AVERAGING BASED DISTRIBUTED ESTIMATION ALGORITHM FOR SENSOR NETWORKS WITH QUANTIZED AND DIRECTED COMMUNICATION

Shanying Zhu, Yeng Chai Soh, Lihua Xie and Shuai Liu

EXQUISITUS, Centre for E-City, School of Electrical and Electronic Engineering,
Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

ABSTRACT

In this paper, we consider the distributed parameter estimation problem over sensor networks in the presence of quantized data and directed communication links. We propose a two-stage algorithm aiming at achieving the centralized sample mean estimate in a distributed manner. The running average technique is utilized in the proposed algorithm to smear out the randomness caused by the probabilistic quantization scheme. It is shown that the centralized estimate can be achieved in the mean square sense, which is not observed in the conventional consensus algorithms. Simulation results are presented to illustrate the effectiveness of the proposed algorithm and highlight the improvements by using running average technique.

Index Terms— Distributed estimation, probabilistic quantization, running average, directed topology

1. INTRODUCTION

Ad hoc sensor networks, composed of a large number of signal processing devices (nodes), are massively distributed systems for sensing and processing of spatial data. A popular application of sensor networks is decentralized estimation of unknown parameters using samples collected from nodes [1–4]. In a typical estimation problem over networks, nodes make noisy measurements of a scalar of interest. The main concern is how to utilize the samples to produce a desired estimate by only exchanging information between neighboring nodes.

Distributed estimation in ad hoc networks is usually based on successive refinements of local estimates maintained at individual nodes. Often nodes are powered by batteries and thus have limited computing and communication capabilities. Another aspect is bandwidth constraint, which renders the transmission of real-valued data impractical. Thus data needs to be quantized prior to transmission. However, this process introduces certain quantization errors which is accumulated throughout the iterations, making the estimation process fluctuating or even divergent [5].

Recently, much attention has been paid to the effect of quantization on distributed consensus algorithms. For instance, deterministic quantizers are used in [6–10], where convergence can only be guaranteed up to a neighborhood of the average of the initial states. Another thread is to adopt probabilistic quantization schemes [6, 11]. It was shown that almost surely consensus can be reached with a shift from the desired average. In fact, even employing the decaying link weights satisfying a persistence condition can not guarantee the convergence to the desired average [12]. By exploring the temporal information of the successive states, Ref. [13] showed that the desired average can be obtained in the mean square sense. All the above works assume symmetric communication. Actually, in ad hod networks, communication links between certain pairs of nodes may be directed, which could be caused by non-homogeneous interference and so on.

To further address the residual issue of quantization, dynamic encoding/decoding schemes were proposed in [14, 15] to ensure the convergence to the exact average value. A similar idea was adopted in [16] to design a progressive quantizer such that the quantization intervals could be reduced progressively during the convergence of the algorithm. Although the dynamic quantizations perform well, some spectral properties of the Laplacian matrix of the underlying topology have to be known in advance based on which the encoder-decoder parameters are carefully chosen.

In this paper, we propose a two-stage distributed estimation algorithm for ad hoc networks with quantized communication and directed topologies. The running average technique is utilized to limit the quantization effect on the estimation process. Unlike [6–9, 11, 12, 17], our algorithm can be run on any strongly connected graphs without any knowledge of the out-neighbor information of nodes and the left eigenvector of the corresponding Laplacian matrix. By defining a new quantity rather than the final state as the estimate, we show that the centralized estimate can be achieved in the mean square sense. This is the distinct feature of the averaging based estimation algorithm that differs from the existing ones. The results extend the one in [13] from undirected

This research is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore.

graphs to directed graphs. Moreover, the simple probabilistic quantizer is used in the proposed algorithm which does not depend on the complicated design of dynamic encoder-decoders as in [14–17].

2. PROBLEM FORMULATION

Consider the estimation problem in an ad hoc sensor network consisting of N nodes, each making observations of an unknown parameter $\theta \in \mathbb{R}$,

$$y_i = \theta + n_i,\ i = 1, 2, \ldots, N,$$

where n_i are zero mean, i.i.d. Gaussian noises. In the centralized case, we know that the linear minimum mean square error estimate can be computed using the sample mean estimator $\hat{\theta} \triangleq \frac{1}{N}\sum_{i=1}^{N} y_i$, which attains the Cramér-Rao lower bound [18].

