

ABSTRACT

PILGRIM, RYAN ZACHARY. Source Coding Optimization for Distributed Average Consensus. (Under the direction of Dror Baron.)

Consensus is a common method for computing a function of the data distributed among the nodes of a network. Of particular interest is distributed average consensus, whereby the nodes iteratively compute the sample average of the data stored at all the nodes of the network using only near-neighbor communications. In real-world scenarios, these communications must undergo quantization, which introduces distortion to the internode messages. In this thesis, a model for the evolution of the network state statistics at each iteration is developed under the assumptions of Gaussian data and additive quantization error. It is shown that minimization of the communication load in terms of aggregate source coding rate can be posed as a generalized geometric program, for which an equivalent convex optimization can efficiently solve for the global minimum. Optimization procedures are developed for rate-distortion-optimal vector quantization, uniform entropy-coded scalar quantization, and fixed-rate uniform quantization. Numerical results demonstrate the performance of these approaches. For small numbers of iterations, the fixed-rate optimizations are verified using exhaustive search. Comparison to the prior art suggests competitive performance under certain circumstances but strongly motivates the incorporation of more sophisticated coding strategies, such as differential, predictive, or Wyner-Ziv coding.

© Copyright 2017 by Ryan Zachary Pilgrim

All Rights Reserved

Source Coding Optimization for Distributed Average Consensus

by
Ryan Zachary Pilgrim

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Electrical Engineering

Raleigh, North Carolina

2017

APPROVED BY:

David Lalush

Michael Daniele

Dror Baron
Chair of Advisory Committee

DEDICATION

To everyone who has stood by me in tough times: my parents, my brothers, and my girlfriend, Kristin. My success rests on a foundation of love and support from you all, and I am grateful.

BIOGRAPHY

The author was born in Boulder, Colorado in 1991. His childhood was spent in Colorado, Virginia, Mississippi, and North Carolina. After graduating from Hickory High school, he attended the University of North Carolina at Asheville. He then transferred to North Carolina State University to study Biomedical Engineering, graduating *magna cum laude* with a Bachelor of Science in 2014.

His research interests span statistical signal processing, compressive sensing, and machine learning. He also enjoys programming and embedded systems development. In his spare time, he plays disc golf and the guitar.

The author completed an internship at Southwest Research Institute in 2015, where he helped create a GPS spoofing detection system. He is currently a Project Engineer at Vadum, Inc. in Raleigh, North Carolina.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dror Baron, for his patient mentorship and excellent advice. Thanks also to my collaborators Waheed Bajwa and Junan Zhu for their many helpful inputs regarding consensus and the research process. I would furthermore like to acknowledge Yanting Ma for helping to expand the main theorem of the thesis to the more general, variable distortion setting.

Special thanks to Mehmet Ercan Yildiz and Anna Scaglione for graciously providing their code for comparison, and to Yaoqing Yang and Pulkit Grover for helpful discussions about their information theoretic work with Soumya Kar on the consensus problem.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
Chapter 1 INTRODUCTION	1
1.1 Summary of prior art	2
1.1.1 Overview	2
1.1.2 Static, dynamic, and dithered quantization schemes	3
1.1.3 Topology and weight tuning	5
1.1.4 Sequence filtering	6
1.1.5 Wireless considerations	6
1.1.6 Information theoretic approaches	7
1.2 Motivation and contributions	8
1.3 Organization	9
1.4 Notation and acronyms	10
1.4.1 Notation	10
1.4.2 Acronyms	12
Chapter 2 QUANTIZED DISTRIBUTED AVERAGE CONSENSUS	14
2.1 Overview	14
2.2 Mathematical preliminaries	15
2.2.1 Matrix and graph theory concepts	15
2.3 The consensus problem	16
2.3.1 Lossy consensus	18
2.4 Source coding, quantization, and rate-distortion theory	21
2.4.1 Coding and information theory	21
2.4.2 Rate-distortion theory	22
2.4.3 Uniform and nonuniform quantization	26
2.4.4 The quantization noise model and dithering	27
Chapter 3 CODING RATE OPTIMIZATION IN CONSENSUS	30
3.1 Problem formulation	31
3.2 Abstractions for vector-valued data	33
3.3 Assumptions and analytical results	36
3.3.1 Additive noise model	37
3.3.2 Preservation of Gaussian distribution	37
3.3.3 State evolution for vector consensus	39
3.4 Optimization via generalized geometric programming	45
3.4.1 Cost function formulation	47
3.4.2 Basics of generalized geometric programming	50
3.4.3 Proof that the optimization is a GGP	51
3.5 Efficient implementation of GGP optimization	54

3.5.1	Equal-distortion simplification and implementation	54
3.5.2	Efficient search heuristic for fixed-rate quantizers	56
Chapter 4	NUMERICAL RESULTS	58
4.1	Structure of solutions	60
4.1.1	Nonmonotone behavior	62
4.2	Comparison to prior art	70
4.2.1	Implementation details	71
4.2.2	Discussion of comparison results	72
4.2.3	Comparison to predictive coding scheme	75
4.2.4	Comparison to the ProgQ scheme	78
Chapter 5	CONCLUSION	79
	BIBLIOGRAPHY	81

LIST OF TABLES

LIST OF FIGURES

Figure 2.1	The RD function $R(D)$ of a Gaussian memoryless source. The function is a nonincreasing, convex function of the normalized distortion $\frac{D}{\sigma^2}$, where σ^2 is the variance of the Gaussian source. The RD pairs above the curve are achievable, while those below the curve are not.	23
Figure 2.2	Operational RD relationships of practical quantizers: ECSQ, Lloyd-Max, and fixed-rate uniform quantization. Note that all of these approach an asymptote that is a constant offset from the Gaussian RD function as $R \rightarrow \infty$. The ECSQ performance corresponds to the uniform scalar quantizer with a level at zero. The representation level placement may be suboptimal, but it allows for coding rates below one bit per symbol. This figure was inspired by a similar plot in lecture notes from a TU Berlin source coding course [56].	25
Figure 3.1	Surface showing the evolution of $MSE_i(t)$ for the case of lossless consensus. Note that the value of $MSE_i(t)$ is not equal across all nodes for a particular iteration, and the evolution of $MSE_i(t)$ is heavily dependent on network topology. Also observe that the MSE decreases monotonically at each node as $t \rightarrow \infty$. The variance $v_i(t)$ evolves in a similar fashion. .	46
Figure 4.1	Optimal rates and MSE sequences from the solution of (3.38) ($T = 5$, $\rho_c = 0.35$, $\sigma_x^2 = 1$, $\sigma_n^2 = 0.5$, $m = 20$). <i>Top</i> : Optimal rate sequences for the variable-distortion optimization problem. The rates are plotted against iteration indices, and each line represents the rates used by a different sensor. <i>Bottom</i> : MSE sequence corresponding to the above rate sequence. The MSE values are plotted against node indices, and each line represents a different iteration. In this case, the MSE values decrease monotonically for all nodes. The lines represent values predicted by the state evolution equations (3.23)–(3.26), and the overlaid circles represent the simulated values.	59
Figure 4.2	Optimal rates and MSE sequences from the solution of (3.39) ($T = 5$, $\rho_c = 0.35$, $\sigma_x^2 = 1$, $\sigma_n^2 = 0.5$, $m = 20$). <i>Top</i> : Optimal rate sequences for the constant-distortion optimization problem. The rates are plotted against iteration indices, and each line represents the rates used by a different sensor. <i>Bottom</i> : MSE sequence corresponding to the above rate sequence. The MSE values are plotted against node indices, and each line represents a different iteration. In this case, the MSE values decrease monotonically at all nodes. The lines represent values predicted by the state evolution equations (3.23)–(3.26), and the overlaid circles represent the simulated values.	61

Figure 4.3	Influence of connectivity on the optimal rates from the solution of (3.39) ($T = 5, \rho_c = 0.35, \sigma_x^2 = 1, \sigma_n^2 = 0.5, m = 20$). <i>Top</i> : Scatter plot of aggregate rate used at each node versus node degree. <i>Bottom</i> : Surface of optimal rate sequences at each node, where the node indices are sorted by node degree in decreasing order.	63
Figure 4.4	Nonmonotone optimal rate sequences for the low-SNR, low-MSE setting ($T = 5, \rho_c = 0.35, \sigma_x^2 = 1, \sigma_n^2 = 20, m = 20$). Each line corresponds to the rate sequence used by a different node. <i>Top</i> : Variable-distortion optimization (3.38) result. <i>Bottom</i> : Constant-distortion optimization (3.39) result.	64
Figure 4.5	Nonmonotone MSE sequences. Each line corresponds to the MSE sequence of a different node ($T = 5, \rho_c = 0.35, \sigma_x^2 = 1, \sigma_n^2 = 0.5$). <i>Top</i> : MSE evolution for the variable-distortion problem (3.38). <i>Bottom</i> : MSE evolution for the constant-distortion problem (3.39).	66
Figure 4.6	Coefficients of the noise variance σ_n^2 (solid blue curve) and distortion $D(0)$ (dashed red curve) in the equation for the MSE (3.31) when the states are initialized as a Gaussian random vector corrupted by AWGN ($\rho_c = 0.35, m = 20$).	69
Figure 4.7	RD trade-off curves for the proposed schemes versus order-one predictive coding [21]. All schemes were simulated for random geometric graphs on a unit torus with connectivity radius $\rho_c = 0.35$. <i>Top</i> : number of iterations $T = 5$. <i>Bottom</i> : number of iterations $T = 7$	73
Figure 4.8	RD trade-off curves for the proposed schemes versus order-one predictive coding [21]. All schemes were simulated for random geometric graphs on a unit torus with connectivity radius $\rho_c = 0.45$. <i>Top</i> : number of iterations $T = 5$. <i>Bottom</i> : number of iterations $T = 7$	74
Figure 4.9	RD trade-off curves for the proposed schemes versus ProgQ [20]. All schemes were simulated for random geometric graphs on a unit torus with connectivity radius $\rho_c = 0.35$. <i>Top</i> : number of iterations $T = 5$. <i>Bottom</i> : number of iterations $T = 7$	76
Figure 4.10	RD trade-off curves for the proposed schemes versus ProgQ [20]. All schemes were simulated for random geometric graphs on a unit torus with connectivity radius $\rho_c = 0.45$. <i>Top</i> : number of iterations $T = 5$. <i>Bottom</i> : number of iterations $T = 7$	77

The wide availability of wireless sensors and large data sets in recent years has provided significant motivation for the development of distributed computing methods. Large sensor networks and big, distributed data sets pose a number of unique optimization opportunities, including energy management in battery-powered sensor networks [1, 2] and run time reduction in cloud computing applications [3].

In many distributed computing problems, it is necessary to compute a function of data that may be dispersed among a number of computing nodes, for instance in sensor networks, where each agent observes a different measurement of a physical process, or large-scale server farms, where the size of the data set requires distributed storage [4]. The class of distributed algorithms considered in this thesis computes these functions with only local, near-neighbor interactions. This distributed approach offers several potential advantages, including robustness to link and node failure and maintenance of privacy [5]. One popular approach to distributed function computation is called consensus [5, 6], which has many variants. Consensus protocols have found application in a wide variety of settings, including oscillator synchronization, robot swarm control, load balancing in computer networks, distributed sensor fusion, belief propagation [6], detection, estimation, target tracking [5], distributed

optimization, filtering, and environmental monitoring [7]. Consensus algorithms have also found applications in computer vision [8], where they can be used for null-space and least-squares estimation, distributed singular value decomposition, principal components and generalized principal components analyses, point triangulation, and linear pose estimation. Furthermore, consensus algorithms have recently found application in distributed eigenvector estimation and dictionary learning [9].

Although it has a wide variety of engineering applications, consensus originated as a model of social interaction in the management science and statistics communities [10] during the 1960s, according to Olfati-Saber *et al.* [6]. Its application to the problem of distributed networked computation was first studied by Borkar and Varaiya [11] and Tsitsiklis [12, 13], where it was applied to distributed estimation, optimization, and decision making. Much of the early work on consensus assumed that the nodes could communicate real-valued data to one another [14]. In realistic scenarios, the nodes must communicate within bandwidth and energy constraints, which can have a significant impact on the convergence of distributed averaging algorithms. Although many papers have been published on quantized consensus in recent years [4, 7, 14–28], a large portion consider trade-offs among run time, communication load, and final accuracy in ways that do not take advantage of the tools of constrained optimization. This work attempts to take a more structured approach to the reduction of communication in consensus than past efforts by considering rate-distortion (RD) theory [29, 30] and convex optimization techniques [31].

1.1 Summary of prior art

1.1.1 Overview

Many previous works have addressed consensus with limited coding rate. Due to the multi-disciplinary nature of the subject, several approaches have been taken to address the issue of transmitting limited-precision information. These include deterministic and dithered quan-

tization strategies using both static and dynamic quantization schemes. Because the classic linear-update consensus algorithms suffer from divergence in the presence of quantized or noisy exchanges [32], many of these works focus on proposing new algorithms and proving their asymptotic convergence properties. A variety of methods has been employed to sidestep the many issues quantization poses, including adapting the quantization range, tuning the weight link sequence, using dithering or randomized quantization, and filtering the past and present state values. This section provides a survey of these works before introducing and distinguishing our approach.

1.1.2 Static, dynamic, and dithered quantization schemes

Early publications on the topic, such as the work of Xiao *et al.* [32], show that introducing perturbations of constant variance (such as quantization error) into the traditional consensus state update prevents convergence due to the limited precision of the quantizer. Similar convergence issues preclude convergence when the quantization range is held constant [14]. Chamie *et al.* [16], like Frasca *et al.* [14], considered the case of nonadaptive, deterministically quantized consensus, and they showed that in finite time, consensus is achieved in the sense that the network state converges to one of the quantization levels.

To surmount the convergence issue associated with quantization, Aysal *et al.* [19] proposed a randomized quantization scheme, and they showed it was equivalent to dithering. With this probabilistic quantization scheme, Aysal *et al.* proved that consensus can be achieved almost surely at one of the neighboring quantization values and analyzed its performance relative to unquantized consensus. In the case of networks with quantization and random link failures (i.e., time-varying topology), Kar and Moura [27] showed that by subjecting the communication links to a persistence condition, dithered quantized consensus can converge. They also derived probability bounds on consensus within a certain mean square error (MSE) range. Under similar topological assumptions, Nedić *et al.* [25] assessed the performance degradation resulting from quantization and provided tight polynomial bounds on the convergence time

for a wide class of distributed averaging algorithms.

Due to the difficulties associated with quantization error, many works address the incorporation of dynamic encoding/decoding strategies into consensus protocols. However, many of these schemes do not explicitly consider the RD trade-off and offer certain heuristics to optimize communication performance within their proposed frameworks.

Carli *et al.* [15, 17] assessed the performance of a “zoom-in, zoom-out” strategy originally studied in the context of quantized feedback control system stabilization. In this scheme, the quantization range grows in the case of saturation and shrinks otherwise. The authors demonstrated convergence for certain topologies, numbers of quantizer levels, and initial quantizer range values. Interestingly, they showed that under certain circumstances, convergence speed can be faster than in the case of ideal, unquantized transmission.

Similarly to Carli *et al.* [15, 17], Rego *et al.* [33] proposed an algorithm with progressively shrinking quantization range and derived conditions on the design parameters to guarantee bounded steady-state error. To demonstrate the efficacy of their approach, they simulated a vehicle formation control problem.

Li *et al.* [34] proposed a control scheme capable of guaranteeing asymptotic consensus with as little as one-bit data exchange per iteration. The authors quantified the convergence rate in terms of network properties and coding rate. To achieve convergence while transmitting a single bit at each node per iteration, the authors relied on differential encoding/decoding [35, Sec. 7.2]. Fang and Li [18] proposed a scheme that adjusts its quantizer parameters by learning from previous runs, and they showed convergence under the assumption that the quantization error converges to zero. Similarly, Thanou *et al.* [20] proposed a differential encoding strategy with exponentially decaying quantization range. Their work is based on an “average-case” analysis that does not provide the same strict performance guarantees as Li *et al.* [34], but allows for better typical performance. The numerical results demonstrate lower MSE than a number of previous works (including Li *et al.* [34], Fang and Li [18], and Carli *et al.* [15, 17]) with equal communication load, and an error decay that matches the rate of the unquantized

algorithm in the limit of many iterations. Both of these works assume a constant coding rate throughout all iterations and rely on differential encoding/decoding to achieve convergence.

Yildiz and Scaglione [21, 22, 36], unlike other authors, explicitly considered the RD trade-off to achieve an asymptotic MSE value in consensus with Gaussian states. They proposed schemes based on differential [22], predictive, and Wyner-Ziv coding [21, 36]. In Yildiz and Scaglione [21], the authors exploited the correlation of the current network state with that of previous iterations, and showed that bounded steady-state error is possible under their schemes using shrinking coding rates. Modeling the quantization error as an additive noise, they also provided a necessary and sufficient condition on the variance of the quantization error to ensure convergence. The main focus of the work [21] is to show that convergence can be achieved using asymptotically decreasing coding rates under predictive and Wyner-Ziv coding schemes. In [22], Yildiz and Scaglione considered a simpler case of differential coding and showed similar results for this approach. Yildiz and Scaglione [22] also imposed a parametric form on the distortion sequence and examined the effect of varying the convergence rate on the aggregate coding rate required to achieve a target asymptotic MSE.

1.1.3 Topology and weight tuning

Another approach to combat the effect of quantization and channel noise, and to improve energy consumption, is to tune the network topology or estimation update rule. In many of the most popular consensus algorithms, the update rule consists of a convex combination of the current node state with those of its neighbors. By adapting which nodes are considered neighbors or by altering the weights associated with each node in the neighborhood, it is possible to improve convergence properties in the presence of quantization. In the absence of quantization, Xiao and Boyd [37] showed that the fastest linear iterative update can be found by semidefinite programming. They also listed a number of heuristics for weight design in the case of incomplete knowledge of the network topology. In their later work, Xiao *et al.* [32] considered the problem of noisy exchanges and showed that the weights corresponding to the

smallest steady-state error can be found via convex optimization.

Mosquera *et al.* [23] considered a greedy approach to updating the weight sequence by minimizing the minimum MSE (MMSE) of the estimates at each node during each iteration and proposed a modified scheme that only requires statistical knowledge about the topology. The modified scheme approximates a random geometric network by a regular graph, for which each node has the same number of neighbors.

1.1.4 Sequence filtering

To suppress the perturbations resulting from quantization, some authors have considered the possibility of using the history of past state values. Zhu *et al.* [26, 38] considered the problem of distributed parameter estimation with quantization error, proposed a scheme to reduce randomness using a moving average, and bounded its almost sure performance. Similarly, Thanou *et al.* [39] considered the problem of quantized distributed averaging and demonstrated a technique to find the optimal polynomial filter coefficients to minimize the effect of quantization error. Fang and Li [40] developed a sequence averaging approach with convergence properties that improved over Frasca *et al.* [14]. In contrast to Zhu *et al.* and Thanou *et al.*, Fang and Li only computed the average in the final iteration. These approaches both reduce the randomness introduced by quantization and accelerate the convergence of consensus algorithms.

1.1.5 Wireless considerations

In cases where the network is wireless, certain aspects of the communication medium can be exploited. Nokleby *et al.* [7] assessed the resource consumption of consensus under a wireless path-loss model. They considered total transmit energy, elapsed time, and time-bandwidth product and showed that by recursively forming geographic clusters, consensus can be achieved with nearly order-optimal performance relative to all three metrics. For comparison, they also assessed the performance of some popular gossip algorithms. Similarly,

Huang and Hua [28] designed an energy planning algorithm for progressive estimation and consensus in multihop wireless sensor networks (WSNs). They formulated energy models based on assumptions on the wireless channels, and proposed energy optimization approaches for both types of estimation in the presence of quantization. However, Huang and Hua [28] only considered the simple fixed-rate uniform quantizer and did not allow the coding rate to vary over the iterations or nodes in their analysis of consensus.

1.1.6 Information theoretic approaches

Although much of the literature considers the design of specific protocols for consensus averaging, a handful of works have explored the fundamental RD limits in the distributed computation problem. Two of these derive bounds on computation time using information theoretic inequalities [41, 42], but they do not consider RD theory [29, 30] in their analyses. Recently, Su and El Gamal [43] considered the problem of computing the RD function for distributed average consensus using peer-to-peer communication protocols. The authors derived a closed form for the RD function for a two-node network, and derived upper and lower bounds on the RD function for weighted-sum and gossip-based protocols in arbitrary networks. However, the authors restricted their attention to a class of protocols that uses communication between node pairs and assumes time-invariant normalized distortion. The analysis of this thesis, by contrast, assumes communication among more than one agent at each time step (corresponding to broadcast, rather than peer-to-peer, protocols), greater flexibility in the selection of edge weights, and time-varying distortions. Furthermore, Su and El Gamal did not present results on the tightness of their weighted-sum or gossip bounds, nor did they demonstrate the achievability of these bounds except in the case of a large star network with a centralized protocol.

In a similar vein, Yang *et al.* [44] considered RD bounds for two lossy in-network function computation scenarios. The first of these is data aggregation, in which data is routed through a tree network to a fusion center. Along the way, each node computes a partial result before

communicating its result to the next node in the path. The second scenario is that of consensus, in which each node forms an estimate of the desired quantity. Although Yang *et al.* [44] provided bounds on the RD relationship for consensus in tree networks and proved the achievability of the derived bounds, their analysis is limited to the setting where the network is tree-structured. Often it is beneficial to consider more flexible topologies, such as random geometric graphs [45], which have been used to model WSNs [46]. In general, random geometric graphs and their real-world WSN counterparts have loops.

1.2 Motivation and contributions

This thesis presents a framework for attaining an estimate of the network sample mean at each node, within a desired average level of accuracy, with finite run time and minimal total communication cost using either deterministic or dithered quantization. Our analysis is informed by the results of RD theory [29, 30] and convex optimization [31], which allow a more structured approach than some of the prior studies in the literature that neglect constrained optimization. The proposed cost function, which originally appeared in the work of Zhu and coauthors [47–51] is simple but capable of modeling diverse networks, from large-scale server farms and cloud services [3] to battery-powered WSNs [1, 2]. Unlike much of the prior art, which is focused on showing asymptotic convergence for particular protocols, this thesis takes a constrained optimization approach similar to Huang and Hua [28] to optimizing the communication scheme for finite run time, subject to an accuracy constraint. Unlike Huang and Hua [28], however, we allow time- and node-varying coding rates. We explicitly assess the trade-offs among run time, total communication load as measured by aggregate source coding rate, and quantizer complexity to attain a final MSE across the network. The focus of this work is on a Gaussian-distributed initial network state, but the results can also be used to design fixed-rate uniform quantization schemes for other distributions. Additionally, for the case of fixed-rate coding, we present a heuristic that reduces optimization complexity but scales well compared to the exact problem. The accuracy of this heuristic is verified using

numerical experiments.

The advantages of our approach are (i) ignorance about the parametric form of the distortions, which allows greater flexibility in the selection of the optimal rates, (ii) support for different rates at each node and iteration of the algorithm, and (iii) optimization with respect to exact MSE quantities for finite iteration count.

Point (i) distinguishes our approach from Yildiz and Scaglione [22], who restricted their study to optimal distortion sequences that formed convergent series. Point (ii) contrasts this thesis with the work of Huang and Hua [28] and Thanou *et al.* [20], who required the use of a fixed-rate uniform quantizer with a rate that was constant over both nodes of the network and iterations of the algorithm. Point (iii) differentiates this thesis from both Huang and Hua [28] and Yildiz and Scaglione [21, 22]; Huang and Hua [28] optimized with respect to a bound on the MSE, and Yildiz and Scaglione [21, 22] considered the *asymptotic* MSE as the number of iterations goes to infinity.

