EURO Journal on Computational Optimization 12 (2024) 100098

Contents lists available at ScienceDirect

EURO Journal on Computational Optimization

journal homepage: www.elsevier.com/locate/ejco

# Communication-efficient ADMM using quantization-aware Gaussian process regression ☆

Aldo Duarte [a,*], Truong X. Nghiem [b], Shuangqing Wei [a]

[a] Division of Electrical and Computer Engineering, School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, LA 70803, United States

[b] Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816, United States

A B S T R A C T

In networks consisting of agents communicating with a central coordinator and working together to solve a global optimization problem in a distributed manner, the agents are often required to solve private proximal minimization subproblems. Such a setting often requires a decomposition method to solve the global distributed problem, resulting in extensive communication overhead. In networks where communication is expensive, it is crucial to reduce the communication overhead of the distributed optimization scheme. Gaussian processes (GPs) are effective at learning the agents' local proximal operators, thereby reducing the communication between the agents and the coordinator. We propose combining this learning method with adaptive uniform quantization for a hybrid approach that can achieve further communication reduction. In our approach, due to data quantization, the GP algorithm is modified to account for the introduced quantization noise statistics. We further improve our approach by introducing an orthogonalization process to the quantizer's input to address the inherent correlation of the input components. We also use dithering to ensure uncorrelation between the quantizer's introduced noise and its input. We propose multiple measures to quantify the trade-off between the communication cost reduction and the optimization solution's accuracy/optimality. Under such metrics, our proposed algorithms can achieve significant communication reduction for distributed optimization with acceptable accuracy, even at low quantization resolutions. This result is demonstrated by simulations of a distributed sharing problem with quadratic cost functions for the agents.

## 1. Introduction

Networked systems have emerged due to the rapid development of communication systems and sensing technologies. Such networks consist of multiple (possibly mobile) agents that cooperate to reach a global objective. Many of those networks can obtain its global objective by convex distributed optimization. In the framework of distributed optimization, some applications for network systems include power systems, sensor networks, smart buildings, and smart manufacturing [1].

Many algorithms are suited to solve distributed convex optimization; see e.g., [2], [3], [4], [5]. Among them, a simple yet powerful algorithm is the Alternating Direction Method of Multipliers (ADMM), first presented in [6]. This algorithm solves an optimization problem by decomposing it into smaller local sub-problems. Then, each agent solves its local sub-problem and sends its results to a coordinator, which combines all the agents' solutions to assemble the global objective. Two major advantages of the ADMM are that it is relatively easy to implement and, because of its decomposing behavior, it is simple to parallelize. As described in [7], the ADMM has broad applications in statistical and machine learning problems including the Lasso, sparse logistic regression, basis pursuit, support vector machines, and many others.

To solve a distributed optimization in a star topology networked system using ADMM, a *query-response* scheme is often employed. In such a scheme, the local sub-problems are cast as *proximal minimization problems* [2], which are regularized versions of the original

☆ This material is based upon work supported by the National Science Foundation under Awards No. 2238296, 2331710, and 2331711.

* Corresponding author.

*E-mail addresses:* aduart3@lsu.edu (A. Duarte), truong.nghiem@ucf.edu (T.X. Nghiem), swei@lsu.edu (S. Wei).

https://doi.org/10.1016/j.ejco.2024.100098

Available online 13 September 2024

sub-problems, to be solved by the agents in response to queries made by the coordinator. Proximal minimization keeps an agent's local function from being revealed to the coordinator, which is ideal for networks with privacy constraints. The queries are calculated and transmitted by the coordinator in each iteration upon receiving the previous agents' responses.

A major drawback of this distributed optimization scheme is that it often incurs extensive communication between the coordinator and agents, increasing communication overhead and communication costs, potentially making the network non-viable if communication is costly. It is therefore critical to reduce the communication load in these query-response distributed optimization schemes. The communication load can be reduced not only by limiting the number of communication rounds directly but by considering the communication overhead, namely the payload size in each iteration of a distributed optimization algorithm. Payload size can be reduced by quantizing the data exchanged between the agents and coordinator.

Our previous work [8] proposed to solve a distributed optimization problem using ADMM where the proximal operators were predicted by Gaussian process (GP) regression, and the communications coming from the agents to the coordinator were quantized. However, it had two limitations: 1) it did not account for the quantization of the training data in the optimization of the GP hyperparameters and in the GP regression; and 2) it did not consider the correlation between quantization noise and inputs, nor mitigation of these correlation issues. Because GP regression assumes a joint Gaussian distribution between any evaluations of the underlying latent function, but the quantization noise is not Gaussian and even correlated with the original function values, the regression modeling had to be adjusted accordingly. The use of inferred values from an incorrectly modeled learning method affects the accuracy of the ADMM algorithm, which may cause an increase in the number of iterations to reach convergence or potential failure to reach convergence.

In this paper, we propose to address these limitations by integrating two components: an adaptive uniform quantizer with joint dithering and orthogonalization, and an improved regression method that takes into consideration the quantization error in the learning data.

Our **main contributions** are summarized below.

1. We study the statistics of the quantization error of the adaptive uniform quantizer proposed in our previous work [8], and characterize its impact on the distributed optimization algorithm.
2. We employ a novel Linear Minimum Mean Square-error Estimator (LMMSE) based regression which takes in consideration the impact of the quantization error to improve the hybrid communication reduction approach from [8]. We also develop an additional LMMSE to more accurately approximate the real response of an agent from its quantized value, to further mitigate the impact of quantization in the ADMM algorithm.
3. We integrate our adaptive uniform quantizer with orthogonal transformations and dithering to account for the inherent correlation of the elements conforming the quantizer's input and to ensure the un-correlation between the quantization error and the quantizer's input, respectively.
4. We validate our approach by extensive simulations of a distributed network solving a sharing problem with a quadratic cost function. For comparison purposes, we also test two baseline methods using the proposed distributed network: vanilla ADMM and ADMM with GP. The simulation results show significant reductions in the total communication cost in all test cases compared to baseline methods, with negligible compromise in optimization performance.

*Paper Organization*: Related works are reviewed in Section 2, followed by the problem formulation in Section 3. An overview of uniform quantization and GP regression is presented in Section 4. Then, Section 5 presents the main mathematical foundation and derivations relevant to our work. A detailed presentation of our proposed approach is shown in Section 6. The simulation results are presented in Section 7. Section 8 discusses the convergence behavior of our proposed approach. Finally, we conclude the paper with the main contributions in Section 9.

## 2. Related works

ADMM has been widely applied for solving distributed optimization problems ([7], [9]), such as consensus problems [10] and sharing problems [11]. Communication reduction in distributed optimization settings has been previously studied. By solving each subsystem via ADMM and using the k-means algorithm to partition a distributed smart grid, the authors of [12] were able to reduce communication complexity. The concept of *the Moreau envelope function* is used in [13] and further developed in [14] to predict the proximal operators of the local agents so that certain communication rounds can be skipped. The same concept was used in [15], where the local proximal operators and their gradients were predicted by GP.

Several works proposed quantization methods to reduce the data exchange size in each algorithmic iteration, resulting in less overall communication overhead. The work in [16] presented a quantized distributed composite optimization problem over relay-assisted networks solved via a simplified augmented Lagrangian method. In [17], a distributed optimization problem affected by quantization was solved using the inexact proximal gradient method. In [18], a distributed optimization problem was solved by a distributed gradient algorithm with adaptive quantization.

Related to GP regression with quantized data is GP regression where part of the data was censored, which has been previously studied. The authors of [19] described a GP framework where all data that was outside of a specific range was fixed to a value. Also, in [20] a system identification with quantized output data modeled with GP was presented, where Gibbs sampler was used for kernel hyperparameters estimation. Finally, in [21] the best locations for sensors in a spatial environment are predicted by GP.

Our work is fundamentally different from the above works because it combines the concepts of ADMM, online learning, and quantization that in previous works were studied separately. Furthermore, our work fully integrates the three concepts by accounting for the quantization error and prediction error to build an approach that correctly models and mitigates the impact of both sources of error.

## 3. Problem formulation

This work deals with a multi-agent optimization problem whose structure takes the form of the sharing problem as considered in [7,11]:

$$\text{minimize} \quad \sum_{i=1}^{n} f_i\left(x_i\right) + h\left(\sum_{i=1}^{n} x_i\right). \tag{1}$$

Here, $n$ agents, each with local decision variables $x_i \in \mathbb{R}^p$, equipped with a proper and convex local cost function $f_i$: $\mathbb{R}^p \mapsto \mathbb{R}$, coordinate to minimize the system cost consisting of all local costs and a proper and convex shared global cost function $h$: $\mathbb{R}^p \mapsto \mathbb{R}$. Each cost function is only known to its corresponding agent and cannot be shared with the coordinator or other agents for privacy reasons. The problem presented in (1) can be solved with the ADMM. By introducing copies $y_i$ of $x_i$, the problem can be formulated equivalently as

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n} f_i\left(x_i\right) + h\left(\sum_{i=1}^{n} y_i\right) \\ &\text{subject to} && x_i - y_i = 0, \quad \forall i = 1, \dots, n. \end{aligned} \tag{2}$$

Because the agents keep their local cost function $f_i$ private, each agent $i$ will only provide the solution to the following local *proximal minimization problem* to the coordinator

$$\mathbf{prox}_{\frac{1}{\rho} f_i}(z_i^k) = \underset{x_i \in \mathbb{R}^p}{\arg\min} \left\{ f_i(x_i) + \frac{\rho}{2} \|x_i - z_i^k\|^2 \right\}, \tag{3}$$

in response to a value (a query) $z_i^k$ sent to it by the coordinator at iteration $k$, where $\rho > 0$ is a penalty parameter. The ADMM works in a query-response manner as follows. At iteration $k$, a query point $z_i^k$ is generated by the coordinator and sent to an agent $i$. Each agent solves its proximal minimization problem at its query point $z_i^k$ and replies with the response vector $\mathbf{prox}_{\frac{1}{\rho} f_i}(z_i^k)$ to the coordinator. The coordinator then updates the dual variables and generates the query points at the next iteration. Mathematically, each ADMM iteration $k$ involves the following updates derived in the analysis in Chapter 7 in [7]:

1. The coordinator updates the average of $y_i$

   $$\bar{y}^{k+1} = \underset{\bar{y} \in \mathbb{R}^p}{\arg\min} \left\{ h(n\bar{y}) + (n\rho/2) \|\bar{y} - \bar{x}^k - u^k\|^2 \right\}$$

   then sends a query $z_i^k = x_i^k - \bar{x}^k + \bar{y}^{k+1} - u^k$ to each agent $i$.
2. Each agent $i$ updates and sends its response $x_i^{k+1} = \mathbf{prox}_{\frac{1}{\rho} f_i}\left(z_i^k\right)$ to the coordinator.
3. The coordinator calculates the average $\bar{x}^{k+1} = (1/n) \sum_{i=1}^{n} x_i^{k+1}$ and updates the scaled dual vector $u^{k+1} = u^k + \bar{x}^{k+1} - \bar{y}^{k+1}$.

This process is repeated until convergence is achieved or until a maximum number of iterations is reached.

### 3.1. Moreau envelope

To reduce the communication overhead in this distributed optimization scheme, the authors of [14] proposed an approach called STEP (STructural Estimation of Proximal operator) which relies on the concept of the Moreau envelope of a function $f$. For brevity, we drop the subscript $i$ and the superscript $k$ in the subsequent equations. For $1/\rho > 0$, the Moreau envelope $f^{\frac{1}{\rho}}$ of $f$ is defined as

$$f^{\frac{1}{\rho}}(z) = \min_{x \in \mathbb{R}^p} \left\{ f(x) + \frac{\rho}{2} \|x - z\|^2 \right\}. \tag{4}$$

When $f$ is a proper and convex function, the Moreau envelope $f^{\frac{1}{\rho}}$ is *convex and differentiable with Lipschitz continuous gradient with constant* $\rho$ [Fact 2.2 in [22]]. Moreover, the unique solution to the proximal minimization $\mathbf{prox}_{\frac{1}{\rho} f}(z)$ is [23, Proposition 5.1.7]

$$\mathbf{prox}_{\frac{1}{\rho} f}(z) = z - \frac{1}{\rho} \nabla f^{\frac{1}{\rho}}(z). \tag{5}$$

Consequently, the gradient $\nabla f^{\frac{1}{\rho}}(z)$ is all that is required to reconstruct the optimizer of (3) following from (5).

