Compressed Sensing with Applications in Wireless Networks

Markus Leinonen[1], Marian Codreanu[2] and Georgios Giannakis[3]

[1] *Centre for Wireless Communications, University of Oulu; markus.leinonen@oulu.fi*

[2] *Department of Science and Technology, Linköping University; marian.codreanu@liu.se*

[3] *Department of Electrical and Computer Engineering, University of Minnesota; georgios@umn.edu*

ABSTRACT

Sparsity is an attribute present in a myriad of natural signals and systems, occurring either inherently or after a suitable projection. Such signals with lots of zeros possess minimal degrees of freedom and are thus attractive from an implementation perspective in wireless networks. While sparsity has appeared for decades in various mathematical fields, the emergence of compressed sensing (CS) – the joint sampling and compression paradigm – in 2006 gave rise to plethora of novel communication designs that can efficiently exploit sparsity. In this monograph, we review several CS frameworks where sparsity is exploited to improve the quality of signal reconstruction/detection while reducing the use of radio and energy resources by decreasing, e.g., the sampling rate, transmission rate, and number of computations. The first part focuses on several advanced CS signal reconstruction techniques along with wireless applications. The second part deals with efficient data gathering and lossy compression techniques in wireless sensor networks. Finally, the third part addresses CS-driven designs for spectrum sensing and multi-user detection for cognitive and wireless communications.

Markus Leinonen, Marian Codreanu and Georgios Giannakis (2019), Compressed Sensing with Applications in Wireless Networks, Foundations and Trends® in Signal Processing: Vol. 13, No. 1-2, pp 1–282. DOI: 10.1561/2000000107.

1

Introduction

Efficient data acquisition and signal reconstruction methods used for various purposes in wireless communication systems have been under extensive research and development for a while. One overarching design objective is to improve the quality of reconstruction, classification, detection etc. while reducing the use of radio and energy resources by decreasing, e.g., the sampling rate, transmission rate, and number of computations. Due to proliferation of the number of deployed wireless devices and the amount of data, implementing advanced signal processing techniques to achieve such objectives becomes of particular interest for emerging multi-user communication systems.

One fundamental feature of a desired information signal that can provide a solution to reach the aforementioned goal is signal *sparsity*. To this end, the key tool for exploiting sparsity is the modern theory of *compressed sensing/compressive sampling* (CS). Accordingly, this monograph reviews several CS techniques to utilize the sparsity of an underlying signal in data gathering, signal reconstruction and detection tasks in wireless networks. In Section 1, we first give a general overview of the different communication frameworks and applications addressed in the monograph. As a continuation, Section 1 elaborates signal sparsity and discusses how sparsity may be present or become available in the considered frameworks. The main idea of CS is detailed in Section 1. Finally, the outline of the monograph is given in Section 1.

Communication Frameworks and Applications

Cognitive Radio Communications: It has been recognized that the licensed radio frequency (RF) spectrum is often severely under-utilized depending on the time and location of communication, in spite of the evident scarcity of the spectral resources due to the growing use of wireless devices [105]. *Cognitive radios (CRs)* aim to mitigate this issue by opportunistically utilizing the unused licensed spectrum through spectrum sensing and dynamic spectrum access in multi-user communication systems.

In particular, *RF cartography* is an instrumental concept for CR tasks [160]. Based on the measurements collected by spatially distributed CR sensors, RF cartography constructs the maps over the space, time, and frequency, portraying the RF landscape in which the CR network is deployed. Notable RF maps that have been proposed include the power spectral density maps, which acquire the ambient interference power distribution, revealing the crowded regions that CR transceivers need to avoid [21]; and the channel gain maps, which capture the channel gains between any two points in space, allowing CR networks to perform accurate spectrum sensing and aggressive spatial reuse [161]. By modelling the channel gains as the tomographic accumulations of an underlying spatial loss field (SLF), the technique captures the attenuation in the signal strength due to the obstacles in the propagation path.

Multi-User Detection: Multiuser detection (MUD) algorithms play a major role for mitigating multi-access interference present in code-division multiple access (CDMA) systems; see e.g., [288] and references therein. These well-appreciated MUD algorithms simultaneously detect the transmitted symbols of all active user terminals. Conventional techniques require knowledge of which terminals are active and exploit no possible user (in)activity. In the design of practical CDMA systems, one is also interested in saving bandwidth and power resources. Such savings are possible by reducing the size of the required spreading gain, which in turn reduces latency and energy consumption. This may be enabled by a low activity factor (probability of each user being active) in which case the system can be designed for spreading gains smaller than the number of candidate users.

Sensor Monitoring: In near future, there will be a burgeoning demand for the deployment of low-power smart sensors, especially to serve the myriad of diverse *internet of things* (IoT) applications including environmental, industrial, healthcare, and military monitoring tasks [316, 119, 213]. Accordingly, *wireless sensor networks* (WSNs) consisting of battery-powered sensors will be a key technology in creating the ubiquitous networked world and smart cities under the IoT framework. IoT opens a new era of intelligent networking, where collaborative sensors sense their environment with no human intervention, enabling to, e.g., automate an underlying process, improve the system performance, and reduce the maintenance costs. By 2020, the number of IoT devices is anticipated to reach hundreds of billions and the IoT market to become on the order of trillions of dollars, with a major portion on healthcare applications [119, 213].

In a typical monitoring task, geographically distributed sensors measure a correlated information source, encode the observations separately, and communicate the information to a sink for joint reconstruction of the source signals. As the sensors have limited batteries, which are often non-rechargeable or irreplaceable, it is crucial to minimize the energy consumption to prolong the lifespan of a WSN. The main contributors to sensors' energy consumption are wireless communications [239], and in certain applications, also the sensing/sampling part [6]. Consequently, it is crucial to minimize the amount of information (i.e., the number of data packets or bits) that must be communicated from each sensor to the sink in order to satisfy given application requirements. Accordingly, an energy-efficient sensor acquires a small number of data samples of a physical phenomenon (e.g., temperature, humidity, or light intensity), and encodes and communicates them at a minimum rate to the sink to reconstruct the information signal with, e.g., a given fidelity or maximum allowed delay. This engenders the need for energy-efficient distributed data gathering techniques that preserve autonomous operation of sensors and have a simple infrastructure with low battery consumption and computational complexity.

Signal Sparsity in the Considered Applications

Sparsity is an attribute present in a plethora of natural signals and systems, occurring either naturally or after projecting them over appropriate bases. A

signal is said to be *sparse* if it has many zero-valued elements or can be represented by few non-zero coefficients under a proper transformation. Similarly, a signal is termed *compressible*, if the energy of transform coefficients is concentrated in a small set of elements. Focusing on sparse signal structures is reasonable not only because nature itself is parsimonious but also because processing and simple models with minimal degrees of freedom are attractive from an implementation perspective. Whereas sparsity is by default an attribute defined for a single signal, it can be extended to characterize joint sparse structures; see, e.g., the regression problems dealing with *group sparsity* [314], and *joint sparsity models* introduced in [19].
Exploitation of sparsity is critical in a wide range of communication tasks. Regarding the frameworks addressed in this monograph, utilization of sparse signals has been investigated in the following applications.

- **WSNs:** Sparse signals are encountered in diverse WSN applications in, e.g., environmental monitoring [19, 238], source localization [201], and biomedical sensing [83]. For instance, universal transformations suitable for revealing the underlying sparsity of many smooth/piecewise smooth signals include the discrete Fourier, cosine, and wavelet transform (DFT, DCT, DWT), respectively [202, 15, 36, 37]. In particular, the efficacy of the DWT matrices in sparsifying signals of several natural phenomena such as temperature, humidity, and light has been especially reported in, e.g., [197, 37]. Sparsity in the sensed data allows to reduce the computations of simple sensor devices, and most importantly, to significantly reduce sensors' energy consumption for communicating the data to a fusion center [175, 176, 178]. One particular direction of great interest is so-called *quantized CS* [129, 273, 333, 151, 180, 184, 182, 181] – a lossy compression setup where the CS measurements are converted into finite-rate bit sequences, and the aim is to design efficient quantization-aware CS reconstruction algorithms and analyze their rate-distortion performance. Application requirements may demand as low quantization rate as one bit per measurement sample, referred as *1-bit CS* [152].

- **Cellular Networks:** Sparsity has been utilized in, e.g., estimation of wireless multipath channels [71, 275], estimation of parameters of communication systems [53, 23, 9, 109], and sparse sphere decoding [277]. As for MUD algorithms, the sparsity arises because the active terminals are unknown and the activity factor is low in a typical system [322]. Sparsity can be exploited to either relax or judiciously search over subsets of the alphabet of the desired symbol vector so that the resultant MUD algorithms trade off optimality in detection error performance with computational complexity. Moreover, source localization based on the DoA estimation typically involves sparsity in the angular domain under certain radio propagation features [118, 201, 228].

- **CR Applications:** Distributed spectrum sensing for CR communications is a crucial task and has been addressed in, e.g., [20, 10, 212, 261]. In spectrum sensing, the sparsity manifests itself in two forms: 1) narrow-band nature of transmit power spectral density relative to the broad range of usable spectrum, and 2) sparsely located active radios in the operational space [20]. This type of *compressive* wideband power spectrum estimation allows to recover an unknown power spectrum of a wide-sense stationary signal from samples obtained at a sub-Nyquist rate [10], even if the samples were coarsely quantized sensors' measurements [212].

  Another emergent topic in CR networks is RF cartography. In RF cartography, SLF may have a low-rank structure potentially corrupted by sparse outliers [171]. Such a model is particularly appealing for urban and indoor propagation scenarios, where regular placement of buildings and walls renders a scene inherently of low rank, while sparse outliers can pick up the artifacts that do not conform to the low-rank model. Earlier works on sparsity-leveraging cartography include network anomaly monitoring in [205].

Applications from other research fields which deal with sparse signals include variable selection in linear regression models for diabetes [278], image compression [46], signal decomposition using overcomplete bases [61], and more.

The Concept of Compressive Sampling

Sampling Followed by Compression: The key principle underlying the data sampling methods and analog-to-digital conversion in modern consumer devices is the Nyquist-Shannon sampling theorem – the celebrated result of the seminal works by Nyquist [227] and Shannon [259]. The theorem states that if the sampling rate of a signal is at least twice its maximum frequency component, the signal can be perfectly reconstructed. This threshold rate is called the Nyquist rate. In a resource-limited digital sensor, the acquisition of signal samples is typically followed by data compression which aims to encode the information with fewer bits. Consequently, a substantial portion of expensively acquired data is eventually discarded prior to storage or transmission. Fortunately, if a signal has certain additional features, perfect reconstruction may be possible even below the Nyquist rate. Namely, the inefficiency caused by the separate sampling and compression may be alleviated by *sub-Nyquist sampling* – an unorthodox paradigm violating the conventional sampling notion.

Compressive Sampling: A feature that enables sub-Nyquist sampling is the sparsity/compressibility of a signal, discussed in Section 1. While sparsity has been exploited for a while in numerical linear algebra, statistics, and signal processing, renewed interest emerged in recent years because sparsity plays an instrumental role in modern *compressive sampling/compressed sensing* (CS) theory and applications [43, 87, 44, 146, 46, 38, 94, 17, 49]. CS is a joint sampling and compression paradigm which enables a sparse/compressible length-N signal to be accurately reconstructed from its $M < N$ (random) linear measurements. This engenders the sub-Nyquist sampling interpretation of CS [217]: instead of sampling at a rate proportional to the signal bandwidth, the sampling rate in CS is dictated by the signal's "information content" [218]. The primary asset of CS is its simple and universal encoding since most computational work load is shifted to the decoder [94]. As a rough comparison, computational complexity at the encoder for CS scales as MN (at most for a dense measurement matrix), whereas for a standard compression method like fast Fourier transform it scales as $N\log N$ [112, Appendix C.1]. While for high-dimensional signals $MN > N\log N$, the use of sparse measurements matrices drastically reduces the computational and memory requirements for CS [28]. The other benefits of CS include robustness to measurement/quantization noise, resiliency to packet losses, security via pseudorandom projections, and the gradual improvement of reconstruction accuracy from increased number of measurements [94]. All above benefits are especially beneficial for low-power sensor applications.

A Historical Note: Prior to the CS era launched in 2006, there has already been lots of research interest to tackle signal processing tasks involving sparse signals. This becomes also evident in Table 1 of Section 2 where many popular present-day CS reconstruction algorithms trace back to early 1990's. Indeed, in a diversity of engineering applications, one encounters solving ill-posed/underdetermined inverse problems, i.e., problems where the number of available measurements is smaller than the dimension of the signal/model to be estimated. Luckily, in many such practical situations, the encountered models have structural constraints similar to sparsity, i.e., they can be described by only a few degrees of freedom relative to their ambient dimension or as linear combinations of a few basic building blocks.

One cornerstone of the emerging area of CS is the notion of *variable selection* (VS) associated with sparse linear regression [278]. VS is a combinatorially complex task closely related (but not identical) to the well-known *model order selection* problem tackled through Akaike's information [2], Bayesian information [258], and risk inflation [111] criteria. A typically convex function of the model fitting error is penalized with the ℓ_0-norm of the unknown vector which equals the number of nonzero entries, and thus accounts for model complexity (degrees of freedom). To bypass the non-convexity of the ℓ_0-norm, VS and CS approaches replace it with convex penalty terms (e.g., the ℓ_1-norm) that capture sparsity but also lead to computationally efficient solvers. One another line of work preceding the CS era is the concept "sampling signals with *finite rate of innovation*" introduced in [289], which generalizes the classic sampling theorem of bandlimited signals with sinc kernels. The rate of innovation is a number that describes a finite number of degrees of freedom (i.e., sparsity) per unit of time for certain classes of signals.

Outline of the Monograph

This monograph addresses several CS techniques to utilize sparsity of an underlying signal in data gathering, signal reconstruction and detection tasks in wireless networks. In this monograph, the "sub-Nyquist feature" of CS refers to measuring an underlying (continuous-time) analog source via dimensionality reducing projections, which can be represented by discrete-time measurements of form (1) (see also Fig. 1). Another line of work is analog-to-digital compression, where analog signals are encoded into bits via a combined sub-Nyquist sampling and quantization process [162]. A hardware implementation of a sub-Nyquist sampling system was presented in [215], and a unified Xampling framework was introduced in [216].

This monograph is organized as follows. We first introduce fundamentals of CS and give an overview about common reconstruction techniques of sparse signals from compressed measurements in Section 2. After that, the monograph is split into three parts as follows:

- The first part focuses on several advanced CS signal reconstruction techniques along with wireless applications. Accordingly, Section 3 develops sparsity-aware recursive algorithms for estimating and tracking sparse and (possibly time-varying) signals. Section 4 devises regularized total least-squares algorithms under sparsity constraints for a perturbed CS signal model, along with applying them to directions-of-arrival estimation.

- The second part deals with efficient data gathering and lossy compression techniques in wireless sensor networks. Compressed acquisition of streaming correlated data in WSNs is presented in Section 5. Section 6 and Section 7 consider CS signal acquisition setups under quantization of measurements. Accordingly, Section 6 devises an efficient quantized CS algorithm for distributed source coding of correlated sparse sources in WSNs. Rate-distortion performance of a quantized CS setup is investigated in Section 7, including both information theoretic analyses and the design of several types of practical quantized CS algorithms.

- The third part addresses CS-driven designs for spectrum sensing and multi-user detection for cognitive and wireless communications. Section 8 addresses channel gain cartography for CR networks under a limited number of available measurements. Section 9 proposes efficient sparsity-utilizing MUD algorithms in CDMA systems.

Finally, the monograph is concluded with the summary in Section 10.

2

Fundamentals of Compressed Sensing

Compressed Sensing Basics

The standard CS problem is introduced herein. Let $\mathbf{x} \in \mathbb{R}^N$ be a real-valued vector that can be represented in basis $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$ as $\mathbf{x} = \mathbf{\Psi}\boldsymbol{\theta}$, where $\boldsymbol{\theta} = [\theta_1 \cdots \theta_N]^{\mathrm{T}}$ is the transform domain coefficient vector. Vector $\mathbf{x}$ is K-*sparse* in basis $\mathbf{\Psi}$ if $\boldsymbol{\theta}$ has at most $K \leq N$ non-zero entries, i.e., $\|\boldsymbol{\theta}\|_0 \leq K$, where $\|\boldsymbol{\theta}\|_0 = |\mathrm{supp}(\boldsymbol{\theta})|$, $\mathrm{supp}(\boldsymbol{\theta}) = \{i|\theta_i \neq 0, i = 1, \ldots, N\}$ is the support of $\boldsymbol{\theta}$, and for a discrete set $|\cdot|$ denotes its cardinality. While many natural signals are not exactly sparse, they are termed *compressible*, if the energy in $\boldsymbol{\theta}$ is concentrated, i.e., the ordered coefficients $|\theta_{(1)}| \geq \ldots \geq |\theta_{(N)}|$ exhibit a power law decay satisfying $|\theta_{(i)}| \leq C_p i^{-p}$, $i = 1, \ldots, N$, where $p \geq 1$ affects the rate of decay, and C_p is a constant depending only on p [38, 44, 145, 46, 81]. This signal class includes smooth and piecewise smooth signals, and images with bounded variations [44, 38].
A CS based sensor acquires $M \leq N$ linear measurements $\mathbf{y} \in \mathbb{R}^M$ of $\mathbf{x}$ as

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{\Psi}\boldsymbol{\theta}, \tag{1}$$

where $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is a fixed (and known) measurement matrix. From the compression perspective, it is desired that $M << N$. A CS measurement setup for $N = 9$, $M = 5$, and $K = 3$ is illustrated in Fig. 1.

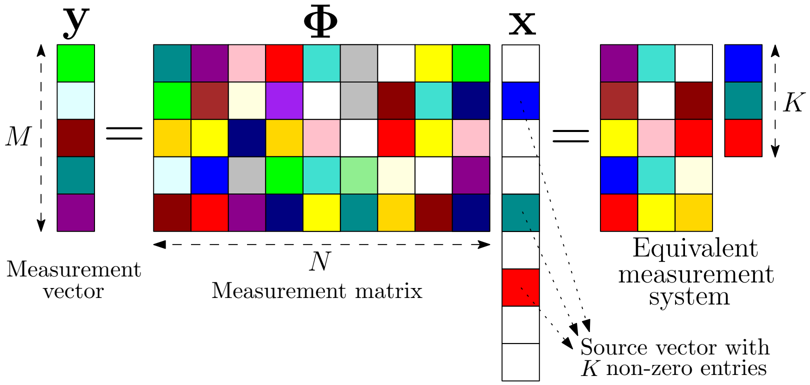


Figure 1: A CS measurement setup for $N = 9$, $M = 5$, and $K = 3$.

For reconstruction, one seeks the sparsest $\mathbf{x}$ that is consistent with measurements $\mathbf{y}$. For $M \geq 2K$, a measurement matrix $\boldsymbol{\Phi}$ exists so that the optimal solution can be found via a combinatorial (non-deterministic polynomial-time) hard problem [112, Theorem 2.14]

$$\hat{\boldsymbol{\theta}} := \underset{\tilde{\boldsymbol{\theta}}}{\operatorname{argmin}} \|\tilde{\boldsymbol{\theta}}\|_0 \quad \text{such that} \quad \mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\tilde{\boldsymbol{\theta}}, \tag{2}$$

resulting in an estimate $\hat{\mathbf{x}} = \boldsymbol{\Psi}\hat{\boldsymbol{\theta}}$. Owing to the main result of CS, the sparsity-promoting ℓ_0-term is replaced by its best convex approximation, the ℓ_1-norm, and one solves the *basis pursuit* problem [45, 38, 46, 87, 44, 88, 16]

$$\hat{\boldsymbol{\theta}} := \underset{\tilde{\boldsymbol{\theta}}}{\operatorname{argmin}} \|\tilde{\boldsymbol{\theta}}\|_1 \quad \text{such that} \quad \mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\tilde{\boldsymbol{\theta}}, \tag{3}$$

yielding an estimate $\hat{\mathbf{x}} = \boldsymbol{\Psi}\hat{\boldsymbol{\theta}}$. Since (3) can be cast as a linear program, it can be solved via polynomial-complexity solvers [81, 88]. Note that the encoding (1) is universal in the sense that an appropriate $\boldsymbol{\Psi}$ is selected only in the decoding stage (3).

Indeed, in the noise-free setup assumed above, CS holds promise to address problems as fundamental as solving exactly under-determined systems of linear equations when the unknown vector is sparse [39]. Variants of CS for the "noisy setup" are rooted in the basis pursuit (BP) approach [61], which deals with fitting sparse linear representations to perturbed measurements – a task of major importance for signal compression and feature extraction. The

Lagrangian form of BP is also popular in statistics for fitting sparse linear regression models, using the so-termed least-absolute shrinkage and selection operator (Lasso); see e.g., [278], [144], and references thereof. Least-squares based variants of Lasso were developed in [7] for a recursive sensing setup, and in [324] for a perturbed sensing setup.

Two essential properties of $\mathbf{\Phi}$ and $\mathbf{\Psi}$ which play important roles in stable and accurate CS signal recovery are the restricted isometry property (RIP) of $\mathbf{\Phi\Psi}$ [39, 49, 93], [100, Sect. 1.4.2] and [112, Ch. 6], and the mutual coherence between $\mathbf{\Phi}$ and $\mathbf{\Psi}$ [88, 42, 49, 93] and [100, Sect. 1.4.3]. For instance, $\mathbf{\Phi}$ with i.i.d. Gaussian or binary entries is highly incoherent with any basis $\mathbf{\Psi}$ so that the reconstruction in (3) is exact with high probability if $M \geq C_0 K \log(N/K)$, where C_0 is a positive constant [17, 49] and [112, Ch. 1].

Compressed Sensing Reconstruction Methods

This section discusses briefly several main classes of the existing CS signal reconstruction methods. The focus is on methods that are designed to recover real-valued/analog signals or signals with high-precision numerical representations; CS methods that are devised to operate on quantized measurements are discussed in Sections 6 and 7.

The conventional CS signal reconstruction methods are based on optimization techniques that rely on certain assumptions on the knowledge of the underlying sparsity model, signal dynamics, sensing setup etc. While the ℓ_1-minimization (3) is admittedly the trademark of CS decoding, there exists a variety of sparse signal reconstruction algorithms of different complexity and performance. Table 1 highlights several categories of commonly used existing methods, albeit with no purpose of serving as an exhaustive list. More algorithms can be found in, e.g., [100, Sect. 1.6], [112, Sect. 1.3], [51, Sect. 1.3.2], and a recent review paper [240].

Moreover, instead of sparse signal *reconstruction*, the principles of CS decoding have also been utilized for *support recovery* of sparse signals; see, e.g., [290, 3, 246]. Formulating a support recovery problem as a *multiple hypothesis testing* problem, estimation of a shared support of common sparse signals was analyzed in [274].

Operating in a Compressed Domain: Besides the CS algorithms discussed above, there exist also several other approaches and viewpoints to the compressive signal processing tasks, which, similarly to the ideology behind CS, aim at utilizing sparse structures in an underlying signal to reduce costs related to, e.g., data transmission, storage, and computations. With no purpose of making any strict division between CS and its related frameworks, one important class of compressed domain signal recovery is *subspace methods*. Matched subspace detectors to detect *subspace signals* under subspace interference and broadband noise were proposed in [255]. The authors in [101] introduced a framework for robust and efficient recovery of signals lying in a *union of subspaces* from a given set of samples. Instead of signal reconstruction, [299] addressed the problem of subspace recovery, i.e., aiming at identifying the particular subspace in which the signal lies in. Another class sharing design objectives with CS is *data reduction methods*, which are not restricted to sparse signals per se. Accordingly, a thorough treatment of *sparse sensing* for statistical inference is presented in [63], dealing with an optimal design of deterministic and structured sensing functions to achieve a targeted inference accuracy using a few data samples of a given (possibly non-sparse) signal. The authors in [56] introduced a convex optimization framework that incorporates *atomic norms* as a means of efficient acquisition of signals consisting of a few *atoms*. A data reduction technique, *interval censoring*, was proposed in [222], where, by utilizing statistical distribution of sensor data, distributed estimation of non-sparse sensor signals is possible by receiving measurements from only a subset of sensors. Alternative methods of estimating a sparse parameter vector include two hyperparameter-free methods, SPICE and LIKES, proposed in [270], which can circumvent a laborious search for appropriate hyper-parameters of conventional sparse signal recovery problems.

This monograph: Owing to the fact that efficient reconstruction of sparse signals has merits to, e.g., reduce energy consumption and computational requirements in numerous wireless communication systems, the following two sections – Section 3 and Section 4 – present advanced, general-purpose CS based sparse signal reconstruction techniques, along with several interesting wireless applications. The rest of the sections of the monograph deal with various wireless applications of CS and present sophisticated CS data acquisition and signal processing techniques tailored to each such framework.

Table 1: Various classes of existing CS reconstruction algorithms.

Class	Algorithm
Greedy methods	- Matching pursuit [203] - Orthogonal matching pursuit [231] - Iterative hard thresholding [31] - Subspace pursuit [77] - Compressive sampling matching pursuit (CoSaMP) [224]
Convex optimization	- Basis pursuit [61] - Basis pursuit denoising [61] - Least absolute shrinkage and selection operator (Lasso) [278] - The Dantzig selector [47] - Total variation denoising [252] - Atomic norm minimization [56]
Weighted-norm minimization	- Focal underdetermined system solver (FOCUSS) [128] - Iterative reweighted nonconvex ℓ_p-minimization $(0 < p < 1)$ [60] - Iterative reweighted ℓ_1-minimization [48] - Iteratively reweighted least squares [214] - Homotopy based reweighted ℓ_1-minimization [12]
Bayesian methods	- Sparse Bayesian learning [279, 301, 296] - Bayesian CS [154] - Fast Bayesian matching pursuit [257] - MMSE estimation for CS [99, 235, 281] and [98, Sect. 11.5]
Message-passing algorithms	- Approximate message-passing [89] - Bayesian CS via belief propagation [18] - Generalized approximate message-passing [156]
CS with prior information	- Kalman filtered CS [284] - Least squares CS residual [285] - Modified CS / regularized modified CS [286] - Regularized modified basis pursuit denoising [195] - ℓ_1-ℓ_1-minimization and ℓ_1-ℓ_2-minimization [221, 321]

Part I

Advanced Signal Reconstruction from Compressive Measurements

3

Online Adaptive Estimation of Sparse Signals: Where RLS Meets the ℓ_1-Norm

This section presents an advanced recursive signal reconstruction method for CS. Motivated by the high complexity, increasing memory requirements, and the lack of tracking capability that batch Lasso estimators face when processing observations sequentially, a time-weighted Lasso (TWL) algorithm is devised. Performance analysis reveals that TWL cannot estimate consistently the desired signal support without compromising the rate of convergence. This motivates the development of a time- and norm-weighted Lasso (TNWL) scheme where the ℓ_1-norm weights are obtained from the recursive least-squares (RLS) algorithm. The resultant algorithm consistently estimates the support of sparse signals without reducing the convergence rate. To cope with sparsity-aware recursive real-time processing, adaptive algorithms are also developed to enable online coordinate descent solvers of TWL and TNWL that provably converge to the true sparse signal in the time-invariant case. Simulated tests compare competing alternatives and corroborate the performance of the devised algorithms in estimating time-invariant signals and tracking time-varying signals under sparsity constraints.

Related Works

Research on CS and variable selection (VS) has concentrated on batch processing, and various algorithms for sparse linear regression are available. Those include the basis pursuit and Lasso [278, 61], the Dantzig selector [47], and the ℓ_2-norm constrained ℓ_1-norm minimizer [44]. CS and VS estimators are nonlinear functions of the available observations which they process in a *batch* form using iterative algorithms.

However, many sparse signals encountered in practice must be estimated based on noisy observations that become available *sequentially* in time. For such cases, batch signal estimators typically incur complexity and memory requirements that grow as time progresses. In addition, the sparse signal may vary with time both in its nonzero support, as well as in the values of its nonzero entries. Due to these reasons, adaptive estimation of sparse signals has also been considered. Sequential noise-free signal recovery was considered in [200], and a sparsity-aware least mean-square (LMS) algorithm was pursued in [138]. Sparsity-aware "RLS-like" algorithms were reported in [8], while a framework combining Kalman filtering and CS can be found in [284]. For more discussion on existing works on dynamic and recursive CS, see Section 5.

System Model and Problem Statement

Consider a vector $\mathbf{x}_o \in \mathbb{R}^P$ which is sparse, meaning that only a few of its entries $x_o(p)$, $p = 1, \ldots, P$, are nonzero. Let $\mathcal{S}_{\mathbf{x}_o} := \{p : x_o(p) \neq 0\}$ denote its support, $P_1 := |\mathcal{S}_{\mathbf{x}_o}|$ the number of non-zero entries, and $P_0 := P - P_1$. Sparsity amounts to having $P_1 \ll P_0$. Suppose that such a sparse vector is to be estimated sequentially in time from scalar observations obeying the linear regression model

$$y_n := \mathbf{h}_n^T \mathbf{x}_o + v_n, \quad n = 1, \ldots, N \tag{4}$$

where $\mathbf{h}_n \in \mathbb{R}^P$ is the regression vector at time n, and the additive noise v_n is assumed uncorrelated with $\mathbf{h}_n$, white, with mean zero, and variance σ^2. The goal is to develop sequential and adaptive estimators of $\mathbf{x}_o$ that is a priori known to be sparse, and perhaps slowly varying with n.

The least-squares (LS) criterion is the "workhorse" for linear regression analysis [254, p. 658]. If $\mathbf{y}_N := [y_1, \ldots, y_N]^T$ and $\mathbf{H}_N := [\mathbf{h}_1, \ldots, \mathbf{h}_N]^T$, the LS estimator of $\mathbf{x}_o$ at time N solves the minimization problem

$$\hat{\mathbf{x}}_N^{\text{LS}} := \arg \min_{\mathbf{x} \in \mathbb{R}^P} \|\mathbf{y}_N - \mathbf{H}_N \mathbf{x}\|_2^2. \tag{5}$$

If $N < P$ or $\mathbf{H}_N$ is not full column rank, the problem in (5) does not admit a unique solution. For such cases, minimizing also the ℓ_2-norm of $\mathbf{x}$ renders the LS solver unique, and expressible as $\hat{\mathbf{x}}_N^{\text{LS}} = \mathbf{H}_N^{\dagger} \mathbf{y}_N$, where † denotes matrix pseudo-inverse defined as in, e.g., [127, p. 275].

In the sequential context considered herein, LS faces three challenges: i) increasing memory requirements for storing $\mathbf{y}_N$ and $\mathbf{H}_N$ as N grows large; ii) complexity of order $\mathcal{O}(P^3)$ per time instant n to perform the inversion in $\mathbf{H}_N^{\dagger}$; and iii) lack of capability to track possible variations of $\mathbf{x}_o$ with n.

These challenges are met by the recursive least-squares (RLS) estimator obtained as [254, Chap. 12]

$$\hat{\mathbf{x}}_N^{\text{RLS}} := \arg \min_{\mathbf{x} \in \mathbb{R}^P} \sum_{n=1}^{N} \beta_{N,n} (y_n - \mathbf{h}_n^T \mathbf{x})^2, \quad N = 1, 2, \ldots \tag{6}$$

where the so-called "forgetting factor" $\beta_{N,n}$ describes one of the following data windowing choices:

(w1) *Infinite window* with $\beta_{N,n} = 1$. This choice is adopted for time-invariant signals and, with proper initialization renders RLS equivalent to LS at complexity $\mathcal{O}(P^2)$ per datum.

(w2) *Exponentially decaying window* with $\beta_{N,n} = \beta^{N-n}$, and $0 \ll \beta < 1$. With this choice, RLS downweighs old samples, and can track time-varying signals.

(w3) *Finite window* with $\beta_{N,n} = 1$ if $N - n \leq M - 1$ and $\beta_{N,n} = 0$ otherwise. Here, only the most recent M samples are utilized to form $\hat{\mathbf{x}}_N^{\text{RLS}}$ while the rest are discarded.

The RLS estimator in (6) can be expressed recursively in terms of $\hat{\mathbf{x}}_{N-1}^{\text{RLS}}$. Supposing that $\{\mathbf{h}_n\}_{n=1}^{P}$ are linearly independent, setting $\beta_{N,n} = 1$, and initializing this recursion with the LS solution for $N - 1 = P$, that is $\hat{\mathbf{x}}_{N-1}^{\text{RLS}} =$

$\widehat{\mathbf{x}}_P^{\text{LS}}$, the RLS coincides with the LS for successive instants $N > P$, provided that $\mathbf{x}_o$ remains invariant [254, p. 740].

For $N < P$ or when $\{\mathbf{h}_n\}_{n=1}^P$ are linearly dependent, the RLS estimator can be regularized by augmenting the LS cost with a scaled ℓ_2-norm of $\mathbf{x}$ [254, p. 739]. Specifically, the regularized RLS is

$$\widehat{\mathbf{x}}_N^{\text{RLS}} := \arg \min_{\mathbf{x}\in\mathbb{R}^P} \left[\sum_{n=1}^{N} \beta_{N,n}(y_n - \mathbf{h}_n^T\mathbf{x})^2 + \gamma_N \|\mathbf{x}\|_2^2 \right], \quad N = 1, 2, \ldots \tag{7}$$

where $\gamma_N > 0$ is a pre-selected decreasing function of N that depends on the selected window and its effect vanishes for large N. Clearly, for $\gamma_N = 0$ the regularized RLS in (7) reduces to the ordinary one in (6), but both do *not* exploit the sparsity present in $\mathbf{x}_o$.

Sparse linear regression is a topic of intense research in the last decade and Lasso is one of the most widely applied sparsity-aware estimators [278, 46]. The Lasso estimator is given by

$$\widehat{\mathbf{x}}_N^{\text{Lasso}} := \arg \min_{\mathbf{x}\in\mathbb{R}^P} \left[\frac{1}{2} \|\mathbf{y}_N - \mathbf{H}_N\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right]. \tag{8}$$

Thanks to the scaled ℓ_1-norm, the cost encourages sparse solutions [278]: the larger the chosen λ is, the more components are shrunk to zero. Interestingly, the Lasso performs well in sparse problems also when $N < P$, and the convex ℓ_1-norm regularization which can afford efficient solvers when optimizing (8) given batch data, performs similarly to its non-convex ℓ_0-norm counterpart [44]. The question that arises is how the ℓ_1-norm regularization can be effectively utilized in adaptive signal processing.

Specifically, given $\{y_n, \mathbf{h}_n\}_{n=1}^N$, we wish to develop recursive schemes to estimate the sparse signal of interest with: i) minimal memory requirements; ii) tracking capability; and iii) limited complexity.

Adaptive Pseudo-Real Time Lasso

Motivated by (6), a time-weighted Lasso (TWL) approach emerges naturally to endow the batch Lasso in (8) with ability to handle sequential processing. Specifically, the proposed TWL estimator is

$$\widehat{\mathbf{x}}_N^{\text{TWL}} := \arg \min_{\mathbf{x}\in\mathbb{R}^P} J_N^{\text{TWL}}(\mathbf{x}), \quad N = 1, 2, \ldots \tag{9}$$

where $J_N^{\mathrm{TWL}}(\mathbf{x}) := \frac{1}{2}\sum_{n=1}^{N}\beta_{N,n}(y_n - \mathbf{h}_n^T\mathbf{x})^2 + \lambda_N\|\mathbf{x}\|_1$. In addition to windowing, note that λ_N is now allowed to vary with N.
Neglecting constant terms, the cost function in (9) can be re-written as

$$J_N^{\mathrm{TWL}}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{R}_N\mathbf{x} - \mathbf{x}^T\mathbf{r}_N + \lambda_N\|\mathbf{x}\|_1 \tag{10}$$

where

$$\mathbf{r}_N := \sum_{n=1}^{N}\beta_{N,n}y_n\mathbf{h}_n, \qquad \mathbf{R}_N := \sum_{n=1}^{N}\beta_{N,n}\mathbf{h}_n\mathbf{h}_n^T. \tag{11}$$

Due to data windowing, $\mathbf{r}_N$ and $\mathbf{R}_N$ can be updated recursively as [cf. (w1)-(w3)]

$$\text{(w1)}: \quad \mathbf{r}_N = \mathbf{r}_{N-1} + y_N\mathbf{h}_N, \quad \mathbf{R}_N = \mathbf{R}_{N-1} + \mathbf{h}_N\mathbf{h}_N^T \tag{12a}$$

$$\text{(w2)}: \quad \mathbf{r}_N = \beta\mathbf{r}_{N-1} + y_N\mathbf{h}_N, \quad \mathbf{R}_N = \beta\mathbf{R}_{N-1} + \mathbf{h}_N\mathbf{h}_N^T \tag{12b}$$

$$\text{(w3)}: \quad \mathbf{r}_N = \mathbf{r}_{N-1} + y_N\mathbf{h}_N - y_{N-M}\mathbf{h}_{N-M}, \tag{12c}$$

$$\mathbf{R}_N = \mathbf{R}_{N-1} + \mathbf{h}_N\mathbf{h}_N^T - \mathbf{h}_{N-M}\mathbf{h}_{N-M}^T. \tag{12d}$$

Relative to the batch Lasso in (8), any of the TWL updates in (12) offers memory savings. Clearly, choices (w2) and (w3) allow also the signal of interest to vary slowly with time. With respect to RLS in (7), the TWL estimator inherits the properties brought by the ℓ_1-norm, namely sparsity awareness and ability to deal with under-determined systems ($N < P$). Summarizing, the attractive features of TWL are:

i) Reduced memory requirements with respect to batch Lasso.

ii) Improved performance relative to RLS when $\mathbf{x}_o$ is sparse and time-invariant;

iii) Enhanced tracking capability when $\mathbf{x}_o$ is sparse and time-varying, relative to batch Lasso and RLS for windows of size less than the dimension P of $\mathbf{x}_o$.

Despite these attractive features, the main limitation of TWL is that a convex program has to be solved per time N. While the RLS cost is differentiable, and thus amenable to closed-form minimization, $J_N^{\mathrm{TWL}}(\mathbf{x})$ is not. However, initializing the convex program at time N with the solution $\widehat{\mathbf{x}}_{N-1}^{\mathrm{TWL}}$ at time $N-1$ provides a “warm start-up,” which speeds up convergence to the optimum $\hat{\mathbf{x}}_N^{\text{TWL}}$. For these reasons, TWL is a “pseudo-real time” algorithm. Low-complexity real-time algorithms will be developed in Section 3. But for now, it is worth checking TWL for consistency.

(In)Consistency of the TWL Estimator

Since the non-zero support of $\mathbf{x}_o$ is unknown, and sparse vector estimators are nonlinear functions of the data not expressible in closed form, performance analysis is distinct from and far more challenging than that of LS estimators. Consider for simplicity that $\mathbf{x}_o$ is time-invariant for which (w1) is prudent to adopt, and suppose that the regressors and noise satisfy these regularity (ergodicity) conditions:

(r1) $\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} \mathbf{h}_n\mathbf{h}_n^T = \mathbf{R}_\infty$ with probability (w.p.) 1, with $\mathbf{R}_\infty$ positive definite; and

(r2) $\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} v_n\mathbf{h}_n = \mathbf{r}_\infty^{vh}$ w.p. 1.

If the noise v_n and the regressors $\{\mathbf{h}_n\}$ are mixing, which is the case for most stationary processes with vanishing memory in practice, then (r1) and (r2) are readily met. Since v_n in (4) is zero mean and uncorrelated with $\mathbf{h}_n$, the cross-covariance in (r2) vanishes. With $\mathbf{r}_\infty^{vh} = \mathbf{0}$, it follows readily from (4) that $\mathbf{r}_\infty := \lim_{N\to\infty} N^{-1}\sum_{n=1}^{N} y_n\mathbf{h}_n = \mathbf{R}_\infty\mathbf{x}_o$ w.p. 1. Upon dividing both sides of (10) by N and taking limits, (r1) and (r2) then imply that $\lim_{N\to\infty}(1/N)J_N^{\text{TWL}}(\mathbf{x}) = (1/2)\mathbf{x}^T\mathbf{R}_\infty\mathbf{x} - \mathbf{x}^T\mathbf{r}_\infty := J_\infty(\mathbf{x})$ w.p. 1, if λ_N is chosen to grow slower than N. In this case, as $N \to \infty$ it holds that $\hat{\mathbf{x}}_N^{\text{TWL}} = \arg\min_{\mathbf{x}} J_N^{\text{TWL}}(\mathbf{x}) \to \arg\min_{\mathbf{x}} J_\infty(\mathbf{x}) := \mathbf{R}_\infty^{-1}\mathbf{r}_\infty = \mathbf{x}_o$ w.p. 1. This proves the following result.

