Communication-efficient ADMM using Quantization-aware Gaussian Process Regression

Aldo Duarte Vera Tudela [1], Shuangqing Wei [1], and Truong Nghiem [1]

[1]Affiliation not available

October 30, 2023

Abstract

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, the GP algorithm is modified to account for the introduced quantization noise statistics due to data quantization. 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.

Communication-efficient ADMM using Quantization-aware Gaussian Process Regression

Aldo Duarte[‡], Truong X. Nghiem[§], and Shuangqing Wei[‡]
[‡] Louisiana State University. [§] Northern Arizona University.

***Abstract*—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.**

***Index Terms*—distributed optimization, ADMM, proximal operator, communication reduction, Gaussian Process, quantization**

I. 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 (as listed in [1]) include power systems, sensor networks, smart buildings, and smart manufacturing.

A simple yet powerful algorithm suited to distributed convex optimization, first presented in [2], is the Alternating Direction Method of Multipliers (ADMM). In this algorithm, the optimization is solved by decomposing the global objective problem into smaller local sub-problems. Then, each agent solve its local sub-problem and send its results to the coordinator which combines all the agent's solutions to assemble the global objective. Also, ADMM is relatively easy to implement and, because of its decomposing behavior, it is simple to parallelize. As mentioned in [3], 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* [4], which are regularized versions of the original 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 agents' responses in the previous iteration.

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 distributed optimization solved via query-response schemes. This 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 [5] proposed to solve a distributed optimization problem using ADMM where the proximal operators were predicted by Gaussian Processes (GP) regression, and the communications coming from the agents to the coordinator were quantized. This study faced 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. Such limitations are critical because GP regression is based on an assumption of joint Gaussian distribution in the underlying conditional mean evaluation, exploiting the knowledge of past function values to infer a new sampled value. Since the quantization noise was 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. This can cause an increase in the number of iterations to reach convergence

or potential failure to reach convergence.

In this paper we propose to address the limitations of our previous work [5] and to integrate 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 [5], and characterize its impact on the distributed optimization algorithm.
2) We employ a novel Linear Minimum Mean Square Estimator (LMMSE) based regression which takes in consideration the impact of the quantization error to improve the hybrid communication reduction approach from [5]. We also develop an additional LMMSE to be used only when a communication between coordinator and agent is required. Since the agent's response is quantized, this additional LMMSE approximates the quantized response to the real value calculated by the agent to further mitigate the impact of quantization in the ADMM algorithm.
3) We integrate our adaptive uniform quantizer with orthogonal transformations and dithering. These additions serve the purpose of taking into account the inherent correlation of the elements conforming the quantizer's input and ensuring the un-correlation between the quantization error and the quantizer's input, respectively.
4) We validate our approach and algorithms by running extensive simulations of a distributed network solving a sharing problem with a quadratic cost function. For comparison purposes, we also test three baseline methods using the proposed distributed network: vanilla ADMM, ADMM with uniform quantization, and ADMM with GP. The simulation results show significant reductions in the total communication expenditure in all test cases when compared against the baseline methods, with negligible compromise in the optimization performances.

Paper Organization: This paper begins with the problem formulation in Section III. An overview of our proposed adaptive uniform quantization scheme and new GP algorithms is presented in Section IV. The main mathematical foundation and derivations are presented in Section V. A detailed presentation of our complete proposed approach is shown in Section VI. The simulation results are presented in Section VII. This paper concludes with the main contributions in Section VIII.

II. Related Works

In the context of ADMM solving distributed optimization problems examples include [6] and [7], where ADMM was used to solve consensus and sharing problems, respectively. Also, in [8] ADMM is used with particle swarm optimization (PSO) for task offloading in vehicular networks with hybrid fog/cloud computing. Furthermore in [9], ADMM with proximal operator is used to minimize the fixed-point error in a reinforcement learning problem.

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 [10] were able to reduce communication complexity. The concept of *the Moreau envelope function* is used in [11] and further developed in [12] to predict the proximal operators of the local agents so that certain communication rounds can be skipped. The same concept was used in [13], where the local proximal operators and their gradients were predicted by GP with derivative observations. The GP models of the local proximal operators were updated online and provided the predicted proximal operators at new query points and their prediction uncertainties.

Several works proposed quantization methods to reduce the size of the data exchanged in each algorithmic iteration, resulting in less overall communication overhead. The work in [14] presented a quantized distributed composite optimization problem over relay-assisted networks solved via a simplified augmented Lagrangian method. In [15], the stabilization problem for switched linear systems with quantization and event-triggered control is studied. In [16], a distributed optimization problem affected with quantization was solved using the inexact proximal gradient method. This work also explored the conditions to ensure convergence in this setting. In [17], a distributed optimization problem was solved by a distributed gradient algorithm with an adaptive quantization scheme.

Distributed optimization problems running GP regression where part of the data were censored was previously studied. Authors of [18] explained a GP framework where all data that was outside of a specific range was fixed to a value. Also, in [19] a system identification with quantized output data modeled with GP was presented, where Gibbs sampler was used to estimate the kernel hyperparameters. Finally, in [20] GP was used to predict the best locations for sensors in a spatial environment.

Our work is fundamentally different since it combines the concepts of ADMM, online learning, and quantization that in previous works were studied separately. Also, our work not only put the three concepts together but considers the presence of the quantization error and prediction error to build an approach that do a correct modeling and mitigates the impact of both sources of error.

III. Problem Formulation

This work deals with a multi-agent optimization problem whose structure takes the form of the sharing problem as considered in [3], [7]:

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

Here, n agents, each with local decision variables $x_i \in \mathbb{R}^p$ 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 convex shared global cost function $h : \mathbb{R}^p \mapsto \mathbb{R}$. Each cost function is only known to its corresponding agent, and for privacy reasons, cannot be shared with the coordinator or other agents. The problem is solved with information exchange between only the coordinator and the agents.

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

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

Because agents must 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:

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.

As discussed in Section I, a drawback of the vanilla ADMM is that it often incurs extensive communications between the coordinator and agents. To reduce the communication overhead in this distributed optimization scheme, the authors of [12] proposed an approach called STEP (STructural Estimation of Proximal operator). The concept of the Moreau envelope of a function f underlies the STEP approach. 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

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

When f is a convex function, the Moreau envelope $f^{\frac{1}{\rho}}$ is *convex and differentiable with Lipschitz continuous gradient with constant* ρ. Moreover, the unique solution to the proximal minimization $\mathbf{prox}_{\frac{1}{\rho} f}(z)$ is [21, 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).

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 [13] 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). This approach is named STEP-GP.

The STEP and STEP-GP methods only consider reducing the number of agents communicating simultaneously but do not consider the *payload size* of each transmission. The communication expenditure can be reduced further if the learning component is combined with quantization of the communications between agents and coordinator. Our work [5] presented some preliminary results on a hybrid approach combining learning with quantization for further reducing the communication overhead.

This paper expands on our preliminary results in [5] by the four major contributions listed in Section I. An overview of our proposed approach will be described in the next section.

IV. Proposed Approach Overview

Prior work: The goal of the STEP-GP approach [13] is to learn the Moreau envelope function $f_i^{1/\rho} : \mathbb{R}^{n_i} \mapsto \mathbb{R}$ of each agent i by a GP, called a proxGP, from past queries with the agent. In particular, the coordinator will keep a proxGP for every agent i, which is trained and continuously improved on the query data $\{z_i^k, f_i^{1/\rho}(z_i^k), \nabla f_i^{1/\rho}(z_i^k)\}_k$, where the available derivatives $\nabla f_i^{1/\rho}(z_i^k)$ are incorporated into proxGP training to improve its accuracy [22]. The proxGP is used by the coordinator to predict the gradient $\nabla f_i^{1/\rho}(z_i^k)$ of the agent's Moreau envelope in response to a new query point z_i^k at the current algorithmic iteration k, which has a multivariate Gaussian distribution. The agent's proximal operator is calculated from the predicted gradient following (5). The coordinator then decides, using a heuristic criterion utilizing the predictive covariance matrix of $\nabla f_i^{1/\rho}(z_i^k)$, whether an actual query should be communicated with the agent to obtain its exact response. Our previous work [5] proposed the use of adaptive quantization of the response data to reduce the payload size of each transmission from agent to coordinator. This works to further reduce communication overhead. Our hybrid approach combines the learning-based method of STEP-GP with adaptive quantization approaches that reduce the payload size of each communication packet sent by agents to the coordinator. Each agent also maintains a proxGP identical to the proxGP at the coordinator. The predicted mean and covariance matrix of $\nabla f_i^{1/\rho}(z_i^k)$ for each agent helps to adapt common quantization methods, such as uniform quantization, to enhance their accuracy even at low quantization resolutions.