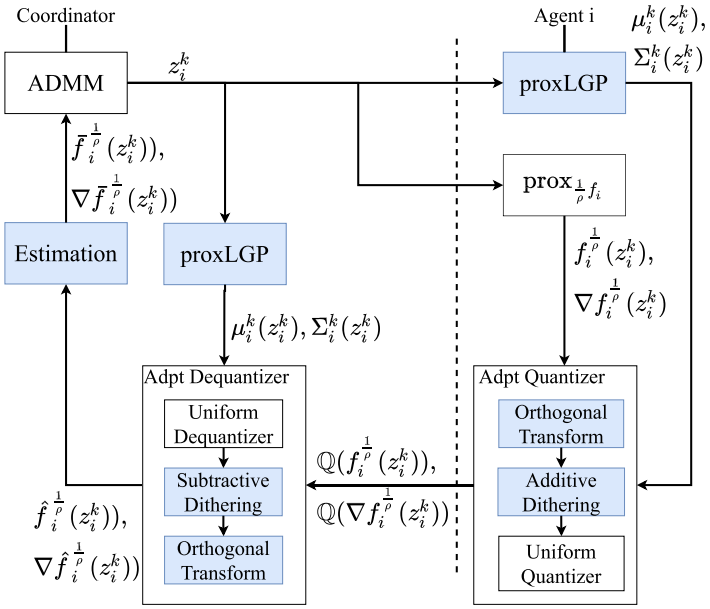


**Fig. 1.** Flow diagram of a query and response between the coordinator and an agent in the proposed approach. The enhancements contributed by this work, compared with the original approach in [8], are highlighted in the blue-shaded boxes.

The STEP approach estimates the unknown gradient $\nabla f^{\frac{1}{\rho}}(z)$ at any query point $z$ by constructing a set of possible gradients at $z$ based on past queries and then selecting a gradient that is "most likely" the true gradient. The work presented in [15] improved STEP by learning the Moreau envelopes corresponding to the local proximal operators with GP, which are updated online from past query data and used to predict the gradient $\nabla f^{\frac{1}{\rho}}(z)$ for estimating the proximal operators (3) of the agents by (5).

### 3.2. Proposed solution overview

The communication expenditure can be reduced further if the learning component is combined with the quantization of the communications between agents and coordinator. Our work [8] presented some preliminary results on a hybrid approach combining learning with quantization for further reducing communication overhead. This paper builds upon our hybrid approach [8] by further analyzing and mitigating the impact of quantization errors. Our improved hybrid approach is depicted in the diagram in Fig. 1, which describes the communication and computation processes between the coordinator and an agent $i$ at ADMM iteration $k$. In the colored boxes are new or modified components developed in this work compared to the approach in [8]. The blocks colored blue indicate the processes that were added or improved compared with our work in [8].

In Fig. 1, if the coordinator determines that a communication with agent $i$ is necessary at iteration $k$, it will send the query point $z_i^k$ to the agent. The Moreau envelope $f_i^{1/\rho}(z_i^k)$ and its gradient $\nabla f_i^{1/\rho}(z_i^k)$ are then calculated. A regression is performed simultaneously by the agent's proxLGP (identical to the coordinator's proxLGP), to obtain the predictive mean $\mu_i^k(z_i^k)$ and the covariance matrix $\Sigma_i^k(z_i^k)$ of the agent's response. These values are used to parameterize the quantization process of the exact response $\left\{ f_i^{1/\rho}(z_i^k), \nabla f_i^{1/\rho}(z_i^k) \right\}$ to reduce the quantization error. The rationale is that if the exact values fall with high probability inside a range (determined by the predictive covariance matrix) around the predictive mean, then the quantization error is reduced and diminished as the proxGP becomes increasingly accurate, ensuring the optimization's convergence [17]. The quantized response $\left\{ \left( \mathbb{Q}\left(f_i^{1/\rho}(z_i^k)\right), \mathbb{Q}\left(\nabla f_i^{1/\rho}(z_i^k)\right) \right) \right\}$ from agent $i$ is sent back to the coordinator, which uses a similar dequantization process based on the same predictive mean $\mu_i^k(z_i^k)$ and covariance matrix $\Sigma_i^k(z_i^k)$ to obtain the dequantized approximate response $\left\{ \hat{f}_i^{1/\rho}(z_i^k), \nabla \hat{f}_i^{1/\rho}(z_i^k) \right\}$. The dequantized values are used both for the ADMM calculations and for updating the proxGP.

In the next section, we present a review of the important theoretical results relevant to our work.

## 4. Review of Gaussian process and quantization

### 4.1. Gaussian process with derivative observations

Let us assume that we have $m$ observations of a random variable, and $X \in \mathbb{R}^{m \times p}$ whose rows $x_i$ ($i \in [1, m]$) are observed inputs vectors. Considering a mean function $\mu(x_i)$ and the co-variance function $\phi(x_i, x_i')$ of a real process $f(x_i) \in \mathbb{R}$ satisfying positive definite conditions as presented in Chapter 4 of [24], the GP can be written as $f(x_i) \sim \mathcal{GP}\left(\mu(x_i), \phi(x_i, x_i')\right)$.

Now, consider the case where we have extended function values at $x_i \in \mathbb{R}^{1\times p}$ including both the function value and its gradients at $x_i$, denoted by $\left[f(x_i); \nabla f(x_i)\right]$, where $\nabla f(x_i) = \left[\frac{\partial f(x_i)}{\partial x_i^{(d)}}\right]_{d=1,\dots,p}$, and $x_i^{(d)}$ is the $d$-th element of $x_i$. Following [25], the covariance matrix is correspondingly expanded, for any pair of points $s, l \in [1, m]$, resulting in the covariances between the observations and its partial derivatives given by

$$\mathrm{Cov}\left[\frac{\partial f(x_s)}{\partial x_s^{(d_s)}}, f(x_l)\right] = \frac{\partial}{\partial x_s^{(d_s)}}\phi\left(x_s, x_l\right),$$

and between the partial derivatives given by

$$\mathrm{Cov}\left[\frac{\partial f(x_s)}{\partial x_s^{(d_s)}}, \frac{\partial f(x_l)}{\partial x_l^{(d_l)}}\right] = \frac{\partial^2}{\partial x_s^{(d_s)} \partial x_l^{(d_l)}}\phi\left(x_s, x_l\right),$$

where $1 \le d_s, d_l \le p$. The GP then will have its predicted mean and covariance as presented in Chapter 2 of [24].

### 4.2. Uniform quantization

We consider a uniform quantizer $\mathbb{Q}_{\mathrm{u}}$ of the mid-tread type [26], where the input-output relation is given by

$$\mathbb{Q}_{\mathrm{u}}(y; \bar{y}, q) = \bar{y} + q\left(\left\lfloor \frac{y - \bar{y}}{q} \right\rceil + \frac{1}{2}\right),$$

in which $q > 0$ is the quantization window length, $\bar{y}$ is the mid-value, and $\lfloor y \rceil$ denotes the integer closest to $y$ towards 0. Here, $q = \frac{l}{2^b}$, where $l$ is the range of the quantization interval and $b$ is the bit resolution of the quantizer. Let $\hat{y} = \mathbb{Q}_{\mathrm{u}}(y; \bar{y}, q)$, then the quantization error (or quantization noise) is defined as $\epsilon_{\mathbb{Q}} = y - \hat{y}$. The statistics of the quantization error for this uniform quantizer are characterized in Section V-A in [27].

## 5. GP regression under adaptive quantization

In this section, we present the derivations and principles of our proposed approach. We present our proposed adaptive quantization scheme and its properties, the new regression mechanism, and an approximation method to deal with the quantized data.

### 5.1. Adaptive uniform quantization

We propose a quantizer that adapts the standard (non-adaptive) uniform quantizer. Given an input $y$ which is a sample of a Gaussian distribution $\mathcal{N}\left(\mu_y, \sigma_y^2\right)$, we adapt a uniform quantizer by setting its mid-value $\bar{y} = \mu_y$ and its range $l = 2c\sigma_y$, for some given $c > 0$ that controls how many standard deviations apart from the mean $\mu_y$ are set for the range of values for quantization, which determines how confident we are that the quantizer's input is within the defined quantization range. The proposed adaptive quantizer $\mathbb{Q}_{\mathrm{ua}}$ on $y$, given by $\mathbb{Q}_{\mathrm{ua}}(y; \mu_y, \sigma_y, c, b) = \mathbb{Q}_{\mathrm{u}}\left(y; \mu_y, \frac{2c\sigma_y}{2^b}\right) = \mu_y + \frac{2c\sigma_y}{2^b}\left(\left\lfloor \frac{2^b(y-\mu_y)}{2c\sigma_y} \right\rceil + \frac{1}{2}\right)$, therefore has parameters that are adapted for a quantization resolution appropriate for the most likely values of $f(x)$.

The following result characterizes the error statistics of the adaptive uniform quantizer, which will play an important role in the analysis of our proposed adaptive quantization methods throughout the rest of the paper. Its proof is presented in Appendix A.

**Proposition 1.** *Consider a sample $y$ of a Gaussian distribution $\mathcal{N}\left(\mu_y, \sigma_y^2\right)$ and an adaptive uniform quantizer $\mathbb{Q}_{\mathrm{ua}}(y; \mu_y, \sigma_y, c, b)$ on $y$. Define the quantization error $\epsilon_{\mathbb{Q}} = y - \mathbb{Q}_{\mathrm{ua}}(y; \mu_y, \sigma_y, c, b)$. Then the mean and variance of the quantization error are*

$$\mathbb{E}[\epsilon_{\mathbb{Q}}] = 0$$

$$\mathbb{E}[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}] = \frac{q^2}{12}\upsilon(r),$$

*where $q = \frac{2c\sigma_y}{2^b}$, $r = \frac{2^b}{2c}$, and*

$$\upsilon(r) = 1 + \frac{12}{\pi^2}\sum_{m=1}^{\infty}\frac{(-1)^m}{m^2}\exp\left(-2\pi^2 m^2 r^2\right). \tag{6}$$

*Furthermore, the correlation between the input $y$ and the quantization error is given by,*

$$\mathbb{E}[y\epsilon_{\mathbb{Q}}] = 2\sigma_y \sum_{m=1}^{\infty}(-1)^m \exp\left(-2\pi^2 m^2 r^2\right). \tag{7}$$

While $\upsilon(r)$ and $\mathbb{E}[y\epsilon_{\mathbb{Q}}]$, given in (6) and (7), involve complex mathematical series, we will show that when the ratio $r = \frac{2^b}{2c}$ exceeds 1, $\upsilon(r)$ becomes approximately 1 and the correlation $\mathbb{E}[y\epsilon_{\mathbb{Q}}]$ becomes negligible. The following lemmas establish the monotonicity and the negative values of these series. Their proofs can be found in Appendix B.

**Lemma 1.** *The series* $\sum_{m=1}^{\infty} \frac{(-1)^m}{m^2} \exp\left(-2\pi^2 m^2 r^2\right)$ *is negative and increasing with* $r$.

**Lemma 2.** *The series* $\sum_{m=1}^{\infty} (-1)^m \exp\left(-2\pi^2 m^2 r^2\right)$ *is negative. Furthermore, for* $r > \frac{1}{\sqrt{2\pi}} \approx 0.225$, *it is increasing with* $r$.

It follows from these lemmas and equations (6) and (7) that $\upsilon(r) < 1$ and increasing with $r$ for all $r > 0$, and $\mathbb{E}[x\epsilon_{\mathbb{Q}}] < 0$ and increasing with $r$ for all $r > \frac{1}{\sqrt{2\pi}} \approx 0.225$. In practice, the ratio $r = \frac{2^b}{2c}$ is at least 1 and often much greater than 1. Indeed, with the typically chosen $c = 3$ (giving a confidence of 99.7% that the quantizer's input is within the quantization range), at a resolution of just $b = 3$ bits, $r = 4/3 > 1$ and increases exponentially with $b$. At $r = 1$, we have $\upsilon(1) = 1 - 3.253 \times 10^{-9}$, and $\mathbb{E}[y\epsilon_{\mathbb{Q}}] = -5.351 \times 10^{-9}\sigma_y$. Therefore, for all practical purposes, we have $1 - 3.253 \times 10^{-9} \leq \upsilon(r) < 1$, thus we can consider $\upsilon(r) = 1$ and hence $\mathbb{E}[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}] = \frac{q^2}{12}$. In addition, we have $-5.351 \times 10^{-9}\sigma_y \leq \mathbb{E}[y\epsilon_{\mathbb{Q}}] < 0$, thus we can consider $\mathbb{E}[y\epsilon_{\mathbb{Q}}] = 0$.