Proposition 1. *For the model in* (4) *with* (r1), (r2)*, and* (w1) *in effect, the TWL estimator is strongly consistent, provided that* λ_N *is chosen to satisfy* $\lim_{N\to\infty}\frac{\lambda_N}{N} = 0$.

At this point it is instructive to recall that under the conditions of Proposition 1, the LS estimator $\hat{\mathbf{x}}_N^{\text{LS}} = \mathbf{R}_N^{-1}\mathbf{r}_N$ also converges w.p. 1 to $\mathbf{x}_o = \mathbf{R}_\infty^{-1}\mathbf{r}_\infty$, and is thus strongly consistent [167]. For this reason, to assess performance of TWL and differentiate it from that of LS it is pertinent to consider sufficiently large (but preferably finite) N for which the standard sparsity-agnostic LS is

The version of record is available at: http://dx.doi.org/10.1561/2000000107

unable to accurately estimate the zero entries of $\mathbf{x}_o$. It is thus of interest to check whether TWL can estimate jointly the nonzero support and the nonzero entries of $\mathbf{x}_o$ consistently for sufficiently large N. To this end, suppose that the first P_1 entries of $\mathbf{x}_o$ are non-zero; i.e., $\mathcal{S}_{\mathbf{x}_o} := \{1, \ldots, P_1\}$; and partition accordingly the $\mathbf{R}_\infty$ matrix as

$$\mathbf{R}_\infty = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{10} \\ \mathbf{R}_{01} & \mathbf{R}_{00} \end{pmatrix}.$$

Following the definitions in [103], *support consistency* amounts to having

$$\lim_{N\to\infty} \text{Prob}\left[\mathcal{S}_{\widehat{\mathbf{x}}_N} = \mathcal{S}_{\mathbf{x}_o}\right] = 1 \tag{13}$$

and $\sqrt{N}$-*estimation consistency* requires convergence in distribution ($\to_d$), that is

$$\sqrt{N}(\widehat{\mathbf{x}}_N^{P_1} - \mathbf{x}_o^{P_1}) \to_d \mathcal{N}(\mathbf{0}_{P_1}, \sigma^2 \mathbf{R}_{11}^{-1}) \tag{14}$$

where $\mathbf{x}^{P_1}$ denotes the $P_1 \times 1$ vector obtained by extracting the first P_1 components of $\mathbf{x}$. Properties (13) and (14) are referred to as *oracle properties* because a sparsity-aware estimator possessing these properties is asymptotically as good as if the support $\mathcal{S}_{\mathbf{x}_o}$ was known in advance [103].
Under (w1), the TWL corresponds to a sequential version of the batch Lasso estimator. Hence, asymptotic properties of the latter derived in [331, 167] carry over to the TWL estimator introduced here.

Lemma 1. (See also [331, Prop. 1]). *For the model in* (4) *with* (r1), (r2), *and* (w1) *in effect, if* $\lim_{N\to\infty} \frac{\lambda_N}{\sqrt{N}} = \lambda_0 \geq 0$, *then* $\lim_{N\to\infty} \text{Prob}\left[\mathcal{S}_{\widehat{\mathbf{x}}_N^{\text{TWL}}} = \mathcal{S}_{\mathbf{x}_o}\right] = c(\lambda_0) < 1$ *with* $c(\lambda_0)$ *denoting an increasing function of* λ_0.

In words, Lemma 1 asserts that if λ_N grows as $\sqrt{N}$, support consistency cannot be achieved. Since $c(\lambda_0)$ increases with λ_0, the hope for the TWL to satisfy the oracle properties is left for cases wherein λ_N grows faster than $\sqrt{N}$. Unfortunately, the next result discourages this.

Lemma 2. (See also [331, Lemma 3]). *For the model in* (4) *with* (r1), (r2), *and* (w1) *in effect, if* $\lim_{N\to\infty} \frac{\lambda_N}{\sqrt{N}} = \infty$ *and* $\lim_{N\to\infty} \frac{\lambda_N}{N} = 0$, *then* $\lim_{N\to\infty} \frac{N}{\lambda_N}(\widehat{\mathbf{x}}_N^{\text{TWL}} - \mathbf{x}_o) = C$, *where* C *is a non-random constant.*

The version of record is available at: http://dx.doi.org/10.1561/2000000107

Lemma 2 states that if λ_N grows faster than $\sqrt{N}$ but slower than N, the rate of convergence is $\frac{N}{\lambda_N}$, that is slower than $\sqrt{N}$; hence, $\sqrt{N}(\hat{\mathbf{x}}_N^{\text{TWL}} - \mathbf{x}_o)$ diverges. Combining Lemmas 1 and 2, the following negative result holds for batch Lasso and thus for TWL.

Proposition 2. *For the model in* (4) *with* (r1), (r2), *and* (w1) *in effect, the TWL estimator can not achieve the oracle properties for any choice of* λ_N.

Before exploring alternatives to TWL that satisfy the oracle properties, one remark is in order.

Remark 1. If instead of the convex ℓ_1-norm the LS cost is regularized with suitably chosen non-convex functions of $\mathbf{x}$, it is possible to construct sparsity-aware estimators that asymptotically possess the oracle properties [103]. Of course, the price paid is inefficient optimization due to non-convexity. These considerations motivate searching for convex regularizing terms which result in pseudo real-time Lasso estimators satisfying the oracle properties. Such a class is developed next using the weighted ℓ_1-norm regularization, introduced in [331, 332] for the batch Lasso.

Time- and Norm-Weighted Lasso

Let $u(\cdot)$ denote the step function, $\{\mu_N\}$ a positive sequence dependent on the sample size, and $a > 1$ a constant tuning parameter. Based on these, define the weight function $w_{\mu_N}(\cdot) : \mathbb{R}^+ \to [0, 1]$ as

$$w_{\mu_N}(x) := \frac{[a\mu_N - x]_+}{(a-1)\mu_N}\, u\left(x - \mu_N\right) \,+\, u\left(\mu_N - x\right) \tag{15}$$

where $[\alpha]_+ := \max(\alpha, 0)$. Using w_{μ_N}, the proposed time- *and* norm-weighted Lasso (TNWL) estimator weighs the ℓ_1-norm with coefficients depending on the entries of $\mathbf{x}_N^{\text{RLS}}$; that is,

$$\begin{aligned}\hat{\mathbf{x}}_N^{\text{TNWL}} := \arg\min\nolimits_{\mathbf{x}\in\mathbb{R}^P} \Bigg[\frac{1}{2}\sum_{n=1}^{N} \beta_{N,n}(y_n - \mathbf{h}_n^T\mathbf{x})^2 \\ +\lambda_N \sum_{p=1}^{P} w_{\mu_N}\left(|\hat{x}_N^{\text{RLS}}(p)|\right)|x(p)|\Bigg],\ N = 1, 2, \ldots\end{aligned} \tag{16}$$

Fig. 2 shows the weight function $w_{\mu_N}(x)$ for $\mu_N = 0.2$ and $a = 3.7$. Notice that while λ_N in TWL weighs identically all summands $|x(p)|$ in the ℓ_1-norm,

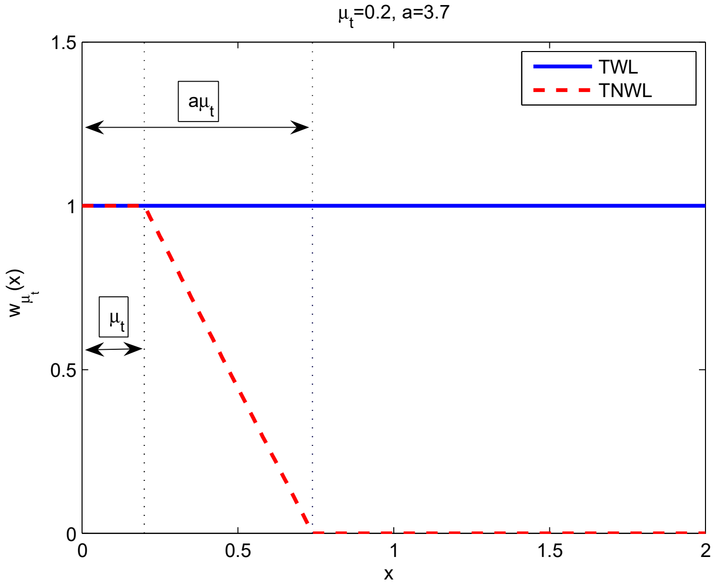


Figure 2: Weight functions for TWL and TNWL estimators.

the TNWL estimator places higher weight to small entries, and lower weight to entries with large amplitudes. In fact, RLS estimates of size less than μ_N are penalized as in TWL, while estimates between μ_N and $a\mu_N$ are penalized in a linearly decreasing manner. Finally, RLS estimates larger than $a\mu_N$ are not penalized at all (cf. Fig. 2).

It is worth recalling at this point that albeit sparsity-agnostic, the RLS estimator is $\sqrt{N}$-estimation consistent [167], that is

$$\sqrt{N}(\hat{\mathbf{x}}_N^{\mathrm{RLS}} - \mathbf{x}_o) \rightarrow_d \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{R}_\infty^{-1}). \tag{17}$$

Based on (17), it is possible to establish the following result.

Proposition 3. (See also [332, Theorem 4]). *For the model in* (4)*, with* (r1)*,* (r2)*, and* (w1) *in effect, if* $\lim_{N\to\infty} \frac{\lambda_N}{\sqrt{N}} = \infty$, $\lim_{N\to\infty} \frac{\lambda_N}{N} = 0$ *and* $\mu_N = \frac{\lambda_N}{N}$*, the TNWL estimator satisfies the oracle properties* (13) *and* (14)*.*

Weighted ℓ_1-norm regularization was introduced in [48], [331], and [332] using different weight functions to effect sparsity and satisfy the oracle properties of the *batch* weighted Lasso estimator. The weight function in (15) corresponds to the local linear approximation of the smoothly clipped absolute deviation regularizer introduced by [332]. The difference here is its coupling with RLS to ensure consistency of the devised *adaptive* TNWL estimator.

Next, the implications of Propositions 2 and 3 are demonstrated through simulated tests.

Numerical Examples

Gaussian observations are generated according to (4) with a time-invariant $\mathbf{x}_o$, $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, $\mathbf{h}_n \sim \mathcal{N}\left(\mathbf{0}_P, \frac{1}{P}\mathbf{I}_P\right)$ and infinite windowing as in (w1). The penalty scale is set to $\lambda_N = \sqrt{2\sigma^2 N \log P}$ for the TWL (see also [331]), and $\lambda_N = \sqrt{2\sigma^2 N^{\frac{4}{3}} \log P}$ for TNWL with $\mu_N = \frac{\lambda_N}{N}$ and $a = 3.7$. The first three entries of $\mathbf{x}_o$ are chosen equal to unity, and all other entries are set to zero. Fig. 3 depicts the mean-square error (MSE), $\mathrm{E}[\|\widehat{\mathbf{x}}_N - \mathbf{x}_o\|^2]$, across time for the TWL, TNWL, and RLS along with what is termed genie-aided (GA) RLS, which knows in advance the support and performs standard RLS to estimate the non-zero components. The convex optimization problem per time N is solved using the SeDuMi package [271] interfaced with Yalmip [193]. Observe that while TWL outperforms RLS, it is outperformed by TNWL, whose performance approaches that of the GA-RLS benchmark. Indeed, the TNWL does achieve the oracle properties in the considered simulation setting.

Next, Gaussian observations are generated according to (4) with a time-varying $\mathbf{x}_o$ (henceforth denoted as $\mathbf{x}_n$), and parameters $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, and $\mathbf{h}_n \sim \mathcal{N}\left(\mathbf{0}_P, \frac{1}{P}\mathbf{I}_P\right)$. A Gauss-Markov model is assumed for $\mathbf{x}_n$; that is, $x_n(p) = \alpha x_{n-1}(p) + w_n(p)$ with $x_0(p) \sim \mathcal{N}(0, 1)$, $\alpha = 0.99$, and $w_n(p) \sim \mathcal{N}\left(0, 1 - \alpha^2\right)$ for $p = 1, 2, 3$. Without loss of generality, (w2) is adopted with $\beta = 0.9$, and $\lambda_N = \sqrt{2\sigma^2 \log P}\sqrt{\sum_{n=1}^{N} \beta^{2(N-n)}}$ for both TWL and TNWL and $\mu_N = \frac{\lambda_N}{\sum_{n=1}^{N} \beta^{N-n}}$. Clearly, in a time-varying setting these estimators are not expected to achieve the consistency properties established when $\mathbf{x}_o$ remains time-invariant. Fig. 4 depicts the squared error (SE) for a realization of the RLS, GA-RLS, TWL and TNWL. In the considered setting, TWL and TNWL perform similarly and both outperform RLS while approaching the performance of the GA-RLS benchmark.

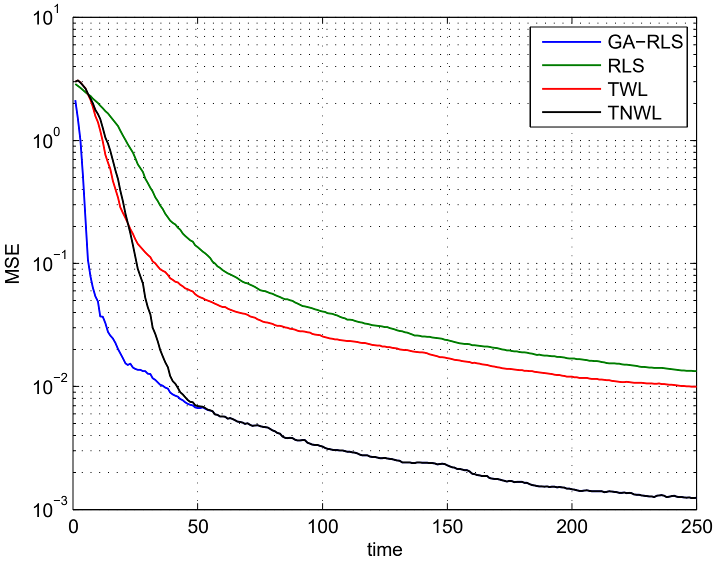


Figure 3: MSE comparisons of pseudo real-time estimators (time-invariant $\mathbf{x}_o$).

Next, Gaussian observations are generated according to (4) with a time-varying $\mathbf{x}_n$, and parameters $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-2}$, and $\mathbf{h}_n \sim \mathcal{N}\left(\mathbf{0}_P, \frac{1}{P}\mathbf{I}_P\right)$. A Gauss-Markov model is assumed for $\mathbf{x}_n$ with $\alpha = 0.995$, and (w3) windowing is adopted. For brevity, only the regularized RLS in (7) with constant γ_N is shown along with the TWL estimators. In Fig. 5, two window sizes of length $M = 15$ and $M = 30$ are simulated. Interestingly, while RLS with $M = 30$ outperforms RLS with $M = 15$, TWL with $M = 15$ outperforms TWL with $M = 30$ and achieves the overall best performance. In fact, TWL performs well even for small window sizes, $M < P$, when the signal of interest is sparse. Thus, TWL exhibits better tracking capability than RLS which requires longer window size, and thus can track signals with slower variations.

Adaptive Real-Time Lasso

As mentioned earlier, TWL and TNWL estimators are not suitable for real-time implementation. In this section, online algorithms are developed and analyzed.

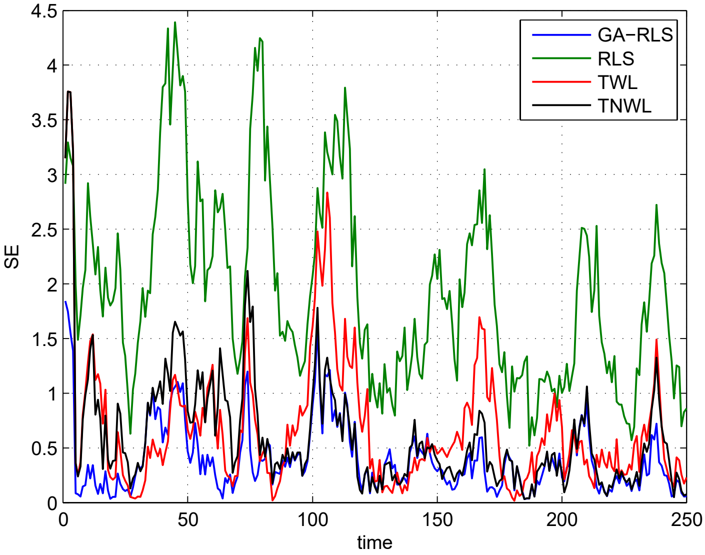


Figure 4: Squared error comparisons of pseudo real-time estimators (time-varying $\mathbf{x}_n$ with exponentially decreasing window).

The vector iterates developed in the next subsection provide online solvers of (9) and (16), admit a closed-form solution per iteration, and are proved convergent to $\mathbf{x}_o$ when the unknown vector is time-invariant. For notational brevity, the algorithms are developed for the TWL estimator but carry over to TNWL as well.

Online Coordinate Descent

One approach to finding the solution $\hat{\mathbf{x}}_N^{\mathrm{TWL}}$ in (9) is to run a cyclic coordinate descent (CCD) algorithm, which in its simplest form entails cyclic iterative minimization of $J_N^{\mathrm{TWL}}(\mathbf{x})$ in (10) with respect to one coordinate per iteration cycle. Let $\mathbf{x}_N^{(i-1)}$ denote the solution at time N and iteration $i-1$. The pth variable at the ith iteration is updated as

$$x_N^{(i)}(p) := \arg\min_x J_N^{\mathrm{TWL}}\Big(\begin{array}{l} x_N^{(i)}(1),\ldots,x_N^{(i)}(p-1),x, \\ x_N^{(i-1)}(p+1),\ldots,x_N^{(i-1)}(P)\Big) \end{array} \tag{18}$$

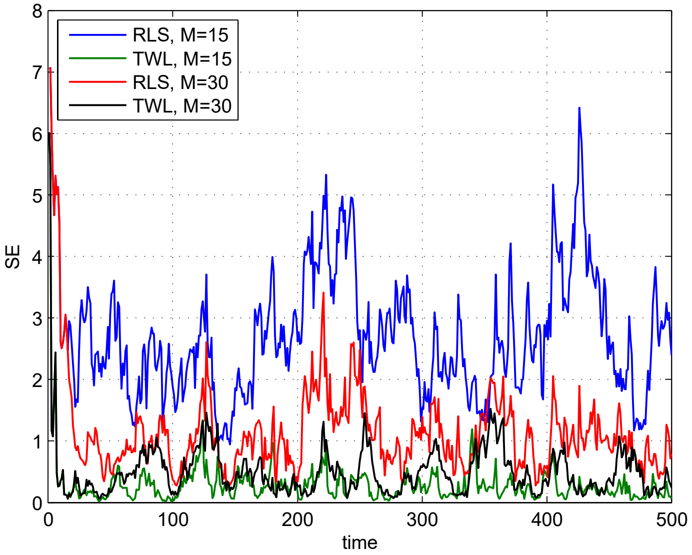


Figure 5: Squared error comparisons of pseudo real-time estimators (time-varying $\mathbf{x}_n$ with finite window).

for $p = 1, \ldots, P$. During the ith cycle each coordinate (here the pth) is optimized, while the pre-cursor coordinates (those with $p' < p$) are kept fixed to their values at the ith cycle, and the post-cursor coordinates (those with $p' > p$) are kept fixed to their values at the $(i-1)$st cycle.

Albeit convex, the cost $J_N^{\mathrm{TWL}}(\mathbf{x})$ is non-differentiable. Nonetheless, convergence of the CCD algorithm for Lasso-type problems follows readily using the results of [280]. In addition to affording effective initialization (with the all-zero vector), another attractive feature of CCD Lasso solvers is that each coordinate-wise minimizer is available in closed form. Recent comparative studies show that CCD exhibits computational complexity similar (if not lower) than state-of-the-art batch Lasso solvers and is numerically stable [115, 308].

The online coordinate descent (OCD) algorithm introduced next can be viewed as an adaptive counterpart of CCD Lasso, where a new datum is incorporated at each iteration; that is, the iteration index (i) in CCD is replaced in OCD by the time index N. The challenge arises because the cost function changes with

N. The crux of OCD is to update only one variable per datum in the spirit of e.g., the partial least mean-squares (PLMS) algorithm [198]. Notwithstanding, PLMS is a sparsity-agnostic first-order algorithm, whereas OCD is sparsity-cognizant, it capitalizes on second-order statistics similar to RLS, and it is also provably convergent.

For notational convenience, express the time index as $N = kP + p$, where $p \in \{1, \ldots, P\}$ corresponds to the only entry of $\mathbf{x}$ to be updated at time N, and $k = \lceil \frac{N}{P} \rceil - 1$ indexes the number of cycles; that is, how many times the pth coordinate is updated. Let $\hat{\mathbf{x}}_{N-1}^{\text{OCD}}$ denote the solution of the OCD algorithm at time $N-1$ and $\hat{x}_N^{\text{OCD}}(q) = \hat{x}_{N-1}^{\text{OCD}}(q)$ for $q \neq p$, which amounts to setting all but the pth coordinate at time N equal to those at time $N-1$, and selecting the pth one by minimizing $J_N^{\text{TWL}}(\mathbf{x})$; that is,

$$\begin{aligned}\hat{x}_N^{\text{OCD}}(p) := \arg\min\nolimits_x J_N^{\text{TWL}}(\; & \hat{x}_{N-1}^{\text{OCD}}(1), \ldots, \hat{x}_{N-1}^{\text{OCD}}(p-1), x, \\ & \hat{x}_{N-1}^{\text{OCD}}(p+1), \ldots, \hat{x}_{N-1}^{\text{OCD}}(P)).\end{aligned} \tag{19}$$

In the cyclic update (19), the pre-cursor coordinates $\{\hat{x}_{N-1}^{\text{OCD}}(q), \; q < p\}$ have been updated $k+1$ times, while the post-cursor entries $\{\hat{x}_{N-1}^{\text{OCD}}(q), \; q > p\}$ have been updated k times.

After isolating from $J_N^{\text{TWL}}(\mathbf{x})$ only terms which depend on the pth coordinate that is currently optimized, recursion (19) can be rewritten as (cf. (9))

$$\hat{x}_N^{\text{OCD}}(p) \quad = \quad \arg\min_x \left[\frac{1}{2} R_N(p,p)x^2 - r_{N,p}x + \lambda_N |x|\right] \tag{20}$$

$$r_{N,p} \quad := \quad r_N(p) - \sum_{q \neq p} R_N(p,q)\hat{x}_{N-1}^{\text{OCD}}(q). \tag{21}$$

Being a scalar optimization problem, it is well known that the minimization problem in (20) accepts a closed-form solution, namely [115]

$$\hat{x}_N^{\text{OCD}}(p) = \frac{\text{sgn}(r_{N,p})}{R_N(p,p)} \left[|r_{N,p}| - \lambda_N\right]_+. \tag{22}$$

Equation (22) amounts to a soft-thresholding operation that sets to zero inactive entries, thus facilitating convergence to sparse iterates. The OCD-TWL scheme is tabulated as Algorithm 1.

Convergence of OCD-TWL is established in Appendix A, and the main result can be summarized as follows.

Algorithm 1 OCD-TWL

```
Initialize with $\hat{\mathbf{x}}_0^{\text{OCD}} = \mathbf{0},\ p = 1, \ldots, P$
for $k = 0, 1, \ldots$ do
    for $p = 1, \ldots, P$ do
        S1. Acquire datum $y_N$, and regressor $\mathbf{h}_N$, $N = kP + p$.
        S2. Obtain $\mathbf{r}_N$ and $\mathbf{R}_N$ as in (11).
        S3. Set $\hat{x}_N^{\text{OCD}}(q) = \hat{x}_{N-1}^{\text{OCD}}(q)$ for all $q \neq p$.
        S4. Compute $r_{N,p}$ via (21).
        S5. Update $\hat{x}_N^{\text{OCD}}(p)$ as in (22).
    end for
end for
```

Proposition 4. *For the model in* (4) *with* (r1), (r2), *and* (w1) *in effect, if* $\lim_{N\to\infty} \frac{\lambda_N}{N} = 0$*, it holds w.p. 1 that* $\lim_{N\to\infty} \hat{\mathbf{x}}_N^{\text{OCD}} = \mathbf{x}_o$.

In words, Proposition 4 asserts that the OCD-TWL estimator is strongly consistent.

Online Selective Coordinate Descent

The OCD-TWL solver has low complexity but may exhibit slow convergence since each variable is updated every P observations. But since $P_1 \ll P_0$ due to sparsity, most of the time OCD-TWL re-sets to zero inactive entries of $\mathbf{x}_o$. On the other hand, updating zero variables cannot be skipped a priori since new nonzero entries may arise in time-varying scenarios. To address this dilemma, it is prudent to select which coordinate to update. A related selective approach has been pursued for batch Lasso in [308], and is extended here to the following OCD solver.

Let $d_{\mathbf{e}_p} J_N^{\text{TWL}}(\hat{\mathbf{x}}_{N-1}^{\text{OSCD}})$ and $d_{-\mathbf{e}_p} J_N^{\text{TWL}}(\hat{\mathbf{x}}_{N-1}^{\text{OSCD}})$ denote the forward and backward directional derivatives w.r.t. $x(p)$ evaluated at $\hat{\mathbf{x}}_{N-1}^{\text{OSCD}}$, which denotes the online selective coordinate descent (OSCD) estimate at time $N-1$. Define also the vectors $\mathbf{d}^+, \mathbf{d}^- \in \mathbb{R}^P$ whose pth entries are $d_{\mathbf{e}_p} J_N^{\text{TWL}}(\hat{\mathbf{x}}_{N-1}^{\text{OSCD}})$ and $d_{-\mathbf{e}_p} J_N^{\text{TWL}}(\hat{\mathbf{x}}_{N-1}^{\text{OSCD}})$, respectively. It is not difficult to verify that (see also

Algorithm 2 OSCD-TWL

```
  Initialize $\widehat{\mathbf{x}}_0^{\text{OSCD}} = \mathbf{0},\ p = 1, \ldots, P$.
  for $N = 1, 2, \ldots$ do
      S1. Acquire datum $y_N$, and regressor $\mathbf{h}_N$.
      S2. Evaluate $\mathbf{d}^+$ and $\mathbf{d}_-$ as in (23) and (24).
      S3. Select $p^* = \arg\min_p \{d^+(p), d^-(p)\}_{p=1}^P$.
      S4. Update $\widehat{x}_N^{\text{OSCD}}(q) = \widehat{x}_{N-1}^{\text{OSCD}}(q)$ for all $q \neq p^*$.
      S5. Compute $r_{N,p^*}$ via (21).
      S6. Update $\widehat{x}_N^{\text{OSCD}}(p^*)$ as in (22).
  end for
```

[308])

$$\mathbf{d}^+ = \mathbf{R}_N \widehat{\mathbf{x}}_{N-1}^{\text{OSCD}} - \mathbf{r}_N + \lambda_N \mathbf{s}^+ \tag{23}$$

$$\mathbf{d}^- = \mathbf{r}_N - \mathbf{R}_N \widehat{\mathbf{x}}_{N-1}^{\text{OSCD}} + \lambda_N \mathbf{s}^- \tag{24}$$

with $\mathbf{s}^+, \mathbf{s}^- \in \mathbb{R}^P$, $s^+(p) = 1$ if $\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}} \geq 0$ and $s^+(p) = -1$ otherwise; while $s^-(p) = 1$ if $\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}} \leq 0$, and $s^+(p) = -1$ otherwise. After evaluating (23) and (24), the coordinate with the most negative directional derivative, either forward or backward, is updated. The OSCD-TWL scheme is summarized as Algorithm 2.

Remark 2. A subgradient-based LMS-like is developed in [8] for sparsity-aware online estimation. However, subgradient methods are first-order algorithms that posses slow convergence. For this reason, the OCD and OSCD alternatives developed here should be preferred.

Complexity Issues

Recall that the RLS algorithm requires $\mathcal{O}(P^2)$ algebraic operations per datum. On the other hand, the OCD-TWL Algorithm 1 requires $r_{N,p}$, whose computational burden is $\mathcal{O}(P)$ given $\mathbf{R}_N$ and $\mathbf{r}_N$. As far as OSCD is concerned, the selection step requires evaluation of $\mathbf{d}^+$ and $\mathbf{d}^-$ whose computation entails $\mathcal{O}(PP_{1,N})$ algebraic operations, where $P_{1,N}$ denotes the number of non-zero entries of $\widehat{\mathbf{x}}_N^{\text{OSCD}}$. However, the overall computational burden of the OCD-TWL algorithm is dominated by the update of $\mathbf{R}_N$, which requires $\mathcal{O}(P^2)$ algebraic operations. In this general case, the OCD (OSCD) can be imple-

Algorithm 3 OCCD-TWL

```
Initialize $\widehat{\mathbf{x}}_0^{\text{OCCD}} = \mathbf{0},\ p = 1, \ldots, P$
for $N = 1, 2, \ldots$ do
    S1. Acquire datum $y_N$, and regressor $\mathbf{h}_N$.
    S2. Obtain $\mathbf{r}_N$ and $\mathbf{R}_N$ as in (11).
    for $p = 1, \ldots, P$ do
        S3. Evaluate $r_{N,p} = r_N(p) - \sum_{q \neq p} R_N(p,q)\widehat{x}_{N-1}^{\text{OCCD}}(q)$
        S4. Update $\widehat{x}_N^{\text{OCCD}}(p)$ via (22).
    end for
end for
```

mented cyclically to update each coordinate per datum without affecting the overall complexity in order to speed up convergence. We summarize the online cyclic coordinate descent (OCCD) TWL as in Algorithm 3.

An important simplification which appears in problems such as system identification and beamforming is that regressors are sliding with time; that is $\mathbf{h}_n := [h(n), h(n-1), \ldots, h(n-P+1)]^T$ with $\mathbf{h}_0 = \mathbf{0}_P$. In this case, $\mathbf{R}_N$ updates and the RLS estimates incur complexity $\mathcal{O}(P)$ [254, p. 816], [295]. Likewise, OCD-TWL and OSCD-TWL in Algorithms 1 and 2 can be also implemented with complexity $\mathcal{O}(P)$.

Same conclusions can be drawn for online implementations of the TNWL through OCD or OSCD. In a nutshell, the devised online algorithms entail complexity analogous to RLS.

Simulated Tests

The online algorithms developed in Section 3 are simulated here and compared with the TWL and TNWL algorithms of Section 3 and also with the RLS in both time-invariant as well as time-varying scenarios.

Gaussian observations are generated according to (4) with a time-invariant $\mathbf{x}_o$, $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, $\mathbf{h}_n \sim \mathcal{N}\left(\mathbf{0}_P, \frac{1}{P}\mathbf{I}_P\right)$, and windowing as in (w1). The first three entries of $\mathbf{x}_o$ are chosen equal to unity, and all other entries are set to zero. Fig. 6 depicts the MSE of the OCD-TWL, OCCD-TWL, OSCD-TWL, and TWL versus time. The scale is set to $\lambda_N = \sqrt{2\sigma^2 N \log P}$. As expected, the OCD-TWL converges to the TWL

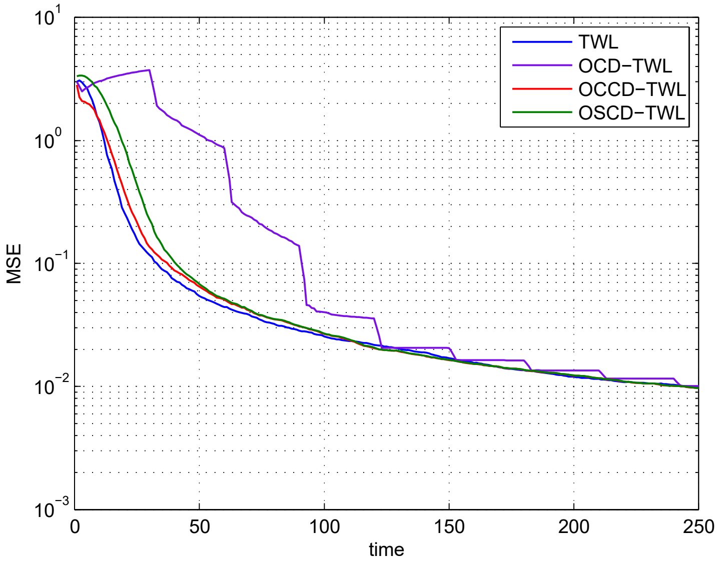


Figure 6: MSE comparisons of online estimators (time-invariant $\mathbf{x}_o$).

which requires the solution of a convex program per time N. Similar results hold for the OCCD-TWL and OSCD-TWL algorithms that also provide a means of enhancing the convergence speed.

Fig. 7 depicts the MSE of the OCD-TNWL, OCCD-TNWL, OSCD-TNWL, and TNWL versus time. The scale is set to $\lambda_N = \sqrt{2\sigma^2 N^{\frac{4}{3}} \log P}$ with $\mu_N = \frac{\lambda_N}{N}$ and $a = 3.7$ Also in this case the online algorithms converge to their pseudo real-time counterparts.

Next, Gaussian observations are generated according to (4) with a time-varying $\mathbf{x}_n$, and parameters $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, and $\mathbf{h}_n \sim \mathcal{N}\left(\mathbf{0}_P, \frac{1}{P}\mathbf{I}_P\right)$. A Gauss-Markov model is assumed for $\mathbf{x}_n$ with entries generated according to $x_n(p) = \alpha x_{n-1}(p) + w_n(p)$ with $x_0(p) \sim \mathcal{N}(0, 1)$, $\alpha = 0.99$, and $w_n(p) \sim \mathcal{N}(0, 1 - \alpha^2)$ for $p = 1, 2, 3$; (w2) is adopted with $\beta = 0.9$, and scale $\lambda_N = \sqrt{2\sigma^2 \log P}\sqrt{\sum_{n=1}^{N} \beta^{2(N-n)}}$. Fig. 8 shows a realization of the squared error (SE) for the OCD-TWL, OCCD-TWL, OSCD-TWL, TWL, and RLS. The OCD-TWL exhibits performance similar to that of the RLS. Indeed, updating one coordinate per observation in

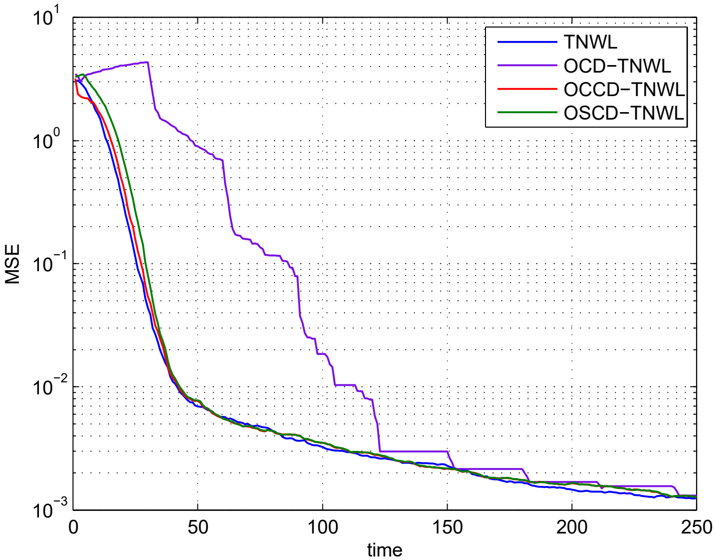


Figure 7: MSE comparisons of online weighted-norm estimators (time-invariant $\mathbf{x}_o$).

time-varying settings weakens tracking capabilities [198]. However, the performance OCCD-TWL and OSCD-TWL approaches that of TWL, and both outperform the RLS algorithm.

Successively, simulated tests are performed to assess performance when the support of $\mathbf{x}_n$ changes with time. The setting in this example is identical to that of Fig. 8, except that here the support of the sparse $\mathbf{x}_n$ also undergoes step changes. Specifically, at $N = 125$ the third entry of $\mathbf{x}_n$ starts decreasing, and after $N = 150$ the same entry is set to zero. In addition, at $N = 125$ the fourth entry becomes nonzero. Fig. 9 and 10 depict, respectively, the true variations of $x_n(3)$ and $x_n(4)$ across time, along with their estimates obtained using the RLS, the OCCD-TWL, and the OCCD-TNWL with $\mu_N = \frac{\lambda_N}{\sum_{n=1}^{N} \beta^{N-n}}$ and $a = 3.7$. Observe that the developed sparsity-aware algorithms can set to zero inactive entries while RLS estimates are not sparse and yield a nonzero value even if the true entry is zero. Moreover, after a few instants from the changing support points, the developed algorithms are able to track entries that become nonzero, and are further able to set to zero entries that disappear.

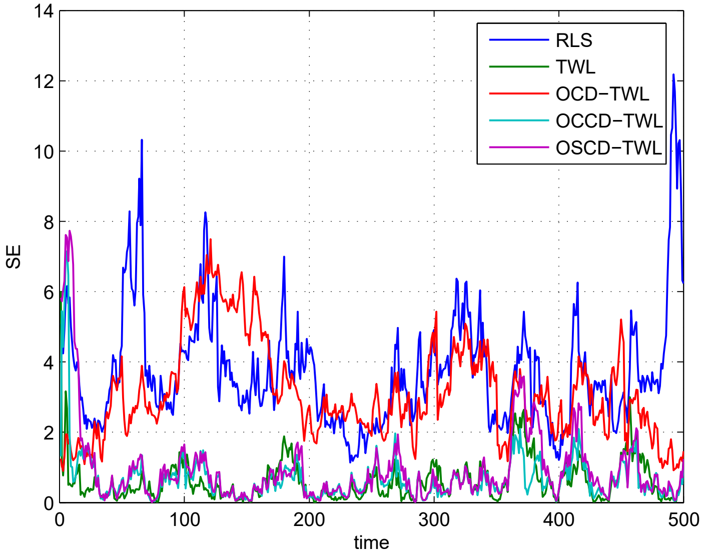


Figure 8: Squared error comparisons of online estimators (time-varying $\mathbf{x}_n$ with exponentially decreasing window).

Finally, the developed algorithms are tested for identifying the sparse, finite impulse response of a discrete-time, linear system using input and noisy output data satisfying the input-output relationship

$$y_n = \sum_{p=0}^{P-1} h_p x_{n-p} + v_n = \mathbf{x}_n^T \mathbf{h}_o + v_n, \quad n = 1, \ldots, N \tag{25}$$

where $\mathbf{h}_o := [h_0, \ldots, h_{P-1}]^T$ collects the unknown impulse response coefficients, $\mathbf{x}_n := [x_n, \ldots, x_{n-P+1}]^T$ denotes the given input data (the regressor vector in (4)), and y_n the output at time n. As the system order maybe unknown, a large known upper bound P is selected. Since many entries of $\mathbf{h}_o$ maybe zero or negligible, the impulse response is sparse. In addition, nonzero entries may exhibit slow time variations, which gives rise to a time-varying impulse response $\mathbf{h}_n$. To assess performance of the introduced algorithms, a system with $P = 128$ and $P_1 = 6$ nonzero entries at unknown locations is simulated. The input sequence is assumed zero-mean, white, Gaussian, with unit variance, and $v_n \sim \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 10^{-2}$. RLS, OSCD-TWL, and

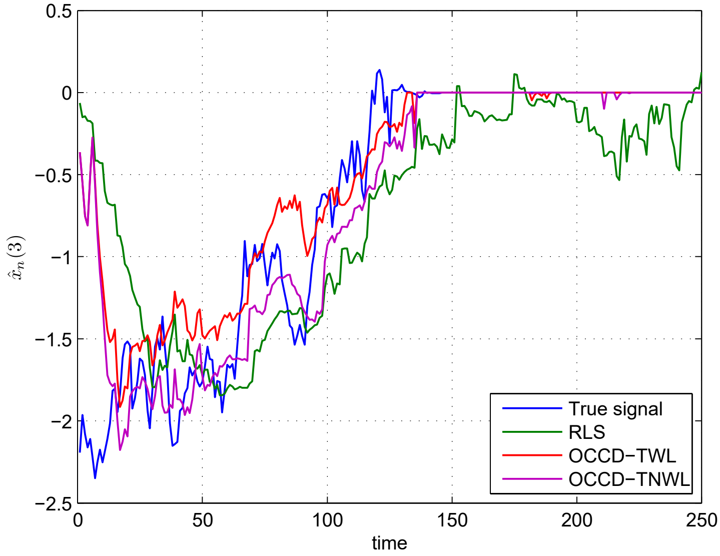


Figure 9: Trajectory of a varying entry of the true signal vector and its estimates across time (tracking of a disappearing entry).