Current work: We showed in [5], through numerical experiments, that our hybrid approach could achieve significant reduction in transmission time compared with the vanilla ADMM without learning (by up to 99%) and with the original STEP-GP method without quantization (by up to 88%). This paper builds upon our hybrid approach [5] by further analyzing and mitigating the impact of quantization errors. Our improved hybrid approach is depicted in the diagram in Figure 1, which describes the communication and computation processes between the coordinator and an agent i at ADMM iteration k. In

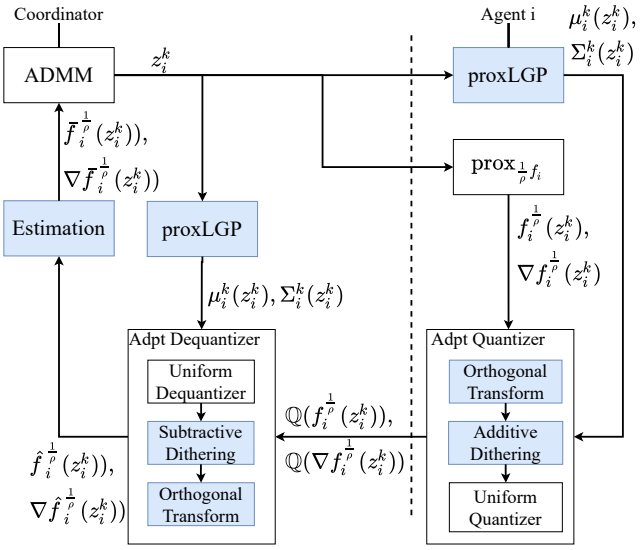

Fig. 1: Flow diagram of a query and response between the coordinator and an agent in the proposed approach.

the colored boxes are new or modified components developed in this work compared to the approach in [5]. These improvements and our overall approach are briefly described below.

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 proxGP (identical to the coordinator's proxGP), 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 $\{f_i^{1/\rho}(z_i^k), \nabla f_i^{1/\rho}(z_i^k)\}$ 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 [16]. The quantized response $\left\{\left(\mathbb{Q}\big(f_i^{1/\rho}(z_i^k)\big), \mathbb{Q}\big(\nabla f_i^{1/\rho}(z_i^k)\big)\right)\right\}$ 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 $\{\hat{f}_i^{1/\rho}(z_i^k), \nabla \hat{f}_i^{1/\rho}(z_i^k)\}$. The dequantized values are used both for the ADMM calculations and for updating the proxGP.

1) Our **first improvement** is related to the regression and update of the proxGP. In our original method [5], the query data were quantized by an adaptive quantization process, but the non-Gaussian quantization errors were not considered during the proxGP prediction and training. This work derived a linear minimum mean square error estimator (LMMSE) approach for GP, named LGP, to address the impact of the quantization errors to improve the accuracy of the regression and update of the proxGP. The "proxLGP" blocks in Figure 1 represent the new method for proxGP regression and update.
2) Our **second improvement** is related to the adaptive quantization and dequantization processes, represented by the "Adaptive Quantizer" and "Adaptive Dequantizer" blocks in Figure 1. The new methods include pre-processing the inputs by orthogonal transformation and dithering. These steps are included to further reduce the impact of the quantization errors.
3) In our original approach [5], the dequantized value $\nabla \hat{f}_i^{1/\rho}(z_i^k)$ was directly used to perform the ADMM update, which was affected by the quantization error. To mitigate this issue, our **third improvement** is an estimation mechanism based on LMMSE for post-processing the dequantized value to generate a new approximation $\nabla \bar{f}_i^{1/\rho}(z_i^k))$ of the true $\nabla f_i^{1/\rho}(z_i^k))$, thereby further reducing the impact of the quantization error. This new process is represented in Figure 1 by the "Estimation" block.

In the next section, we present the theoretical foundation upon which these improvements are developed.

V. Theoretical Foundation

A. Gaussian Process Regression with Derivative Observations

Following the definition given in [23], a GP is a collection of random variables, any finite number of which have a joint Gaussian distribution. Also, it is completely specified by its mean function and co-variance function. The concept of GP is further illustrated in the supplementary file in Figure S2.

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 [23], the GP can be written as $f(x_i) \sim \mathcal{GP}(\mu(x_i), \phi(x_i, x_i'))$.

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 $[f(x_i); \nabla f(x_i)]$, 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. In this scenario, the GP will use values of the function and its gradient to estimate an unknown function value and its gradient. Also, the consideration of derivative components will modify the way the co-variance function is evaluated. Following [22], the covariance matrix is correspondingly expanded, for any pair of points $s, l \in [1, m]$, with the covariances between the observations and its partial derivatives given by,

$$\begin{aligned}\operatorname{Cov}\left[\frac{\partial f(x_s)}{\partial x_s^{(d_s)}}, f(x_l)\right] &= \frac{\partial}{\partial x_s^{(d_s)}} \operatorname{Cov}\left[f(x_s), f(x_l)\right] \\ &= \frac{\partial}{\partial x_s^{(d_s)}} \phi\left(x_s, x_l\right), \end{aligned} \tag{6}$$

and the co-variances between the partial derivatives given by

$$\begin{aligned}\operatorname{Cov}\left[\frac{\partial f(x_s)}{\partial x_s^{(d_s)}}, \frac{\partial f(x_l)}{\partial x_l^{(d_l)}}\right] &= \frac{\partial^2}{\partial x_s^{(d_s)} \partial x_l^{(d_l)}} \operatorname{Cov}\left[f(x_s), f(x_l)\right] \\ &= \frac{\partial^2}{\partial x_s^{(d_s)} \partial x_l^{(d_l)}} \phi\left(x_s, x_l\right) \end{aligned} \tag{7}$$

where $1 \le d_s, d_l \le p$.

In a communication affected by noise, the extended functions at x_i are given by

$$y_i = [f(x_i); \nabla f(x_i)] + \epsilon_{\mathrm{n}},$$

where $\epsilon_{\mathrm{n}} \in \mathbb{R}^{p+1}$ is a vector whose elements are independent identically distributed zero mean Gaussian noise with variance σ_n^2. Redefining the training set of m observations as $\mathcal{D} = (X, Y)$, where $X = [X_1; X_2; \ldots; X_m] \in \mathbb{R}^{m(p+1)\times p}$ with $X_i = [x_i; x_i \ldots; x_i] \in \mathbb{R}^{(p+1)\times p}$, and $Y = [y_1; y_2; \ldots; y_m] \in \mathbb{R}^{m(p+1)\times 1}$, the prior on the noisy observations becomes

$$\mathrm{Cov}(Y) = \Phi(X, X) + \sigma_n^2 I_{m(p+1)},$$

with $I_{m(p+1)}$ being the $m(p+1) \times m(p+1)$ identity matrix. The matrix $\Phi(X, X) \in \mathbb{R}^{m(p+1)\times m(p+1)}$ will have entries given by $E[f(x_s)f(x_l)] = \phi(x_s, x_l)$ where $E[.]$ is the expected value, $E[f(x_s)\nabla f(x_l)]$ following (6), and $E[\nabla f(x_s)\nabla f(x_l)^T]$ following (7).

Given a new input $x_* \in \mathbb{R}^{1\times p}$, we want to predict the extended function value and its gradient depicted by the vector $y_* = [f(x_*); \nabla f(x_*)]$. The predicted value of y_* will be given by the conditional mean $\mu(x_*)$ denoted by

$$\mu(x_*) = \Phi(X_*, X)(\Phi(X, X) + \sigma_n^2 I_{m(p+1)})^{-1} Y \quad (8)$$

where $X_* \in \mathbb{R}^{(p+1)\times p}$ contains a copy of x_* in each of its rows. The matrix $\Phi(X_*, X)$ is given by $E\big[[f(x_*); \nabla f(x_*)][f(x_1), \nabla f(x_1)^T, \ldots, f(x_m), \nabla f(x_m)^T]\big]$, with its entries given by $E[f(x_*)f(x_i)] = \phi(x_*, x_i)$, $E[f(x_*)\nabla f(x_i)]$, or $E[\nabla f(x_*)\nabla f(x_i)^T]$, which will follow (6) and (7) respectively. The uncertainty of such prediction given by the conditional co-variance

$$\Sigma(x_*) = \Phi(X_*, X_*) - \Phi(X_*, X)(\Phi(X, X) + \sigma_n^2 I_{m(p+1)})^{-1}\Phi(X, X_*), \quad (9)$$

where the matrix

$$\Phi(X_*, X_*) = E\big[[f(x_*); \nabla f(x_*)^T][f(x_*), \nabla f(x_*)]\big]$$

will have its entries given by $E[f(x_*)f(x_*)] = \phi(x_*, x_*)$, $E[f(x_*)\nabla f(x_*)]$, or $E[\nabla f(x_*)\nabla f(x_*)^T]$ which will follow (6) and (7), respectively.