### 5.2. Adaptive uniform quantization with vector input

Consider the case where the input to the quantizer is a Gaussian random vector $y$ with conditional mean vector $\mu_y$ and conditional co-variance matrix $\Sigma_y$. The previously presented adaptive quantization scheme must be adjusted to handle the multidimensional nature of the input. We propose two schemes described below: one ignores the correlations among the input values and the other takes these correlations into account.

*Adaptive Scheme Ignoring Correlation.* Quantization is performed element-wise, using each element of the quantizer's input with its corresponding element of the conditional mean vector $\mu_y$ and the diagonal of the co-variance matrix $\Sigma_y$ for adaptation. Therefore, we have a vector of window lengths $q$ with the $i^{th}$ entry given by

$$q_i = \frac{2c\sqrt{\Sigma_{y[ii]}}}{2^b}, \tag{8}$$

where $\Sigma_{y[ii]}$ is the $i^{th}$ entry of the diagonal of $\Sigma_y$.

Using Proposition 1, we can characterize the quantization error under the proposed scheme, as stated in the following proposition.

**Proposition 2.** *Under the Adaptive Scheme Ignoring Correlation, an adaptive uniform quantizer* $\mathbb{Q}_{\text{ua}}(y; \mu_y, \Sigma_y, c, b)$ *has a quantization error vector* $\epsilon_{\mathbb{Q}}$ *whose components are uncorrelated. The correlation matrix, defined as* $\Delta_{\text{un}} = \mathbb{E}[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}]$, *is a diagonal matrix with its diagonal given by the vector* $\frac{q^2}{12}\upsilon(2^b/2c)$, *with the entries of vector* $q$ *defined in* (8) *and* $\upsilon(\cdot)$ *defined in Proposition 1.*

*Correlated Adaptive Scheme.* The use of an orthogonal transformation of the quantizer's input $y$ allows us to consider the correlation between its elements, and to perform quantization over the transformed input similarly as in the previously defined *Adaptive Scheme Ignoring Correlation*.

Using the above notations, the orthogonal transformation to the quantizer's input is expressed as

$$y^A = A(y - \mu_y), \tag{9}$$

where $A$ is the transformation matrix. The conditional mean of $y$ is subtracted to have a zero-mean quantizer's input. Then, the way $A$ is determined will define our orthogonal *pre-filtering* of the quantizer's input.

*Pre-filtering*: *The transformation matrix* $A$ *used in* (9) *is obtained by applying an eigenvalue decomposition of matrix* $\Sigma_y$, *in which* $\Sigma_y = U\Lambda U'$, *with* $\Lambda$ *being a diagonal matrix with the eigenvalues of* $\Sigma_y$ *and* $U$ *being a square matrix whose columns are eigenvectors of* $\Sigma_y$. *The matrix* $A$ *can be expressed in two ways;* $A_1 = (\Sigma_y)^{-1/2}$ *or* $A_2 = U'$, *where* $(\Sigma_y)^{1/2}$ *is a matrix such that* $(\Sigma_y)^{1/2}(\Sigma_y)^{1/2} = \Sigma_y$. *The use of* $A_1$ *will result in a whitening procedure where the result will be a zero-mean unit variance vector with independent components. The use of* $A_2$ *will result in a decoupling procedure where the result will be a zero-mean vector whose variances are determined by the eigenvalues in* $\Lambda$.

Following this pre-filtering, $y^A$ will be element-wise quantized given by:

$$\mathbb{Q}_{\text{ua}}(y^A; 0, \Sigma_w, c, b) = y^A + \epsilon_{\mathbb{Q}},$$

where $\Sigma_w$ represents the identity matrix (when $A = A_1$) or a diagonal matrix with entries given by the eigenvalues of $\Sigma_y$ (when $A = A_2$).

**Proposition 3.** *Under the Correlated Adaptive Scheme and the proposed Pre-filtering, an adaptive uniform quantizer* $\mathbb{Q}_{\text{ua}}(y^A; 0, \Sigma_y, c, b)$, *where the input vector is transformed following* (9), *has a quantization error vector* $\epsilon_{\mathbb{Q}}$ *whose components are correlated with each other. The*

*correlation matrix, defined as* $\Delta\text{co} = E[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}]$*, is independent of the choice of the transformation matrix* $A$ *and is given by* $\Delta\text{co} = \frac{c^2 v(2^b/2c)}{3(2^b)^2}\Sigma_y$*, with* $v(\cdot)$ *as defined in Proposition 1.*

**Proof.** The proof is presented in Appendix C. □

### 5.3. LMMSE regression with quantization

In this subsection, we consider a GP regression as presented in Section 4.1, but when the training set $\mathcal{D}$ is affected by adaptive quantization. In this scenario, we do not have access to the exact extended values $y_i$ but a quantized version of them $\hat{y}_i = \left[\mathbb{Q}_{\text{ua}}(f(x_i)); \mathbb{Q}_{\text{u}}(\nabla f(x_i))^T\right] + \epsilon^i_{\text{n}}$, which are quantized following the proposed adaptive quantization with vector inputs presented in Section 5.2. These quantized extended values are also expressed as $\hat{y}_i = \left[f(x_i); \nabla f(x_i)^T\right] + \epsilon^i_{\text{n}} + \epsilon^i_{\mathbb{Q}}$, where $\epsilon^i_{\mathbb{Q}}$ refers to the quantization error vector for the observation $i$ and $\epsilon^i_{\text{n}}$ is a vector whose entries follow the same Gaussian distribution with zero mean, $\sigma_n^2$ variance at observation $i$. Such Gaussian noise is not a physical noise but one added to avoid possible matrix singularity.

The added non-Gaussian quantization noise invalidates the Gaussian noise assumption of the regular GP regression. In this case, the regression cannot be a Minimum Mean Square-error Estimator (MMSE) anymore, so we must compute the conditional mean which requires a more involved computation. To overcome this challenge, we adopt a Linear Minimum Mean Square-error Estimator (LMMSE). This allows us to balance the accuracy and complexity of the estimator while preserving the advantages of GP. With this premise we will derive two estimators under two scenarios regarding the training set $\mathcal{D}$.

#### 5.3.1. Linear GP regression (LGP-R)

This estimator is used to predict the extended values of an input $x_*$ given a training set where the observed extended values are affected by quantization. In this case, we only have access to quantized values of the extended values. For a new input $x_*$ we want to predict $y_*$, leading to the following theorem, whose proof is presented in Appendix D. This estimation is performed at every iteration, and for every agent to assess the quality of regression.

**Theorem 1.** *The LGP-R Estimator has an input* $x_* \in \mathbb{R}^p$ *and a training set containing* $m$ *past observations with quantized extended values* $\mathcal{D} = (X, \hat{Y})$*, with* $X \in \mathbb{R}^{m(p+1)\times p}$ *being a collection of the past inputs* $x_i \in \mathbb{R}^{(p+1)\times p}$ *and* $\hat{Y} \in \mathbb{R}^{m(p+1)\times 1}$ *being a collection of the past quantized extended observation values* $\hat{y}_i \in \mathbb{R}^{(p+1)\times 1}$*. This estimator has its predicted mean*

$$\mu(x_*) = \Phi(X_*, X)\left(\Phi(X, X) + \sigma_n^2 I_{m(p+1)} + \Delta + 2\mathbb{E}[Y\epsilon'_{\mathbb{Q}}]\right)^{-1}\hat{Y},$$

*and predicted covariance matrix*

$$\Sigma(x_*) = \Phi(X_*, X_*) - \Phi(X_*, X)\left(\Phi(X, X) + \sigma_n^2 I_{m(p+1)} + \Delta + 2\mathbb{E}[Y\epsilon'_{\mathbb{Q}}]\right)^{-1}\Phi(X, X_*),$$

*where* $X_* \in \mathbb{R}^{(p+1)\times p}$ *contains a copy of* $x_*$ *in each of its rows, the entries of the matrices* $\Phi(X_*, X_*)$*,* $\Phi(X_*, X)$*, and* $\Phi(X, X)$ *are as detailed in Subsection 4.1,* $\Delta = \mathbb{E}[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}]$ *contains the information of the uniform quantization error of all extended values observations of the training set* $\mathcal{D}$*, and the entries corresponding to each observation in* $\Delta$ *are added block-wise following the expression given by* $\Delta_{\text{un}}$ *in Proposition 2 or* $\Delta_{\text{co}}$ *in Proposition 3 (depending on the quantization scheme selected), and* $\mathbb{E}[Y\epsilon'_{\mathbb{Q}}]$ *is the correlations between the extended observation values* $Y$ *in the training set* $\mathcal{D}$ *and their corresponding uniform quantization errors, calculated as shown in Proposition 1.*

#### 5.3.2. Linear GP approximation (LGP-A)

Consider the case where we perform adaptive uniform quantization on the extended values at $x_*$, resulting in the quantized version of $y_*$ given by $\hat{y}_*$. Such adaptive quantization uses the conditional mean and conditional covariance given by LGP-R. It is possible to approximate the real value $y_*$ if $\hat{y}_*$ and the statistics that adapt the quantizer are known. To do so, we propose the construction of a LMMSE named LGP-A to be performed after the quantization process. This estimation is only performed when communication is required and after receiving the reply from the agent.

The estimation could be performed by updating the training set with the new input and the quantized extended values. Input $x_*$ could then be reinserted to the estimator presented in Theorem 1. To avoid such redundancy we consider an approximator that deals with a zero-mean input $\hat{y}_* - \mu(x_*)$, and since $\hat{y}_*$ already has the information of the past training set, we then have the following theorem, whose proof is presented in Appendix E.

**Theorem 2.** *The LGP-A Estimator has a training set containing* $m$ *past inputs, past quantized extended observation values, and the current input* $x_*$ *and its quantized extended observation value* $\hat{y}_*$*, leading to the training set* $\mathcal{D} = ([X; x_*], [\hat{Y}; \hat{y}_*])$*, with* $X \in \mathbb{R}^{m(p+1)\times p}$ *being a collection of the past inputs* $x_i \in \mathbb{R}^{(p+1)\times p}$*, and* $\hat{Y} \in \mathbb{R}^{m(p+1)\times 1}$ *being a collection of the past quantized extended observation values* $\hat{y}_i \in \mathbb{R}^{(p+1)\times 1}$*. LGP-A estimates the target value* $y_*$ *by*

$$\bar{y}_* = B\left(\hat{y}_* - \mu(x_*)\right) + \mu(x_*),$$

*where* $B = \Sigma(x_*)\left(\Sigma(x_*) + \Delta_{p+1} + \sigma_n I_{p+1} + 2\mathbb{E}[y_*\epsilon'_{\mathbb{Q}*}]\right)^{-1}$*, with* $\mu(x_*)$ *and* $\Sigma(x_*)$ *as presented in Theorem 1 and* $\Delta_{p+1}$ *is given by* $\Delta_{\text{un}}$ *in Proposition 2 or* $\Delta_{\text{co}}$ *in Proposition 3 depending on the quantization scheme selected,* $\epsilon_{\mathbb{Q}*}$ *is the quantization error of only the quantized values in the current iteration, and* $\mathbb{E}[y_*\epsilon'_{\mathbb{Q}*}]$ *is calculated as shown in Proposition 1.*

## 6. Proposed approach

### 6.1. Proposed adaptive uniform quantization scheme

This section combines the overview presented in Section 3 with the results presented in Section 5 to present our complete proposed approach in more detail.

In Fig. 1, upon receiving the query point $z_i^k \in \mathbb{R}^{1\times p}$ from the coordinator (left side), agent $i$ (right side) solves the proximal minimization problem (3) (the box $\mathbf{prox}_{1/\rho f_i}$) and obtains the exact values of $f_i^{1/\rho}(z_i^k) \in \mathbb{R}$ and $\nabla f_i^{1/\rho}(z_i^k) \in \mathbb{R}^{p\times 1}$. Simultaneously, it uses the regression process, depicted in the block 'proxLGP', to obtain the conditional mean $\mu_i^k(z_i^k)$, which stores the predicted values of $f_i^{1/\rho}(z_i^k)$ and $\nabla f_i^{1/\rho}(z_i^k)$, and the conditional covariance matrix $\Sigma_i^k(z_i^k)$. We can adopt the same adaptive uniform quantization scheme presented in Section 5.1, as the exact values follow a Gaussian distribution (under the LGP model). We will denote the quantized values of the query response as $\left[\hat{f}_i^{1/\rho}(z_i^k); \nabla \hat{f}_i^{1/\rho}(z_i^k)\right] = \mathbb{Q}_{\text{ua}}\left(\left[f_i^{1/\rho}(z_i^k); \nabla f_i^{1/\rho}(z_i^k)\right]; \mu_i^k(z_i^k), \Sigma_i^k(z_i^k), c, b\right)$. The output of the quantizer is transmitted from the agent (right side) to the coordinator (left side). The dequantized values $\hat{f}_i^{1/\rho}(z_i^k)$ and $\nabla \hat{f}_i^{1/\rho}(z_i^k)$ are used by the ADMM algorithm and to update the corresponding 'proxLGP' of agent $i$.