OSCD-TNWL are tested along with the GA-RLS for an exponential window with $\beta = 0.95$. Since the regressors here are shift-invariant, all algorithms incur computational burden that scales linearly with P. The impulse response is generated according to a first-order Gauss-Markov process with $\alpha = 0.999$. The tuning parameters of the OSCD-TWL and OSCD-TNWL have been chosen as in Fig. 9. Fig. 11 depicts the MSE (averaged over 100 realizations) across time. It is clear that both OSCD-TWL and OSCD-TNWL outperform the RLS. In particular, the gain of the OSCD-TNWL is more than one order of magnitude.

Conclusions

In this section, recursive algorithms were developed for estimation of (possibly time-varying) sparse signals based on observations that obey a linear regression model, and become available sequentially in time. The presented TWL and TNWL algorithms can be viewed as ℓ_1-norm regularized versions of the

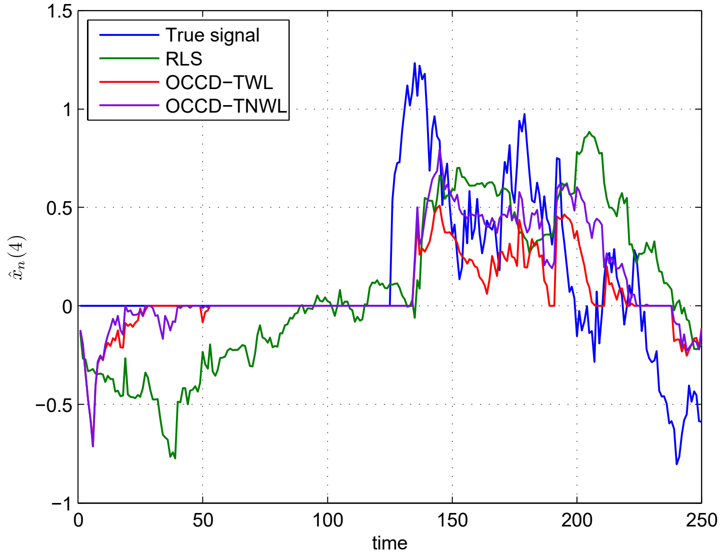


Figure 10: Trajectory of a varying entry of the true signal vector and its estimates across time (tracking of an emerging entry).

RLS. Simulations illustrated that TWL outperforms the sparsity-agnostic RLS scheme when estimating time-invariant and slowly-varying sparse signals. Moreover, the presented algorithms exhibit enhanced tracking capability with respect to RLS especially for short observation windows. Performance analysis revealed that TWL estimates cannot simultaneously recover the signal support and maintain convergence of RLS. This prompted the development of TNWL, which for proper selection of design parameters can achieve oracle consistency properties for time-invariant sparse signals. However, TWL and TNWL require solving a convex problem per time step, and may be less desirable for real-time applications. To overcome this limitation, low-complexity sparsity-aware online schemes were also developed. The crux of these schemes is a novel optimization algorithm that implements the basic coordinate descent iteration online. Albeit simple, the resulting OCD-TWL (OCD-TNWL) algorithm was proved convergent when the sparse signal is time invariant. At complexity comparable to OCD-TWL (OCD-TNWL) but with improved convergence

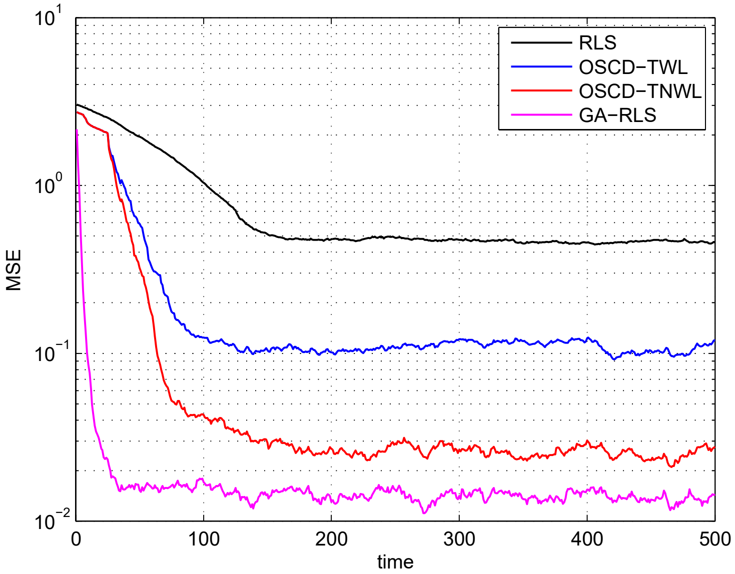


Figure 11: MSE comparison of adaptive algorithms for estimating a sparse, linear, time-varying impulse response.

speed, online selective variants choose the best coordinate to optimize and exhibit performance similar to the pseudo real-time TWL (TNWL).

4

Sparsity-Cognizant Total Least-Squares for Perturbed Compressive Sampling

In line with Section 3, this section mainly focuses on devising sophisticated signal reconstruction techniques for CS, while also applying the methods to wireless applications to assess their performance. The considered framework rests on the observation that solving linear regression problems based on the total least-squares (TLS) criterion has well-documented merits in various applications, where perturbations appear both in the data vector as well as in the regression matrix. However, conventional TLS approaches do not account for sparsity possibly present in the unknown vector of regression coefficients. On the other hand, sparsity is the key attribute exploited by modern CS and VS approaches to linear regression, which include noise in the data, but do not account for perturbations in the regression matrix. This section fills this gap by formulating and solving (regularized) TLS optimization problems under sparsity constraints. Near-optimum and reduced-complexity suboptimum sparse (S-) TLS algorithms are developed to address the perturbed CS (and the related dictionary learning) challenge, when there is a mismatch between the true and adopted bases over which the unknown vector is sparse. The developed S-TLS schemes also allow for perturbations in the regression matrix of the Lasso, and endow TLS approaches with ability to cope with sparse, under-determined "errors-in-variables" models. Interesting generalizations can further exploit prior knowledge on the perturbations to obtain novel weighted and structured S-TLS solvers. Analysis and simulations demonstrate the practical impact of S-TLS in calibrating the mismatch effects of contemporary grid-based approaches to cognitive radio sensing, and robust direction-of-arrival estimation using antenna arrays.

Related Works

Traditional CS, basis pursuit (BP), and Lasso-based approaches do not account for perturbations present in the matrix of equations, which in the BP (respectively Lasso) parlance is referred to as the representation basis or dictionary (correspondingly regression) matrix. Such perturbations appear when there is a *mismatch* between the adopted basis matrix and the actual but *unknown* one – a performance-critical issue in, e.g., sparsity-exploiting approaches to localization, time delay, and Doppler estimation in communications, radar, and sonar applications [117], [201], [228], [23], [9]. Performance analysis of CS and BP approaches for the partially-perturbed linear model with perturbations only in the basis matrix, as well as for the fully-perturbed one with perturbations present also in the measurements, was pursued in [148], [64], and [55]. However, devising a systematic approach to reconstruct sparse vectors under either type of perturbed models was left open.

Interestingly, for *non-sparse* over-determined linear systems, such an approach is available within the framework of TLS – the basic generalization of LS tailored for fitting fully-perturbed linear models [283]. TLS and its variants involving regularization with the ℓ_2-norm of the unknown vector [76], have found widespread applications in diverse areas, including system identification with errors-in-variables (EIV), retrieval of spatial and temporal harmonics, reconstruction of medical images, and forecasting of financial data [209]. TLS was also utilized by [70] for dictionary learning, but the problem reduces to an over-determined linear system with a non-sparse unknown vector. Unfortunately, TLS approaches, with or without aforementioned regularization terms, cannot yield consistent estimators when the linear model is under-determined, nor they account for sparsity present in the unknown vector of regression coefficients.

System Model and Problem Statement

Consider the *under-determined* linear system of equations, $\mathbf{y} = \mathbf{C}\boldsymbol{\theta}_o$, where the unknown $n \times 1$ vector $\boldsymbol{\theta}_o$ is to be recovered from the given $m \times 1$ data vector $\mathbf{y}$ and the $m \times n$ matrix $\mathbf{C}$. With $m < n$ and no further assumption, only approximations of $\boldsymbol{\theta}_o$ are possible using the minimum-norm solution; or, the least-squares (LS) regularized by the ℓ_2-norm, which solves in closed form the quadratic problem: $\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{C}\boldsymbol{\theta}\|_2^2 + \gamma\|\boldsymbol{\theta}\|_2^2$ for some chosen $\gamma > 0$. Suppose instead that over a known basis matrix $\mathbf{B}$, the unknown vector satisfies $\boldsymbol{\theta}_o = \mathbf{B}\mathbf{x}_o$ with $\mathbf{x}_o$ being *sparse*, meaning that:
(as0) *The $n \times 1$ vector $\mathbf{x}_o$ contains more than $n - m$ zero elements at unknown entries.*
Under (as0) and certain conditions on the matrix $\mathbf{A} := \mathbf{CB}$, compressive sampling (CS) theory asserts that exact recovery of $\mathbf{x}_o$ can be guaranteed by solving the nonconvex, combinatorially complex problem: $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ subject to (s.to) $\mathbf{y} = \mathbf{A}\mathbf{x}$. More interestingly, the same assertion holds with quantifiable chances if one relaxes the ℓ_0- via the ℓ_1-norm, and solves efficiently the convex problem: $\min_{\mathbf{x}} \|\mathbf{x}\|_1$ s.to $\mathbf{y} = \mathbf{A}\mathbf{x}$ [61, 17, 39].
Suppose now that due to data perturbations the available vector $\mathbf{y}$ adheres only approximately to the linear model $\mathbf{A}\mathbf{x}_o$. The ℓ_1-norm based formulation accounting for the said perturbations is known as basis pursuit (BP) [61], and the corresponding convex problem written in its Lagrangian form is: $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda_1\|\mathbf{x}\|_1$, where $\lambda_1 > 0$ is a sparsity-tuning parameter. (For large λ_1, the solution is driven toward the all-zero vector; whereas for small λ_1 it tends to the LS solution.) This form of BP coincides with the Lasso approach developed for variable selection in linear regression problems [144, 278]. For uniformity with related problems, the BP/Lasso solvers can be equivalently written as

$$\{\hat{\mathbf{x}}_{Lasso}, \hat{\mathbf{e}}_{Lasso}\} := \arg\min_{\mathbf{x},\mathbf{e}} \ \|\mathbf{e}\|_2^2 + \lambda_1\|\mathbf{x}\|_1 \tag{26a}$$

$$\text{s. to } \ \mathbf{y} + \mathbf{e} = \mathbf{A}\mathbf{x}\,. \tag{26b}$$

Two interesting questions arise at this point: i) How is the performance of CS and BP/Lasso based reconstruction affected if perturbations appear also in $\mathbf{A}$? and ii) How can sparse vectors be efficiently reconstructed from over- and especially under-determined linear regression models while accounting for perturbations present in $\mathbf{y}$ and/or $\mathbf{A}$?

In the context of CS, perturbations in $\mathbf{A}$ can be due to disturbances in the compressing matrix $\mathbf{C}$, in the basis matrix $\mathbf{B}$, or in both. Those in $\mathbf{C}$ can be due to non-idealities in the analog implementation of CS; while those in $\mathbf{B}$ can also emerge because of mismatch between the adopted basis $\mathbf{B}$ and the actual one, which being unknown, is modeled as $\mathbf{B} + \mathbf{E}_B$. This mismatch emerges with grid-based approaches to localization, time delay, and spatio-temporal frequency or Doppler estimation [9, 20, 23, 54, 117, 118]. In these applications, the entries of $\boldsymbol{\theta}_o$ have e.g., a sparse discrete-time Fourier transform with peaks off the frequency grid $\{2\pi k/n\}_{k=0}^{n-1}$, but the postulated $\mathbf{B}$ is the fast Fourier transform (FFT) matrix built from this canonical grid. In this case, the actual linear relationship is $\boldsymbol{\theta}_o = (\mathbf{B} + \mathbf{E}_B)\mathbf{x}_o$ with $\mathbf{x}_o$ sparse. Bounds on the CS reconstruction error under basis mismatch are provided in [64]; see also [55], where the mismatch-induced error was reduced by increasing the grid density. Performance of BP/Lasso approaches for the under-determined, fully-perturbed (in both $\mathbf{y}$ and $\mathbf{A}$) linear model was analyzed in [148] by bounding the reconstruction error, and comparing it against its counterpart derived for the partially-perturbed (only in $\mathbf{y}$) model derived in [39]. Collectively, [64] and [148] address the performance question i), but provide no algorithms to address the open research issue ii).

The overarching theme of the considered approach is to address this issue by developing a sparse total least-squares (S-TLS) framework. Without exploiting sparsity, TLS has well-documented impact in applications as broad as linear prediction, system identification with errors-in-variables (EIV), spectral analysis, image reconstruction, speech, and audio processing, to name a few; see [283] and references therein. For over-determined models with unknown vectors $\mathbf{x}_o$ not abiding with (as0), TLS estimates are given by

$$\left\{\hat{\mathbf{x}}_{TLS}, \hat{\mathbf{E}}_{TLS}, \hat{\mathbf{e}}_{TLS}\right\} := \arg\min_{\mathbf{x},\mathbf{E},\mathbf{e}} \|[\mathbf{E}\ \mathbf{e}]\|_F^2 \tag{27a}$$

$$\text{s. to } \ \mathbf{y} + \mathbf{e} = (\mathbf{A} + \mathbf{E})\mathbf{x}. \tag{27b}$$

To cope with ill-conditioned matrices $\mathbf{A}$, an extra constraint bounding $\|\boldsymbol{\Gamma}\mathbf{x}\|_2$ is typically added in (27) to obtain different regularized TLS estimates depending on the choice of matrix $\boldsymbol{\Gamma}$ [76, 22].

The distinct objective of S-TLS relative to (regularized) TLS is twofold: account for sparsity as per (as0), and develop S-TLS solvers especially for under-determined, fully-perturbed linear models. To accomplish these goals, one must solve the S-TLS problem formulated (for $\lambda > 0$) as [cf. (26), (27)]

$$\left\{\hat{\mathbf{x}}_{S-TLS}, \hat{\mathbf{E}}_{S-TLS}, \hat{\mathbf{e}}_{S-TLS}\right\} := \arg\min_{\mathbf{x},\mathbf{e},\mathbf{E}} \|[\mathbf{E}\ \mathbf{e}]\|_F^2 + \lambda\|\mathbf{x}\|_1 \tag{28a}$$

$$\text{s. to } \mathbf{y} + \mathbf{e} = (\mathbf{A} + \mathbf{E})\mathbf{x}\,. \tag{28b}$$

The main goal is to develop efficient algorithms attaining at least the local and hopefully the global optimum of (28) – a challenging task since presence of the product $\mathbf{E}\mathbf{x}$ reveals that the problem is generally nonconvex. Similar to LS, BP, Lasso, and TLS, it is also worth stressing that the S-TLS estimates sought in (28) are universal in the sense that perturbations in $\mathbf{y}$ and $\mathbf{A}$ can be random or deterministic with or without a priori known structure.

But if prior knowledge is available on the perturbations, can weighted and structured S-TLS problems be formulated and solved? Can the scope of S-TLS be generalized (e.g., to recover a sparse matrix $\mathbf{X}_o$ using $\mathbf{A}$ and a data matrix $\mathbf{Y}$), and thus have impact in classical applications such as calibration of antenna arrays, or contemporary ones, such as cognitive radio sensing? Can S-TLS estimates be (e.g., Bayes) optimal if additional modeling assumptions are invoked? These questions will be addressed in the ensuing sections, starting from the last one.

MAP Optimality of S-TLS for EIV Models

Consider the EIV model with perturbed input ($\mathbf{A}$) and perturbed output ($\mathbf{y}$) obeying the relationship

$$\mathbf{y} = \mathbf{A}_o\mathbf{x}_o + (-\mathbf{e}_y)\,, \quad \mathbf{A} = \mathbf{A}_o + (-\mathbf{E}_A) \tag{29}$$

where the notation of the model perturbations $\mathbf{e}_y$ and $\mathbf{E}_A$ stresses their difference with $\mathbf{e}$ and $\mathbf{E}$, which are variables selected to yield the optimal S-TLS fit in (28). In a system identification setting, $\mathbf{e}_y$ and $\mathbf{E}_A$ are random perturbations giving rise to noisy output/input data $\mathbf{y}/\mathbf{A}$, based on which the task is to estimate the system vector $\mathbf{x}_o$ (comprising e.g., impulse response or pole-zero parameters), and possibly the inaccessible input matrix $\mathbf{A}_o$. To assess statistical optimality of the resultant estimators, collect the model perturbations in a column-vector form as $\text{vec}([\mathbf{E}_A\ \mathbf{e}_y])$, and further assume that:

(as1) *Perturbations of the EIV model in (29) are independent identically distributed (i.i.d.), Gaussian r.v.s, i.e., vec*$([\mathbf{E}_A\ \mathbf{e}_y]) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$*, independent from* $\mathbf{A}_o$ *and* $\mathbf{x}_o$*. Entries of* $\mathbf{x}_o$ *are zero-mean, i.i.d., according to a common Laplace distribution. In addition, either* **(a)** *the entries of* $\mathbf{x}_o$ *have common Laplacian parameter* $2/\lambda$*, and are independent from* $\mathbf{A}_o$*, which has i.i.d. entries drawn from a zero-mean uniform (i.e., non-informative) prior pdf; or,* **(b)** *the common Laplacian parameter of* $\mathbf{x}_o$ *entries is* $2(\sigma^2+1)/(\lambda\sigma^2)$*, and* $\mathbf{A}_o$ *conditioned on* $\mathbf{x}_o$ *has i.i.d. rows with pdf* $\mathcal{N}(\mathbf{0}, \sigma^2[\mathbf{I} - (1 + \|\mathbf{x}_o\|_2^2)^{-1}\mathbf{x}_o\mathbf{x}_o^T])$.
Note that the heavy-tailed Laplacian prior on $\mathbf{x}_o$ under (as1) is in par with the "non-probabilistic" sparsity attribute in (as0). It has been used to establish that the Lasso estimator in (26) is optimal, in the maximum a posteriori (MAP) sense, when $\mathbf{E}_A \equiv \mathbf{0}$ [278]. If on the other hand, $\mathbf{x}_o$ is viewed as non-sparse, deterministic and $\mathbf{A}_o$ as deterministic or as adhering to (as1b), it is known that the TLS estimator in (27) is optimum in the maximum likelihood (ML) sense for the EIV model in (29); see [209] and [225].

Aiming to establish optimality of S-TLS under (as1), it is useful to re-cast (28) as described in the following lemma. (This lemma will be used also in developing S-TLS solvers in Section 4.)

Lemma 1: *The constrained S-TLS formulation in (28) is equivalent to two unconstrained (also nonconvex) optimization problems: (a) one involving* $\mathbf{x}$ *and* $\mathbf{E}$ *variables, namely*

$$\left\{\hat{\mathbf{x}}_{S-TLS}, \hat{\mathbf{E}}_{S-TLS}\right\} = \arg\min_{\mathbf{x},\mathbf{E}} \left[\|\mathbf{y} - (\mathbf{A}+\mathbf{E})\mathbf{x}\|_2^2 + \|\mathbf{E}\|_F^2 + \lambda\|\mathbf{x}\|_1\right] \tag{30}$$

and (b) one of fractional form involving only the variable $\mathbf{x}$*, expressed as*

$$\hat{\mathbf{x}}_{S-TLS} := \arg\min_{\mathbf{x}} \frac{\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2}{1 + \|\mathbf{x}\|_2^2} + \lambda\|\mathbf{x}\|_1. \tag{31}$$

Proof: To establish the equivalence of (30) with (28), simply eliminate $\mathbf{e}$ by substituting the constraint (28b) into the cost function of (28a). For (31), let $\mathbf{v} := \text{vec}([\mathbf{E}\ \mathbf{e}])$, and re-write the cost in (28a) as $\|[\mathbf{E}\ \mathbf{e}]\|_F^2 = \|\mathbf{v}\|_2^2$; and the constraint (28b) as $\mathbf{y} - \mathbf{A}\mathbf{x} = \mathbf{G}(\mathbf{x})\mathbf{v}$, where $\mathbf{G}(\mathbf{x}) := \mathbf{I} \otimes [\mathbf{x}^T, -1]$. With $\mathbf{x}$ fixed, the ℓ_1-norm can be dropped from (28a), and the reformulated optimization becomes: $\min_{\mathbf{v}}\ \|\mathbf{v}\|_2^2$ s. to $\mathbf{y} - \mathbf{A}\mathbf{x} = \mathbf{G}(\mathbf{x})\mathbf{v}$. But the latter is a minimum-norm LS problem, admitting the closed-form solution

$$\begin{aligned}\mathbf{v}(\mathbf{x}) &= \mathbf{G}^T(\mathbf{x})[\mathbf{G}(\mathbf{x})\mathbf{G}^T(\mathbf{x})]^{-1}(\mathbf{y}-\mathbf{A}\mathbf{x}) \\ &= (1+\|\mathbf{x}\|_2^2)^{-1}\mathbf{G}^T(\mathbf{x})(\mathbf{y}-\mathbf{A}\mathbf{x}) \qquad (32)\end{aligned}$$

where the second equality holds because $\mathbf{G}(\mathbf{x})\mathbf{G}^T(\mathbf{x}) = \|[\mathbf{x}^T, -1]\|_2^2\mathbf{I} = (1+\|\mathbf{x}\|_2^2)\mathbf{I}$. Substituting (32) back into the cost $\|\mathbf{v}\|_2^2$, yields readily the fractional form in (31), which depends solely on $\mathbf{x}$. ■

Using Lemma 1, it is possible to establish MAP optimality of the S-TLS estimator as follows.

Proposition 1: *(MAP optimality). Under (as1), the S-TLS estimator in* (28) *is MAP optimal for the EIV model in* (29)*. Specifically,* (30) *is MAP optimal for estimating both* $\mathbf{x}_o$ *and* $\mathbf{A}_o$ *under (as1a), while* (31) *is MAP optimal for estimating only* $\mathbf{x}_o$ *under (as1b).*

Proof: Given $\mathbf{y}$ and $\mathbf{A}$, the MAP approach to estimating both $\mathbf{x}_o$ and $\mathbf{A}_o$ in (29) amounts to maximizing with respect to (wrt) $\mathbf{x}$ and $\mathbf{E}$ the logarithm of the posterior pdf denoted as $\ln p[\mathbf{x}_o = \mathbf{x}, \mathbf{A}_o = \mathbf{A}+\mathbf{E}|\mathbf{y}, \mathbf{A}]$. Recalling that $\mathbf{x}_o$ and $\mathbf{A}_o$ are independent under (as1a), Bayes' rule implies that this is equivalent to: $\min_{\mathbf{x},\mathbf{E}} -\{\ln p[\mathbf{y}, \mathbf{A}|\mathbf{x}_o = \mathbf{x}, \mathbf{A}_o = \mathbf{A}+\mathbf{E}] + \ln p[\mathbf{x}_o = \mathbf{x}] + \ln p[\mathbf{A}_o = \mathbf{A}+\mathbf{E}]\}$, where the summands correspond to the (conditional) log-likelihood and the log-prior pdfs, respectively. The log-prior associated with the Laplacian pdf of $\mathbf{x}_o$ is given by

$$\begin{aligned}\ln p[\mathbf{x}_o = \mathbf{x}] &= \ln \prod_{\nu=1}^{n} [(\lambda/4)\exp(-\lambda|x_\nu|/2)] \\ &= -(\lambda/2)\sum_{\nu=1}^{n} |x_\nu| + n\ln(\lambda/4) \qquad (33)\end{aligned}$$

while the log-prior associated with the uniform pdf of $\mathbf{A}_o$ is constant under (as1a), and thus does not affect the MAP criterion. Conditioning the log-likelihood on $\mathbf{x}_o$ and $\mathbf{A}_o$, implies that the only sources of randomness in the data $[\mathbf{y}\ \mathbf{A}]$ are the EIV model perturbations, which under (as1) are independent, standardized Gaussian; thus, the conditional log-likelihood is $\ln p[\mathbf{y}, \mathbf{A}|\mathbf{x}_o = \mathbf{x}, \mathbf{A}_o = \mathbf{A}+\mathbf{E}] = \ln[\mathbf{e}_y = \mathbf{y} - (\mathbf{A}+\mathbf{E})\mathbf{x}] + \ln p[\mathbf{E}_A = \mathbf{E}]$. After omitting terms not dependent on the variables $\mathbf{x}$ and $\mathbf{E}$, the latter shows that the log-likelihood contributes to the MAP criterion two quadratic terms (sum of two Gaussian exponents): $(1/2)\{\|\mathbf{y} - (\mathbf{A}+\mathbf{E})\mathbf{x}\|_2^2 + \|\mathbf{E}\|\}_F^2\}$. Upon combining these quadratic terms with the ℓ_1-norm coming from the sum in (33), the log-posterior pdf boils down to the form minimized in (30), which per Lemma 1 is equivalent to (28), and thus establishes MAP optimality of S-TLS under (as1a).

Proceeding to prove optimality under (as1b), given again the data $\mathbf{y}$ and $\mathbf{A}$, consider the MAP approach now to estimate only $\mathbf{x}_o$ in (29), treating $\mathbf{A}_o$ as a nuisance parameter matrix that satisfies (as1b). MAP here amounts to maximizing (wrt $\mathbf{x}$ only) the criterion $\ln p[\mathbf{x}_o = \mathbf{x}|\mathbf{y}, \mathbf{A}]$; and Bayes' rule leads to the equivalent problem $\min_{\mathbf{x}} -\{\ln p[\mathbf{y}, \mathbf{A}|\mathbf{x}_o = \mathbf{x}] + \ln p[\mathbf{x}_o = \mathbf{x}]\}$. But conditioned on $\mathbf{x}_o$, (as1b) dictates that $\mathbf{A}_o$ and $[\mathbf{E}_A\ \mathbf{e}_y]$ are zero-mean Gaussian and independent. Thus, linearity of the EIV model (29) implies that $\mathbf{y}$ and $\mathbf{A}$ are zero-mean jointly Gaussian in the conditional log-likelihood. Since rows of $\mathbf{A}_o$ and $[\mathbf{E}_A\ \mathbf{e}_y]$ are (conditionally) i.i.d. under (as1b), the rows of matrix $[\mathbf{A}\ \mathbf{y}]$ are independent. In addition, the ρth-row of $[\mathbf{A}\ \mathbf{y}]$ denoted as $[\mathbf{a}_\rho^T\ y_\rho]$, has inverse (conditional) covariance matrix

$$\begin{aligned}
&\mathbb{E}\left[\left[\begin{array}{c}\mathbf{a}_\rho\\ y_\rho\end{array}\right][\mathbf{a}_\rho^T\ y_\rho]\middle|\,\mathbf{x}_o=\mathbf{x}\right]^{-1}\\
&=\left[\begin{array}{cc}(\sigma^2+1)\mathbf{I}-\sigma^2\mathbf{x}\mathbf{x}^T/\left(1+\|\mathbf{x}\|_2^2\right) & \sigma^2\mathbf{x}/\left(1+\|\mathbf{x}\|_2^2\right)\\ \sigma^2\mathbf{x}^T/\left(1+\|\mathbf{x}\|_2^2\right) & 1+\sigma^2\|\mathbf{x}\|_2^2/\left(1+\|\mathbf{x}\|_2^2\right)\end{array}\right]^{-1}\\
&=\frac{1}{\sigma^2+1}\left\{\mathbf{I}+\frac{\sigma^2}{1+\|\mathbf{x}\|_2^2}\left[\begin{array}{c}\mathbf{x}\\ -1\end{array}\right][\mathbf{x}^T\ \ -1]\right\} \qquad (34)
\end{aligned}$$

with determinant $1/(\sigma^2+1)^n$ not a function of $\mathbf{x}$. After omitting such terms not dependent on $\mathbf{x}$, and using the independence among rows and their inverse covariance in (34), the conditional log-likelihood boils down to the fractional form $\frac{\sigma^2}{2(\sigma^2+1)}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2/\left(1+\|\mathbf{x}\|_2^2\right)$. Since the Laplacian parameter under (as1b) equals $2(\sigma^2+1)/(\lambda\sigma^2)$, the log-prior in (33) changes accordingly; and together with the fractional form of the log-likelihood reduces the negative log-posterior to the cost in (31). This establishes MAP optimality of the equivalent S-TLS in (28) for estimating only $\mathbf{x}_o$ in (29), under (as1b). ■

Proposition 1 will be generalized in Section 4 to account for structured and correlated perturbations with known covariance matrix. But before pursuing these generalizations, S-TLS solvers of the problem in (28) are in order.

S-TLS Solvers

Two iterative algorithms are developed in this section to solve the S-TLS problem in (28), which was equivalently re-formulated as in (30) and (31). The first algorithm can approach the *global optimum* but is computationally demanding; while the second one guarantees convergence to a *local optimum* but is computationally efficient. Thus, in addition to being attractive on its own, the second algorithm can serve as initialization to speed up convergence (and thus, reduce computational burden) of the first one. To appreciate the challenge and the associated performance-complexity trade-offs in developing algorithms for optimizing S-TLS criteria, it is useful to recall that all S-TLS problems are nonconvex; hence, unlike ordinary TLS that can be globally optimized (e.g., via SVD [209]), no efficient convex optimization solver is available with guaranteed convergence to the global optimum of (28), (30), or (31).

Bisection-Based ε-Optimal Algorithm

Viewing the cost in (31) as a Lagrangian function, allows casting this unconstrained minimization problem as a constrained one. Indeed, sufficiency of the Lagrange multiplier theory implies that [29, Sec. 3.3.4]: using the solution $\hat{\mathbf{x}}_{S-TLS}$ of (31) for a given multiplier $\lambda > 0$ and letting $\mu := \|\hat{\mathbf{x}}_{S-TLS}\|_1$, the pertinent constraint is $\mathcal{X}_1(\mu) := \{\mathbf{x} \in \mathbb{R}^n : \ \|\mathbf{x}\|_1 \leq \mu\}$; and the equivalent constrained minimization problem is [cf. (31)]

$$\hat{\mathbf{x}}_{S-TLS} := \arg \min_{\mathbf{x} \in \mathcal{X}_1(\mu)} f(\mathbf{x}) \ , \quad f(\mathbf{x}) := \frac{\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2}{1 + \|\mathbf{x}\|_2^2} \ . \tag{35}$$

There is no need to solve (31) in order to specify μ, because a cross-validation scheme can be implemented to specify μ in the stand-alone problem (35), along the lines used by e.g., [232] to determine λ in (31). The remainder of this subsection will thus develop an iterative scheme converging to the global optimum of (35), bearing in mind that this equivalently solves (31), (30) and (28) too.

From a high-level view, the presented scheme comprises an *outer iteration* loop based on the bisection method [82], and an *inner iteration* loop that relies on a variant of the branch-and-bound (BB) method [150]. A related approach was pursued in [22] to solve the clairvoyant TLS problem (27) under ℓ_2-norm regularization constraints. The challenging difference with the S-TLS here is precisely the non-differentiable ℓ_1-norm constraint in $\mathcal{X}_1(\mu)$. The outer iteration "squeezes" the minimum cost $f(\mathbf{x})$ in (35) between successively shrinking lower and upper bounds expressible through a parameter a. Per outer iteration, these bounds are obtained via inner iterations equivalently minimizing a surrogate quadratic function $g(\mathbf{x}, a)$, which does not have fractional form, and is thus more convenient to optimize than $f(\mathbf{x})$.

Given an upper bound a on $f(\mathbf{x})$, the link between $f(\mathbf{x})$ and $g(\mathbf{x}, a)$ follows if ones notes that

$$0 \leq a^\star := \min_{\mathbf{x} \in \mathcal{X}_1(\mu)} f(\mathbf{x}) = \min_{\mathbf{x} \in \mathcal{X}_1(\mu)} \frac{\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2}{1 + \|\mathbf{x}\|_2^2} \leq a \tag{36}$$

is equivalent to

$$g^\star(a) := \min_{\mathbf{x} \in \mathcal{X}_1(\mu)} g(\mathbf{x}, a) = \min_{\mathbf{x} \in \mathcal{X}_1(\mu)} \left\{ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 - a(1 + \|\mathbf{x}\|_2^2) \right\} \leq 0. \tag{37}$$

Suppose that after outer iteration i the optimum $a^\star$ in (36) belongs to a known interval $\mathcal{I}_i := [l_i, u_i]$. Suppose further that the inner loop yields the global optimum in (37) for $a = (l_i + u_i)/2$, and consider evaluating the sign of $g^\star(a)$ at this middle point $a = (l_i + u_i)/2$ of the interval $\mathcal{I}_i$. If $g^\star((l_i + u_i)/2) > 0$, the equivalence between (37) and (36) implies that $a^\star > (l_i + u_i)/2 > l_i$; and hence, $a^\star \in \mathcal{I}_{i+1} := [(l_i + u_i)/2, u_i]$, which yields a reduced-size interval $\mathcal{I}_{i+1}$ by shrinking $\mathcal{I}_i$ from the left. On the other hand, if $g^\star((l_i + u_i)/2) < 0$, the said equivalence will imply that $a^\star \in \mathcal{I}_{i+1} := [l_i, (l_i + u_i)/2]$, which shrinks the $\mathcal{I}_i$ interval from the right. This successive shrinkage through bisection explains how the outer iteration converges to the global optimum of (35).

What is left before asserting rigorously this convergence, is to develop the inner iteration which ensures that the global optimum in (37) can be approached for any given a specified by the outer bisection-based iteration. To appreciate the difficulty here note that the Hessian of $g(\mathbf{x}, a)$ is given by $\mathbf{H} := 2(\mathbf{A}^T\mathbf{A} - a\mathbf{I})$. Clearly, $\mathbf{H}$ is not guaranteed to be positive or negative definite since a is positive. As a result, the cost $g(\mathbf{x}, a)$ in (37) bypasses the fractional form of $f(\mathbf{x})$ but it is still an indefinite quadratic, and hence nonconvex. Nonetheless, the quadratic form of $g(\mathbf{x}, a)$ allows adapting the BB iteration of [150], which can yield a feasible and δ-optimum solution $\mathbf{x}_g^\star$ satisfying: a) $\mathbf{x}_g^\star \in \mathcal{X}_1(\mu)$; and b) $g^\star(a) \leq g(\mathbf{x}_g^\star, a) \leq g^\star(a) + \delta$, where δ denotes a pre-specified margin.

In the present context, the BB algorithm finds successive upper and lower bounds of the function

$$g^{\star}_{\text{box}}(a) := \min_{\mathbf{x}\in\mathcal{X}_1(\mu),\mathbf{x}_L\leq\mathbf{x}\leq\mathbf{x}_U} g(\mathbf{x},a) \tag{38}$$

where the constraint $\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U$ represents a box that shrinks as iterations progress. Upon converting the constraints of (38) to linear ones, upper bounds $\mathcal{U}$ on the function $g^{\star}_{\text{box}}(a)$ in (38) can be readily obtained via suboptimum solvers of the constrained optimization of the indefinite quadratic cost $g(\mathbf{x},a)$; see e.g., [29, Chp. 2]. Lower bounds on $g^{\star}_{\text{box}}(a)$ can be obtained by minimizing a convex function $g_L(\mathbf{x},a)$, which under-approximates $g(\mathbf{x},a)$ over the interval $\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U$. This convex approximant is given by

$$g_L(\mathbf{x},a) = g(\mathbf{x},a) + (\mathbf{x}-\mathbf{x}_L)^T\mathbf{D}(\mathbf{x}-\mathbf{x}_U) \tag{39}$$

where $\mathbf{D}$ is a diagonal positive semi-definite matrix chosen to ensure that $g_L(\mathbf{x},a)$ is convex, and stays as close as possible below $g(\mathbf{x},a)$. Such a matrix $\mathbf{D}$ can be found by minimizing the maximum distance between $g_L(\mathbf{x},a)$ and $g(\mathbf{x},a)$, and comes out as the solution of the following minimization problem:

$$\min_{\mathbf{D}} \ (\mathbf{x}_U - \mathbf{x}_L)^T\mathbf{D}(\mathbf{x}_U - \mathbf{x}_L) \ \text{ s. to } \ \mathbf{H} + 2\mathbf{D} \succeq \mathbf{0} \tag{40}$$

where the constraint on the Hessian ensures that $g_L(\mathbf{x},a)$ remains convex. Since (40) is a semi-definite program, it can be solved efficiently using available convex optimization software; e.g., the interior point optimization routine in SeDuMi [271]. Having selected $\mathbf{D}$ as in (40), $\min_{\mathbf{x}\in\mathcal{X}_1(\mu),\mathbf{x}_L\leq\mathbf{x}\leq\mathbf{x}_U}$ $g_L(\mathbf{x},a)$ is a convex problem (quadratic cost under linear constraints); thus, similar to the upper bound $\mathcal{U}$, the lower bound $\mathcal{L}$ on $g^{\star}_{\text{box}}(a)$ can be obtained efficiently.

The detailed inner loop (BB scheme) is tabulated as Algorithm 4-a. It amounts to successively splitting the initial box $-\mu\mathbf{1} \leq \mathbf{x} \leq \mu\mathbf{1}$, which is the smallest one containing $\mathcal{X}_1(\mu)$. Per inner iteration i, variable $\mathcal{U}$ keeps track of the upper bound on $g^{\star}_{\text{box}}(a)$, which at the end outputs to the outer loop the nearest estimate of $g^{\star}(a)$. Concurrently, the lower bound $\mathcal{L}$ on $g^{\star}_{\text{box}}(a)$ determines whether the current box needs to be further split, or discarded, if the difference $\mathcal{U} - \mathcal{L}$ is smaller than the pre-selected margin δ. This iterative splitting leads to a decreasing $\mathcal{U}$ and a tighter $\mathcal{L}$, both of which prevent further splitting.