B. Adaptive Uniform Quantization

We consider a mid-tread type of uniform quantization [24] where the input-output relation of the quantizer $\mathbb{Q}_{\mathrm{u}}$ is given by

$$\mathbb{Q}_{\mathrm{u}}(y; \bar{y}, q) = \bar{y} + q\left(\left\lfloor \frac{y - \bar{y}}{q} \right\rfloor + \frac{1}{2}\right), \quad (10)$$

in which $q > 0$ is the quantization window length, $\bar{y}$ is the mid-value, and $\lfloor y \rfloor$ 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. Defining $\hat{y} = \mathbb{Q}_{\mathrm{u}}(y; \bar{y}, q)$, 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 the following result [25].

Lemma 1 ([25]): If the input y of a uniform quantizer defined in (10) *follows a Gaussian distribution, $y \sim \mathcal{N}(\mu_y, \sigma_y^2)$, then the probability density of the quantization error $f_{\epsilon_{\mathbb{Q}}}(\epsilon)$ is given by: $f_{\epsilon_{\mathbb{Q}}}(\epsilon) = \frac{1}{q}\left(1 + \sum_{i=1}^{\infty} \cos(\frac{2\pi i\epsilon}{q}) \exp\left(-\frac{2\pi^2 i^2 \sigma_y^2}{q^2}\right)\right)$ if $-q/2 \le \epsilon \le q/2$, and $f_{\epsilon_{\mathbb{Q}}}(\epsilon) = 0$ otherwise. The mean of the quantization error is $E[\epsilon_{\mathbb{Q}}] = 0$ and its variance given by*

$$\mathrm{Var}(\epsilon_{\mathbb{Q}}) = \frac{q^2}{12}\left(1 + \frac{12}{\pi^2}\sum_{i=1}^{\infty}\frac{(-1)^i}{i^2}\exp\left(-\frac{2\pi^2 i^2 \sigma_y^2}{q^2}\right)\right) \quad (11)$$

Remark 1: It can be seen in Lemma 1 that as σ_y/q increases, the discrepancy between the uniform noise model $(1/q)$ and the actual quantization noise reduces as far as the first-order statistics are concerned. As seen in [25], if $\sigma_y > q$, then the quantization error ϵ_q approximately follows a uniform distribution given by

$$\epsilon_{\mathbb{Q}} \sim \mathcal{U}[-q/2, q/2]. \quad (12)$$

Furthermore as shown in [25], when $\sigma_y/q \ge 1$, the correlation between the quantizer's input and the quantization error becomes negligible. The results presented in [25] assumed an infinite number of quantization levels.

In the next subsection, we present our proposed adaptation for the uniform quantizer which will adapt its mid-value and windows length so the conditions presented in Lemma 1 and its subsequent Remark can be satisfied.

1) Proposed Uniform Quantization Adaptation: We propose a quantizer which adapts the standard (non-adaptive) uniform quantizer. Given the quantizer's input $y = f(x)$ being 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 $c > 0$. The proposed quantizer (denoted by $\mathbb{Q}_{\mathrm{ua}}(y; \mu_y, \sigma_y, c, b)$ has parameters that are adapted for a quantization resolution appropriate for the most likely values of $f(x)$. The proposed quantization adaptation is further illustrated in the supplementary file in Figure S1.

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 $\mu(x)$ and conditional co-variance matrix $\Sigma(x)$. The previously presented adaptive quantization scheme must be adjusted to handle the multi-dimensional 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.

a) Adaptive Scheme Ignoring Correlation: Quantization is performed element-wise, using the corresponding elements of with its corresponding elements of the conditional mean vector $\mu(x)$ and the diagonal of the co-variance matrix $\Sigma(x)$ for adaptation. Therefore, we have a vector of window lengths q with the i^{th} entry given by

$$q_i = \frac{2c\sqrt{\Sigma(x)_{ii}}}{2^b} \quad (13)$$

where $\Sigma(x)_{ii}$ is the i^{th} entry of the diagonal of $\Sigma(x)$.

Condition 1: The quantizer $\mathbb{Q}_{\mathrm{ua}}(y; \mu(x), \Sigma(x), c, b)$ has its parameters set such that $\sigma_y > q$ so the quantization error distribution can be approximated and the correlation between such error and the quantizer's input becomes negible as presented in Remark 1.

Under Condition 1, the quantization error $\epsilon_{\mathbb{Q}}$ can be approximated to follow a uniform distribution. This leads to the following proposition.

Proposition 1: Under the Adaptive Scheme Ignoring Correlation and Condition 1, an adaptive uniform quantizer $\mathbb{Q}_{\mathrm{ua}}(y; \mu(x), \Sigma(x), c, b)$ will have a quantization error vector $\epsilon_{\mathbb{Q}}$ whose components are assumed to be uncorrelated. This will

lead to a correlation expression, defined as $\Delta_{\text{un}} = E[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}]$, *being a diagonal matrix with diagonal* $\Delta_{\text{un}(ii)} = \frac{q_i^2}{12}$, *with the entries of vector* q *as defined in* (13).

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

Following the same notation as in the previous scheme, the orthogonal transformation to the quantizer's input is expressed as

$$y^A = A(y - \mu(x)) \tag{14}$$

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* (14) *is obtained by applying an eigenvalue decomposition of matrix* $\Sigma(x)$, *in which* $\Sigma(x) = U\Lambda U'$, *with* Λ *being a diagonal matrix with the eigenvalues of* $\Sigma(x)$ *and* U *being a square matrix whose columns are eigenvectors of* $\Sigma(x)$. *The matrix* A *can be expressed in two ways;* $A_1 = (\Sigma(x))^{-1/2}$ *or* $A_2 = U'$, *where* $(\Sigma(x))^{1/2}$ *is a matrix such that* $(\Sigma(x))^{1/2}(\Sigma(x))^{1/2} = \Sigma(x)$. *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* Λ.

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

$$\mathbb{Q}_{\text{ua}}(y^A; 0, \Sigma_w, c, b) = y^A + \epsilon_{\mathbb{Q}} \tag{15}$$

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

Proposition 2: Under the Correlated Quantization Scheme, Condition 1, and the proposed Pre-filtering, an adaptive uniform quantizer $\mathbb{Q}_{\text{ua}}(y^A; \mu(x), \Sigma(x), c, b)$, *where the input vector is transformed following* (14), *have a quantization error vector* $\epsilon_{\mathbb{Q}}$ *whose components are correlated with each other. This will lead to a correlation expression, defined as* $\Delta\text{co} = E[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}]$, *which is independent of the choice of transformation matrix* A *and is given by* $\Delta\text{co} = \frac{c^2}{3(2^b)^2}\Sigma(x)$.

Proof: The proof is presented in the supplementary file in Section II-A.

C. LMMSE Regression with Quantization

In this subsection, we consider a GP regression as presented in Section V-A, 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 $\hat{y}_i = [\mathbb{Q}_{\text{u}}(f(x_i)); \mathbb{Q}_{\text{u}}(\nabla f(x_i))^T] + \epsilon^i_{\text{n}}$, also expressed as

$$\hat{y}_i = [f(x_i); \nabla f(x_i)^T] + \epsilon^i_{\text{n}} + \epsilon^i_{\mathbb{Q}} \tag{16}$$

where $\epsilon^i_{\mathbb{Q}}$ refers to the quantization error vector for the observation i and ϵ^i_{n} is a vector whose entries follow the same Gaussian distribution with zero mean, σ_n^2 variance at observation i. Such Gaussian noise is not a physical noise but an artificial one added to avoid possible matrix singularity.

The added non-Gaussian quantization noise invalidates the Gaussian noise assumption of the GP regression expressed in (8). In this case, the regression cannot be a Minimum Mean Square 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 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 $\mathcal{D}$.

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. This LMMSE is constructed under the following condition.

Condition 2: The 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* $X \in \mathbb{R}^{m(p+1)\times p}$ *and* $\hat{Y} \in \mathbb{R}^{m(p+1)\times 1}$. *Such training set will be used to estimate the corresponding extended values at* x_* *given by* y_*. *The quantizer used over the elements of* $\hat{Y}$ *is performed element-wise and fulfills the conditions presented in Remark 1.*

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 result.