1.3 Organization

The remainder of this thesis is organized as follows. Chapter 2 presents the previous results in the literature, including the basics of consensus, quantization, rate-distortion theory, the additive quantization noise model, and dithering. Therefore, Chapter 2 does not contain any original results. In Chapter 3, we present the main contributions of this thesis, namely the state-evolution model for the modified consensus iteration of Frasca *et al.* [14] and the application of generalized geometric programming (GGP) to optimize the source coding in consensus. Chapter 4 presents numerical results demonstrating the performance of the GGP approach and compares its performance to some of the prior art. Finally, Chapter 5 includes some concluding remarks on the contributions of this thesis, the merits and drawbacks of the presented approach, and possible topics for future research.

1.4 Notation and acronyms

1.4.1 Notation

In this thesis, uppercase bold letters (e.g., \mathbf{A}) will be used to denote matrices, and lowercase bold letters (e.g., \mathbf{x}) will be used to denote vectors. Vectors that vary with time are assigned a time index, so that a time-varying \mathbf{x} becomes $\mathbf{x}(t)$. Scalars that vary with time are denoted similarly (e.g., $x(t)$). A superscript on a square matrix (e.g., \mathbf{A}^k , $\mathbf{A} \in \mathbb{R}^{n \times n}$, k a positive integer) denotes raising that matrix to the k^{th} power (i.e, multiplying that matrix by itself $k - 1$ times). Random variables (RVs) are not distinguished by notation to avoid confusing vectors and matrices.

The following list enumerates frequently used notation and variables and their associated meanings. The meaning of the following notation may not be clear until later, and it is provided here for the reader's reference.

- $\{\cdot\}^{\top}$: transpose
- $Q(\cdot)$: quantization function
- \mathbf{I} : identity matrix
- $\mathbf{0}$: the matrix or vector of all zeros
- $\mathbf{1}$: vector of all ones
- \mathbb{R} : real numbers
- \mathbb{Z} : integers
- $\mathbb{Z}_{\geq 0}$: nonnegative integers
- $\mathbb{Z}_{>0}$: positive integers
- $\mathbb{E}[\cdot]$: statistical expectation

- $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$: multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
- $\|\cdot\|_p$: ℓ_p norm
- \mathcal{G} : undirected graph
- \mathcal{V} : vertex (node) set of a graph
- \mathcal{E} : edge set of a graph
- \mathcal{N}_i : neighborhood of node i
- \mathcal{X}, \mathcal{Y} : finite subsets of \mathbb{R}
- $H(x)$: entropy of the random variable x
- $I(x; y)$: mutual information between the random variables x and y
- $p(x)$: probability mass function (PMF) of the discrete random variable x
- $f(x)$: probability distribution function (pdf) of the continuous random variable x
- $\Phi_x(\omega)$: characteristic function of the random variable x
- $\mathcal{U}(a, b)$: uniform distribution with minimum a and maximum b
- $[\cdot]_i$: i^{th} component of a vector
- $[\cdot]_{ij}$: $(i, j)^{\text{th}}$ component of a matrix
- $\boldsymbol{\mu}_{\mathbf{x}}(t)$: mean of the random vector $\mathbf{x}(t)$
- $\boldsymbol{\Sigma}_{\mathbf{x}}(t)$: covariance matrix of the random vector $\mathbf{x}(t)$
- $\text{tr}(\cdot)$: the trace of a matrix
- $|\cdot|$: the cardinality of a set
- $\text{deg } i$: the degree of node i (i.e., $|\mathcal{N}_i|$)

1.4.2 Acronyms

Here we define a number of acronyms that will be used throughout the work.

- AWGN: additive white Gaussian noise
- CLT: central limit theorem
- ECSQ: entropy-coded scalar (uniform) quantization/quantizer
- GP: geometric program/programming
- GGP: generalized geometric program/programming
- i.i.d.: independent and identically distributed
- LMMSE: linear minimum mean square error
- LSE: log-sum-exponential or log-sum-exp
- MMSE: minimum mean square error
- MSE: mean square error
- PMF: probability mass function
- pdf: probability density function
- RD: rate-distortion
- RV: random variable
- RVec: random vector
- SDR: signal-to-distortion ratio
- SNR: signal-to-noise ratio
- VQ: vector quantization/quantizer

- WSN: wireless sensor network

This chapter presents much of the background required to understand consensus and source coding. The results presented here are not original, and they are presented to make the thesis self-contained. Our original results appear in Chapters 3 and 4.

2.1 Overview

One of the many attractive features of the variety of distributed average consensus algorithms explored in this thesis is its linearity. At each iteration, every node takes a weighted average of incoming messages from its neighbors, and in the absence of quantization errors, the performance is elegantly described using concepts from spectral graph theory [6, 37].

Unfortunately, quantization introduces nonlinearity into the state update, which complicates analysis. Using the additive quantization noise model [52] and dithering [53], however, it is possible to linearize the impact of quantization on the performance. The use of dithering has been found in previous work [19] to greatly simplify the design of distributed averaging algorithms, because the quantization error becomes uncorrelated with the quantizer input signal.

Although dithering and high-resolution assumptions greatly simplify the analysis of distributed average consensus, close attention must be paid to the design of the state update. If quantization errors are introduced into the lossless algorithm, then the network state will not converge to the true sample average [21, 32]. However, by making a simple modification to the state update, Frasca *et al.* [14] guaranteed convergence to within one quantization bin. In contrast, prior efforts [21, 22, 32] could only guarantee bounded asymptotic mean square error (MSE) due to the drift from the sample average in the presence of quantization error.

In this chapter, we explore the fundamentals of distributed average consensus, algorithmic modifications to account for quantization effects, rate-distortion (RD) theory, and the basics of the additive noise model for quantization error.

2.2 Mathematical preliminaries

Before introducing the consensus problem, it is first necessary to introduce a number of mathematical concepts related to linear algebra and graph theory.

2.2.1 Matrix and graph theory concepts

To represent the network of interest, it is necessary to model the nodes, which represent the computing elements or agents. These can be wireless sensors [7, 28], servers [9], cameras [8], or robots [6]. We assume that each node can only communicate with a subset of the other nodes of the network, so it is also necessary to model the presence or absence of communication links between them, which can be wireless channels or wired connections. These relationships are modeled by a graph [37].

In this thesis, the communication links are bidirectional connections, so we model the network as an undirected graph. This graph can be represented in a number of ways. One of

the simplest representations of the graph is a pair of sets,

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}, \quad (2.1)$$

where the graph \mathcal{G} is comprised of a set of vertices (nodes) \mathcal{V} and a set of edges \mathcal{E} between pairs of vertices [54, Sec. 1.2]. Because the communication links are bidirectional, each edge $\{i, j\} \in \mathcal{E}$ is represented as an unordered pair of vertices i and j [37].

2.3 The consensus problem

In the simplest case of the consensus problem, each node $i \in \{1, \dots, m\}$ has an initial scalar quantity

$$z_i(0) \in \mathbb{R}, \quad (2.2)$$

and the goal is to have all nodes of the network agree upon the sample mean of these quantities by iteratively exchanging messages with their neighbors [37]. The quantities $z_i(t)$ will be referred to as “states,” which in this thesis are assumed to be real-valued scalar random variables (RVs) with known distribution. More formally, let the (discrete) iteration index be a nonnegative integer,¹ $t \in \mathbb{Z}_{\geq 0}$. At $t = 0$, the states $\{z_i(t)\}_{i=1}^m$ are the initial values to be averaged by the consensus algorithm. For $t \geq 1$, the state $z_i(t)$ represents the estimate of the sample average

$$\bar{z} := \frac{1}{m} \sum_{i=1}^m z_i(0)$$

at node i . The objective of consensus is for the state $z_i(t)$ to eventually equal the sample mean of the initial states [37]. Mathematically, this is expressed as $\lim_{t \rightarrow \infty} z_i(t) = \bar{z}, \forall i \in \{1, \dots, m\}$ [37].

¹We denote the positive subset of a set \mathcal{S} by $\mathcal{S}_{>0}$. The nonnegative subset is similarly denoted $\mathcal{S}_{\geq 0}$. The integers are denoted by \mathbb{Z} , and the real numbers by \mathbb{R} .

We leave the discussion of vector-valued states for the following chapter. In this thesis, we restrict our attention to deterministic, synchronous-update consensus algorithms. We assume the following: (i) the communication link topology of the network is fixed and does not change with time, (ii) at each iteration, every node of the network exchanges messages with only its neighbors, and (iii) the communication channels between nodes are noiseless.

Given the above assumptions on communication, one of the most popular algorithms for consensus relies on linear updates [6, 37]. Each node is assigned an index $i \in \{1, \dots, m\}$. Let $z_i(t)$ denote the state of the i^{th} node at iteration t . Each node updates its state by taking a weighted sum of its own state with those of its neighbors [37],

$$z_i(t+1) = w_{ii}z_i(t) + \sum_{j \in \mathcal{N}_i} w_{ij}z_j(t), \quad (2.3)$$

where $w_{ij} > 0 \forall i, j$, $\sum_{k=1}^m w_{ik} = 1$, and \mathcal{N}_i denotes the neighborhood of node i ,

$$\mathcal{N}_i := \{j \mid \{i, j\} \in \mathcal{E}\}.$$

The degree of node i is defined as

$$\deg i := |\mathcal{N}_i|,$$

where $|\cdot|$ represents set cardinality. By this definition, $\deg i$ is the number of neighbors of node i . The weights w_{ij} are designed such that [37]

$$\lim_{t \rightarrow \infty} z_i(t) = \bar{z}.$$

If the state of each node of the network is collected in a vector

$$\mathbf{z}(t) := [z_1(t), \dots, z_m(t)]^\top, \quad (2.4)$$

and the averaging weights w_{ij} are collected in a matrix,

$$[\mathbf{W}]_{ij} := w_{ij}, \quad (2.5)$$

then the above update equation (2.3) can be written in matrix-vector form as [37]

$$\mathbf{z}(t+1) = \mathbf{W}\mathbf{z}(t). \quad (2.6)$$

The design of the weight matrix \mathbf{W} that yields the fastest convergence is a well studied problem; the interested reader is referred to Xiao and Boyd [37]. To converge asymptotically, that is,

$$\lim_{t \rightarrow \infty} \mathbf{z}(t) = \frac{1}{m} \mathbf{1}\mathbf{1}^\top \mathbf{z}(0) = \bar{z} \mathbf{1},$$

the weight matrix must be doubly stochastic, $\sum_i w_{ij} = \sum_j w_{ij} = 1$, and the modulus of its largest eigenvalue must be less than unity [37].

2.3.1 Lossy consensus

By introducing quantization error into the internode messages, the simple linear iteration above (2.6) is not guaranteed to converge. Instead, we use the modified iteration proposed by Frasca *et al.* [14], which allows the sample average to be preserved in the presence of quantization error.

Let $Q : \mathbb{R}^m \rightarrow \mathcal{X}^m$ represent quantization to a finite set of representation levels $\mathcal{X}^m \subset \mathbb{R}^m$ (i.e., $Q(\mathbf{z}(t)) = [Q_1(z_1(t)), \dots, Q_m(z_m(t))]^\top$). The subscripts on Q indicate that each node can use a different quantizer in general. State updates for the case of consensus with quantized messages have been studied by a number of authors, as discussed in Chapter 1. In this thesis,

we use the update proposed by Frasca *et al.* [14], which is

$$z_i(t+1) = z_i(t) + \sum_{j=1}^m w_{ij} (Q_j(z_j(t)) - Q_i(z_i(t))), \quad (2.7)$$

or in matrix-vector form,

$$\mathbf{z}(t+1) = \mathbf{z}(t) + (\mathbf{W} - \mathbf{I}) Q(\mathbf{z}(t)). \quad (2.8)$$

The key advantage of this update is that the average $\frac{1}{m} \sum_{i=1}^m z_i(t)$ of the states $z_i(t)$ is preserved at each step t , despite the presence of quantization error [14]. Defining the quantization error

$$\boldsymbol{\epsilon}(t) := Q(\mathbf{z}(t)) - \mathbf{z}(t), \quad (2.9)$$

the preservation of the sample mean can be seen by rewriting (2.8) as

$$\begin{aligned} \mathbf{z}(t+1) &= \mathbf{z}(t) + (\mathbf{W} - \mathbf{I}) (\mathbf{z}(t) + \boldsymbol{\epsilon}(t)) \\ &= \mathbf{W}\mathbf{z}(t) + (\mathbf{W} - \mathbf{I}) \boldsymbol{\epsilon}(t), \end{aligned} \quad (2.10)$$

and then taking the sample mean [14],

$$\frac{1}{m} \mathbf{1}^\top \mathbf{z}(t+1) = \frac{1}{m} \mathbf{1}^\top \mathbf{W}\mathbf{z}(t) + \frac{1}{m} \mathbf{1}^\top (\mathbf{W} - \mathbf{I}) \boldsymbol{\epsilon}(t).$$

Because \mathbf{W} is doubly stochastic, $\mathbf{1}^\top \mathbf{W} = \mathbf{1}^\top$, and the mean at iteration $t+1$ is equal to the

mean at iteration t [14]:

$$\begin{aligned}
\frac{1}{m} \sum_{i=1}^m z_i(t+1) &= \frac{1}{m} \mathbf{1}^\top \mathbf{z}(t+1) \\
&= \frac{1}{m} \mathbf{1}^\top \mathbf{W} \mathbf{z}(t) + \frac{1}{m} \mathbf{1}^\top (\mathbf{W} - \mathbf{I}) \boldsymbol{\epsilon}(t) \\
&= \frac{1}{m} \mathbf{1}^\top \mathbf{z}(t) + \frac{1}{m} \underbrace{(\mathbf{1}^\top - \mathbf{1}^\top)}_{=0} \boldsymbol{\epsilon}(t) \\
&= \frac{1}{m} \mathbf{1}^\top \mathbf{z}(t) = \frac{1}{m} \sum_{i=1}^m z_i(t).
\end{aligned} \tag{2.11}$$

In this thesis, we model the quantization error $\boldsymbol{\epsilon}(t)$ as additive noise.

Note that the target state of consensus, termed the *average consensus state*, can be written,

$$\mathbf{z}^* := \bar{z} \mathbf{1},$$

which means that $z_i(t) = \bar{z}, \forall i \in \{1, \dots, m\}$. We also define the *average consensus operator* $\frac{1}{m} \mathbf{1} \mathbf{1}^\top$, so that [37]

$$\frac{1}{m} \mathbf{1} \mathbf{1}^\top \mathbf{z}(0) = \mathbf{z}^*.$$

Defining the error from the true sample mean,

$$\begin{aligned}
\mathbf{e}(t) &:= \mathbf{z}(t) - \mathbf{z}^* \\
&= \mathbf{z}(t) - \frac{1}{m} \mathbf{1} \mathbf{1}^\top \mathbf{z}(0),
\end{aligned} \tag{2.12}$$

and noting that the average is preserved over the iterations, (i.e., $\frac{1}{m} \mathbf{1}^\top \mathbf{z}(t) = \frac{1}{m} \mathbf{1}^\top \mathbf{z}(t+1)$), the error $\mathbf{e}(t)$ can be expressed as [14]

$$\mathbf{e}(t) = \left(\mathbf{I} - \frac{1}{m} \mathbf{1} \mathbf{1}^\top \right) \mathbf{z}(t).$$

2.4 Source coding, quantization, and rate-distortion theory

Digital systems rely on the ability to transmit and store information. To perform these tasks reliably, the information takes the form of strings of symbols belonging to some finite alphabet [30]. As a cornerstone of digital communication and storage, coding has been studied quite extensively. In this section, we present some of the fundamental coding and quantization concepts required to understand this thesis.

2.4.1 Coding and information theory

Understanding data compression requires a few concepts from information theory [30], initially developed by Shannon in the 1940s [55]. This theory offers insight into the fundamental limiting performance of communication and compression systems [30]. In this subsection, we focus on the case where the source to be encoded, $\mathbf{x} \in \mathbb{R}^n$, is mapped to a point $\hat{\mathbf{x}}$ in some finite set \mathcal{X}^n by the quantization operator $Q(\cdot)$.

The *entropy* of a scalar RV $x \in \mathcal{X}$ with probability mass function (PMF) $p(x)$ is given by [30]

$$H(x) := -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x).$$

Intuitively, the entropy of an RV is a measure of its uncertainty or information content [30]. Let $\mathbf{x} \in \mathbb{R}^n$ represent a long sequence x , the entries of which are independent and identically distributed (i.i.d.) according to the probability mass function (PMF) $p(x)$. Then $nH(x)$ is the minimum expected binary sequence length required to describe \mathbf{x} without error [30]. That is, if we wish to describe the sequence \mathbf{x} by a string of binary digits $\tilde{\mathbf{x}} \in \{0, 1\}^M$, then a code exists such that the original sequence \mathbf{x} can be noiselessly reconstructed from $\tilde{\mathbf{x}}$ provided that $\mathbb{E}[M] \geq nH(x)$ [30]. To extend this concept to sequences of arbitrary length, we define the *coding rate* per symbol as $\frac{1}{n}\mathbb{E}[M]$, which is the average number of bits used to describe a single source symbol [30].

Another important quantity in information theory is the mutual information between two

RVs. Let x and y be RVs with joint PMF $p(x, y)$ and marginal PMFs $p(x)$ and $p(y)$, respectively.² Then the *mutual information* between $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ is given by [30]

$$I(x; y) := \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}.$$

The utility of these information-theoretic quantities will become apparent during the discussion of RD theory.

2.4.2 Rate-distortion theory

The previous discussion of coding considered only RVs that take on a finite set of values. If we wish to digitally communicate or store a continuous source, it must first be quantized [35, Ch. 1]. The two most basic elements of a quantizer are a set of *representation levels*, which are used to approximate the unquantized signal, and a set of *decision thresholds*, which determine the mapping from the input set to the output set [35, pp. 133–135]. If we imagine the source data as an RV $\mathbf{x} \in \mathbb{R}^n$, then the quantizer $Q : \mathbb{R}^n \rightarrow \mathcal{X}^n$ maps \mathbf{x} to one of finitely many representations $\hat{\mathbf{x}} \in \mathcal{X}^n$ [35]. For real-valued sources, quantizers necessarily introduce a certain expected distortion D into their representation of the input signal [35, pp. 144–145]. This distortion can be quantified using a number of metrics, but for the purpose of this thesis, we use the square error metric [35]

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2,$$

so that the expected distortion per entry or *per letter* is given by $D = \frac{1}{n} \mathbb{E} [\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2]$. In general, using a higher coding rate R results in a lower distortion D , with the drawback of greater communication load. RD theory [29, 30] quantifies the best possible trade-off between coding rate and distortion using the tools of information theory.

²This is an abuse of notation: it would be more correct to write $p_x(x)$ and $p_y(y)$, indicating that the two functions are, in general, different. Here the subscripts are omitted for simplicity.

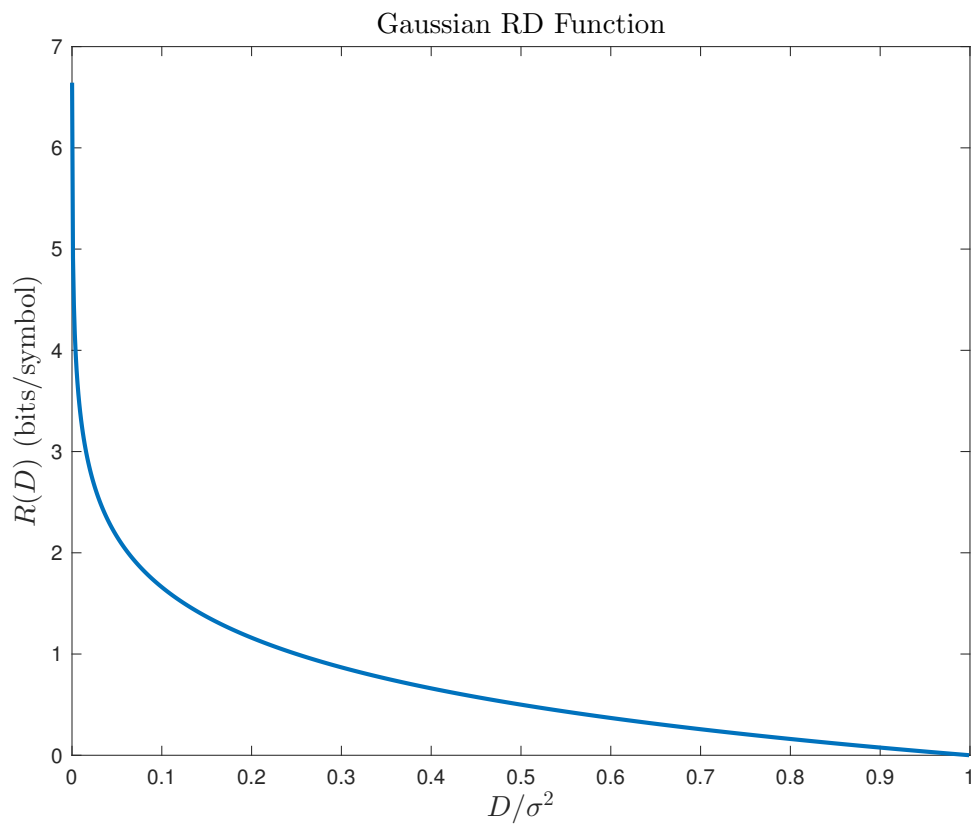


Figure 2.1 The RD function $R(D)$ of a Gaussian memoryless source. The function is a nonincreasing, convex function of the normalized distortion $\frac{D}{\sigma^2}$, where σ^2 is the variance of the Gaussian source. The RD pairs above the curve are achievable, while those below the curve are not.

Assume that we wish to encode a memoryless source, which is represented by the random vector (RVec) $\mathbf{x} \in \mathbb{R}^n$, x_i i.i.d. $\sim f(x_i), \forall i \in \{1, \dots, n\}$. For the remainder of this subsection, we drop the indices on x_i due to the assumption that all entries of \mathbf{x} are i.i.d. The minimum coding rate R required to produce an expected distortion less than or equal to a particular value D is given by the so-called RD function $R(D)$ [29, 30],

$$R(D) := \min_{p(\hat{x}|x): \mathbb{E}[d(x, \hat{x})] \leq D} I(x; \hat{x}).$$

In words, the RD function is the minimum of the mutual information over all possible “test channels” $p(\hat{x}|x)$, subject to the constraint that the expected distortion per entry $\mathbb{E}[d(x, \hat{x})]$ is less than a specified value D [30]. Operationally, the RD function is the minimum number of bits per symbol required to describe a long i.i.d. source within the prescribed distortion D [30]. The computation of a closed form for $R(D)$ is difficult in general; however, $R(D)$ can be computed numerically [57–59]. When a particular quantizer is used, it will often have an RD trade-off curve that differs from $R(D)$, which is a bound on the best possible performance [35]. In this thesis, we term such a trade-off curve for a particular practical quantizer an *operational RD relationship* to avoid ambiguity. An example RD function for the Gaussian case is given in Figure 2.1. Figure 2.2 shows the performance of practical quantizers compared to both the Gaussian RD function and their respective operational RD relationships. Note that the signal-to-distortion ratio (SDR),

$$\text{SDR} := \frac{\text{var}(x)}{D},$$

increases approximately exponentially in the high-rate limit for all practical quantizers in Figure 2.2. This exponential growth is a common feature of Gaussian operational RD relationships that will be exploited in the following chapter.