Algorithm 4-a (BB): Input $\mathbf{y}$, $\mathbf{A}$, a, and δ. Output a δ-optimal solution $\mathbf{x}_g^\star$ of (37)

```
Set $\mathbf{x}_L = -\mu\mathbf{1}$, $\mathbf{x}_U = \mu\mathbf{1}$, $\Omega := \{(\mathbf{x}_L, \mathbf{x}_U, -\infty\}$, and initialize with $\mathcal{U} = \infty$.
repeat
  Let $(\mathbf{x}_L, \mathbf{x}_U, c)$ be one triplet of $\Omega$ with the smallest $c$; and set $\Omega = \Omega \backslash (\mathbf{x}_L, \mathbf{x}_U, c)$.
  Solve (38) locally to obtain $\hat{\mathbf{x}}_g^\star$.
  if $g(\hat{\mathbf{x}}_g^\star, a) < \mathcal{U}$ then
    Set $\mathcal{U} = g(\hat{\mathbf{x}}_g^\star, a)$ and $\mathbf{x}_g^\star = \hat{\mathbf{x}}_g^\star$. {update the minimum}
  end if
  Minimize globally the convex $g_L(\mathbf{x}, a)$ in (39) with the optimum $\mathbf{D}$ in (40), to obtain $\check{\mathbf{x}}_g^\star$ and $\mathcal{L} := g_L(\check{\mathbf{x}}_g^\star, a)$.
  if $\mathcal{U} - \mathcal{L} > \delta$ {need to split} then
    Find $i = \arg\max_n([\mathbf{x}_U]_n - [\mathbf{x}_L]_n)$.
    Set $\mathbf{x}_{L,1}$ ($\mathbf{x}_{U,1}$) and $\mathbf{x}_{L,2}$ ($\mathbf{x}_{U,2}$) equal to $\mathbf{x}_L$ ($\mathbf{x}_U$) except for the $i$-th entry. {split the maximum separation}
    Set $[\mathbf{x}_{L,1}]_i = [\mathbf{x}_L]_i$, $[\mathbf{x}_{U,1}]_i = ([\mathbf{x}_U]_i - [\mathbf{x}_L]_i)/2$, $[\mathbf{x}_{L,2}]_i = ([\mathbf{x}_U]_i - [\mathbf{x}_L]_i)/2$, and $[\mathbf{x}_{U,2}]_i = [\mathbf{x}_U]_i$.
    Augment the set of unsolved boxes $\Omega = \Omega \bigcup \{(\mathbf{x}_{L,1}, \mathbf{x}_{U,1}, \mathcal{L}), (\mathbf{x}_{L,2}, \mathbf{x}_{U,2}, \mathcal{L})\}$.
  end if
until $\Omega = \emptyset$
```

Recapitulating, the outer bisection-based iteration tabulated as Algorithm 4-b calls Algorithm 4-a to find a feasible δ-optimal solution $\mathbf{x}_g^\star$ to evaluate the sign of $g^\star(a)$ in (37). Since $\mathbf{x}_g^\star$ is not the exact global minimum of (37), positivity of $g(\mathbf{x}_g^\star, a)$ does not necessarily imply $g^\star(a) > 0$. But $\mathbf{x}_g^\star$ is δ-optimal, meaning that $g^\star(a) \geq g(\mathbf{x}_g^\star, a) - \delta$; thus, $g(\mathbf{x}_g^\star, a) > \delta$, in which case the lower bound l_{i+1} is updated to $(l_i + u_i)/2$; otherwise, if $g(\mathbf{x}_g^\star, a) \in (0, \delta)$, then l_{i+1} should be set to $(l_i + u_i)/2 - \delta$.

As far as convergence is concerned, the following result can be established.

Algorithm 4-b (Bisection): Input $\mathbf{y}$, $\mathbf{A}$, and tolerances ε and δ. Output an ε-optimal solution $\mathbf{x}_\varepsilon^\star$ to (35)

```
Set $l_0 = 0$, $u_0 = \|\mathbf{y}\|_2^2$, iteration index $i = 0$, and initialize the achievable cost $f_m = u_0$ with $\mathbf{x}_\varepsilon^\star = \mathbf{0}_{n\times 1}$.
repeat
    Let $a = (l_i + u_i)/2$ and call Algorithm 4-a to find a feasible $\delta$-optimal solution $\mathbf{x}_g^\star$ to (37).
    Calculate $f_g = f(\mathbf{x}_g^\star)$, and update the iteration $i = i + 1$.
    if $f_g < f_m$ then
        Set $f_m = f_g$ and $\mathbf{x}_\varepsilon^\star = \mathbf{x}_g^\star$. {update the minimum}
    end if
    if $g(\mathbf{x}_g^\star, a) \leq 0$ then
        Update $u_i = a$ and $l_i = l_{i-1}$.
    else if $g(\mathbf{x}_g^\star, a) \geq \delta$ then
        Update $l_i = a$ and $u_i = u_{i-1}$.
    else
        Update $l_i = a - \delta$ and $u_i = u_{i-1}$.
    end if
    Set $u_i = \min(u_i, f_g)$.
until $u_i - l_i \leq \varepsilon$
```

Proposition 2: *(ε-optimal convergence) After at most $\left\lceil \ln\left(\frac{u_0}{\varepsilon - 2\delta}\right) / \ln(2) \right\rceil$ iterations, Algorithm 4-b outputs an ε-optimal solution $\mathbf{x}_\varepsilon^\star$ to (35); that is,*

$$\mathbf{x}_\varepsilon^\star \in \mathcal{X}_1(\mu), \ \ \textit{and} \ \ a^\star \leq f(\mathbf{x}_\varepsilon^\star) \leq a^\star + \varepsilon. \tag{41}$$

Proof: Upon updating the lower and upper bounds, it holds per outer iteration $i \geq 1$ that $u_i - l_i \leq \frac{1}{2}(u_{i-1} - l_{i-1}) + \delta$; and by induction, $u_i - l_i \leq \left(\frac{1}{2}\right)^i u_0 + 2\delta$, when $l_0 = 0$. The latter implies that if the number of iterations $i \geq \left\lceil \ln\left(\frac{u_0}{\varepsilon - 2\delta}\right) / \ln(2) \right\rceil$, the distance $u_i - l_i \leq \varepsilon$ is satisfied.

Since per outer iteration Algorithm 4-a outputs $\mathbf{x}_g^\star \in \mathcal{X}_1(\mu)$, it holds that the updated $\mathbf{x}_\varepsilon^\star$ is also feasible. Further, the bisection process guarantees that $l_i \leq a^\star \leq f(\mathbf{x}_\varepsilon^\star) \leq u_i$ per iteration i. Since Algorithm 4-b ends with $u_i - l_i \leq \varepsilon$, the inequality in (41) follows readily. ■

Proposition 2 quantifies the number of outer iterations needed by the bisection-based Algorithm 4-b to approach within ε the global optimum of (35). In addition, the inner (BB) iterations bounding $g^{\star}_{\text{box}}(a)$ are expected to be fast converging because the box function in (38) is tailored for the box constraints induced by the ℓ_1-norm regularization. Nonetheless, similar to all BB algorithms, the complexity of Algorithm 4-a does not have guaranteed polynomial complexity on average. The latter necessitates as few calls of Algorithm 4-a, which means as few outer iterations. Proposition 2 reveals that critical to this end is the initial upper bound u_0 (Algorithm 4-b simply initializes with $u_0 = f(\mathbf{0})$).

This motivates the efficient suboptimal S-TLS solver of the next subsection, which is of paramount importance not only on its own, but also for initializing the ε-optimal algorithm.

Alternating Descent Sub-Optimal Algorithm

The starting point for a computationally efficient S-TLS solver is the formulation in (30). Given $\mathbf{E}$, the cost in (30) has the form of the Lasso problem in (26); while given $\mathbf{x}$, it reduces to a quadratic form, which admits closed-form solution wrt $\mathbf{E}$. These observations suggest an iterative block coordinate descent algorithm yielding successive estimates of $\mathbf{x}$ with $\mathbf{E}$ fixed, and alternately of $\mathbf{E}$ with $\mathbf{x}$ fixed. Specifically, with the iterate $\mathbf{E}(i)$ given per iteration $i \geq 0$, the iterate $\mathbf{x}(i)$ is obtained by solving the Lasso-like convex problem as [cf. (26)]

$$\mathbf{x}(i) = \arg\min_{\mathbf{x}} \|\mathbf{y} - [\mathbf{A} + \mathbf{E}(i)]\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 \ . \tag{42}$$

With $\mathbf{x}(i)$ available, $\mathbf{E}(i+1)$ for the ensuing iteration is found as

$$\mathbf{E}(i+1) = \arg\min_{\mathbf{E}} \|\mathbf{y} - \mathbf{A}\mathbf{x}(i) - \mathbf{E}\mathbf{x}(i)\|_2^2 + \|\mathbf{E}\|_F^2. \tag{43}$$

By setting the first-order derivative of the cost wrt $\mathbf{E}$ equal to zero, the optimal solution to the quadratic problem (43) is obtained in closed form as

$$\mathbf{E}(i+1) = (1 + \|\mathbf{x}(i)\|_2^2)^{-1}[\mathbf{y} - \mathbf{A}\mathbf{x}(i)]\mathbf{x}^T(i). \tag{44}$$

The iterations are initialized at $i = 0$ by setting $\mathbf{E}(0) = \mathbf{0}_{m\times n}$. Substituting the latter into (42), yields $\mathbf{x}(0) = \hat{\mathbf{x}}_{Lasso}$ in (26). That this is a good initial estimate is corroborated by the result in [148], which shows that even with perturbations present in both $\mathbf{A}$ and $\mathbf{y}$, the CS (and thus Lasso) estimators yield accurate reconstruction. In view of the fact that the block coordinate descent iterations ensure that the cost in (30) is non-increasing, the final estimates upon convergence will be at least as accurate.

The block coordinate descent algorithm is provably convergent to a stationary point of the S-TLS cost in (30), and thus to its equivalent forms in (28), (31) and (35), as asserted in the following proposition.

Proposition 3: *(Convergence of alternating descent) Given arbitrary initialization, the iterates* $\{\mathbf{E}(i), \mathbf{x}(i)\}$ *given by* (42) *and* (44) *converge monotonically at least to a stationary point of the S-TLS problem* (28).

Proof: The argument relies on the basic convergence result in [280]. The alternating descent algorithm specified by (42) and (44) is a special case of the block coordinate descent method using the cyclic rule for minimizing the cost in (30). The first two summands of this cost are differentiable wrt the optimization variables, while the non-differential third term (ℓ_1-norm regularization) is separable in the entries of $\mathbf{x}$. Hence, the three summands satisfy the assumptions (B1)–(B3) and (C2) in [280]. Convergence of the iterates $\{\mathbf{E}(i), \mathbf{x}(i)\}$ to a coordinate minimum point of the cost thus follows by appealing to [280, Thm. 5.1]. Moreover, the first summand is Gâteaux-differentiable over its domain which is open. Hence, the cost in (30) is regular at each coordinate's minimum point, and every coordinate's minimum point becomes a stationary point; see [280, Lemma 3.1]. Monotonicity of the convergence follows simply because the cost per iteration may either reduce or maintain its value. ■

Proposition 3 solidifies the merits of the alternating descent S-TLS solver. Simulated tests will further demonstrate that the local optimum guaranteed by this computationally efficient scheme is very close to the global optimum attained by the more complex scheme of the previous subsection.

Since estimating $\mathbf{E}$ is simple using the closed form in (43), it is useful at this point to explore modifications, extensions and tailored solvers for the problem in (42) by adapting to the present setup existing results from the Lasso literature dealing with problem (26). From the plethora of available options to solve (42), it is worth mentioning two computationally efficient ones: the least-angle regression (LARS), and the coordinate descent (CD); see e.g., [144]. LARS provides the entire "solution path" of (42) for all $\lambda > 0$ at complexity comparable to LS. On the other hand, if a single "best" value of λ is fixed using the cross-validation scheme [232], then CD is the state-of-the-art choice for solving (42).

CD in the present context cycles between iterates $\mathbf{E}(i)$, and scalar iterates of the $\mathbf{x}(i)$ entries. Suppose that the ν-th entry $x_\nu(i)$ is to be found. Precursor entries $\{x_1(i), \ldots, x_{\nu-1}(i)\}$ have been already obtained in the i-th iteration, and postcursor entries $\{x_{\nu+1}(i-1), \ldots, x_n(i-1)\}$ are also available from the previous $(i-1)$-st iteration along with $\mathbf{E}(i)$ obtained in closed form as in (44). If $\boldsymbol{\alpha}_\nu(i)$ denotes the ν-th column of $[\mathbf{A} + \mathbf{E}(i)]$, the effect of these known entries can be removed from $\mathbf{y}$ by forming

$$\mathbf{e}_\nu(i) := \mathbf{y} - \sum_{j=1}^{\nu-1} \boldsymbol{\alpha}_j(i) x_j(i) - \sum_{j=\nu+1}^{n} \boldsymbol{\alpha}_j(i) x_j(i-1) \,. \tag{45}$$

Using (45), the vector optimization problem in (42) reduces to the following scalar one with $x_\nu(i)$ as unknown: $x_\nu(i) = \arg\min_{x_\nu} [\|\mathbf{e}_\nu(i) - \boldsymbol{\alpha}_\nu(i) x_\nu\|_2^2 + \lambda |x_\nu|]$. This *scalar* Lasso problem is known to admit a closed-form solution expressed in terms of a soft thresholding operator (see e.g., [144])

$$x_\nu(i) = \operatorname{sign}\left(\mathbf{e}_\nu^T(i) \boldsymbol{\alpha}_\nu(i)\right) \left[\frac{|\mathbf{e}_\nu^T(i) \boldsymbol{\alpha}_\nu(i)|}{\|\boldsymbol{\alpha}_\nu(i)\|_2^2} - \frac{\lambda}{2\|\boldsymbol{\alpha}_\nu(i)\|_2^2} \right]_+ , \ \nu = 1, \ldots, n \tag{46}$$

where $\operatorname{sign}(\cdot)$ denotes the sign operator, and $[\chi]_+ := \chi$, if $\chi > 0$, and zero otherwise.

Cycling through the closed forms (44)-(46) explains why CD here is faster than, and thus preferable over general-purpose convex optimization solvers of (42). Another factor contributing to its speed is the sparsity of $\mathbf{x}(i)$, which implies that starting up with the all-zero vector, namely $\mathbf{x}(-1) = \mathbf{0}_{n\times 1}$, offers initialization close to a stationary point of the cost in (30). Convergence to this stationary point is guaranteed by using the results in [280], along the lines of Proposition 3. Note also that larger values of λ in (46) force more entries of $\mathbf{x}(i)$ to be shrunk to zero, which corroborates the role of λ as a sparsity-tuning parameter. The CD based S-TLS solver is tabulated as Algorithm 5.

Remark 1: *(Regularization options for S-TLS)* Lasso estimators are known to be biased, but modifications are available to remedy bias effects. One such modification is the weighted Lasso, which replaces the ℓ_1-norm in (28) by its weighted version, namely $\sum_{\nu=1}^{n} w_\nu |x_\nu|$, where the weights $\{w_\nu\}$ are chosen

Algorithm 5 (CD): Input $\mathbf{y}$, $\mathbf{A}$, and coefficient λ. Output the iterates $\mathbf{E}(i)$ and $\mathbf{x}(i)$ upon convergence.

```
Initialize with $\mathbf{E}(0) = \mathbf{0}_{m\times n}$ and $\mathbf{x}(-1) = \mathbf{0}_{n\times 1}$
for $i = 0, 1, \ldots$ do
  for $\nu = 1, \ldots, n$ do
    Compute the residual $\mathbf{e}_\nu(i)$ as in (45).
    Update the scalar $x_\nu(i)$ via (46).
  end for
  Update the iterate $\mathbf{E}(i+1)$ as in (44).
end for
```

using the LS solution [331]. An alternative popular choice is to replace the ℓ_1-norm with concave regularization terms [103], such as $\sum_{\nu=1}^{n} \log(x_\nu + \delta_1)$, where δ_1 is a small positive constant introduced to avoid numerical instability. In addition to mitigating bias effects, concave regularization terms provide tighter approximations to the ℓ_0-(pseudo)norm, and although they render the cost in (28) nonconvex, they are known to converge very fast to an improved estimate of $\mathbf{x}$, when initialized with the Lasso solution [103].

Remark 2: *(Group Lasso and Matrix S-TLS)* When groups $\{\mathbf{x}_g\}_{g=1}^{G}$ of $\mathbf{x}$ entries are a priori known to be zero or nonzero (as a group), the ℓ_1-norm in (28) must be replaced by the sum of ℓ_2-norms, namely $\sum_{g=1}^{G} \|\mathbf{x}_g\|_2$. The resulting group S-TLS estimate can be obtained using the group-Lasso solver [144]. In the present context, this is further useful if one considers the matrix counterpart of the S-TLS problem in (28), which in its unconstrained form can be written as [cf. (30)]

$$\left\{\hat{\mathbf{X}}_{S-TLS}, \hat{\mathbf{E}}_{S-TLS}\right\} = \arg\min_{\mathbf{X},\mathbf{E}} \Big[\|\mathbf{Y} - (\mathbf{A}+\mathbf{E})\mathbf{X}\|_F^2 + \|\mathbf{E}\|_F^2 + \lambda \sum_{\nu=1}^{n} \|\mathbf{x}_\nu^T\|_2 \Big] \tag{47}$$

where $\mathbf{x}_\nu^T$ denotes the ν-th row of the $n \times L$ unknown matrix $\mathbf{X}$, which is sparse in the sense that a number of its rows are zero, and has to be estimated using an $m \times L$ data matrix $\mathbf{Y}$ along with the regression matrix $\mathbf{A}$, both with perturbations present. Problem (47) can be solved using block coordinate descent cycling between iterates $\mathbf{E}(i)$ and rows $\mathbf{x}_\nu^T(i)$ as opposed to scalar entries as in (46).

Weighted and Structured S-TLS

Apart from the optimality links established in Proposition 1 under (as1), the S-TLS criteria in (28), (30), and (31) make no assumption on the perturbations $[\mathbf{E}\ \mathbf{e}]$. In this sense, the S-TLS solvers of the previous section find universal applicability. However, one expects that exploiting prior information on $[\mathbf{E}\ \mathbf{e}]$, can only lead to improved performance. Thinking for instance along the lines of weighted LS, one is motivated to *weight* $\|\mathbf{E}\|_F^2$ and $\|\mathbf{e}\|_2^2$ in (30) by the inverse covariance matrix of $\mathbf{E}$ and $\mathbf{e}$, respectively, whenever those are known and are not both equal to $\mathbf{I}$. As a second motivating example, normal equations, involved in e.g., linear prediction, entail *structure* in $\mathbf{E}$ and $\mathbf{e}$ that capture sample estimation errors present in the matrix $[\mathbf{A}\ \mathbf{y}]$, which is Toeplitz. Prompted by these examples, this section is about broadening the scope of S-TLS with weighted and structured forms capitalizing on prior information available about the matrix $[\mathbf{E}\ \mathbf{e}]$. To this end, it is prudent to quantify first the notion of structure.

Definition 1. *The* $m \times (n+1)$ *data matrix* $[\mathbf{A}\mathbf{y}](\mathbf{p})$ *has structure characterized by an* $n_p \times 1$ *parameter vector* $\mathbf{p}$, *if and only if there is a mapping such that* $\mathbf{p} \in \mathbb{R}^{n_p} \to [\mathbf{A}\ \mathbf{y}](\mathbf{p}) := \mathbf{S}(\mathbf{p}) \in \mathbb{R}^{m\times(n+1)}$.

Definition 1 is general enough to encompass any (even unstructured) matrix $[\mathbf{A}\ \mathbf{y}](\mathbf{p})$, by simply letting $\mathbf{p} := \text{vec}([\mathbf{A}\ \mathbf{y}]) \in \mathbb{R}^{m(n+1)}$ comprise all entries of $[\mathbf{A}\ \mathbf{y}]$. However, it becomes more relevant when $n_p \ll m(n+1)$, the case in which $\mathbf{p}$ characterizes $[\mathbf{A}\ \mathbf{y}]$ parsimoniously. Application examples are abundant: structure in Toeplitz and Hankel matrices encountered with system identification, deconvolution, and linear prediction; as well as in circulant and Vandermonde matrices showing up in spatio-temporal harmonic retrieval problems [209]. Structured matrices $\mathbf{A}$ and sparse vectors $\mathbf{x}_o$ emerge also in contemporary CS gridding-based applications e.g., for spectral analysis and estimation of time-varying channels, where rows of the FFT matrix are selected at random. (This last setting appears when training orthogonal frequency-division multiplexing (OFDM) input symbols are used to estimate communication links exhibiting variations due to mobility-induced Doppler effects [23].)

Consider now re-casting the S-TLS criteria in terms of $\mathbf{p}$, and its associated perturbation vector denoted by $\boldsymbol{\epsilon} \in \mathbb{R}^{n_p}$. The Frobenius norm in the cost of (28a) is mapped to the ℓ_2-norm of $\boldsymbol{\epsilon}$; and to allow for weighting the structured perturbation vector using a symmetric positive definite matrix $\mathbf{W} \in \mathbb{R}^{n_p \times n_p}$, the weighted counterpart of $\|[\mathbf{E}\ \mathbf{e}]\|_F^2$ becomes $\boldsymbol{\epsilon}^T\mathbf{W}\boldsymbol{\epsilon}$. With regards to the constraint, recall first from Definition 1 that $\mathbf{S}(\mathbf{p}) = [\mathbf{A}\ \mathbf{y}]$, which implies $\mathbf{S}(\mathbf{p}+\boldsymbol{\epsilon}) = [\mathbf{A}+\mathbf{E}\ \mathbf{y}+\mathbf{e}]$; hence, re-writing (28b) as $[\mathbf{A}+\mathbf{E}\ \mathbf{y}+\mathbf{e}]\left[\mathbf{x}^T, -1\right]^T = \mathbf{0}$, yields the structured constraint as $\mathbf{S}(\mathbf{p}+\boldsymbol{\epsilon})\left[\mathbf{x}^T, -1\right]^T = \mathbf{0}$. Putting things together, leads to the combined weighted-structured S-TLS version of (28) as

$$\min_{\mathbf{x},\boldsymbol{\epsilon}} \quad \boldsymbol{\epsilon}^T\mathbf{W}\boldsymbol{\epsilon} + \lambda\|\mathbf{x}\|_1 \tag{48a}$$

$$\text{s. to} \quad \mathbf{S}(\mathbf{p}+\boldsymbol{\epsilon})\begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix} = \mathbf{0} \tag{48b}$$

which clearly subsumes the structure-only form as a special case corresponding to $\mathbf{W} = \mathbf{I}$.

To confine the structure quantified in Definition 1, two conditions will be imposed, which are commonly adopted by TLS approaches [209], and are satisfied by most applications mentioned so far.

(as2) *The structure mapping in Definition 1 is separable, meaning that with* $\mathbf{p} = [(\mathbf{p}^A)^T\ (\mathbf{p}^y)^T]^T$, *where* $\mathbf{p}^A \in \mathbb{R}^{n_A}$ *and* $\mathbf{p}^y \in \mathbb{R}^{n_y}$, *it holds that* $\mathbf{S}(\mathbf{p}) := [\mathbf{A}\ \mathbf{y}](\mathbf{p}) = [\mathbf{A}(\mathbf{p}^A)\ \mathbf{y}(\mathbf{p}^y)]$. *In addition, the separable structure mapping is linear (more precisely affine), if and only if the* $\mathbf{S}(\mathbf{p})$ *matrix is composed of known structural elements, namely "matrix atoms"* $\mathbf{S}_0$, $\{\mathbf{S}_k^A\}_{k=1}^{n_A}$ *and "vector atoms"* $\{\mathbf{s}_k^y\}_{k=1}^{n_y}$, *so that*

$$\mathbf{S}(\mathbf{p}) = \mathbf{S}_0 + \left[\sum_{k=1}^{n_A} p_k^A \mathbf{S}_k^A \quad \sum_{k=1}^{n_y} p_k^y \mathbf{s}_k^y\right] \tag{49}$$

where p_k^A *(*p_k^y*) denotes the* k*-th entry of* $\mathbf{p}^A(\mathbf{p}^y)$*.*

Similar to Definition 1, (49) is general enough to encompass even unstructured matrices $\mathbf{S}(\mathbf{p}) := [\mathbf{A}\ \mathbf{y}]$, by setting $\mathbf{S}_0 = \mathbf{0}$, $\mathbf{p} := \text{vec}([\mathbf{A}\ \mathbf{y}]) \in \mathbb{R}^{m(n+1)}$, $n_p := n_A + n_y = mn + m$, and selecting the m vector atoms (mn matrix

atoms) as the canonical vectors (matrices), each with one entry equal to 1 and all others equal to 0. Again, interesting structures are those with $n_A \ll mn$ and/or $n_y \ll m$. (Consider for instance a circulant $m \times n$ matrix $\mathbf{A}$, which can be represented as in (49) using $n_A = m$ matrix atoms.)
Separability and linearity will turn out to simplify the constraint in (48b) for some given matrix atoms and vector atoms collected for notational brevity in the matrices

$$\mathbf{S}^A := [\mathbf{S}_1^A \cdots \mathbf{S}_{n_A}^A] \quad \text{and} \quad \mathbf{S}^y := [\mathbf{s}_1^y \ \cdots \ \mathbf{s}_{n_y}^y] \in \mathbb{R}^{m \times n_y} \,. \tag{50}$$

Indeed, linearity in (as2) allows one to write $\mathbf{S}(\mathbf{p}+\boldsymbol{\epsilon}) = \mathbf{S}(\mathbf{p}) + \mathbf{S}(\boldsymbol{\epsilon})$, and the constraint (48b) as: $\mathbf{S}(\boldsymbol{\epsilon})[\mathbf{x}^T, -1]^T = -\mathbf{S}(\mathbf{p})[\mathbf{x}^T, -1]^T = \mathbf{y} - \mathbf{A}\mathbf{x}$; while separability implies that

$$\begin{aligned}\mathbf{S}(\boldsymbol{\epsilon})\left[\mathbf{x}^T, -1\right]^T &= \left[\sum_{k=1}^{n_A} \epsilon_k^A \mathbf{S}_k^A \quad \sum_{k=1}^{n_y} \epsilon_k^y \mathbf{s}_k^y\right]\left[\mathbf{x}^T, -1\right]^T \\ &= \mathbf{S}^A(\mathbf{I} \otimes \mathbf{x})\boldsymbol{\epsilon}^A - \mathbf{S}^y \boldsymbol{\epsilon}^y,\end{aligned}$$

where the definitions $\boldsymbol{\epsilon} := [(\boldsymbol{\epsilon}^A)^T \ (\boldsymbol{\epsilon}^y)^T]^T$ and (50) were used in the last equality along with the identity $\sum_{k=1}^{n_A} \epsilon_k^A \mathbf{S}_k^A \mathbf{x} = \mathbf{S}^A(\mathbf{I} \otimes \mathbf{x})\boldsymbol{\epsilon}^A$. In a nutshell, (48b) under (as2) becomes $\mathbf{S}^A(\mathbf{I} \otimes \mathbf{x})\boldsymbol{\epsilon}^A - \mathbf{S}^y\boldsymbol{\epsilon}^y = \mathbf{y} - \mathbf{A}\mathbf{x}$, in which $\boldsymbol{\epsilon}^A$ is decoupled from $\boldsymbol{\epsilon}^y$.
Therefore, the weighted and structured (WS)S-TLS problem in (48) reduces to [cf. (28)]

$$\min_{\mathbf{x}, \boldsymbol{\epsilon}^A, \boldsymbol{\epsilon}^y} \ \begin{bmatrix} \boldsymbol{\epsilon}^A \\ \boldsymbol{\epsilon}^y \end{bmatrix}^T \mathbf{W} \begin{bmatrix} \boldsymbol{\epsilon}^A \\ \boldsymbol{\epsilon}^y \end{bmatrix} + \lambda \|\mathbf{x}\|_1 \tag{51a}$$

$$\text{s. to } \ [\mathbf{S}^A(\mathbf{I} \otimes \mathbf{x}) \ \ -\mathbf{S}^y] \begin{bmatrix} \boldsymbol{\epsilon}^A \\ \boldsymbol{\epsilon}^y \end{bmatrix} = \mathbf{y} - \mathbf{A}\mathbf{x} \tag{51b}$$

or in a more compact form as: $\min_{\mathbf{x}, \boldsymbol{\epsilon}} \{\boldsymbol{\epsilon}^T \mathbf{W} \, \boldsymbol{\epsilon} + \lambda \|\mathbf{x}\|_1\}$ s.t. $\mathbf{G}(\mathbf{x})\boldsymbol{\epsilon} = \mathbf{r}(\mathbf{x})$, after defining

$$\mathbf{G}(\mathbf{x}) := [\mathbf{S}^A(\mathbf{I} \otimes \mathbf{x}) \ \ \mathbf{S}^y] \quad \text{and} \quad \mathbf{r}(\mathbf{x}) := \mathbf{y} - \mathbf{A}\mathbf{x} \,. \tag{52}$$

Comparing (28) with (51) allows one to draw apparent analogies: both involve three sets of optimization variables, and both are nonconvex because two of these sets enter the corresponding constraints in a bilinear fashion [cf. product of $\mathbf{E}$ with $\mathbf{x}$ in (28b), and $\boldsymbol{\epsilon}^A$ with $\mathbf{x}$ in (51b)].

Building on these analogies, the following lemma shows how to formulate WSS-TLS criteria, paralleling those of Lemma 1, where one or two sets of variables were eliminated to obtain efficient, provably convergent solvers, and establish statistical optimality links within the EIV model in (29).

Lemma 2: *The constrained WSS-TLS form in* (48) *is equivalent to two unconstrained nonconvex optimization problems: (a) one involving* $\mathbf{x}$ *and* $\boldsymbol{\epsilon}^A$ *variables, namely*

$$\{\hat{\mathbf{x}}, \hat{\boldsymbol{\epsilon}}^A\}_{WSS-TLS} = \arg\min_{\mathbf{x},\boldsymbol{\epsilon}^A} \begin{bmatrix} \boldsymbol{\epsilon}^A \\ (\mathbf{S}^y)^\dagger \left[\mathbf{S}^A(\mathbf{I}\otimes\mathbf{x})\boldsymbol{\epsilon}^A - \mathbf{r}(\mathbf{x})\right] \end{bmatrix}^T \mathbf{W} \\ \times \begin{bmatrix} \boldsymbol{\epsilon}^A \\ (\mathbf{S}^y)^\dagger[\mathbf{S}^A(\mathbf{I}\otimes\mathbf{x})\boldsymbol{\epsilon}^A - \mathbf{r}(\mathbf{x})] \end{bmatrix} + \lambda\|\mathbf{x}\|_1 \quad (53)$$

where $\mathbf{S}^y$ *is assumed full rank and square*[1]*, i.e.,* $m = n_y$ *in* (52)*; and also (b) one involving only the variable* $\mathbf{x}$*, expressed using the definitions in* (52)*, as*

$$\hat{\mathbf{x}}_{WSS-TLS} = \arg\min_{\mathbf{x}} \left\{\mathbf{r}^T(\mathbf{x}) \left[\mathbf{G}(\mathbf{x})\,\mathbf{W}^{-1}\,\mathbf{G}^T(\mathbf{x})\right]^\dagger \mathbf{r}(\mathbf{x}) + \lambda\|\mathbf{x}\|_1\right\}. \quad (54)$$

Proof: Constraint (51b) can be solved uniquely for $\boldsymbol{\epsilon}^y$ to obtain $\boldsymbol{\epsilon}^y = (\mathbf{S}^y)^\dagger[\mathbf{S}^A(\mathbf{I}\otimes\mathbf{x})\boldsymbol{\epsilon}^A - (\mathbf{y}-\mathbf{A}\mathbf{x})]$. Plug the latter with the definition of $\mathbf{r}(\mathbf{x})$ from (52) into the quadratic form in (51a) to recognize that (51) is equivalent to the unconstrained form in (53) with the $\boldsymbol{\epsilon}^y$ variable eliminated.

To arrive at (54), suppose that $\mathbf{x}$ is given and view the compact form of (51) (after ignoring $\lambda\|\mathbf{x}\|_1$) as the following weighted minimum-norm LS problem: $\min_{\mathbf{x},\boldsymbol{\epsilon}}\{\boldsymbol{\epsilon}^T\mathbf{W}\boldsymbol{\epsilon}+\lambda\|\mathbf{x}\|_1\}$ s.to $\mathbf{G}(\mathbf{x})\boldsymbol{\epsilon} = \mathbf{r}(\mathbf{x})$. Solving the latter in closed form expresses $\boldsymbol{\epsilon}$ in terms of $\mathbf{x}$ as: $\boldsymbol{\epsilon}(\mathbf{x}) = \mathbf{W}^{-1}\mathbf{G}^T(\mathbf{x})\left[\mathbf{G}(\mathbf{x})\mathbf{W}^{-1}\mathbf{G}^T(\mathbf{x})\right]^\dagger \mathbf{r}(\mathbf{x})$. Substitute now $\boldsymbol{\epsilon}(\mathbf{x})$ back into the cost, and reinstate $\lambda\|\mathbf{x}\|_1$, to obtain (54). ■

The formulation in (53) suggests directly an iterative WSS-TLS solver based on the block coordinate descent method. Specifically, suppose that the estimate

[1] Tall $\mathbf{S}^y$ matrices with full column rank can be handled too for block diagonal weight matrices $\mathbf{W}$ typically adopted with separable structures; see also [323]. This explains why the pseudo-inverse of $\mathbf{S}^y$ is used in this section instead of its inverse; but exposition of the proof simplifies considerably for the square case. Note also that the full rank assumption is not practically restrictive because data matrices perturbed by noise of absolutely continuous pdf have full rank almost surely.

$\boldsymbol{\epsilon}^A(i)$ of $\boldsymbol{\epsilon}^A$ is available at iteration i. Substituting $\boldsymbol{\epsilon}^A(i)$ into (53), allows estimating $\mathbf{x}$ as

$$\begin{aligned}\mathbf{x}(i) = \arg\min_{\mathbf{x}} & \begin{bmatrix} \boldsymbol{\epsilon}^A(i) \\ (\mathbf{S}^y)^\dagger \left[\mathbf{S}^A(\mathbf{I}\otimes\mathbf{x})\boldsymbol{\epsilon}^A(i) - \mathbf{r}(\mathbf{x})\right] \end{bmatrix}^T \mathbf{W} \\ & \times \begin{bmatrix} \boldsymbol{\epsilon}^A(i) \\ (\mathbf{S}^y)^\dagger \left[\mathbf{S}^A(\mathbf{I}\otimes\mathbf{x})\boldsymbol{\epsilon}^A(i) - \mathbf{r}(\mathbf{x})\right] \end{bmatrix} + \lambda\|\mathbf{x}\|_1. \end{aligned} \tag{55}$$

Since $\mathbf{r}(\mathbf{x})$ is linear in $\mathbf{x}$ [cf. (52)], the cost in (55) is convex (quadratic regularized by the ℓ_1-norm as in the Lasso cost in (26)); thus, it can be minimized efficiently. Likewise, given $\mathbf{x}(i)$ the perturbation vector for the ensuing iteration can be found in closed form since the pertinent cost is quadratic; that is,

$$\begin{aligned}\boldsymbol{\epsilon}^A(i+1) = \arg\min_{\boldsymbol{\epsilon}^A} & \begin{bmatrix} \boldsymbol{\epsilon}^A \\ (\mathbf{S}^y)^\dagger \left[\mathbf{S}^A(\mathbf{I}\otimes\mathbf{x}(i))\boldsymbol{\epsilon}^A - \mathbf{r}(\mathbf{x}(i))\right] \end{bmatrix}^T \\ & \times \mathbf{W} \begin{bmatrix} \boldsymbol{\epsilon}^A \\ (\mathbf{S}^y)^\dagger \left[\mathbf{S}^A(\mathbf{I}\otimes\mathbf{x}(i))\boldsymbol{\epsilon}^A - \mathbf{r}(\mathbf{x}(i))\right] \end{bmatrix}. \end{aligned} \tag{56}$$

To express $\boldsymbol{\epsilon}^A(i+1)$ compactly, partition $\mathbf{W}$ in accordance with $\mathbf{p} = [(\mathbf{p}^A)^T,\ (\mathbf{p}^y)^T]^T$; i.e., let

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{AA} & \mathbf{W}_{Ay} \\ \mathbf{W}_{Ay}^T & \mathbf{W}_{yy} \end{bmatrix}. \tag{57}$$

Using (57), and equating to zero the gradient (wrt $\boldsymbol{\epsilon}^A$) of the cost in (56), yields the closed form

$$\boldsymbol{\epsilon}^A(i+1) = \left\{\check{\mathbf{S}}(\mathbf{x}(i))\mathbf{W}\check{\mathbf{S}}^T(\mathbf{x}(i))\right\}^\dagger \check{\mathbf{S}}(\mathbf{x}(i)) \left[\mathbf{W}_{Ay}^T,\ \mathbf{W}_{yy}^T\right]^T (\mathbf{S}^y)^\dagger \mathbf{r}(\mathbf{x}(i)) \tag{58}$$

where $\check{\mathbf{S}}(\mathbf{x}(i)) := \left[\mathbf{I},\ \left[(\mathbf{S}^y)^\dagger\mathbf{S}^A(\mathbf{I}\otimes\mathbf{x}(i))\right]^T\right]$.

Initialized with $\boldsymbol{\epsilon}^A(0) = \mathbf{0}_{n_A\times 1}$, the algorithm cycles between iterations (55) and (58). Mimicking the steps of Proposition 3, it is easy to show that these iterations are convergent as asserted in the following.

Proposition 4: *(Convergence). The iterates in* (55) *and* (58) *converge monotonically at least to a stationary point of the cost in* (48)*, provided that* $\mathbf{S}^y$ *in* (52) *has full column rank.*

As with the solver of Section 4, CD is also applicable to the WSS-TLS solver, by cycling between $\boldsymbol{\epsilon}^A(i)$ and scalar iterates of the $\mathbf{x}(i)$ entries. To update the ν-th entry $x_\nu(i)$, suppose precursor entries $\{x_1(i), \ldots, x_{\nu-1}(i)\}$ have been already obtained in the i-th iteration, and postcursor entries $\{x_{\nu+1}(i-1), \ldots, x_n(i-1)\}$ are also available from the previous $(i-1)$-st iteration along with $\boldsymbol{\epsilon}^A(i)$, found in closed form as in (58). Letting $\boldsymbol{\alpha}_\nu(i)$ denote the ν-th column of $\left[(\mathbf{S}^y)^\dagger\left(\mathbf{A} + \sum_{k=1} \epsilon_k^A \mathbf{S}_k^A\right)\right]$, the effect of these known entries can be removed from $\mathbf{y}$ by forming [cf. (45)]

$$\mathbf{e}_\nu(i) := (\mathbf{S}^y)^\dagger \mathbf{y} - \sum_{j=1}^{\nu-1} \boldsymbol{\alpha}_j(i) x_j(i) - \sum_{j=\nu+1}^{n} \boldsymbol{\alpha}_j(i) x_j(i-1) \,. \tag{59}$$

Using (59), the vector optimization problem in (55) now reduces to the following scalar one with $x_\nu(i)$ as unknown: $x_\nu(i) = \arg\min_{x_\nu}\{\|\boldsymbol{\alpha}_\nu(i)x_\nu - \mathbf{e}_\nu(i)\|^2_{W_{yy}} + 2[\boldsymbol{\epsilon}^A(i)]^T\, \mathbf{W}_{Ay}\boldsymbol{\alpha}_\nu(i)x_\nu + \lambda|x_\nu|\}$, where $\|\cdot\|_{W_{yy}}$ denotes the ℓ_2-norm weighted by $\mathbf{W}_{yy}$. The solution of this scalar Lasso problem can be expressed using the same soft-thresholding form as in (46), and is given by

$$\begin{aligned} x_\nu(i) &= \operatorname{sign}\left(\left[\mathbf{e}_\nu(i)\mathbf{W}_{yy} - \boldsymbol{\epsilon}^A(i)\mathbf{W}_{Ay}^T\right]^T \boldsymbol{\alpha}_\nu(i)\right) \\ &\times \left[\frac{\left|\left[\mathbf{e}_\nu(i)\mathbf{W}_{yy} - \boldsymbol{\epsilon}^A(i)\mathbf{W}_{Ay}^T\right]^T \boldsymbol{\alpha}_\nu(i)\right|}{\|\boldsymbol{\alpha}_\nu(i)\|^2_{W_{yy}}} - \frac{\lambda}{2\|\boldsymbol{\alpha}_\nu(i)\|^2_{W_{yy}}}\right]_+ \,. \end{aligned} \tag{60}$$

This block CD algorithm enjoys fast convergence (at least) to a stationary point, thanks both to the simplicity of (60), and the sparsity of $\mathbf{x}(i)$, as explained in Section IV-B.