Theorem 1: Following Condition 2 the LGP-R Estimator has its predicted mean vector

$$\mu(x_*) = \Phi(X_*, X)(\Phi(X, X) + \sigma_n^2 I_{m(p+1)} + \Delta)^{-1}\hat{Y}$$

and predicted covariance matrix

$$\Sigma(x_*) = \Phi(X_*, X_*) - \Phi(X_*, X)(\Phi(X, X) + \sigma_n^2 I_{m(p+1)} + \Delta)^{-1}\Phi(X, X_*)$$

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 V-A,* $\Delta = E[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}]$ *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* Δ *are added block-wise following the expression given by* Δ_{un} *in Proposition 1 or* Δ_{co} *in Proposition 2 (depending on the quantization scheme selected).*

Proof: The proof is presented in the supplementary file in Section II-B.

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 was adapted using 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 which relies on the following condition.

Condition 3: The estimator has an input $x_* \in \mathbb{R}^p$, *a training set containing* m *past observations, and its extended function values follow a zero-mean multivariate Gaussian Distribution. Also, it has a training set containing past observations and the quantized extended values of* x_* *leading to the set* $\mathcal{D} = ([X; x_*], [\hat{Y}; \hat{y}_*])$, *with* $X \in \mathbb{R}^{m(p+1)\times p}$ *and*

$\hat{Y} \in \mathbb{R}^{m(p+1)\times 1}$. *Such training set will be used to estimate the corresponding function values and its gradient at x_* given by y_*. The quantization over the elements of $\hat{Y}$ and $\hat{y}_*$ is performed element-wise and the quantizer fulfills Condition 1.* 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 result.

Theorem 2: Following Condition 3 LGP-A estimates the target value y_* by

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

where $B = \Sigma(x_*)(\Sigma(x_*) + \Delta_{p+1} + \sigma_n I_{p+1})^{-1}$, with $\mu(x_*)$ and $\Sigma(x_*)$ as presented in Theorem 1 and Δ_{p+1} is given by Δ_{un} in Proposition 1 or Δ_{co} in Proposition 2 depending on the quantization scheme selected.

Proof: The proof is presented in the supplementary file in Section II-C.

VI. Proposed Approach Refined

A. Proposed Adaptive Uniform Quantization Scheme

This section combines the overview presented in Section IV with the mathematical derivations presented in Section V to present our complete proposed approach in more detail.

In Figure 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) (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 V-B as the exact values follow a Gaussian distribution (under the LGP model). We will denote the quantized values of the query response as $[\hat{f}_i^{1/\rho}(z_i^k); \nabla \hat{f}_i^{1/\rho}(z_i^k)] = \mathbb{Q}_{\mathrm{ua}}([f_i^{1/\rho}(z_i^k); \nabla f_i^{1/\rho}(z_i^k)]; \mu_i^k(z_i^k), \Sigma_i^k(z_i^k), c, b))$. 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.

B. LGP-R based Regression in our Proposed Approach

The 'proxLGP' block on the coordinator side of Figure 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 $\mathbb{Q}_{\mathrm{ua}}$ (defined in Section V-B) and following (13) , if c is chosen such that $2c < 2^b$, for any element r of the quantizer input the condition $\sqrt{\Sigma_{i[rr]}^k(z_i^k)} > q_{i[r]}^k$ of Remark 1 is met. The condition for the correlation between the quantizer's input and the quantization error to be negligible from Remark 1, $\sqrt{\Sigma_{i[rr]}^k(z_i^k)}/q_{i[r]}^k \geq 1$, is fulfilled when b is large enough. We assume the same parameters for the next derivation, which will result in the adaptive quantizer $\mathbb{Q}_{\mathrm{ua}}$ satisfying the conditions in Remark 1. In that case, Condition 1 for $\mathbb{Q}_{\mathrm{ua}}$ holds then Condition 2 will also hold. 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. Now, defining $g_i^{1/\rho}(z_i^k) = [f_i^{1/\rho}(z_i^k); \nabla f_i^{1/\rho}(z_i^k)]$ 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

$$\mu_i^k(z_i^k) = \Phi(Z_{i*}^k, Z_i^k)(\Phi(Z_i^k, Z_i^k) + \sigma_n^2 I_{m(p+1)} + \Delta_i)^{-1}\hat{G}_i^k \tag{17}$$

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 $\{\mathbb{Q}_{\mathrm{ua}}(g_i^{1/\rho}(z_i^j); \mu_i^j(z_i^j), \Sigma_i^j(z_i^j), c, b)\}_{j\in\mathcal{J}_i}$, $\sigma_n^2 I_{m(p+1)}$, Δ_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 V-A with a covariance function given by the square exponential kernel function.

Following the same notation, we have that the covariance matrix given by the LGP-R will be

$$\begin{aligned}\Sigma_i^k(z_i^k) = \ & \Phi(Z_{i*}^k, Z_{i*}^k) - \\ & \Phi(Z_{i*}^k, Z_i^k)(\Phi(Z_i^k, Z_i^k) + \sigma_n^2 I_{m(p+1)} + \Delta_i)^{-1}\Phi(Z_i^k, Z_{i*}^k)\end{aligned} \tag{18}$$

The matrix Δ_i will be updated block-wise by inserting the corresponding quantization error covariance matrix of the query round, which follows Proposition 1 or Proposition 2 depending the quantization scheme used. Henceforth, we will use Δ_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 Δ_i.

C. LGP-A Aproximation in our Proposed Approach

In Figure 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 VI-B 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, we have the quantized value of $g_i^{1/\rho}(z_i^k)$, $\hat{g}_i^{1/\rho}(z_i^k)$, sent from the agent, and Δ_i has been updated with the block Δ_i^k. Hence, Condition 3 holds. Therefore, we can obtain the desired approximation $\bar{g}_i^{1/\rho}(z_i^k)$ following the derivation from Theorem 2, which gives us

$$\bar{g}_i^{1/\rho}(z_i^k) = (B_i^k(\hat{g}_i^{1/\rho}(z_i^k) - \mu_i^k(z_i^k)))) + \mu_i^k(z_i^k) \tag{19}$$

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

D. Dithering

From Remark 1 and (13), 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 ([26]) explores the use of quantization with dithering to determine which distribution the substractive dithering follows. The work presented in [27] 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 $d_{i[r]}^k \sim \mathcal{U}(\frac{-q_{i[r]}^k}{2}, \frac{q_{i[r]}^k}{2})$, where the window length $q_{i[r]}^k$ is as defined in (13). 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 V-B2, the quantizer input with dithering is given by

$$g_i^{A[d]}(z_i^k) = g_i^A(z_i^k) + d_i^k \tag{20}$$

where $g_i^A(z_i^k) = A(g_i^{1/\rho}(z_i^k) - \mu_i^k(z_i^k))$, 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 subtract the noise added to the input before adding back its mean. The value $\hat{g}_i^{1/\rho}(z_i^k)$ is given by

$$\hat{g}_i^{1/\rho}(z_i^k) = A^{-1}((g_i^{A[d]}(z_i^k) + \epsilon_{\mathbb{Q}i}^k - d_i^k) + \mu_i^k(z_i^k) \tag{21}$$

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

In the next section, we will test different ways of integrating learning and adaptive quantization methods in a numerical example.

VII. Numerical Simulations

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

A. Sharing Problem

1) Problem Definition: Our testing problem is based on the application presented in [7]. 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:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n}(x_i-\theta_i)^T\Upsilon_i(x_i-\theta_i) + \zeta\|\sum_{i=1}^{n} y_i\|_1 \\ \text{subject to} \quad & x_i - y_i = 0 \end{aligned} \tag{22}$$

where for $i = 1, \cdots, n$, variables $x_i, y_i \in \mathbb{R}^p$, $\theta_i \in \mathbb{R}^p$, $\Upsilon_i \in \mathbb{R}^{p\times p}$ positive definite, and $\zeta > 0$ are given problem parameters.

As presented in [7], the problem in (22) 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, $i = 1, 2, \ldots, n$.

2) Generation of Variables θ_i and Υ_i: The details are presented in Section III-A of the supplementary file.