### 6.2. LGP-R based regression in our proposed approach

The 'proxLGP' block on the coordinator side of Fig. 1 runs at every iteration and its resulting covariance matrix is used to determine whether to send $z_i^k$ to agent $i$.

Using the quantization scheme for vector inputs $\mathbb{Q}_{\text{ua}}$ (defined in Section 5.2) and following (8), the results presented in Propositions 1-3 apply to the adaptive quantizer $\mathbb{Q}_{\text{ua}}$. Hence, we can use the previously derived regression scheme LGP-R presented in Theorem 1 as the regression scheme to be used in this work. Using the results in Section 5.1 that $\mathbb{E}[y\epsilon'_{\mathbb{Q}}] \approx 0$ and $\upsilon(r) \approx 1$, we henceforth remove the correlation $\mathbb{E}[y\epsilon'_{\mathbb{Q}}]$ present in Theorems 1 and 2, and remove the term $\upsilon(2^b/2c)$ used in the characterization of the variance of the quantization error in Propositions 2 and 3.

Now, defining $g_i^{1/\rho}(z_i^k) = \left[f_i^{1/\rho}(z_i^k); \nabla f_i^{1/\rho}(z_i^k)\right]$, we have that, given the new query point $z_i^k$, the predicted value of the vector $g_i^{1/\rho}(z_i^k)$ using LGP-R will be given by

$$\mu_i^k(z_i^k) = \Phi(Z_{i*}^k, Z_i^k)\left(\Phi(Z_i^k, Z_i^k) + \sigma_n^2 I_{m(p+1)} + \Delta_i\right)^{-1} \hat{G}_i^k, \tag{10}$$

where $Z_{i*}^k \in \mathbb{R}^{(p+1)\times p}$ contains a copy of $z_i^k$ in each of its rows, $Z_i^k$ is the training input set containing queries sent to agent $i$ up to time $k$ in the set $\{z_i^j\}_{j\in\mathcal{J}_i}$, $\mathcal{J}_i^k$ contains the indices of the iterations where a query was sent to agent $i$ by the coordinator up to the current algorithmic iteration, $m$ is the number of elements in set $\mathcal{J}_i^k$, $\hat{G}_i^k$ is the quantized training target set containing the local quantized proximal minimization problem results sent from agent $i$ to the coordinator up to time $k$ in the set $\left\{\mathbb{Q}_{\text{ua}}\left(g_i^{1/\rho}(z_i^j); \mu_i^j(z_i^j), \Sigma_i^j(z_i^j), c, b\right)\right\}_{j\in\mathcal{J}_i}$, $\sigma_n^2 I_{m(p+1)}$, $\Delta_i$ are defined in Theorem 1, and the entries of $\Phi(Z_{i*}^k, Z_i^k)$ and $\Phi(Z_i^k, Z_i^k)$ are detailed in Subsection 4.1 with a covariance function given by the square exponential kernel function.

Using the same notation, the covariance matrix given by the LGP-R is

$$\Sigma_i^k(z_i^k) = \Phi(Z_{i*}^k, Z_{i*}^k) - \Phi(Z_{i*}^k, Z_i^k)\left(\Phi(Z_i^k, Z_i^k) + \sigma_n^2 I_{m(p+1)} + \Delta_i\right)^{-1} \Phi(Z_i^k, Z_{i*}^k). \tag{11}$$

The matrix $\Delta_i$ will be updated block-wise by inserting the corresponding quantization error covariance matrix of the query round, which follows Proposition 2 or Proposition 3 depending on the quantization scheme used. Henceforth, we will use $\Delta_i^k$ to refer to the resulting quantization error covariance matrix obtained after a query process in iteration $k$, which will be then added to $\Delta_i$.

### 6.3. LGP-A approximation in our proposed approach

In Fig. 1 we can see that the coordinator receives the quantized version $\nabla \hat{f}_i^{1/\rho}(z_i^k)$ of the exact value $\nabla f_i^{1/\rho}(z_i^k)$. To improve the accuracy of the gradient values used in the ADMM updates at the coordinator, we estimate these values with a LMMSE estimator rather than using the inexact quantized values directly. The estimator derived in this subsection is different from that in subsection 6.2 because it is applied only when a query is performed, which only uses the newly added entry in the training set. The result is further used by the ADMM process.

After a query undergoes a communication round, the quantized value of $g_i^{1/\rho}(z_i^k)$, $\hat{g}_i^{1/\rho}(z_i^k)$, is added to the regression training set, and $\Delta_i$ is updated with the block $\Delta_i^k$. Therefore, we can obtain the desired approximation $\bar{g}_i^{1/\rho}(z_i^k)$ following the derivation from Theorem 2, which gives us

$$\bar{g}_i^{1/\rho}(z_i^k) = B_i^k\left(\hat{g}_i^{1/\rho}(z_i^k) - \mu_i^k(z_i^k)\right) + \mu_i^k(z_i^k), \tag{12}$$

where $B_i^k = \Sigma_i^k(z_i^k)\left(\Sigma_i^k(z_i^k) + \sigma_n I_{p+1} + \Delta_i^k\right)^{-1}$.

**Algorithm 1** LGP: Distributed Optimization with Estimated Proximal Operator Based on Gaussian Processes with Adaptive Uniform Quantization.

**Require:** $x_i^0 \in \mathbb{R}^p$, $\bar{y}^0 \in \mathbb{R}^p$, $u^0 \in \mathbb{R}^p$, $c \in \mathbb{N}$, $b \in \mathbb{N}$
**for** $k = 0, 1, \ldots, k_{\text{stop}}$ **do**
  $\bar{y}^{k+1} \leftarrow \underset{\bar{y}\in\mathbb{R}^p}{\arg\min} \left\{ h(n\bar{y}) + (n\rho/2)\|\bar{y} - \bar{x}^k - u^k\|^2 \right\}$
  **for** each agent $i$ **do**
    $z_i^k \leftarrow x_i^k - \bar{x}^k + \bar{y}^{k+1} - u^k$
    Calculate $\mu_i^k(z_i^k)$ and $\Sigma_i^k(z_i^k)$ from (10) and (11)
    **if** $\max\left(\text{diag}\left(\Sigma_i^k(z_i^k)\right)\right) > \psi_i^k$ **then**
      Send $z_i^k$ to Agent $i$
      $\hat{g}_i^{1/\rho} \leftarrow \text{QUERYAGENT}(z_i^k)$ ▷ Agent $i$
      Compute $\bar{g}_i^{1/\rho}$ from (12)
      Add $\left(z_i^k, \hat{g}_i^{1/\rho}(z_i^k)\right)$ to the GP training set
      Perform the GP hyperparameter update.
      $x_i^{k+1} \leftarrow z_i^k - (1/\rho)\nabla \bar{f}_i^{1/\rho}(z_i^k)$
    **else**
      $x_i^{k+1} \leftarrow z_i^k - (1/\rho)\mu_i^k(z_i^k)$
    **end if**
  **end for**
  $\bar{x}^{k+1} \leftarrow (1/n)\sum_{i=1}^n x_i^{k+1}$
  $u^{k+1} \leftarrow u^k + \bar{x}^{k+1} - \bar{y}^{k+1}$
  If $\|\bar{x}^k - \bar{y}^k\|_\infty \leq \epsilon_p\left(1 + \|\lambda^k/\rho\|_\infty\right)$ then Terminate.
**end for**

### 6.4. Dithering

From Section 5.1, we have that the correlation between the quantization noise and the input is negligible when the quantization bit resolution ($b$) becomes larger and we fix a small value for $c$. If $b$ is too small, we can introduce dithering to randomize the quantization error and break the correlation between this error and the quantizer input.

A recent study ([28]) explores the use of quantization with dithering to determine which distribution the subtractive dithering follows. The work presented in [29] shows that the use of dithering with quantization could be improved if an orthogonal transformation was performed on the quantizer input prior to the quantization process. We thus adopt dithering as part of quantization after orthogonal transformation is performed at the quantizer's input.

When the uniform quantizer is used with a zero-mean Gaussian input, the dithering variable $d_i^k$ will be a random number coming from a uniform distribution $d_{i[r]}^k \sim \mathcal{U}\left(\frac{-q_{i[r]}^k}{2}, \frac{q_{i[r]}^k}{2}\right)$, where the window length $q_{i[r]}^k$ is as defined in (8). The dithering will be performed element-wise, so $d_i^k$ will have the same dimension as the quantizer input. Following the orthogonal transformation as in Section 5.2, the quantizer input with dithering is given by

$$g_i^{A[d]}(z_i^k) = g_i^A(z_i^k) + d_i^k, \tag{13}$$

where $g_i^A(z_i^k) = A\left(g_i^{1/\rho}(z_i^k) - \mu_i^k(z_i^k)\right)$, with $A$ as presented in the *Pre-filtering*. Then, $g_i^{A[d]}(z_i^k)$ will be quantized and sent to the coordinator. The coordinator then performs the dequantization process and subtracts the noise added to the input before adding back its mean. The value $\hat{g}_i^{1/\rho}(z_i^k)$ is given by

$$\hat{g}_i^{1/\rho}(z_i^k) = A^{-1}\left(g_i^{A[d]}(z_i^k) + \epsilon_{\mathbb{Q}i}^k - d_i^k\right) + \mu_i^k(z_i^k),$$

where $\epsilon_{\mathbb{Q}i}^k$ is the $i^{\text{th}}$ agent quantization noise at iteration $k$.

### 6.5. LGP pseudo-code

The complete LGP algorithm considering all its different variations is presented in Algorithm 1.

## 7. Numerical simulations

In this section, we evaluate the methods proposed in this work by solving a sharing problem where the agent's sub-problems are quadratic. The specifics of the sharing problem considered, the simulation settings, and the results obtained are presented next.

### 7.1. Sharing problem

#### 7.1.1. Problem definition

Our testing problem is based on the application presented in [11]. In this example, a dynamic sharing problem where the problem's variables change at each iteration is presented and solved via ADMM. In our work, those varying variables are fixed and do not vary at each algorithmic step. We consider the following sharing problem:

**Algorithm 2** Query Process at the Agent Side.

**procedure** QUERYAGENT($z_i^k$)
  Compute $f_i^{1/\rho}(z_i^k)$ and $\nabla f_i^{1/\rho}(z_i^k)$ from (4)
  $g_i^{1/\rho} \leftarrow \left[f_i^{1/\rho}(z_i^k); \nabla f_i^{1/\rho}(z_i^k)\right]$
  **if** Using *Adaptive Scheme Ignoring Correlation* **then**
    $\hat{g}_i^{1/\rho} \leftarrow \mathbb{Q}_{\text{ua}}\left(g_i^{1/\rho}; \mu_i^k(z_i^k), \Sigma_i^k(z_i^k), c, b\right)$
  **else**
    Perform decomposition $\Sigma_i^k(z_i^k) = U_i^k \Lambda_i^k U_i^{k'}$
    **if** Using *Whitening Transformation* **then**
      $A_i^k \leftarrow \left(\Sigma_i^k(z_i^k)\right)^{-1/2}$
    **end if**
    **if** Using *Decoupling Transformation* **then**
      $A_i^k \leftarrow U_i^{k'}$
    **end if**
    $g_i^A \leftarrow A_i^k \left[g_i^{1/\rho} - \mu_i^k(z_i^k)\right]$
    **if** Using *Dithering* **then**
      Compute $g_i^{A[d]}$ as in (13)
      $\hat{g}_i^{1/\rho} \leftarrow \mathbb{Q}_{\text{ua}}\left(g_i^{A[d]}; 0, \Sigma_i^k(z_i^k), c, b\right) + \mu_i^k(z_i^k)$
    **else**
      $\hat{g}_i^{1/\rho} \leftarrow \mathbb{Q}_{\text{ua}}\left(g_i^A; 0, \Sigma_i^k(z_i^k), c, b\right) + \mu_i^k(z_i^k)$
    **end if**
  **end if**
  **return** $\hat{g}_i^{1/\rho}$
**end procedure**

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n}(x_i-\theta_i)^T\Upsilon_i(x_i-\theta_i) + \zeta\|\sum_{i=1}^{n} y_i\|_1 \\ &\text{subject to} && x_i - y_i = 0 \end{aligned} \tag{14}$$

where $x_i, y_i \in \mathbb{R}^p$, $\theta_i \in \mathbb{R}^p$, $\Upsilon_i \in \mathbb{R}^{p\times p}$ positive definite, and $\zeta > 0$ are given problem parameters.

