, while Section VIII provides error bounds for the unconstrained case. Section IX next illustrates these results in simulation. Finally, Section X concludes.

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 for agents to choose these parameters independently. We next give a formal problem statement 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 $i \in [N] := \{1, \ldots, N\}$. 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 assume the following.

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

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 $x := (x[1]^T, \ldots, x[N]^T)^T \in \mathbb{X} \subset \mathbb{R}^n$, 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 $r \in \mathbb{R}^n$. Here $Q$ is the Hessian matrix of $f$ and $r^T x$ is a linear cost term. We assume the following.

**Assumption 2:** $Q$ is symmetric and positive definite.

Note that symmetry 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. In addition, $\nabla f = Q x + r$, and $\nabla f$ is Lipschitz with constant $\|Q\|_2$, which is equal to the maximum singular value of $Q$.

In this paper, we divide $n \times n$ matrices into blocks. Given a matrix $B \in \mathbb{R}^{n \times n}$, where $n = \sum_{i \in [N]} n_i$, the $i$th block of $B$, denoted $B[i]$, is the $n_i \times n_i$ 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 $b$, $b[i]$ is the first $n_i$ entries of $b$, $b[2]$ is the next $n_2$ entries, etc. We further define the notation of a sub-block $B[j]$, where $B[i] = B[1]^T B[2] \cdots B[j]^T$. That is, $B[i]$ is the first $n_i$ columns of $B$, $B[i]_j$ is the next $n_j$ columns, etc. For notational simplicity, $B = [B[j]]_p$ 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,

$$B = [B[j]]_p = \begin{bmatrix} [B[1]^T] & [B[2]^T] & \cdots & [B[N]^T] \\
B[1] & B[2] & \cdots & B[N] \\
\vdots & \vdots & \ddots & \vdots \\
B[1]^T & B[2]^T & \cdots & B[N]^T 
\end{bmatrix}$$

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

Previous work has shown that totally asynchronous algorithms may diverge if $Q$ is not strictly diagonally dominant [14, Example 3.1, Chapter 6]. While enforcing that condition is sufficient to ensure a totally asynchronous update scheme will converge, in this paper we will instead require the weaker condition of strict block diagonal dominance.

**Definition 1:** Let the matrix $B = [B[j]]_p$, 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
\[
\left( \left\| B_i^{[i]} \right\|_2^{-1} \right)^{-1} > \sum_{j=1 \atop j \neq i}^N \left\| B_j^{[j]} \right\|_2 \quad \text{for all } i \in [N]
\]
then \( B \) is said to be \textit{strictly block diagonally dominant} relative to the choice of partitioning \( p \).

Note that just as in [14, Ex. 3.1, Ch. 6], we require a \textit{strict} notion of diagonal dominance, as opposed to \textit{weak} block diagonal dominance, in which case equality could hold in Equation (1). We elaborate in Section VIII-C about why strictness is essential to showing convergence. We also point out that this partitioning scheme is a generalization of previous work: setting each block size to 1 recovers ordinary strict diagonal dominance, as required in [14, Chapter 6].

In the forthcoming analysis, we will use the gap between the left and right sides of Equation (1), which we define as
\[
\delta(B^{[i]}) = \left( \left\| B_i^{[i]} \right\|_2^{-1} \right)^{-1} - \sum_{j=1 \atop j \neq i}^N \left\| B_j^{[j]} \right\|_2.
\]

Note that \( p = [1, 1, \ldots, 1]^T \) reduces Definition 1 to ordinary diagonal dominance. We now make the following assumption:

\textbf{Assumption 3:} In \( f, Q = [Q^{[i]}]_p \) 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 \( i \in [N] \).

\section*{B. Problem Statements}

To reduce parametric coupling between agents, we allow agents to select stepsizes independently. In particular, we wish for the stepsize for block \( i \), denoted \( \gamma_i \), to be chosen using only knowledge of \( Q^{[i]} \). The selection of \( \gamma_i \) should not depend on any other block \( Q^{[j]} \), \( j \neq i \), or any stepsize choice, \( \gamma_j \), for any other block. Allowing independent stepsizes will preclude the need for agents to agree on a single value before optimizing.

Additionally, although it is not necessary for convergence, we wish to allow for regularizations as well. Regularizations are commonly used for centralized quadratic programs to improve convergence properties, and we wish to allow that same capability here. However, in keeping with the independence of agents’ parameters, we wish to allow agents to choose independent regularizations: as with stepsizes, we wish for the regularization for block \( i \), denoted \( \alpha_i \), to be chosen using only knowledge of \( Q^{[i]} \). The following problem will be one focus of the remainder of the paper.

\textbf{Problem 1:} Design a distributed algorithm to solve
\[
\begin{align*}
\min_{x \in \mathbb{R}^N} & \quad \frac{1}{2} x^T Q x + r^T x \\
\text{subject to} & \quad x \in \mathbb{R}^N
\end{align*}
\]
where only agent \( i \) updates \( x^{[i]} \), and
1) Is totally asynchronous
2) Allows agents to choose stepsizes independently
3) Allows agents to choose regularizations independently.

Section III will specify the structure of the asynchronous communications and computations used to solve Problem 1. To maintain the flow of the paper and control the order in which concepts are introduced, we will satisfy conditions 1 and 2 of Problem 1 in Section IV. Afterwards, we will satisfy condition 3 in Section V.

\section*{III. Block-Based Multiagent 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 \( x_i \in \mathbb{R}^n \), and this is agent \( i \)’s local copy of the global state. Agent \( i \)’s own states in this vector are denoted by \( x_i^{[i]} \in \mathbb{R}^{n_i} \). The current values stored onboard agent \( i \) for agent \( j \)’s states are denoted by \( x_i^{[j]} \in \mathbb{R}^{n_j} \). In the forthcoming update law, agent \( i \) will only compute updates for \( x_i^{[i]} \), and it will share only \( x_j^{[i]} \) 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 \( x_i^{[i]} \).