3) Solution with ADMM: The problem presented in (22) has the same form as (2) in Section III based on which the ADMM updates for this case are expressed as

$$\begin{aligned} x_i^{k+1} &= \underset{x_i\in\mathbb{R}^p}{\arg\min}\left\{f_i(x_i) + (\rho/2)\|x_i - z_i^k\|_2^2\right\} \\ \bar{y}^{k+1} &= \underset{\bar{y}\in\mathbb{R}^p}{\arg\min}\left\{\zeta\|n\bar{y}\|_1 + (n\rho/2)\|\bar{y} - \bar{x}^{k+1} - (1/\rho)\lambda^k\|_2^2\right\} \\ \lambda^{k+1} &= \lambda^k + \rho(\bar{x}^{k+1} - \bar{y}^{k+1}) \end{aligned} \tag{23}$$

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 (23), and setting them equal to zero. Following this, x_i^{k+1} can be expressed by the closed form solution

$$x_i^{k+1} = (2\Upsilon_i + \rho I_p)^{-1}(2\Upsilon_i\theta_i + \rho(x_i^k - \bar{x}^k + \bar{y}^k) - \lambda^k), \tag{24}$$

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

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

$$\bar{y}^{k+1} = \begin{cases} (\bar{x}^{k+1} + \lambda^k/\rho) - \frac{\zeta}{\rho}, & \text{if } \bar{x}^{k+1} + \lambda^k/\rho > \frac{\zeta}{\rho} \\ 0, & \text{if } |\bar{x}^{k+1} + \lambda^k/\rho| \le \frac{\zeta}{\rho} \\ (\bar{x}^{k+1} + \lambda^k/\rho) + \frac{\zeta}{\rho}, & \text{if } \bar{x}^{k+1} + \lambda^k/\rho < -\frac{\zeta}{\rho} \end{cases} \tag{25}$$

B. Simulation Implementation

We consider two cases where $n \in \{10, 30\}$. The problem described in (22) 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 (23), which simplifies to (24) and (25) with $\rho = 10$.
3) *STEP-GP*: the algorithm proposed in [13], which combines 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 VI-B, the LMMSE approximation presented in Section VI-C, and the adaptive quantization method developed in Section VI-A.

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 [16] 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

TABLE I: Elements associated with each of the proposed methods.

	GP Regression	LGP Regression	Adpt Uni Quant	Decoupling	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		✓	✓		✓	✓

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 VI-A and performed element-wise following the *Uncorrelated Adaptive Scheme* as presented in Section V-B2a.
- *UniAd-Dec*: this is the adaptive uniform quantization method as presented in Section VI-A and following the *Correlated Quantization Scheme* as presented in Section V-B2b with decoupling.
- *UniAd-DecDit*: same as *UniAd-Dec* but adding the dithering procedure as presented in Section VI-D.
- *UniAd-Whit*: this is the adaptive uniform quantization method as presented in Section VI-A and following the *Correlated Quantization Scheme* with whitening.
- *UniAd-WhitDit*: same as *UniAd-Whit* but adding the dithering procedure as presented in Section VI-D.

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*. The algorithmic components of each of the proposed combinations are summarized in Table I.

The simulations were implemented in MATLAB. The solution of the minimization problems (22) are obtained directly using a convex solver from the YALMIP toolbox [28]. For the regression training and inference, we use the GPstuff toolbox [29]. The computation was conducted with high performance computational resources provided by Louisiana State University (http://www.hpc.lsu.edu).

C. Metrics and Considerations

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 [30], 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_{round} is the expected transmission time in one iteration round. The specifics of how this metric was obtained are presented in Section III-B of the supplementary file.

2) ADMM Termination Criterion: We propose a termination criterion for ADMM using the concept of primal-residual as shown in [3], having the form:

$$\|x^k - y^k\|_\infty \leq \epsilon_p(1 + \|\lambda^k/\rho\|_\infty) \tag{26}$$

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

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

$$LOT = -\log(|J_{gt} - J_*|/J_{gt})/Tx_t \tag{27}$$

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 VII-C1. 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.

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 $\max\left(\text{Var}\left(\nabla f_i^{1/\rho}(z_i^k)\right)\right) > \left(\psi_i^k\right)^2$ then communication is needed, otherwise it is not. The threshold ψ_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 α, 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 $\psi_i^{k_0} = \iota \max\left(\text{Var}\left(\nabla f_i^{1/\rho}(z_i^k)\right)\right)$, where $0 < \iota < 1$. At iteration $k > k_0$, no matter the communication decision made by agent i, the threshold will be updated as $\psi_i^k = \psi_i^{k_0}(\alpha)^{k-k_0}$.

D. 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 ι for the querying mechanism described in Section VII-C4 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 θ_i and Υ_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 $\epsilon_p = 10^{-6}$, and $x_i^0 = \bar{z}^0 = \lambda^0 = 0$.

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 VII-C3 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 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.

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 outperformed 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.

E. 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 Figure S3 in the supplementary file.

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 VII-C3 through the various quantization resolutions tested. The minimum resolution 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 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 Figure S3 (left) of the supplementary file. 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 significant better performance compared against 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.

2) Results for 30 agents: Also, we generated the results for 30 agents following the same procedure as in the previous subsection.

In Figure S3 (right) of the supplementary file we can see that the performance in this case was similar than 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 beated STEP-GP. Different to 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 methods, however they still attained the best performance for the 7 bits case.

F. Tuning

The presented results apart from the parameters explicitly mentioned that change upon cases, commonly share most of the initial parameters. Such approach is fair for comparison purposes, however we believe that a proper tuning of the initial parameters could lead to better results. We acknowledge that real life applications do not allow tuning parameters extensively, nonetheless we wanted to show what happens when tuning one of the initial parameters is allowed. In this subsection we varied the initial querying threshold by tuning the variable ι, which for all the previous simulation was set to 0.6. For the 10 agents case with $p = 5$, we ran simulations varying ι in the interval between 1 and 0.4. For each trial, we used 5 different sets of randomly generated θ_i and Υ_i, and for each set we ran simulations for Sync:Exact, STEP-GP, STEP-LGP:UniAd, STEP-LGP:UniAd-Dec, and STEP-LGP:UniAd-DecDit.

As shown in Fig. 2 (left), the median values for the LOT metric at 7 and 8 bits were better for STEP-LGP:UniAd compared to STEP-LGP:UniAd-Dec and STEP-LGP:UniAd-DecDit. When compared to STEP-GP, STEP-LGP:UniAd presented higher LOT results for 7 and 8 bits while STEP-LGP:UniAd-Dec and STEP-LGP:UniAd-DecDit had better performance only in the 8 bits case. In Table II we present

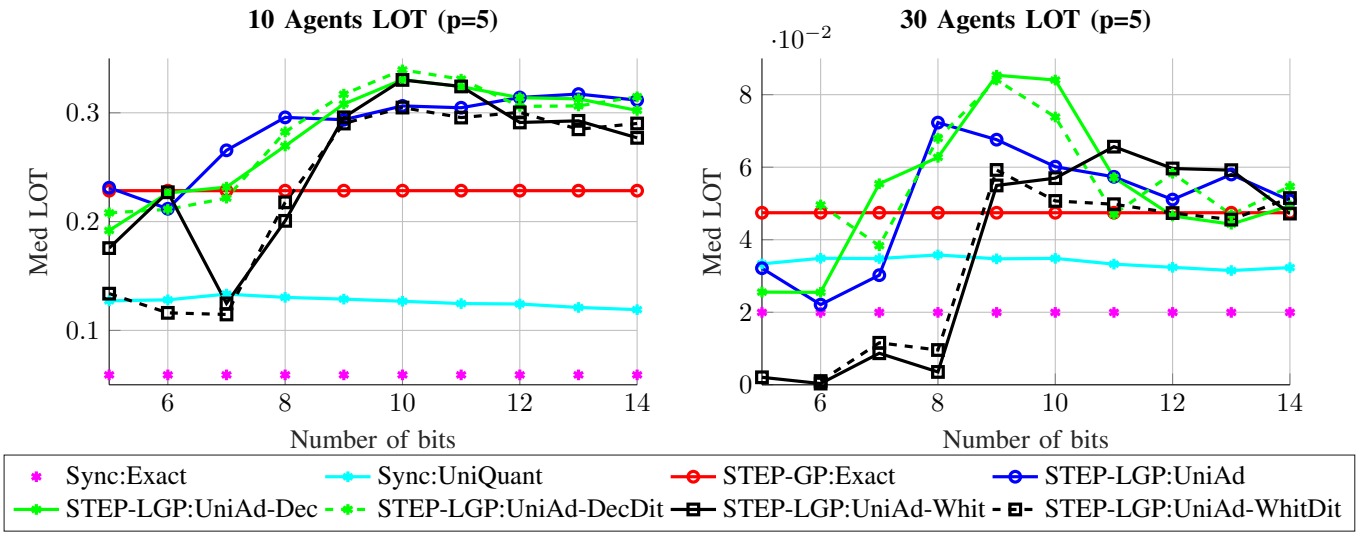

Fig. 2: Performance in the LOT metric of the adaptive quantization methods at different bit resolutions for 10 agents (left) and 30 agents (right) with $p = 5$. The plots show the median LOT of 100 simulations for different sets of parameters θ_i and Υ_i.

the best value of LOT that we achieved for all the ι used for quantization levels fixed at 7 and 8 bits. For the 7 bits case, contrary to the results in Fig. 2 (left), we can see that STEP-LGP:UniAd and STEP-LGP:UniAd-Dec beated STEP-GP in 3 of the 5 cases considered, while STEP-LGP:UniAd-DecDit did the same in 4 cases. Only by comparing between the quantization-based algorithms we got that STEP-LGP:UniAd did not achieve the highest LOT value in any case, STEP-LGP:UniAd-Dec was the best for sets 3 and 4, and STEP-LGP:UniAd-DecDit got the best results for sets 1, 2, and 5.