The WSS-TLS criterion in (53) is also useful to establish its statistical optimality under a structured EIV model, with output-input data obeying the relationships

$$\mathbf{y} = \mathbf{A}(\mathbf{p}_o^A)\mathbf{x}_o + (-\mathbf{S}^y\boldsymbol{\epsilon}_y)\,, \quad \mathbf{A} = \mathbf{A}(\mathbf{p}_o^A) + \left(-\sum_{k=1}^{n_A} \epsilon_{A,k}\mathbf{S}_k^A\right) \tag{61}$$

where perturbation vectors $\boldsymbol{\epsilon}_A$ and $\boldsymbol{\epsilon}_y$ play the role of $\mathbf{E}_A$ and $\mathbf{e}_y$ in (29), and differ from the optimization variables $\boldsymbol{\epsilon}^A$ and $\boldsymbol{\epsilon}^y$ in (51). Unknown are the vector $\mathbf{x}_o$, and the inaccessible input matrix $\mathbf{A}(\mathbf{p}_o^A)$, characterized by the vector $\mathbf{p}_o^A$. The model in (61) obeys the following structured counterpart of (as1a).

(as1′) *Perturbations in* (61) *are jointly Gaussian, i.e.,* $[\boldsymbol{\epsilon}_A\ \boldsymbol{\epsilon}_y] \sim \mathcal{N}(\mathbf{0}, \mathbf{W}^{-1})$*, as well as independent from* $\mathbf{p}_o^A$ *and* $\mathbf{x}_o$*. Vector* $\mathbf{x}_o$ *has i.i.d. entries with the same prior as in (as1a); and it is independent from* $\mathbf{p}_o^A$*, which has i.i.d. entries drawn from a zero-mean uniform (i.e., non-informative) prior pdf.*

The following optimality claim holds for the WSS-TLS estimator in (53), assured to be equivalent to the solution of problem (51) by Lemma 2.

Proposition 5: *(MAP optimality of WSS-TLS). Under (as1′) and (as2), the equivalent WSS-TLS problem in* (53) *yields the MAP optimal estimator of* $\mathbf{x}_o$ *and* $\mathbf{p}_A$ *in the structured EIV model* (61)*.*

Proof: The proof follows the lines used in proving the MAP optimality of (30) under (as1a) in Proposition 1. The log-prior pdf of $\mathbf{x}_o$ contains an ℓ_1-norm term as in (33), while the uniform prior on $\mathbf{p}_o^A$ is constant under (as1′). Furthermore, given the structure mapping $[\mathbf{A}\ \mathbf{y}] = \mathbf{S}(\mathbf{p})$, the conditional log-likelihood here can be expressed in terms of $\mathbf{x}$ and $\boldsymbol{\epsilon}^A$, as $\ln p[\mathbf{y}, \mathbf{A}|\mathbf{x}_o = \mathbf{x}, \mathbf{p}_o^A = \mathbf{p}^A + \boldsymbol{\epsilon}^A] = \ln p\Big[\boldsymbol{\epsilon}_A = \boldsymbol{\epsilon}^A,\ \boldsymbol{\epsilon}_y = -(\mathbf{S}^y)^\dagger[\mathbf{y} - (\mathbf{A} + \sum_{k=1}^{n_A} \epsilon_k^A \mathbf{S}_k^A)\mathbf{x}] = (\mathbf{S}^y)^\dagger[\mathbf{S}^A(\mathbf{I} \otimes \mathbf{x})\boldsymbol{\epsilon}^A - \mathbf{r}(\mathbf{x})]\Big]$. After omitting terms not dependent on $\mathbf{x}$ and $\boldsymbol{\epsilon}^A$, the conditional log-likelihood under the joint Gaussian distribution in (as1′) boils down to half of the quadratic cost in (53). Combining the latter with $\|\mathbf{x}\|_1$ from the log-prior pdf, it follows that maximizing the log-posterior pdf amounts to minimizing the unconstrained sum of the two, which establishes MAP optimality of the WSS-TLS estimator in (53). ■

S-TLS Applications

In this section, the practical impact of accounting for perturbations present in the data matrix $[\mathbf{A}\ \mathbf{y}]$ will be demonstrated via two sensing applications involving reconstruction of sparse vectors. In both, the perturbation $\mathbf{E}_A$ comes from inaccurate modeling of the underlying actual matrix $\mathbf{A}_o$, while $\mathbf{e}_y$ is due to measurement noise.

Cognitive Radio Sensing

Consider N_s sources located at unknown positions, each transmitting an RF signal with power spectral density (PSD) $\Phi_s(f)$ that is well approximated by a basis expansion model: $\Phi_s(f) = \sum_{\nu=1}^{N_b} x_{s\nu} b_\nu(f)$, where $\{b_\nu(f)\}_{\nu=1}^{N_b}$ are known (e.g., rectangular) basis functions, and $\{x_{s\nu}\}_{s=1}^{N_s}$ are unknown power coefficients. As source positions are also unknown, a Cartesian grid of known points $\{\mathbf{l}_g\}_{g=1}^{N_g}$ is adopted to describe *candidate* locations that transmitting radios could be positioned [54, 20]; see also Fig. 12.

The task is to estimate the locations and powers of active sources based on PSD samples measured at N_r cognitive radios (CRs) at known locations $\{\boldsymbol{\ell}_r\}_{r=1}^{N_r}$. Per frequency f_k, these samples obey the model

$$\begin{aligned}\hat{\Phi}_r(f_k) &= \sum_{g=1}^{N_g} \gamma_{gr} \Phi_g(f_k) + \sigma_r^2 + e_r(f_k) \\ &= \left[\sum_{g=1}^{N_g} \sum_{\nu=1}^{N_b} \gamma_{gr} b_\nu(f_k)\right] x_{g\nu} + \sigma_r^2 + e_r(f_k) \\ &= \mathbf{a}_r^T(f_k)\mathbf{x}_o + e_r(f_k) \qquad (62)\end{aligned}$$

where the PSD $\Phi_g(f)$ is nonzero only if a transmitting source is present at $\mathbf{l}_g$; γ_{gr} represents the channel gain from the candidate source at $\mathbf{l}_g$ to the CR at $\boldsymbol{\ell}_r$ that is assumed to follow a known pathloss function of the distance $||\mathbf{l}_g - \boldsymbol{\ell}_r||$; σ_r^2 denotes the known noise variance at receiver r; the $N_b N_g \times 1$ vector $\mathbf{a}_r(f_k)$ collects products $\gamma_{gr} b_\nu(f_k)$; vector $\mathbf{x}_o$ contains the $N_b N_g$ unknown power coefficients $x_{g\nu}$; and $e_r(f_k)$ captures the error between the true, $\Phi_r(f_k)$, and estimated, $\hat{\Phi}_r(f_k)$, PSDs.

Estimated PSD samples at K frequencies from all N_r receivers are first compensated by subtracting the corresponding noise variances, and subsequently collected to form the data vector $\mathbf{y}$ of length $m = KN_r$. Noise terms $-e_r(f_k)$ are similarly collected to build the perturbation vector $\boldsymbol{\epsilon}_y$. Likewise, row vectors $\mathbf{a}_r^T(f_k)$ of length $n = N_b N_g$ are concatenated to form the $m \times n$ matrix $\mathbf{A}$. The latter is perturbed (relative to the inaccessible $\mathbf{A}_o$) by a matrix $\mathbf{E}_A$, which accounts for the mismatch between grid location vectors, $\{\mathbf{l}_g\}_{g=1}^{N_g}$, and those of the actual sources, $\{\boldsymbol{\kappa}_s\}_{s=1}^{N_s}$. To specify $\mathbf{E}_A$, let $\epsilon_{gr}^s := \gamma_{sr} - \gamma_{gr}$ for the source at $\boldsymbol{\kappa}_s$ closest to $\mathbf{l}_g$, and substitute $\gamma_{gr} = \gamma_{sr} - \epsilon_{gr}^s$ into the double sum inside the square brackets of (62). This allows writing $\mathbf{A} = \mathbf{A}_o - \mathbf{E}_A$,

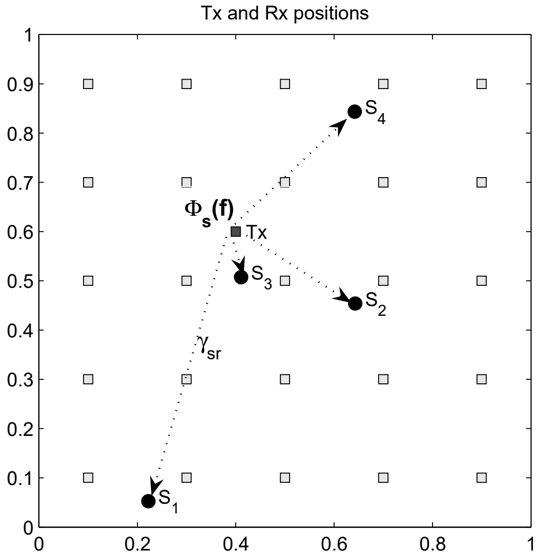


Figure 12: Grid topology with $N_g = 25$ candidate locations, $N_s = 1$ transmitting source, and $N_r = 4$ receiving CRs.

where $\mathbf{E}_A$ is affine structured with coefficients $\{\epsilon^s_{gr}\}$ and matrix atoms formed by $\{b_\nu(f_k)\}$. All in all, the setup fits nicely the structured EIV model in (61). Together with $\{\epsilon^s_{gr}\}$, the support of $\mathbf{x}_o$ estimates the locations of sources, and the nonzero entries of $\mathbf{x}_o$ their transmit powers. Remarkably, this grid-based approach reduces localization – traditionally a nonlinear estimation task – to a linear one, by increasing the problem dimensionality ($N_g \gg N_s$). What is more, $\mathbf{x}_o$ is sparse for two reasons: a) relative to the swath of available bandwidth, the transmitted PSDs are narrowband; hence, the number of nonzero $x_{g\nu}$s is small relative to N_b; and (b) the number of actual sources (N_s) is much smaller than the number of grid points (N_g) that is chosen large enough to localize sources with sufficiently high resolution. Existing sparsity-exploiting approaches to CR sensing rely on BP/Lasso, and do not take into account the mismatch arising due to griding [54, 20]. Simulations in Section 4 will demonstrate that sensing accuracy improves considerably if one accounts for grid-induced errors through the EIV model, and compensates for them via the devised WSS-TLS estimators.

DoA Estimation via Sparse Linear Regression

The setup here is the classical one in sensor array processing: plane waves from N_s far-field, narrowband sources impinge on a uniformly-spaced linear array (ULA) of N_r (possibly uncalibrated) antenna elements. Based on as few $N_r \times 1$ vectors of spatial samples collected across the ULA per time instant t (snapshot), the task is to localize sources by estimating their directions-of-arrival (DoA) denoted as $\{\vartheta_s\}_{s=1}^{N_s}$. High-resolution, (weighted) subspace-based DoA estimators are nonlinear, and rely on the sample covariance matrix of these spatio-temporal samples, which requires a relatively large number of snapshots for reliable estimation especially when the array is not calibrated; see e.g., [153]. This has prompted recent DoA estimators based on sparse linear regression, which rely on a uniform polar grid of N_g points describing *candidate* DoAs $\{\theta_g\}_{g=1}^{N_g}$ [118, 201, 228]. Similar to the CR sensing problem, the g-th entry x_g of the $N_g \times 1$ unknown vector of regression coefficients, $\mathbf{x}_{o,t}$, is nonzero and equal to the transmit-source signal power, if a source is impinging at angle θ_g, and zero otherwise.

The $N_r \times 1$ array response vector to a candidate source at DoA θ_g is $\mathbf{a}(\theta_g) = [1\ e^{-j\alpha_g}\ \cdots\ e^{-j\alpha_g(N_r-1)}]^T$, where $\alpha_g := 2\pi d \sin(\theta_g)$ denotes the phase shift relative to the source signal wavelength between neighboring ULA elements separated by distance d. The per-snapshot received data vector $\mathbf{y}_t$ of length $m = N_r$ obeys the EIV model: $\mathbf{y}_t = (\mathbf{A} + \mathbf{E}_A)\,\mathbf{x}_{o,t} + (-\mathbf{e}_{y,t})$, where $-\mathbf{e}_{y,t}$ represents the additive noise across the array elements; the $m \times n$ matrix $\mathbf{A} := [\mathbf{a}(\theta_1)\ \cdots\ \mathbf{a}(\theta_{N_g})]$ denotes the grid angle scanning matrix of $n = N_g$ columns; and $\mathbf{E}_A$ represents perturbations arising because DoAs from actual sources do not necessarily lie on the postulated grid points. Matrix $\mathbf{E}_A$ can also account for gain, phase, and position errors of antenna elements when the array is uncalibrated.

To demonstrate how a structured S-TLS approach applies to the DoA estimation problem at hand, consider for simplicity one source from direction ϑ_s, whose nearest grid angle is θ_g; and let $\epsilon_g^s := \vartheta_s - \theta_g$ be the corresponding error that vanishes as the grid density grows large. For small ϵ_g^s, the actual source-array phase shift $\alpha_s := 2\pi d \sin(\theta_g + \epsilon_g^s)$ can be safely approximated as $2\pi d[\sin(\theta_g)\cos(\epsilon_g^s) + \cos(\theta_g)\sin(\epsilon_g^s)] \approx 2\pi d[\sin(\theta_g) + \epsilon_g^s \cos(\theta_g)]$; or, more compactly as $\alpha_s \approx \alpha_g + \epsilon_g \beta_g$, where $\beta_g := 2\pi d \cos(\theta_g)$. As a result, using the approximation $\exp(-j\alpha_s) \approx \exp[-j(\alpha_g + \epsilon_g^s \beta_g)] \approx (1 - j\epsilon_g^s \beta_g)\exp(-j\alpha_g)$, the actual array response vector can be approximated as a linear function of ϵ_g^s; thus, it be expressed as

$$\begin{aligned}&\mathbf{a}(\vartheta_s) = \mathbf{a}(\theta_g) + \epsilon_g^s i \boldsymbol{\phi}(\theta_g)\\ &\boldsymbol{\phi}(\theta_g) := \Big[0 \;\; -j\beta_g e^{-j\alpha_g}, \ldots, -j\beta_g (N_r - 1) e^{-j\alpha_g (N_r - 1)}\Big]^T .\end{aligned} \tag{63}$$

With columns obeying (63), the actual array manifold is modeled as $\mathbf{A}_o = \mathbf{A} + \mathbf{E}_A$, where the perturbation matrix is structured as $\mathbf{E}_A = \sum_{g=1}^{N_g} \epsilon_g^s \mathbf{S}_g^A$, with the $N_r \times N_g$ matrix $\mathbf{S}_g^A$ having all zero entries, except for the g-th column that equals $\boldsymbol{\phi}(\theta_g)$. With such an array manifold and $\mathbf{S}^y = \mathbf{I}$, the grid-based DoA setup matches precisely the structured EIV model in (61). The simulated tests in the ensuing section will illustrate, among other things, the merits of employing WSS-TLS solvers to estimate $\mathbf{x}_{o,t}$ and ϵ_g^s based on data collected by possibly antenna arrays. But before this, a final remark is in order.

Remark 3: *(Relationships with [117] and [201])* Although $\mathbf{E}_A$ is not explicitly included in the model of existing grid-based approaches, this mismatch has been mitigated either by iteratively refining the grid around the region where sources are present [201], or, by invoking the minimum description length (MDL) test to estimate the number of actual sources N_s, followed by spatial interpolation to estimate their DoAs [117]. These remedies require post-processing the initial estimates obtained by sparse linear regression. In contrast, the proposed *structured* S-TLS based approach *jointly* estimates the nonzero support of $\mathbf{x}_{o,t}$ along with grid-induced perturbations. This allows for direct compensation of the angle errors to obtain high-resolution DoA estimates in a single step, and in certain cases, without requiring multiple snapshots. Of course, multiple snapshots are expected to improve estimation performance using the matrix S-TLS solver mentioned in Remark 2.

Simulated Tests

Four simulated tests are presented in this section to illustrate the merits of the S-TLS approach, starting from the algorithms of Section 4.

Test Case 1: *(Optimum vs. suboptimum S-TLS)* The EIV model in (29) is simulated here with a 6×10 matrix $\mathbf{A}$, whose entries are i.i.d. Gaussian having variance $1/6$, so that the expected ℓ_2-norm of each column equals 1. The entries of $\mathbf{E}_A$ and $\mathbf{e}_y$ are also i.i.d. Gaussian with variance $0.0025/6$

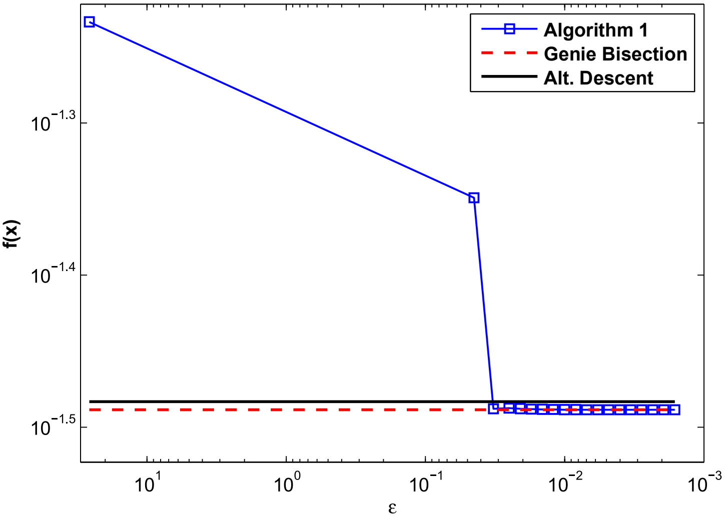


Figure 13: Attained $f(\mathbf{x})$ for variable tolerance values ε by the global Algorithm 4-b, compared to the alternating descent local algorithm, and the genie-aided global solver.

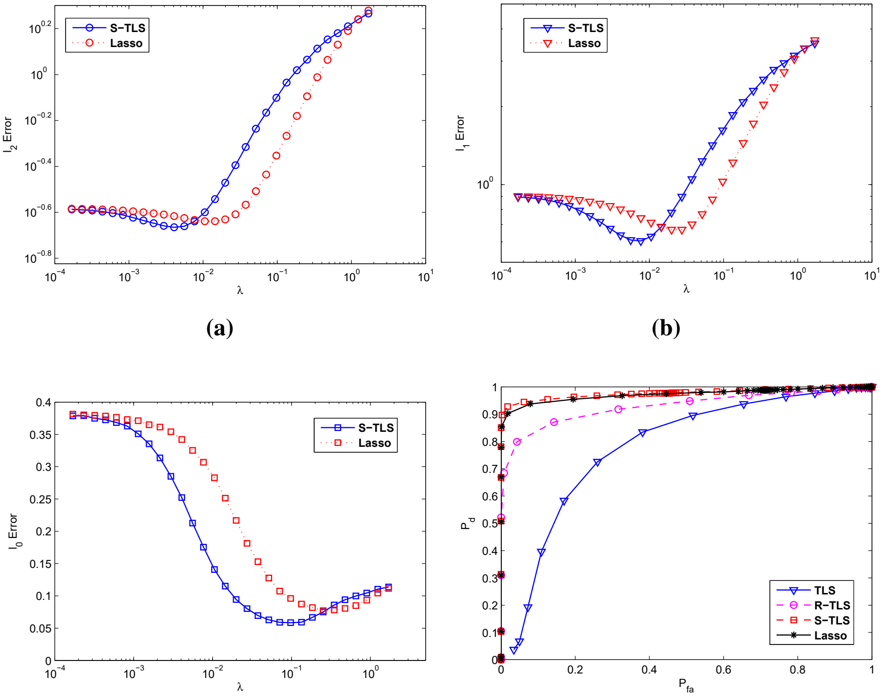
corresponding to entry-wise signal-to-noise ratio (SNR) of 26dB. Vector $\mathbf{x}_o$ has only nonzero elements in the two first entries: $x_{o,1} = -1.3$ and $x_{o,2} = 5$. Algorithm 1 is tested with $\mu = 5$ against Algorithm 5 implemented with different values of λ to obtain a solution satisfying $\|\hat{\mathbf{x}}_{S-TLS}\|_1 = \mu$. For variable tolerance values ε in Algorithm 4-b, the attained minimum cost $f(\mathbf{x})$ in (36) is plotted in Fig. 13. To serve as a benchmark, a genie-aided globally optimum scheme is also tested with the support of $\mathbf{x}$ known and equal to that of $\mathbf{x}_o$. Specifically, the genie-aided scheme minimizes $f(\mathbf{x})$ over all points with ℓ_1-norm equal to μ, and all entries being 0 except for the first two. Using the equivalence between (36) and (37), the genie-aided scheme per iteration amounts to minimizing a scalar quadratic program under linear constrains, which is solved efficiently using the interior-point optimization routine in [271].

Fig. 13 shows that as ε becomes smaller, the minimum achieved value $f(\mathbf{x})$ decreases monotonically, and drops sharply to the global minimum attained by the genie-aided bisection scheme. Interestingly, the alternating descent algorithm that guarantees convergence to a stationary point, exhibits performance comparable to the global algorithm. For this reason, only the alternating descent algorithm is used in all subsequent tests. Next, S-TLS estimates are compared with those obtained via BP/Lasso and (regularized) TLS in the context of the CR sensing and array processing applications outlined in Section 4.

Test Case 2: *(S-TLS vs. Lasso vs. TLS)* The setup here is also based on the EIV model (29), with $\mathbf{A}$ of size 20×40 having i.i.d. Gaussian entries; and $\mathbf{x}_o$ having 5 nonzero i.i.d. standardized Gaussian entries. All other parameters are as in Test Case 1 adapted to the different problem size here. By averaging results over 200 Monte Carlo runs, the S-TLS solution is compared against the Lasso one for 20 values of λ (uniformly spaced in log-scale), based on the ℓ_2, ℓ_1, and ℓ_0 errors of the estimated vectors relative to $\mathbf{x}_o$. (The ℓ_0 error equals the percentage of entries for which the support of the two vectors is different.) Fig. 14 corroborates the improvement of S-TLS over Lasso, especially in the ℓ_0 norm. Fig. 14(c) further demonstrates that over a range of moderate λ values, S-TLS consistently outperforms Lasso in recovering the true support of $\mathbf{x}_o$. For high λ's, both estimates come close to the all-zero vector, so that the ℓ_0 errors become approximately the same, even though the ℓ_2 and ℓ_1 errors are smaller for Lasso. However, for both error norms S-TLS has a slight edge over moderate values of λ.

Receiver operating characteristic (ROC) curves are plotted in Fig. 14(d) to illustrate the merits of S-TLS and Lasso over (regularized) TLS in recovering the correct support. The "best" λ for the S-TLS and Lasso algorithms is chosen using cross-validation [232]. As TLS cannot be applied to underdetermined systems, a 40×40 matrix $\mathbf{A}$ is selected. Since TLS and LS under an ℓ_2-norm constraint $\|\mathbf{x}\|_2 \leq \mu$ are known to be equivalent when μ is small [76], the regularized TLS is tested using the function `lsqi` for regularized LS from [143]. The probability of correct detection, P_d, is calculated as the probability of identifying correctly the support over nonzero entries of $\mathbf{x}_o$, and the probability of false alarms, P_{fa}, as that of incorrectly deciding zero entries to be nonzero. The ROC curves in Fig. 14(d) demonstrate the advantage of Lasso, and more clearly that of S-TLS, in recovering the correct support.

Test Case 3: *(CR Spectrum Sensing)* This simulation is performed with reference to the CR network in the region $[0\ 1] \times [0\ 1]$ in Fig. 12. The setup includes $N_r = 4$ CRs deployed to estimate the power and location of a single source



(a) **(b)**

(c) **(d)**

Figure 14: Comparison between S-TLS and Lasso in terms of: (a) ℓ_2-norm, (b) ℓ_1-norm, and (c) ℓ_0-norm of the estimation errors; (d) Probability of detection versus probability of false alarms for the TLS, ℓ_2 regularized (R-)TLS, S-TLS and Lasso algorithms.

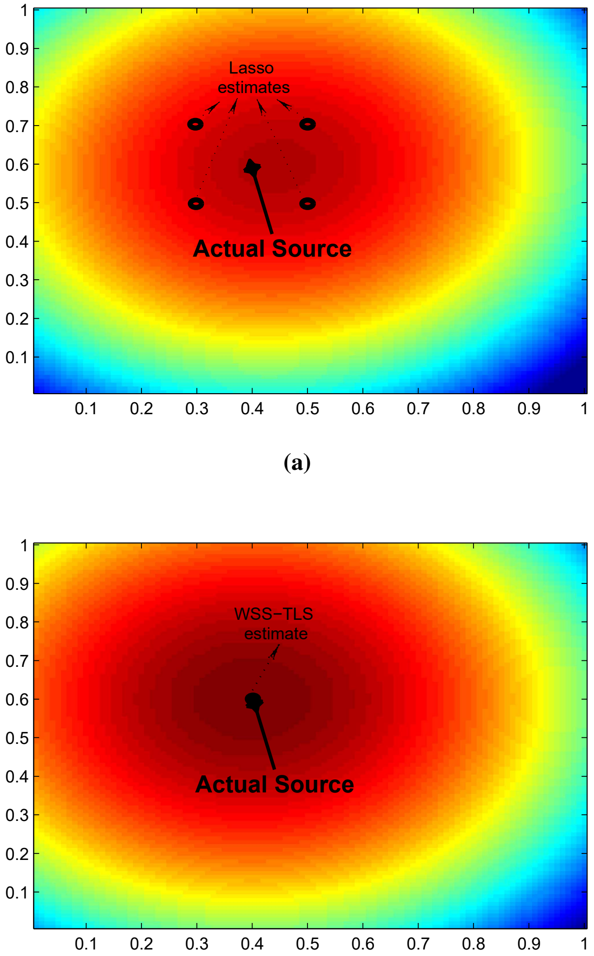
with position vector $[0.4\ 0.6]$, located at the center of four neighboring grid points. The CRs scan $K = 128$ frequencies from 15MHz to 30MHz, and adopt the basis expansion model in Section 4-A with $N_b = 16$ rectangular $b_\nu(f)$ functions, each of bandwidth 1MHz. The actual source only transmits over the $\nu = 6$-th band. The channel gains are exponentially decaying in distances with exponent $-1/2$. The received data are generated using the transmit PSD described earlier, a regular Rayleigh fading channel with 6 taps, and additive white Gaussian receiver noise at SNR=0dB. Receive-PSDs are obtained using exponentially weighted periodograms (with weight 0.99) averaged over 1,000 coherence blocks; see also [20] for more details of a related simulation. The

WSS-TLS approach is used to account for perturbations ϵ_{gr}^{s} in the channel gains. A diagonal matrix $\mathbf{W}$ is used with each diagonal entry equal to $\hat{\sigma}_{\epsilon}^{-2}$ (inversely proportional to the average of sample variances of ϵ_{gr}^{s}).

With λ chosen as in [61], both Lasso and WSS-TLS identify the active frequency band correctly (only the entries $\{x_{g6}\}_{g=1}^{16}$ were estimated as nonzero). However, Lasso identifies four transmitting sources at positions $[0.3(0.5)\ 0.5(0.7)]$, the four grid points closest to $[0.4\ 0.6]$. WSS-TLS returns only one source at position $[0.5\ 0.5]$, along with the estimated $\hat{\epsilon}_{gr}^{s}$ that yields $\hat{\gamma}_{sr} = \hat{\epsilon}_{gr}^{s} + \gamma_{gr}$. Concatenate the latter to form $\hat{\boldsymbol{\gamma}}_{s}$ of length $N_r = 4 \ll m$. Using a refined grid of 25 points uniformly spaced over the "zoom-in" region $[0.3\ 0.7] \times [0.3\ 0.7]$ centered at $[0.5\ 0.5]$, correlation coefficients between $\hat{\boldsymbol{\gamma}}_{s}$ and those of each candidate point are evaluated. The source position is estimated as the point with maximum correlation coefficient, which for WSS-TLS occurs at the true location $[0.4\ 0.6]$. To illustrate graphically the two alternatives, the estimated maps of the spatial PSDs at the 6th frequency band are plotted in Fig. 15(a) using the Lasso, and in Fig. 15(b) using WSS-TLS. The marked point indicates the actual source location $[0.4\ 0.6]$ in both maps. Unlike Lasso, the WSS-TLS identifies correctly the true position of the source.

Test Case 4: *(DoA Estimation)* The setup here entails a ULA consisting of $N_r = 8$ antenna elements with inter-element spacing $d = 1/2$, and a grid of $N_g = 90$ scanning angles from $-90°$ to $90°$ wrt the array boresight. Two sources ($N_s = 2$) of unit amplitude impinge from angles $\vartheta_s = 1°$ and $-9°$, both $1°$ off their nearest grid DoAs. As in the single-snapshot test in [201], the SNR is set to 20dB. The variance of ϵ_{g}^{s} in (63) is obtained from the uniform distribution in $[-1°, 1°]$. Selecting λ according to the noise level as in [201], Lasso returns four nonzero entries, two around each source at $\vartheta_s \pm 1°$; while WSS-TLS gives two nonzero θ_g estimates at $-10°$ ($g = 40$) and $0°$ ($g = 45$), along with perturbation estimates $\hat{\epsilon}_{40}^{s}$ and $\hat{\epsilon}_{45}^{s}$. Using the latter, the DoAs are estimated as $\hat{\vartheta}_s := \hat{\theta}_g + \hat{\epsilon}_{g}^{s}$ for $g = 40,\ 45$. The angle spectra using Lasso, and WSS-TLS with estimated $\hat{\vartheta}_s$, are compared in Fig. 16(a). The two black arrows depict the actual source angles, and benchmark the true angular spectrum.

To further illustrate the merits of WSS-TLS in estimating correctly the closest grid point and subsequently each DoA, the sample variance of a DoA estimate is plotted versus SNR in Fig. 16(b) using Monte Carlo runs, each with a single source randomly placed over $[-1°, 1°]$. Both WSS-TLS and Lasso are

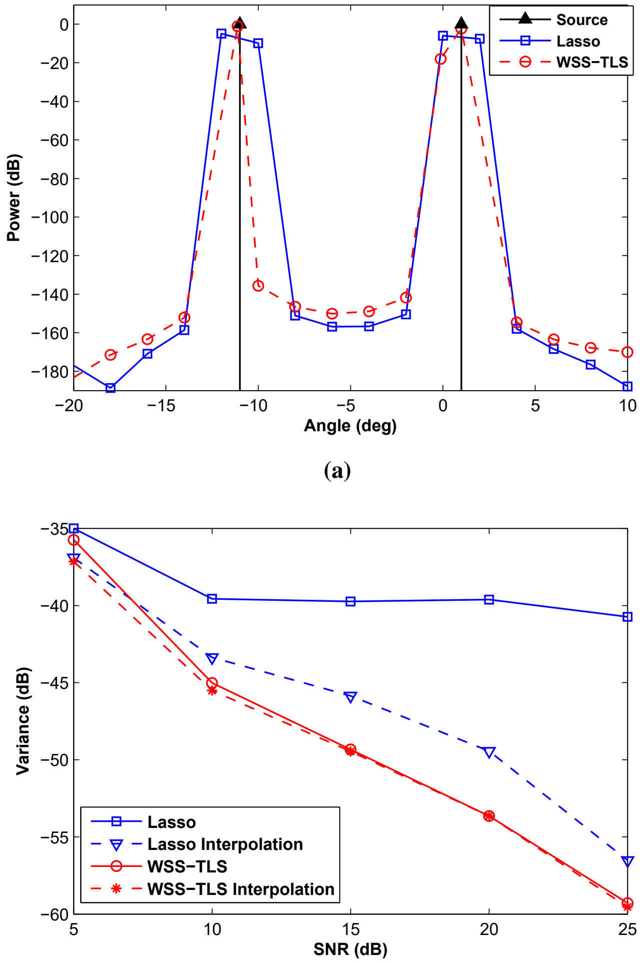


(a)

(b)

Figure 15: Comparison between PSD maps estimated by (a) Lasso, and (b) WSS-TLS for the CR network in Fig. 12.

post-processed by interpolating peaks in the obtained spectra from two nearest grid points, linearly weighted by the estimated amplitudes as in [118]. Both curves confirm that WSS-TLS outperforms the Lasso. More interestingly,



(a)

(b)

Figure 16: (a) Angular spectra estimated using Lasso and WSS-TLS as compared to the actual transmission pattern; (b) Comparison of angle estimation variances of Lasso, WSS-TLS, without and with interpolation.

the two WSS-TLS curves almost coincide, which further corroborates that WSS-TLS manages in a single step to identify correctly the support of $\mathbf{x}_{o,t}$ without requiring post processing.

Conclusions

An innovative approach was proposed to account for sparsity in estimating coefficient vectors of fully-perturbed linear regression models. This approach enriches TLS criteria that have been traditionally used to fit such models with the ability to handle under-determined linear systems. The presented S-TLS framework also enables sparsity-exploiting approaches (CS, BP, and Lasso) to cope with perturbations present not only in the data but also in the regression matrix.

Near-optimum and reduced-complexity suboptimum solvers with global and local convergence guarantees were also developed to optimize the generally nonconvex S-TLS criteria. They rely on bisection, branch-and-bound, or coordinate descent iterations, and have universal applicability regardless of whether perturbations are modeled as deterministic or random. Valuable generalizations were also provided when prior information is available on the deterministic structure or statistics of the associated (augmented) data matrix. Under specific statistical models with errors-in-variables, the resultant (generally weighted and structured) S-TLS estimators were proved to be optimal in the MAP sense. Simulated tests corroborated the analytical claims, compared competing alternatives, and demonstrated the practical impact of the presented S-TLS framework to grid-based sparsity-exploiting approaches for cognitive radio sensing, and direction-of-arrival estimation with possibly uncalibrated antenna arrays.

Part II

Compressive Data Gathering in Wireless Sensor Networks

5

Compressed Acquisition of Correlated Streaming Sensor Data

This section focuses on distributed sequential compressed acquisition and progressive reconstruction of spatially and temporally correlated sensor data streams in WSNs through CS. A sliding window based recursive CS data collection method is devised. The method uses ℓ_2-regularization and iterative reweighted ℓ_1-minimization (IRW-ℓ_1) [48] as the key techniques to incorporate prior signal information from preceding decoding instants to improve reconstruction accuracy while reducing the necessary communications. As main benefits, the method enjoys a low decoding delay, allows progressive refinement of past readings, and can trade off between the CS recovery performance and decoding complexity. The simulation results demonstrate that the proposed method achieves superior performance compared to the baseline CS methods.

Related Works

Exploiting Signal Correlation via CS

CS is an efficient means to reduce energy consumption of gathering sparse correlated signals in WSNs [94, 145, 186]. Particular attention is needed regarding the structure of a measurement matrix $\mathbf{\Phi}$; besides affecting the reconstruction performance (see Section 2), it dictates the measurement collection structure for $\mathbf{y} = \mathbf{\Phi x}$, thereby influencing the sensors' energy expenditure. Different CS data collection techniques classified based on the utilization of signal correlation are reviewed next.

Spatial Correlation: Compressive wireless sensing, where phase-coherent analog projections are sent from each sensor to a fusion center through multiple access channels, was introduced in [14]. Luo *et al.* [197] proposed a measurement matrix design to collect spatially (i.e., inter-signal) correlated data so that the nodes distant from a sink send original data, while the rest linearly combine their measurements along multi-hop routing. The minimum spanning tree based routing of the measurements was addressed in [309]. These in-network aggregation methods are highly topology-dependent and are mostly suitable for large-scale WSNs. This can be circumvented by spatial sub-sampling where the measurements are collected only from a (random) subset of sensors at each sampling instant [238, 104]. Sub-sampling via sleeping modes in distributed local signal recovery was proposed in [190], and sub-sampling combined with weighing each sensor's transmissions based on the harvested energy in [313].

Temporal Correlation: If the N readings in $\mathbf{x}$ possess temporal (i.e., intra-signal) correlations, each sensor may communicate only $M < N$ measurements $\mathbf{y} = \mathbf{\Phi x}$ to the sink for CS decoding [186]. Such an in-node technique is localized and network-independent, but it causes an inevitable decoding delay in the sensors' readings. The combination of CS and random linear network coding to efficiently acquire temporally correlated vital signals was addressed in [170].

Joint Spatio-Temporal Correlation: Besides operating in a single dimension at a time, CS has been used to acquire multi-dimensional correlated signals. There are two main classes. The first one is distributed CS [19, 95] – a unified decentralized CS framework where intra-signal and inter-signal dependencies are exploited via joint sparsity models (JSMs). The second is *Kronecker CS* (KCS) [93] which exploits general correlation structures by combining the (possibly distinct) sparsifying bases of each signal dimension into a single basis matrix [93, 248]. Distributed CS and KCS applied to various WSN signals have been empirically shown to outperform single-dimensional CS approaches in terms of compression performance and sensors' energy expenditure [37].

Dynamic CS Framework

Despite the excellence of distributed CS and KCS for distributed data acquisition, their block-wise processing neglects the inherent dynamic nature of sensor data streams. One remedy is *dynamic CS* where temporally correlated sparse signals are measured and decoded sequentially [13, 114, 284, 285, 286, 195, 318, 287]. Such streaming processing enables continually incorporating prior signal knowledge into the CS decoding to improve reconstruction accuracy. This also enables a reduction in the data block size during reconstruction, thus circumventing the complexity issues involved in distributed CS and KCS. See also Section 3 for more discussion on recursive CS frameworks.

Modified CS Reconstruction: The works in [284, 285, 286, 195] assume slowly varying signal supports and transform domain coefficients, and developed modified CS reconstruction algorithms that add regularization terms and constraints to incorporate partial support knowledge or/and signal value estimates from the previous decoding instant to improve the reconstruction performance. The works [114, 12, 13, 296] employed sliding window processing to exploit successively decoded overlapping signal portions either to improve the reconstruction fidelity or to speed up the iterative signal recovery. The reconstruction accuracy and stability over time of such recursive algorithms are affected for example by temporal signal characteristics, algorithm initialization, and intermittent signal information updates [286, 318]. For other applications on recursive CS, see, e.g., [200, 291, 67].

Weighted-Norm Minimization: The intrinsic prior information in dynamic CS can also be utilized via the *iterative reweighted ℓ_1-minimization* (IRW-ℓ_1) [48, 223, 302, 159, 312, 12, 13]. The core idea of the IRW-ℓ_1 is to suppress the magnitude dependency of the ℓ_1-norm so that the reweighted ℓ_1-minimization becomes closer to the sparsity-optimal ℓ_0-minimization [48]. The IRW-ℓ_1 alternates between solving a weighted ℓ_1-minimization problem and updating the weights based on the current solution [48]. While the IRW-ℓ_1 has mainly been applied for single finite-length signals, the work in [13] adapts the technique to a recursive CS framework. In such a setup, the estimates from the preceding decoding window can be sophisticatedly used for weight initialization. Other variants aiming at improving the performance of conventional ℓ_1-minimization include iteratively reweighted least squares [214, 52, 302], and non-convex ℓ_p-minimization for $0 < p < 1$ [59, 60].

System Model

Consider a multi-hop single-sink data gathering WSN which consists of a set of battery-powered sensors $\mathcal{N} = \{1, \ldots, N\}$, capable of transmitting, receiving, and relaying data. The sensors are deployed in an event area to periodically monitor a physical phenomenon (e.g., light intensity, wind speed, temperature, humidity etc.) at a pre-defined rate and to disseminate the acquired information to the sink. For each sensing period, the sink is responsible for obtaining an accurate reconstruction of the monitored field, i.e., it recovers the readings of all N sensors.