An update law 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 [20], [27], [28], 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 \( \nabla^{[i]} f(x) = Q^{[i]} x + r^{[i]} \). We propose the update law
\[
x_i^{[i]}(k + 1) = \begin{cases} 
\Pi_{K^i} \left[ x_i^{[i]}(k) - \gamma_i \left( Q^{[i]} x_i^{[i]}(k) + r^{[i]} \right) \right], & k \in K^i \\
x_i^{[i]}(k), & k \notin K^i 
\end{cases}
\]
where \( \Pi_{K^i} \) is the Euclidean projection operator onto \( X_i \) and \( \gamma_i > 0 \) is a stepsize. 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 significantly 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 \( \tau_i^j \) is defined so that \( \tau_i^j(k_a) = k_a \). For \( K^i \) and \( \tau_i^j \), we assume the following:

\textbf{Assumption 4:} For all \( i \in [N] \), the set \( K^i \) is infinite. Moreover, for all \( i \in [N] \) and \( j \in [N] \setminus \{i\} \), if \( \{k_a\} \in \mathbb{N} \) is a sequence in \( K^i \) tending to infinity, then \( \lim_{d \to \infty} \tau_i^j(k_a) = d \).

Assumption 4 is simply a formalization of the requirement that no agent ever permanently stops updating and sharing its own state with any other agent. Specifically, \( K^i \) and \( \tau_i^j \) for all \( i, j \in [N] \) define a particular run of the algorithm by
specifying all time indices at which each agent either computes, communicates, or receives an update. We will show that any execution that satisfies Assumption 4 will converge, regardless of the other details of the communication protocol, such as whether it is stochastic or deterministic. In particular, we do not require a particular protocol to be used, and any protocol satisfying Assumption 4 is admissible. Our proposed update law for all agents can then be written as follows.

Algorithm 1: For all \(i \in [N] \) and \( j \in [N] \setminus \{i\} \), execute

\[
x_i^{[i]}(k + 1) = \begin{cases} \Pi X_i \left[ x_i^{[i]}(k) - \gamma_i \left( Q_i^{[i]} x_i(k) + r_i^{[i]} \right) \right] & k \in K_i^i \\
x_i^{[i]}(k) & k \notin K_i^i
\end{cases}
\]

\[
x_j^{[j]}(k + 1) = \left( x_j^{[j]}(k + 1) \right) i \text{ receives } j \text{'s state at } k + 1 \text{ otherwise}.
\]

In Algorithm 1 we see that \( x_j^{[j]} \) 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 sets that behave like Lyapunov sub-level sets. Then, in Section V, we will leverage this same construction to show that convergence is in fact geometric in the number of operations agents complete, in a sense to be made precise below.

A. Block-Maximum Norms

Convergence of Algorithm 1 will be shown using a block-maximum norm as in [14], [25], and [29]. Below, we define the block-maximum norm in terms of the partitioning vector \( p \).

**Definition 2:** Let \( x = [x^{[i]}]^p \in \mathbb{R}^n \), where \( p = [n_1, n_2, \ldots, n_N]^T \). The norm of \( x \) is defined as the maximum 2-norm of any block, i.e., \( \|x\|_{2,p} := \max_{i \in [N]} \|x^{[i]}\|_2 \).

The following lemma upper-bounds the induced block-maximum norm by the norms of the individual blocks.

**Lemma 1:** For the matrix \( B = [B_j^{[i]}]^p \) and induced matrix norm \( \| \cdot \|_{2,p} \), we have \( \|B\|_{2,p} \leq \max_{i \in [N]} \sum_{j=1}^N \|B_j^{[i]}\|_2 \).

**Proof:** Proof in Appendix A.

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 [14], and [29]. Below, we use \( \hat{x} := \arg \min_{x \in X} f(x) \) for the minimizer of \( f \). We state the following proposition concerning these sets, and below we construct a sequence of sets such that this proposition is true.

**Proposition 1:** 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 \( X(s) = \bigcap_{i=1}^N X_i(s) \times \cdots \times X_N(s) \)
4. \( \theta_i(y) \in X_i(s + 1) \) where \( \theta_i(y) := \Pi X_i[y^{[i]} - \gamma_i \nabla f(y)] \) for all \( y \in X(s) \), all \( s \in \mathbb{N} \), and all \( i \in [N] \).

Propositions 1.1 and 1.2 jointly imply that the collection \( \{X(s)\}_{s \in \mathbb{N}} \) is nested and that the sets converge to a singleton containing \( \hat{x} \). Proposition 1.3 allows for the blocks of \( x \) to be updated independently by the agents, which allows for decoupled update laws. Proposition 1.4 implies that state updates make only forward progress toward \( \hat{x} \), which ensures that each set is forward-invariant in time. It is shown in [29] 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 \( D_o := \max_{i \in [N]} \|x_i(0) - \hat{x}\|_{2,p} \), which is the block furthest from \( \hat{x} \) onboard any agent at timestep zero, and where we define the constant

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

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 \( \gamma_i \) 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 that Algorithm 1 converges if Proposition 1 is true.

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

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

The following lemma states an equivalent condition for Equation (2), which demonstrates the necessity and sufficiency of strict block diagonal dominance.

**Lemma 2:** Let \( Q = Q^F = [Q_j^{[i]}]^p \), where \( p = [n_1, n_2, \ldots, n_N]^T \). Additionally, let \( \{\gamma_1, \gamma_2, \ldots, \gamma_N\} \) be a set of \( N \) positive scalars. Then

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

if and only if

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

for all \( i \in [N] \).