For the cases following a quantization level of 8, the quantization-based methods always outperformed STEP-GP with the only exception being STEP-LGP:UniAd-Dec for Set 3. In this case, STEP-LGP:UniAd only outperformed the other quantization-based algorithm in Set 2 and 4, STEP-LGP:UniAd-Dec did not achieved the highest LOT value in any case, and STEP-LGP:UniAd-DecDit got the best results for sets 1, 3, and 5.

As a general observation, excluding the Set 5, all the methods that had their initial threshold tuned presented a higher value of LOT than its corresponding median value presented in Fig. 2 (left). The results presented in Table II show the potential of fine tuning initial parameters.

G. 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. Furthermore, a proper tuning of the initial parameters can significantly improve the overall performance of the tested algorithms in terms of the trade-off between communication reduction and accuracy. 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 a 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. However, as shown in the previous subsection the elements that are part of the nature of the problem affect the relative performance of the proposed methods. For that reason, further research needs to be done to determine under what conditions each of the proposed methods achieve the best performance.

VIII. 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 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.

TABLE II: Best result obtained for 10 agents with $p = 5$ in terms of the LOT metric from simulations varying the the initial querying threshold by tuning the value of ι in the interval $[0.4, 1]$. Such tuning was performed for the STEP-GP, STEP-LGP:UniAd, STEP-LGP:UniAd-Dec, and STEP-LGP:UniAd-DecDit methods for fixed quantization resolutions of 7 and 8 bits.

	Quantization resolution $b = 7$					Quantization resolution $b = 8$				
	Set 1	Set 2	Set 3	Set 4	Set 5	Set 1	Set 2	Set 3	Set 4	Set 5
Sync:Exact	0.12077	0.11357	0.11704	0.05357	0.01589	0.12077	0.11357	0.11704	0.05357	0.01589
STEP-GP	0.46526	0.41836	0.50458	0.23658	0.07632	0.46526	0.41836	0.50458	0.23658	0.07632
STEP-LGP:UniAd	0.52190	0.39249	0.41624	0.29317	0.10001	0.48892	0.53054	0.51621	0.32744	0.11247
STEP-LGP:UniAd-Dec	0.51851	0.36936	0.42839	0.29719	0.09726	0.50445	0.48677	0.49199	0.32325	0.11474
STEP-LGP:UniAd-DecDit	0.54340	0.46426	0.38651	0.29297	0.10678	0.54909	0.48731	0.53442	0.30638	0.11748

REFERENCES

[1] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.

[2] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, 2011.

[4] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[5] T. X. Nghiem, A. Duarte, and S. Wei, "Learning-based Adaptive Quantization for Communication-efficient Distributed Optimization with ADMM," in *Annual Asilomar Conference on Signals, Systems, and Computers*, California, USA, Nov. 2020.

[6] S. Kumar, R. Jain, and K. Rajawat, "Asynchronous optimization over heterogeneous networks via consensus admm," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 1, pp. 114–129, 2017.

[7] X. Cao and K. J. R. Liu, "Dynamic sharing through the ADMM," *IEEE Transactions on Automatic Control*, vol. 65, no. 5, pp. 2215–2222, 2020.

[8] Z. Liu, P. Dai, H. Xing, Z. Yu, and W. Zhang, "A distributed algorithm for task offloading in vehicular networks with hybrid fog/cloud computing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–14, 2021.

[9] T. Song, D. Li, Q. Jin, and K. Hirasawa, "Sparse proximal reinforcement learning via nested optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4020–4032, 2020.

[10] D. Du, X. Li, W. Li, R. Chen, M. Fei, and L. Wu, "Admm-based distributed state estimation of smart grid under data deception and denial of service attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 8, pp. 1698–1711, 2019.

[11] G. Stathopoulos and 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 2018, Limassol, Cyprus*, 2018.

[12] G. Stathopoulos and C. Jones, "Communication reduction in distributed optimization via estimation of the proximal operator," *arXiv preprint arXiv:1803.07143*, 03 2018.

[13] T. X. Nghiem, G. Stathopoulos, and C. Jones, "Learning Proximal Operators with Gaussian Processes," in *Annual Allerton Conference on Communication, Control, and Computing*, Illinois, USA, Oct. 2018.

[14] C.-X. Shi and G.-H. Yang, "Distributed composite optimization over relay-assisted networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 10, pp. 6587–6598, 2021.

[15] R. Zhao, Z. Zuo, and Y. Wang, "Event-triggered control for networked switched systems with quantization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–9, 2022.

[16] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Quantization Design for Distributed Optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2107–2120, May 2017.

[17] T. T. Doan, S. T. Maguluri, and J. Romberg, "Fast Convergence Rates of Distributed Subgradient Methods with Adaptive Quantization," *arXiv:1810.13245 [math]*, Oct. 2018.

[18] P. Groot and P. J. Lucas, "Gaussian process regression with censored data using expectation propagation," 01 2012, pp. 115–122.

[19] G. Bottegal, H. Hjalmarsson, and G. Pillonetto, "A new kernel-based approach to system identification with quantized output data," *Automatica*, vol. 85, pp. 145–152, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109817303989

[20] L. V. Nguyen, G. Hu, and C. J. Spanos, "Efficient sensor deployments for spatio-temporal environmental monitoring," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 12, pp. 5306–5316, 2020.

[21] D. P. Bertsekas, *Convex Optimization Algorithms*. Athena Scientific, 2015.

[22] E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, and C. E. Rasmussen, "Derivative observations in gaussian process models of dynamic systems," in *Advances in neural information processing systems*, 2003, pp. 1057–1064.

[23] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, 2006, vol. 1.

[24] A. Grami, "Chapter 5 - analog-to-digital conversion," in *Introduction to Digital Communications*, A. Grami, Ed. Boston: Academic Press, 2016, pp. 217 – 264.

[25] A. Sripad and D. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 5, pp. 442–448, October 1977.

[26] J. Rapp, R. M. A. Dawson, and V. K. Goyal, "Estimation from quantized gaussian measurements: When and how to use dither," *IEEE Transactions on Signal Processing*, vol. 67, no. 13, pp. 3424–3438, 2019.

[27] R. Hadad and U. Erez, "Dithered quantization via orthogonal transformations," *IEEE Transactions on Signal Processing*, vol. 64, no. 22, pp. 5887–5900, 2016.

[28] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. of the CACSD Conference*, Taipei, Taiwan, 2004.

[29] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari, "GPstuff: Bayesian modeling with gaussian processes," *Journal of Machine Learning Research*, vol. 14, pp. 1175–1179, 2013.

[30] N. A. NAGENDRA. (2013) Ieee 802.11 mac protocol. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/44110-ieee-802-11-mac-protocol

Supplementary Material of "Communication-efficient ADMM using Quantization-Aware Gaussian Process Regression"

Aldo Duarte[‡], Truong X. Nghiem[§], and Shuangqing Wei[‡]
[‡] Louisiana State University. [§] Northern Arizona University.

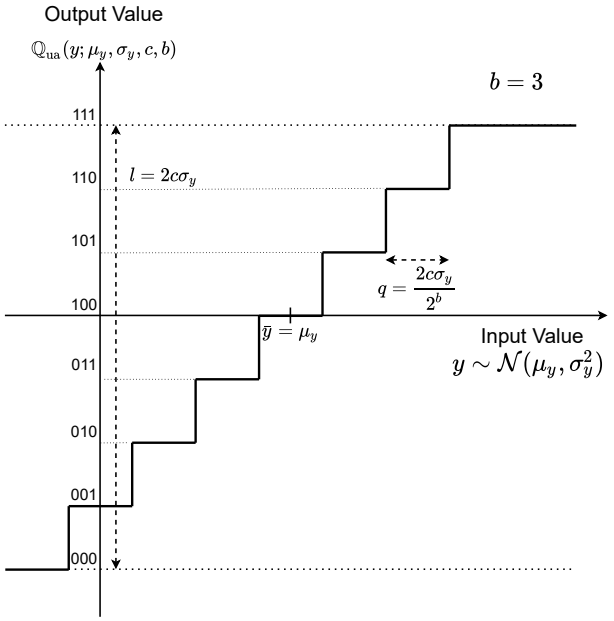

Fig. S1: Diagram representing the proposed adaptation of the uniform quantizer using the statistics of the Gaussian input y. The uniform quantizer's mid-point is set as the mean of y and the range depends on the variance of y.