Source Model

The sensor observations are assumed to encompass both spatial and temporal correlations, accounting for various typical environmental sensing applications in densely deployed WSNs [5, 197]. The prevailing physical phenomenon in the monitoring field is generated by a set of independent, randomly located, time-varying sources $\mathcal{S} = \{1, \ldots, S\}$. These sources could be, e.g., server racks representing heat sources during data center temperature monitoring, or lamps corresponding to light sources in an automated illumination control system. A data gathering WSN with $N = 8$ and $S = 5$ is illustrated in Fig. 17. The impact of each source $s \in \mathcal{S}$ on the sensors' readings is modeled through time-invariant, real-valued influence functions $h_{ns} \in \mathbb{R}$, $n \in \mathcal{N}$. Let $x_n(t)$ and $\beta_s(t)$ denote the reading of sensor $n \in \mathcal{N}$ and the magnitude of source $s \in \mathcal{S}$ at time instant t, respectively. The sources are assumed to affect the sensor observations collectively. Thus, a sensor reading $x_n(t)$ is given by the superposition of the influences from all sources as [190]

$$x_n(t) = \textstyle\sum_{s \in \mathcal{S}} h_{ns}\beta_s(t) = \mathbf{h}_n^{\mathrm{T}}\boldsymbol{\beta}(t), \; n \in \mathcal{N}, \; t = 1, 2, \ldots, \qquad (64)$$

where vector $\boldsymbol{\beta}(t) = [\beta_1(t) \; \cdots \; \beta_S(t)]^{\mathrm{T}}$ consists of the source magnitudes at time instant t, and $\mathbf{h}_n = [h_{n1} \; \cdots \; h_{nS}]^{\mathrm{T}}$ is the aggregate influence vector on sensor $n \in \mathcal{N}$.

According to (64), the spatial and temporal correlation properties of the sensors' readings are separately characterized by the types of influence functions and the stochastic properties of the source magnitude sequences, respectively. Without a loss of generality, a common class of distance-dependent influences

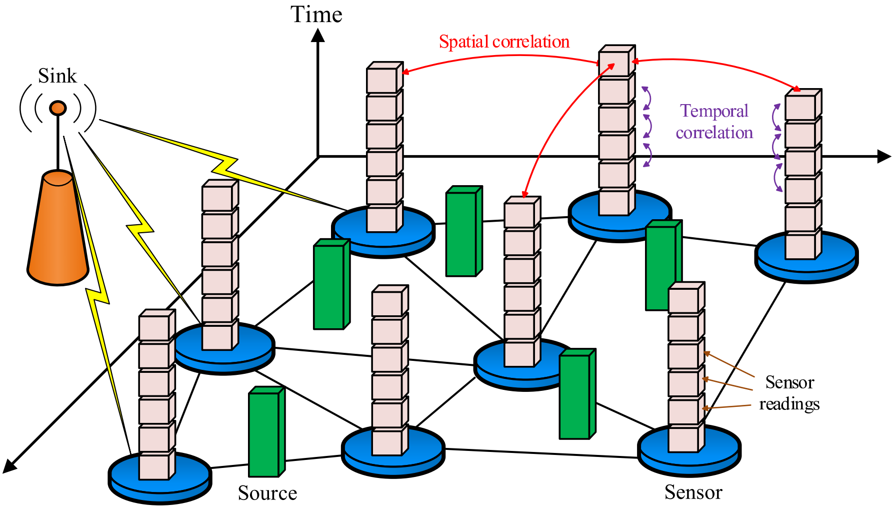


Figure 17: A data gathering WSN with spatially and temporally correlated sensor data streams for $N = 8$ and $S = 5$.

[24, 5, 330] of the form $h_{ns}(d_{ns})$ is used, where d_{ns} is the distance between sensor $n \in \mathcal{N}$ and source $s \in \mathcal{S}$. Moreover, it is assumed that the evolution of source sequences $\{\beta_s(1), \beta_s(2), \ldots\}$, $s \in \mathcal{S}$, is *piecewise smooth*. This family of models covers a general WSN monitoring framework, where the readings of the sensors close to each other encompass a high degree of spatial correlation and where occasional abrupt changes can occur on top of slow temporal variations in the underlying phenomenon. In fact, the data recorded from monitoring the temperature, humidity, and solar radiation in real WSN test beds have been reported to encounter high degrees of temporal correlation with slow magnitude variations [196, 197, 330, 238].

Energy Consumption Model

Since the wireless data transmissions are one of the most energy-demanding operations for a sensor, the amount of communications needed to report the observed readings (64) to the sink should be minimized. For assessing the energy consumption of wireless access, we use a customary approach of CS data gathering works [238]: the transmit power usage of a sensor is quantified in proportion to the number of transmitted measurements (this will become more clear in Section 5). Although this model does not take into account the details of the physical layer (i.e., coding, modulation, channel capacity), the data link layer (i.e., medium access control), and the network layer (i.e., data routing and forwarding schemes), the number of transmitted data units serves as a legitimate quantity for the sensors' energy consumption.

Sequential CS of Correlated Sensor Data Streams

In this section, we present a sequential CS framework for delivering jointly correlated sensor data streams, i.e., the observed sensor readings $x_n(t)$ of (64), to the sink with an aim of minimizing the necessary number of transmitted data units. The sensors' readings are periodically delivered to the sink in the form of measurements, which are used to sequentially reconstruct portions of the sensor data streams via sliding window processing while exploiting the joint compressibility using Kronecker sparsifying bases. The method produces estimates for the current sensors' readings without an additional decoding delay and flexibly implements trade-offs between the CS recovery performance and decoding complexity via the window size.

Sliding Data Window

Let $\mathbf{X}(t) \in \mathbb{R}^{N \times W}$ denote a *data window* at time instant t with a window size[2] $W \geq 1$. The matrix $\mathbf{X}(t)$, $t \geq W$, consists of W consecutive readings of all N sensors at time instants $\{t - W + 1, \ldots, t\}$ as

$$\mathbf{X}(t) = \begin{bmatrix} x_1(t-W+1) & \cdots & x_1(t) \\ \vdots & \ddots & \vdots \\ x_N(t-W+1) & \cdots & x_N(t) \end{bmatrix}, \tag{65}$$

where $x_n(t)$ is the reading of sensor $n \in \mathcal{N}$ at time instant t given in (64). Let vector $\mathbf{x}_n^{\mathrm{T}}(t) = [x_n(t-W+1) \cdots x_n(t)]$ denote the nth row of $\mathbf{X}(t)$, containing the readings of sensor n at time instants $\{t - W + 1, \ldots, t\}$. Similarly, let vector $\underline{\mathbf{x}}(t) = [x_1(t) \cdots x_N(t)]^{\mathrm{T}}$ denote the columns of $\mathbf{X}(t)$, each of which contains the sensors' readings at a time instant t. Accordingly, $\mathbf{X}(t)$

[2] A fixed window length W is assumed for clarity; using a varying window size over the time slots could be beneficial in certain applications.

can be expressed as

$$\mathbf{X}(t) = [\underline{\mathbf{x}}(t-W+1) \cdots \underline{\mathbf{x}}(t)] = [\mathbf{x}_1(t) \cdots \mathbf{x}_N(t)]^{\mathrm{T}}. \tag{66}$$

The sliding window structure with respect to the sensor data streams is illustrated in Fig. 18(a).

Exploiting Joint Spatio-Temporal Correlation

Recall that according to the sensor data model defined in Section 5, each data window $\mathbf{X}(t)$ in (65) encompasses both a spatial and temporal correlation. We assume that there exists a basis $\boldsymbol{\Psi}_{\mathrm{S}} \in \mathbb{R}^{N\times N}$ for the spatial domain in which each column of $\mathbf{X}(t)$ has a compressible representation, i.e., $\underline{\mathbf{x}}(t) = \boldsymbol{\Psi}_{\mathrm{S}}\boldsymbol{\theta}_{\mathrm{S}}(t)$, where vector $\boldsymbol{\theta}_{\mathrm{S}}(t) \in \mathbb{R}^N$ contains the spatial transform coefficients at slot t. Accordingly, $\mathbf{X}(t)$ can be written as

$$\begin{aligned}\mathbf{X}(t) &= [\underline{\mathbf{x}}(t-W+1) \cdots \underline{\mathbf{x}}(t)] \\ &= \boldsymbol{\Psi}_{\mathrm{S}}\,[\boldsymbol{\theta}_{\mathrm{S}}(t-W+1) \cdots \boldsymbol{\theta}_{\mathrm{S}}(t)] \\ &= \boldsymbol{\Psi}_{\mathrm{S}}\boldsymbol{\Theta}_{\mathrm{S}}(t),\end{aligned} \tag{67}$$

where $\boldsymbol{\Theta}_{\mathrm{S}}(t) \triangleq [\boldsymbol{\theta}_{\mathrm{S}}(t-W+1) \cdots \boldsymbol{\theta}_{\mathrm{S}}(t)]$. Similarly, there exists a temporal domain basis $\boldsymbol{\Psi}_{\mathrm{T}} \in \mathbb{R}^{W\times W}$ in which each row of $\mathbf{X}(t)$ has a compressible representation, i.e., $\mathbf{x}_n(t) = \boldsymbol{\Psi}_{\mathrm{T}}\boldsymbol{\theta}_{\mathrm{T},n}(t)$, where vector $\boldsymbol{\theta}_{\mathrm{T},n}(t) \in \mathbb{R}^W$ contains the temporal transform coefficients of sensor n. Hence, $\mathbf{X}(t)$ can be expressed as

$$\begin{aligned}\mathbf{X}^{\mathrm{T}}(t) &= [\mathbf{x}_1(t) \cdots \mathbf{x}_N(t)] \\ &= \boldsymbol{\Psi}_{\mathrm{T}}\,[\boldsymbol{\theta}_{\mathrm{T},1}(t) \cdots \boldsymbol{\theta}_{\mathrm{T},N}(t)] \\ &= \boldsymbol{\Psi}_{\mathrm{T}}\boldsymbol{\Theta}_{\mathrm{T}}(t),\end{aligned} \tag{68}$$

where $\boldsymbol{\Theta}_{\mathrm{T}}(t) \triangleq [\boldsymbol{\theta}_{\mathrm{T},1}(t) \cdots \boldsymbol{\theta}_{\mathrm{T},N}(t)]$.
Kronecker sparsifying bases can succinctly combine the individual sparsifying bases of each signal dimension into a single transformation matrix [92, 93]. Thus, by merging the transformations in (67) and (68), $\mathbf{X}(t)$ can be represented as

$$\begin{aligned}\mathbf{x}(t) = \mathrm{vec}\,(\mathbf{X}(t)) &= \mathrm{vec}\,(\boldsymbol{\Psi}_{\mathrm{S}}\mathbf{Z}(t)\boldsymbol{\Psi}_{\mathrm{T}}^{\mathrm{T}}) \\ &= (\boldsymbol{\Psi}_{\mathrm{T}}\otimes\boldsymbol{\Psi}_{\mathrm{S}})\mathrm{vec}\,(\mathbf{Z}(t)) \\ &= \boldsymbol{\Psi}\mathbf{z}(t),\end{aligned} \tag{69}$$

where $\mathbf{x}(t) = [\underline{\mathbf{x}}^{\mathrm{T}}(t-W+1) \cdots \underline{\mathbf{x}}^{\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^{NW}$ is the vector-reshaped data window, the operator $\mathrm{vec}(\mathbf{A})$ stacks the columns of matrix

$\mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_N]$ into the vector form $\mathbf{a} = [\mathbf{a}_1^{\mathrm{T}} \cdots \mathbf{a}_N^{\mathrm{T}}]^{\mathrm{T}}$, $\mathbf{Z}(t) \in \mathbb{R}^{N \times W}$ is a matrix of the joint transform coefficients for $\mathbf{X}(t)$, $\otimes$ denotes the Kronecker product, $\mathbf{\Psi} = (\mathbf{\Psi}_{\mathrm{T}} \otimes \mathbf{\Psi}_{\mathrm{S}}) \in \mathbb{R}^{NW \times NW}$ is the Kronecker sparsifying basis, and $\mathbf{z}(t) = \mathrm{vec}\left(\mathbf{Z}(t)\right) \in \mathbb{R}^{NW}$ is the joint transform coefficient vector. In summary, $\mathbf{X}(t)$ has a 2D-separable transform $\mathbf{Z}(t) = \mathbf{\Psi}_{\mathrm{S}}^{-1} \mathbf{X}(t) \mathbf{\Psi}_{\mathrm{T}}^{-\mathrm{T}}$, where $\mathbf{\Psi}_{\mathrm{S}}^{-1}$ operates on the columns of $\mathbf{X}(t)$ and $\mathbf{\Psi}_{\mathrm{T}}^{-\mathrm{T}}$ on its rows [248].

Data-independent bases $\mathbf{\Psi}_{\mathrm{S}}$ and $\mathbf{\Psi}_{\mathrm{T}}$ are considered in this work. Such universal transformations are suitable for revealing the underlying sparsity of many smooth/piecewise smooth signals and include the discrete Fourier, cosine, and wavelet transform (DFT, DCT, DWT), respectively [202, 15, 36, 37]. The efficacy of the DWT matrices in sparsifying signals of various natural phenomena (e.g., temperature, humidity, and light intensity) has been reported in, e.g., [197, 37]. Other transformations include data-dependent basis learning via principal component analysis (PCA) [210, 238], and topology-dependent diffusion and graph wavelets [145].

CS Encoding and Decoding

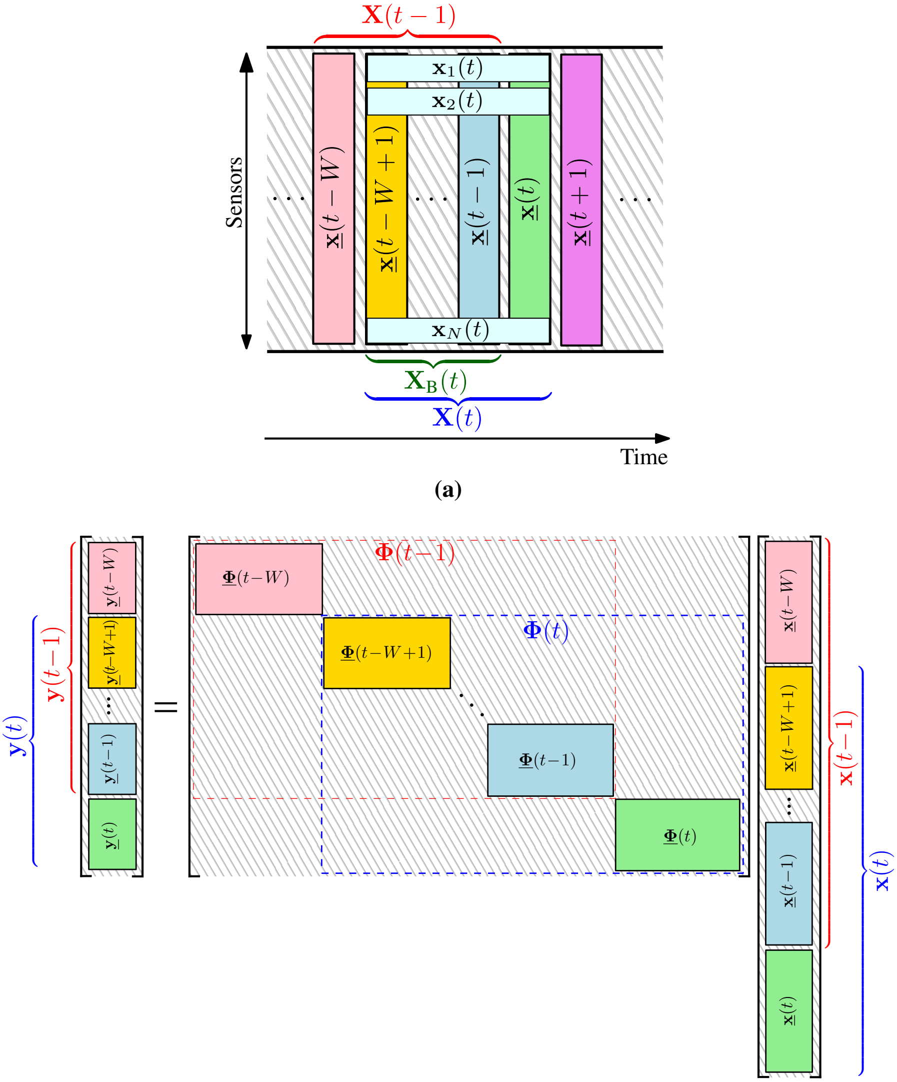
Next, we present the CS encoding and decoding processes in the data gathering framework.

CS Encoding

We consider a CS encoding procedure, where at each time instant $t \geq 1$, the measurements are taken with respect to the current sensors' readings $\underline{\mathbf{x}}(t)$. Accordingly, the sink acquires $M(t)$ linear compressive measurements $\underline{\mathbf{y}}(t) = [\underline{y}_1(t) \cdots \underline{y}_{M(t)}(t)]^{\mathrm{T}} \in \mathbb{R}^{M(t)}$ as

$$\underline{\mathbf{y}}(t) = \underline{\mathbf{\Phi}}(t) \underline{\mathbf{x}}(t), \; t \geq 1, \tag{70}$$

where $\underline{\mathbf{\Phi}}(t) \in \mathbb{R}^{M(t) \times N}$, $M(t) \leq N$, is the measurement matrix for time instant t. According to (70), the measurement ensemble with respect to each


$\mathbf{X}(t-1)$
$\mathbf{x}_1(t)$
$\mathbf{x}_2(t)$
Sensors
$\cdots$
$\underline{\mathbf{x}}(t-W)$
$\underline{\mathbf{x}}(t-W+1)$
$\cdots$
$\underline{\mathbf{x}}(t-1)$
$\underline{\mathbf{x}}(t)$
$\underline{\mathbf{x}}(t+1)$
$\cdots$
$\mathbf{x}_N(t)$
$\mathbf{X}_\mathrm{B}(t)$
$\mathbf{X}(t)$
Time
$\mathbf{y}(t)$
$\mathbf{y}(t-1)$
$\underline{\mathbf{y}}(t-W)$
$\underline{\mathbf{y}}(t-W+1)$
$\vdots$
$\underline{\mathbf{y}}(t-1)$
$\underline{\mathbf{y}}(t)$
=
$\mathbf{\Phi}(t-1)$
$\underline{\mathbf{\Phi}}(t-W)$
$\mathbf{\Phi}(t)$
$\underline{\mathbf{\Phi}}(t-W+1)$
$\ddots$
$\underline{\mathbf{\Phi}}(t-1)$
$\underline{\mathbf{\Phi}}(t)$
$\underline{\mathbf{x}}(t-W)$
$\underline{\mathbf{x}}(t-W+1)$
$\vdots$
$\underline{\mathbf{x}}(t-1)$
$\underline{\mathbf{x}}(t)$
$\mathbf{x}(t-1)$
$\mathbf{x}(t)$

(a)

(b)

Figure 18: A block diagram to illustrate the sliding window processing with respect to (a) the sensor data streams, and (b) measurement ensembles.

data window $\mathbf{X}(t)$ has a block-diagonal structure[3] as

$$\begin{bmatrix} \underline{\mathbf{y}}(t-W+1) \\ \vdots \\ \underline{\mathbf{y}}(t) \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{\Phi}}(t-W+1) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \underline{\mathbf{\Phi}}(t) \end{bmatrix} \begin{bmatrix} \underline{\mathbf{x}}(t-W+1) \\ \vdots \\ \underline{\mathbf{x}}(t) \end{bmatrix}. \tag{71}$$

[3] For a line of work with overlapping measurement systems, see, e.g., [114, 13].

By forming the concatenated measurement vector $\mathbf{y}(t) = \left[\underline{\mathbf{y}}^{\mathrm{T}}(t-W+1) \cdots \underline{\mathbf{y}}^{\mathrm{T}}(t)\right]^{\mathrm{T}} \in \mathbb{R}^{\sum_{\tau=t-W+1}^{t} M(\tau)}$ and the corresponding measurement matrix $\mathbf{\Phi}(t) = \operatorname{diag}\left(\underline{\mathbf{\Phi}}(t-W+1), \ldots, \underline{\mathbf{\Phi}}(t)\right) \in \mathbb{R}^{\sum_{\tau=t-W+1}^{t} M(\tau) \times NW}$, the measurement ensemble in (71) can be compactly written as

$$\mathbf{y}(t) = \mathbf{\Phi}(t)\mathbf{x}(t). \tag{72}$$

Note that the measurement matrices $\underline{\mathbf{\Phi}}(t) \in \mathbb{R}^{M(t) \times N}$, $t = 1, 2, \ldots,$ are in general unique, i.e., they can have different structures and varying numbers of measurements $M(t)$. Measurement ensembles with respect to successive data windows are depicted in Fig. 18(b).

Delivery of Measurements

As the structure of each $\underline{\mathbf{\Phi}}(t)$ dictates how the measurements in (70) are delivered to the sink, each $\underline{\mathbf{\Phi}}(t)$ has a great impact on the sensors' energy expenditure via the required wireless transmissions across the WSN [4]. Hence, to achieve efficient and viable measurement acquisition scheme for (70), the following sparse binary matrix $\underline{\mathbf{\Phi}}(t) \in \mathbb{B}^{M(t) \times N}$ is used [210, 238]: all its entries are zeros, except for a single "1" in each row, and at most a single ”1” in each column. Consequentially, at each data gathering period t, the sink receives $M(t)$ readings from a random subset of sensors, denoted by $\mathcal{N}(t) \subseteq \mathcal{N}$, $|\mathcal{N}(t)| = M(t)$, i.e., the set of reading values becomes $\{x_n(t) | n \in \mathcal{N}(t)\}$. This scheme has been empirically shown to greatly lower communication costs in data gathering WSNs, while allowing fast and efficient implementation of matrix multiplications in the CS decoding [204, 37].

The considered *sub-sampling* type $\underline{\mathbf{\Phi}}(t)$, and consequently $\mathbf{\Phi}(t)$, fall into the class of binary sparse measurement matrices [292, 28, 27, 173, 172, 237, 84, 37, 204], which are capable of achieving CS recovery performance comparable to that of dense ones. The performance is affected by the mutual coherence between $\mathbf{\Phi}(t)$ and $\mathbf{\Psi}$, and by the RIP-p property – a weaker form of the RIP introduced in [27]. Analyses and empirical studies on the mutual coherence between binary sparse measurement matrices and different sparsifying bases can be found in, e.g., [237, 204, 84]. Intuitively, as the projections via the sub-sampling $\mathbf{\Phi}(t)$ become highly localized, successful CS decoding necessitates that the energy of the readings (64) is sufficiently dispersed across the sensors, i.e., $\mathbf{\Psi}$ is dense [173, 172, 84]. Fortunately, this is the case in typical WSN applications where the correlated data has a compressible representation under a transformation like DCT or DWT.

Furthermore, since the underlying physical phenomena may be geographically localized in WSNs, the measurements (70) should have a sufficient spatial distribution/resolution to capture enough signal information for successful CS recovery. This can be achieved by a distributed random sampling scheme [238], where at each time instant t, each sensor $n \in \mathcal{N}$ independently decides to transmit its reading $x_n(t)$ (along with a time stamp t and node index n) with the probability[4] $p_t^{\mathrm{tx}} \in [0, 1]$. Consequently, for each t, on average $p_t^{\mathrm{tx}} N$ sensors communicate the readings to the sink. The scheme is desirable in WSNs because the routing becomes simple and topology-independent with a low communication overhead. Moreover, it can be realized with standard multi-hop routing protocols, including asynchronous individual transmissions and packet aggregation techniques [94, 238]. A sensor may also switch into a sleep mode, if during a data gathering round t, the sensor does not belong to $\mathcal{N}(t)$ and does not act as a relaying node for multi-hop packet forwarding.

CS Decoding

By exploiting the *joint spatio-temporal* compressibility (69), each data window $\mathbf{X}(t)$ can be recovered from measurements (72) by solving the ℓ_1-minimization problem (cf. (3))

$$\hat{\mathbf{z}}(t) := \underset{\tilde{\mathbf{z}}}{\operatorname{argmin}} \, \|\tilde{\mathbf{z}}\|_1 \quad \text{such that} \quad \mathbf{y}(t) = \mathbf{\Phi}(t)\mathbf{\Psi}\tilde{\mathbf{z}}, \tag{73}$$

where $\hat{\mathbf{z}}(t)$ is an estimate of $\mathbf{z}(t)$ so that an estimate of $\mathbf{x}(t)$ is $\hat{\mathbf{x}}(t) = \mathbf{\Psi}\hat{\mathbf{z}}(t)$. Reshaping the decoded data window $\hat{\mathbf{x}}(t)$ gives $\hat{\mathbf{X}}(t) = [\underline{\hat{\mathbf{x}}}(t - W + 1) \cdots \underline{\hat{\mathbf{x}}}(t)]$, where $\underline{\hat{\mathbf{x}}}(t) = [\hat{x}_1(t) \cdots \hat{x}_N(t)]^{\mathrm{T}}$ contains the estimates of the sensors' readings $x_n(t)$ of (64).

Each decoding instant (73) produces estimates for the current sensors' readings[5] and the $W - 1$ previous ones. Note that for $W = 1$, (73) reduces to

[4] The classic CS sampling process relies on non-adaptive incoherent measurements, i.e., each measurement has an equal priority [94]. For a line of work on CS with adaptive measurements, see, e.g., [147].

[5] We assume a *slotted* sliding window processing, i.e., at each time slot t, $\underline{\mathbf{x}}(t)$ is referred to as the current sensors' readings which are reconstructed by (73) within the same slot.

reconstructing each $\underline{\mathbf{x}}(t)$ *separately* from measurements (70) by utilizing only the spatial domain compressibility (67), i.e., solving

$$\hat{\boldsymbol{\theta}}_{\mathrm{S}}(t) := \underset{\tilde{\boldsymbol{\theta}}_{\mathrm{S}}}{\operatorname{argmin}} \|\tilde{\boldsymbol{\theta}}_{\mathrm{S}}\|_1 \quad \text{such that} \quad \underline{\mathbf{y}}(t) = \underline{\boldsymbol{\Phi}}(t)\boldsymbol{\Psi}_{\mathrm{S}}\tilde{\boldsymbol{\theta}}_{\mathrm{S}}, \tag{74}$$

and reconstructing $\hat{\underline{\mathbf{x}}}(t) = \boldsymbol{\Psi}_{\mathrm{S}}\hat{\boldsymbol{\theta}}_{\mathrm{S}}(t)$.

Remark 5.1. *The encoding/decoding in* (73) *without streaming processing constitutes a Kronecker CS (KCS) scheme [93]: a single data window* $\mathbf{X}(t)$ *is reconstructed from the associated measurement ensemble by treating* $\mathbf{X}(t)$ *as a finite-length signal. In general, a large* W *is preferable for the KCS as it allows to utilize the temporal domain compressibility of* $\mathbf{X}(t)$ *over a longer interval improving the signal recovery performance. However, this induces two prominent drawbacks: 1) involving* W *consecutive sensors' readings in the encoding induces a* decoding delay *proportional to* W *prior to obtaining estimates for all the involved readings, and 2) the decoding complexity of* (73) *can grow excessively high.*

Progressive CS Signal Reconstruction With Prior Information

This section presents the development of a sequential CS method. Differently from the KCS that processes the data in non-overlapping blocks, the proposed method treats the sensor data as *continuous-time streams*. This philosophy is inspired by the fact that at each time instant $t \geq W$, (73) utilizes the measurements and joint compressibility associated with the current sensor reading vector $\underline{\mathbf{x}}(t)$, and the $W-1$ past ones $\underline{\mathbf{x}}(t-W+1), \ldots, \underline{\mathbf{x}}(t-1)$. The streaming processing mitigates the disadvantages of the KCS by 1) eliminating the decoding delay to obtain the current sensors' readings, and 2) allowing a trade-off between the decoding complexity and the CS recovery performance by adjusting W. These benefits are illustrated by the numerical results in Section 5.

Modified CS Reconstruction Problem

For window sizes $W > 1$, the sensors' readings $\underline{\mathbf{x}}(t)$ reappear in the W consecutive data windows $\mathbf{X}(t), \ldots, \mathbf{X}(t+W-1)$ (see (66) and Fig. 18(a)). Hence, as $\underline{\mathbf{x}}(t)$ is involved in the W successive measurement vectors

$\mathbf{y}(t), \ldots, \mathbf{y}(t+W-1)$ by (71), it will be decoded W times via (73). Due to this, a recursive CS recovery method is derived by modifying (73) so that, at each time instant $t \geq W$, it utilizes the previously decoded estimates of $\underline{\mathbf{x}}(t-W+1), \ldots, \underline{\mathbf{x}}(t-1)$ to facilitate the reconstruction of the current sensors' readings $\underline{\mathbf{x}}(t)$. Moreover, this improves the reconstruction accuracy of the past readings $\underline{\mathbf{x}}(t-W+1), \ldots, \underline{\mathbf{x}}(t-1)$. As demonstrated in Section 5, the developed method substantially reduces the amount of necessary sensor communications. Albeit beyond the scope of this work, the overlapping signal portions could further be utilized to develop iterative warm-start based recovery algorithms with fast convergence [114, 12, 13, 296].

Decoder Buffer

Let $\mathbf{X}_{\mathrm{B}}(t) \in \mathbb{R}^{N \times (W-1)}$ denote the first $W-1$ columns of data window $\mathbf{X}(t)$ (See Fig. 18(a)). Thus, at time instant $t \geq W$, it consists of the $W-1$ previous sensors' readings, i.e., $\mathbf{X}_{\mathrm{B}}(t) = [\underline{\mathbf{x}}(t-W+1) \cdots \underline{\mathbf{x}}(t-1)]$. Accordingly, let $\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t) \in \mathbb{R}^{N \times (W-1)}$ denote a *decoder buffer* at time instant t, which consists of the estimates of $\mathbf{X}_{\mathrm{B}}(t)$ obtained at the previous instant $t-1$, i.e.,

$$\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t) = \left[\hat{\underline{\mathbf{x}}}^{(t-1)}(t-W+1) \cdots \hat{\underline{\mathbf{x}}}^{(t-1)}(t-1)\right], \tag{75}$$

where each vector $\hat{\underline{\mathbf{x}}}^{(t-1)}(t-d) = \left[\hat{x}_1^{(t-1)}(t-d) \cdots \hat{x}_N^{(t-1)}(t-d)\right]^{\mathrm{T}}$, $d = 1, \ldots, W-1$, contains the estimates of $\underline{\mathbf{x}}(t-d)$ obtained at the decoding instant $t-1$, i.e., $\hat{x}_n^{(t-1)}(t-d)$ denotes the estimate of sensor reading $x_n(t-d)$ of (64) obtained at decoding instant $t-1$. In other words, the last column of $\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)$ stores the estimates of the preceding sensors' readings, whereas its first column contains the estimates of the most outdated ones.

Problem Formulation

Recall that the CS decoding problems (73) at consecutive time instants $t-1$ and t reconstruct the data windows $\mathbf{X}(t-1) = [\underline{\mathbf{x}}(t-W) \;\; \mathbf{X}_{\mathrm{B}}(t)]$ and $\mathbf{X}(t) = [\mathbf{X}_{\mathrm{B}}(t) \;\; \underline{\mathbf{x}}(t)]$, respectively. Owing to the fact that they share the common signal part $\mathbf{X}_{\mathrm{B}}(t)$ (see Fig. 18(a)), the recovery problem (73) is next modified such that, at each time instant $t \geq W$, it utilizes the previous estimate of $\mathbf{X}_{\mathrm{B}}(t)$, i.e., the decoder buffer $\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)$ in (75), to reconstruct $\mathbf{X}(t)$. To this end, the objective function of (73) will be augmented with a regularization term which induces an additional penalty in relation to $\left\|\hat{\mathbf{X}}_{\mathrm{B}}^{(t)}(t)-\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)\right\|_{\mathrm{F}}$, i.e., the deviation between the estimates of $\mathbf{X}_{\mathrm{B}}(t)$ obtained at the consecutive decoding instants t and $t-1$.

Let $\mathbf{\Psi}_{\mathrm{T}}' \in \mathbb{R}^{(W-1)\times W}$ denote the matrix consisting of the first $W-1$ rows of temporal domain basis $\mathbf{\Psi}_{\mathrm{T}}$ in (68), i.e., $\mathbf{\Psi}_{\mathrm{T}}' = [\boldsymbol{\psi}_{\mathrm{T},1} \cdots \boldsymbol{\psi}_{\mathrm{T},(W-1)}]^{\mathrm{T}}$, where $\boldsymbol{\psi}_{\mathrm{T},i} = [\psi_{\mathrm{T},i1} \cdots \psi_{\mathrm{T},iW}]^{\mathrm{T}}$ is the ith row of $\mathbf{\Psi}_{\mathrm{T}}$, $i=1,\ldots,W-1$. Then, let us form a matrix $\mathbf{\Psi}' \in \mathbb{R}^{N(W-1)\times NW}$ as

$$\begin{aligned}\mathbf{\Psi}' &= \mathbf{\Psi}_{\mathrm{T}}' \otimes \mathbf{\Psi}_{\mathrm{S}} \\ &= [\boldsymbol{\psi}_1 \cdots \boldsymbol{\psi}_{N(W-1)}]^{\mathrm{T}},\end{aligned} \tag{76}$$

which extracts the first $N(W-1)$ rows of Kronecker sparsifying basis $\mathbf{\Psi}$ in (69), where $\boldsymbol{\psi}_i = [\psi_{i1} \cdots \psi_{i(NW)}]^{\mathrm{T}}$ is the ith row of $\mathbf{\Psi}$, $i=1,\ldots,N(W-1)$ (See Appendix B for more details).

Regularization: Using $\mathbf{\Psi}'$ in (76) and $\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)$ in (75), the following regularization term is added in the objective function of (73):

$$\gamma_{\mathrm{B}}\left\|\mathbf{\Psi}'\tilde{\mathbf{z}} - \mathrm{vec}\left(\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)\right)\right\|_2, \tag{77}$$

where $\gamma_{\mathrm{B}} \geq 0$ is a non-negative regularization weight parameter, and $\tilde{\mathbf{z}} \in \mathbb{R}^{NW}$ are the optimization variables. The first term in (77), which, according to (76) is equal to $\mathbf{\Psi}'\tilde{\mathbf{z}} = [\boldsymbol{\psi}_1 \cdots \boldsymbol{\psi}_{N(W-1)}]^{\mathrm{T}}\tilde{\mathbf{z}}$, carries the following logic: by solving (73) with added regularization (77), the term $\mathbf{\Psi}'\tilde{\mathbf{z}}$ constitutes an estimate of $\mathrm{vec}\left(\mathbf{X}_{\mathrm{B}}(t)\right)$ at time instant t, i.e., $\mathbf{\Psi}'\tilde{\mathbf{z}} \triangleq \mathrm{vec}\left(\hat{\mathbf{X}}_{\mathrm{B}}^{(t)}(t)\right)$. Therefore, the regularization term can be interpreted as $\gamma_{\mathrm{B}}\left\|\mathrm{vec}\left(\hat{\mathbf{X}}_{\mathrm{B}}^{(t)}(t)\right) - \mathrm{vec}\left(\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)\right)\right\|_2$, i.e., it assigns a cost for the deviation between the consecutive estimates of $\mathbf{X}_{\mathrm{B}}(t)$. The weight parameter γ_{B} controls the emphasis between the regularization term and the sparsity-promoting ℓ_1-norm term.

IRW-ℓ_1: In addition to the ℓ_2-regularization, the iterative reweighted ℓ_1-minimization (IRW-ℓ_1) algorithm [48] is adapted to problem (73). Accordingly, at each decoding instant t, the IRW-ℓ_1 alternates between solving a weighted ℓ_1-minimization problem, and updating the weights based on the obtained solution [48]. Specifically, $\|\tilde{\mathbf{z}}\|_1$ in (73) is replaced with $\left\|\mathbf{G}^{(k)}(t)\tilde{\mathbf{z}}\right\|_1$, where $\mathbf{G}^{(k)}(t) = \mathrm{diag}\left(g_1^{(k)}(t),\ldots,g_{NW}^{(k)}(t)\right)$ is a diagonal weight matrix at decoding instant t and iteration k with positive weights $g_i^{(k)}(t) > 0$, $i=1,\ldots,NW$.

By combining the IRW-ℓ_1 with the ℓ_2-regularization term (77), at each decoding instant $t \geq W$ and iteration k, the decoder first solves a *modified CS recovery problem*

$$\begin{aligned} \hat{\mathbf{z}}^{(k)}(t) := & \underset{\tilde{\mathbf{z}}}{\operatorname{argmin}} \left\{ \left\| \mathbf{G}^{(k)}(t)\tilde{\mathbf{z}} \right\|_1 + \gamma_{\mathrm{B}} \left\| \mathbf{\Psi}'\tilde{\mathbf{z}} - \hat{\mathbf{x}}_{\mathrm{B}}^{(t-1)}(t) \right\|_2 \right\} \\ & \text{such that} \quad \mathbf{y}(t) = \mathbf{\Phi}(t)\mathbf{\Psi}\tilde{\mathbf{z}} \end{aligned} \tag{78}$$

to obtain an estimate of $\mathbf{z}(t)$ (See (69)), where $\hat{\mathbf{x}}_{\mathrm{B}}^{(t-1)}(t) = \operatorname{vec}\left(\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)\right)$ is the vector-reshaped decoder buffer of (75). Then, the decoder uses the obtained estimate vector $\hat{\mathbf{z}}^{(k)}(t) = \left[\hat{z}_1^{(k)}(t) \ \cdots \ \hat{z}_{NW}^{(k)}(t)\right]^{\mathrm{T}}$ to update the weights as [48]

$$g_i^{(k+1)}(t) := \left(\left|\hat{z}_i^{(k)}(t)\right| + \epsilon_0\right)^{-1}, \ i = 1, \ldots, NW, \tag{79}$$

where $\epsilon_0 > 0$ is a small positive stability parameter. In practice, the two iteration steps (78) and (79) of the IRW-ℓ_1 are alternated until either the weights have converged (e.g., when $\left\|\mathbf{G}^{(k+1)}(t) - \mathbf{G}^{(k)}(t)\right\|_{\mathrm{F}} / \left\|\mathbf{G}^{(k)}(t)\right\|_{\mathrm{F}} \leq \epsilon_{\mathrm{G}}$ with a pre-defined tolerance $\epsilon_{\mathrm{G}} > 0$), or until a maximum number of iterations $k_{\max}$ is reached. Typically, most of the gain is achieved in the first few iterations of the IRW-ℓ_1 [48].

Prior to describing the complete algorithm in Section 5, the two modifications – the ℓ_2-regularization term (77) and the IRW-ℓ_1 – which differentiate problem (78) from (73) are separately elaborated. In particular, an effective weight initialization strategy for the IRW-ℓ_1 is proposed. Note that for $\mathbf{G}^{(k)}(t) = \mathbf{I}_{NW}$ and $\gamma_{\mathrm{B}} = 0$, the problem (78) is equivalent to (73).

The ℓ_2-Regularization Term

Recall that $\gamma_{\mathrm{B}}\left\|\mathbf{\Psi}'\tilde{\mathbf{z}} - \operatorname{vec}\left(\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)\right)\right\|_2$ adds regularization to the variations in the estimates $\hat{\mathbf{X}}_{\mathrm{B}}^{(t)}(t)$ and $\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)$ obtained at the successive decoding instants. In other words, at time t, the regularization induces an extra cost for the inconsistency of the new estimate $\hat{\mathbf{X}}_{\mathrm{B}}^{(t)}(t)$ compared to the previous one $\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)$. Due to the characteristics of the ℓ_2-norm, the regularization assigns very small weights to small residuals in $\left\|\operatorname{vec}\left(\hat{\mathbf{X}}_{\mathrm{B}}^{(t)}(t)\right) - \operatorname{vec}\left(\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)\right)\right\|_2$, i.e., in $\left(\sum_{n=1}^{N}\sum_{\tau=t-W+1}^{t-1}\left(\hat{x}_n^{(t)}(\tau) - \hat{x}_n^{(t-1)}(\tau)\right)^2\right)^{1/2}$ [34, Ch. 6]. Correspondingly, it has a low incentive to make small deviations even smaller, and, thus, the ℓ_2-regularization allows $\mathbf{\Psi}'\tilde{\mathbf{z}} = \hat{\mathbf{x}}_{\mathrm{B}}^{(t)}(t)$ to slightly differ from the previous estimate $\hat{\mathbf{x}}_{\mathrm{B}}^{(t-1)}(t)$. Thus, besides facilitating the recovery of the current sensors' readings $\underline{\mathbf{x}}(t)$, the ℓ_2-regularization allows the algorithm to refine the previous estimates $\hat{\underline{\mathbf{x}}}(t-W+1), \ldots, \hat{\underline{\mathbf{x}}}(t-1)$ through the sliding window processing. This is also supported by the empirical findings in Section 5.