**Proof:** Proof in Appendix B.

Note that \( \gamma_i \) only depends on \( Q_i^{[i]} \). This lemma implies that \( \gamma_i \) 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
\[
X(s) = \{ y \in X : \|y - \hat{x}\|_{2,p} \leq q^s D_o \} \tag{3}
\]
we next show that Proposition 1 is true for this construction of sets, thereby ensuring convergence of Algorithm 1.

**Theorem 1:** If Assumptions 1-4 hold and all entries of $\Gamma = \text{diag}(\gamma_1I_{n_1}, \gamma_2I_{n_2}, \ldots, \gamma_N I_{n_N})$ satisfy the conditions in Lemma 2, then Proposition 1 is true for the collection of sets $\{X(s)\}_{s \in \mathbb{N}}$ as defined in Equation (3).

**Proof:** See Appendix C.

Regarding Problem 1, we therefore state the following:

**Lemma 3:** Algorithm 1 satisfies conditions 1 and 2 of Problem 1 and asymptotically converges to $\hat{x}$.

**Proof:** See Appendix D.

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

Here we make a quick remark regarding our assumption that $Q$ is strictly block diagonally dominant. In the proof of Lemma 3, one can see that if $\|I - \Gamma Q\|_{2,p} < 1$, then our system would still converge, even if $Q$ is not necessarily diagonally dominant. However, the $2,p$-norm is difficult to compute, and must be done so numerically. There is no closed form of $Q$ that makes the above inequality true, and there is no simple or intuitive way to check if this is possible for a particular $Q$. However, if we upper bound this norm by $q$, we derive a simple compatibility check for $Q$ in the form of the block diagonal dominance condition and a closed form solution for $\Gamma$, both given in Lemma 2. This diagonal dominance condition can be interpreted as a restriction on how tightly coupled a QP can be while still allowing for total asynchrony.

**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 Proposition 1, means the convergence rate is geometric in the number of cycles completed, which we now show.

**Theorem 2:** Let Assumptions 1-5 hold and let $\gamma_i \in (0, 1 - \frac{2}{\sum_{j=1}^N \|Q_{ij}\|_2})$ for all $i$. 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:** See Appendix E.

From the definition of $q$, we write $q = \max_{i \in [N]} q_i$, where
\[
q_i = \left\| I - \gamma_i Q_i \right\|_2 + \gamma_i \sum_{j=1}^N \|Q_{ij}\|_2 \tag{4}
\]
which illustrates the dependence of each $q_i$ upon $\gamma_i$. 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 3:** $q$ is minimized when, for every $i \in [N]$,
\[
\gamma_i = \frac{2}{\lambda_{\max} (Q_i^{[i]}) + \lambda_{\min} (Q_i^{[i]})}.
\]

**Proof:** See Appendix F.

**VI. REGULARIZATION AND CONVERGENCE RATE**

In centralized optimization, regularization can be used to accelerate convergence by reducing the condition number of $Q$, denoted $k_Q$. 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 a common upper bound on regularizations to be agreed on, and enforces a common lower bound on agents’ regularizations, all of which lead to the type of parametric coupling that we wish to avoid.

As stated in Problem 1, we want to allow agents to choose regularization parameters independently. Here, we therefore only require that agent $i$ use a nonnegative regularization parameter $\alpha_i \geq 0$. In Algorithm 1, this changes only agent $i$’s updates to $x_i^{[i]}$, which now take the form
\[
x_i^{[i]} = \Pi_{\mathcal{X}_i} \left[ x_i^{[i]}(k) - \gamma_i \left( Q_i^{[i]} x_i(k) + r^{[i]}(k) + \alpha_i x_i^{[i]}(k) \right) \right].
\]
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. The regularized cost function is
\[
f_A(x) := \frac{1}{2} x^T (Q + A)x + r^T x
\]
where $Q + A$ is symmetric and positive definite because $Q = Q^T > 0$. We now state the following theorem that confirms that minimizing $f_A$ succeeds.

**Theorem 4:** Suppose that $A = \text{diag}(\alpha_1 I_{n_1}, \ldots, \alpha_N I_{n_N}) > 0$, where agent $i$ chooses $\alpha_i$ independently of all other agents. Then Algorithm 1 satisfies all conditions stated in Problem 1 when $f_A$ is minimized.
Proof: Replacing $Q$ with $Q + A$, all assumptions and conditions used to prove Lemma 3 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 Lemma 3 and are therefore omitted.

Problem 1 is now solved. Theorem 4 establishes that regularizing preserves asymptotic convergence, and we next analyze convergence rates. Because the condition number $k_Q$ 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 $q_i$ 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 Theorem 3. We then have

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

We now consider the case where this problem is regularized with some arbitrary $A$. Because $Q$ has been replaced with $Q + A$, our stepsize choice for each block $i \in [N]$ is now

$$\gamma_i = \frac{2}{\lambda_{\max}(Q_i^i) + \lambda_{\min}(Q_i^i) + 2\alpha_i}.$$  

When this is done, the convergence parameter becomes $q_{AI} = \max_i q_{\alpha_i}$, where

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

This shows that 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 the guaranteed convergence rate. Using this fact, we can tailor parameter selections to attain a desired convergence rate. Assume we have a desired convergence rate for our system, denoted $q^*$. If we want to set $q_{AI} \leq q^*$, we need $q_{\alpha_i} \leq q^*$ for all $i \in [N]$. Some algebraic manipulation of the above equation shows we therefore need

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

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 5**: Given $q^* \in (0, 1)$, for all $i$ agent $i$ uses

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

and $\gamma_i$ according to Equation (5), then $q_{AI} \leq q^*$.

Proof: Define $A$ as above, and substitute Equation (5) into Equation (4), replacing $Q$ with $Q + A$.  