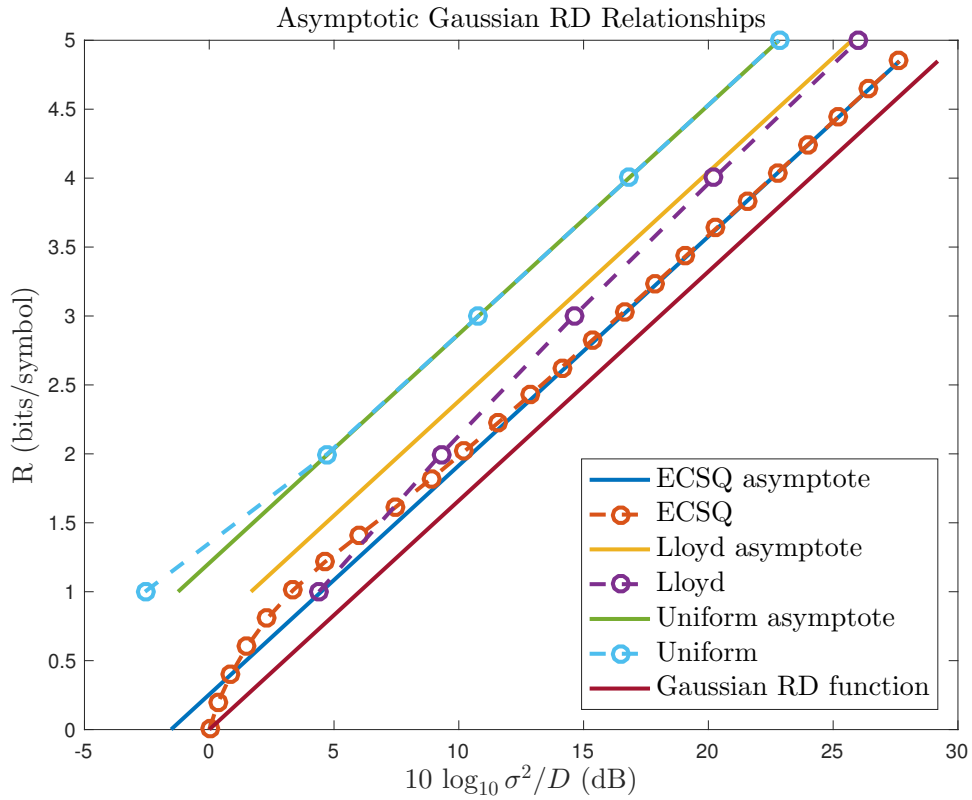


Figure 2.2 Operational RD relationships of practical quantizers: ECSQ, Lloyd-Max, and fixed-rate uniform quantization. Note that all of these approach an asymptote that is a constant offset from the Gaussian RD function as $R \rightarrow \infty$. The ECSQ performance corresponds to the uniform scalar quantizer with a level at zero. The representation level placement may be suboptimal, but it allows for coding rates below one bit per symbol. This figure was inspired by a similar plot in lecture notes from a TU Berlin source coding course [56].

2.4.3 Uniform and nonuniform quantization

From a conceptual standpoint, one of the simplest quantizers is the familiar *uniform scalar quantizer*, for which both the representation levels and the decision thresholds are uniformly spaced [35]. For a given input \mathbf{x} , a uniform scalar quantizer simply rounds each x_i to the representation level \hat{x}_i nearest to x_i [35]. This type of quantizer is frequently encountered in analog-to-digital conversion and various digital signal processing systems. Despite its simplicity, this quantizer performs surprisingly well in certain scenarios.

Consider a source $\mathbf{x} \in \mathbb{R}^n$ that is not uniformly distributed. For a uniform scalar quantizer, each representation level \hat{x}_i will occur with different probability. In this case, we can choose the representation levels based on the input statistics to obtain a low-distortion quantizer [35]. This idea was explored by Lloyd [60] and Max [61], who developed necessary and sufficient conditions for quantizer optimality, given a fixed number of representation levels and a certain source distribution. Lloyd and Max also developed algorithms for achieving an efficient quantizer by iteratively updating the thresholds and representation levels [35]. Because the quantizer is optimized for a particular number of representation levels, Lloyd-Max quantizers are useful for fixed-length codes, where the encoded sequence $\tilde{\mathbf{x}} \in \{0, 1\}^{nR}$. However, more sophisticated coding techniques, such as Huffman coding [62], can be used to approach the previously discussed entropy lower bound on code length if the length M of the binary code sequence $\tilde{\mathbf{x}} \in \{0, 1\}^M$ is allowed to vary with the input [35].

In the case of variable-length codes, we encounter a surprising result. For certain source distributions, uniform scalar quantization followed by entropy coding, called entropy-coded scalar quantization (ECSQ), is the optimal scalar quantizer asymptotically as $R \rightarrow \infty$ [35, 63]. In fact, for the memoryless Gaussian source, ECSQ has been proved to approach the RD limit within 0.255 bits as $R \rightarrow \infty$ [35, 64]. For many memoryless sources, the performance of ECSQ is similar [35, 64]: Farvardin and Modestino [65] showed less than 0.3 bits redundancy over $R(D)$ in the worst case for all the memoryless distributions they studied. Similarly, the operational RD relationships for Lloyd-Max quantization and fixed-rate uniform quantization

approach asymptotes that are a constant offset from the $R(D)$ function in the limit $R \rightarrow \infty$ [35]. The asymptotic offset is illustrated in Figure 2.2.

The curious reader might wonder how the optimal quantizer can differ from the bound if the bound is, in fact, achievable. Better performance can be achieved by quantizing long sequences of the source jointly. Interestingly, this is advantageous even if the source symbols are independent [30]. This technique is termed *vector quantization (VQ)* [35].

Because of the performance improvement associated with VQ, we also consider the multidimensional extension of the scalar quantizer, which is the lattice quantizer [35]. These quantizers approach the RD lower bound for memoryless Gaussian sources [66] as the block dimension approaches infinity, while also permitting the application of dithering, which we will discuss in the following sections.

2.4.4 The quantization noise model and dithering

One of the nicest properties of distributed averaging algorithms is their linearity [14]. At each iteration, a weighted sum of the incoming messages is computed at each node [6, 37]. This allows us to exploit the properties of linear operators, and, for Gaussian-distributed initial states, we can describe the entire network state statistics using linear algebra [21]. However, quantization is inherently nonlinear, which complicates the analysis [14]. Luckily, under certain conditions, the quantization error can be modeled as an additive noise that is uncorrelated with the source [52, 53, 67].

In undithered quantization systems, the conditions for i.i.d. uniform quantization error are rather restrictive [53]. In most cases, these conditions are not met exactly, and the additive quantization noise model is used as an approximation to the true behavior that simplifies system design [52]. In general, the approximation resulting from assuming the additive quantization noise model improves with higher coding rates [52, Sec. 9.5].

More precisely, let $\mathbf{x} \in \mathbb{R}^n$ represent a vector to be quantized. The quantizer $Q(\cdot) : \mathbb{R}^n \rightarrow \mathcal{X}^n$ maps the values of \mathbf{x} to a set of representation levels $\hat{\mathbf{x}} \in \mathcal{X}^n$ [35]. Defining the quantization

error,

$$\boldsymbol{\epsilon} := \hat{\mathbf{x}} - \mathbf{x},$$

the quantization noise model can, for our purposes, be summarized by the following three relationships: (i) the quantization error is zero-mean,

$$\mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0},$$

(ii) the quantization error is uncorrelated with the source,

$$\mathbb{E}[\boldsymbol{\epsilon}\mathbf{x}^\top] = \mathbf{0},$$

and (iii) the quantization error entries $[\boldsymbol{\epsilon}]_i$ and $[\boldsymbol{\epsilon}]_j$, $i \neq j$, are uncorrelated,

$$\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top] = \frac{b^2}{12}\mathbf{I},$$

where b is the quantizer bin size and \mathbf{I} is the identity matrix [52].

In situations that discourage or forbid the use of high rates, a technique called *dithering* can be used to cause the quantization error to be uncorrelated with the source [53]. Dithering is the process of intentionally adding noise to the source signal $\mathbf{x} \in \mathbb{R}^n$ to randomize the quantization error, which is otherwise a deterministic function of the quantizer input [53]. If the i.i.d. noise $\mathbf{w} \in \mathbb{R}^n$ satisfies certain technical conditions, and the receiver subtracts \mathbf{w} from the reconstructed signal $Q(\mathbf{x})$, then the previously discussed additive quantization noise model becomes exact in the sense that [53]

$$\mathbb{E}[(Q(\mathbf{x} + \mathbf{w}) - (\mathbf{x} + \mathbf{w})) \mathbf{x}^\top] = \mathbf{0}$$

and

$$\mathbb{E} \left[(Q(\mathbf{x} + \mathbf{w}) - (\mathbf{x} + \mathbf{w})) (Q(\mathbf{x} + \mathbf{w}) - (\mathbf{x} + \mathbf{w}))^\top \right] = \frac{b^2}{12} \mathbf{I}.$$

Therefore, the quantization error can be modeled as an additive white noise with variance $\frac{b^2}{12}$, which is uncorrelated with the input signal \mathbf{x} [53]. The subtraction of the dither can be achieved in practice using a particular pseudorandom number generator with a seed that is known to both the transmitter and receiver.

Perhaps the simplest dither signal that causes the quantization error to be independent of the source and spectrally white is the uniform distribution $\mathcal{U}(-\frac{b}{2}, \frac{b}{2})$ [53]. Uniformly distributed dither $\mathbf{w} \sim \mathcal{U}(-\frac{b}{2}, \frac{b}{2})$ is used throughout this thesis whenever dithering is mentioned.

Depending on the system of interest, communication may be more or less expensive relative to run time [47]. For instance, consider two extremes of distributed computing: wireless sensor networks (WSNs) [1, 2] and multiserver networks [3]. WSNs have limited battery life and must communicate over wireless channels, which consumes a large amount of power. In many WSN applications, communication is expensive due to the low-power constraints at each node [47]. By contrast, multiserver networks, such as those used in cloud services [3] have less stringent communication requirements, and the run time is more significant to the user [47]. In these networks, the nodes communicate over wired connections and do not have battery power constraints. To produce an optimization that is useful in diverse distributed settings, then, we pursue a general and simple approach that models the relative costs of communication and computation.

The contents of this chapter constitute the main analytical results of this thesis. The contributions of this chapter are twofold: (i) a set of state evolution equations for the network's sufficient statistics are derived under the assumption of initial Gaussian distribution and additive white quantization error, and (ii) it is proven that the problem of minimizing the aggregate rate required to achieve a certain mean square error (MSE) can be posed as a

generalized geometric program (GGP).

The first contribution is minor, since similar approaches to computing the network state statistics were derived in Yildiz and Scaglione [21] and Huang and Hua [28]; however, our derivation allows for the explicit computation of MSE expressions in closed form using the iteration (2.7) introduced in Frasca *et al.* [14]. Yildiz and Scaglione [21] only derived a closed form for the asymptotic MSE as the iteration index $t \rightarrow \infty$, and Huang and Hua optimized with respect to an MSE bound. Our analysis allows an exact optimization to be performed in the finite-iteration (total iterations $T < \infty$) case.

The second contribution allows the problem to be formulated in a well-understood framework. The GGP model can be converted into convex form and efficiently solved using common numerical techniques [31]. Furthermore, due to the convexity of the equivalent transformed optimization problem, the solver is guaranteed to find the globally optimal solution [31]. In addition to the GGP model, we present a heuristic approach for optimization of fixed-rate source coding schemes. This model reduces the size of the optimization problem, which still provides the same solutions as exhaustive search.

3.1 Problem formulation

At every iteration $t \in \{0, \dots, T - 1\}$ of the consensus process, each node $i \in \{1, \dots, m\}$ uses a coding rate $R_i(t)$ to encode its state for transmission to the neighboring nodes.¹ In general, $R_i(t)$ can vary across both nodes and iterations. In practice, we want to terminate the consensus process after a finite number of iterations T . We simplify notation by defining the *coding rate vector*

$$\mathbf{R} := [R_1(0), \dots, R_m(0), \dots, R_1(T - 1), \dots, R_m(T - 1)]^\top. \quad (3.1)$$

¹The choice of zero as the initial iteration is arbitrary.

We denote the expected distortions per entry incurred by transmitting states using the coding rates \mathbf{R} by the *distortion vector*

$$\mathbf{D} := [D_1(0), \dots, D_m(0), \dots, D_1(T-1), \dots, D_m(T-1)]^\top. \quad (3.2)$$

One key quantity we use to determine the cost of running the consensus process is the *aggregate coding rate* [47, 51]:

$$R_{\text{agg}} := \sum_{t=0}^{T-1} \sum_{i=1}^m R_i(t), \quad (3.3)$$

which represents the total coding rate used over the T iterations of the consensus algorithm by all m nodes of the network. To address the trade-off between communication and computation, we borrow the following composite cost function from Zhu and coauthors [47–51] that models both communication and run-time costs:

$$C_{\text{total}}(\mathbf{R}, T) := K_1 R_{\text{agg}} + K_2 T, \quad (3.4)$$

where K_1 and K_2 represent the costs of communication and computation, respectively. Since the iteration (2.8) consists only of a quantization and a weighted sum at each node, the total run-time cost is modeled as linear in the number of iterations. This cost function can be used to model both battery-powered sensor networks and server networks by varying the relative values of K_1 and K_2 [47, 51].

In this chapter, we present minimization strategies of the subcost

$$C_{\text{comm}}(\mathbf{R}, T) := K_1 R_{\text{agg}} \quad (3.5)$$

for fixed- and variable-length codes for (i) Gaussian-distributed sources using a variety of quantizers, and (ii) arbitrarily distributed sources using a specific fixed-rate uniform quantizer. For both proposed schemes, the overall solution to the minimization of (3.4) can

be found by fixing T and minimizing the communication cost $C_{\text{comm}}(\mathbf{R}, T)$, subject to the constraint that \mathbf{R} achieves a target mean square error (MSE) value. Repeating this procedure for $T \in \{T_{\min}, \dots, T_{\max}\}$ is equivalent to minimizing over T and \mathbf{R} jointly.

To efficiently encode the data stored across the network, it is necessary to understand the evolution of its distribution over time. In general, this is not tractable due to the need to track dependences among source elements and transformations of source densities. Instead, we propose an efficient optimization scheme for entropy-coded uniform scalar quantization (ECSQ) [35] of stationary Gaussian states and rate-distortion-optimal (RD-optimal) vector quantization (VQ) [29, 30] of memoryless Gaussian-distributed initial states.² We also use the resulting strategies to design a coding rate optimization heuristic for a particular dithered fixed-rate uniform quantizer [52, 53], for which the expected distortion is the same regardless of source distribution [53].

In this thesis, we assume that the node-to-node communication is taking place in a *broadcast* manner [36], which means that the total rate expended by a transmitting node is independent of the size of its neighborhood. This also simplifies the analysis of the introduced distortion, because each neighbor of a particular node will receive the same quantized message.

3.2 Abstractions for vector-valued data

The concepts presented in this section are included for clarity. We will later present simplifications for the case that each node has a stationary Gaussian source. Although not used directly in our optimization model, the mathematics presented in this section is useful for computing the source statistics across the network in a general setting.

To model the vector-valued data stored at each node during the averaging process, a useful abstraction is introduced, based on similar approaches by Huang and Hua [28]. Let each node $i \in \{1, \dots, m\}$ at iteration $t \in \{0, \dots, T - 1\}$ of consensus have state $\zeta_i(t) \in \mathbb{R}^L$. We represent the

²For the scalar quantization schemes, we assume that the state of each node is a sample from a stationary, ergodic Gaussian random process. The scalar quantizer acts elementwise, so we expect the RD performance of correlated sources with scalar quantization and entropy coding to be the same as the memoryless RD performance.

j^{th} element of the state $\zeta_i(t)$ as

$$\zeta_i^j(t) := [\zeta_i(t)]_j. \quad (3.6)$$

These states are Gaussian random vectors (RVecs) that may be correlated, so that $\mathbb{E} [\zeta_i^k(t)\zeta_j^l(t)] \neq 0$ in general.

To model the state of the entire network, we define the *network state supervector* $\zeta(t) \in \mathbb{R}^{mL}$ as follows:

$$\zeta(t) := \left[\zeta_1^1(t) \cdots \zeta_m^1(t) \mid \zeta_1^2(t) \cdots \zeta_m^2(t) \mid \cdots \mid \zeta_1^L(t) \cdots \zeta_m^L(t) \right]^\top. \quad (3.7)$$

During consensus, each partition of $\zeta(t)$ in (3.7) is averaged independently (i.e., the vector states are averaged over the network nodes elementwise). To represent this elementwise action, we use the *direct sum* operator for matrices [68]:

$$\bigoplus_{i=1}^n \mathbf{A}_i := \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_n \end{bmatrix}.$$

Therefore, the operator \bigoplus creates a block-diagonal matrix from the matrices $\{\mathbf{A}_i\}_{i=1}^n$. Note that $\bigoplus_{i=1}^n \mathbf{A}_i$ is shorthand for $\mathbf{A}_1 \oplus \mathbf{A}_2 \oplus \cdots \oplus \mathbf{A}_n$. This is an alternative to the Kronecker product notation and the matrix structure introduced by Huang and Hua [28].

Defining the weighted averaging matrix $\mathbf{\Omega} \in \mathbb{R}^{mL \times mL}$:

$$\mathbf{\Omega} := \bigoplus_{l=1}^L \mathbf{W} \quad (3.8)$$

as the operator equivalent to the weight matrix \mathbf{W} (2.5) for vector consensus, the state update

equivalent to (2.8) for vector consensus is

$$\zeta(t+1) = \zeta(t) + (\mathbf{\Omega} - \mathbf{I})Q(\zeta(t)), \quad (3.9)$$

where

$$Q(\zeta(t)) := \left[Q_1(\zeta_1^1(t)) \cdots Q_m(\zeta_m^1(t)) \mid Q_1(\zeta_1^2(t)) \cdots Q_m(\zeta_m^2(t)) \mid \cdots \mid Q_1(\zeta_1^L(t)) \cdots Q_m(\zeta_m^L(t)) \right]^\top.$$

As in the scalar consensus case, we define the quantization error,

$$\delta(t) := Q(\zeta(t)) - \zeta(t). \quad (3.10)$$

This is the vector consensus analog to $\epsilon(t)$ (2.9). Using the preceding definitions, the state update (3.9) can be rewritten as

$$\zeta(t+1) = \mathbf{\Omega}\zeta(t) + (\mathbf{\Omega} - \mathbf{I})\delta(t). \quad (3.11)$$

The goal of vector consensus is to iteratively drive each $\zeta_i(t)$ to the sample mean of the initial states $\zeta_i(0)$. Define $\bar{\zeta} \in \mathbb{R}^L$ as the sample mean (over nodes) of the initial states,

$$\bar{\zeta} := \frac{1}{m} \sum_{i=1}^m \zeta_i(0).$$

The goal of consensus is thus

$$\lim_{t \rightarrow \infty} \zeta_i(t) = \bar{\zeta}, \quad \forall i \in \{1, \dots, m\}.$$

We term the state supervector $\zeta^* \in \mathbb{R}^{mL}$ for which $\zeta_i(t) = \bar{\zeta}, \forall i \in \{1, \dots, m\}$, the *vector average*

consensus state. This state has the following form:

$$\zeta^* = \left[[\bar{\zeta}]_1 \cdots [\bar{\zeta}]_1 \mid [\bar{\zeta}]_2 \cdots [\bar{\zeta}]_2 \mid \cdots \mid [\bar{\zeta}]_L \cdots [\bar{\zeta}]_L \right]^\top.$$

The *vector average consensus operator* is defined as

$$\mathbf{M} := \bigoplus_{l=1}^L \frac{1}{m} \mathbf{1}\mathbf{1}^\top, \quad (3.12)$$

so that

$$\mathbf{M}\zeta(0) = \zeta^*.$$

The *state estimation error*, which is the error in the current estimate of the true average, is defined as

$$\boldsymbol{\eta}(t) := \zeta(t) - \mathbf{M}\zeta(0). \quad (3.13)$$

Because $\mathbf{M}\zeta(t+1) = \mathbf{M}\zeta(t) = \mathbf{M}\zeta(0)$ [this follows from (2.11)], the error (3.13) can also be expressed as

$$\boldsymbol{\eta}(t) = (\mathbf{I} - \mathbf{M}) \zeta(t).$$

3.3 Assumptions and analytical results

In this section, we present certain assumptions and results on the distribution of $\zeta(t)$ (3.7), the network graph structure (i.e., $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ (2.1)), and the quantization error $\delta(t)$ (3.10). We also present relationships among the MSE, the variance of $\zeta_i^j(t)$ (3.6), and the distortions introduced, \mathbf{D} (3.2).

3.3.1 Additive noise model

Following Frasca *et al.* [14] and Yildiz and Scaglione [21, 22], we adopt the additive model for quantization noise [52]. This model assumes that the errors introduced by the quantization process are uncorrelated with the quantizer input. A formal statement of the corresponding assumptions is provided.

We assume that the quantization error $\delta(t)$ is spectrally white:

Assumption 1 *The quantization error element $\delta_i^k(t)$ corresponding to $\zeta_i^k(t)$ (3.6) (i.e., quantization error at the k^{th} element of the i^{th} node at iteration t) is uncorrelated with all other quantization error elements: if $j \neq i$, $l \neq k$, or $\tau \neq t$, $\mathbb{E} [\delta_i^k(t) \delta_j^l(\tau)] = \mathbb{E} [\delta_i^k(t)] \mathbb{E} [\delta_j^l(\tau)] = 0$.*

We further assume that the quantization error is independent of the source to be quantized:

Assumption 2 *The quantization error $\delta(t)$ is orthogonal to the state supervector $\zeta(t)$ with respect to node index (i.e., $\mathbb{E} [\zeta(t) \delta^\top(t)] = \mathbf{0}$).*

These assumptions are well motivated for uniform quantizers with high coding rate [52], or for uniform quantizers using dither at arbitrary rates [53]. In general, probability-density-optimized (pdf-optimized) quantizers will have quantization error that is linearly correlated with the input signal (i.e., $\mathbb{E} [x(Q(x) - x)] = -D$, where x is a scalar RV to be quantized, $Q(x)$ is the result of quantization, and D is the expected distortion) [35, p. 182, 69]. Although dither for nonuniform quantization is not as well studied as for uniform quantization, the problem of correlated quantization noise may be alleviated by dithering using a compander as shown in a recent paper by Aykol and Rose [70]. For quantizers with nonuniform level spacing, Assumptions 1 and 2 are not rigorously justified, but numerical results are included to assess performance.

3.3.2 Preservation of Gaussian distribution

In this work, we assume that the initial state supervector $\zeta(t)$ (3.7) has a multidimensional Gaussian distribution. Under certain conditions on the connectivity of the network, the

distribution of the state supervector $\zeta(t)$ will be Gaussian for all $t \in \{0, \dots, T-1\}$.