As presented in [11], the problem in (14) can be applied to data flow in communication networks or currents in power grids, where there are $n$ subsystems and $p$ quantities distributed over such subsystems. The vector $x_i$ describes the $p$ quantities at subsystem $i$, and the goal is to determine the solution vectors $x_i$, $i = 1, 2, \ldots, n$.

### 7.1.2. Generation of parameters $\theta_i$ and $\Upsilon_i$

The details are presented in Appendix F.

### 7.1.3. Solution with ADMM

The problem presented in (14) has the same form as (2) in Section 3 based on which the ADMM updates for this case are expressed as

$$\begin{aligned} x_i^{k+1} &= \underset{x_i\in\mathbb{R}^p}{\arg\min}\left\{f_i(x_i) + (\rho/2)\|x_i - z_i^k\|_2^2\right\} \\ \bar{y}^{k+1} &= \underset{\bar{y}\in\mathbb{R}^p}{\arg\min}\left\{\zeta\|n\bar{y}\|_1 + (n\rho/2)\|\bar{y} - \bar{x}^{k+1} - (1/\rho)\lambda^k\|_2^2\right\} \\ \lambda^{k+1} &= \lambda^k + \rho\left(\bar{x}^{k+1} - \bar{y}^{k+1}\right) \end{aligned} \tag{15}$$

where $f_i(x_i) = (x_i-\theta_i)^T\Upsilon_i(x_i-\theta_i)$, $\bar{x}^k = (1/n)\sum_{i=1}^n x_i^k$, $\bar{y}^k = (1/n)\sum_{i=1}^n y_i^k$, and $z_i^k = x_i^k - \bar{x}^k + \bar{y}^k - (1/\rho)\lambda^k$.

Since the functions $f_i$ and the $l_1$ norm are strongly convex, the ADMM updates for $x_i^{k+1}$ and $\bar{y}^{k+1}$ are solutions to unconstrained convex optimization problems. Thus, those problems can be solved by calculating the derivatives of the objective functions in (15), and setting them equal to zero. Following this, $x_i^{k+1}$ can be expressed by the closed form solution

$$x_i^{k+1} = \left(2\Upsilon_i + \rho I_p\right)^{-1}\left(2\Upsilon_i\theta_i + \rho(x_i^k - \bar{x}^k + \bar{y}^k) - \lambda^k\right), \tag{16}$$

where $I_p$ is the $p\times p$ identity matrix.

Similarly, the $\bar{y}$ update can expressed as

$$\bar{y}^{k+1} = \begin{cases} \left(\bar{x}^{k+1} + \lambda^k/\rho\right) - \frac{\zeta}{\rho}, & \text{if } \bar{x}^{k+1} + \lambda^k/\rho > \frac{\zeta}{\rho} \\ 0, & \text{if } |\bar{x}^{k+1} + \lambda^k/\rho| \le \frac{\zeta}{\rho} \\ (\bar{x}^{k+1} + \lambda^k/\rho) + \frac{\zeta}{\rho}, & \text{if } \bar{x}^{k+1} + \lambda^k/\rho < -\frac{\zeta}{\rho}. \end{cases} \tag{17}$$

**Table 1**
Elements associated with each of the proposed methods.

| | GP Reg | LGP Reg | Uni Quant | Decoup | Whitening | Dithering |
|---|---|---|---|---|---|---|
| Sync:UniQuant | | | ✓ | | | |
| STEP-GP:Exact | ✓ | | | | | |
| STEP-LGP:UniAd | | ✓ | ✓ | | | |
| STEP-LGP:UniAd-Dec | | ✓ | ✓ | ✓ | | |
| STEP-LGP:UniAd-DecDit | | ✓ | ✓ | ✓ | | ✓ |
| STEP-LGP:UniAd-Whit | | ✓ | ✓ | | ✓ | |
| STEP-LGP:UniAd-WhitDit | | ✓ | ✓ | | ✓ | ✓ |

### 7.2. Simulation implementation

We consider two cases where $n \in \{10, 30\}$. The problem described in (14) is solved with four different methods:

1. *Direct*: this method uses a convex solver to solve the problem directly. The knowledge of the true solution is used to construct the comparative metric which is introduced in the following subsection.
2. *Sync*: this algorithm uses ADMM with proximal operator as in (15), which simplifies to (16) and (17) with $\rho = 10$.
3. *STEP-GP*: the algorithm proposed in [15] combining ADMM with proximal operator with GP regression.
4. *STEP-LGP*: the hybrid algorithm proposed in this paper, which combines the regression algorithm developed in Section 6.2, the LMMSE approximation presented in Section 6.3, and the adaptive quantization method developed in Section 6.1.

For each of the above algorithms, different quantization methods, or no quantization at all, are considered as follows:

- *Exact*: this method does not employ any quantization but uses 64-bit floating point numbers.
- *UniQuant*: this uniform quantization adaptation scheme is proposed in [17] to quantize the communications between agents in a connected network using the Proximal Gradient Method (PGM). In case the quantizer's input is a vector the quantization is performed element-wise. For each element of the quantizer's input, an initial quantizer's range is set which decreases at a linear rate over the algorithmic iterations and the quantizer's mid-value is set to be the previous quantized value.
- *UniAd*: this is the adaptive uniform quantization method as presented in Section 6.1 and performed element-wise following the *Uncorrelated Adaptive Scheme* as presented in Section 5.2a.
- *UniAd-Dec*: this is the adaptive uniform quantization method as presented in Section 6.1 and following the *Correlated Quantization Scheme* as presented in Section 5.2b with decoupling.
- *UniAd-DecDit*: same as *UniAd-Dec* but adding the dithering procedure as presented in Section 6.4.
- *UniAd-Whit*: this is the adaptive uniform quantization method as presented in Section 6.1 and following the *Correlated Quantization Scheme* with whitening.
- *UniAd-WhitDit*: same as *UniAd-Whit* but adding the dithering procedure as presented in Section 6.4.

In our simulations, we consider the following combinations: *Sync:Exact*, *Sync:UniQuant*, *STEP-GP:Exact*, *STEP-LGP:UniAd*, *STEP-LGP:UniAd-Dec*, *STEP-LGP:UniAd-DecDit*, *STEP-LGP:UniAd-Whit*, and *STEP-LGP:UniAd-WhitDit*. Table 1 summarizes each proposed combination's algorithmic components.

The simulations were implemented in MATLAB. The solution of the minimization problems (14) is obtained directly using a convex solver from the YALMIP toolbox [30]. We used the GPstuff toolbox [31] for the regression training and inference. The computation was conducted with high-performance computational resources provided by Louisiana State University (http://www.hpc.lsu.edu).

### 7.3. Metrics and considerations

#### 7.3.1. MAC metric

To consider a more realistic communication process, we include a simulation component to reflect the channel contention. By modifying the simulator in [32], we get that the total transmission time will be $Tx_t = \sum_{k=1}^{N} T^k_{\text{round}}$, where $N$ is the number of iterations taken to reach convergence, and $T^k_{\text{round}}$ is the expected transmission time in one iteration round. Appendix G presents the specifics of how this metric was obtained.

#### 7.3.2. ADMM termination criterion

We propose a termination criterion for ADMM using the concept of primal-residual as shown in [7], having the form:

$$\|\bar{x}^k - \bar{y}^k\|_\infty \leq \epsilon_p \left(1 + \|\lambda^k/\rho\|_\infty\right),$$

where $x^k$, $y^k$, and $\lambda^k$ are the variables used in the ADMM (see Section 3) and $\epsilon_p$ is an adjustable tolerance whose value will affect the trade-off between communication reduction and accuracy.

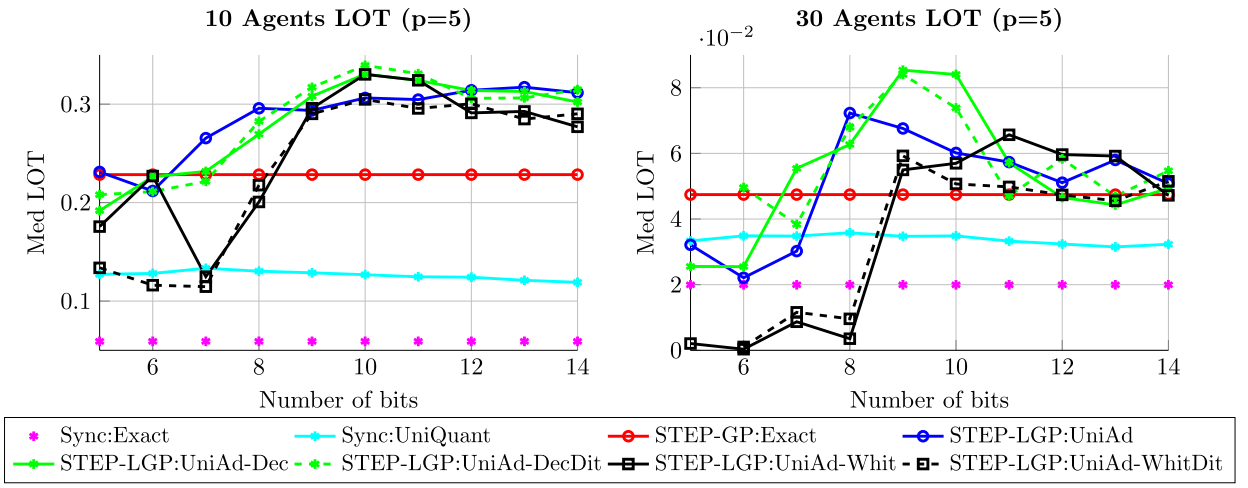


**Fig. 2.** Performance in the LOT metric of the adaptive quantization methods at different bit resolutions for 10 agents (left) and 30 agents (right) with $p = 5$. The plots show the median LOT of 100 simulations for different sets of parameters $\theta_i$ and $\Upsilon_i$.

### 7.3.3. Performance metric

To compare our results, we propose the *Log Optimality over Transmission time (LOT)* performance metric

$$LOT = -\log\left(\left|J_{gt} - J_*\right| / J_{gt}\right) / Tx_t$$

where $J_{gt}$ is the true optimal value obtained by the *Direct* method, $J_*$ is the objective value obtained by a particular approach, and $Tx_t$ the total transmission time defined in Section 7.3.1. This metric reflects both communication cost and efficacy of a given approach. In particular, we want both the absolute error in the numerator and the transmission time in the denominator to be small, hence a higher LOT value is better.

### 7.3.4. Querying mechanism

The coordinator decides if a query should be sent to agent $i$ using a heuristic criterion utilizing the maximum component of the diagonal of the covariance matrix of the gradients of the Moreau Envelope. Specifically, if $\max\left(\text{diag}\left(\Sigma_i^k(z_i^k)\right)\right) > \psi_i^k$ then communication is needed, otherwise it is not. The threshold $\psi_i^k$ is adapted at the coordinator side based on the setting of an initial threshold which will decrease at each iteration according to a decay rate $\alpha$, such that $0 < \alpha < 1$. At $k_0$, which is the iteration where the GP regression is used for the first time, the initial threshold for agent $i$ ($\psi_i^{k_0}$) is calculated following $\psi_i^{k_0} = \iota \max\left(\text{diag}\left(\Sigma_i^{k_0}(z_i^{k_0})\right)\right)$, where $0 < \iota < 1$. At iteration $k > k_0$, no matter the communication decision made by agent $i$, the threshold will be updated as $\psi_i^k = \psi_i^{k_0}(\alpha)^{k-k_0}$.

## 7.4. Simulation results with $p = 5$

In this subsection, we present the results for 10 and 30 agents when the dimension of the variables is set to be $p = 5$. We also set the variable $\iota$ for the querying mechanism described in Section 7.3.4 to be 0.6 for all agents. Each algorithm with the different combinations of quantization methods was run 100 times with different sets of randomly generated $\theta_i$ and $\Upsilon_i$, and the results are shown in terms of the median statistic among all simulations. We used such metric to mitigate the effect of outliers. The median is taken considering only the convergent cases for each method across the considered quantization levels. We consider a case to be non-convergent when the ADMM algorithm do not stop before reaching the maximum number of iterations manually set by us. In our simulations, we considered a maximum iteration count of 250 for a network of 10 agents and 300 when considering 30 agents. This set of results considered values of $\eta = 0.2$, $\epsilon = \zeta = 1$, $\rho = 10$, $p = 5$, a tolerance value of $\epsilon_p = 10^{-6}$, $x_i^0 = \bar{z}^0 = \lambda^0 = 0$, and constant $c = 3$ for quantization.