### VII. REGULARIZATION ABSOLUTE ERROR BOUND: SET CONSTRAINED CASE

Regularization inherently results in a suboptimal solution because the system converges to $\hat{x}_A$ rather than $\hat{x}$. We therefore wish to bound this error by a function of the regularization matrix $A$. We define this error in two ways, $||\hat{x} - \hat{x}_A||_2, p = \max_i ||\hat{x}_i - \hat{x}_A||_2$, which is the largest error of any one block in the network, and $|f(\hat{x}) - f(\hat{x}_A)|$, which is the difference in cost for the system between the regularized and unconstrained 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. We present these two error bounds as Propositions, followed by their proofs.

**Proposition 2**: When $\gamma_i$ is chosen according to Equation (5) for all $i \in [N]$, then $||\hat{x} - \hat{x}_A||_2, p \leq E$, where

$$E = \frac{\max_{i \in [N]} \left(1 - \frac{q_{\alpha_i}}{q_i}\right) M_{X_i}}{1 - q_{AI}}$$  

where $M_{X_i} = \max_{x \in X_i} ||x||_2$.  

Proof: Because $\hat{x} = \Pi_X [\hat{x} - \Gamma(\hat{x} + r)]$ and $\hat{x}_A = \Pi_X [\hat{x}_A - \Gamma((Q + A)\hat{x} + r)]$ for appropriately chosen $\Gamma$, we can choose $\Gamma = \Gamma_1$, meaning every stepsize is chosen according to Equation (5). Doing so, we find

$$||\hat{x} - \hat{x}_A||_2, p \leq \max_{i \in [N]} \left(1 - \frac{q_{\alpha_i}}{q_i}\right) M_{X_i}$$  

where the first inequality follows from the non-expansive property of the Euclidean projection operator, and the last follows from the definitions of $q_{\alpha_i}$ and the $(2, p)$-norm. Through algebraic manipulation, when $\gamma_i$ is chosen according to Equation (5), we can see $\gamma_i \alpha_i = 1 - \frac{q_{\alpha_i}}{q_i}$. Using this and rearranging, we find

$$(1 - q_{AI}) ||\hat{x} - \hat{x}_A||_2, p \leq \max_{i \in [N]} \left(1 - \frac{q_{\alpha_i}}{q_i}\right) M_{X_i}$$  

where the second inequality follows from the non-expansive property of the Euclidean projection operator, and the last follows from the definition of $M_{X_i}$. Rearranging again completes the proof.

Note that from Assumption 1, the set constraint for each block is compact, meaning agent $i$ can explicitly calculate $M_{X_i}$. The significance of this error bound is that if a network operator has access to $q_{\alpha_i}$, $q_i$, and $M_{X_i}$ for all $i \in [N]$, which are locally computable by every agent, the network operator can compute these bounds. We now analyze the error in cost due to regularizing.

**Proposition 3**: When $\gamma_i$ is chosen according to Equation (5) for all $i \in [N]$, then

$$|f(\hat{x}) - f(\hat{x}_A)| \leq E \left(\sum_{i=1}^{N} M_{X_i} \max_{i \in [N]} \sum_{j=1}^{N} ||Q_j^i||_2^2 + \sum_{i=1}^{N} ||r^i||_2\right)$$  

where $E$ is defined as in Proposition 2.  

$\triangle$
\textbf{Proof:} Defining \(\Delta X_A = \hat{x} - \hat{x}_A\), we find that \(f(\hat{x}) - f(\hat{x}_A) = \frac{1}{2}(\hat{x} + \hat{x}_A)^T Q(\Delta X_A) + r^T (\Delta X_A)\), which gives
\[
|f(\hat{x}) - f(\hat{x}_A)| &= \left| \frac{1}{2}(\hat{x} + \hat{x}_A)^T Q(\Delta X_A) + r^T (\Delta X_A) \right| \\
&\leq \left\| \frac{1}{2}(\hat{x} + \hat{x}_A)^T Q + r^T \right\|_{2,p} \|\Delta X_A\|_{2,p} \\
&\leq \left\| \left( \frac{1}{2}(\hat{x} + \hat{x}_A)^T \left\| Q \right\|_{2,p} + \|r^T\|_{2,p} \right) \|\Delta X_A\|_{2,p}.
\]

By definition, \(\|x^p\|_{2,p} = \sum_{i=1}^N \|x_i^p\|_2\), and by Lemma 1 \(\|B\|_{2,p} \leq \max_i \sum_{j=1}^N \|B_{ij}\|_2\). Combining with the definition of \(M_X\), and Proposition 2 proves the proposition.

\textbf{Remark 1:} A major motivation for error bounds of this form is that they do not depend on global knowledge of the system, e.g., \(\|Q\|_2\). A network operator can construct these error bounds using only information that is locally computable by each agent. However, it is useful to compare these error bounds to those that a network operator with global knowledge may construct. First, note that \(|\Delta X_A| \leq \sqrt{N} |\Delta X_A|_{2,p}\), and therefore \(\|\Delta X_A\|_2 \leq \sqrt{N} E\). We can find an error bound using global information, which we call \(E_{global}\), by following equivalent steps in the proof of Proposition 2 using the 2-norm instead of the \((2, p)\)-norm. This gives us
\[
E_{global} = \sqrt{\sum_{i \in [N]} \left( 1 - \frac{q_{n,i}}{q_{T,i}} \right)^2 M_X^2}.
\]

Note that \(q_A \geq \|I - \Gamma_A (Q + A)\|_2\), which can be proven using Lemma 4. The gap in this inequality depends on the gap between the largest eigenvalue of \(I - \Gamma_A (Q + A)\) and the Gershgorin Circle Theorem bound provided in Section VIII-A. Additionally, we can rewrite the numerator of \(E\) as \(\sqrt{N} \|v\|_\infty\), and the numerator of \(E_{global}\) as \(\|v\|_2\), where \(v\) is an \(N\)-dimensional vector with \(i^{th}\) component \(\left( 1 - \frac{q_{n,i}}{q_{T,i}} \right) M_X \times \sqrt{N} \|v\|_\infty \geq \|v\|_2\). This implies \(E \geq E_{global}\). The gap between these bounds is highly problem-dependent, and the formulas for them can be used to calculate this gap on a case-by-case basis. Similarly, using equivalent steps to the proof of Proposition 3 yields \(|f(\hat{x}) - f(\hat{x}_A)| \leq (M_X \|Q\|_2 + \|r\|_2) E_{global}\), which is tighter than the bound given in Proposition 3.