Lemma 9 *Assume the technical conditions of Assumption 1 hold. Further assume that the degree of each node is bounded from below by a monotone increasing function $g(m)$ of the network size m . Then the state supervector $\zeta(t)$, $t \in \{0, \dots, T-1\}$, will be Gaussian distributed.*

Proof: The proof follows from a simple application of the central limit theorem (CLT). At the beginning of iteration zero, the random vector $\zeta(0)$ is jointly Gaussian by assumption. The state supervector is updated as (3.11),

$$\zeta(t+1) = \mathbf{\Omega}\zeta(t) + (\mathbf{\Omega} - \mathbf{I})\delta(t).$$

If the elements of $\zeta(t)$ are jointly Gaussian, then the term $\mathbf{\Omega}\zeta(t)$ will also be Gaussian distributed, because it is a linear transformation of jointly Gaussian RVs. As $m \rightarrow \infty$, $|\mathcal{N}_i| \geq g(m) \rightarrow \infty, \forall i$. Define the “aggregate quantization error” as

$$\delta_{\text{agg}}(t) := (\mathbf{\Omega} - \mathbf{I})\delta(t).$$

Because the degree of each node $\deg i \rightarrow \infty, \forall i \in \{1, \dots, m\}$, each element $[\delta_{\text{agg}}(t)]_k$ of $\delta_{\text{agg}}(t)$ is a sum of many independent RVs. By the CLT and our assumption of zero-mean independent quantization error, $\delta_{\text{agg}}(t)$ converges in distribution to $\mathcal{N}(\mathbf{0}, (\mathbf{\Omega} - \mathbf{I})\mathbf{\Sigma}_{\delta}(t)(\mathbf{\Omega} - \mathbf{I}))$, where $\mathbf{\Sigma}_{\delta}(t) := \mathbb{E}[\delta(t)\delta^{\top}(t)]$ is the covariance matrix of the quantization error $\delta(t)$. Because both terms of (3.11) are Gaussian distributed, if $\zeta(t)$ is Gaussian, then $\zeta(t+1)$ will be Gaussian. Therefore, by our assumption of initial Gaussian state, the state $\zeta(t)$ will be Gaussian distributed for all $t \in \{0, \dots, T-1\}$. \square

Note: For high-dimensional VQ, the quantization noise approaches a white Gaussian process [66], and the above node degree bounding assumptions are unnecessary.

3.3.3 State evolution for vector consensus

Because the network state supervector $\zeta(t)$ (3.7) is Gaussian for all t , the sufficient statistics of $\zeta(t)$ are a mean vector and covariance matrix. Likewise, because the error $\eta(t)$ (3.13) is a linear transformation of $\zeta(t)$, it has a multidimensional Gaussian distribution. By deriving formulas for the evolution of these statistics, we can minimize the cost function (3.5) to find the optimal source coding rates \mathbf{R}_{opt} .

We define the mean of $\zeta(t)$ (3.7) as

$$\boldsymbol{\mu}_\zeta(t) := \mathbb{E} [\zeta(t)], \quad (3.14)$$

and the mean of $\eta(t)$ (3.13) as

$$\boldsymbol{\mu}_\eta(t) := \mathbb{E} [\eta(t)]. \quad (3.15)$$

The covariance matrices of $\zeta(t)$ and $\eta(t)$ are defined as

$$\boldsymbol{\Sigma}_\zeta(t) := \mathbb{E} [\zeta(t)\zeta^\top(t)] - \boldsymbol{\mu}_\zeta(t)\boldsymbol{\mu}_\zeta^\top(t), \quad (3.16)$$

and

$$\boldsymbol{\Sigma}_\eta(t) := \mathbb{E} [\eta(t)\eta^\top(t)] - \boldsymbol{\mu}_\eta(t)\boldsymbol{\mu}_\eta^\top(t), \quad (3.17)$$

respectively.

Lemma 10 *The quantities (3.14)–(3.17) can be computed as follows,*

$$\boldsymbol{\mu}_\zeta(t) = \boldsymbol{\Omega}^t \boldsymbol{\mu}_\zeta(0), \quad (3.18)$$

$$\boldsymbol{\mu}_\eta(t) = (\mathbf{I} - \mathbf{M}) \boldsymbol{\mu}_\zeta(0), \quad (3.19)$$

$$\boldsymbol{\Sigma}_\zeta(t) = \boldsymbol{\Omega}^t \boldsymbol{\Sigma}_\zeta(0) \boldsymbol{\Omega}^t + \sum_{s=0}^{t-1} \boldsymbol{\Omega}^{t-s-1} (\boldsymbol{\Omega} - \mathbf{I}) \boldsymbol{\Sigma}_\delta(s) (\boldsymbol{\Omega} - \mathbf{I}) \boldsymbol{\Omega}^{t-s-1}, \quad (3.20)$$

and

$$\boldsymbol{\Sigma}_\eta(t) = (\mathbf{I} - \mathbf{M}) \boldsymbol{\Sigma}_\zeta(t) (\mathbf{I} - \mathbf{M}), \quad (3.21)$$

where $\boldsymbol{\Sigma}_\delta(t) := \mathbb{E} [\boldsymbol{\delta}(t) \boldsymbol{\delta}^\top(t)]$ is the quantization error (3.10) covariance matrix.

Proof: The state update used in vector consensus is (3.9)

$$\boldsymbol{\zeta}(t+1) = \boldsymbol{\zeta}(t) + (\boldsymbol{\Omega} - \mathbf{I}) \mathcal{Q}(\boldsymbol{\zeta}(t)),$$

where $\boldsymbol{\zeta}(t)$ is the state supervector (3.7). The mean $\boldsymbol{\mu}_\zeta(t)$ can be written in terms of the weight matrix $\boldsymbol{\Omega}$ (3.8) and the quantization error $\boldsymbol{\delta}(t)$ (3.10) as

$$\boldsymbol{\mu}_\zeta(t) = \mathbb{E} [\boldsymbol{\Omega} \boldsymbol{\zeta}(t-1)] + \mathbb{E} [(\boldsymbol{\Omega} - \mathbf{I}) \boldsymbol{\delta}(t-1)].$$

Because $\boldsymbol{\Omega}$ and $(\boldsymbol{\Omega} - \mathbf{I})$ are constants, this expression is equivalent to

$$\boldsymbol{\mu}_\zeta(t) = \boldsymbol{\Omega} \mathbb{E} [\boldsymbol{\zeta}(t-1)] + (\boldsymbol{\Omega} - \mathbf{I}) \mathbb{E} [\boldsymbol{\delta}(t-1)].$$

Recalling that the mean of the quantization error is assumed to be zero,

$$\boldsymbol{\mu}_\zeta(t) = \boldsymbol{\Omega} \boldsymbol{\mu}_\zeta(t-1).$$

By recursion on the above equation,

$$\boldsymbol{\mu}_\zeta(t) = \boldsymbol{\Omega}^t \boldsymbol{\mu}_\zeta(0),$$

which is (3.18).

The covariance $\boldsymbol{\Sigma}_\zeta(t)$ can be written

$$\begin{aligned} \boldsymbol{\Sigma}_\zeta(t) = & \mathbb{E} \left[(\boldsymbol{\Omega}\zeta(t-1) + (\boldsymbol{\Omega} - \mathbf{I})\delta(t-1)) (\boldsymbol{\Omega}\zeta(t-1) + (\boldsymbol{\Omega} - \mathbf{I})\delta(t-1))^\top \right] \\ & - \mathbb{E} [\boldsymbol{\Omega}\zeta(t-1)] \mathbb{E} [(\boldsymbol{\Omega}\zeta(t-1))^\top]. \end{aligned}$$

Noting that $\boldsymbol{\Omega}$ is a constant, and that $\boldsymbol{\Omega} = \boldsymbol{\Omega}^\top$, expanding the above expression gives³

$$\begin{aligned} \boldsymbol{\Sigma}_\zeta(t) = & \mathbb{E} \left[\boldsymbol{\Omega}\zeta(t-1)\zeta^\top(t-1)\boldsymbol{\Omega} \right] + \mathbb{E} \left[(\boldsymbol{\Omega} - \mathbf{I})\delta(t-1)\zeta^\top(t-1)\boldsymbol{\Omega} \right] \\ & + \mathbb{E} \left[\boldsymbol{\Omega}\zeta(t-1)\delta^\top(t-1)(\boldsymbol{\Omega} - \mathbf{I}) \right] + \mathbb{E} \left[(\boldsymbol{\Omega} - \mathbf{I})\delta(t-1)\delta^\top(t-1)(\boldsymbol{\Omega} - \mathbf{I}) \right] \\ & - \boldsymbol{\Omega} \mathbb{E} [\zeta(t-1)] \mathbb{E} [\zeta^\top(t-1)] \boldsymbol{\Omega}. \end{aligned}$$

By Assumption 2, the second and third terms above are zero, so that

$$\boldsymbol{\Sigma}_\zeta(t) = \boldsymbol{\Omega} \mathbb{E} \left[\zeta(t-1)\zeta^\top(t-1) \right] \boldsymbol{\Omega} + (\boldsymbol{\Omega} - \mathbf{I})\boldsymbol{\Sigma}_\delta(t-1)(\boldsymbol{\Omega} - \mathbf{I}) - \boldsymbol{\Omega}\boldsymbol{\mu}_\zeta(t-1)\boldsymbol{\mu}_\zeta^\top(t-1)\boldsymbol{\Omega}.$$

Grouping terms,

$$\boldsymbol{\Sigma}_\zeta(t) = \boldsymbol{\Omega} \left(\mathbb{E} \left[\zeta(t-1)\zeta^\top(t-1) \right] - \boldsymbol{\mu}_\zeta(t-1)\boldsymbol{\mu}_\zeta^\top(t-1) \right) \boldsymbol{\Omega} + (\boldsymbol{\Omega} - \mathbf{I})\boldsymbol{\Sigma}_\delta(t-1)(\boldsymbol{\Omega} - \mathbf{I}).$$

Therefore,

$$\boldsymbol{\Sigma}_\zeta(t) = \boldsymbol{\Omega}\boldsymbol{\Sigma}_\zeta(t-1)\boldsymbol{\Omega} + (\boldsymbol{\Omega} - \mathbf{I})\boldsymbol{\Sigma}_\delta(t-1)(\boldsymbol{\Omega} - \mathbf{I}).$$

³The weight matrix \mathbf{W} is symmetric for the optimal weights found by Xiao and coauthors [32, 37] and other simple weighting strategies, such as Metropolis-Hastings [32] and max-degree [37].

Performing recursion on the above equation gives

$$\boldsymbol{\Sigma}_\zeta(t) = \boldsymbol{\Omega}^t \boldsymbol{\Sigma}_\zeta(0) \boldsymbol{\Omega}^t + \sum_{s=0}^{t-1} \boldsymbol{\Omega}^{t-s-1} (\boldsymbol{\Omega} - \mathbf{I}) \boldsymbol{\Sigma}_\delta(s) (\boldsymbol{\Omega} - \mathbf{I}) \boldsymbol{\Omega}^{t-s-1},$$

which is (3.20).

Because the error from the true sample mean is $\boldsymbol{\eta}(t) = (\mathbf{I} - \mathbf{M}) \boldsymbol{\zeta}(t)$ (from (3.12) and (3.13)) and $(\mathbf{I} - \mathbf{M}) = (\mathbf{I} - \mathbf{M})^\top$, it follows that

$$\boldsymbol{\mu}_\eta(t) = (\mathbf{I} - \mathbf{M}) \boldsymbol{\mu}_\zeta(t),$$

which is (3.19), and

$$\boldsymbol{\Sigma}_\eta(t) = (\mathbf{I} - \mathbf{M}) \boldsymbol{\Sigma}_\zeta(t) (\mathbf{I} - \mathbf{M}),$$

which is (3.21). Similar steps are taken to obtain the simplified forms of these expressions for scalar consensus. \square

The equations (3.18)–(3.21) can be simplified for optimization modeling in the case that the states $\zeta_i(0)$ are stationary Gaussian processes. In this scenario, the distortion incurred by uniform scalar quantization at node i and iteration t can be modeled as the distortion from quantization of an independent and identically distributed (i.i.d.) source under certain conditions [52, Sec. 9.5]. In particular, the RD performance can be described well by the marginal variance of the state $\zeta_i(t)$,

$$v_i(\mathbf{D}, t) = \mathbb{E} \left[\left(\zeta_i^1(t) \right)^2 \right] - \mathbb{E} \left[\zeta_i^1(t) \right]^2 = \dots = \mathbb{E} \left[\left(\zeta_i^L(t) \right)^2 \right] - \mathbb{E} \left[\zeta_i^L(t) \right]^2,$$

provided that the correlations $\mathbb{E} [\zeta_i^k(t) \zeta_i^l(t)]$, $k \neq l$, are not extreme and the quantizer bin size b is not too coarse [52, Sec. 9.5]. For this scenario, we can consider the scalar states $z_i(t) \in \mathbb{R}$ (2.2), $i \in \{1, \dots, m\}$ and the state vector $\mathbf{z}(t) \in \mathbb{R}^m$ (2.4) in place of $\zeta_i(t) \in \mathbb{R}^L$, $i \in \{1, \dots, m\}$ and $\boldsymbol{\zeta}(t) \in \mathbb{R}^{mL}$, respectively, where $z_i(t)$ is distributed according to the marginal pdf $f(\zeta_i^1(t))$,

$\forall i \in \{1, \dots, m\}$.

We define the means of $\mathbf{z}(t)$ (2.4) and $\mathbf{e}(t)$ (2.12),

$$\boldsymbol{\mu}_{\mathbf{z}}(t) := \mathbb{E}[\mathbf{z}(t)]$$

and

$$\boldsymbol{\mu}_{\mathbf{e}}(t) := \mathbb{E}[\mathbf{e}(t)],$$

and the covariances of $\mathbf{z}(t)$ and $\mathbf{e}(t)$,

$$\boldsymbol{\Sigma}_{\mathbf{z}}(t) := \mathbb{E}[\mathbf{z}(t)\mathbf{z}^\top(t)] - \boldsymbol{\mu}_{\mathbf{z}}(t)\boldsymbol{\mu}_{\mathbf{z}}^\top(t) \quad (3.22)$$

and

$$\boldsymbol{\Sigma}_{\mathbf{e}}(t) := \mathbb{E}[\mathbf{e}(t)\mathbf{e}^\top(t)] - \boldsymbol{\mu}_{\mathbf{e}}(t)\boldsymbol{\mu}_{\mathbf{e}}^\top(t).$$

The equations (3.18)–(3.21) reduce to

$$\boldsymbol{\mu}_{\mathbf{z}}(t) = \mathbf{W}^t \boldsymbol{\mu}_{\mathbf{z}}(0), \quad (3.23)$$

$$\boldsymbol{\mu}_{\mathbf{e}}(t) = \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \mathbf{W}^t \boldsymbol{\mu}_{\mathbf{e}}(0), \quad (3.24)$$

$$\boldsymbol{\Sigma}_{\mathbf{z}}(t) = \mathbf{W}^t \boldsymbol{\Sigma}_{\mathbf{z}}(0) \mathbf{W}^t + \sum_{s=0}^{t-1} \mathbf{W}^{t-s-1} (\mathbf{W} - \mathbf{I}) \boldsymbol{\Sigma}_{\mathbf{e}}(s) (\mathbf{W} - \mathbf{I}) \mathbf{W}^{t-s-1}, \quad (3.25)$$

and

$$\boldsymbol{\Sigma}_{\mathbf{e}}(t) = \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \boldsymbol{\Sigma}_{\mathbf{z}}(t) \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right). \quad (3.26)$$

Given the above state-evolution equations (3.23)–(3.26), we can determine the variance $v_i(\mathbf{D}, t)$ and MSE, $\text{MSE}_i(\mathbf{D}, t)$, at each node $i \in \{1, \dots, m\}$. The source variance at node i and iteration t is defined as

$$v_i(\mathbf{D}, t) := \mathbb{E} [z_i^2(t)] - \mathbb{E} [z_i(t)]^2. \quad (3.27)$$

Let us discuss our state estimation error (2.12) metrics. We define the *node MSE*, which represents the MSE at node i and iteration t , as

$$\text{MSE}_i(\mathbf{D}, t) := \mathbb{E} [e_i^2(t)], \quad (3.28)$$

where $e_i(t)$ is the i^{th} element of $\mathbf{e}(t)$ (2.12). We now turn to quantifying the estimation error over the entire network. For this purpose, we define the *network MSE*, which is the arithmetic mean of the node MSEs, $\{\text{MSE}_i(\mathbf{D}, t)\}_{i=1}^m$,

$$\text{MSE}(\mathbf{D}, t) := \frac{1}{m} \sum_{i=1}^m \text{MSE}_i(\mathbf{D}, t). \quad (3.29)$$

Using these definitions, we present the following mathematical relationships, which we term the *state evolution equations*. These equations allow us to perform the optimization of the coding rate vector \mathbf{R} using the cost function (3.5).

Lemma 11 *Given the above equations (3.25) and (3.26) for the state and error covariances, the variance at node i at iteration t (3.27) is given by*

$$v_i(\mathbf{D}, t) = [\boldsymbol{\Sigma}_{\mathbf{z}}(t)]_{ii}, \quad (3.30)$$

the node MSE (3.28) is given by

$$\text{MSE}_i(\mathbf{D}, t) = \left[\boldsymbol{\Sigma}_e(t) + \boldsymbol{\mu}_e(t)\boldsymbol{\mu}_e^\top(t) \right]_{ii}, \quad (3.31)$$

and the network MSE (3.29) is given by

$$\text{MSE}(\mathbf{D}, t) = \frac{1}{m} \text{tr} \left(\boldsymbol{\Sigma}_e(t) + \boldsymbol{\mu}_e(t)\boldsymbol{\mu}_e^\top(t) \right), \quad (3.32)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix.

The derivations of equations (3.23)–(3.26) take almost identical steps to those in the proof of Lemma 10. An example of the $\text{MSE}_i(t)$ predicted by (3.31) is given in Figure 3.1.

3.4 Optimization via generalized geometric programming

The key insight of this work is the ability to pose the optimization of the coding rate vector \mathbf{R} as a generalized geometric program (GGP), whose global optimum can be found efficiently [71]. Using variable-length codes [62], the coding rates

$$R_i(t) \in \mathbb{R}_{>0}, \quad \forall i \in \{1, \dots, m\}, \quad \forall t \in \{0, \dots, T-1\},$$

in contrast to the fixed-length coding case where $R_i(t) \in \mathbb{Z}_{>0}, \forall i \in \{1, \dots, m\}, \forall t \in \{0, \dots, T-1\}$. For the case of variable-length codes, we propose a high-rate approximation to the operational RD relationship to find an efficient rate vector \mathbf{R} capable of achieving a target value of $\text{MSE}(\mathbf{D}, T)$ (3.32).⁴ This modeling strategy will later be used to design an optimization procedure for fixed-length coding, where $\mathbf{R} \in \mathbb{Z}_{>0}^{mT}$.

⁴Note that the final network MSE corresponds to iteration index T , and not $T-1$. This is because $\text{MSE}(\mathbf{D}, t)$ is the network MSE *after the end of* t iterations. The quantity $\text{MSE}(\mathbf{D}, t-1)$ would thus correspond to the end of the $(t-1)$ th iteration, or the beginning of the t th iteration.

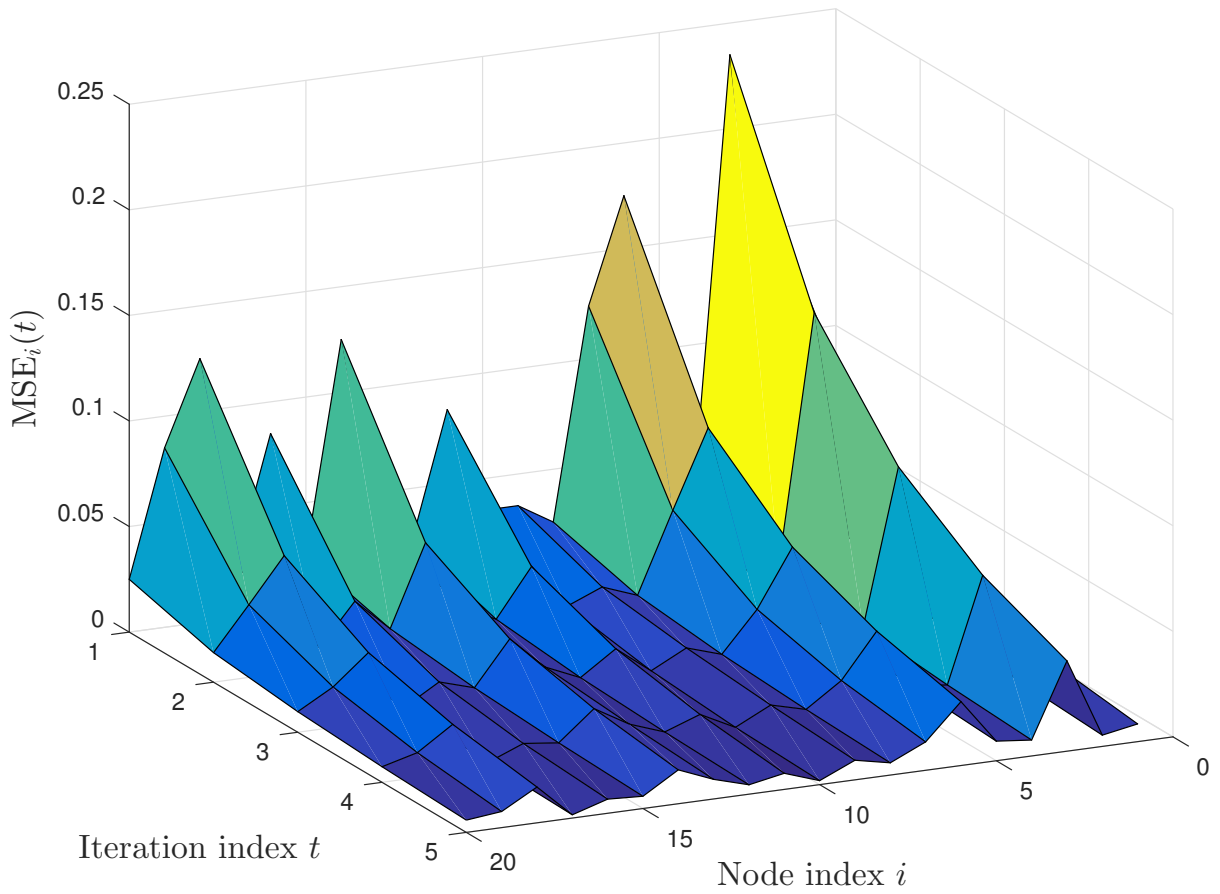


Figure 3.1 Surface showing the evolution of $MSE_i(t)$ for the case of lossless consensus. Note that the value of $MSE_i(t)$ is not equal across all nodes for a particular iteration, and the evolution of $MSE_i(t)$ is heavily dependent on network topology. Also observe that the MSE decreases monotonically at each node as $t \rightarrow \infty$. The variance $v_i(t)$ evolves in a similar fashion.

3.4.1 Cost function formulation

For the quantizers of interest operating on Gaussian sources, the operational RD relationship (the reader is reminded that this term was defined in Section 2.4.2) in the high-rate regime has the form

$$R(D) \approx \begin{cases} \frac{1}{2} \log_2 \left(\frac{\sigma^2}{D} \right) + R_c, & D \in (0, \sigma^2 D_{\max}] \\ 0, & \text{otherwise} \end{cases}, \quad (3.33)$$

where σ^2 represents the variance of the data to be encoded and D_{\max} is some constant [63]. In some cases, such as infinite-dimensional VQ with Gaussian memoryless sources and dithered scalar uniform quantization [66], the relationship (3.33) holds for all rates. Note that (3.33) is equal to the Gaussian RD function

$$R_G(D) = \begin{cases} \frac{1}{2} \log_2 \left(\frac{\sigma^2}{D} \right), & D \in (0, \sigma^2] \\ 0, & \text{otherwise} \end{cases},$$

with a modified domain and offset. In the following discussion, it is easier to consider the inverse to the operational RD relationship, denoted $D(R)$.

For uniform scalar quantization of a memoryless Gaussian source followed by block entropy coding, the true operational RD relationship is approximated well in the high-rate regime by the Gaussian RD function plus 0.255 bits [35, p. 302, 64]. More generally, the operational distortion-rate relationship for ECSQ at high rates is determined by

$$D(R) \approx \frac{1}{12} 2^{2h(x)} 2^{-2R}, \quad (3.34)$$

where $h(x)$ is the differential entropy of the source x [64]. Similar results hold for Lloyd-Max quantization [64, 72] and fixed-rate uniform quantization [64]. For Lloyd-Max quantization of memoryless Gaussian sources at high resolution, the operational RD relationship is approxi-

mated by

$$D(R) \approx \frac{1}{12} \left[\int_{-\infty}^{\infty} f^{1/3}(x) dx \right]^3 2^{-2R}, \quad (3.35)$$

where $f(x)$ is the probability distribution of the source x [64, 72]. For Gaussian densities, both (3.34) and (3.35) scale with the source variance σ^2 . This scaling can be proven by straightforward computation.