The IRW-ℓ_1

The weighted ℓ_1-norm is motivated by the imbalance between the ℓ_0- and ℓ_1-penalty: while the ℓ_0-term sets an equal penalty for each non-zero coefficient, the ℓ_1-norm penalizes them linearly in proportion to their magnitudes [48]. Therefore, although the ℓ_1-minimization most likely identifies the locations of large entries in $\mathbf{z}(t)$, the magnitudes – especially those of small elements – can remain inaccurate. As a remedy, the reweighting (79) compensates for the magnitude dependency of the ℓ_1-norm: it assigns a weight $g_i^{(k)}(t)$ inversely proportionally to the corresponding (expected) coefficient magnitude $|z_i|$ so that the terms in the objective of (78) become roughly equalized as $\left|g_i^{(k)}(t)\tilde{z}_i\right| \approx 1$, $i = 1, \ldots, NW$. Hence, in the course of iterations, the weighted ℓ_1-minimization starts to resemble the sparsity-optimal ℓ_0-minimization, thereby improving the CS decoding accuracy [48].

Weight Initialization: Obviously, the CS recovery performance of the recursive IRW-ℓ_1 depends on the initial point, because all successive iterations for $k \geq 2$ implicitly rely on that solution [48]. An intuitive strategy also found to work well in practice is to use $\mathbf{G}^{(1)}(t) = \mathbf{I}_{NW}$ (i.e., the ℓ_1-minimization) [48]. With no prior signal information, this is a reasonable choice, as the ℓ_1-minimization has the best theoretically established recovery threshold amongst polynomial-complexity decoding algorithms for sparse signals [312]. However, the decoding problems (78) at consecutive instants $t-1$ and t reconstruct the estimates for $\mathbf{X}(t-1) = [\underline{\mathbf{x}}(t-W) \;\; \mathbf{X}_{\mathrm{B}}(t)]$ and $\mathbf{X}(t) = [\mathbf{X}_{\mathrm{B}}(t) \;\; \underline{\mathbf{x}}(t)]$, respectively. Thus, because the data windows share the overlapping block of temporally (and spatially) correlated sensors' readings $\mathbf{X}_{\mathrm{B}}(t)$, the corresponding joint transform coefficients $\mathbf{z}(t-1) = \mathbf{\Psi}^{-1}\mathrm{vec}\left(\mathbf{X}(t-1)\right)$ and $\mathbf{z}(t) = \mathbf{\Psi}^{-1}\mathrm{vec}\left(\mathbf{X}(t)\right)$ can be expected to be close to each other. This supplementary information will be used in the following weight initialization to improve the reconstruction of the sensor data streams.

By the above reasonings, the weights at decoding instant $t > W$ are initialized with the final weights used in the previous instant $t - 1$[6]. Let κ_t denote the last iteration k taken for the IRW-ℓ_1 at decoding instant t. Accordingly, $\hat{\mathbf{z}}^{(\kappa_t)}(t) = \left[\hat{z}_1^{(\kappa_t)}(t) \cdots \hat{z}_{NW}^{(\kappa_t)}(t)\right]^{\mathrm{T}}$ are the final variables obtained for (78) at decoding instant t; $\mathbf{G}^{(\kappa_t)}(t)$ is the corresponding weight matrix. Hence, instead of confining to the ℓ_1-minimization ($\mathbf{G}^{(1)}(t) := \mathbf{I}_{NW}$) in the first iteration, the weight matrix is set as $\mathbf{G}^{(1)}(t) := \mathbf{G}^{(\kappa_{t-1})}(t-1)$ at each $t > W$, i.e., the initial weights become

$$g_i^{(1)}(t) := \left(\left|\hat{z}_i^{(\kappa_{t-1}-1)}(t-1)\right| + \epsilon_0\right)^{-1},\ i = 1, \ldots, NW. \tag{80}$$

The compression gains brought by the weight initialization are numerically demonstrated in Section 5.

Algorithm Summary

The presented sequential compressed data acquisition method with progressive CS signal recovery based on the ℓ_2-norm regularization and the IRW-ℓ_1 (termed Seq-Prog-CS) is described in Algorithm 6. The main operations are summarized as follows. At each time slot $t \geq W$, the sink gathers the measurements (70) by acquiring the readings from a subset of the sensors (Step I.). Then, the decoder solves the modified CS recovery problem (78) via the IRW-ℓ_1, resulting in an estimate of $\mathbf{X}(t)$ as $\hat{\mathbf{X}}^{(t)}(t) = \left[\hat{\underline{\mathbf{x}}}^{(t)}(t-W+1) \cdots \hat{\underline{\mathbf{x}}}^{(t)}(t)\right]$ (Step II.). Once reconstructed, its $W-1$ last columns $\left[\hat{\underline{\mathbf{x}}}^{(t)}(t-W+2) \cdots \hat{\underline{\mathbf{x}}}^{(t)}(t)\right]$ are used to form the decoder buffer $\hat{\mathbf{X}}_{\mathrm{B}}^{(t)}(t+1)$ in (75) for the next instant $t+1$ (Step III.). Then, $\hat{\mathbf{X}}^{(t)}(t)$ is used to update the estimates of $\underline{\mathbf{x}}(t-W+1), \ldots, \underline{\mathbf{x}}(t)$ as $\{\hat{\underline{\mathbf{x}}}(t-W+1), \ldots, \hat{\underline{\mathbf{x}}}(t)\} \leftarrow \left\{\hat{\underline{\mathbf{x}}}^{(t)}(t-W+1) \cdots \hat{\underline{\mathbf{x}}}^{(t)}(t)\right\}$, i.e., the estimates of the sensor data streams are gradually refined based on the most recently obtained estimates (Step IV.). Finally, the sliding window is advanced by one step (Step V.), and the procedure is repeated.

At the starting phase, estimates of $\underline{\mathbf{x}}(1), \ldots, \underline{\mathbf{x}}(W-1)$ are required to form the decoder buffer $\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)$ and initialize the IRW-ℓ_1. These can be attained

[6] The authors of [13] propose to initialize the weights based on the signal estimate from the previous instant along with the available measurements, albeit mainly to speed up the iterative solution process.

Algorithm 6 Sequential compressed data acquisition with progressive CS signal reconstruction (Seq-Prog-CS)

```
Parameters: $M_t$, $W$, $k_{\max}$, $\gamma_{\mathrm{B}}$, $\epsilon_0$ and $\epsilon_{\mathrm{G}}$.
Initializations: (i) Set $t = W$; (ii) obtain $\left[\underline{\hat{\mathbf{x}}}(1) \cdots \underline{\hat{\mathbf{x}}}(t-1)\right]$ to form
$\hat{\mathbf{X}}_{\mathrm{B}}^{(t-1)}(t)$; (iii) set $\kappa_{t-1} = 1$ and $\mathbf{G}^{(1)}(t-1) = \mathbf{I}_{NW}$.
I. Measurements
Deliver the measurements $\underline{\mathbf{y}}(t) = \underline{\mathbf{\Phi}}(t)\underline{\mathbf{x}}(t)$ in (70) to the sink.
II. Progressive signal reconstruction
a) Construct $\mathbf{y}(t) = \left[\underline{\mathbf{y}}^{\mathrm{T}}(t-W+1) \cdots \underline{\mathbf{y}}^{\mathrm{T}}(t)\right]^{\mathrm{T}}$.
b) Set $k = 1$, and initialize $\mathbf{G}^{(k)}(t)$ according to (80).
c) Run the IRW-$\ell_1$ to obtain $\hat{\mathbf{z}}^{(\kappa_t)}(t)$ and $\mathbf{G}^{(\kappa_t)}(t)$:
repeat
    1) Solve (78) to obtain $\hat{\mathbf{z}}^{(k)}(t) = \left[\hat{z}_1^{(k)}(t) \cdots \hat{z}_{NW}^{(k)}(t)\right]^{\mathrm{T}}$.
    2) Set weight matrix $\mathbf{G}^{(k+1)}(t)$ according to (79).
    3) Set $\kappa_t = k$ and $k = k+1$.
until $\|\mathbf{G}^{(k)}(t) - \mathbf{G}^{(k-1)}(t)\|_{\mathrm{F}} / \|\mathbf{G}^{(k-1)}(t)\|_{\mathrm{F}} \leq \epsilon_{\mathrm{G}}$ or $k > k_{\max}$.
d) Reconstruct the estimate for $\mathbf{X}(t)$ as $\mathrm{vec}(\hat{\mathbf{X}}^{(t)}(t)) = \mathbf{\Psi}\hat{\mathbf{z}}^{(\kappa_t)}(t)$, and
reshape it as $\hat{\mathbf{X}}^{(t)}(t) = \left[\underline{\hat{\mathbf{x}}}^{(t)}(t-W+1) \cdots \underline{\hat{\mathbf{x}}}^{(t)}(t)\right]$.
III. Decoder buffer update
Set the decoder buffer in (75) as
$\hat{\mathbf{X}}_{\mathrm{B}}^{(t)}(t+1) = \left[\underline{\hat{\mathbf{x}}}^{(t)}(t-W+2) \cdots \underline{\hat{\mathbf{x}}}^{(t)}(t)\right]$.
IV. Estimate update
Set $\{\underline{\hat{\mathbf{x}}}(t-W+1), \ldots, \underline{\hat{\mathbf{x}}}(t)\} \leftarrow \left\{\underline{\hat{\mathbf{x}}}^{(t)}(t-W+1) \cdots \underline{\hat{\mathbf{x}}}^{(t)}(t)\right\}$.
V. Sliding window advance step
Set $t = t+1$, and go to Step I.
```

for example via (73). As there is no prior signal information available, it may be necessary to use more measurements compared to the subsequent steps $t \geq W$ [286]. In general, due to the recursive decoding, all preceding estimates from time slots $\{\ldots, t-2, t-1\}$ implicitly affect the recovery performance at the time slot t. Thus, inaccurate estimates may induce error propagation. The stability over time can be improved via intermittently requesting for more measurements from the sensors (up to $M(t) = N$), and consequently, to provide more reliable prior information for the sliding window processing.

Simulation Results

Numerical experiments are presented to illustrate the performance of the devised Seq-Prog-CS method in terms of achievable energy savings and signal reconstruction accuracy as compared to several baseline CS methods.

Simulation Setup

Consider WSN topologies with $N = 9$, $N = 16$, and $N = 25$. The sensors monitor a phenomenon over T sampling instants, resulting in a matrix of the sensors' readings as $\mathbf{X} = [\underline{\mathbf{x}}(1) \cdots \underline{\mathbf{x}}(T)] \in \mathbb{R}^{N\times T}$. The sensors are deployed in an observation field of size $100\sqrt{N}\times 100\sqrt{N}$ units as follows: the field is divided into a $\sqrt{N}\times\sqrt{N}$-grid of square areas, where each 100×100 square is randomly deployed with one sensor according to a uniform distribution. The sensors use a shortest path tree to route their data to the sink, which is located at the center of a WSN. Two sensors n and j are able to communicate if $d_{nj} \leq 100\sqrt{5}$, $n, j \in \mathcal{N}$, $n \neq j$, guaranteeing a connected WSN.

Generation of WSN Data

Each WSN consists of $S = \sqrt{N}$ independent, randomly located sources. Spatially and temporally correlated sensor data $\mathbf{X}$ is generated as follows.

Spatial Correlation: A power exponential correlation model [24, 5, 330] is used to model the influence functions in (64): $h_{ns}(d_{ns}) = \exp\{-(d_{ns}/\rho_1)^{\rho_2}\}$, where parameters $\rho_1 > 0$ and $\rho_2 \in [0, 2]$ adjust the correlation decay rate and geometrical properties of the field, respectively [24].

Temporal Correlation: Temporally correlated, piecewise smooth source sequences $\boldsymbol{\beta}_s \in \mathbb{R}^T$, $s \in \mathcal{S}$, in (64), are generated as follows.

A. Smooth part: For each $s \in \mathcal{S}$, a sequence $\boldsymbol{\lambda}_s = [\lambda_s(1) \cdots \lambda_s(T)]^{\mathrm{T}}$ is created according to a Gauss-Markov process as

$$\lambda_s(t) = \alpha_s[\lambda_s(t-1) - \mu_s] + (1-\alpha_s)\varsigma_s(t) + \mu_s,\ t = 2, \ldots, T, \tag{81}$$

where $\alpha_s \in [0, 1]$ is a correlation parameter, $\varsigma_s(t) \sim \mathcal{N}(0, \sigma_\varsigma^2)$ is the innovation component, and μ_s is the mean component [326, 327, 241]. Thus, for $\alpha_s = 1$, $\boldsymbol{\lambda}_s$ remains constant, whereas for $\alpha_s = 0$, it behaves as an uncorrelated Gaussian random process. In order to create smoothly evolving

sequences, each $\boldsymbol{\lambda}_s$ is low-pass filtered by applying the T-point DFT as $\mathbf{f}_{\text{dft},s} = \mathbf{F}_{\text{dft}}\boldsymbol{\lambda}_s$, where $\mathbf{F}_{\text{dft}} \in \mathbb{C}^{T\times T}$ is the DFT-matrix, and $\mathbf{f}_{\text{dft},s} \in \mathbb{C}^T$ is the vector of the DFT-coefficients. Each $\mathbf{f}_{\text{dft},s}$ is used to form a vector $\tilde{\mathbf{f}}_{\text{dft},s} \in \mathbb{C}^T$ by preserving $\eta_{\text{lpf}} \in [1, T/2]$ lowest frequency components of $\mathbf{f}_{\text{dft},s}$, while setting the rest to zero. Finally, taking the T-point inverse DFT, a low-pass version of $\boldsymbol{\lambda}_s$ is obtained as $\tilde{\boldsymbol{\lambda}}_s = \mathbf{F}_{\text{dft}}^{-1}\tilde{\mathbf{f}}_{\text{dft},s}$. The parameters are set as $\alpha_s = 0.9$, $\lambda_s(1) \sim \mathcal{N}(0, \sigma_\lambda^2)$ and $\mu_s = \lambda_s(1)$, $\forall s \in \mathcal{S}$, $\sigma_\lambda^2 = \sigma_\varsigma^2 = 1$, and $\eta_{\text{lpf}} = \lceil 0.3(T/2)\rceil$, where $\lceil\cdot\rceil$ denotes rounding up to the nearest integer.
B. Abrupt changes: Let $\boldsymbol{\chi}_s = [\chi_s(1) \cdots \chi_s(T)]^{\mathrm{T}}$, $s \in \mathcal{S}$, be a Markov chain sequence given by the state transition matrix $\mathbf{P}^{\text{mc}} \in \mathbb{R}^{|\mathcal{Q}_s|\times|\mathcal{Q}_s|}$ and the state space $\mathcal{Q}_s = \{q_{s1}, \ldots, q_{s|\mathcal{Q}_s|}\}$ [185]. A (k, l)th entry of $\mathbf{P}^{\text{mc}}$, p_{kl}^{mc}, gives the probability of changing the state from $q_{sk} \in \mathcal{Q}_s$ to $q_{sl} \in \mathcal{Q}_s$. The parameters are set as $p_{kl}^{\text{mc}} = P_1 \triangleq (1 - P_0)/(|\mathcal{Q}_s| - 1)$, $\forall k \neq l$, where $P_0 = p_{kk}^{\text{mc}}$. Thus, for $P_0 = 1$, $\boldsymbol{\chi}_s$ remains constant, whereas for $P_0 = 0$, the state changes at every step. We set $|\mathcal{Q}_s| = 10$ and $q_{sk} \sim \mathcal{N}(0, 1)$, $\forall s \in \mathcal{S}$, $k = 1, \ldots, |\mathcal{Q}_s|$. Finally, by forming

$$\boldsymbol{\beta}_s = \tilde{\boldsymbol{\lambda}}_s + \boldsymbol{\chi}_s, \;\; s \in \mathcal{S}, \tag{82}$$

the resulting source sequences of form (64) are piecewise smooth; the smoothness and the degree of temporal correlation are adjusted by $\eta_{\text{lpf}} \in [1, T/2]$ and $\alpha_s \in [0, 1]$, and the number of unpredictable changes by $P_0 \in [0, 1]$. It is worth emphasizing that owing to the used influence functions $h_{ns}(d_{ns})$ and the generated source magnitude sequences $\boldsymbol{\beta}_s$, $s \in \mathcal{S}$, the sensor data streams of (64) are only *compressible* (i.e., not exactly sparse) in both the spatial and temporal domain.

Spatial Domain Transform

2D-DCT is applied to $\mathbf{X}$ to obtain a compressible spatial domain representation of each $\underline{\mathbf{x}}(t)$, $t = 1, \ldots, T$. Firstly, each $\underline{\mathbf{x}}(t)$ is reorganized into a matrix $\underline{\mathbf{X}}(t) \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$, whose (n, j)th entry contains the reading of the sensor at the (n, j)th square of the $\sqrt{N}\times\sqrt{N}$-grid, $n, j = 1, \ldots, \sqrt{N}$. Similarly to (69), $\underline{\mathbf{X}}(t)$ can be expressed as $\underline{\mathbf{X}}(t) = \mathbf{F}_{\text{dct}}^{-1}\underline{\boldsymbol{\Theta}}_{\text{S}}(t)\mathbf{F}_{\text{dct}}^{-\text{T}}$, i.e., $\underline{\mathbf{x}}(t) = \text{vec}\left(\underline{\mathbf{X}}(t)\right) = \boldsymbol{\Psi}_{\text{S}}\boldsymbol{\theta}_{\text{S}}(t)$, where $\mathbf{F}_{\text{dct}}^{-1} \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$ is the inverse of a DCT-matrix, $\underline{\boldsymbol{\Theta}}_{\text{S}}(t) \in \mathbb{R}^{\sqrt{N}\times\sqrt{N}}$ contains the DCT-coefficients in a matrix form, $\mathbf{\Psi}_{\mathrm{S}} = (\mathbf{F}_{\mathrm{dct}}^{-1} \otimes \mathbf{F}_{\mathrm{dct}}^{-1})$ is the (Kronecker) sparsifying basis for the spatial domain, and $\boldsymbol{\theta}_{\mathrm{S}}(t) = \mathrm{vec}\left(\underline{\mathbf{\Theta}}_{\mathrm{S}}(t)\right) \in \mathbb{R}^{N}$.

Performance Metrics

The performance of Seq-Prog-CS method is evaluated in terms of the data transportation costs and signal reconstruction accuracy. The total cost of delivering the necessary amount of data to the sink to reconstruct $\mathbf{X}$ is measured via a normalized communication cost [238]

$$C = \frac{\sum_{t=1}^{T} \sum_{n=1}^{N} E_n I_n(t)}{\sum_{t=1}^{T} \sum_{n=1}^{N} E_n}, \tag{83}$$

where E_n denotes the number of hops from sensor $n \in \mathcal{N}$ to the sink, and $I_n(t)$ is an indicator function with $I_n(t) = 1$, if $n \in \mathcal{N}(t)$, and $I_n(t) = 0$ otherwise. Thus, E_n captures the impact of multi-hop routing on the energy expenditure of sensor n, and $I_n(t)$ indicates whether the sensor transmits its reading at slot t. By the normalization, C compares the total number of transmitted data units, $x_n(t)$, for a CS method to that of the multi-hop forwarding of all NT readings. Assuming a unit cost for transmitting each sensor reading $x_n(t)$, $C < 1$ means that a CS method decreases the sensor communications[7].
For a given C, the CS reconstruction error is measured as

$$\xi_{\mathrm{cs}} = \frac{\left\| \mathrm{vec}(\hat{\mathbf{X}}) - \mathrm{vec}(\mathbf{X}) \right\|_2}{\left\| \mathrm{vec}(\mathbf{X}) \right\|_2}, \tag{84}$$

where $\hat{\mathbf{X}}$ denotes the estimate of $\mathbf{X}$.

CS Methods

The Seq-Prog-CS method is compared against three baseline CS methods:

1. KCS (Kron-CS) that splits $\mathbf{X}$ into B_{kcs} consecutive non-overlapping blocks[8] as $\mathbf{X} = [\mathbf{X}_{[1]} \cdots \mathbf{X}_{[B_{\mathrm{kcs}}]}]$, where $\mathbf{X}_{[l]} \in \mathbb{R}^{N \times (T/B_{\mathrm{kcs}})}$, $l = 1, \ldots, B_{\mathrm{kcs}}$, and reconstructs them separately via (73) with $W := T/B_{\mathrm{kcs}}$.

[7] While C neglects the effects of the packet transmission protocol, overhead, and channel access etc., it assesses the main factors contributing to the sensors' energy consumption.

[8] In order to restrain the decoding complexity in the KCS, the reconstruction of $\mathbf{X}$ is partitioned into smaller blocks.

2. Spatial CS (Spat-CS) that reconstructs each $\underline{\mathbf{x}}(t)$, $t = 1, \dots, T$, separately via (74).

3. A dynamic version of the regularized modified-CS (Reg-Mod-CS) proposed in [286], which utilizes the support and estimate knowledge recursively from the preceding decoding instant.

For the ease of comparison, the same randomly generated $\underline{\mathbf{\Phi}}(t)$ with $M(t) = M$ is used for each CS method at each time instant t, guaranteeing equal C in (83) for all methods. For given N and W, empirically tuned and fixed regularization weight parameters γ_{B} in Seq-Prog-CS and γ_{reg} in Reg-Mod-CS are used, i.e., they are not specifically fine-tuned for each network realization. The estimates for the first W sensors' readings in the Seq-Prog-CS and Reg-Mod-CS methods are reconstructed via (73). To illustrate the gradual estimate refinement, the Seq-Prog-CS method is reported in respect to 1) the first estimates obtained for each $\underline{\mathbf{x}}(t)$, and 2) the estimates after decoded W times, which are marked with the superscripts $(\cdot)^{(t)}$ and $(\cdot)^{(t+W-1)}$, respectively. The Seq-Prog-CS$^{(t)}$ is also included without the ℓ_2-regularization and the IRW-ℓ_1 (Seq-CS), which reconstructs each $\mathbf{X}(t)$ non-recursively (yet with streaming processing) via (73). For the IRW-ℓ_1, the parameters are set as $k_{\max} = 5$, $\epsilon_0 = 1 \times 10^{-1}$, and $\epsilon_{\mathrm{G}} = 1 \times 10^{-3}$. For Reg-Mod-CS, $b = 99.5$ is used in the support detection [286, Sect. V].
The optimization problems were solved using ℓ_1-MAGIC [41] and CVX [130].
The DWT-matrices were generated with the Wavelab toolbox [90].

Performance of the Proposed Method

Influence of Window Size

The influence of window size W on the CS recovery performance of the Seq-Prog-CS method is investigated in the setup with $N = 16$, $T = 512$, and $B_{\mathrm{kcs}} = 8$. The spatial correlation is set by $\rho_1 = 1 \times 10^2$ and $\rho_2 = 2$, and the temporal by $P_0 = 1.0$ (smooth signals). The DCT-matrix is used for $\mathbf{\Psi}_{\mathrm{T}}^{-1}$. In order to highlight the impact of W, we ran Seq-CS instead of Seq-Prog-CS.
Fig. 19 shows the average CS recovery error ξ_{cs} against communication cost C for the Spat-CS, Kron-CS, and Seq-CS for window sizes $W = \{4, 8, 16, 32, 64\}$. Because the Spat-CS neglects compressibility in the

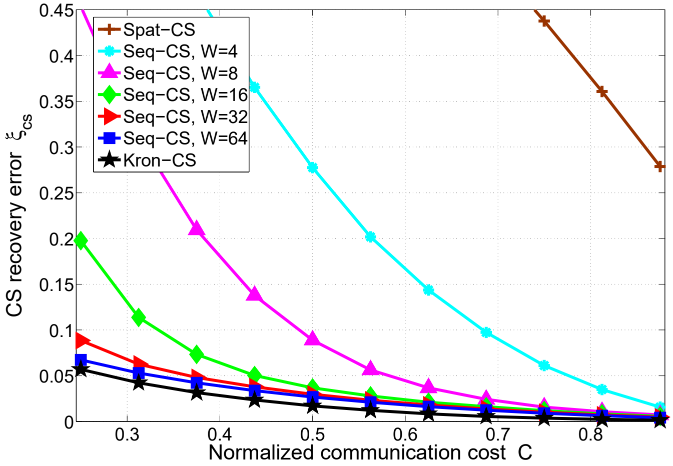


Figure 19: Reconstruction performance of the Seq-CS method with varying window sizes W against the Spat-CS and Kron-CS methods for $N = 16$, $T = 512$, $B_{\text{kcs}} = 8$, and $P_0 = 1.0$ (smooth signals).

temporal domain, its performance is inferior to all other methods. In fact, utilizing also the compressibility in the temporal domain significantly reduces the necessary sensor communications: the Seq-CS method with $W = 16$ obtained reconstruction accuracy of around $\xi_{\text{cs}} = 0.04$ while requiring only a half of the sensors' readings to be communicated to the sink. The figure also visualizes the trade-off between the decoding complexity and the CS recovery performance for Seq-CS: when W increases, the joint correlation structure of the sensor data is more efficiently utilized, and the performance gradually approaches that of the Kron-CS method, yet at the cost of increased decoding complexity. Recall, however, that by means of the sliding window processing, the Seq-CS method obtains the estimates for the current sensors' readings with no extra delay as opposed to the block-wise Kron-CS approach, for which the delay is proportional to $T/B_{\text{kcs}} = 64$.

Different Network Sizes

To illustrate the performance and scalability of the Seq-Prog-CS method for various network sizes, consider setups with $N = 9$, $N = 16$, and $N = 25$, where the other parameters are $T = 256$, $B_{\mathrm{kcs}} = 4$, $\rho_1 = 5{\times}10^2$, $\rho_2 = 2$, and $P_0 = 0.97$. The matrix $\mathbf{\Psi}_{\mathrm{T}}^{-1}$ is set as the Daubechies-4 DWT-matrix. Fig. 20 presents the resulting average CS recovery error ξ_{cs} versus communication cost C for each CS method.

In line with Fig. 19, by exploiting the spatio-temporal correlation in Seq-CS, Seq-Prog-CS, Reg-Mod-CS, and Kron-CS, the methods significantly reduce the sensors' communication costs compared to the Spat-CS method. Moreover, the benefits of utilizing prior information in the CS decoding are clearly visible: the Seq-Prog-CS method substantially improves the CS recovery performance compared to the Seq-CS method, especially for small numbers of measurements M (i.e., low values of C). Substantial improvements are also achieved by the other recursive CS method, Reg-Mod-CS, although it has lower accuracy than Seq-Prog-CS for all N. Interestingly, the performance of Seq-Prog-CS$^{(t)}$ with $W = 4$ almost matches that of Kron-CS, and with $W = 8$ and $W = 16$, Seq-Prog-CS$^{(t)}$ even outperforms the Kron-CS method. In summary, the proposed method is able to periodically reconstruct estimates for the current sensors' readings with notably reduced sensor communications. Recall that as compared to Kron-CS, the decoding complexity of the Seq-Prog-CS method is lower by a factor of $W/(T/B_{\mathrm{kcs}}) = \{1/16, 1/8, 1/4\}$ for $W = \{4, 8, 16\}$, respectively. Finally, as shown by the Seq-Prog-CS$^{(t+W-1)}$ curves, Seq-Prog-CS also substantially improves the reconstruction accuracy of the past sensors' readings via the progressive decoding.

Influence of Prior Information

Next, the impact of incorporating different types of prior signal information in the decoding process of the Seq-Prog-CS method is examined in the case with $N = 16$, $T = 256$, $B_{\mathrm{kcs}} = 4$, and $W = 8$. Particularly, the influences of the IRW-ℓ_1 (i.e., $k_{\max} > 1$), the ℓ_2-regularization (i.e., $\gamma_{\mathrm{B}} > 0$), and the weight initialization ($\mathrm{G}_{\mathrm{init}}$) of (80) (i.e., $\mathbf{G}^{(1)}(t) := \mathbf{G}^{(\kappa_{t-1})}(t-1)$ as opposed to $\mathbf{G}^{(1)}(t) := \mathbf{I}_{NW}$) are studied by running the following six variants of Seq-Prog-CS:

- Seq-CS, which excludes both the IRW-ℓ_1 and ℓ_2-reg

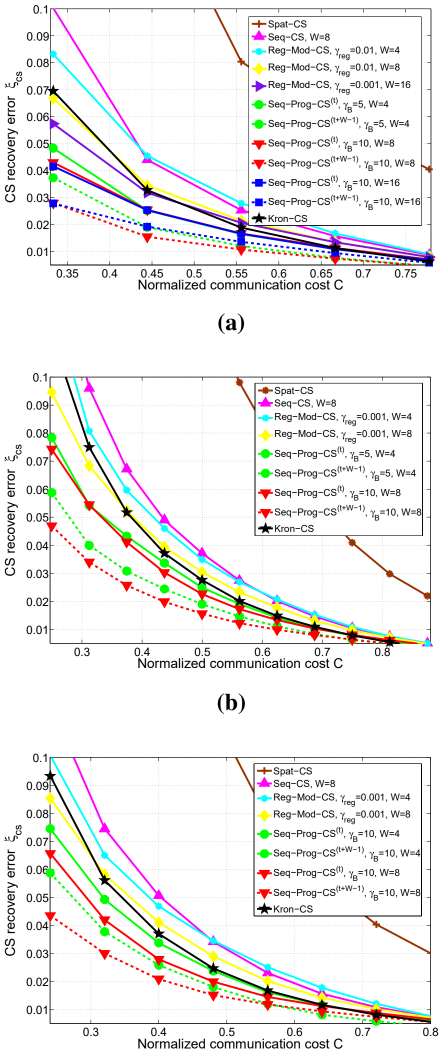


Figure 20: The CS reconstruction performance of the Spat-CS, Kron-CS, Reg-Mod-CS, Seq-Prog-CS$^{(t)}$ (solid line), and Seq-Prog-CS$^{(t+W-1)}$ (dashed line) methods for various window sizes W with $T = 256$ and $B_{\text{kcs}} = 4$ for (a) $N = 9$, (b) $N = 16$, and (c) $N = 25$.

- The proposed method in Algorithm 6, which incorporates all three forms of prior knowledge, termed Seq-Prog-CS [IRW-ℓ_1+ℓ_2-reg+G$_{\text{init}}$] ($\gamma_{\text{B}} = 10$)

- Seq-Prog-CS [IRW-ℓ_1]
- Seq-Prog-CS [IRW-ℓ_1+$\mathrm{G}_{\mathrm{init}}$]
- Seq-Prog-CS [ℓ_2-reg] ($\gamma_{\mathrm{B}} = 3$)
- Seq-Prog-CS [IRW-ℓ_1+ℓ_2-reg] ($\gamma_{\mathrm{B}} = 10$)

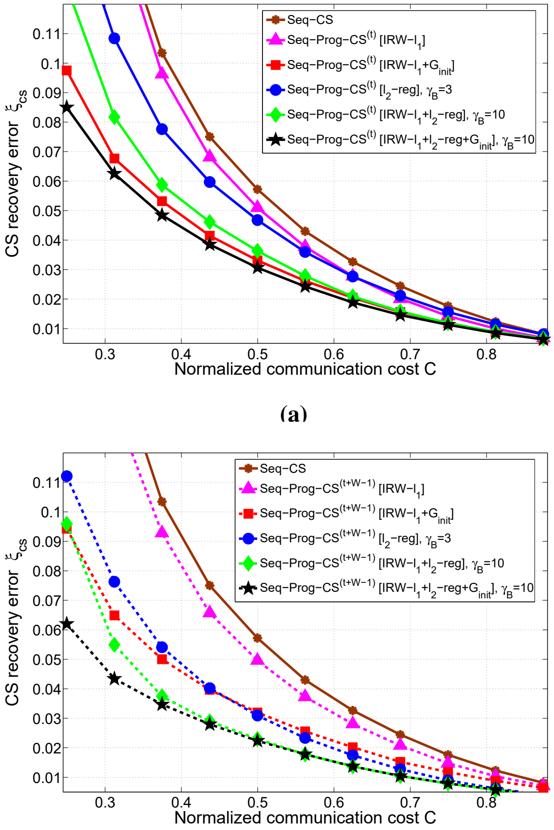
The correlation is adjusted by $\rho_1 = 5\times10^2$, $\rho_2 = 2$, and $P_0 = 0.97$, and the Daubechies-8 DWT-matrix is used for $\mathbf{\Psi}_{\mathrm{T}}^{-1}$.

Figs. 21(a) and (b) depict the resulting average CS recovery error ξ_{cs} versus the communication cost C for the above methods in comparison to the Seq-Prog-CS$^{(t)}$ and Seq-Prog-CS$^{(t+W-1)}$ methods, respectively. Fig. 21 highlights the significance of an appropriate starting point for the IRW-ℓ_1: using the ℓ_1-minimization as the initial step, [IRW-ℓ_1] can only slightly improve the CS performance from that of the Seq-CS method. However, with the proposed weight initialization (80), Seq-Prog-CS$^{(t)}$[IRW-ℓ_1+$\mathrm{G}_{\mathrm{init}}$] outperforms both [$\ell_2$-reg] and [IRW-$\ell_1$+$\ell_2$-reg]. Because the proposed method with [IRW-ℓ_1+ℓ_2-reg+$\mathrm{G}_{\mathrm{init}}$] incorporates all three forms of prior signal knowledge, it indisputably yields the best CS reconstruction accuracy amongst all the methods.

Influence of Spatial and Temporal Correlation

The CS methods were also tested under different signal characteristics by varying the parameters adjusting the spatial (ρ_1) and temporal (P_0 and α_s) correlation properties of the sensor data streams. We considered the case with $N = 16$, $M = 6$, $T = 256$, $B_{\mathrm{kcs}} = 4$, $W = 8$, and $\rho_2 = 2$. The Daubechies-4 DWT-matrix was used for $\mathbf{\Psi}_{\mathrm{T}}^{-1}$ for $P_0 < 1.0$, and the DCT-matrix for $P_0 = 1.0$.

Table 2 reports the average error ξ_{cs} for the CS methods under various signal statistics. Obviously, as the temporal correlation degrades, i.e., the parameters P_0 or/and α_s decrease, the performance of the Kron-CS, Seq-CS, and Seq-Prog-CS methods relying on temporal compressibility gradually decreases. Nevertheless, the reconstruction by the Seq-Prog-CS method is more resilient to the degraded correlation; in most of the cases, it clearly outperforms all the other methods while producing estimates with an accuracy of $\xi_{\mathrm{cs}} < 0.07$. The only exception is Kron-CS with highly correlated smooth signals (the first column), where Kron-CS logically makes the most of processing



(a)

(b)

Figure 21: The CS reconstruction performance with different types of prior signal information used in the signal recovery with $N = 16$, $T = 256$, $B_{\text{kcs}} = 4$, and $W = 8$ in terms of (a) Seq-Prog-CS$^{(t)}$, and (b) Seq-Prog-CS$^{(t+W-1)}$.

large batches of size $T/B_{\text{kcs}} = 64$. As a final remark, no method was capable of producing accurate estimates for signals with very poor spatial correlation (the last column).

Conclusions

This section addressed the framework of distributed compressed acquisition and progressive reconstruction of spatially and temporally correlated sensor

Table 2: CS reconstruction performance ($\xi_{\text{cs}} \times 100$ [%]) with different parameters for spatial (ρ_1) and temporal (P_0, α_s) correlation with $N = 16$, $T = 256$, $B_{\text{kcs}} = 4$, $W = 8$, and $M = 6$.

ρ_1	$5{\times}10^2$								$1{\times}10^3$	$1{\times}10^2$
P_0	1.0	0.97	0.97	0.97	0.90	0.80	0.80	0.7	0.97	0.97
α_s	0.95	0.90	0.70	0.50	0.90	0.90	0.5	0.4	0.90	0.90
Spat-CS	17.7	23.4	22.9	22.6	21.9	21.2	21.1	20.7	12.5	93.2
Seq-CS	1.46	6.72	7.57	8.19	9.27	11.0	11.1	11.9	2.04	38.7
SeqProgCS$^{(t)}$	0.392	4.12	4.21	4.44	5.36	6.06	6.05	6.51	1.13	35.5
SeqProgCS$^{(t+W-1)}$	0.281	2.57	3.02	3.28	3.70	4.64	4.65	5.13	0.809	21.8
Kron-CS	0.188	5.17	6.17	6.99	7.92	9.76	10.1	11.1	1.60	27.4

data streams in multi-hop WSNs. A sequential CS method relying on sliding window processing summarized in Algorithm 6 was devised. By means of the Kronecker sparsifying basis, ℓ_2-regularization, and adaption of the IRW-ℓ_1, the proposed method efficiently utilizes the spatio-temporal signal correlation and the estimates from the successive sliding windows to improve the signal recovery performance. The simulation results illustrated the benefits of utilizing the joint signal dependencies and prior information in the CS recovery: by adjusting the window size, the proposed method achieved higher reconstruction accuracy with a smaller number of required transmissions, and with less decoding delay and complexity compared to several baseline CS methods.

Owing to the high compression performance demonstrated by the numerical experiments, the presented method has great potential to prolong the lifetime of battery-powered sensors in various monitoring applications. Furthermore, the ability to trade off between the performance and complexity via the window size makes the algorithm versatile for applications with different requirements for, e.g., computation power and reconstruction fidelity. Thanks to the innate properties of the CS, the method benefits of the simple, decentralized and universal encoding strategy at the sensors, whereas the acquisition of the global correlation structure and the implementation of a computationally demanding decoding algorithm is shifted to the sink/fusion center. In summary, the presented method is a competent candidate for streaming-type compression applications when striving for high signal reconstruction accuracy in a cost-effective manner.

6

Distributed Source Coding via Quantized Compressed Sensing

This section addresses lossy *distributed source coding* (DSC) for acquiring correlated sparse sources using CS in WSNs. Accordingly, noisy compressive measurements are separately encoded at a finite rate by each sensor, followed by joint reconstruction of the sources at the decoder. The main distinction from Section 5 is that the framework now involves *quantization* of compressive measurements, which is a practical necessity for finite-rate data communications/storage. We present a distributed complexity-constrained variable-rate quantized CS method that minimizes a weighted sum between the mean square error signal reconstruction distortion and the average encoding rate.

One key objective is to restrain the encoding complexity of each sensor while targeting to high distortion-rate performance. To this end, we pre-quantize the encoder inputs, i.e., the compressive measurements, via vector quantization at each sensor. Differently from the prior works [265, 266], pre-quantization eliminates the need of reconstructing exponentially complex estimates at each sensor. It is worth noting that while the use of VQ is in favor of good compression, it limits the proposed method to setups with moderate signal dimensions. Hence, techniques to lower the encoding complexity will also be discussed.

In particular, each encoder is modeled as a quantizer followed by a lossless entropy encoder. In this work, variable-rate coding is incorporated via rate measures of particular entropy bounds. We confine to a two-sensor system and present the algorithm design methodology along with practical training algorithms and complexity analysis. Simulation results illustrate that the proposed method achieves superior compression performance compared to several baseline methods and lends itself to versatile setups with different performance requirements.

Related Works

Distributed Joint Estimation and Compression

Owing to the decentralized network structure of a WSN, the compression of sensor signals calls for DSC [233, 311, 91], commonly known as *Slepian-Wolf (SW) coding* [268]. In a DSC setup, multiple sensors separately measure, encode, and communicate one or multiple (correlated) sources to a decoder. Since typical sensor signals are analog/continuous-valued, quantization [135] is inevitable, i.e., the compression becomes *lossy* [26]. Furthermore, as encoding at each CS based sensor relies only on indirect observations of a source, the setup necessitates *remote source coding* [85]. Thus, the compression in a distributed CS setup under quantization of measurements extends to distributed joint estimation and compression [310].
Vector quantization (VQ) [133] is a practical compression method capable to achieve performance arbitrary close to the theoretical optimum [136]. VQ operates in a *symbol-by-symbol* fashion by mapping a single observation sample into a digital output at a time. The inevitable loss from a finite VQ dimension can be compensated for by using *variable-length coding* [149, 329] and [74, Ch. 5], where the main idea is to assign shorter codewords to more frequent symbols to minimize the average transmission rate. In a lossy compression context, such *entropy coding* can be incorporated via entropy-constrained VQ (ECVQ) design [66], which models the encoder as a VQ followed by a lossless entropy encoder for the index sequence. Entropy coding under uniform quantization has been studied in, e.g., [125, 328]. In practice,

VQs/ECVQs can be optimized by the Linde-Buzo-Gray (LBG) algorithm [188], descending from the iterative Lloyd algorithm[9] [211, 192].