In this attachment we present supplementary information to the one presented in the main document "Communication-efficient ADMM using Quantization-Aware Gaussian Process Regression".

I. Supplementary Illustrations

A. Illustration on the Adaptive Uniform Quantization presented in Section V-B1

The proposed quantization adaptation of a mid-thread uniform quantizer presented in Section V-B1 is illustrated in Figure S1. Such a figure presents a quantizer with 8 quantization levels in which the quantizer's input y will be expressed by 3 bits at the quantizer's output. The illustration shows how the statistics of the quantizer's Gaussian input y are used to set the mid-value $\bar{y}$ and the range l of the uniform quantizer. Given the quantizer's input $y\sim\mathcal{N}\left(\mu_y,\sigma_y^2\right)$, we adapt a uniform quantizer by setting its mid-value $\bar{y}=\mu_y$ and its range $l=2c\sigma_y$, for some given $c>0$. The proposed quantizer (denoted by $\mathbb{Q}_{\text{ua}}(y;\mu_y,\sigma_y,c,b)$ has parameters that are adapted for a quantization resolution appropriate for the most likely values of y. The quantization levels will be constructed by setting uniform quantization intervals around the mid-value where the length of each interval is given by $q=2c\sigma_y/2^b$. Depending on which interval the quantizer's input belongs to, it will be assigned one of the 8 different 3-bits binary numbers considered in this case.

B. Illustration on the Gaussian Process concept presented in Section V-A

In Figure S2 the concept of GP is depicted. The distribution of a Gaussian process is the joint distribution of infinitely many random variables. Every finite collection of those random variables has a multivariate normal distribution, i.e. every finite linear combination of them is normally distributed. The blue lines in Figure S2 represent such a collection of random variables. The crosses in the graph represent the observed values of the function $f(x)$ and we can see that the many random variables all converge to those points. This shows that for a given set of training points, there are potentially infinitely many functions that fit the data. Gaussian processes assign a probability to each of these functions and the mean of this probability distribution then represents the most probable characterization of the data. The black curve in Figure S2 is the mean function of the GP. Finally, the use of a probabilistic approach allows us to incorporate the confidence of the prediction into the regression result. Such confidence region is represented in the gray area in the illustration and uses the second-order statistics of the joint distribution to be constructed. It is noticeable that the farther we are from a point of the training set, then the mentioned region becomes bigger making the predictive mean more unreliable.

II. Mathematical Proofs

A. Proof of Proposition 2

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

$$\hat{y}=A^{-1}[A(y-\mu(x))+\epsilon_{\mathbb{Q}}]+\mu(x)=y+A^{-1}\epsilon_{\mathbb{Q}}=y+\hat{\epsilon}_{\mathbb{Q}} \tag{1}$$

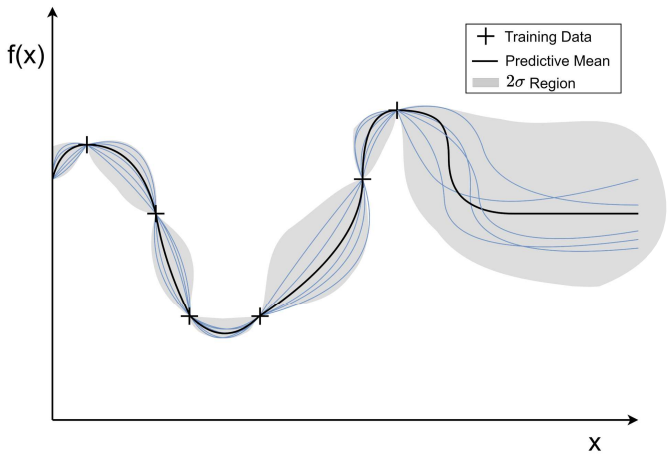

Fig. S2: Diagram representing the GP regression.

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

$$E[\hat{\epsilon}_{\mathbb{Q}}\hat{\epsilon}'_{\mathbb{Q}}] = (A)^{-1}E[\epsilon_{\mathbb{Q}}\ \epsilon'_{\mathbb{Q}}]((A)^{-1})' = (A)^{-1}\Lambda_{\epsilon_{\mathbb{Q}}}((A)^{-1})' \tag{2}$$

where $E[\epsilon_{\mathbb{Q}}\ \epsilon'_{\mathbb{Q}}]$ is the auto correlation of the quantization error and $\Lambda_{\epsilon_{\mathbb{Q}}}$ is a diagonal matrix with entries given by $\frac{1}{12}\tilde{q}^2$.

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

$$E[\hat{\epsilon}_{\mathbb{Q}}\hat{\epsilon}'_{\mathbb{Q}}] = A^{-1}\Lambda_{\epsilon_{\mathbb{Q}}}(A^{-1}) = \frac{\Gamma^2(b,c)}{12}(A^{-1}\tilde{\Lambda}_{\epsilon_{\mathbb{Q}}}(A^{-1})') \tag{3}$$

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

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

$$E[\hat{\epsilon}_{\mathbb{Q}}\hat{\epsilon}'_{\mathbb{Q}}] = \frac{\Gamma^2(b,c)}{12}\Sigma(x) = \Delta \tag{4}$$

B. Proof of Theorem 1

The proposed LMMSE will be given by the linear combination

$$\mu(x_*) = H\hat{Y} \tag{5}$$

Then, if (5) is a LMMSE then it must follow the orthogonal principle which will be given by $E[(\mu(x_*) - \hat{y}_*)(\hat{Y})'] = 0$. From this point we can obtain an expression for H

$$E[(H\hat{Y} - \hat{y}_*)(\hat{Y})'] = 0$$
$$HE[\hat{Y}(\hat{Y})'] = E[\hat{y}_*(\hat{Y})']$$
$$HE[(Y + \epsilon_n + \epsilon_{\mathbb{Q}})(Y + \epsilon_n + \epsilon_{\mathbb{Q}})'] = \Phi(x_*, X) \tag{6}$$

Since $\epsilon_{\mathbb{Q}}$ is uncorrelated from y and ϵ_n is independent from the rest, all cross products will be turn to zero by the expectation. Therefore we can simplify the expression to

$$H[\Phi(X,X) + E[\epsilon_{\mathbb{Q}}\epsilon'_{\mathbb{Q}}] + \sigma_n I_{m(p+1)}] = \Phi(x_*, X) \tag{7}$$

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

$$H = \Phi(x_*, X)[\Phi(X,X) + \Delta + \sigma_n I_{m(p+1)}]^{-1} \tag{8}$$

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

$$\Sigma(x_*) = E[(\hat{y}_* - \mu(x_*))(\hat{y}_* - \mu(x_*))^T] \tag{9}$$
$$\Sigma(x_*) = E[(\hat{y}_* - H\hat{Y})(\hat{y}_* - H\hat{Y})^T] \tag{10}$$

Expanding this expression and operating the expectations we get

$$\Sigma(x_*) = \Phi(X_*, X_*) - H^T\Phi(X, X_*) - \Phi(X_*, X)H - H^T\Phi(X,X)H$$

Finally, introducing the expression of H in (8) we get

$$\Sigma(x_*) = \Phi(X_*, X_*) - \Phi(X_*, X)[\Phi(X,X) + \sigma_n^2 I_{m(p+1)} + \Delta]^{-1}\Phi(X, X_*)$$

C. Proof of Theorem 2

The expression for our estimator will be defined as

$$\bar{y}_* - \mu(x_*) = B(\hat{y}_* - \mu(x_*)) \tag{11}$$

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

$$\begin{aligned} E[(B(\hat{y}_* - \mu(x_*)) - (\hat{y}_* - \mu(x_*)))(\hat{y}_* - \mu(x_*))'] = 0 \\ B\ E[(\hat{y}_* - \mu(x_*))(\hat{y}_* - \mu(x_*))'] = E[(\hat{y}_* - \mu(x_*))(\hat{y}_* - \mu(x_*))'] \end{aligned} \tag{12}$$