The distributed estimation problem consists in computing the centralized $\hat{\theta}$ at every node without requiring global knowledge of $\{y_i\}_{i=1}^{N}$. We model the communication topology over which information is exchanged as a weighted digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{1, 2, \ldots, N\}$, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denotes all the communication links and $\mathcal{A} = [a_{ij}]_{N\times N}$ is composed of weights $a_{ij} > 0$ associated with each edge $(j, i) \in \mathcal{E}$. The directed edge (j, i) means that node i can receive data from node j. All these nodes are denoted as neighbors $\mathcal{N}_i$ of node i. We assume that

(**A1**) The graph $\mathcal{G}$ is strongly connected, i.e., for any two nodes i and j, there exists a directed path from i to j.

In the case of limited communication rate between nodes, each node needs to quantize the data prior to transmission. We adopt the following estimation algorithm at each node i,

$$\begin{aligned} x_i(t+1) = x_i(t) + \epsilon_i(t) \\ +\alpha \sum_{j\in\mathcal{N}_i} a_{ij}[\mathcal{Q}(x_j(t) + \epsilon_j(t)) - \mathcal{Q}(x_i(t) + \epsilon_i(t))], \end{aligned} \tag{1}$$

with initial guess $x_i(0) = y_i$, where $\alpha > 0$ is the weight, $\mathcal{Q}(\cdot)$ denotes the quantization operation and $\epsilon_i(t)$ is a correction term to be designed latter.

Remark 1 *For standard consensus algorithms in the absence of quantized communication, it is known that the state $x_i(t)$ will converge to the weighted average of $x_i(0)$ rather than $\hat{\theta}$, if we set $\epsilon_i(t) = 0$ [19]. The introduction of the correction term $\epsilon_i(t) \neq 0$ in* (1) *is meant to regulate the weighted average such that the desired sample mean can be achieved.*

Each node is equipped with a probabilistic quantizer $\mathcal{Q}(\cdot) : \mathbb{R} \to \mathcal{S}_\Delta \triangleq \{k\Delta : k \in \mathbb{Z}\}$, where Δ is the quantization step-size. For any $x \in \mathbb{R}$, it is quantized in a probabilistic manner as follows

$$\mathcal{Q}(x) = \begin{cases} \left\lceil \frac{x}{\Delta} \right\rceil \Delta, & \text{with probability } p, \\ \left\lfloor \frac{x}{\Delta} \right\rfloor \Delta, & \text{with probability } 1-p, \end{cases}$$

where $p = x/\Delta - \lfloor x/\Delta \rfloor$, $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ denote the floor and ceiling functions, respectively. We can prove that $\mathcal{Q}(x)$ is an unbiased estimator of x with finite variance [6, 11], that is, $\mathbb{E}\{\mathcal{Q}(x)\} = x$ and $\mathbb{E}\left\{(\mathcal{Q}(x) - x)^2\right\} \leq \Delta^2/4$. Further, $|\mathcal{Q}(x) - x| \leq \Delta$. Actually, the above quantization is equivalent to a substractively dithered method [11, 20]. We make the following natural assumption:

(**A2**) The quantization errors are independent from the data and are temporally independent.

As discussed previously, the existing consensus algorithms with quantized transmission can not achieve the exact $\hat{\theta}$ even for undirected graphs. Motivated by a rule of thumb in statistics, i.e., large samples have smoothing effects [21], we use the following running average to smooth the samples

$$\bar{x}_i(K) \triangleq \frac{1}{K}\sum_{t=t_0}^{t_0+K-1} x_i(t),\ \ \forall i \in \mathcal{V}, \tag{2}$$

where $t_0 \in \mathbb{Z}_{\geq 1}$ is the starting point of the averaging. The new quantity $\bar{x}_i$ will be used as the estimate of node i, which is different from those used for existing consensus algorithms.

3. DISTRIBUTED ESTIMATION ALGORITHM VIA RUNNING AVERAGE

In this section, we describe the proposed two-stage algorithm: At the first stage, we estimate the normalized left eigenvector ω of the Laplacian $L \triangleq D - A$ corresponding to the zero eigenvalue, i.e., $\omega^T L = 0$ and $\mathbf{1}^T\omega = 1$, where $D \triangleq \text{diag}\{d_1, d_2, \ldots, d_N\}$ and $d_i = \sum_{j\in\mathcal{N}_i} a_{ij}$ denotes the in-degree of node i; At the second stage, we design the correction term $\epsilon(t)$ in (1) by using the estimates of the left eigenvector obtained at the first stage.