\section{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 regularization. This bound is descriptive in the sense that given the agents’ regularization choices, one can derive a bound describing the 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 \(\alpha_i\) 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 \(Q_i\).

\textbf{Problem 2:} Given the restriction that \(\alpha_i\) can be chosen using only knowledge of \(Q_i\) and \(\epsilon\), where \(\epsilon \in (0, 1)\), develop independent regularization selection guidelines that guarantee
\[
\frac{|f(\hat{x}) - f(\hat{x}_A)|}{|f(\hat{x})|} \leq \epsilon.
\]

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 \(f(\hat{x}) \leq f(\hat{x}_A)\) is not. That is, the cost can be upper-bounded by zero for both the regularized and unregularized cases using \(x = 0\). Therefore the optimal cost in both cases is 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 have \(f(\hat{x}) - f(\hat{x}_A) \geq 0\). Then
\[
\frac{|f(\hat{x}) - f(\hat{x}_A)|}{|f(\hat{x})|} \leq \epsilon \text{ if and only if } \frac{f(\hat{x}) - f(\hat{x}_A)}{f(\hat{x})} \leq \epsilon.
\]

The solution to Problem 2 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 2 such that a certain eigenvalue condition of the matrix \(A^T Q A\) is satisfied. Afterward, it will be shown that this condition on \(A^T Q\) is sufficient to bound the error by \(\epsilon\).

\section{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 \(\lambda_k(B) \in \bigcup_{k=1}^n \{ b_{k,k} - \sum_{j \neq k} |b_{k,j}|, b_{k,k} + \sum_{j \neq k} |b_{k,j}| \}\) for all \(k = 1, \ldots, n\). 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 \(\lambda_{\min}(B) \geq \lambda_{\min}(b_{k,k} - \sum_{j \neq k} |b_{k,j}|)\). In the event that \(B\) is a strictly diagonally dominant matrix in the usual sense, i.e., \(n_k = 1\) for all \(i \in [N]\), with \(b_{k,k} \geq 0\) for all \(k = 1, \ldots, n\), this implies that every eigenvalue of \(B\) is positive, because \(\lambda_{\min}(B) \geq \lambda_{\min}(b_{k,k} - \sum_{j \neq k} |b_{k,j}|) > 0\) for all \(k = 1, \ldots, n\). Note also that if \(C\) is an \(m \times n\) positive definite diagonal matrix, then \(\lambda_{\min}(C B) \geq \lambda_{\min}(c_{k,k}(b_{k,k} - \sum_{j \neq k} |b_{k,j}|)) > 0\). That is, if \(B\) is strictly diagonally dominant and \(C\) is positive definite and diagonal, then \(C B\) 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 $c_{k,k}(b_{k,k} - \sum_{j \neq k} |b_{k,j}|) \geq l$ for all $k = 1, \ldots, n$. This condition can be restated as

$$\text{if } c_{k,k} \geq \frac{l}{b_{k,k} - \sum_{j \neq k} |b_{k,j}|} \text{ for all } k \in [n],$$

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

That is, given a strictly diagonally dominant matrix $B$ and a positive constant $l$, the $k^{th}$ diagonal element of $C$ can be chosen using only knowledge of the $k^{th}$ 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 $B = [B_{i,j}^{[i]}]_p$, where $p = [n_1, n_2, \ldots, n_N]^T$, each eigenvalue $\lambda(B)$ satisfies

$$\left(\left\| \left(B_{i,i}^{[i]} - \lambda(B)I\right)^{-1}\right\|_2\right)^{-1} \leq \sum_{j \neq i} \left\| B_{j,i}^{[i]} \right\|_2$$

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

**Proof:** See Theorem 2 in [30].

For $\lambda_{\min}(B)$, the lower bound in Lemma 4 becomes

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

Let $\mu(B_{i,i}^{[i]}) = \arg \min_{\lambda_i} \left| \lambda_{\min}(B) - \lambda_i \left(B_{i,i}^{[i]}\right) \right|$, the eigenvalue of $B_{i,i}^{[i]}$ closest to the minimum eigenvalue of $B$. Then,

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

From the block Gershgorin Circle Theorem, we then have

$$\lambda_{\min}(B) \geq \mu(B_{i,i}^{[i]}) - \sum_{j \neq i} \left\| B_{j,i}^{[i]} \right\|_2$$

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

Because $\mu(B_{i,i}^{[i]}) \geq \lambda_{\min}(B_{i,i}^{[i]})$, we can say $\lambda_{\min}(B) \geq \delta(B_{i,i}^{[i]})$ 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 $C = [C_{i,i}^{[i]}]_p$, with $C_{i,i}^{[i]} = c_i I$ for every $i \in [N]$ and $C_{j,i}^{[i]} = 0$ when $j \neq i$. In the same manner as above, we find

$$c_i \geq \frac{l}{\delta(B_{i,i}^{[i]})} \text{ for all } i \in [N]$$

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

That is, $c_i$ can be chosen using only knowledge of $B_{i,i}^{[i]}$ and $l$. This brings us back to the restrictions imposed in Problem 2. For reasons that will be shown in the following subsection, choose $B = Q$, $C = A^{-1}$, and $l = \frac{1 - \sqrt{\epsilon}}{\epsilon}$.