As in Section 2.4.2, we denote the representation levels of the N -codeword quantizer as $\{\hat{x}_i\}_{i=1}^N$. The maximum representation level of the quantizer is denoted \hat{x}_{\max} , and the minimum representation level is denoted \hat{x}_{\min} . For subtractively dithered fixed-rate uniform quantization of any source at all rates with $\Delta := \frac{1}{\sigma}(\hat{x}_{\max} - \hat{x}_{\min})$ (neglecting overload distortion) [64],

$$D(R) = \frac{\Delta^2 \sigma^2}{12} 2^{-2R}.$$

In our case, the source variance is $v_i(\mathbf{D}, t)$, defined in (3.30), which is a function of the initial state vector covariance $\Sigma_{\mathbf{z}}(0)$ (3.22) and the distortion vector \mathbf{D} (3.2). It evolves as described by (3.30). The operational RD relationship at all nodes $i \in \{1, \dots, m\}$ and iterations $t \in \{0, \dots, T-1\}$ can be expressed as follows. Any Gaussian scalar quantizer will have a maximum possible distortion for a particular source. Because the distortion is proportional to the source variance $v_i(\mathbf{D}, t)$ in the Gaussian case, we consider the maximum distortion when $v_i(\mathbf{D}, t) = 1$. Letting D_{\max} denote this maximum possible distortion for a unit-variance Gaussian source,

$$R_i(\mathbf{D}, t) = \begin{cases} \frac{1}{2} \log_2 \left(\frac{v_i(\mathbf{D}, t)}{D_i(t)} \right) + R_c, & D_i(t) \in (0, v_i(\mathbf{D}, t) D_{\max}] \\ 0, & D_i(t) > v_i(\mathbf{D}, t) D_{\max} \end{cases}.$$

This operational RD relationship is equivalent to

$$\begin{aligned} R_i(\mathbf{D}, t) &= \frac{1}{2} \max \left\{ \log_2 \left(\frac{v_i(\mathbf{D}, t)}{D_i(t)} \right), c \right\} + R_c \\ &= \frac{1}{2} \log_2 \left(\max \left\{ \frac{v_i(\mathbf{D}, t)}{D_i(t)}, k \right\} \right) + R_c, \quad D_i(t) > 0, \end{aligned} \quad (3.36)$$

where c and k are terms (constant in \mathbf{D} and t) that depend on D_{\max} . Concretely, $k = 2^{-2R_c} > 0$.

Given a number of iterations T , the goal is to minimize the aggregate coding rate (3.3), subject to a constraint on the final MSE, $\text{MSE}(\mathbf{D}, T) \leq \text{MSE}^*$ (3.32). For a particular number of iterations T and target MSE^* , the minimum number of iterations required to achieve that MSE, $T_{\min} = \arg \min_T \{T \mid \text{MSE}_{\text{lossless}}(\mathbf{D}, T) < \text{MSE}^*\}$, can be readily obtained using the state-evolution equations (3.23)–(3.26). More formally, using the operational RD relationship (3.36), the optimization problem is

$$\begin{aligned} \underset{\mathbf{D}}{\text{minimize}} \quad & \sum_{t=0}^{T-1} \sum_{i=1}^m \frac{1}{2} \log_2 \left(\max \left\{ \frac{v_i(\mathbf{D}, t)}{D_i(t)}, k \right\} \right) + R_c, \\ \text{subject to} \quad & \text{MSE}(\mathbf{D}, T) \leq \text{MSE}^*, \\ & D_i(t) > 0, \quad \forall i, t. \end{aligned}$$

Note that the above optimization can be rewritten as

$$\begin{aligned} \underset{\mathbf{D}}{\text{minimize}} \quad & \ln \left(\prod_{t=0}^{T-1} \prod_{i=1}^m \max \left\{ \frac{v_i(\mathbf{D}, t)}{D_i(t)}, k \right\} \right), \\ \text{subject to} \quad & \text{MSE}(\mathbf{D}, T) \leq \text{MSE}^*, \\ & D_i(t) > 0, \quad \forall i, t, \end{aligned} \quad (3.37)$$

where the constants $\frac{1}{2 \ln(2)}$ and R_c are omitted, because they make no difference to the optimum point of the minimization. We will now introduce GGP and show that the optimization (3.37) reduces to such a problem.

3.4.2 Basics of generalized geometric programming

The following information can be found in the book *Convex Optimization* by Boyd and Vandenberghe [31]. For this tutorial, we stay close to the authors' original notation. In the language of geometric programming (GP), a function of the form

$$f(x) = cx_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}, \quad c > 0, x_i > 0 \quad \forall i, a_i \in \mathbb{R} \quad \forall i,$$

is called a *monomial* [31, 71]. Similarly, a function of the form

$$f(x) = \sum_{i=1}^k g_i(x_1, \dots, x_n),$$

where the $g_i(x_1, \dots, x_n)$ are monomials, is termed a *posynomial* (not to be confused with the word *polynomial* from algebra) [71]. The word posynomial is intended to convey that the coefficients and values of the variables are constrained to be positive [71]. Boyd and Vandenberghe [31] state a certain number of closure properties:

Posynomials are closed under addition, multiplication, and nonnegative scaling. Monomials are closed under multiplication and division. If a posynomial is multiplied by a monomial, the result is a posynomial; similarly, a posynomial can be divided by a monomial, with the result a posynomial [31, p. 161].

By introducing dummy variables and additional constraints, more general cost and constraint functions, called generalized posynomials, can be converted into posynomials and used to formulate GPs [71]. A *generalized posynomial* is a function formed from posynomials using addition, multiplication, the maximum operation, and positive exponentiation [71]. Generalized posynomials are closed under addition, multiplication, division by monomials, and the maximum operation [31, 71]. A standard inequality-constrained generalized GGP is of

the form

$$\begin{aligned} & \underset{x_1, \dots, x_n}{\text{minimize}} && C(x_1, \dots, x_n), \\ & \text{subject to} && f_i(x_1, \dots, x_n) \leq 1, \quad \forall i \in \{1, \dots, n_f\}, \\ & && g_i(x_1, \dots, x_n) = 1, \quad \forall i \in \{1, \dots, n_g\}, \end{aligned}$$

where the cost $C(x_1, \dots, x_n)$ and all the inequality constraints $f_i(x_1, \dots, x_n)$ are generalized posynomials, and all the equality constraints $g_i(x_1, \dots, x_n)$ are monomials [31]. By a logarithmic change of variables and objective function transformation, GGP can be cast in convex form and efficiently solved using numerical techniques [71]. Note that there is an implicit constraint that $x_i > 0 \forall i \in \{1, \dots, n\}$ [71].

We now turn to the problem of posing the optimization (3.37) as a GGP. Boyd and Vandenberghe [31] state that applying a monotonically increasing function to the objective results in an equivalent problem. Therefore, by applying the exponential function to the cost function (3.37), we obtain the equivalent problem,

$$\begin{aligned} & \underset{\mathbf{D}}{\text{minimize}} && \prod_{t=0}^{T-1} \prod_{i=1}^m \max \left\{ \frac{v_i(\mathbf{D}, t)}{D_i(t)}, k \right\}, \\ & \text{subject to} && \text{MSE}(\mathbf{D}, T) \leq \text{MSE}^*, \\ & && D_i(t) > 0, \quad \forall i, t. \end{aligned} \tag{3.38}$$

3.4.3 Proof that the optimization is a GGP

Until now, the analysis has resembled the prior art. The state evolution equations (3.23)–(3.26) were derived for a different update equation than ours by Yildiz and Scaglione [21]. Huang and Hua [28] extended analysis of consensus to vectors, although they also used a different state update. However, neither of these works optimize with respect to exact MSE values (3.28), and Huang and Hua did not account for the possibilities of different coding rates at each node and iteration or entropy coding.

The main original contribution of this work is to pose the coding rate optimization problem

with node- and iteration-dependent rates as a GGP, which can solve quickly and reliably for the global optimum coding rate vector \mathbf{R}_{opt} . The following theorem proves that the optimization (3.38) can be cast as such a problem.

Theorem *The above optimization problem (3.38) is a GGP.*

Proof: To show that (3.38) is a GGP, our argument takes the following steps: (i) show that $v_i(\mathbf{D}, t)$ is a posynomial, (ii) show that $\frac{v_i(\mathbf{D}, t)}{D_i(t)}$ is a posynomial, (iii) show that $\max\{\frac{v_i(\mathbf{D}, t)}{D_i(t)}, k\}$ is a generalized posynomial, (iv) show that $\prod_{t=0}^{T-1} \prod_{i=1}^m \max\{\frac{v_i(\mathbf{D}, t)}{D_i(t)}, k\}$ is a generalized posynomial, and (v) show that $\frac{1}{\text{MSE}^*} \text{MSE}(\mathbf{D}, t)$ is a posynomial.

(i) To show that the variance $v_i(\mathbf{D}, t)$ is a posynomial, it is necessary to express $v_i(\mathbf{D}, t)$ in terms of the individual distortions $D_i(t)$. From the state-evolution equation (3.30),

$$v_i(\mathbf{D}, t) = \left[\mathbf{W}^t \boldsymbol{\Sigma}_{\mathbf{z}}(0) \mathbf{W}^t + \sum_{s=0}^{t-1} \mathbf{W}^{t-s-1} (\mathbf{W} - \mathbf{I}) \boldsymbol{\Sigma}_{\epsilon}(s) (\mathbf{W} - \mathbf{I}) \mathbf{W}^{t-s-1} \right]_{ii}.$$

Recall that $\boldsymbol{\Sigma}_{\epsilon}(t)$ is a diagonal matrix by Assumption 1 and (2.9),

$$[\boldsymbol{\Sigma}_{\epsilon}(t)]_{ij} = \begin{cases} D_i(t), & i = j \\ 0, & \text{otherwise} \end{cases}.$$

Let $\widehat{\mathbf{W}}(t, s) = \mathbf{W}^{t-s-1} (\mathbf{W} - \mathbf{I})$, and denote the i^{th} column of $\widehat{\mathbf{W}}(t, s)$ as $\widehat{\mathbf{w}}_i(t, s) \in \mathbb{R}^m$. Each term $\widehat{\mathbf{W}}(t, s) \boldsymbol{\Sigma}_{\epsilon}(s) \widehat{\mathbf{W}}^{\top}(t, s)$ in the sum is then equal to

$$\widehat{\mathbf{W}}(t, s) \boldsymbol{\Sigma}_{\epsilon}(s) \widehat{\mathbf{W}}^{\top}(t, s) = \sum_{k=1}^m D_k(s) \widehat{\mathbf{w}}_k(t, s) \widehat{\mathbf{w}}_k^{\top}(t, s).$$

Therefore,

$$v_i(\mathbf{D}, t) = [\mathbf{W}^t \boldsymbol{\Sigma}_{\mathbf{z}}(0) \mathbf{W}^t]_{ii} + \sum_{s=0}^{t-1} \sum_{k=1}^m D_k(s) \left[\widehat{\mathbf{w}}_k(t, s) \widehat{\mathbf{w}}_k^{\top}(t, s) \right]_{ii}.$$

Because the diagonals of the outer product of a real vector with itself are always positive, each

coefficient in the above summation is positive and $v_i(\mathbf{D}, t)$ is a posynomial.

(ii) $D_i(t)$ is a monomial, and a posynomial divided by a monomial is a posynomial. Therefore, $\frac{v_i(\mathbf{D}, t)}{D_i(t)}$ is a posynomial.

(iii) The maximum of a posynomial and a monomial is a generalized posynomial. Because k is positive, it is trivially a monomial. Therefore, $\max\{\frac{v_i(\mathbf{D}, t)}{D_i(t)}, k\}$ is a generalized posynomial.

(iv) Generalized posynomials are closed under multiplication. Therefore, $\prod_{i,t} \max\{\frac{v_i(\mathbf{D}, t)}{D_i(t)}, k\}$ is a generalized posynomial.

(v) Define $\tilde{\mathbf{W}}(t, s) = (\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top)\mathbf{W}^{t-s-1}(\mathbf{W} - \mathbf{I})$ and let $\tilde{\mathbf{w}}_i(t, s) \in \mathbb{R}^m$ denote the i^{th} column of $\tilde{\mathbf{W}}(t, s)$. The MSE at each node (3.28) is given by

$$\text{MSE}_i(\mathbf{D}, t) = \left[\left(\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \right) \mathbf{W}^t \boldsymbol{\Sigma}_e(0) \mathbf{W}^t \left(\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \right) + \sum_{s=0}^{t-1} \tilde{\mathbf{W}}(t, s) \boldsymbol{\Sigma}_e(s) \tilde{\mathbf{W}}^\top(t, s) + \boldsymbol{\mu}_e(t) \boldsymbol{\mu}_e^\top(t) \right]_{ii}.$$

As in the proof of (i), the above equation reduces to

$$\text{MSE}_i(\mathbf{D}, t) = \left[\left(\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \right) \mathbf{W}^t \boldsymbol{\Sigma}_e(0) \mathbf{W}^t \left(\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \right) + \sum_{s=0}^{t-1} \sum_{k=1}^m D_k(s) \tilde{\mathbf{w}}_k(t, s) \tilde{\mathbf{w}}_k^\top(t, s) + \boldsymbol{\mu}_e(t) \boldsymbol{\mu}_e^\top(t) \right]_{ii}.$$

By the same argument as (i), $\text{MSE}_i(\mathbf{D}, t)$ is a posynomial. Likewise, the network MSE, $\text{MSE}(\mathbf{D}, t) = \frac{1}{m} \sum_{i=1}^m \text{MSE}_i(\mathbf{D}, t)$ is a posynomial.

Because the objective and constraint functions are posynomials, and the variables $D_i(t)$ are constrained to be positive, the optimization (3.38) is a GGP. \square

Two aspects of the above proof should be highlighted. Because the constraints are allowed to be generalized posynomials, one could also optimize with respect to a constraint on the

maximum node MSE, for example

$$\max_i \{\text{MSE}_i(\mathbf{D}, T)\}_{i=1}^m \leq \text{MSE}^*,$$

or constraints on each of the node MSE values,

$$\text{MSE}_i(\mathbf{D}, T) \leq \text{MSE}_i^*, \quad \forall i \in \{1, \dots, m\},$$

where $\text{MSE}_i(\mathbf{D}, t)$ is defined in (3.31). Also, in its most general form, the optimization allows each node to use a different rate or distortion. In the interest of designing a distributed protocol (or for computational efficiency), one may wish to constrain the rates or distortions at each node to be the same. The constraint that all distortions be the same is a straightforward modification of (3.38) and is also provably a GGP.

3.5 Efficient implementation of GGP optimization

Solving the exact optimization problem (3.38) naively requires explicit representation of all mT distortions $D_i(t)$ and all the coefficients of the log-sum-exp (LSE) model⁵ required to compute $\text{MSE}_i(\mathbf{D}, t)$ and $v_i(\mathbf{D}, t)$ from \mathbf{D} . The result of this explicit representation is large time and memory complexity. In this section, we explore a simplification of the optimization problem to combat these issues, and we present a heuristic to apply the GGP optimization method to fixed-rate coding schemes.

3.5.1 Equal-distortion simplification and implementation

According to Boyd and coauthors [31, 71], GGPs can be converted into convex optimization problems by a change of variables and a logarithmic transformation of the objective function. For a detailed treatment in the case of GP, the reader is referred to the above references. GGPs

⁵GGPs are converted to convex LSE form for solution [73, 74].

can be converted in a similar fashion; however, the conversion is somewhat more complicated. In some software packages, such as CVX [73, 74], GGP's can be specified directly, and they are automatically converted into corresponding convex programs [71]. This approach guarantees that the global optimum can be found with little additional effort by the programmer.

In practice, for large networks, (i.e., $m > 20$ and $T \geq 6$) the memory and time requirements of the optimization (3.38) seem to grow very quickly. Furthermore, using CVX [73, 74], all compatible solvers tested displayed poor stability for large networks. Attempts to use alternative modeling frameworks, such as GPkit [75], CVXPY [76], and YALMIP [77], failed due to the high memory requirements of the model. Manually building a GGP model is memory- and time-intensive; if explicit model representation can be avoided, however, it may be possible to apply other convex optimization methods without these scaling issues. In particular, avoiding the explicit representation of posynomial coefficients and exponents would probably greatly improve performance for large problem sizes m and T . To provide a program that is more easily solved in practice, we make two simplifications. First, we constrain the distortions to be equal at each node, which is well motivated for the end goal of designing a truly distributed protocol. Next, the program can be cast as

$$\begin{aligned}
 & \underset{\mathbf{D}}{\text{minimize}} && \prod_{t=0}^{T-1} \prod_{i=1}^m \max \left\{ \frac{v_i(t)}{D(t)}, k \right\}, \\
 & \text{subject to} && \text{MSE}(\mathbf{D}, T) \leq \text{MSE}^*, \\
 & && D(t) > 0, \quad \forall t.
 \end{aligned} \tag{3.39}$$

In the following chapter, the results of solving this simplified problem (3.39) are compared to the solutions of the exact program (3.38) and the prior art [20, 21]. Surprisingly, the above approximation provides competitive results for random geometric networks [45], with significant reduction in memory and run-time requirements.

3.5.2 Efficient search heuristic for fixed-rate quantizers

Using fixed-length coding with uniform scalar quantization, the rates to be used are integer-valued [64], and arbitrary distortions are not achievable for a given source variance. To account for this constraint, we suggest a search heuristic. Given a particular T and MSE^* , there exist a finite number of rate vectors $\mathbf{R} \in \mathbb{Z}_{>0}^{mT}$ (3.1). Clearly, if each node is allowed to use different rates, then the size of the search space becomes too large to be practical, and we constrain our attention to the case that rates are constant over nodes for a particular iteration: $R_i(t) = R_j(t) = R(t), \forall i, j \in \{1, \dots, m\}, t \in \{0, \dots, T-1\}$. Even in the node-constant rate case, the number of possible sequences such that $\min_{i,t} R_i(t) = 1$ and $\max_{i,t} R_i(t) = R_{\max}$ grows combinatorially in T .

To find an efficient (though perhaps suboptimal) sequence, we propose to use the output of the GGP (3.38) as a starting point for the optimization. Because constraining the rates to be equal at each node is not straightforward, we instead constrain the distortions to be equal at each node as in (3.39). The optimal rates $R_i(t)$ from the solution of (3.39) are then averaged into a single sequence $\{R_{\text{GGP}}(t)\}_{t=0}^{T-1}$,

$$R_{\text{GGP}}(t) := \frac{1}{m} \sum_{i=1}^m R_i(t).$$

A neighborhood is formed as follows. To the original sequence

$$\mathbf{R}_{\text{GGP}} := [R_{\text{GGP}}(0), \dots, R_{\text{GGP}}(T-1)],$$

we add $\ell \in \{-1, +1\}^T$. Each possible ℓ is added, and the result is stored, until all 2^T possible transformations of \mathbf{R}_{GGP} have been computed. Because the minimum fixed rate is one bit per symbol, all rates less than one bit per symbol from the previous step are re-assigned to one bit per symbol. Next, we apply the operators $A \in \{[\cdot], [\cdot]^\top\}^T$ to the resulting sequences, where, for example, $[[\cdot], [\cdot], [\cdot]] [x, y, z]^\top := [[x], [y], [z]]^\top$. Finally, we eliminate duplicate sequences.

This results in a “trellis” containing the integers about each $R_{\text{GGP}}(t)$. The maximum possible total number of sequences in the search space is thus 4^T . Brute-force search is conducted on this subset of sequences to find the one that satisfies the MSE constraint with the minimum aggregate rate $R_{\text{agg}} = m \sum_{t=0}^{T-1} R(t)$. In the case that multiple sequences achieve the MSE constraint with equal aggregate rate, the one that produces the lowest MSE is selected.

In this chapter, we present numerical results that provide further insight into the optimal rate sequences and their relationship to the target mean square error (MSE) and the network topology. We further compare the performance of the proposed generalized geometric program (GGP) and heuristic optimizations to the prior art. To test the effectiveness of the proposed approach, we used MATLAB [78] for simulation. We used the CVX toolbox [73, 74] to solve the GGP presented in Chapter 3, which converts the GGP model to a convex problem automatically.

In the results presented, the networks were generated by random geometric graph models [45], and each node state $\zeta_i(0)$ was initialized with the same independent and identically distributed (i.i.d.) variance- σ_x^2 zero-mean Gaussian vector $\mathbf{x} \in \mathbb{R}^L$ corrupted by a different variance- σ_n^2 , zero-mean Gaussian noise $\mathbf{n}_i \in \mathbb{R}^L, i \in \{1, \dots, m\}$. More precisely,

$$\zeta_i(0) = \mathbf{x} + \mathbf{n}_i, \quad \forall i \in \{1, \dots, m\}, \quad (4.1)$$

where $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma_x^2 \mathbf{I})$ and $\mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}), \forall i \in \{1, \dots, m\}$. The random geometric graphs were generated on the unit torus (i.e., edge effects were neglected by “wrapping” the top and bottom and left and right edges of $[0, 1]^2$) [45, 79, 80].

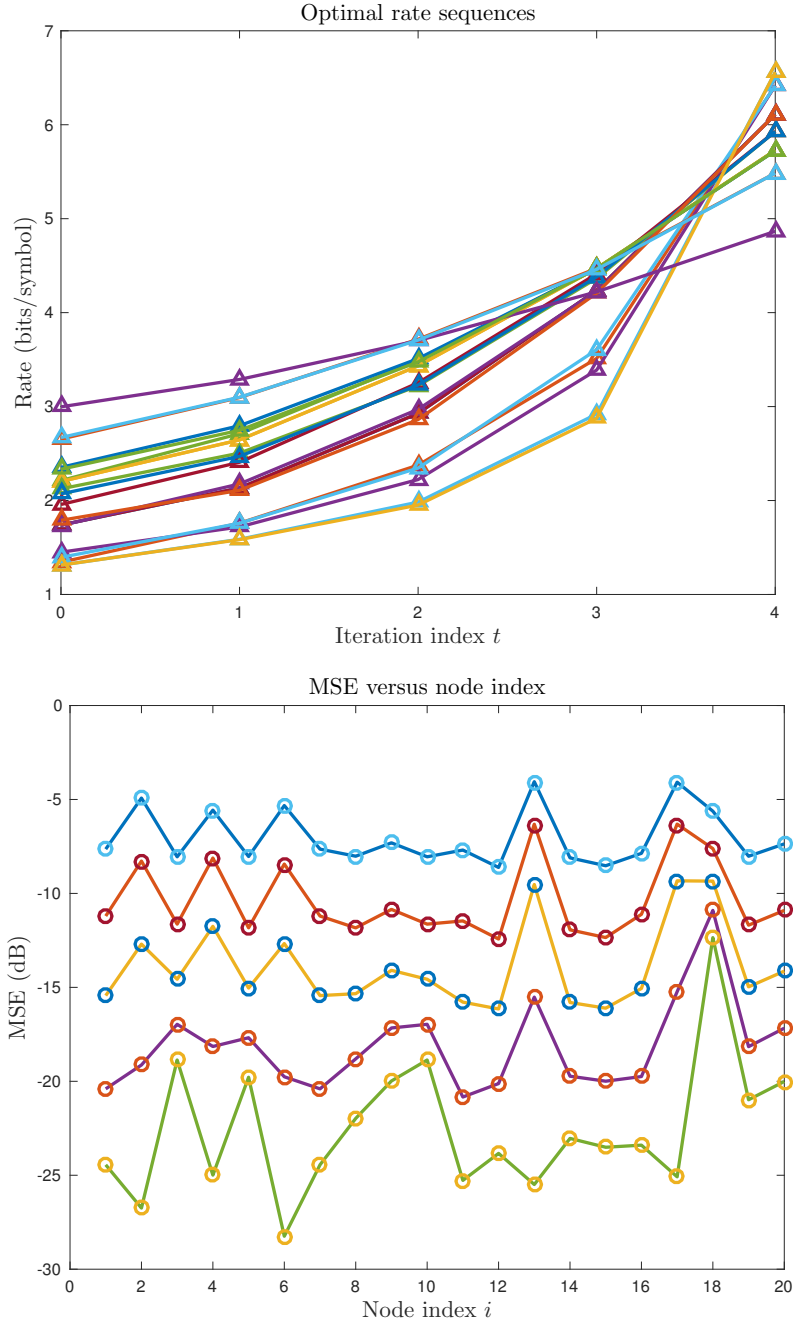


Figure 4.1 Optimal rates and MSE sequences from the solution of (3.38) ($T = 5$, $\rho_c = 0.35$, $\sigma_x^2 = 1$, $\sigma_n^2 = 0.5$, $m = 20$). *Top*: Optimal rate sequences for the variable-distortion optimization problem. The rates are plotted against iteration indices, and each line represents the rates used by a different sensor. *Bottom*: MSE sequence corresponding to the above rate sequence. The MSE values are plotted against node indices, and each line represents a different iteration. In this case, the MSE values decrease monotonically for all nodes. The lines represent values predicted by the state evolution equations (3.23)–(3.26), and the overlaid circles represent the simulated values.