3.1. Distributed estimation of the left eigenvector ω

At the first stage, each node i maintains a vector $z_i = [z_{i1}, z_{i2}, \ldots, z_{iN}]^T \in \mathbb{R}^N$ to store the estimate of ω. At each iteration, the nodes update their variables as follows:

$$z_i(t+1) = z_i(t) + \alpha \sum_{j\in\mathcal{N}_i} a_{ij}\left[\mathcal{Q}(z_j(t)) - \mathcal{Q}(z_i(t))\right], \tag{3}$$

with initial values $z_{ii}(0) = 1$, $z_{ij}(0) = 0$, $\forall j \neq i$, where $0 < \alpha < \frac{1}{\max_i d_i}$ and $\mathcal{Q}(\cdot)$ is componentwise for vectors.

We apply the running average technique to the left eigenvector estimation problem, and adopt the running average $\bar{z}_i(t) = \frac{1}{t-k_0+1}\sum_{k=k_0}^{t} z_i(k)$ as the estimate of ω, where $k_0 \in \mathbb{Z}_{\geq 1}$. Algorithm 1 shows the distributed estimation algorithm of the left eigenvector ω at the t-th iteration run by node i. In the algorithm, we take $z_{ii}(0) = N^\kappa$ with $\kappa \geq 0$ instead of the original $z_{ii}(0) = 1$. One reason behind this operation is that convergence of the original $\bar{z}_{ii}$ to ω_i is equivalent to its convergence to $N^\kappa\omega_i$ in the new scale. Actually, we can choose different κ_i for each node.

Algorithm 1 Distributed estimation of ω at node i

Input: α, N, κ, a_{ij}, k_0.
Output: $\bar{z}_i/N^{\kappa}$.
1: **Initialization:** $z_{ii}(0) = N^{\kappa}$, $z_{ij}(0) = 0, \forall j \neq i$.
2: Receive data from neighbors: $\mathcal{Q}(z_j(t))$, $j \in \mathcal{N}_i$.
3: Update the estimate of ω via (3).
4: **if** $t < k_0$ **then**
5: $\quad \bar{z}_i(t) = z_i(t)$.
6: **else**
7: $\quad$ Update the average $\bar{z}_i(t)$:
$\quad \bar{z}_i(t+1) = \frac{t-k_0+1}{t-k_0+2}\bar{z}_i(t) + \frac{1}{t-k_0+2} z_i(t+1)$.
8: **end if**

The next result presents the mean square performance of Algorithm 1, whose proof can be found in [22].

Theorem 1 *Suppose that (**A1**) and (**A2**) hold, then the running average $\bar{z}_i(t)$ converges to the left eigenvector ω in mean square for each node $i \in \mathcal{V}$. And for all large t, the rate of mean square convergence of Algorithm 1 is given by*

$$\mathbb{E}\left\{\|\bar{z}_i(t) - \omega\|_2^2\right\} = \mathcal{O}\left(\frac{1}{t}\right).$$

Moreover, for any constant $0 < \eta < 1$, there exists $t_\eta \in \mathbb{Z}_{\geq 0}$ such that

$$\bar{z}_{ii}(t) \geq \eta w_i \text{ a.s. } \forall t \in \mathbb{Z}_{\geq t_\eta}, i \in \mathcal{V}.$$

Remark 2 *Different from the standard consensus algorithm [19, 23], the averaging based method has a universal convergence rate of $\mathcal{O}(t^{-1})$, independent of the network topology. The possible effect of the network topology only lies in the rate coefficient, which depends on α, N, L, ω and Δ.*

3.2. Design of the correction term $\epsilon(t)$

The second stage is concerned with the design of an appropriate correction term $\epsilon(t)$ in (1) to compensate the unidirectional effect of the communication links.