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 $A = [A_{j,j}^{[i]}]_p$, with $A_{j,j}^{[i]} = \alpha_i I$ for every $i \in [N]$ and $A_{j,j}^{[i]} = 0$ when $j \neq i$. If we have $\alpha_i \leq \frac{\sqrt{\epsilon}}{1 - \sqrt{\epsilon}} \delta(Q^{[i]})$ for all $i \in [N]$ then $\lambda_{\min}(A^{-1}Q) \geq \frac{1 - \sqrt{\epsilon}}{\epsilon}$.

**Proof:** Use (6) with $C = A^{-1}$, $B = Q$, and $l = \frac{1 - \sqrt{\epsilon}}{\epsilon}$.

We have shown this eigenvalue condition can be satisfied according to the conditions in Problem 2, i.e., $A^{[i]}$ is chosen using only knowledge of $Q^{[i]}$ and $\epsilon$. The next subsection shows this is sufficient to satisfy the error bound in Problem 2.

### B. Error Bound Satisfaction

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

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

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

then $\left| f(\hat{x}) - f(\hat{x}_A) \right| \leq \epsilon \left| f(\hat{x}) \right|$. 

**Proof:** Proof in Appendix G.

With these lemmas, we now present the following theorem.

**Theorem 6:** 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 $A = [A_{j,j}^{[i]}]_p$, with $A_{j,j}^{[i]} = \alpha_i I$ for every $i \in [N]$ and $A_{j,j}^{[i]} = 0$ when $j \neq i$. Let $f(x) = \frac{1}{2} x^T Qx + r^T x$ where $x \in \mathbb{R}^n$. Let $\hat{x} = \arg \min_{x \in \mathbb{R}^n} f(x)$ and $\hat{x}_A = \arg \min_{x \in \mathbb{R}^n} f(x) + \frac{1}{2} x^T Ax$ where $A \succ 0$ and diagonal. Additionally, let $\epsilon \in [0, 1]$. If

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

then $\left| f(\hat{x}) - f(\hat{x}_A) \right| \leq \epsilon \left| f(\hat{x}) \right|$. 

**Proof:** Lemma 5 shows that the regularization selection rules presented above, along with Assumption 3, imply that $\frac{1 - \sqrt{\epsilon}}{\epsilon} \leq \lambda_{\min}(A^{-1}Q)$. Lemma 6 shows that $\frac{1 - \sqrt{\epsilon}}{\epsilon} \leq \lambda_{\min}(A^{-1}Q)$ implies that $\left| f(\hat{x}) - f(\hat{x}_A) \right| \leq \epsilon \left| f(\hat{x}) \right|$. Additionally, we can derive a similar bound for relative error in the solution itself. Because in the unconstrained case $\hat{x} = -Q^{-1}r$ and $\hat{x}_A = -(Q + A)^{-1}r$, we see $\left\| \hat{x} - \hat{x}_A \right\|_2 \leq \left\| (Q^{-1} - (Q + A)^{-1})r \right\|_2$. 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

$$\left\| \hat{x} - \hat{x}_A \right\|_2 \leq \left\| (I + A^{-1}Q)^{-1} \right\|_2 \left\| Q^{-1} \right\|_2 \left\| r \right\|_2. $$

Additionally, we make use of the following lemma:

**Lemma 7:** If there is a block strictly diagonally dominant matrix $B = [B_{i,j}^{[i]}]_p$, where $p = [n_1, n_2, \ldots, n_N]^T$, and $\beta_p(B) = \min_i \left( \left\| B_{i,i}^{[i]} \right\|_2^{-1} - \sum_{j \neq i} \left\| B_{j,i}^{[i]} \right\|_2 \right)$, then

$$\left\| B^{-1} \right\|_2 \leq \beta_p^{-1}(B).$$

**Proof:** Theorem 2 in [31] establishes the above result for $\left\| \cdot \right\|_\infty$, and the proof for $\left\| \cdot \right\|_2$ follows identical steps.
Defining this error as $\| \hat{x} - \hat{x}_A \|_{2,p}$ and using Equation (7) and Lemma 7 we see

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

If we wish for agents to select regularizations such that the above error is less than some $\eta > 0$, this is accomplished if

$$\frac{1}{\eta} \leq \min_{i \in [N]} \left[ 1 + \alpha_i^{-1} \delta(Q^{(i)}) \right]$$

which is true if and only if

$$\alpha_i \leq \eta \frac{1}{1 - \eta} \delta(Q^{(i)})$$

for all $i \in [N]$.

This rule has the same structure as the one in Theorem 6, with the only difference being there is no square root taken of $\eta$.

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. if $q_i < q^*$, that agent can choose $\alpha_i = 0$. This is because all of the above analysis holds if $\alpha_i$ is chosen to be a small positive value, which can be set arbitrarily small.