In the case of very low rates using entropy-coded uniform scalar quantization (ECSQ), nothing would be communicated (all elements transmitted/received will be zeros). To prevent this behavior, the maximum normalized distortion D_{\max} allowed was set such that 1% of the elements were nonzero.

4.1 Structure of solutions

To verify the state-evolution equations (3.23)–(3.26) and better understand the structure of the solutions to the optimization problems (3.38) and (3.39), we present some anecdotal simulation results. Each of these results is taken from a different single instantiation of the optimization problem.

The optimal rate sequences $\{R_i(t)\}_{t=0}^{T-1}$, $i \in \{1, \dots, m\}$ for both the variable-distortion (3.38) and constant-distortion (3.39) problems typically exhibit monotonically nondecreasing structure, with an increasing rate of change toward the final iterations. In the constant-distortion case, the rates $R_i(t) \approx \frac{1}{2} \log_2 \left(\frac{v_i(\mathbf{D}, t)}{D_i(t)} \right) + R_c$ are similar because the ratios $\frac{v_i(\mathbf{D}, t)}{D_i(t)}$ in the Gaussian operational rate-distortion (RD) relationship (3.33) are similar across the network. Examples of the optimal rate sequences and corresponding MSE state evolution are provided for both variants of the optimization problem (3.38) and (3.39) in Figures 4.1 and 4.2.

The reader may wonder why the rate sequences of Figures 4.1 and 4.2 look different. In Figure 4.1, the distortions $\{D_i(t)\}_{i=1, t=0}^{m, T-1}$ corresponding to the optimization of (3.38) are allowed to vary across both the node index $i \in \{1, \dots, m\}$ and the iteration index $t \in \{0, \dots, T-1\}$. This optimization is more flexible with respect to the choice of rates $\{R_i(t)\}_{i=1, t=0}^{m, T-1}$ corresponding to $\{D_i(t)\}_{i=1, t=0}^{m, T-1}$. In Figure 4.2, however, the distortion is constrained to be the same the network for a particular iteration (i.e., $\{D_i(t)\}_{i=1, t=0}^{m, T-1}$ varies only across $t \in \{1, \dots, T-1\}$). This corresponds to the solution of (3.39).

The pattern of these rate sequences is intuitive and mirrors the results of Zhu and coauthors' study of multiprocessor approximate message passing [47, 48, 51]. As the estimate of the sample mean at each node increases in precision, higher-resolution messages must be

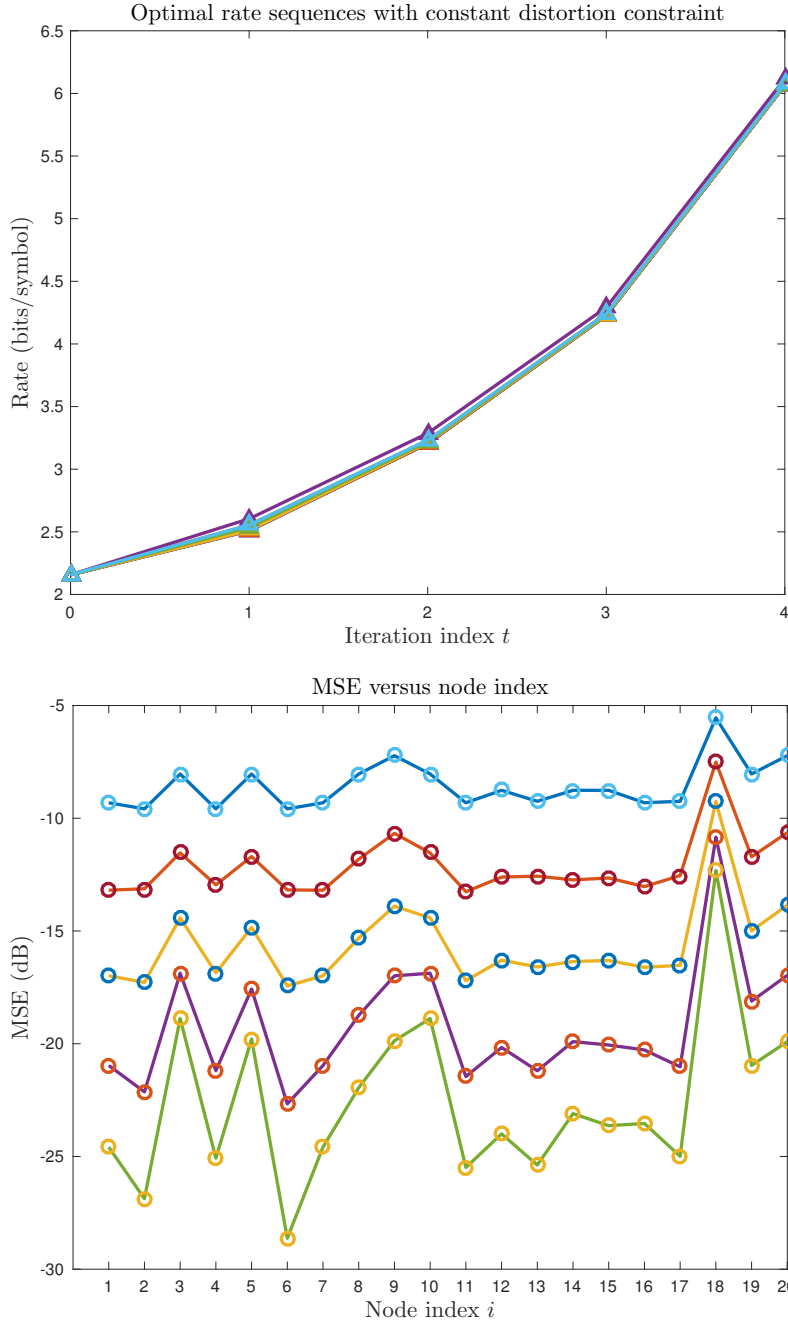


Figure 4.2 Optimal rates and MSE sequences from the solution of (3.39) ($T = 5$, $\rho_c = 0.35$, $\sigma_x^2 = 1$, $\sigma_n^2 = 0.5$, $m = 20$). *Top*: Optimal rate sequences for the constant-distortion optimization problem. The rates are plotted against iteration indices, and each line represents the rates used by a different sensor. *Bottom*: MSE sequence corresponding to the above rate sequence. The MSE values are plotted against node indices, and each line represents a different iteration. In this case, the MSE values decrease monotonically at all nodes. The lines represent values predicted by the state evolution equations (3.23)–(3.26), and the overlaid circles represent the simulated values.

exchanged among processors to achieve increasing estimate quality. In the case of coding without side information, this implies the use of larger coding rates in the later iterations of the algorithm.

The results of the variable-distortion problem (3.38) also exhibit structure based on the connectivity of each node. Figure 4.3 contains a scatter plot of the aggregate rate versus the node degree, which is a proxy for the “connectedness” of each node of the network. There is a negative correlation between the node degree and the aggregate rate used by each node. Figure 4.3 also contains a surface demonstrating the changing shape of the rate sequence as the node degree decreases. In general, the experimental results suggest that nodes with higher degree tend to have flatter optimal rate sequences, whereas the nodes with lower degree ramp up their coding rates more aggressively from beginning to end.

Intuitively, this pattern can be explained by considering the overall goal of consensus and the impact of quantization on the information content [44] of the network’s sample mean estimates. The more poorly connected nodes must use greater rates at each iteration to diffuse their information throughout the network, due to topological bottleneck effects [41]. The more highly connected nodes, however, can save coding rate by exploiting their topological advantage, as the information content of their messages will be diffused more quickly.

In all cases simulated, the differences in aggregate rate between the exact optimization problem (3.38) and the constant-distortion problem (3.39) were small, on the order of 5%. For the sequences plotted in Figures 4.1 and 4.2, the difference was 4.45%. This similarity may be an effect of the highly regular random geometric graph structure. For random geometric networks on squares rather than tori, the difference may be more significant due to the irregularity arising from edge effects.

4.1.1 Nonmonotone behavior

Intuitively, it seems reasonable to expect the MSE to decrease monotonically, and the rates to increase monotonically, since greater precision is required to encode the increasingly precise

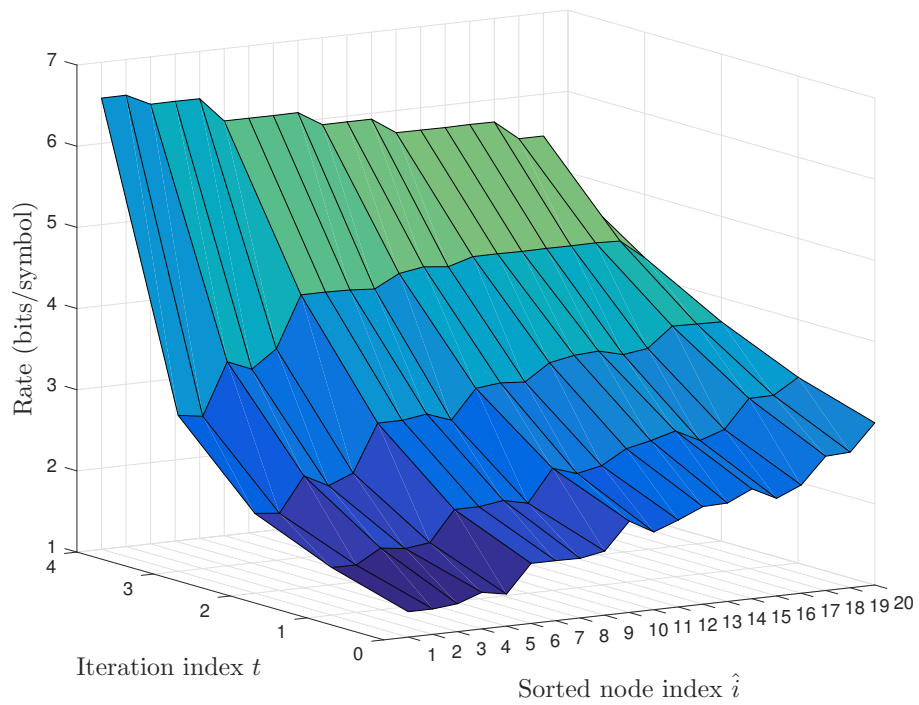
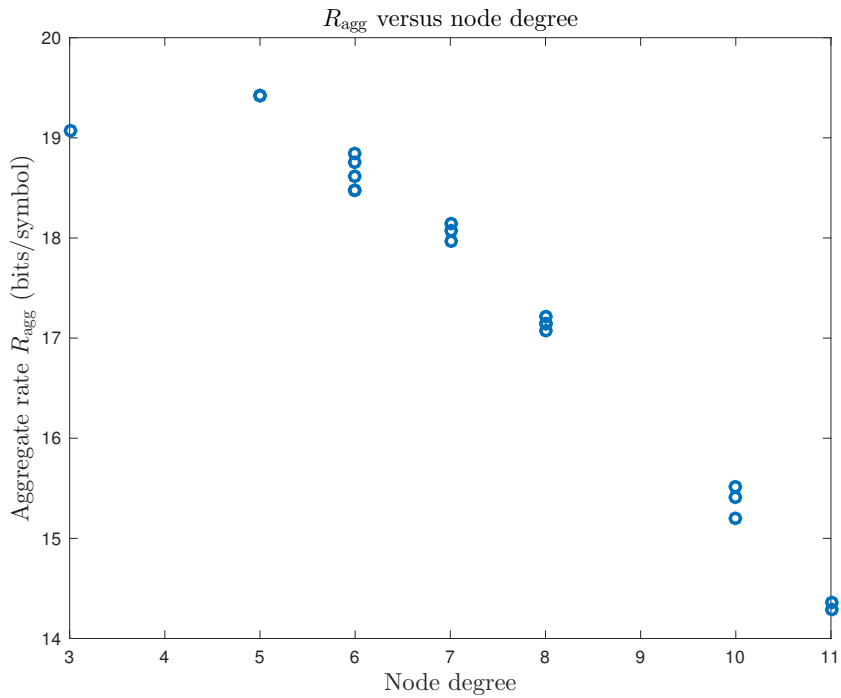


Figure 4.3 Influence of connectivity on the optimal rates from the solution of (3.39) ($T = 5$, $\rho_c = 0.35$, $\sigma_x^2 = 1$, $\sigma_n^2 = 0.5$, $m = 20$). *Top*: Scatter plot of aggregate rate used at each node versus node degree. *Bottom*: Surface of optimal rate sequences at each node, where the node indices are sorted by node degree in decreasing order.

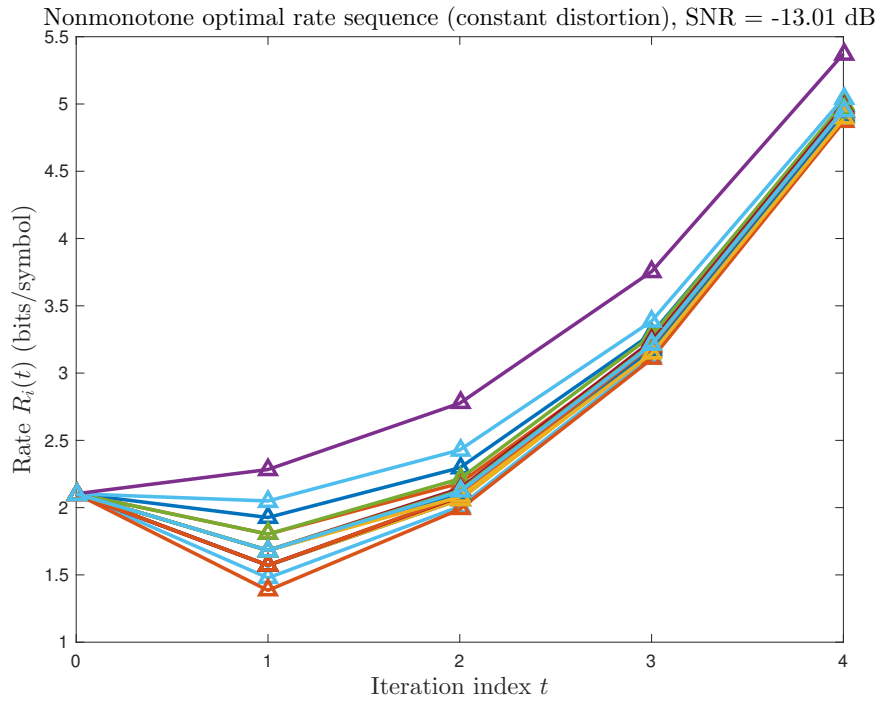
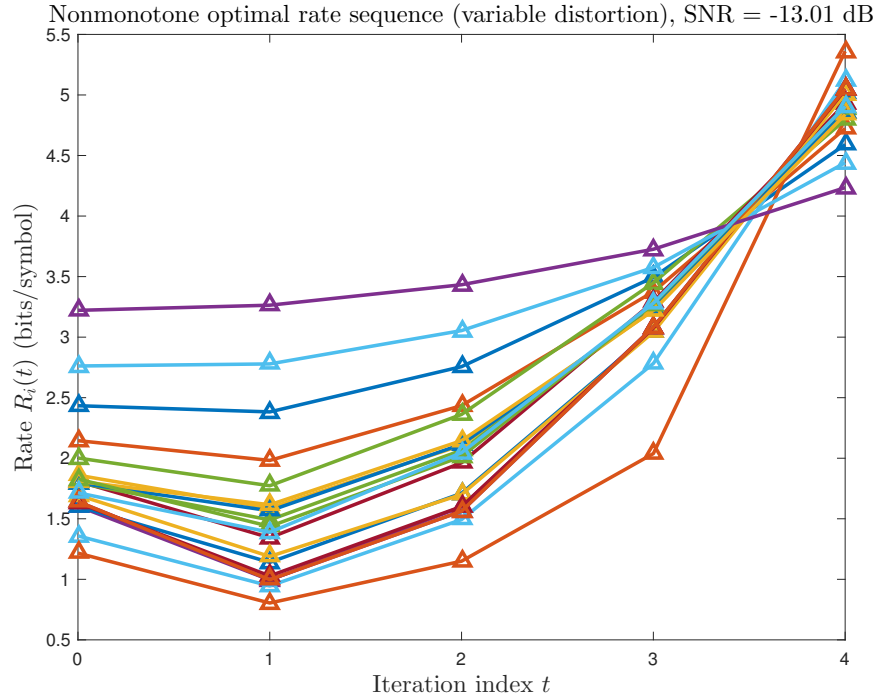


Figure 4.4 Nonmonotone optimal rate sequences for the low-SNR, low-MSE setting ($T = 5$, $\rho_c = 0.35$, $\sigma_x^2 = 1$, $\sigma_n^2 = 20$, $m = 20$). Each line corresponds to the rate sequence used by a different node. *Top*: Variable-distortion optimization (3.38) result. *Bottom*: Constant-distortion optimization (3.39) result.

estimates of the sample average available to each node. However, in certain edge cases, neither of these patterns holds. In this subsection, we discuss these behaviors and present some hypothesized reasons for these nonintuitive results.

Figure 4.4 demonstrates a case in which the optimal rate sequences are nonmonotone. In this case, the signal-to-noise ratio (SNR),

$$\text{SNR} := 10 \log_{10} \frac{\sigma_x^2}{\sigma_n^2},$$

is low (-13.01 dB), which introduces a trade-off for the rates $R_i(t)$ corresponding to the first few iterations of the consensus algorithm. Recall that the simulation initializes each node to have access to the same zero-mean Gaussian signal plus its own realization of additive white Gaussian noise (AWGN). In this case, the covariance matrix (3.25) has the form

$$\Sigma_{\mathbf{z}}(t=0) = \sigma_x^2 \mathbf{1}\mathbf{1}^\top + \sigma_n^2 \mathbf{I}. \quad (4.2)$$

If the signal variance is set to zero (corresponding to the case of averaging independent AWGN processes), then the diagonals of the covariance matrix (3.25) will converge to $\frac{\sigma_n^2}{m}$ as the iteration index $t \rightarrow \infty$, which is significantly less than the original variance σ_n^2 for large networks.

Therefore, for low SNR, the impact of quantization error on the variance becomes significant. The variance state-evolution equation is (3.25)

$$v_i(t) = \left[\mathbf{W}^t \Sigma_{\mathbf{z}}(0) \mathbf{W}^t + \sum_{s=0}^{t-1} \mathbf{W}^{t-s-1} (\mathbf{W} - \mathbf{I}) \Sigma_{\epsilon}(s) (\mathbf{W} - \mathbf{I}) \mathbf{W}^{t-s-1} \right]_{ii}.$$

It can be seen from the above equation that both the original covariance matrix and the distortions introduced in past iterations play a role in the variance at any iteration $t \in \mathbb{Z}_{>0}$.

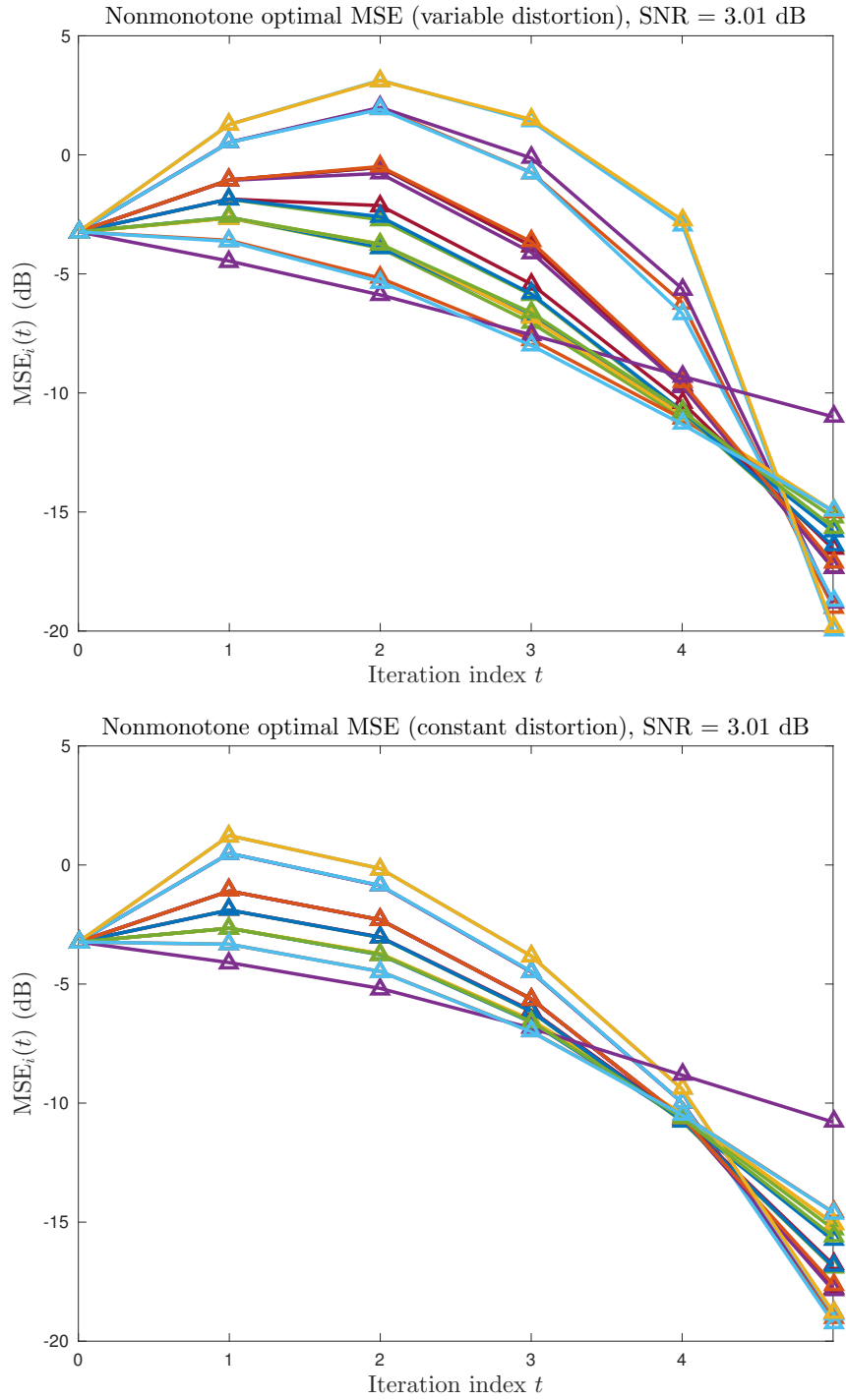


Figure 4.5 Nonmonotone MSE sequences. Each line corresponds to the MSE sequence of a different node ($T = 5, \rho_c = 0.35, \sigma_x^2 = 1, \sigma_n^2 = 0.5$). *Top*: MSE evolution for the variable-distortion problem (3.38). *Bottom*: MSE evolution for the constant-distortion problem (3.39).