Let $\tau \in \mathbb{Z}_{\geq 1}$ be the integer triggering the estimation algorithm (1). Motivated by the consensus algorithm proposed in [24], we design the correction term $\epsilon_i(t)$ for each $i \in \mathcal{V}$

$$\epsilon_i(t) \triangleq \begin{cases} \left(\frac{1}{N\bar{z}_{ii}(\tau)} - 1\right)x_i(0), & t = 0, \\ \left(\frac{1}{N\bar{z}_{ii}(t+\tau)} - \frac{1}{N\bar{z}_{ii}(t+\tau-1)}\right)x_i(0), & t \in \mathbb{Z}_{\geq 1}. \end{cases} \quad (4)$$

One issue remaining before the implementation of the algorithm is the well-definedness of $\epsilon_i(t), \forall i \in \mathcal{V}$. This is validated by Theorem 1, which shows that the denominators of $\epsilon(t)$ is almost surely non-zero for large t. As for implementation, we may choose $\tau = t_\eta$. In this case, we almost surely have $\epsilon(t) > 0$ for all $t \in \mathbb{Z}_{\geq 0}$. Actually, with the setup in Algorithm 1, it is possible to choose a much smaller $\tau \ll t_\eta$. This is verified by the simulation results in Section 4.

The proposed algorithm of the t-th iteration run by node i at the second stage is shown in Algorithm 2. In the algorithm, we modify the definition of the correction $\epsilon(t)$ to accommodate the setup in Algorithm 1 (see lines 1 and 4).

Algorithm 2 Distributed estimation algorithm with quantization via running average at node i

Input: α, N, κ, a_{ij}, τ, t_0, $x_i(0)$.
Output: $\bar{x}_i$.
1: **Initialization:** $\epsilon_i(0) = \left(\frac{N^{\kappa-1}}{\bar{z}_{ii}(\tau)} - 1\right)x_i(0)$.
2: Receive data from neighbors: $\mathcal{Q}(x_j(t) + \epsilon_j(t))$, $j \in \mathcal{N}_i$.
3: Update $x_i(t)$ via (1).
4: Compute the correction:
$\epsilon_i(t+1) = N^{\kappa-1}x_i(0)\frac{\bar{z}_{ii}(t+\tau) - \bar{z}_{ii}(t+\tau+1)}{\bar{z}_{ii}(t+\tau)\bar{z}_{ii}(t+\tau+1)}$.
5: **if** $t \geq t_0$ **then**
6: $\quad$ Update the average $\bar{x}_i(K)$:
$\quad \bar{x}_i(K+1) = \frac{K}{K+1}\bar{x}_i(K) + \frac{1}{K+1}x_i(t)$.
7: **end if**

The following result summarizes the convergence result of Algorithm 2. The proof is referred to [22].

Theorem 2 *Suppose that (**A1**) and (**A2**) hold, then at each node $i \in \mathcal{V}$, the running average $\bar{x}_i(K)$ converges to the centralized sample mean estimate $\hat{\theta}$ in the mean square. Moreover, the rate of convergence is given by*

$$\mathbb{E}\left\{(\bar{x}_i(K) - \hat{\theta})^2\right\} = \mathcal{O}\left(\frac{\ln(t_0 + \tau - 2 + K)}{K}\right).$$

Remark 3 *Note that $\ln(t_0 + \tau - 2 + K) \approx \ln K$, for large K, this means that we can start the averaging $\bar{x}(K)$ at any time during the iteration. For example, at each stage, we can initiate the original consensus algorithm to achieve a better decaying rate, then start the running average to get higher accuracy. This is exactly what we have done in Algorithm 2.*

3.3. Summary of the algorithm

The proposed distributed estimation algorithm with quantized data is composed of Algorithm 1 and Algorithm 2. We remark that the adjustment of the initial values in line 1 of Algorithm 1 has another consequence. It can be shown that $0 < \omega_i < 1$ for all $i = 1, 2, \ldots, N$, provided that Assumption (**A1**) is satisfied [19]. And for certain topologies, some ω_i's are rather close to 0. It is probable that zeros would occur in the denominators in ϵ_i (4) during the quantization process for the first several iterations. In this case, the correction term in line 4 of Algorithm 2 will be meaningless and we have to wait a long time before triggering Algorithm 2. Increasing the initial values from 1 to a quite larger N^{κ} is meant to tackle this concern. We also emphasize that no further buffer is needed to store the previous states $\bar{z}_i(t)$ and $\bar{x}_i(K)$ (see line 7 of Algorithm 1 and line 6 of Algorithm 2 for their iterative implementations).

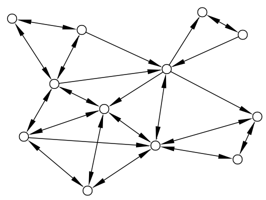


Fig. 1: A directed communication topology with 12 nodes.

4. PERFORMANCE EVALUATION

In this section, simulation results are provided to illustrate the effectiveness of the proposed algorithm.