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 5 provides a lower bound on $\alpha_i$ that allows agents to converge at any speed we wish, while Theorem 6 provides an upper bound on $\alpha_i$ that allows us to bound the cost error between the solution we find and the optimal solution. Practically, this means the network operator has two options. The first is to choose a desired convergence rate, and then agents may select regularizations according to Theorem 6 to achieve that convergence rate while inducing the minimum amount of error. The second option is to choose a maximum tolerable error, and allow agents to select regularizations according to Theorem 7 to achieve the fastest convergence rate while obeying that error bound. The following inequalities demonstrate how a network operator’s choice of either $\epsilon$ or $q^*$ affects the possible values of the other:

$$q^* \geq \max_{i \in [N]} \left[ \frac{(1 - \sqrt{\epsilon})q_i}{(1 - \sqrt{\epsilon})q_i + \frac{2\delta(Q^{(i)})}{\lambda_{\min}(Q^{(i)}) + \lambda_{\max}(Q^{(i)})}} \right]$$

$$\sqrt{\epsilon} \geq \max_{i \in [N]} \left[ \frac{(q_i - q^*)}{(q_i - q^*) + \frac{2\delta(Q^{(i)})}{\lambda_{\min}(Q^{(i)}) + \lambda_{\max}(Q^{(i)})}} \right]. \quad (8)$$

We also observe why it is critical that $Q$ be strictly block diagonally dominant. If block $i$ of $Q$ is weakly diagonally dominant, then $q_i = 1$, and convergence cannot be guaranteed in a totally asynchronous setting. It may be tempting to simply add a regularization parameter $\alpha_i$ to block $i$ to make it strictly block diagonally dominant, but the above inequalities demonstrated why this cannot be done. If block $i$ of $Q$ is weakly diagonally dominant, then $\delta(Q^{(i)}) = 0$, and both inequalities collapse to 1. This implies that any attempt to reduce the contraction term below 1 will result in an error bound of 100%, effectively meaning that no guarantees can be made about the quality of the solution. While there are other methods to bound the error without requiring strict diagonal dominance, they are inherently centralized, and do not allow agents to select parameters independently.

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 $q_{\text{initial}} = 0.99, 0.95, 0.85, 0.70, 0.50, 0.30,$ and 0.01. For each QP, $q^*$ is ranged from $q$ to 0, and the percentage reduction from $q$ to $q^*$ is plotted against the corresponding error bound given by Equation (8). For example, the data for the QP with $q_{\text{initial}} = 0.85$ is plotted by the yellow dotted line in Fig. 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 Fig. 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 $\hat{x}_A \to 0$ as the diagonal entries of $A$ tend toward $\infty$, $f(\hat{x}_A) \to 0$ as $\hat{x}_A \to 0$, and $\epsilon \to 1$ as $f(\hat{x}_A) \to 0$. The second is that the larger $q_{\text{initial}}$ is, the more sensitive the error bound for the QP is to regularizing. That is, if $q_{\text{initial}}$ is thought of as a condition number, then “poorly conditioned” QPs will have larger errors due to regularizing.
as is visible in the sub-window in Fig. 2. However, Lemma 3 guarantees these growths are bounded and that each case will converge to its unique solution, which is either \( \hat{x} \) or \( \hat{x}_A \).

\section{Conclusions}

We have developed a distributed quadratic programming framework that converges under totally asynchronous conditions while letting each agent independently select its stepsize and regularization parameters. Future work will apply these developments to quadratic programs with functional constraints.

\appendix

\section{Proof of Lemma 1:}
By definition of the maximum norm,
\[
\| B \|_{2,p} = \sup_{\| x \|_{2,p} = 1} \| Bx \|_2 = \sup_{\| x \|_{2,p} = 1} \| x \|_2 = 1 \frac{\| B \|_p}{\| B \|_S}.
\]

Since \( B[i] = [B_1[i]^T B_2[i]^T \ldots B_N[i]^T] \), we can now write
\[
B[i]x = B_1[i]x[1] + B_2[i]x[2] + \ldots + B_N[i]x[N].
\]

Therefore,
\[
\| B \|_{2,p} = \sup_{\| x \|_{2,p} = 1} \| B[i]x[1] + \ldots + B_N[i]x[N] \|_2.
\]

By the triangle inequality, we have
\[
\| B \|_{2,p} \leq \sup_{\| x \|_{2,p} = 1} \max_{1 \leq i \leq N} \| B_j[i]x[j] \|_2.
\]

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

\section{Proof of Lemma 2:}
Because \( Q_i[i] = Q_i[i]^T \neq 0 \), we see that
\[
\| I - \gamma_i Q_i[i] \|_2 = \max \left\{ |\lambda_{min} \left( I - \gamma_i Q_i[i] \right)|, |\lambda_{max} \left( I - \gamma_i Q_i[i] \right)| \right\}
\]
\[
\quad = \max \left\{ |\lambda_{min} (Q_i[i])|, |\gamma_i \lambda_{max} (Q_i[i])| \right\},
\]
which allows us to write
\[
\| I - \gamma_i Q_i[i] \|_2 + \gamma_i \| \sum_{j \neq i} Q_j[i] \|_2 < 1
\]
if and only if \( |1 - \gamma_i \lambda(Q_i[i])| < 1 - \gamma_i \sum_{j \neq i} \| Q_j[i] \|_2 \) is true for both \( \lambda = \lambda_{min} \) and \( \lambda = \lambda_{max} \). These two inequalities are true if and only if both

\[
\lambda_{min}(Q_i[i]) > \frac{\sum_{j=1}^{N} Q_j[i]}{\lambda_{max}(Q_i[i]) + \sum_{j=1}^{N} Q_j[i]} \quad \text{and} \quad \gamma_i < \frac{2}{\lambda_{max}(Q_i[i]) + \sum_{j=1}^{N} Q_j[i]}. \]

Therefore, we can write \( \| I - \gamma_i Q_i[i] \|_2 + \gamma_i \| \sum_{j \neq i} Q_j[i] \|_2 < 1 \) if and only if the previous two inequalities hold.
APPENDIX C

Proof of Theorem 1: For Proposition 1.1, by definition
\[ X(s+1) = \{ y \in X : \| y - \hat{x} \|_2 \leq q^{s+1} D_o \}. \]
Since \( q \in (0, 1) \), we have \( q^{s+1} < q^s \), which results in \( \| y - \hat{x} \|_2 \leq q^{s+1} D_o < q^s D_o \). Then \( y \in X(s) \) implies \( y \in X(s) \) and \( X(s+1) \subset X(s) \subset X \), as desired.

