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.

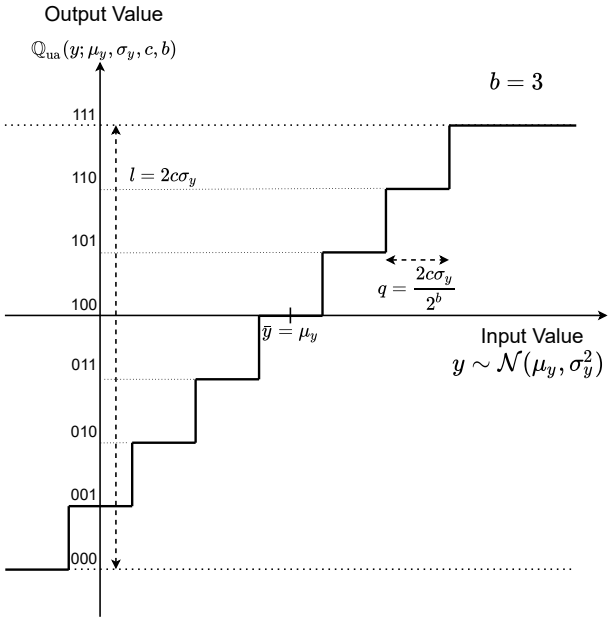

Fig. S1: Diagram representing the proposed adaptation of the uniform quantizer using the statistics of the Gaussian input y. The uniform quantizer's mid-point is set as the mean of y and the range depends on the variance of y.

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

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

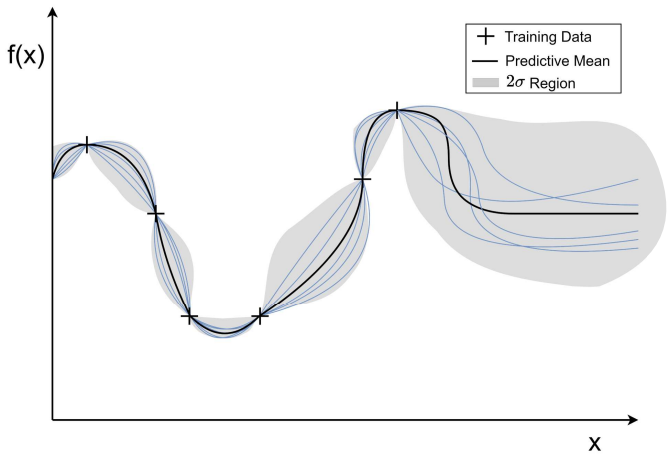

Fig. S2: Diagram representing the GP regression.

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.

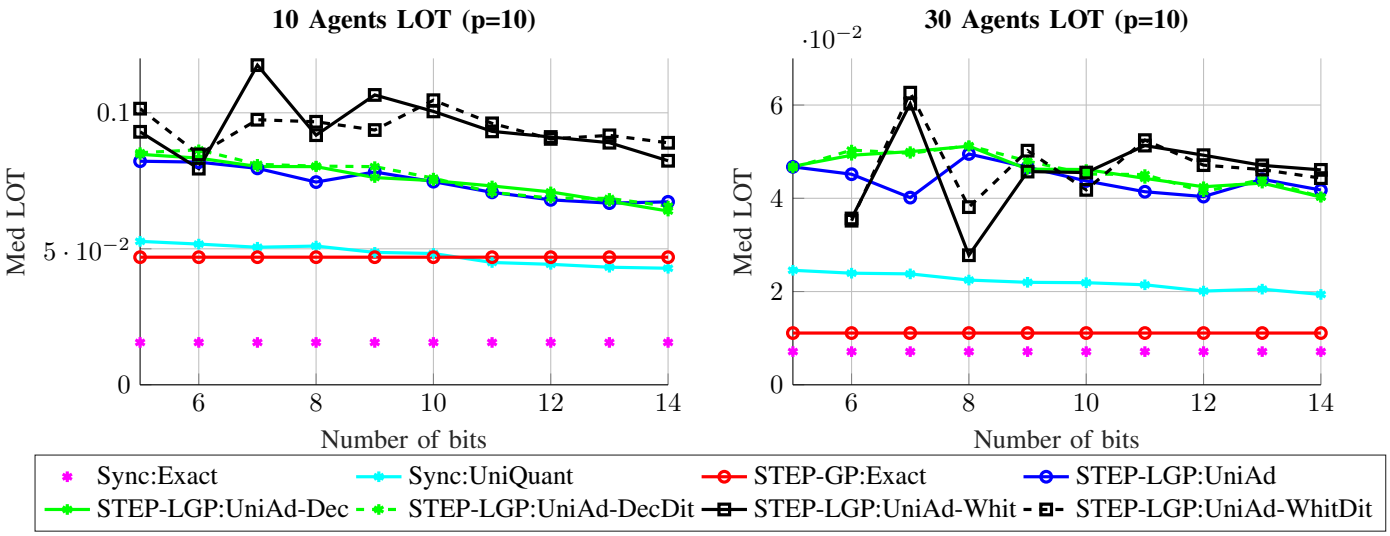

Fig. S3: Performance in the LOT metric of the adaptive quantization methods at different bit resolutions for 10 agents (left) and 30 agents (right) with $p = 10$. The plots show the median LOT of 100 simulations for different sets of parameters θ_i and Υ_i.

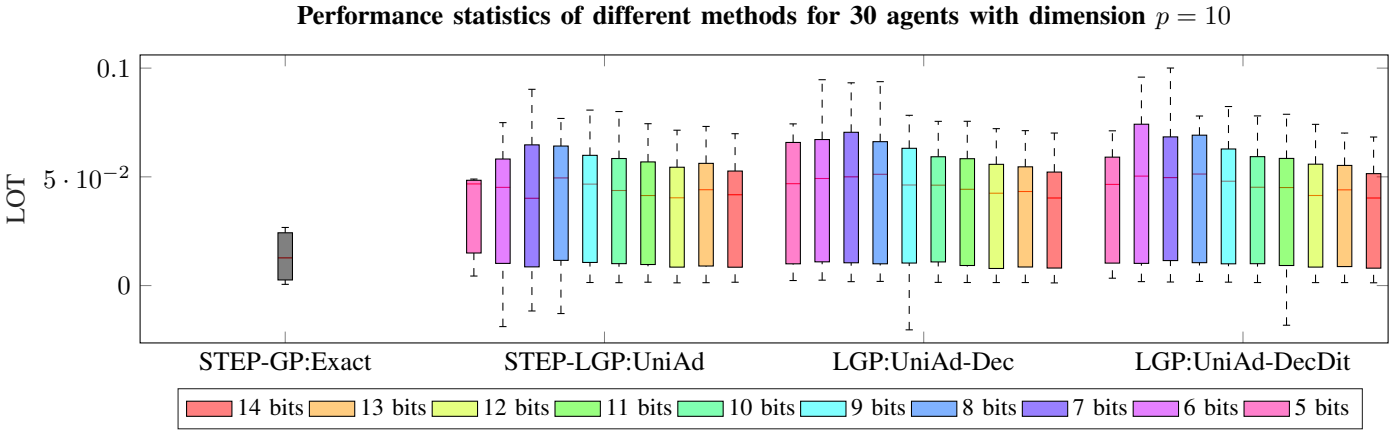

Fig. S4: Boxplot comparing the methods STEP-GP:Exact, LGP:UniAd, LGP:UniAd-Dec, and LGP:UniAd-DecDit for 30 agents with $p = 10$ in terms of the LOT metric for different resolutions. The results presented gather the information out of 100 simulations for different sets of parameters θ_i and Υ_i. Since 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