Consider a sensor network with 12 nodes deployed to monitor an unknown parameter $\theta = 2$. The directed communication topology is shown in Fig. 1. Each node observes $y_i = \theta + n_i$, where n_i is the white Gaussian noise with zero mean and unit variance. We choose $\alpha = 1$ and $a_{ij} = (1 + d_i)^{-1}$, if $j \in \mathcal{N}_i$ and 0, otherwise, for both Algorithm 1 and Algorithm 2. For each implementation, the initial state $x_i(0)$ is randomly chosen from the interval $[y_i - 1, y_i + 1]$, $\forall i \in \mathcal{V}$. In the following simulations, we consider both the uniform quantizer (UnifQ) [16] and probabilistic quantizer (ProbQ) [11] . The proposed algorithm is denoted as ProbQ-RA. Simulation results are presented by averaging over 100 independent runs.

First, we simulate Algorithm 1. Here, $\kappa = 1.15$ and the starting point for the running average of z is taken as $k_0 = 25$. Fig. 2 depicts the estimate of the left eigenvector ω at the first node for $\Delta = 1$. From the results, we observe that steady residues occur for UnifQ, and there are fluctuations for ProbQ due to the random nature of the quantizer. While for the proposed ProbQ-RA, the running average has an obvious smoothing effect. The performance of ProbQ-RA is satisfactory even for a low quantization resolution $\Delta = 1$ compared with the large residues observed both in UnifQ and ProbQ.

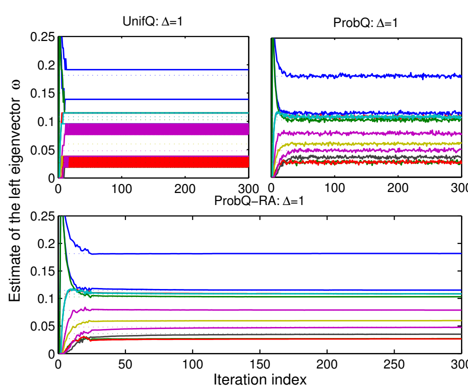
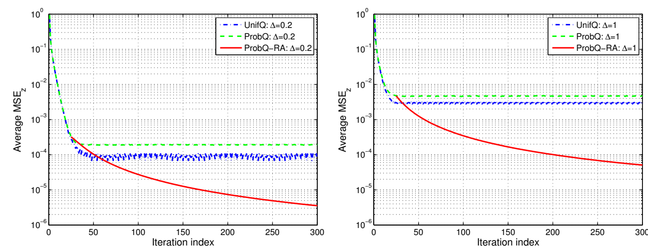


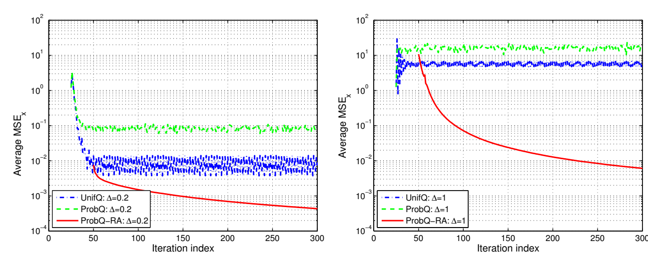
Fig. 2: Estimate of ω at the first node for $\Delta = 1$.

To quantify the performance of the proposed algorithm, we use the average of the mean square error: $\text{MSE}_z = N^{-1}\sum_{i=1}^{N}\|\bar{z}_i(t) - \omega\|^2$ and $\text{MSE}_x = N^{-1}\sum_{i=1}^{N}(\bar{x}_i(K) - \hat{\theta})^2$ for Algorithm 1 and Algorithm 2, respectively. We set $\tau = 1$ and $t_0 = 25$ for Algorithm 2. The results are shown in Fig. 3 and Fig. 4. It can be seen that the proposed ProbQ-RA outperforms UnifQ and ProbQ in both cases with $\Delta = 0.2$ and 1. From Fig. 3, we can see that the performances of UnifQ and ProbQ are acceptable for the estimates of the left eigenvector ω in both cases . However, with the errors accumulated from the first stage to the second stage, they degrade significantly for lower quantization resolution $\Delta = 1$ (see Fig. 4). Compared with UnifQ and ProbQ, the proposed ProbQ-RA degrades quite smoothly. There is only a modest increase of MSE with decreasing quantization resolution, i.e., increasing Δ from 0.2 to 1.