For Proposition 1.2 we find
\[ \lim_{s \to \infty} X(s) = \lim_{s \to \infty} \{ y \in X : \| y - \hat{x} \|_2 \leq q^s D_o \} = \{ \hat{x} \}. \]
The structure of the weighted block-maximum norm then allows us to see that \( \| y - \hat{x} \|_2 \leq q^s D_o \), and only if \( \| y^{[i]} - \hat{x}^{[i]} \|_2 \leq q^s D_o \) for all \( i \in [N] \). It then follows that
\[ X_i(s) = \{ y^{[i]} \in X_i : \| y^{[i]} - \hat{x}^{[i]} \|_2 \leq q^s D_o \} \]
which gives \( X(s) = X_1(s) \times \ldots \times X_N(s) \), thus satisfying Proposition 1.3.

We then see that, for \( y \in X(s) \),
\[ \| \theta_i(y) - \hat{\theta}^{[i]} \|_2 \leq \| y^{[i]} - \hat{x}^{[i]} - \gamma_i Q^{[i]}(y - \hat{x}) \|_2 \]
which follows from the definition of \( \theta_i(y) \) and the fact that \( \hat{x}^{[i]} = X_i[\theta_i(\hat{x})] \). Using the non-expansive property of the projection operator \( X_i[\cdot] \), we find
\[ \| \theta_i(y) - \hat{x}^{[i]} \|_2 \leq \| y^{[i]} - \hat{x}^{[i]} - \gamma_i Q^{[i]}(y - \hat{x}) \|_2 \]
\[ \leq \max_{i \in [N]} \| (I^i - \gamma_i Q^{[i]})(y - \hat{x}) \|_2 \]
\[ = \| (I - \Gamma_Q)(y - \hat{x}) \|_2 \]
\[ \leq \| I - \Gamma_Q \|_{2,p} \| y - \hat{x} \|_2 \]
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
\[ \| \theta_i(y) - \hat{x}^{[i]} \|_2 \leq q \| y - \hat{x} \|_2 \leq q^{s+1} D_o, \]
and \( \theta_i(y) \in X_i(s+1) \). Then Proposition 1.4 is satisfied..

APPENDIX D

Proof of Theorem 3: Theorem 1 shows Proposition 1 is true for the construction of the sets \( \{ X(s) \}_{s \in \mathbb{N}} \), and from [14] and [29] we see this implies asymptotic convergence of Algorithm 1 for all \( i \in [N] \). Condition 1 of Problem 1 is satisfied by not requiring delay bounds, and condition 2 is satisfied because the bound on \( \gamma_i \) depends only upon \( Q^{[i]} \).

APPENDIX E

Proof of Theorem 2: 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 3: If \( \gamma_i \leq \frac{2}{\lambda_{\max}(Q^{[i]}i) + \lambda_{\min}(Q^{[i]}i)} \), then
\[ q_i = 1 - \gamma_i \left( \lambda_{\min} \left( Q^{[i]}_i \right) - \sum_{j=1}^{N} \left\| Q^{[i]}_j \right\|_2 \right), \]
and if \( \gamma_i \geq \frac{2}{\lambda_{\max}(Q^{[i]}i) + \lambda_{\min}(Q^{[i]}i)} \), then
\[ q_i = -1 + \gamma_i \left( \lambda_{\max} \left( Q^{[i]}_i \right) + \sum_{j=1}^{N} \left\| Q_j^{[i]} \right\|_2 \right). \]
That is, when \( \gamma_i \leq \frac{2}{\lambda_{\max}(Q^{[i]}i) + \lambda_{\min}(Q^{[i]}i)} \), the relationship between \( q_i \) and \( \gamma_i \) is a line with negative slope, and when \( \gamma_i \geq \frac{2}{\lambda_{\max}(Q^{[i]}i) + \lambda_{\min}(Q^{[i]}i)} \), the relationship is a line with positive slope. Then \( q_i \) is minimized at the point where the slope changes sign, which occurs when \( \gamma_i = \frac{2}{\lambda_{\max}(Q^{[i]}i) + \lambda_{\min}(Q^{[i]}i)} \).

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 \( B = B^T \leq 0 \) and \( C \) is an invertible matrix of the same dimension, then \( \lambda_i(C^{-1}BC) \leq 0 \) 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 [32, 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 $- \epsilon \leq - \lambda_{\text{max}}((P^{-1} A)^2)$. From Fact 3,

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

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

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

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

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

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

Note that $P^{-1} + P^{-1} A^{-1}P^{-1} Q = P^{-1} + P^{-1}(P - Q)P^{-1} = 2P^{-1} - P^{-1} Q P^{-1}$, therefore

$$1 - \epsilon \leq \lambda_{\text{min}}((2P^{-1} - P^{-1} Q P^{-1})),$$

which implies

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

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

From Fact 3, $0 \leq -\lambda_{\text{max}}((1 - \epsilon)I - (2P^{-1} - P^{-1} Q P^{-1}))Q$ and \( \lambda_{\text{max}}((1 - \epsilon)I - (2P^{-1} - P^{-1} Q P^{-1}))Q \leq 0 \).

From Fact 4, taking \( C = Q^{-\frac{1}{2}} \)

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

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

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

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

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

Assuming \( x \neq 0 \), \( x^TQ^{-1}x \) is a positive scalar. Dividing both sides by this term gives

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

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

$$-\frac{1}{2}r^TQ^{-1}r - (\frac{1}{2}r^TQ^{-1}P^2r + r^TP^{-1}r) \leq \epsilon,$$

and substituting returns the desired result.

\[ \]