Furthermore, recall the family of operational RD relationships we study (3.33),

$$R_i(\mathbf{D}, t) \approx \frac{1}{2} \log_2 \left(\max \left\{ \frac{v_i(\mathbf{D}, t)}{D_i(t)}, k \right\} \right) + R_c,$$

which indicates that, holding the distortion $D_i(t)$ constant, larger coding rates are required if the source has higher variance $v_i(t)$. From (3.30) and (4.2), we can see that introducing larger distortions will increase the source variance for all future iterations, and thus increase the communication cost. Therefore, using higher coding rates (introducing lower distortions) in the initial iterations may be preferable in settings where the source variance makes a significant contribution to the overall cost, as in the low-SNR setting explored here.

It seems intuitive to expect the MSE to decrease monotonically in iterations as $t \rightarrow \infty$, but this has been observed not to be true in general. If the target MSE^* is much higher than the $\text{MSE}(\mathbf{D}, T)$ (3.32) achievable in the lossless case (i.e., $R_i(t) \rightarrow \infty, \forall i \in \{1, \dots, m\}$, $t \in \{0, \dots, T-1\}$), then the MSE at each node might increase initially, followed by a rapid decrease in the later iterations. This behavior is illustrated in Figure 4.5. We hypothesize that this nonmonotonicity is an effect of the different treatment of states and quantization errors in the state update. Recall the iteration we use (2.8), borrowed from Frasca *et al.* [14]:

$$\zeta(t+1) = \zeta(t) + (\mathbf{\Omega} - \mathbf{I}) Q(\zeta(t)).$$

This equation can be rewritten as (2.10),

$$\zeta(t+1) = \mathbf{\Omega} \zeta(t) + (\mathbf{\Omega} - \mathbf{I}) \delta(t),$$

where the reader is reminded that $\zeta(t) \in \mathbb{R}^{mL}$ is the state supervector (3.7) that contains the vector-valued states $\zeta_i(t)$ at each node $i \in \{1, \dots, m\}$, and $\delta(t) \in \mathbb{R}^{mL}$ is the quantization error (3.10) corresponding to each element of $\zeta(t)$. Because the state $\zeta(t)$ and quantization error $\delta(t)$ are incorporated into the state update differently, they have different impacts on the

MSE. In our case, this difference appears to be favorable in the sense that it allows the impact of distortions to decay quickly.

Mathematically, the reason for the nonmonotone optimal MSE behavior may be discerned from the state evolution equations. The reader is reminded of the MSE state evolution relationship (3.31),

$$\begin{aligned} \text{MSE}_i(t) = & \left[\left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \mathbf{W}^t \boldsymbol{\Sigma}_e(0) \mathbf{W}^t \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \right. \\ & \left. + \sum_{s=0}^{t-1} \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \mathbf{W}^{t-s-1} (\mathbf{W} - \mathbf{I}) \boldsymbol{\Sigma}_e(s) (\mathbf{W} - \mathbf{I}) \mathbf{W}^{t-s-1} \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \right]_{ii}, \end{aligned} \quad (4.3)$$

where the $\boldsymbol{\mu}_e(t) \boldsymbol{\mu}_e^\top(t)$ term is omitted because the simulated signal and noise are zero-mean.

To simplify the explanation of the nonmonotone MSE behavior, we will consider the constant-distortion optimization, for which the above (4.3) reduces to

$$\begin{aligned} \text{MSE}_i(t) = & \left[\left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \mathbf{W}^t \boldsymbol{\Sigma}_e(0) \mathbf{W}^t \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \right. \\ & \left. + \sum_{s=0}^{t-1} D(s) \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \mathbf{W}^{t-s-1} (\mathbf{W} - \mathbf{I})^2 \mathbf{W}^{t-s-1} \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \right]_{ii}. \end{aligned}$$

In the Gaussian-signal-plus-noise setting (4.1) mentioned at the beginning of this chapter, the initial error covariance matrix (3.26) is $\boldsymbol{\Sigma}_e(0) = \sigma_n^2 (\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top)$. Because $\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top$ is idempotent,¹ the first term in the above equation reduces to $\sigma_n^2 (\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top) \mathbf{W}^{2t} (\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top)$, and we can assess the different contributions of noise variance σ_n^2 and distortion $D(t)$ to the network MSE, $\text{MSE}(\mathbf{D}, t)$, by examining the traces of the matrices

$$\left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right) \mathbf{W}^{2t} \left(\mathbf{I} - \frac{1}{m} \mathbf{1}\mathbf{1}^\top \right)$$

¹An idempotent matrix \mathbf{A} is one for which $\mathbf{A}^2 = \mathbf{A}$ [81].

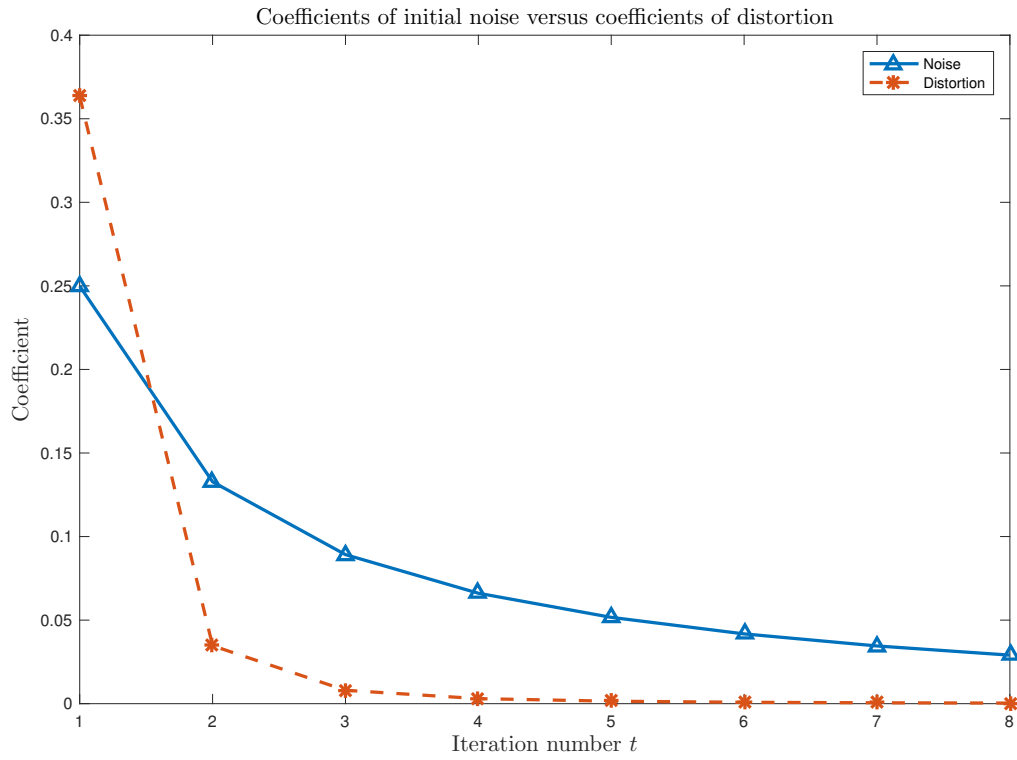


Figure 4.6 Coefficients of the noise variance σ_n^2 (solid blue curve) and distortion $D(0)$ (dashed red curve) in the equation for the MSE (3.31) when the states are initialized as a Gaussian random vector corrupted by AWGN ($\rho_c = 0.35$, $m = 20$).

and

$$\left(\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top\right)\mathbf{W}^t(\mathbf{W} - \mathbf{I})^2\mathbf{W}^t\left(\mathbf{I} - \frac{1}{m}\mathbf{1}\mathbf{1}^\top\right),$$

for $t \in \{0, \dots, T-1\}$. The reader is reminded that the trace of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, written $\text{tr}(\mathbf{A})$, is the sum of its diagonal elements (i.e., $\text{tr}(\mathbf{A}) := \sum_{i=1}^n a_{ii}$).

These diagonal components are the coefficients of the noise variance σ_n^2 and the distortions $D_i(t)$ in the equation for the MSE. Figure 4.6 plots the diagonal values for the above matrices, $\forall t \in \{0, \dots, T-1\}$. The coefficients of $D(t)$ decrease sharply after a single iteration, meaning that the distortion contribution to the MSE vanishes quickly compared to the contribution of the initial state variance.

4.2 Comparison to prior art

To compare our work to the prior art [20, 21], we generated 32 random geometric networks [45] with connectivity radius² $\rho_c \in \{0.35, 0.45\}$ on a two-dimensional unit torus. For each of these networks, consensus was run on 1000 realizations of the initial states, which were length-10000 i.i.d. Gaussian vectors $\mathbf{z}_i(0) = \mathbf{x} + \mathbf{n}_i$, $\forall i \in \{1, \dots, m\}$, $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{n}_i \sim \mathcal{N}(\mathbf{0}, 0.5\mathbf{I})$ (SNR = 2, which is 3.01 dB).

We simulated ProgQ [20] and order-one predictive coding [21], using initial rates $R_i(0) \in \{4, \dots, 7\} \forall i$ and $R_i(0) \in \{3, \dots, 6\} \forall i$, respectively. The measured final MSE (3.31) for these schemes were set as the target values for the proposed GGP and heuristic optimizations.

For all schemes, $\text{MSE}(\mathbf{D}, T)$ and R_{agg} were computed. These values were averaged over all 32 realizations of each $(\rho_c, R_i(0), T)$ setting, and the resulting averages were plotted against each other. Assuming an additive noise model for the quantization error means that the MSE for quantized communication will always be higher than the MSE of the unquantized algorithm. We therefore introduce two terms to define the MSE performance relative to the

²A random geometric network is one for which each node $V_i \in \mathcal{V}$ is associated with a geographic coordinate \mathbf{v}_i . Nodes V_i, V_j are connected if $\|\mathbf{v}_i - \mathbf{v}_j\|_2 \leq \rho_c$ [46].

ideal, unquantized algorithm. To compensate for the effect of network topology on the MSE, we define the *lossless MSE*,

$$\text{MSE}_{\text{lossless}}(t) := \text{MSE}(\mathbf{D}, t) \Big|_{\mathbf{D}=\mathbf{0}}.$$

We then define the *excess MSE* (EMSE) as

$$\text{EMSE} := 10 \log_{10} \frac{\text{MSE}(\mathbf{D}, T)}{\text{MSE}_{\text{lossless}}(T)},$$

which represents the increase in MSE over lossless consensus resulting from distortion. The EMSE was used for the generation of RD trade-off curves.

4.2.1 Implementation details

To generate the weight matrix \mathbf{W} , the max-degree heuristic [37, 82] was used, so that

$$\mathbf{W} = \mathbf{I} - \alpha \mathbf{L} = \mathbf{I} - \frac{0.9}{\max_i \{\text{deg } i\}_{i=1}^m} \mathbf{L}, \quad (4.4)$$

where $\text{deg } i$ is the degree of node i and \mathbf{L} is the graph Laplacian [54, Sec. 10.9]. This weighting scheme was selected for simplicity, following the implementation of Yildiz and Scaglione [21]. The factor of 0.9 is used to ensure numerical stability, since the coefficient α of \mathbf{L} in (4.4) must satisfy $\alpha \in (0, 1/\max_i \{\text{deg } i\}_{i=1}^m)$ to guarantee convergence [37]. Based on (4.4), \mathbf{W} is symmetric and satisfies all the requirements of the matrix properties discussed in Chapter 2.

The implementation of Yildiz and Scaglione’s order-one predictive coding [21], which was graciously provided by the authors, was modified to use fixed-rate uniform quantization but allow for the rate to vary with iteration and node indices. This capability was implemented by running two rate update recursions—one to keep track of the ideal (real-valued) rates given by the quantization noise variance recursion [21], and another to perform the predictive coding using rates that were rounded to the nearest integral value. Interested readers can access the

code, which will be posted online, for further details.

The optimization modeler for the GGP and its associated solvers suffered from poor stability at high iteration count, and poor scaling in the network size and number of iterations for the exact program (3.38). Therefore, only the fixed-distortion problem (3.39) was solved for comparison to the prior art.

The bin size of all fixed-rate uniform quantizers was set to 12 times the standard deviation of the Gaussian data to prevent clipping.

4.2.2 Discussion of comparison results

To adequately discuss the RD results, it is first necessary to comment on some properties of each of the schemes presented. The ProgQ algorithm of Thanou *et al.* [20] uses a time- and node-invariant fixed-rate uniform quantizer (i.e., $R_i(t) = R, \forall i \in \{1, \dots, m\}, t \in \{0, \dots, T-1\}$), whereas Yildiz and Scaglione [21] allow the use of different rates at each node and iteration.

Thanou *et al.* [20] use the same state update as ours (2.8), but Yildiz and Scaglione [21] use a different update that is incapable of truly converging in the presence of quantization error. The final asymptotic MSE for the predictive scheme depends on the sum of distortions $D_i(t), t \in \mathbb{Z}_{\geq 0}$. If these distortions are chosen to form a convergent series, then the MSE will converge to a nonzero, but bounded, value. Because of this limitation, the predictive scheme [21] is heavily dependent on the starting rates $R_i(0)$.

For each fixed-rate uniform quantizer sequence in the prior art comparisons, the result of the heuristic from Sec. 3.4 was compared to the optimum obtained by an exhaustive search over all possible rate sequences with a maximum aggregate rate value per node of $\sum_{t=0}^{T-1} R_{GGP}(t)$. In every case, the heuristic was able to find the optimal result. These results seem to justify our heuristic search. Despite the reduced search complexity (exponential instead of combinatorial) of this heuristic, the simulated performance was no worse. Therefore, at the least, this heuristic seems to be a promising approach whenever fixed-rate coding is used. The RD performances of the proposed optimization schemes are compared to the predictive coding scheme of Yildiz

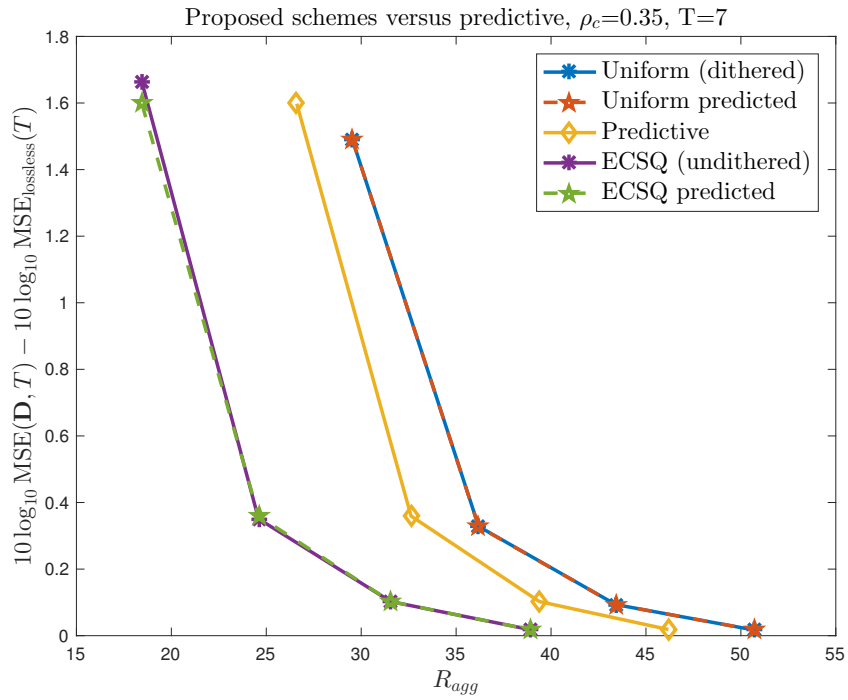
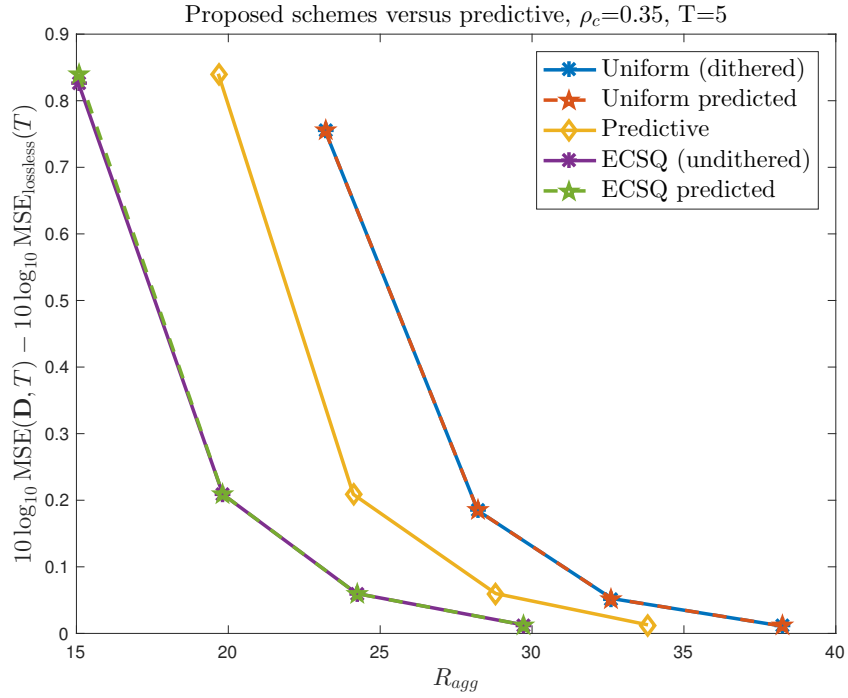


Figure 4.7 RD trade-off curves for the proposed schemes versus order-one predictive coding [21]. All schemes were simulated for random geometric graphs on a unit torus with connectivity radius $\rho_c = 0.35$. *Top*: number of iterations $T = 5$. *Bottom*: number of iterations $T = 7$.

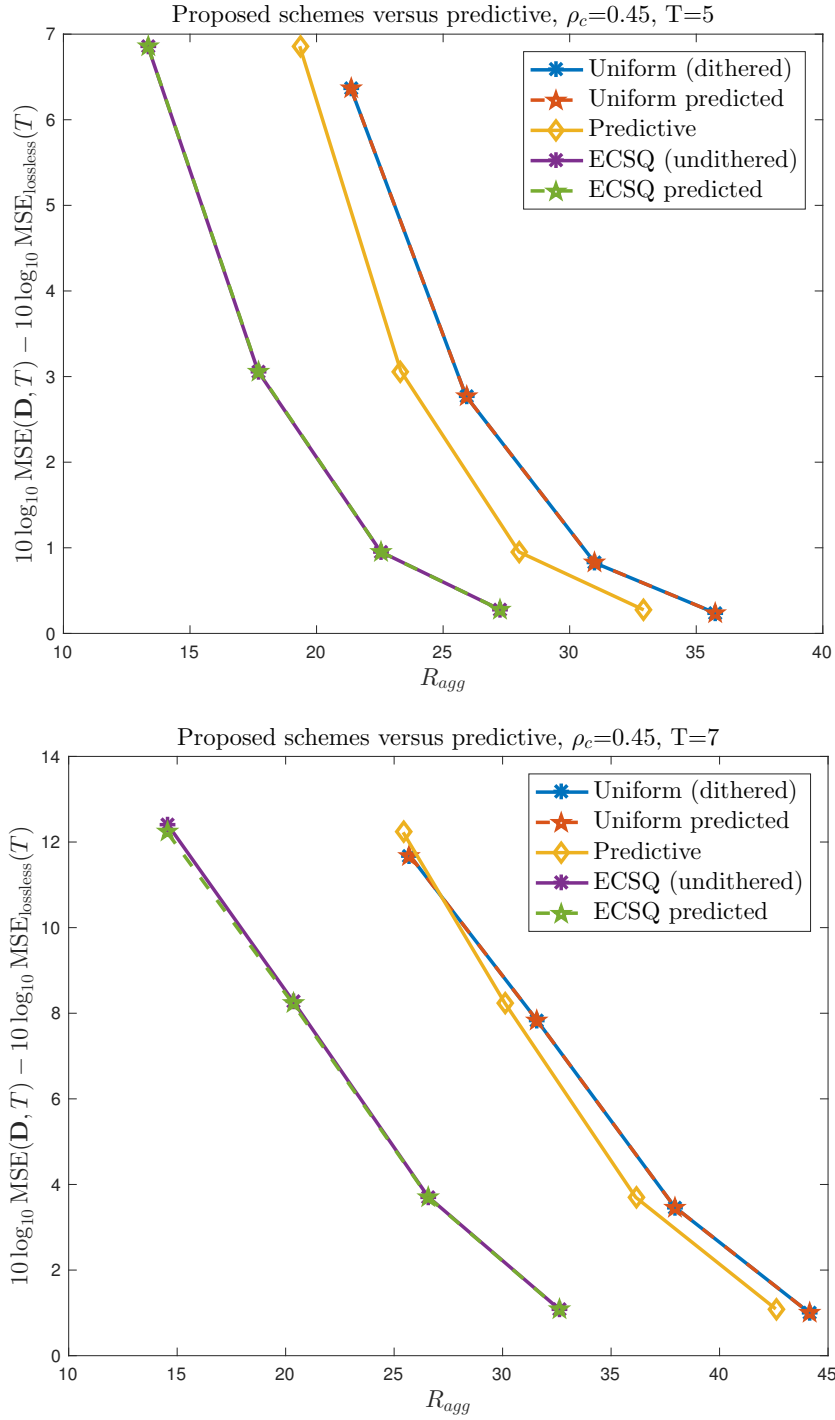


Figure 4.8 RD trade-off curves for the proposed schemes versus order-one predictive coding [21]. All schemes were simulated for random geometric graphs on a unit torus with connectivity radius $\rho_c = 0.45$. *Top*: number of iterations $T = 5$. *Bottom*: number of iterations $T = 7$.

and Scaglione [21] in Figures 4.7 and 4.8 and to the ProgQ algorithm of Thanou *et al.* [20] in Figures 4.9 and 4.10.

These RD trade-off curves (Figures 4.7–4.10) contain the performance of both ECSQ and fixed-rate uniform quantization. For both of our schemes, the predicted RD performance (as computed by the state evolution equations (3.23)–(3.26)) is compared to the actual performance. The distortion (measured by the EMSE) is given by the y -axis, and the aggregate rate R_{agg} by the x -axis. The predicted performance is denoted by a dashed line, while the simulated performance is represented as a solid line. The legend disambiguates the many curves that are plotted. Alongside our approaches, we plot the RD performance of one of the comparators. A curve closer to the bottom left corner of these figures indicates better performance, meaning lower aggregate rate R_{agg} to achieve the same EMSE, or alternatively lower EMSE for a particular R_{agg} investment.

In some cases, the predicted performance and measured performance of ECSQ do not match. Because the ECSQ used in the simulations is not dithered, the additive quantization model only holds approximately. As the aggregate rate increases, the performance improves, and better adherence to predicted performance can be accomplished using dithering.