### 7.4.1. Results for 10 agents

Fig. 2 (left) shows the results of the median of the 100 simulations for ADMM, STEP-GP and STEP-LGP based methods using the metric presented in Section 7.3.3 through the various quantization resolutions tested. The minimum resolution for which any quantization method achieved convergence was 5 bits.

In terms of the LOT metric, STEP-GP presented a better performance in all cases compared to the baseline approaches Sync:UniQuant and Sync:Exact. Also, it can be seen that starting from a resolution of 9 bits the performance of any STEP-LGP based method was better than STEP-GP, Sync:UniQuant, and Sync:Exact, with the peak of performance occurring at 10 bits for STEP-LGP:UniAd-DecDit. For resolutions below 9 bits, STEP-LGP:UniAd outperformed the STEP-GP case starting from 7 bits while STEP-LGP:UniAd-Dec and STEP-LGP:UniAd-DecDit did it starting from 8 bits. For 8 and 7 bits, it is STEP-LGP:UniAd which achieved the best overall performance while STEP-LGP:UniAd-Whit and STEP-LGP:UniAd-WhitDit could not beat the STEP-GP algorithm. Overall, STEP-LGP:UniAd

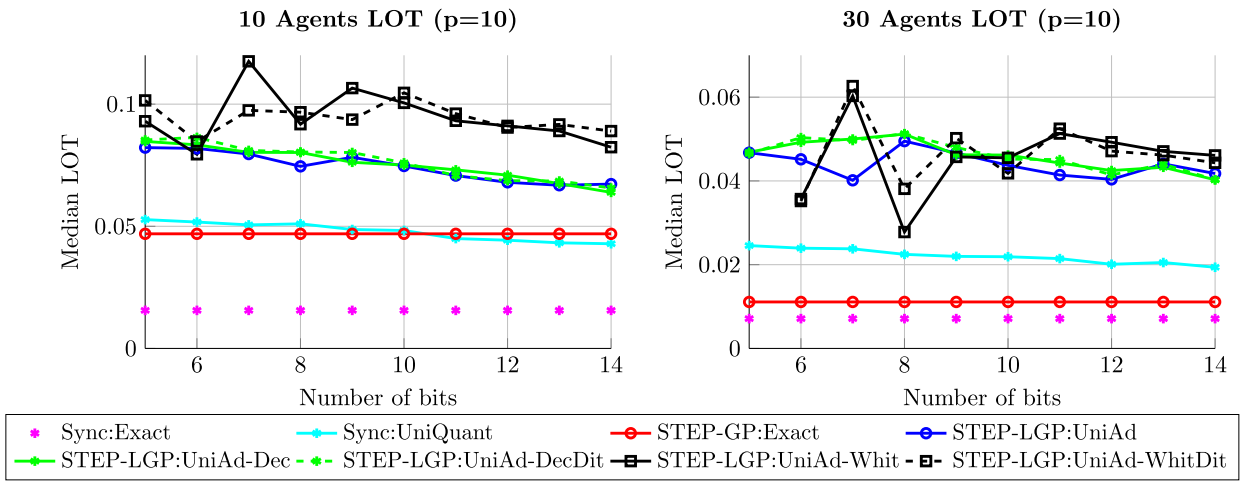


**Fig. 3.** Performance in the LOT metric of the adaptive quantization methods at different bit resolutions for 10 agents (left) and 30 agents (right) with $p = 10$. The plots show the median LOT of 100 simulations for different sets of parameters $\theta_i$ and $\Upsilon_i$.

performed consistently good for all the presented resolutions with STEP-LGP:UniAd-Dec and STEP-LGP:UniAd-DecDit presenting the peak of performance starting from a quantization resolution of 9 bits.

### *7.4.2. Results for 30 agents*

The performance, in this case, is different than the 10 agents case according to Fig. 2 (right) in terms of the LOT metric. It can be seen that STEP-GP presented a better performance in all cases compared to the baseline approaches Sync:UniQuant and Sync:Exact, however the difference in performance is not as notorious as in the previous case. Similarly to the 10 agents case, STEP-LGP:UniAd-DecDit presented the peak of performance but this time it does for the 9 bits case. Between the 5-8 bits interval, STEP-LGP:UniAd-Whit and STEP-LGP:UniAd-WhitDit could not outperform STEP-GP, Sync:UniQuant, or Sync:Exact, while the rest of methods using LGP regression always outperformed Sync:Exact and were all able to outperform STEP-GP and Sync:UniQuant starting from the 8 bits case. For 9 and 10 bits, all LGP-based methods presented better performance than STEP-GP with STEP-LGP:UniAd-Dec and STEP-LGP:UniAd-DecDit presenting the better LOT values by a significant margin. Between 11 and 14 bits, the best performance was always attained by a method involving quantization. However, it is noted that the margin between STEP-GP and the methods using LGP regression was significantly reduced compared to the 10 agents case.

## *7.5. Simulation results with $p = 10$*

In this subsection, we discuss the results for 10 and 30 agents when the dimension of the variables is set to be $p = 10$. The initialization parameters and constant variables considered are the same as in the previous subsection. The corresponding graphs are presented in Fig. 3.

### *7.5.1. Results for 10 agents*

We generated results of the median of 100 simulations for ADMM, STEP-GP and STEP-LGP-based methods using the metric presented in Section 7.3.3 through the various quantization resolutions tested. The minimum resolution at which any quantization method achieved convergence was 5 bits.

In terms of the LOT metric, STEP-GP presented a better performance compared to Sync:Exact but it was outperformed by Sync:UniQuant in the cases where such a method had a quantization resolution between 5 and 10 bits. Also, it is observed a stable performance of all the methods using LGP regression through all the quantization resolutions tested as shown in Fig. 3 (left). In all the cases, those methods consistently beated STEP-GP. The peak of performance was attained by STEP-LGP:UniAd-Whit at 7 bits beating by a small margin its own result for the 9 bits case. Through all the results it is either STEP-LGP:UniAd-Whit or STEP-LGP:UniAd-WhitDit the method that presented the best performance, with the only exception being the 6 bits case. Starting from 10 bits, the methods using whitening presented a significantly better performance compared to all the other methods. Finally, STEP-LGP:UniAd, STEP-LGP:UniAd-Dec, and STEP-LGP:UniAd-DecDit presented a similar behavior through the different quantization resolutions.

### *7.5.2. Results for 30 agents*

Also, we generated the results for 30 agents following the same procedure as in the previous subsection. In Fig. 3 (right) we can see that the performance, in this case, was similar to the 10 agents case in terms of the LOT metric. The most notorious difference was that STEP-GP was outperformed by Sync:UniQuant for all the tested quantization resolutions. In all the cases, LGP-based methods consistently outperformed STEP-GP. Different from the 10 agents case, the methods STEP-LGP:UniAd-Whit and STEP-LGP:UniAd-WhitDit did not present the same notorious improvement in performance compared to the rest of the methods, however, they still attained the best performance for the 7 bits case.

### 7.6. Overall remarks

The behavior of methods using whitening transformation reflects that a more complex algorithm can achieve the best results under certain conditions but it lacks the robustness shown (especially at lower quantization bits) by the less complex method STEP-LGP:UniAd. The LGP-based algorithms were able to further reduce the communication expenditure compared to the base STEP-GP algorithm. The best behavior in terms of performance and robustness of any of the proposed quantization-based algorithms is achieved for a resolution greater than 8 bits.

The results showed the potential of our proposed methods to achieve a really good accuracy while significantly reducing the communication cost in comparison to the baseline methods Sync:Exact, Sync:UniQuant, and STEP-GP. Even the less complex proposed method STEP-LGP:UniAd is good enough for reducing significantly the communication cost while reaching an acceptable accuracy level with consistent performance. The peak of performance in any of the testing scenarios was achieved by a quantization-based method using orthogonal transformation, either Decoupling or whitening.

## 8. Discussion on convergence behavior

In our recent technical note in [33], we present a convergence analysis for the STEP-GP and LGP algorithms when the querying mechanism is performed comparing the trace of the covariance matrix $\Sigma_i^k(z_i^k)$ to a decaying threshold instead of the maximum element of the diagonal of $\Sigma_i^k(z_i^k)$ as presented in Section 7.3.4. This querying mechanism can be expressed in the following optimization problem for the STEP-GP algorithm:

$$\begin{aligned} \underset{\gamma^k}{\text{minimize}} \quad & \|\gamma^k\|_1 \\ \text{subject to} \quad & \gamma_i^k \in \{0,1\}. \\ & \sum_{i=1}^{n} \left[(1-\gamma_i^k)\text{trace}(\Sigma_i^k(z_i^k))\right] < \psi_t^k, \end{aligned} \tag{18}$$

where $\gamma_i^k$ is the local communication decision variable being 1 if communication is needed and 0 otherwise, $\psi_t^k = \omega(\alpha)^k$, $\alpha \in [0,1]$, and is a positive constant.

**Lemma 3.** *Under the querying mechanism presented in Section 7.3.4, the STEP-GP algorithm converges and does so at a geometric rate.*

**Proof.** The querying mechanism presented in Section 7.3.4 determines if communication is required following

$$\gamma_i^k = \begin{cases} 0, & \text{if } \max\left(\text{diag}\left(\Sigma_i^k(z_i^k)\right)\right) \le \psi_i^k \\ 1, & \text{otherwise,} \end{cases} \tag{19}$$

with local threshold $\psi_i^k = \psi_i^{k_0}(\alpha)^{k-k_0}$.

Since the trace of $\Sigma_i^k(z_i^k)$ is the sum of its diagonal entries, we can establish the following relationship on the constraints presented in (18) and (19)

$$\text{trace}\left(\Sigma_i^k(z_i^k)\right) \le p \max\left(\text{diag}\left(\Sigma_i^k(z_i^k)\right)\right) \le p\psi_i^k.$$

Assuming that the assessment to determine $\gamma_i^k$ for each agent was already made, we take the sum over all agents:

$$\sum_{i=1}^{n} \left[(1-\gamma_i^k)\text{trace}\left(\Sigma_i^k(z_i^k)\right)\right] \le p \sum_{i=1}^{n} \left[(1-\gamma_i^k)\max\left(\text{diag}\left(\Sigma_i^k(z_i^k)\right)\right)\right] \le p \sum_{i=1}^{n} \psi_i^k.$$

The bound imposed on $\sum_{i=1}^{n} \left[(1-\gamma_i^k)\text{trace}(\Sigma_i^k(z_i^k))\right]$ (the same term used in Section 4 in [33]) follows the same form of a constant multiplied by a geometrically decaying term. Since the sum of the maximum variances is bounded by this form of threshold, Theorem 3 and 4 in [33] also apply to the querying mechanism presented in Section 7.3.4. This communication strategy imposes a tighter bound than the one using the trace. □

Section 5 in [33] presents a convergence analysis for the LGP algorithm. Theorems 5 and 6 show the convergence of the LGP algorithm using trace for the communication decision when the coordinator can vary the quantization resolution at each iteration and there is no bound on the value such resolution can take. For the case when the quantization resolution is bounded, we present a discussion in Section 5.4 of [33] where convergence is not concluded but it is shown that the expectation of the ADMM residual is bounded by a decaying bound. We are currently working on the convergence analysis when quantization is present and the querying method presented in this work is used. Those results will be presented in a future work. However, the empirical evidence of the extensive simulations performed suggests that the LGP algorithm converges to an acceptable solution while not dramatically increasing the number of iterations required to reach convergence.

## 9. Conclusion

In this paper, we developed a hybrid approach that combined the Gaussian Process-based learning approach with an adaptive uniform quantization approach to achieve further reduction of the communication cost required in distributed optimization. The resulting quantization error did not follow a Gaussian distribution, so we proposed a new regression algorithm. This algorithm, inspired by GP, resulted in a Linear Minimum Mean Square-error Estimator named LGP-R, which considered the resulting quantization error statistics. Communication was also reduced by refining the uniform quantizer with an orthogonalization process of the quantizer input to handle the inherent correlation of the quantizer's input components, and with dithering to ensure the uncorrelation between the quantizer's introduced noise and the quantizer's input. Simulations of a distributed sharing problem showed that our hybrid approaches significantly decreased total communication cost when compared to baseline methods, being able to find the global solution at even low quantization resolutions.