(a) (b)

Fig. 3: Comparison of MSE of UnifQ, ProbQ and ProbQ-RA for the estimate of ω with respect to $\Delta \in \{0.2, 1\}$.



(a) (b)

Fig. 4: Comparison of MSE of UnifQ, ProbQ and ProbQ-RA for the sample mean $\hat{\theta}$ with respect to $\Delta \in \{0.2, 1\}$.

5. CONCLUSION

We have studied the distributed parameter estimation problem over ad hoc sensor networks in the presence of quantized data and directed communication links. A two-stage algorithm such that the centralized estimate can be achieved in a distributed manner has been proposed. We have presented simulation results illustrating the effectiveness of the proposed algorithm. Comparisons with other algorithms have also been provided to highlight the improvements of the proposed one.

6. REFERENCES

[1] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Trans. Autom. Control*, vol. 58, no. 12, pp. 3112–3125, 2013.

[2] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in Ad Hoc WSNs with noisy links-Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 342–356, 2008.

[3] J. Xiao, A. Ribeiro, Z. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 27–41, 2006.

[4] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Int'l. Conf. Information. Process. Sensor Networks(IPSN)*, Los Angeles, USA, Apr. 2005, pp. 63–70.

[5] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, 2007.

[6] R. Carli, F. Fagnani, P. Frasca, and S. Zampieri, "Gossip consensus algorithms via quantized communication," *Automatica*, vol. 46, no. 1, pp. 70–80, 2010.

[7] M. El Chamie, J. Liu, and T. Başar, "Design and analysis of distributed averaging with quantized communication," in *Proc. 53rd IEEE Conf. Decision and Control(CDC)*, Los Angeles, USA, Dec. 2014, pp. 3860–3865.

[8] S. Liu, T. Li, L. Xie, M. Fu, and J. Zhang, "Continuous-time and sampled-data based average consensus with logarithmic quantizers," *Automatica*, vol. 49, no. 11, pp. 3329–3336, 2013.

[9] A. Nedić, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "On distributed averaging algorithms and quantization effects," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2506–2517, 2009.

[10] A. Kashyap, T. Basar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

[11] T. C. Aysal, M. J. Coates, and M. G. Rabbat, "Distributed average consensus with dithered quantization," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4905–4918, 2008.

[12] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Quantized data and random link failures," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1383–1400, 2010.

[13] J. Fang and H. Li, "Distributed consensus with quantized data via sequence averaging," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 944–948, 2010.

[14] R. Carli, F. Bullo, and S. Zampieri, "Quantized average consensus via dynamic coding/decoding schemes," *Int. J. Robust Nonlinear Control*, vol. 20, no. 2, pp. 156–175, 2010.

[15] T. Li, M. Fu, L. Xie, and J.-F. Zhang, "Distributed consensus with limited communication data rate," *IEEE Trans. Autom. Control*, vol. 56, no. 2, pp. 279–292, 2011.

[16] D. Thanou, E. Kokiopoulou, Y. Pu, and P. Frossard, "Distributed average consensus with quantization refinement," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 194–205, 2013.

[17] D. Li, Q. Liu, X. Wang, and Z. Lin, "Consensus seeking over directed networks with limited information communication," *Automatica*, vol. 49, no. 2, pp. 610–618, 2013.

[18] S. M. Kay, *Fundamentals of Statistical Processing: Estimation Theory*, Prentice Hall, Upper Saddle River, NJ, 1993.

[19] Z. Lin, *Coupled Dynamic Systems: From Structure Towards Stability and Stabilizability*, Ph.D. thesis, University of Toronto, Toronto, Canada, 2005.

[20] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy, "Quantization and dither: A theoretical survey," *J. Audio. Eng. Soc.*, vol. 40, no. 5, pp. 355–375, 1992.

[21] A. Gut, *Probability: A Graduate Course*, Springer, New York, second edition, 2013.

[22] S. Zhu, Y. C. Soh, and L. Xie, "Distributed parameter estimation with quantized communication via running average," submitted to *IEEE Trans. Signal Process.*, 2014. [Available online] http://arxiv.org/abs/1412.7281.

[23] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[24] A. Priolo, A. Gasparri, E. Montijano, and C. Sagues, "A distributed algorithm for average consensus on strongly connected weighted digraphs," *Automatica*, vol. 50, no. 3, pp. 946–951, 2014.