In addition to the ECSQ and fixed-rate uniform quantizers, Lloyd-Max quantization [60, 61] was studied. However, its empirical MSE performance did not match predictions, because its quantization error $\delta(t)$ is correlated with the source $\zeta(t)$. Due to this correlation, the use of Lloyd-Max quantization in our framework seems inappropriate.

4.2.3 Comparison to predictive coding scheme

If we neglect the effect of entropy coding on the RD curves in Figures 4.7 and 4.8, then we can more easily assess the impact of predictive coding on the aggregate rate. In each of these cases, the predictive coding scheme requires lower aggregate rate than the proposed fixed-rate uniform quantization strategy. However, when ECSQ is used in place of fixed-rate uniform quantization, the results improve significantly.

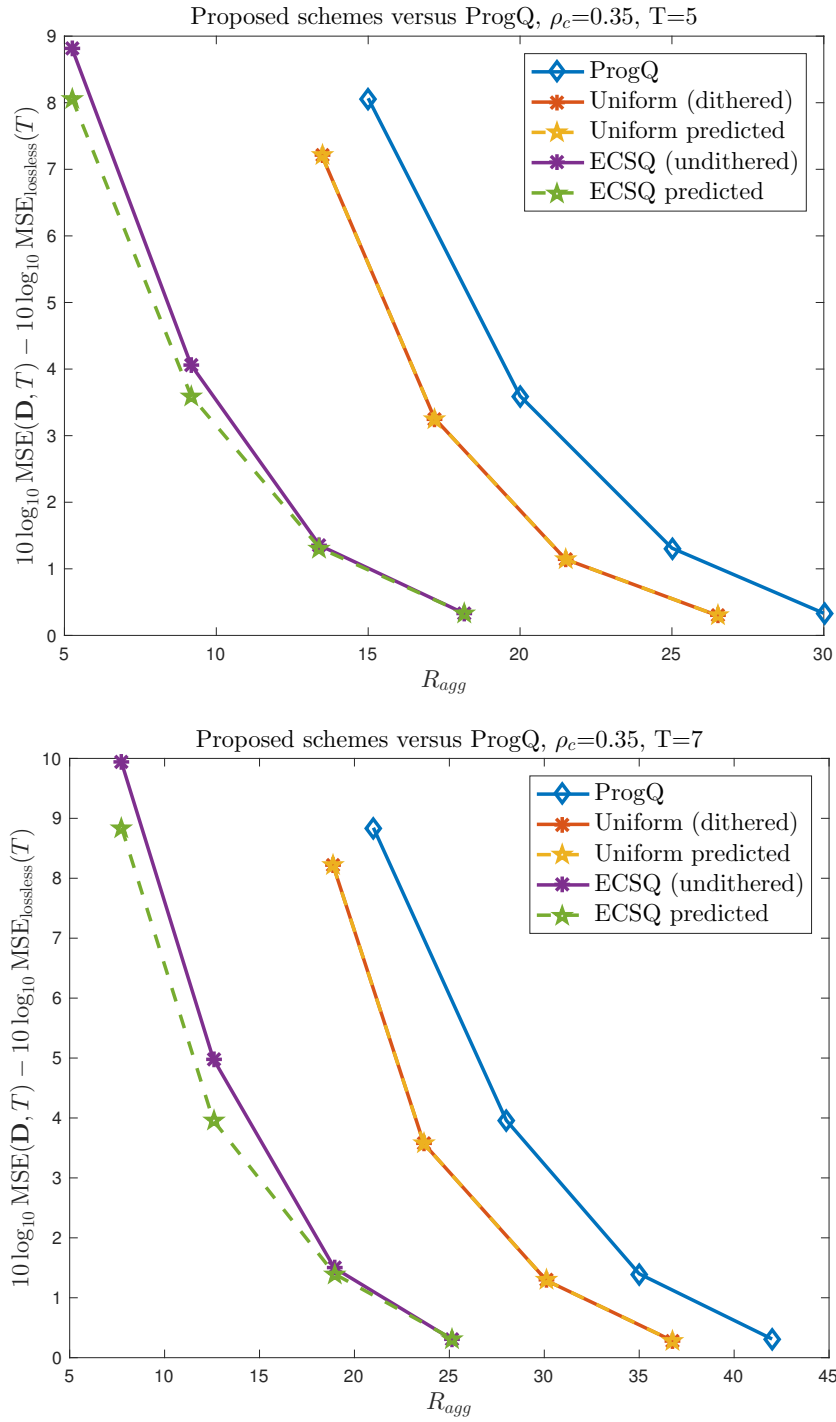


Figure 4.9 RD trade-off curves for the proposed schemes versus ProgQ [20]. All schemes were simulated for random geometric graphs on a unit torus with connectivity radius $\rho_c = 0.35$. *Top*: number of iterations $T = 5$. *Bottom*: number of iterations $T = 7$.

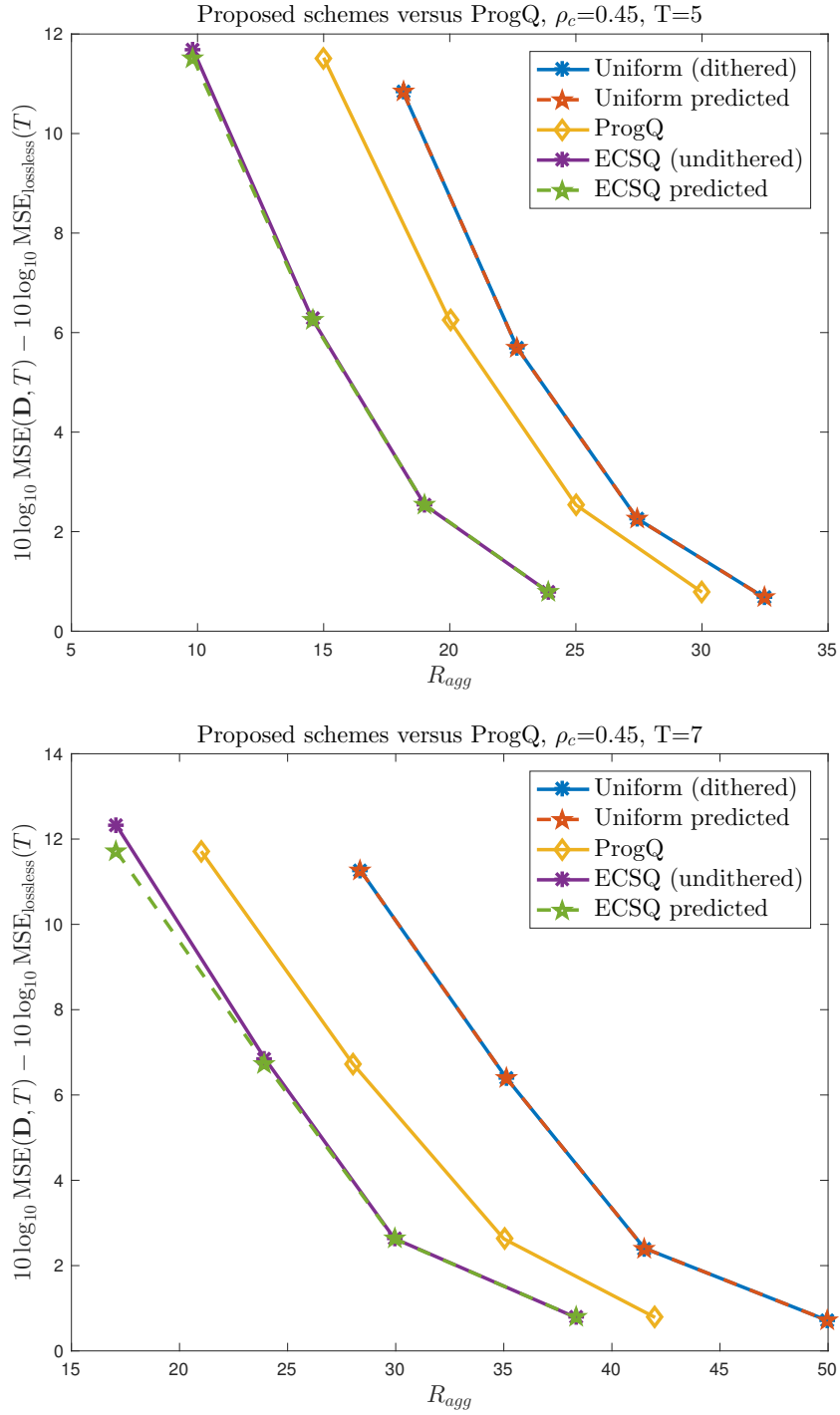


Figure 4.10 RD trade-off curves for the proposed schemes versus ProgQ [20]. All schemes were simulated for random geometric graphs on a unit torus with connectivity radius $\rho_c = 0.45$. *Top*: number of iterations $T = 5$. *Bottom*: number of iterations $T = 7$.

The performance advantage of the predictive scheme stems from its use of linear minimum MSE (LMMSE) prediction [35] to reduce the source variance prior to quantization. As the connectivity parameter ρ_c and total iteration count T increase, there is no significant change in performance relative to the proposed schemes. This lack of improvement may be due to the previously discussed nonzero asymptotic MSE. Clearly, the advantages of the predictive quantization strategy combined with entropy coding and the modified state update of Frasca *et al.* [14] would produce better results, and it is worthwhile to explore whether LMMSE prediction can be incorporated into our framework.

4.2.4 Comparison to the ProgQ scheme

Unlike the predictive scheme [21], ProgQ [20] converges to the true sample average in the limit $t \rightarrow \infty$. In the low-connectivity ($\rho_c = 0.35$) setting, the ProgQ scheme is outdone by our fixed-rate uniform quantizer. Increasing the connectivity to $\rho_c = 0.45$ also increases the speed of convergence. Because the ProgQ scheme uses differential coding, faster convergence leads to lower source variance, and hence lower distortion for a given coding rate (3.36). Therefore, the performance of the ProgQ scheme improves relative to our approaches with higher ρ_c . Furthermore, from Figure 4.10, it is clear that this advantage becomes more significant as the iteration count T increases. Even with constant coding rates, the ProgQ scheme can converge in the limit; however, it is clear from Figure 4.2 that the optimal sequences for our schemes grow with t . Therefore, asymptotically, we expect that ProgQ will outdo our proposed schemes, despite our constrained optimization approach.

In this thesis, we presented a new approach to reducing the communication cost in distributed average consensus using lossy compression and generalized geometric programming (GGP) modeling in the case of initial Gaussian states with entropy-coded scalar uniform quantization (ECSQ), or the case of an arbitrary distribution using fixed-rate uniform quantization. The main results in Chapter 3 of this thesis are (i) the proof that the problem of minimizing the coding rate for a particular number of consensus iterations T , subject to a constraint on the mean square error (MSE), can be posed as a GGP, and (ii) to a lesser extent, the state evolution equations for consensus using Frasca's modified iteration [14]. This analysis allows the minimization of communication cost to be tackled with a structured and principled approach that may be promising for future research.

Numerical results in Chapter 4 verify the state-evolution equations and hint at properties of the optimal rate sequences in the absence of side information at the encoder and decoder. Additionally, in the case that fixed-rate coding is used, a heuristic is presented in Section 3.5.2 that reduces the optimization complexity substantially. The main advantage of the proposed approach is the ability to specify fine trade-offs among accuracy, run time, and communication cost as measured by the aggregate coding rate.

From the comparisons to prior art in Chapter 4, it can be seen that even simple incorporation of side information into the coding strategy can result in significant performance improvements in terms of aggregate rate. Using differential coding, as in the approach of Thanou *et al.* [20], is simple. Exploring its effect on performance would make an interesting future research topic.

Unfortunately, the optimization (3.38) does not scale well, and it may be worthwhile to study the properties of the optimal rate sequences to design some theoretically motivated heuristics, or to investigate alternative optimization implementations that do not rely on explicit GGP modeling. In the case of uniform quantization with fixed-rate coding, numerical results showcased the effectiveness of our proposed heuristic approach. For each simulation conducted, the reduced complexity search heuristic found the same result as an exhaustive search, with a significant reduction in run time.

On the applied front, it would be interesting to explore potential uses of this optimized form of consensus in existing distributed algorithms. One potential path forward is to use consensus to perform the pseudodata fusion step in row-partitioned multiprocessor approximate message passing [47–51]. Because of the nonhomogeneous MSE evolution at each node, this becomes a nontrivial problem. The distributed nature of signal acquisition in sensor networks may make approximate message passing in the multimeasurement vector framework [83] appropriate. Other potential areas for application include dictionary learning [9] and computer vision [8].

Another possible application of this work is in information theory. With relaxations of some of the state-update assumptions, it may be possible to numerically compute the operational rate distortion (RD) relationship for synchronous-time consensus for Gaussian data using our optimization (3.38). This would represent an upper bound on the operational RD relationship in consensus, and it might offer insight into the continuing information-theoretic investigation of source coding for distributed average consensus.

BIBLIOGRAPHY

- [1] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000.
- [2] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Comput.*, vol. 1, no. 1, pp. 59–69, Jan. 2002.
- [3] *Amazon EC2*, <https://aws.amazon.com/ec2/>.
- [4] N. Noorshams and M. J. Wainwright, "Non-asymptotic analysis of an optimal algorithm for network-constrained averaging with noisy links," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 833–844, Aug. 2011.
- [5] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.
- [6] 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, Jan. 2007.
- [7] M. Nokleby, W. U. Bajwa, R. Calderbank, and B. Aazhang, "Toward resource-optimal consensus over the wireless medium," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 284–295, Apr. 2013.
- [8] T. Tron and R. Vidal, "Distributed computer vision algorithms through distributed averaging," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition (CVPR)*, Providence, RI, Jun. 2011, pp. 57–63.
- [9] H. Raja and W. U. Bajwa, "Cloud K-SVD: A collaborative dictionary learning algorithm for big, distributed data," *IEEE Trans. Signal Process.*, vol. 64, no. 1, pp. 173–188, Jan. 2016.
- [10] M. H. DeGroot, "Reaching a consensus," *J. Amer. Statist. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.
- [11] V. Borkar and P. Varaiya, "Asymptotic agreement in distributed estimation," *IEEE Trans. Autom. Control*, vol. AC-27, no. 3, pp. 650–655, Jun. 1982.
- [12] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," PhD thesis, Massachusetts Inst. Technol., Cambridge, MA, Nov. 1984.
- [13] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

- [14] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, "Average consensus on networks with quantized communication," *Int. J. Robust Nonlinear Control*, vol. 19, no. 16, pp. 1787–1816, Nov. 2008.
- [15] 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, May 2009.
- [16] M. E. Chamie, J. Liu, and T. Başar, "Design and analysis of distributed averaging with quantized communication," in *Proc. 53rd IEEE Conf. Decision Control*, Dec. 2014, pp. 3860–3865.
- [17] R. Carli, F. Fagnani, P. Frasca, and S. Zampieri, "Efficient quantization for average consensus," *Arxiv preprint arXiv:0903.1337*, Mar. 2009.
- [18] J. Fang and H. Li, "An adaptive quantization scheme for distributed consensus," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2009, pp. 2777–2780.
- [19] 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, Oct. 2008.
- [20] 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, Jan. 2013.
- [21] M. E. Yildiz and A. Scaglione, "Coding with side information for rate-constrained consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3753–3764, Aug. 2008.
- [22] —, "Limiting rate behavior and rate allocation strategies for average consensus problems with bounded convergence," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2008, pp. 2717–2720.
- [23] C. Mosquera, R. López-Valcarce, and S. K. Jayaweera, "Stepsize sequence design for distributed average consensus," *IEEE Signal Process. Lett.*, vol. 17, no. 2, pp. 169–172, Feb. 2010.
- [24] R. Rajagopal and M. J. Wainwright, "Network-based consensus averaging with general noisy channels," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 373–385, Jan. 2011.
- [25] 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, Nov. 2009.
- [26] S. Zhu, S. Liu, J. Xu, Y. C. Soh, and L. Xie, "Averaging based distributed estimation algorithm for rate-constrained sensor networks with additive quantization model," in *Proc. IEEE Ann. Conf. Decision Control (CDC)*, Osaka, Japan, Dec. 2015, pp. 6245–6250.

- [27] 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, Mar. 2010.
- [28] Y. Huang and Y. Hua, "On energy for progressive and consensus estimation in multihop sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 3863–3875, Aug. 2011.
- [29] T. Berger, *Rate Distortion Theory: Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
- [30] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience, 1991.
- [31] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.
- [32] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distributed Comput.*, vol. 67, no. 1, pp. 33–46, Jan. 2007.
- [33] F. F. C. Rego, Y. Pu, A. Alessandretti, A. P. Aguiar, and C. N. Jones, "A consensus algorithm for networks with process noise and quantization error," in *Proc. 53rd Annu. Allerton Conf. on Commun., Control, and Comput.*, Allerton House, UIUC, IL, Oct. 2015, pp. 488–495.
- [34] 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, Feb. 2011.
- [35] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1993.
- [36] M. E. Yildiz and A. Scaglione, "Differential nested lattice encoding for consensus problems," in *Proc. 6th Int. Symp. Inform. Process. Sensor Networks (IPSN)*, Cambridge, MA, Apr. 2007, pp. 89–98.
- [37] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.
- [38] S. Zhu, Y. C. Soh, and L. Xie, "Distributed parameter estimation with quantized communication via running average," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4634–4646, Sep. 2015.
- [39] D. Thanou, H. Park, E. Kokiopoulou, and P. Frossard, "Polynomial filter design for quantized consensus," in *Proc. 18th European Signal Process. Conf. (EUSIPCO)*, Aalborg, Denmark, Aug. 2010, pp. 184–188.

- [40] J. Fang and H. Li, "Distributed consensus with quantized data via sequence averaging," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 944–948, Feb. 2010.
- [41] O. Ayaso, D. Shah, and M. A. Dahleh, "Information theoretic bounds for distributed computation over networks of point-to-point channels," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6020–6039, Dec. 2010.
- [42] A. Xu and M. Raginsky, "A new information-theoretic lower bound for distributed function computation," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Honolulu, HI, Jun. 2014, pp. 2227–2231.
- [43] H.-I. Su and A. El Gamal, "Distributed lossy averaging," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3422–3437, Jul. 2010.
- [44] Y. Yang, P. Grover, and S. Kar, "Rate distortion for lossy in-network function computation: Information dissipation and sequential reverse water-filling," *IEEE Trans. Inf. Theory*, vol. PP, no. 99, pp. 1–29, May 2017.
- [45] M. Penrose, *Random Geometric Graphs*. New York, NY: Oxford University Press, 2003.
- [46] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Mixing times for random walks on geometric random graphs," in *ALENEX/ANALCO*, Vancouver, British Columbia, Canada, Jan. 2005, pp. 240–249.
- [47] J. Zhu, D. Baron, and A. Beirami, "Optimal trade-offs in multi-processor approximate message passing," *Arxiv preprint arXiv:1601.03790*, Nov. 2016.
- [48] J. Zhu, A. Beirami, and D. Baron, "Performance trade-offs in multi-processor approximate message passing," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 680–684.
- [49] J. Zhu, "Statistical physics and information theory perspectives on linear inverse problems," PhD thesis, North Carolina State University, Raleigh, NC, Jan. 2017.
- [50] P. Han, J. Zhu, R. Niu, and D. Baron, "Multi-processor approximate message passing using lossy compression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 6240–6244.
- [51] J. Zhu, R. Pilgrim, and D. Baron, "An overview of multi-processor approximate message passing," in *Proc. 51st IEEE Conf. Inform. Sci. Syst.*, Baltimore, MD, Mar. 2017.
- [52] B. Widrow and I. Kollár, *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*. New York, NY: Cambridge University Press, 2008.

- [53] 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.
- [54] A. Brandstädt, T. Nishizeki, K. Thulasiraman, and S. Arumugam, Eds., *Handbook of Graph Theory, Combinatorial Optimization, and Algorithms*, ser. Chapman and Hall/CRC Computer and Information Science Series. Boca Raton, FL: Chapman and Hall/CRC, 2016, vol. 34.
- [55] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, pp. 379–423, 623–656, 1948.
- [56] T. Wiegand and H. Schwarz. (Dec. 2012). Quantization, [Online]. Available: https://www.ic.tu-berlin.de/fileadmin/fg121/Source-Coding_WS12/05_Quantization-WS12.pdf.
- [57] S. Arimoto, "An algorithm for computing the capacity of an arbitrary discrete memoryless channel," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 14–20, Jan. 1972.
- [58] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 460–473, Jul. 1972.
- [59] K. Rose, "A mapping approach to rate-distortion computation and analysis," *IEEE Trans. Inf. Theory*, vol. 40, no. 6, pp. 1939–1952, Nov. 1994.
- [60] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
- [61] J. Max, "Quantization for minimum distortion," *IRE Trans. Inf. Theory*, vol. 6, no. 1, pp. 7–12, Mar. 1960.
- [62] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. Inst. Radio Eng.*, vol. 9, no. 40, pp. 1098–1101, Sep. 1952.
- [63] H. Gish and J. Pierce, "Asymptotically efficient quantizing," *IEEE Trans. Inf. Theory*, vol. 14, no. 5, pp. 676–683, Sep. 1968.
- [64] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. IT-44, pp. 2325–2383, Oct. 1998.
- [65] N. Farvardin and J. W. Modestino, "Optimum quantizer performance for a class of non-gaussian memoryless sources," *IEEE Trans. Inf. Theory*, vol. IT-30, no. 3, pp. 485–497, May 1984.
- [66] R. Zamir and M. Feder, "On lattice quantization noise," *IEEE Trans. Inf. Theory*, vol. 42, no. 4, pp. 1152–1159, Jul. 1996.

- [67] A. B. Sripad and D. L. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-25, no. 5, pp. 442–448, Oct. 1977.
- [68] E. W. Weisstein. (2017). Matrix direct sum. From MathWorld—a Wolfram Web Resource, [Online]. Available: <http://mathworld.wolfram.com/MatrixDirectSum.html>.
- [69] P. Kabal, "Correlation properties of quantization noise," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Prague, Czech Republic, May 2011, pp. 5244–5247.
- [70] E. Aykol and K. Rose, "On constrained randomized quantization," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3291–3302, Jul. 2013.
- [71] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optimization Eng.*, vol. 8, no. 67, pp. 67–127, Mar. 2007.
- [72] P. F. Panter and W. Dite, "Quantization distortion in pulse-count modulation with nonuniform spacing of levels," *Proc. IRE*, vol. 39, no. 1, pp. 44–48, Jan. 1951.
- [73] M. Grant and S. Boyd, *CVX: MATLAB software for disciplined convex programming, version 2.1*, <http://cvxr.com/cvx>, Mar. 2014.
- [74] —, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds., http://stanford.edu/~boyd/graph_dcp.html, Springer-Verlag Limited, 2008, pp. 95–110.
- [75] E. Burnell and W. Hoburg, *GPkit software for geometric programming*, <https://github.com/hoburg/gpkit>, Version 0.5.2, 2017.
- [76] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Mach. Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
- [77] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. CACSD Conf.*, Taipei, Taiwan, 2004.
- [78] MATLAB, *Release 2017a*. Natick, MA: The MathWorks, Inc., 2017.
- [79] E. W. Weisstein. (2017). Torus. From MathWorld—a Wolfram Web Resource, [Online]. Available: <http://mathworld.wolfram.com/Torus.html>.
- [80] —, (2017). Square torus. From MathWorld—a Wolfram Web Resource, [Online]. Available: <http://mathworld.wolfram.com/SquareTorus.html>.
- [81] K. Hoffman and R. Kunze, *Linear Algebra*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1971.

- [82] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Inform. Process. Sensor Networks (IPSN)*, Apr. 2005, pp. 63–70.
- [83] J. Zhu, D. Baron, and F. Krzakala, "Performance limits for noisy multimeasurement vector problems," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2444–2454, May 2017.