So, inserting the definition of $\mu(x_*)$ and $\Sigma(x_*)$ from Theorem 1 into (12) will lead to the simplified version

$$B = \Sigma(x_*)[\Sigma(x_*) + \sigma_n I_{p+1} + \Delta_{p+1}]^{-1} \tag{13}$$

III. Supplementary Information to Numerical Results

A. Details on the Calculation of Variables θ_i and Υ_i in Section VII-A2

In [1] the variables θ_i and Υ_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 [1]. In such, to calculate each θ_i we first create θ_i^0 which is a p-dimensional vector with entries randomly generated and uniformly distributed on [-1,1]. Then, the value of θ_i to be used is

$$\theta_i = \theta_i^0 + \eta u_i \tag{14}$$

where η 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 Υ_i we first create Υ_i^0 as a symmetric $p \times p$ matrix whose entries 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, Υ_i is constructed as

$$\Upsilon_i = \begin{cases} \widetilde{\Upsilon}_i, & \text{if } \lambda_{min}(\widetilde{\Upsilon}_i) > \epsilon \\ \widetilde{\Upsilon}_i + \big(\epsilon - \lambda_{min}(\widetilde{\Upsilon}_i)\big) I_p, & \text{otherwise} \end{cases} \tag{15}$$

where $\lambda_{min}(\widetilde{\Upsilon}_i)$ denotes the smallest eigenvalue of $\widetilde{\Upsilon}_i$ and $\epsilon > 0$ is some positive constant. The procedure in (15) is performed to ensure that Υ_i is positive definite.

B. Details of MAC Metric presented in Section VII-C1

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 [2] 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 ξ. 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_{simul}. The expected transmission time in one iteration round will be $T^k_{\text{round}} = T^k_{\text{simul}}/\xi^*$, where ξ^* 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.

C. Complementary Numerical Results Graphs

In Fig. S3 we present the results for 10 and 30 agents when the dimension of the variables is set to be $p = 10$ discussed in Section VII-E.

IV. Supplementary Statistical Results for 30 Agents $p = 10$

The results presented up to this point were only considering the median among the 100 simulation performed for each case. Such approach is useful to visualize the general trend among all cases, however there is information lost by only considering such statistic. For that reason we also generated boxplots to get more insight about the data sets distribution. Such boxplots show how the data of our 100 simulations is spread around the median and in between the first and third quartile. In this subsection, we are going to discuss boxplot result for the data sets coming from 30 agents with $p = 10$. Because not all methods can be presented in a single graph we decided to present the graph comparing only four methods.

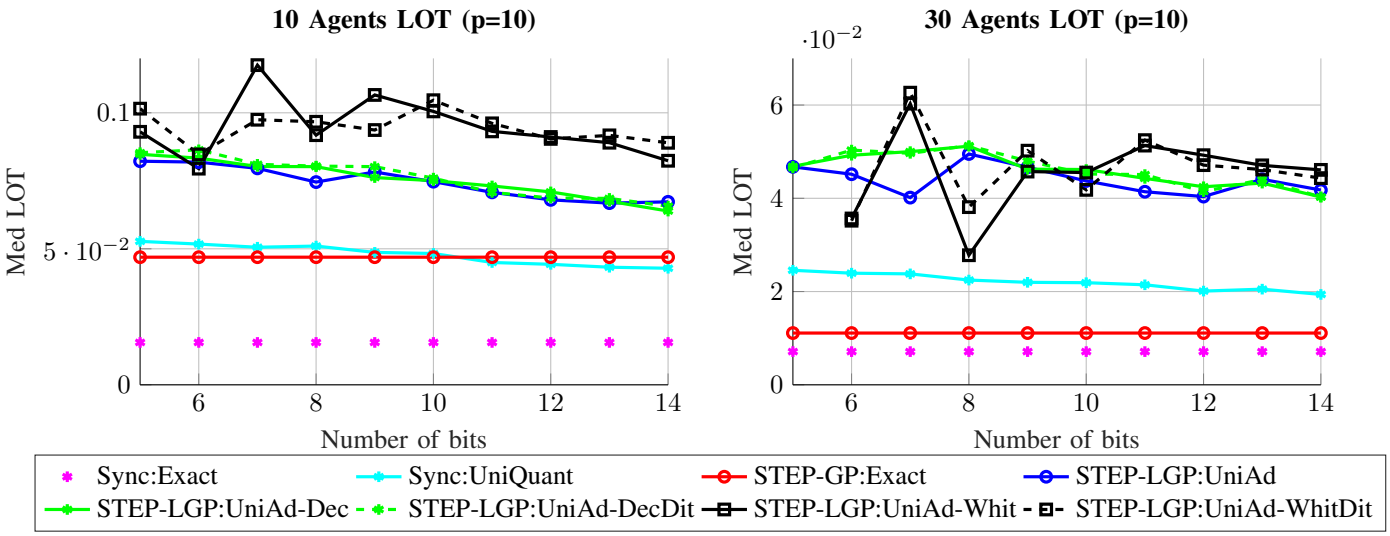

Fig. S3: Performance in the LOT metric of the adaptive quantization methods at different bit resolutions for 10 agents (left) and 30 agents (right) with $p = 10$. The plots show the median LOT of 100 simulations for different sets of parameters θ_i and Υ_i.

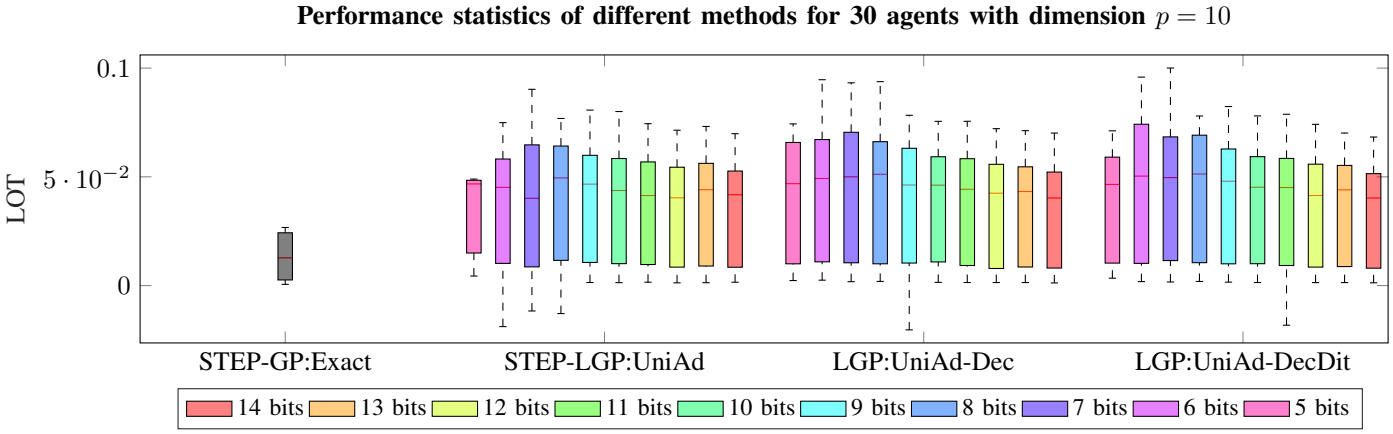

Fig. S4: Boxplot comparing the methods STEP-GP:Exact, LGP:UniAd, LGP:UniAd-Dec, and LGP:UniAd-DecDit for 30 agents with $p = 10$ in terms of the LOT metric for different resolutions. The results presented gather the information out of 100 simulations for different sets of parameters θ_i and Υ_i. Since STEP-GP:Exact is not affected by quantization it is presented with a single boxplot with gray color.

Fig. S4 shows the boxplots comparing STEP-GP, STEP-LGP:UniAd, STEP-LGP:UniAd-Dec, and STEP-LGP:UniAd-DecDit for the case where we have 30 agents with $p = 10$. The Sync:Exact was not shown in the graph since it did not have much variation among the 100 simulations, not giving much more information than the one already presented in the median plots. Since STEP-GP:Exact is not affected by quantization it is presented with a lone boxplot. Comparing STEP-GP:Exact with STEP-LGP:UniAd, we can see that the former presents less spread of its data while STEP-LGP:UniAd has a significant spread for the 7 to 14 bits cases. Also, for all bit resolution STEP-LGP:UniAd-Dec and STEP-LGP:UniAd-DecDit present slightly more variation than STEP-LGP:UniAd.

REFERENCES

[1] X. Cao and K. J. R. Liu, "Dynamic sharing through the ADMM," *IEEE Transactions on Automatic Control*, vol. 65, no. 5, pp. 2215–2222, 2020.
[2] N. A. NAGENDRA. (2013) Ieee 802.11 mac protocol. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/44110-ieee-802-11-mac-protocol