### CRediT authorship contribution statement

**Aldo Duarte:** Investigation, Software, Writing – original draft, Writing – review & editing. **Truong X. Nghiem:** Conceptualization, Supervision, Validation, Writing – review & editing. **Shuangqing Wei:** Formal analysis, Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix A. Proof of Proposition 1

Define $x = y - \mu_y \sim \mathcal{N}\left(0, \sigma_y^2\right)$. The output of the adaptive uniform quantizer is given by the standard uniform quantizer $\mathbb{Q}_{\mathrm{u}}\left(y; \mu_y, \frac{2c\sigma_y}{2^b}\right)$, which is equivalent to $\mu_y + \mathbb{Q}_{\mathrm{u}}\left(x; 0, \frac{2c\sigma_y}{2^b}\right)$. Using the result presented in [27, Section V-A] on the quantization error of a uniform quantizer on a zero-mean Gaussian random variable, we can derive the above equations of $\mathbb{E}[\epsilon_{\mathbb{Q}}]$ and $\mathbb{E}[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}]$.

The correlation between $y$ and $\epsilon_{\mathbb{Q}}$ is

$$\mathbb{E}[y\epsilon_{\mathbb{Q}}] = \mathbb{E}[(x+\mu_y)\epsilon_{\mathbb{Q}}] = \mathbb{E}[x\epsilon_{\mathbb{Q}}] + \mu_y\mathbb{E}[\epsilon_{\mathbb{Q}}] = \mathbb{E}[x\epsilon_{\mathbb{Q}}].$$

Using the result presented in [27, Section V-B] on the correlation between a zero-mean Gaussian random variable and its uniform quantization error, we have that

$$\mathbb{E}[x\epsilon_{\mathbb{Q}}] = 2\sigma_y \sum_{m=1}^{\infty} (-1)^m \exp\left(-2\pi^2 m^2 r^2\right),$$

which results in the same equation for $\mathbb{E}[y\epsilon_{\mathbb{Q}}]$.

## Appendix B. Proof of Lemmas 1 and 2

We first need the following result.

**Proposition 4.** *For* $r > \frac{1}{\sqrt{2\pi}}$,

$$\sum_{m=1}^{\infty} (-1)^m m^2 \exp\left(-2\pi^2 m^2 r^2\right) < 0.$$

**Proof.** Define $S(m) = m^2 \exp\left(-2\pi^2 m^2 r^2\right)$. Then the series is $\sum_{m=1}^{\infty}(-1)^m S(m)$. We have

$$\begin{aligned}\frac{\mathrm{d}S(m)}{\mathrm{d}m} &= 2m\exp\left(-2\pi^2 m^2 r^2\right) - 4\pi^2 r^2 m^3 \exp\left(-2\pi^2 m^2 r^2\right) \\ &= 2m\exp\left(-2\pi^2 m^2 r^2\right)\left(1 - 2\pi^2 r^2 m^2\right).\end{aligned}$$

For $r > \frac{1}{\sqrt{2\pi}}$ and $m \geq 1$, we have $1 - 2\pi^2 r^2 m^2 < 0$, thus $\frac{\mathrm{d}S(m)}{\mathrm{d}m} < 0$, which implies that $S(m)$ is strictly decreasing with $m$, i.e., $S(1) > S(2) > S(3) > S(4) > \ldots$. Therefore, the series is $\sum_{m=1}^{\infty}(-1)^m S(m) = (-S(1) + S(2)) + (-S(3) + S(4)) + \cdots < 0$. □

We will now prove Lemmas 1 and 2. Consider the series $s(r) = \sum_{m=1}^{\infty} \frac{(-1)^m}{m^2} \exp\left(-2\pi^2 m^2 r^2\right)$ as a function of $r$. Define $s_m(r) = \frac{1}{m^2}\exp\left(-2\pi^2 m^2 r^2\right)$. Then $s(r) = \sum_{m=1}^{\infty}(-1)^m s_m(r)$. For an integer $m \geq 1$, we have that $s_m(r) > s_{m+1}(r)$ because

$$\begin{aligned} s_{m+1}(r) &= \frac{1}{(m+1)^2}\exp\left(-2\pi^2(m+1)^2r^2\right) \\ &< \frac{1}{m^2}\exp\left(-2\pi^2(m+1)^2r^2\right) \\ &= \frac{1}{m^2}\exp\left(-2\pi^2m^2r^2\right)\exp\left(-2\pi^2(2m+1)r^2\right) \\ &< \frac{1}{m^2}\exp\left(-2\pi^2m^2r^2\right) \\ &= s_m(r), \end{aligned}$$

where the last inequality holds due to $\exp\left(-2\pi^2(2m+1)r^2\right) < 1$. Therefore

$$s(r) = (-s_1(r) + s_2(r)) + (-s_3(r) + s_4(r)) + \cdots < 0.$$

Using the same approach, we can show that $\sum_{m=1}^{\infty}(-1)^m \exp\left(-2\pi^2m^2r^2\right) < 0$.

To show that $s(r)$ is increasing with $r$, we differentiate it with respect to $r$:

$$\frac{\mathrm{d}s(r)}{\mathrm{d}r} = -4\pi^2 r\sum_{m=1}^{\infty}(-1)^m \exp\left(-2\pi^2m^2r^2\right)$$

which is positive because we have just shown that $\sum_{m=1}^{\infty}(-1)^m \exp\left(-2\pi^2m^2r^2\right) < 0$. Therefore, $s(r)$ is increasing with $r$.

Similarly, for the series in Lemma 2, we have

$$\frac{\mathrm{d}}{\mathrm{d}r}\sum_{m=1}^{\infty}(-1)^m \exp\left(-2\pi^2m^2r^2\right) = -4\pi^2 r\sum_{m=1}^{\infty}(-1)^m m^2 \exp\left(-2\pi^2m^2r^2\right) > 0$$

for all $r > \frac{1}{\sqrt{2}\pi}$, due to Proposition 4. Therefore, the series $\sum_{m=1}^{\infty}(-1)^m \exp\left(-2\pi^2m^2r^2\right)$ is increasing with $r$ for all $r > \frac{1}{\sqrt{2}\pi}$.

## Appendix C. Proof of Proposition 3

The dequantized value $\hat{y}$ will be $\hat{y} = A^{-1}\mathbb{Q}_{\mathrm{ua}}\left(y^A; 0, \sigma_w, c, b\right) + \mu(x)$, but can be also expressed as

$$\begin{aligned} \hat{y} &= A^{-1}\left[A(y-\mu_y) + \epsilon_{\mathbb{Q}}\right] + \mu_y \\ &= y + A^{-1}\epsilon_{\mathbb{Q}} = y + \hat{\epsilon}_{\mathbb{Q}}. \end{aligned}$$

Analyzing the auto correlation of $\hat{\epsilon}_{\mathbb{Q}}$ we have:

$$\begin{aligned} \mathbb{E}[\hat{\epsilon}_{\mathbb{Q}}\hat{\epsilon}'_{\mathbb{Q}}] &= (A)^{-1}\mathbb{E}[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}]\left((A)^{-1}\right)' \\ &= (A)^{-1}\Lambda_{\epsilon_{\mathbb{Q}}}\left((A)^{-1}\right)', \end{aligned}$$

where $\mathbb{E}[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}]$ is the auto correlation of the quantization error and $\Lambda_{\epsilon_{\mathbb{Q}}}$ is a diagonal matrix with its diagonal given by the vector $\frac{v(2^b/2c)}{12}\tilde{q}^2$, with $v(2^b/2c)$ as defined in Proposition 1.

If $A_1$ is used then $\tilde{q}$ will be $\tilde{q} = \frac{2c}{2^b}I_{p+1} = \Gamma(b,c)I_{p+1}$, where $\Gamma(b,c) = \frac{2c}{2^b}$.

On the other hand, if $A_2$ is used then $\tilde{q} = \frac{2c}{2^b}\sqrt{\Lambda} = \Gamma(b,c)\sqrt{\Lambda}$. Therefore we will have that

$$\begin{aligned} \mathbb{E}[\hat{\epsilon}_{\mathbb{Q}}\hat{\epsilon}'_{\mathbb{Q}}] &= A^{-1}\Lambda_{\epsilon_{\mathbb{Q}}}(A^{-1})' \\ &= \frac{\Gamma^2(b,c)v(2^b/2c)}{12}(A^{-1}\tilde{\Lambda}_{\epsilon_{\mathbb{Q}}}(A^{-1})'), \end{aligned}$$

with $\tilde{\Lambda}_{\epsilon_{\mathbb{Q}}}$ being $I_{p+1}$ or $\Lambda$ depending on the selection of A.

Finally, we have that since $A^{-1}\tilde{\Lambda}_{\epsilon_{\mathbb{Q}}}(A^{-1})' = \Sigma_y$, then no matter the selection of A the result will be

$$\mathbb{E}[\hat{\epsilon}_{\mathbb{Q}}\hat{\epsilon}'_{\mathbb{Q}}] = \frac{\Gamma^2(b,c)v(2^b/2c)}{12}\Sigma_y = \Delta.$$

## Appendix D. Proof of Theorem 1

The proposed LMMSE will be given by the linear combination

$$\mu(x_*) = H\hat{Y}. \tag{D.1}$$

Then, if (D.1) is a LMMSE then it must follow the orthogonal principle which will be given by $\mathbb{E}\left[(\mu(x_*) - \hat{y}_*)(\hat{Y})'\right] = 0$. From this point we can obtain an expression for $H$

$$\mathbb{E}\left[(H\hat{Y} - \hat{y}_*)(\hat{Y})'\right] = 0$$
$$H\mathbb{E}\left[(Y + \epsilon_n + \epsilon_{\mathbb{Q}})(Y + \epsilon_n + \epsilon_{\mathbb{Q}})'\right] = \Phi(x_*, X). \tag{D.2}$$

Since $\epsilon_n$ is independent from the rest, all cross products involving $\epsilon_n$ will be turn to zero by the expectation. Therefore we can simplify the expression to

$$H\left(\Phi(X,X) + \mathbb{E}[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}] + \sigma_n I_{m(p+1)} + 2\mathbb{E}[Y\epsilon'_{\mathbb{Q}}]\right) = \Phi(x_*, X). \tag{D.3}$$

Defining $\mathbb{E}[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}] = \Delta$, we have the expression

$$H = \Phi(x_*, X)\left(\Phi(X,X) + \Delta + \sigma_n I_{m(p+1)} + 2\mathbb{E}[Y\epsilon'_{\mathbb{Q}}]\right)^{-1}. \tag{D.4}$$

The term $\mathbb{E}[Y\epsilon'_{\mathbb{Q}}]$ expresses the correlation between the input of the quantizer and the quantization error. In Proposition 1 a way to calculate this correlation is presented when the input of the quantizer is zero mean. Because we subtract the mean of the input of the quantizer before performing the quantization, we have $\mathbb{E}[\epsilon'_{\mathbb{Q}}] = 0$, following Proposition 1. Thus, the following holds true, $\mathbb{E}[Y\epsilon'_{\mathbb{Q}}] = \mathbb{E}[(Y - \mu(Y))\epsilon'_{\mathbb{Q}}] + \mu(Y)\mathbb{E}[\epsilon'_{\mathbb{Q}}] = \mathbb{E}[(Y - \mu(Y))\epsilon'_{\mathbb{Q}}]$. This means that the results of Proposition 1 can be extended to calculate the elements conforming matrix $\mathbb{E}[Y\epsilon'_{\mathbb{Q}}]$. This is done directly for the diagonal terms that come from the same dimension, for example, $\mathbb{E}[Y_{[1]}\epsilon'_{\mathbb{Q}[1]}]$ where $Y_{[1]}$ and $\epsilon'_{\mathbb{Q}[1]}$ refer to the first element of vectors $Y$ and $\epsilon'_{\mathbb{Q}}$, respectively. In case we want to calculate $\mathbb{E}[Y_{[i]}\epsilon'_{\mathbb{Q}[j]}]$, $i \neq j$, we define $\tilde{Y}_{[i]} = Y_{[i]} - \mu(Y_{[i]})$ and do the following:

$$\begin{aligned}\mathbb{E}\left[Y_{[i]}\epsilon'_{\mathbb{Q}[j]}\right] &= \mathbb{E}\left[\tilde{Y}_{[i]}\epsilon'_{\mathbb{Q}[j]}\right] = \mathbb{E}\left[(\tilde{Y}_{[i]} - \xi_{ij}\tilde{Y}_{[j]} + \xi_{ij}\tilde{Y}_{[j]})\epsilon'_{\mathbb{Q}[j]}\right] \\ &= \mathbb{E}\left[(\tilde{Y}_{[i]} - \xi_{ij}\tilde{Y}_{[j]})\epsilon'_{\mathbb{Q}[j]}\right] + \xi_{ij}\mathbb{E}\left[\tilde{Y}_{[j]}\epsilon'_{\mathbb{Q}[j]}\right],\end{aligned}$$

where $\xi_{ij}\tilde{Y}_{[j]}$ is the MMSE of $\tilde{Y}_{[i]}$ with $\xi_{ij}$ being the operator to estimate $\tilde{Y}_{[i]}$ from $\tilde{Y}_{[j]}$. Since the error of the MMSE is given by $\epsilon_{ij} = \tilde{Y}_{[i]} - \xi_{ij}\tilde{Y}_{[j]}$, then $\epsilon_{ij}$ is independent of $\tilde{Y}_{[j]}$. Therefore,

$$\mathbb{E}\left[(\tilde{Y}_{[i]} - \xi_{ij}\tilde{Y}_{[j]})\epsilon'_{\mathbb{Q}[j]}\right] = \mathbb{E}\left[\tilde{Y}_{[i]} - \xi_{ij}\tilde{Y}_{[j]}\right]\mathbb{E}\left[\epsilon'_{\mathbb{Q}[j]}\right] = 0.$$

Thus,

$$\mathbb{E}\left[Y_{[i]}\epsilon'_{\mathbb{Q}[j]}\right] = \xi_{ij}\mathbb{E}\left[\tilde{Y}_{[j]}\epsilon'_{\mathbb{Q}[j]}\right].$$

Consequently, we can calculate any correlation $\mathbb{E}[Y_{[i]}\epsilon'_{\mathbb{Q}[j]}]$ following the correlation expression presented in Proposition 1.

Finally, the error covariance of the estimator will be given by

$$\Sigma(x_*) = \mathbb{E}\left[(\hat{y}_* - H\hat{Y})(\hat{y}_* - H\hat{Y})^T\right].$$

Expanding this expression and operating the expectations we get

$$\Sigma(x_*) = \Phi(X_*, X_*) - H^T\Phi(X, X_*) - \Phi(X_*, X)H - H^T\Phi(X, X)H. \tag{D.5}$$

Finally, introducing the expression of $H$ in (D.4) we get

$$\Sigma(x_*) = \Phi(X_*, X_*) - \Phi(X_*, X)\left(\Phi(X,X) + \sigma_n^2 I_{m(p+1)} + \Delta + 2\mathbb{E}[Y\epsilon'_{\mathbb{Q}}]\right)^{-1}\Phi(X, X_*).$$

## Appendix E. Proof of Theorem 2

The expression for our estimator will be defined as

$$\bar{y}_* - \mu(x_*) = B\left(\hat{y}_* - \mu(x_*)\right),$$

where $B$ is the matrix determined by resorting to the orthogonal principle. Using the orthogonal principle for this LMMSE like in the LGP case the expression for $B$ will be

$$B\ E\left[(\hat{y}_* - \mu(x_*))(\hat{y}_* - \mu(x_*))'\right] = \mathbb{E}\left[(\hat{y}_* - \mu(x_*))(\hat{y}_* - \mu(x_*))'\right]. \tag{E.1}$$

So, inserting the definition of $\mu(x_*)$ and $\Sigma(x_*)$ from Theorem 1 into (E.1) will lead to the simplified version

$$B = \Sigma(x_*)\left(\Sigma(x_*) + \sigma_n I_{p+1} + \Delta_{p+1} + 2\mathbb{E}[y_*\epsilon'_{\mathbb{Q}*}]\right)^{-1}$$

## Appendix F. Details on the calculation of variables $\theta_i$ and $\Upsilon_i$ in Section 7.1.1

In [11] the variables $\theta_i$ and $\Upsilon_i$ are updated at each iteration of the ADMM algorithm. In this work, those variables are fixed by following the variable's initialization for the first iteration made in [11]. As such, to calculate each $\theta_i$ we first create $\theta_i^0$ which is a p-dimensional vector with entries randomly generated and uniformly distributed on [-1,1]. Then, the value of $\theta_i$ to be used is

$\theta_i = \theta_i^0 + \eta u_i$, where $\eta$ is some small positive number, $u_i$ is a p-dimensional vector for agent $i$ whose entries are randomly generated and uniformly distributed on [-1,1].

Next, to calculate each $\Upsilon_i$ we first create $\Upsilon_i^0 = AA'$ as a symmetric $p \times p$ matrix, where the entries of $A \in \mathbb{R}^{p \times p}$ are randomly generated and uniformly distributed on [-1,1]. Then, we generate $\widetilde{\Upsilon}_i = \Upsilon_i^0 + \eta E_i$, where $E_i$ is a symmetric $p \times p$ matrix whose entries are randomly generated and uniformly distributed on [-1,1]. Subsequently, $\Upsilon_i$ is constructed as

$$\Upsilon_i = \begin{cases} \widetilde{\Upsilon}_i, & \text{if } \lambda_{min}(\widetilde{\Upsilon}_i) > \epsilon \\ \widetilde{\Upsilon}_i + \left(\epsilon - \lambda_{min}(\widetilde{\Upsilon}_i)\right) I_p, & \text{otherwise,} \end{cases}$$

where $\lambda_{min}(\widetilde{\Upsilon}_i)$ denotes the smallest eigenvalue of $\widetilde{\Upsilon}_i$ and $\epsilon > 0$ is some positive constant.

## Appendix G. Details of MAC metric presented in Section 7.3.1

Assuming that the coordinator communicates with the agents wirelessly following the IEEE 802.11 specification, a MAC layer simulator was implemented. The 802.11 CSMA/CA simulator presented in [32] was chosen because of its simplicity, which was modified to our purposes. The simulator implemented in MATLAB will return the number of total transmissions, successful transmissions, and an efficiency value defined by $\xi = st/tt$, where $st$ is the successful transmissions observed and $tt$ the total amount of transmissions performed. The simulation was run offline 1000 times to obtain an average efficiency $\xi$. Once the average values are obtained for different payloads and number of agents, those values will be used with the results given by the distributed optimization simulation to calculate the communication time for each round. In particular, at the $k$-th iteration, the coordinator will receive a certain amount of simultaneous responses which are expressed in the variable $T^k_{\text{simul}}$. The expected transmission time in one iteration round will be $T^k_{\text{round}} = T^k_{\text{simul}}/\xi^*$, where $\xi^*$ is the average efficiency in the MAC simulation for the given scenario. The total transmission time will be $Tx_t = \sum_{k=1}^{N} T^k_{\text{round}}$, where $N$ is the number of iterations taken to reach convergence. This metric is not only affected by the total number of communications that were performed but also the number of agents communicating at each iteration and the payload size, thereby making it a more robust metric to compare the performance of the proposed methods.

## References


[1] T. Yang, et al., A survey of distributed optimization, Annu. Rev. Control 47 (2019) 278–305.
[2] N. Parikh, S. Boyd, Proximal algorithms, Found. Trends Optim. 1 (3) (2014) 127–239.
[3] D. Varagnolo, et al., Newton-Raphson consensus for distributed convex optimization, IEEE Trans. Autom. Control 61 (4) (2016).
[4] A. Gourtani, T.-D. Nguyen, H. Xu, A distributionally robust optimization approach for two-stage facility location problems, EURO J. Comput. Optim. 8 (2) (2020).
[5] P. Dvurechensky, et al., Hyperfast second-order local solvers for efficient statistically preconditioned distributed optimization, EURO J. Comput. Optim. 10 (2022).
[6] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Comput. Math. Appl. 2 (1) (1976) 17–40.
[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. (2011).
[8] T.X. Nghiem, A. Duarte, S. Wei, Learning-based adaptive quantization for communication-efficient distributed optimization with admm, in: 2020 54th Asilomar Conference on Signals, Systems, and Computers, 2020, pp. 37–41.
[9] V. Yfantis, et al., Hierarchical distributed optimization of constraint-coupled convex and mixed-integer programs using approximations of the dual function, EURO J. Comput. Optim. 11 (2023).
[10] S. Kumar, R. Jain, K. Rajawat, Asynchronous optimization over heterogeneous networks via consensus admm, IEEE Trans. Signal Inf. Process. Netw. 3 (1) (2017).
[11] X. Cao, K.J.R. Liu, Dynamic sharing through the ADMM, IEEE Trans. Autom. Control 65 (5) (2020).
[12] D. Du, X. Li, W. Li, R. Chen, M. Fei, L. Wu, Admm-based distributed state estimation of smart grid under data deception and denial of service attacks, IEEE Trans. Syst. Man Cybern. Syst. 49 (8) (2019).
[13] G. Stathopoulos, C.N. Jones, A coordinator-driven communication reduction scheme for distributed optimization using the projected gradient method, in: Proceedings of the 17th IEEE European Control Conference, ECC, Limassol, Cyprus, 2018.
[14] G. Stathopoulos, C. Jones, Communication reduction in distributed optimization via estimation of the proximal operator, arXiv: Optimization and Control, 2018.
[15] T.X. Nghiem, G. Stathopoulos, C. Jones, Learning proximal operators with Gaussian processes, in: Annual Allerton Conference on Communication, Control, and Computing, Illinois, USA, 2018.
[16] C.-X. Shi, G.-H. Yang, Distributed composite optimization over relay-assisted networks, IEEE Trans. Syst. Man Cybern. Syst. 51 (10) (2021) 6587–6598.
[17] Y. Pu, M.N. Zeilinger, C.N. Jones, Quantization design for distributed optimization, IEEE Trans. Autom. Control 62 (5) (May 2017).
[18] T. Doan, S. Maguluri, J. Romberg, Fast convergence rates of distributed subgradient methods with adaptive quantization, IEEE Trans. Autom. Control (08 2020).
[19] P. Groot, P.J. Lucas, Gaussian process regression with censored data using expectation propagation, 2012, pp. 115–122.
[20] G. Bottegal, H. Hjalmarsson, G. Pillonetto, A new kernel-based approach to system identification with quantized output data, Automatica 85 (2017) 145–152.
[21] L.V. Nguyen, G. Hu, C.J. Spanos, Efficient sensor deployments for spatio-temporal environmental monitoring, IEEE Trans. Syst. Man Cybern. Syst. 50 (12) (2020).
[22] X. Wang, On Chebyshev functions and Klee functions, J. Math. Anal. Appl. 368 (1) (2010) 293–310.
[23] D.P. Bertsekas, Convex Optimization Algorithms, Athena Scientific, 2015.
[24] C.E. Rasmussen, C.K. Williams, Gaussian Processes for Machine Learning, vol. 1, MIT Press, Cambridge, 2006.
[25] E. Solak, et al., Derivative observations in Gaussian process models of dynamic systems, in: Advances in Neural Information Processing Systems, 2003, pp. 1057–1064.
[26] A. Grami, Chapter 5 - analog-to-digital conversion, in: A. Grami (Ed.), Introduction to Digital Communications, Academic Press, Boston, 2016, pp. 217–264.
[27] A. Sripad, D. Snyder, A necessary and sufficient condition for quantization errors to be uniform and white, IEEE Trans. Acoust. Speech Signal Process. 25 (5) (1977).
[28] J. Rapp, R.M.A. Dawson, V.K. Goyal, Estimation from quantized Gaussian measurements: when and how to use dither, IEEE Trans. Signal Process. 67 (13) (2019).
[29] R. Hadad, U. Erez, Dithered quantization via orthogonal transformations, IEEE Trans. Signal Process. 64 (2016).
[30] J. Löfberg, YALMIP: a toolbox for modeling and optimization in MATLAB, in: Proc. of the CACSD Conference, Taiwan, 2004.

[31] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, A. Vehtari, GPstuff: Bayesian modeling with Gaussian processes, J. Mach. Learn. Res. 14 (2013).
[32] N.A. Nagendra, Ieee 802.11 mac protocol, https://www.mathworks.com/matlabcentral/fileexchange/44110-ieee-802-11-mac-protocol, 2013.
[33] A. Duarte, T. Nghiem, S. Wei, On the convergence of the structural estimation of proximal operator with Gaussian processes (STEP-GP) method with adaptive quantization for communication-efficient distributed optimization, https://doi.org/10.36227/techrxiv.24593196.v1, 11 2023.