Distributed Quantized CS

While the early era of CS exclusively addressed real-valued signals, the inevitable conversion of measurements into bit sequences initiated *quantized CS* (QCS) [129, 273, 333, 151]. The overarching idea behind QCS algorithms is to accommodate the non-linear impact of CS and quantization in the encoder/decoder to ameliorate the signal recovery performance under discretized measurements. The works in [273, 157, 263] devised optimized scalar quantizers (SQs) for a standard CS reconstruction algorithm, whereas quantization-aware CS decoding algorithms for a fixed SQ based encoder were developed in [333, 78, 151, 68, 32]. While the methods outperform the plain quantization-unaware versions, they are suboptimal for minimizing the mean square error (MSE) of the signal reconstruction distortion because they 1) optimize only either the encoder or decoder, 2) use SQ instead of VQ, and/or 3) minimize the measurement quantization distortion, which due to the non-linearities does not in general minimize the signal reconstruction distortion [262, Sect. 3.2.3] and [78].

As a countermeasure, joint optimization of VQ based encoder-decoder pair(s) to minimize signal reconstruction MSE in QCS setups was first proposed in [265, 266, 264, 262]. Shirazinia *et al.* [265] derived necessary conditions for an MSE-optimal fixed-rate single-sensor QCS system for acquiring a sparse source over noisy channels. This was extended to distributed joint estimation and compression of two correlated sparse sources in [264, 266]. However, the enhanced compression entails high encoding complexity; a sensor has to reconstruct a minimum mean square error (MMSE) estimate – a task of exponential complexity [99, 235, 281] – impeding the practical implementation of the methods.

[9] As mentioned in the "Author's Note 1981" section of [192], nearly all Lloyd's results were already presented in an unpublished Bell Labs manuscript in 1957.

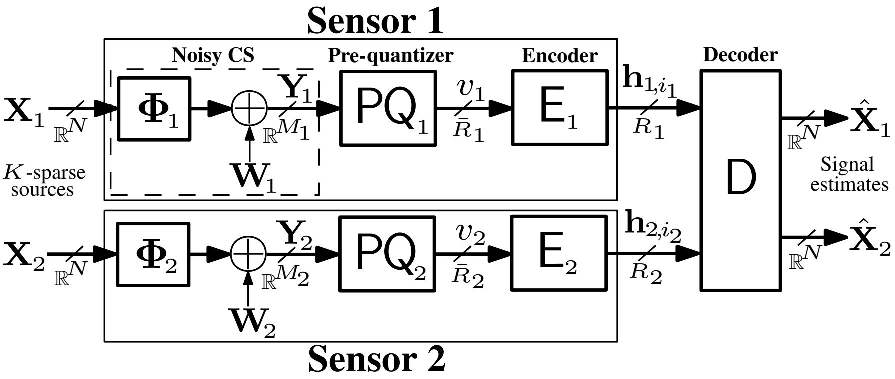


Figure 22: Distributed variable-rate quantized CS acquisition of correlated sparse sources with complexity-constrained encoders.

System Model

Consider a distributed QCS system consisting of two CS based sensors and one sink, as depicted in Fig. 22. In the DSC approach, each sensor acquires noisy compressive measurements of its source, converts them into finite-rate bit sequences via *separate encoding* (i.e., without inter-sensor collaboration), and communicates the messages to the sink for *joint decoding* of both sources. We focus merely on the source coding, and, thus, the transmissions from each sensor to the sink are assumed to be error-free. Note that the presented alternating optimization framework, in which one system block is optimized while keeping the others fixed, can readily be extended to a general multi-sensor system, although the computational and memory requirements rapidly grow intolerably high.

CS Signal Acquisition

The correlated sources $\mathbf{X}_1$[10] and $\mathbf{X}_2$ are given by JSM-2 model [19, 95] as

$$\mathbf{X}_l = \bar{\mathbf{X}} + \mathbf{X}'_l,\ l = 1, 2, \tag{85}$$

[10] In this section, with slight abuse of notation, we use capital bold letters (e.g., $\mathbf{X}$) to refer to *random* vectors, and small bold letters to refer to their corresponding realizations (e.g., $\mathbf{X} = \mathbf{x}$).

where the *common* component $\bar{\mathbf{X}}$ and *innovation* component $\mathbf{X}'_l$ are real-valued length-N random vectors where both are K-sparse and share the same random (unknown) support, i.e., the set of indices of non-zero components. Consequently, each real-valued length-N source random vector $\mathbf{X}_l$ is K-sparse, i.e., for each realization $\|\mathbf{x}_l\|_0 \leq K \leq N$, $l = 1, 2$. The vectors $\bar{\mathbf{X}}$, $\mathbf{X}'_1$, and $\mathbf{X}'_2$ are assumed to be independent of each other. JSM-2 signals are encountered, e.g., in a group of sensors monitoring an audio source or spectrum occupancy [19]. Various common sparse signals have also been studied under compressive support recovery and signal reconstruction problems with multiple measurement vectors in a variety of monitoring applications [72, 300, 325].

Let $\mathcal{T}_s \subseteq \{1, \ldots, N\}$ be an index set representing the sth sparsity pattern with $|\mathcal{T}_s| = K$, $s = 1, \ldots, \binom{N}{K}$. The $\binom{N}{K}$ index sets are different, i.e., $\mathcal{T}_s \setminus \mathcal{T}_{s'} \neq \emptyset$, $\forall s' \neq s = 1, \ldots, \binom{N}{K}$. Each support $\mathcal{T}_s$ is associated with the *a priori* probability $p(\mathcal{T}_s) \in [0, 1]$ with $\sum_{s=1}^{\binom{N}{K}} p(\mathcal{T}_s) = 1$.

Each sensor measures the source $\mathbf{X}_l$ through a fixed (and known) measurement matrix $\mathbf{\Phi}_l \in \mathbb{R}^{M_l \times N}$ as

$$\mathbf{Y}_l = \mathbf{\Phi}_l \mathbf{X}_l + \mathbf{W}_l, \; l = 1, 2, \tag{86}$$

where $\mathbf{Y}_l$ is the length-M_l measurement random vector, and $\mathbf{W}_l \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\mathbf{W}} \mathbf{I}_{M_l})$ is the measurement noise random vector. The structure of each $\mathbf{\Phi}_l$, $l = 1, 2$, has a significant impact on the CS signal recovery performance; see Section 2 for the discussions on the RIP and the coherence of $\mathbf{\Phi}_l$. Nonetheless, no restrictive assumptions of $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$ are needed in the derivations.

Whereas CS typically assumes $K \leq M_l \leq N$, the presented design is not restricted to any particular range for M_l. In fact, while $M_l \leq N$ complies with the fundamental CS theory, over-sampling (i.e., $M_l > N$) may be useful in QCS setups. Namely, given a quantization bit resolution of an analog-to-digital converter (ADC), over-sampling is a practical – and often cost-effective – means to improve the reconstruction accuracy [151]. It is also worth emphasizing that because (86) models the physics of the sensing process, the encoder at each CS based sensor $l = 1, 2$ has no access to $\mathbf{X}_l$, but only to $\mathbf{Y}_l$. Consequently, the compression scheme falls under remote source coding [85].

Measurement Space Pre-Quantization

Prior to the actual (desirably low-rate) source encoding, the input of the encoding system at each sensor $l = 1, 2$, i.e., the measurement random vector $\mathbf{Y}_l$ in (86), is *discretized* with a VQ. Each pre-quantizer[11] PQ_l is a key block to restrain the encoding complexity of a sensor while providing high compression performance. As a by-product, this also simplifies the optimization design by converting the optimization over continuous random variables into optimization over discrete ones. Thus, it facilitates offline training by allowing pre-computation of several required quantities. Next, we present a general description of each pre-quantizer, whereas their optimization is deferred until Section 6.

Let $\mathcal{V}_l \triangleq \{1, \ldots, |\mathcal{V}_l|\}$ be a set of *cell indices* $v_l \in \mathcal{V}_l$ with pre-quantization rate $\bar{R}_l = \log_2 |\mathcal{V}_l|$ bits/vector $\mathbf{Y}_l$ for sensor $l = 1, 2$. Let $\mathcal{G}_l \triangleq \{\mathbf{g}_{l,1}, \ldots, \mathbf{g}_{l,|\mathcal{V}_l|}\}$ be a *pre-quantization codebook* consisting of *codepoints* $\mathbf{g}_{l,v_l} \in \mathbb{R}^{M_l}$. Each pre-quantizer PQ_l is a $|\mathcal{V}_l|$-level VQ that partitions the M_l-dimensional measurement vector space determined by (86) into *cells* $\mathcal{S}_{l,1}, \ldots, \mathcal{S}_{l,|\mathcal{V}_l|}$, i.e., $\mathcal{S}_{l,v_l} \cap \mathcal{S}_{l,v_l'} = \emptyset$, $v_l \neq v_l' \in \mathcal{V}_l$, and $\bigcup_{v_l=1}^{|\mathcal{V}_l|} \mathcal{S}_{l,v_l} = \mathbb{R}^{M_l}$. Thus, PQ_l is a lossy mapping

$$\mathsf{PQ}_l : \mathbb{R}^{M_l} \to \mathcal{V}_l,\ l = 1, 2, \tag{87}$$

i.e., for a given measurement realization, it assigns a cell index as $\mathsf{PQ}_l(\mathbf{y}_l) = v_l \in \mathcal{V}_l$, if $\mathbf{y}_l \in \mathcal{S}_{l,v_l}$.

Remark 6.1. *The codepoints $\mathbf{g}_{l,1}, \ldots, \mathbf{g}_{l,|\mathcal{V}_l|}$ are intermediate quantities for the actual encoding at E_l. That is, their main purpose is to determine to which cell $\mathcal{S}_{l,v_l}$ each realization $\mathbf{y}_l$, $l = 1, 2$, belongs. This will become more clear in the encoding rule defined in Section 6. In conclusion, the quantized version of random vector $\mathbf{Y}_l$ is never reconstructed in the system.*

Encoding and Decoding

The outputs of each PQ_l, i.e., the cell indices $v_l \in \mathcal{V}_l$, are fed to encoder E_l at each sensor $l = 1, 2$. We follow a customary approach and model E_l as the concatenation of a (lossy) quantizer and a lossless entropy encoder [108,

[11] Pre-quantization has also a pragmatic aspect: prior to source encoding, the sensor inputs are necessarily discretized with an ADC in any digital sensor.

244]. Accordingly, each sensor encodes the cell indices into message indices, and further, into binary source codewords. The sink uses the received pairs of codewords to jointly reconstruct estimates of $\mathbf{X}_1$ and $\mathbf{X}_2$. The next subsections describe the distributed quantizer blocks; the treatment of entropy coding is presented in Section 6.

Separate Encoders

Let $\mathcal{I}_l \triangleq \{1, \ldots, |\mathcal{I}_l|\}$ be a set of *message indices* $i_l \in \mathcal{I}_l$ for sensor $l = 1, 2$. Let $\mathcal{H}_l \triangleq \{\mathbf{h}_{l,1}, \ldots, \mathbf{h}_{l,|\mathcal{H}_l|}\}$, $|\mathcal{H}_l| = |\mathcal{I}_l|$, be a *source codebook* consisting of binary *codewords*. Each encoder is a composite mapping $\mathsf{E}_l : \alpha_l \circ \pi_l$ as

$$\mathsf{E}_l : \mathcal{V}_l \rightarrow \mathcal{I}_l \rightarrow \mathcal{H}_l, \; l = 1, 2. \tag{88}$$

While the number of pre-quantization cells $|\mathcal{V}_l|$ is not restricted in the design, $|\mathcal{I}_l| < |\mathcal{V}_l|$ is used for all practical purposes.

The first mapping $\pi_l : \mathcal{V}_l \rightarrow \mathcal{I}_l$, termed the *message index mapping*, maps each cell index $v_l \in \mathcal{V}_l$ into a message index $i_l \in \mathcal{I}_l$, i.e., $\pi_l = \{\pi_l(1), \ldots, \pi_l(|\mathcal{V}_l|)\}$, where $\pi_l(v_l) \in \mathcal{I}_l$. Given π_l, each $i_l \in \mathcal{I}_l$ is associated with a set of cell indices mapped to itself, i.e., the inverse image $\pi_l^{-1}(i_l) = \{v_l \in \mathcal{V}_l | \pi_l(v_l) = i_l\}$. Note that for $|\mathcal{I}_l| < |\mathcal{V}_l|$, π_l is a many-to-one mapping, i.e., it performs *lossy* compression of the cell index V_l. This accounts for the prefix "pre" for PQ_l, as each sensor has a concatenation of two quantizers: the VQ of PQ_l and the message index mapping π_l in E_l. Interconnections between the indices of PQ_l and E_l are illustrated in Fig. 23. Note that the pre-quantization allows assigning non-contiguous cells to the same message index, which is conducive to compression performance.

For a given entropy code, the second mapping $\alpha_l : \mathcal{I}_l \rightarrow \mathcal{H}_l$ is a one-to-one *lossless* mapping from the set of message indices to binary source codewords $\mathbf{h}_{l,i_l} \in \mathcal{H}_l$, i.e., $\alpha_l(i_l) = \mathbf{h}_{l,i_l}$, $i_l \in \mathcal{I}_l$. Fixed-to-variable-length coding is assumed, i.e., each index $i_l \in \mathcal{I}_l$ is mapped to one codeword at a time, whereas the binary representations of $\mathbf{h}_{l,i_l} \in \mathcal{H}_l$ have, in general, different lengths.

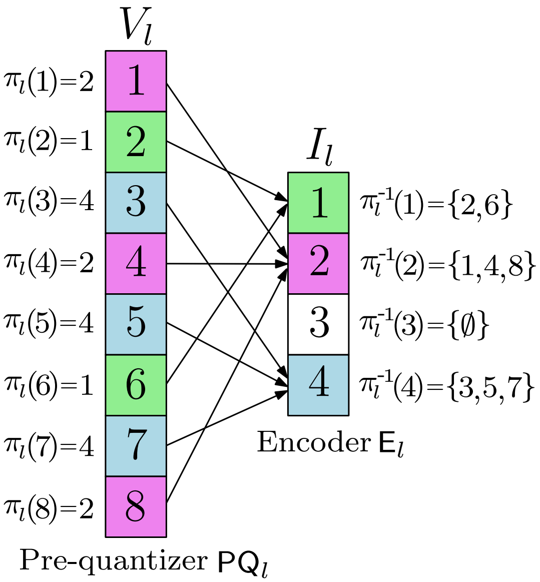


Figure 23: The relationships between the cell indices of PQ_l and message indices of E_l for $|\mathcal{V}_l| = 8$ and $|\mathcal{I}_l| = 4$, $l = 1, 2$. Note that the message index $I_l = 3$ is assigned no cell index $v_l \in \mathcal{V}_l$.

Joint Decoder

The joint decoder comprises of two composite mappings $\mathsf{D} : \{\beta_1 \circ \alpha_1^{-1}, \beta_2 \circ \alpha_2^{-1}\}$ as

$$\mathsf{D} : (\mathcal{H}_1 \times \mathcal{H}_2) \to (\mathcal{I}_1 \times \mathcal{I}_2) \to \mathcal{C}_l,\ l = 1, 2. \tag{89}$$

The first mappings $\alpha_l^{-1} : \mathcal{H}_l \to \mathcal{I}_l$, $l = 1, 2$, decode the entropy codes, i.e., each received codeword $\mathbf{h}_{l,i_l} \in \mathcal{H}_l$ is used to recover the (transmitted) message index $i_l \in \mathcal{I}_l$. By this convention, we confine here to the separate/independent coding[12] of the message indices I_1 and I_2. Owing to the information lossless property, the pairs $(\alpha_1, \alpha_1^{-1})$ and $(\alpha_2, \alpha_2^{-1})$ constitute uniquely decodable codes.

As for the second mappings $\beta_l : (\mathcal{I}_1 \times \mathcal{I}_2) \to \mathcal{C}_l$, the decoded message indices are used to jointly reconstruct estimates of sources

[12] Separate coding enables low-complexity and low-delay coding as the pairs $(\alpha_l, \alpha_l^{-1})$, $l = 1, 2$, can be chosen to constitute two *instantaneous* lossless source codes.

$\mathbf{X}_1$ and $\mathbf{X}_2$ as $\hat{\mathbf{x}}_1 := \beta_1(i_1, i_2) = \mathbf{c}_{1,i_1,i_2}$ and $\hat{\mathbf{x}}_2 := \beta_2(i_1, i_2) = \mathbf{c}_{2,i_1,i_2}$, where $\mathbf{c}_{l,i_1,i_2} \in \mathbb{R}^N$ is the *codevector* of a *reconstruction codebook* $\mathcal{C}_l \triangleq \{\mathbf{c}_{l,1,1}, \ldots, \mathbf{c}_{l,|\mathcal{I}_1|,|\mathcal{I}_2|}\}$, $|\mathcal{C}_l| = |\mathcal{I}_1||\mathcal{I}_2|$, $l = 1, 2$. Thus, the decoder D performs a single operation to 1) take account of the quantization/encoding steps applied to $\mathbf{Y}_1$ and $\mathbf{Y}_2$, and 2) reconstruct the signal estimates of $\mathbf{X}_1$ and $\mathbf{X}_2$, given the underlying distributed CS setup.

Entropy Coding

The pairs of mappings $(\alpha_1, \alpha_1^{-1})$ and $(\alpha_2, \alpha_2^{-1})$ can realize different entropy coding classes in the system. An appropriate choice for the used source code depends on application requirements such as the reconstruction fidelity and maximum allowed delay (i.e., coding block length), and implementation factors such as the sensors' computation and memory capabilities. A unified framework for subsuming entropy coding in the distributed quantizer design is discussed next.

Average Rate

Let R_l be the average encoding rate in bits/vector $\mathbf{Y}_l$ for sensor $l = 1, 2$. Thus, the average sum rate of the sensors is

$$\underline{R} \triangleq R_1 + R_2. \tag{90}$$

In practice, R_l is quantified by the average codeword length of the codebook $\mathcal{H}_l$, i.e.,

$$R_l \triangleq \mathbb{E}[\gamma(\cdot)] = \textstyle\sum_{i_l=1}^{|\mathcal{I}_l|} p(i_l)\gamma(i_l),\ l = 1, 2, \tag{91}$$

where $\gamma(i_l)$ is the length of codeword $\mathbf{h}_{l,i_l} \in \mathcal{H}_l$, and $p(i_l) \triangleq \Pr(I_l = i_l)$ is the probability of index $i_l \in \mathcal{I}_l$. An alternative rate definition is presented in the following.

Rate Measure

Following the approaches in [66, 108, 244] and [243, Sect. 4.2], instead of using (91), the average rate is approximated via the *entropy bound* of a source code[13]. Let $r(p(i_l))$ be a *rate measure* which is a function of the message

[13] The design of practical codes is beyond the scope of this work.

index probabilities $p(i_l)$, $i_l \in \mathcal{I}_l$, $l = 1, 2$. Accordingly, $\underline{R}$ in (90) is given as the expectation of the rate measures, i.e.,

$$\underline{R} \triangleq \textstyle\sum_{l=1}^{2} \sum_{i_l=1}^{|\mathcal{I}_l|} p(i_l) r(p(i_l)). \tag{92}$$

As a major benefit, the rate definition in (92) permits flexible treatment of the entropy coding without tying the design to any particular source code.

Entropy bounds and their associated rate measures for various coding settings have been addressed in, e.g., [244] and [243, Sect. 4.2]. For the considered separate coding, $r(p(i_l)) = -\text{log}_2 p(i_l)$, $i_l \in \mathcal{I}_l$, and, thus, the average sum rate is approximated as $\underline{R} = H(I_1) + H(I_2)$, where $H(I_l) = -\sum_{i_l=1}^{|\mathcal{I}_l|} p(i_l)\text{log}_2 p(i_l)$ is the entropy of message index I_l, $l = 1, 2$.

Distributed Variable-Rate QCS Method

This section describes a distortion-rate (DR) optimization framework for the variable-rate communication system of Fig. 22. We devise a distributed QCS method for the efficient acquisition of the correlated sparse sources $\mathbf{X}_1$ and $\mathbf{X}_2$ under complexity-constrained encoding. Practical training algorithms are presented, and implementation and complexity aspects are discussed.

Problem Formulation

Let $\hat{\mathbf{X}}_l$ be a length-N random vector that represents the estimate of source $\mathbf{X}_l$, $l = 1, 2$, at the output of decoder D. The average sum MSE reconstruction distortion is defined as

$$\underline{D} \triangleq D_1 + D_2 = \textstyle\sum_{l=1}^{2} \mathbb{E}[\|\mathbf{X}_l - \hat{\mathbf{X}}_l\|_2^2], \tag{93}$$

where the expectation $\mathbb{E}[\cdot]$ is taken over the distributions of $\mathbf{X}_l$ and $\mathbf{W}_l$, $l = 1, 2$. Furthermore, let $\mathfrak{L}_\mu(\underline{D}, \underline{R})$ be a weighted DR cost function as

$$\mathfrak{L}_\mu(\underline{D}, \underline{R}) \triangleq (1 - \mu)\underline{D} + \mu\underline{R}, \tag{94}$$

where $\mu \in [0, 1]$ is a weighting parameter for adjusting the DR trade-off, and $\underline{R}$ is the average sum rate given in (92).

Our design objective is to minimize $\mathfrak{L}_\mu(\underline{D}, \underline{R})$ in (94) for a given μ by optimizing the pre-quantizers PQ_l (i.e., the pre-quantization codebooks $\mathcal{G}_l$), the encoders E_l (i.e., the message index mappings π_l), and the decoder D (i.e., the reconstruction codebooks $\mathcal{C}_l$) so that the pre-quantization cells $\mathcal{S}_{l,v_l}$, $v_l \in \mathcal{V}_l$, at each PQ_l, $l = 1, 2$, satisfy a *nearest-neighbor* (NN) condition [122]. This structural constraint is the key to restrain the sensors' encoding complexity. Since the joint optimization of all these blocks is intractable, the design is split into two steps: first, each PQ_l is optimized under the NN constraints (Section 6), followed by the optimization of E_1, E_2, and D for fixed pre-quantizers (Section 6). Despite the sub-optimality, the approach is shown to yield satisfactory empirical performance in Section 6.

Note that the rate term $\mu \underline{R}$ in (94) eliminates the need of constructing specific codebooks $\mathcal{H}_1$ and $\mathcal{H}_2$ in the optimization phase. As demonstrated in [66] (for a point-to-point case), decent performance is achievable by subsuming rate measures $r(p(i_l)) = -\log_2 p(i_l)$, $i_l \in \mathcal{I}_l$, in the optimization loop, followed by a source code whose average codeword length is close to the index entropy [66]. These include Huffman codes [149] and [74, Sect. 5.6] and arithmetic codes [247]. Note that $\mu = 0$ realizes a minimum-distortion fixed-rate method with $R_l = \log_2 |\mathcal{I}_l|$, $l = 1, 2$.

Optimization of Pre-Quantizers

At each PQ_l, the cells $\mathcal{S}_{l,v_l}$ and codepoints $\mathbf{g}_{l,v_l}$, $v_l \in \mathcal{V}_l$, of the $|\mathcal{V}_l|$-level VQ are optimized to minimize the MSE distortion induced by discretizing the measurement vector space (see (86)), i.e., the distortion

$$D_l^{\mathrm{pq}} \triangleq \textstyle\sum_{v_l=1}^{|\mathcal{V}_l|} p(v_l) \mathbb{E}\big[\|\mathbf{Y}_l - \mathbf{g}_{l,v_l}\|_2^2 | V_l = v_l\big], \; l = 1, 2. \tag{95}$$

This approach inherently results in the required NN encoding (cf. (96)). Since finding the *globally* optimal partition and codebook of a quantizer is intractable, the common alternating optimization technique [211, 107, 192, 188, 317, 244, 294, 265, 266] is adopted to derive the necessary optimality conditions. Such conditions serve as a practical means to train each PQ_l, $l = 1, 2$, via principles of the iterative Linde-Buzo-Gray (LBG) algorithm [188], elaborated in Section 6.

For given codepoints $\mathbf{g}_{l,v_l}$, $v_l \in \mathcal{V}_l$, the optimal cells which minimize D_l^{pq} satisfy the NN condition [188],

$$\mathcal{S}_{l,v_l}^* = \Big\{\mathbf{y}_l : \|\mathbf{y}_l - \mathbf{g}_{l,v_l}\|_2^2 \leq \|\mathbf{y}_l - \mathbf{g}_{l,v_l'}\|_2^2, \; \forall v_l' \neq v_l\Big\}, \; v_l \in \mathcal{V}_l, \tag{96}$$

i.e., the cells form a Voronoi partition [134, Sect. 5.1]. For given cells $\mathcal{S}_{l,v_l}$, $v_l \in \mathcal{V}_l$, the optimal codepoints satisfy the centroid condition [188]

$$\mathbf{g}^*_{l,v_l} = \mathbb{E}[\mathbf{Y}_l | V_l = v_l] = \frac{1}{p(v_l)} \int_{\mathbf{y}_l \in \mathcal{S}_{l,v_l}} \mathbf{y}_l f(\mathbf{y}_l) \mathrm{d}\mathbf{y}_l, \; v_l \in \mathcal{V}_l, \tag{97}$$

where $f(\mathbf{y}_l)$ is the probability density function (PDF) of $\mathbf{Y}_l$.

Alternating Optimization of Encoders & Decoder

In this section, the encoders E_1 and E_2, and decoder D are optimized to minimize $\mathfrak{L}_\mu(\underline{D}, \underline{R})$ in (94) for fixed PQ_1 and PQ_2. The optimization involves six sets of optimization variables: the message index mappings π_1 and π_2, the rate measures $r(p(i_1))$ and $r(p(i_2))$, and the codevectors $\mathbf{c}_{1,i_1,i_2}$ and $\mathbf{c}_{2,i_1,i_2}$, $i_1 \in \mathcal{I}_1$, $i_2 \in \mathcal{I}_2$. Due to the intractability of joint optimization, the alternating optimization principles (see Section 6) are applied, and consequently, the necessary optimality conditions for each variable set while keeping the others fixed are derived. The implementation aspects are detailed in Section 6.

Message Index Mappings

The minimization of $\mathfrak{L}_\mu(\underline{D}, \underline{R})$ over the message index mapping π_l is independent of $r(p(i_{l'}))$, $i_{l'} \in \mathcal{I}_{l'}$, $l \neq l'$. Hence, for fixed $r(p(i_l))$, $i_l \in \mathcal{I}_l$, $\mathcal{C}_1$, $\mathcal{C}_2$, and $\pi_{l'}$, the optimal message index mapping π^*_l for sensor $l \neq l'$ is the one that minimizes $\mathfrak{L}_\mu(\underline{D}, \underline{R})$. The distortion term $\underline{D}$ of $\mathfrak{L}_\mu(\underline{D}, \underline{R})$ can be reformulated as

$$\begin{aligned}
\underline{D} = {} & \textstyle\sum_{l=1}^{2} \sum_{v_1 \in \mathcal{V}_1} \sum_{v_2 \in \mathcal{V}_2} \displaystyle\int_{\mathbf{y}_1} \int_{\mathbf{y}_2} p(v_1, v_2 | \mathbf{y}_1, \mathbf{y}_2) \\
& \mathbb{E}[\|\mathbf{X}_l - \mathbf{c}_{l,\pi_1(v_1),\pi_2(v_2)}\|_2^2 | V_1 = v_1, V_2 = v_2, \ldots \\
& \mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2] f(\mathbf{y}_1, \mathbf{y}_2) \mathrm{d}\mathbf{y}_1 \mathrm{d}\mathbf{y}_2 \\
\overset{(a)}{=} {} & \textstyle\sum_{l=1}^{2} \sum_{v_1 \in \mathcal{V}_1} \sum_{v_2 \in \mathcal{V}_2} \displaystyle\int_{\mathbf{y}_1 \in \mathcal{S}_{1,v_1}} \int_{\mathbf{y}_2 \in \mathcal{S}_{2,v_2}} \\
& \Big\{ \mathbb{E}[\|\mathbf{X}_l\|_2^2 | \mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2] + \|\mathbf{c}_{l,\pi_1(v_1),\pi_2(v_2)}\|_2^2 - \\
& 2\mathbf{c}^{\mathrm{T}}_{l,\pi_1(v_1),\pi_2(v_2)} \mathbb{E}[\mathbf{X}_l | \mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2] \Big\} f(\mathbf{y}_1, \mathbf{y}_2) \mathrm{d}\mathbf{y}_1 \mathrm{d}\mathbf{y}_2 \\
\overset{(b)}{=} {} & \textstyle\sum_{l=1}^{2} \sum_{v_1 \in \mathcal{V}_1} \sum_{v_2 \in \mathcal{V}_2} p(v_1, v_2) \Big\{ \mathbb{E}[\|\mathbf{X}_l\|_2^2 | V_1 = v_1, V_2 = v_2] + \\
& \|\mathbf{c}_{l,\pi_1(v_1),\pi_2(v_2)}\|_2^2 - 2\mathbf{c}^{\mathrm{T}}_{l,\pi_1(v_1),\pi_2(v_2)} \mathbf{z}_{l,v_1,v_2} \Big\}
\end{aligned} \tag{98}$$

where (a) follows from i) the Markov properties[14] $V_l \to \mathbf{Y}_l \to \mathbf{Y}_{l'} \to V_{l'}$, $\mathbf{X}_l \to (V_1, V_2) \to \mathcal{C}_l$, $\mathbf{X}_l \to \mathbf{Y}_l \to V_l$, and $\mathbf{X}_l \to \mathbf{Y}_{l'} \to V_{l'}$, $l \neq l'$, and ii) the fact that $p(v_l|\mathbf{y}_l) = 1$, if $\mathbf{y}_l \in \mathcal{S}_{l,v_l}$, and 0 otherwise, $l = 1, 2$; (b) follows from $p(v_1, v_2) = \int_{\mathbf{y}_1 \in \mathcal{S}_{1,v_1}} \int_{\mathbf{y}_2 \in \mathcal{S}_{2,v_2}} f(\mathbf{y}_1, \mathbf{y}_2) \mathrm{d}\mathbf{y}_1 \mathrm{d}\mathbf{y}_2$, and the definition of a vector $\mathbf{z}_{l,v_1,v_2} \in \mathbb{R}^N$ as

$$\begin{aligned} \mathbf{z}_{l,v_1,v_2} &\triangleq \mathbb{E}\left[\mathbf{X}_l | V_1 = v_1, V_2 = v_2\right], \; l = 1, 2 \\ &= \frac{1}{p(v_1, v_2)} \int_{\mathbf{y}_1 \in \mathcal{S}_{1,v_1}} \int_{\mathbf{y}_2 \in \mathcal{S}_{2,v_2}} \\ &\quad \mathbb{E}[\mathbf{X}_l | \mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2] f(\mathbf{y}_1, \mathbf{y}_2) \mathrm{d}\mathbf{y}_1 \mathrm{d}\mathbf{y}_2, \end{aligned} \tag{99}$$

which represents the *centroid* of the minimum mean square error (MMSE) estimates of source $\mathbf{X}_l$, $l = 1, 2$, for those measurement realizations $\mathbf{y}_1$ and $\mathbf{y}_2$ that are pre-quantized to the cell index pair (v_1, v_2), $v_1 \in \mathcal{V}_1$, $v_2 \in \mathcal{V}_2$, at PQ_1 and PQ_2.

Remark 6.2. *A closed-form expression of the MMSE estimate* $\mathbb{E}[\mathbf{X}_l | \mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2]$, $l = 1, 2$, *in* (99) *has been derived for a similar signal setup in, e.g., [266, Proposition 1]. Nevertheless, as it will be pointed out later, computation of these complex estimates is obviated by means of the pre-quantizers* PQ_1 *and* PQ_2 *both in the offline training and online communication phase.*

Finally, the rate term $\underline{R}$ in $\mathfrak{L}_\mu(\underline{D}, \underline{R})$ can be extended as

$$\begin{aligned} \underline{R} &= \textstyle\sum_{l=1}^{2} \sum_{i_l \in \mathcal{I}_l} p(i_l) r(p(i_l)) \\ &= \textstyle\sum_{l=1}^{2} \sum_{v_l \in \mathcal{V}_l} p(v_l) r(p(\pi_l(v_l))). \end{aligned} \tag{100}$$

By combining (98) and (100), and dropping the unnecessary terms $\mathbb{E}[\|\mathbf{X}_l\|_2^2 | V_1 = v_1, V_2 = v_2]$ in the minimization, finding the optimal message index mapping $\pi_l^* = \{\pi_l^*(1), \ldots, \pi_l^*(|\mathcal{V}_l|)\}$ for each sensor $l = 1, 2$ separates into $|\mathcal{V}_l|$ subproblems. Accordingly, the optimal message index for a v_1th cell

[14] Random variables X, Y, and Z form a Markov chain $Z \to Y \to X$ (in this order) if X and Z are conditionally independent given Y [74, Sect. 2.8]. Consequently, $p(x, y, z) = p(x)p(y|x)p(z|y)$.

of sensor 1 is given by the minimization problem

$$\begin{aligned}\pi_1^*(v_1) = \underset{\pi_1(v_1)\in\mathcal{I}_1}{\operatorname{argmin}} \Big\{ & (1-\mu)\textstyle\sum_{l=1}^{2}\sum_{v_2\in\mathcal{V}_2} p(v_1,v_2) \\ & \Big(\|\mathbf{c}_{l,\pi_1(v_1),\pi_2(v_2)}\|_2^2 - 2\mathbf{c}^{\mathrm{T}}_{l,\pi_1(v_1),\pi_2(v_2)}\mathbf{z}_{l,v_1,v_2}\Big) + \\ & \mu p(v_1) r\big(p(\pi_1(v_1))\big)\Big\},\ \forall v_1 \in \mathcal{V}_1.\end{aligned} \tag{101}$$

Each optimal message index $\pi_2^*(v_2)$, $v_2 \in \mathcal{V}_2$, for sensor 2 is found similarly by swapping the roles of the sensor indices.

Rate Measures

The rate measure update follows the procedures in [66, 244]. Accordingly, for fixed π_l, the optimal rate measures for each sensor $l = 1, 2$ are given as

$$\begin{aligned} r^*(p(i_l)) &= -\log_2 p(i_l),\ \forall i_l \in \mathcal{I}_l, \\ &= -\log_2 \textstyle\sum_{v_l \in \pi_l^{-1}(i_l)} p(v_l). \end{aligned} \tag{102}$$

By (102), the average sum rate (92) becomes $\underline{R} = H(I_1) + H(I_2)$, where $H(I_l) = -\sum_{i_l=1}^{|\mathcal{I}_l|} p(i_l)\log_2 p(i_l)$ is the entropy of I_l, $l = 1, 2$.

Reconstruction Codebooks

The minimization of $\mathfrak{L}_\mu(\underline{D}, \underline{R})$ in (94) with respect to the codebook $\mathcal{C}_l$ is independent of $\underline{R}$, and $\mathcal{C}_{l'}$, $l' \neq l$. Accordingly, for a fixed π_1 and π_2, the optimal reconstruction codebook $\mathcal{C}_l^* = \{\mathbf{c}^*_{l,1,1}, \ldots, \mathbf{c}^*_{l,|\mathcal{I}_1|,|\mathcal{I}_2|}\}$ for source $l = 1, 2$ is found from solving $|\mathcal{I}_1||\mathcal{I}_2|$ separate optimization problems for each index pair (i_1, i_2), $i_1 \in \mathcal{I}_1$, $i_2 \in \mathcal{I}_2$, as

$$\mathbf{c}^*_{l,i_1,i_2} = \underset{\mathbf{c}_{l,i_1,i_2}\in\mathbb{R}^N}{\operatorname{argmin}}\ \mathbb{E}\left[\|\mathbf{X}_l - \mathbf{c}_{l,i_1,i_2}\|_2^2 | I_1 = i_1, I_2 = i_2\right]. \tag{103}$$

From (103), each $\mathbf{c}^*_{l,i_1,i_2}$ is given by the MMSE estimate of source $\mathbf{X}_l$, $l = 1, 2$, given the message index pair (i_1, i_2) [266], i.e.,

$$\begin{aligned}
\mathbf{c}^*_{l,i_1,i_2} &= \mathbb{E}\left[\mathbf{X}_l | I_1 = i_1, I_2 = i_2\right], \ \forall i_1 \in \mathcal{I}_1,\ i_2 \in \mathcal{I}_2 \\
&= \textstyle\sum_{v_1 \in \mathcal{V}_1} \sum_{v_2 \in \mathcal{V}_2} p(v_1, v_2 | i_1, i_2) \\
&\quad\ \mathbb{E}\left[\mathbf{X}_l | I_1 = i_1, I_2 = i_2, V_1 = v_1, V_2 = v_2\right] \\
&\overset{(a)}{=} \textstyle\sum_{v_1 \in \mathcal{V}_1} \sum_{v_2 \in \mathcal{V}_2} \dfrac{p(v_1, v_2) p(i_1, i_2 | v_1, v_2)}{p(i_1, i_2)} \mathbb{E}\left[\mathbf{X}_l | V_1 = v_1, V_2 = v_2\right] \\
&\overset{(b)}{=} \textstyle\sum_{v_1 \in \pi_1^{-1}(i_1)} \sum_{v_2 \in \pi_2^{-1}(i_2)} \dfrac{p(v_1, v_2)}{p(i_1, i_2)} \mathbb{E}\left[\mathbf{X}_l | V_1 = v_1, V_2 = v_2\right] \\
&\overset{(c)}{=} \frac{\sum_{v_1 \in \pi_1^{-1}(i_1)} \sum_{v_2 \in \pi_2^{-1}(i_2)} p(v_1, v_2) \mathbf{z}_{l,v_1,v_2}}{\sum_{v_1 \in \pi_1^{-1}(i_1)} \sum_{v_2 \in \pi_2^{-1}(i_2)} p(v_1, v_2)},
\end{aligned} \tag{104}$$

where (a) follows from the Markov properties $\mathbf{X}_l \rightarrow V_l \rightarrow I_l$ and $\mathbf{X}_l \rightarrow V_{l'} \rightarrow I_{l'}$, $l \neq l'$; (b) follows from the Markov property $I_l \rightarrow V_l \rightarrow V_{l'} \rightarrow I_{l'}$, $l \neq l'$, and the fact that $p(i_l | v_l) = 1$, if $\pi_l(v_l) = i_l$, and 0 otherwise; (c) follows from (99).

Algorithm Implementation

Practical implementation aspects of the proposed method, termed DQCS-PQ, are elaborated next. Two offline training algorithms and an algorithm for an online communication phase are presented. Each algorithm's complexity in terms of computational and memory requirements are also discussed.

Offline Training Phase

An *offline* training algorithm for optimizing each PQ_l, $l = 1, 2$, is described in Algorithm 7, and the algorithm for optimizing E_1, E_2, and D is presented in Algorithm 8. The foundation of both algorithms is the iterative Lloyd algorithm [192, 188]: in Algorithm 7, each PQ_l, $l = 1, 2$, is optimized by successively applying the necessary optimality conditions (96) and (97) (Steps 1) and 2)). Similarly, the optimization of E_1, E_2, and D in Algorithm 8 relies on the six-step iteration loop[15], in which the optimality conditions (101), (102), and (104) (Steps 1) – 3)) are first applied in respect to sensor 1, and then, in respect to sensor 2. Once Algorithm 8 has converged, the resulting message index

[15] Other optimization orders within the sequential algorithm could be considered as well.

Algorithm 7 Practical training algorithm for PQ_l, $l = 1, 2$ (offline).

```
Inputs: a) CS matrix $\mathbf{\Phi}_l$; b) measurement training vectors $\{\mathbf{y}_l^{(1)}, \mathbf{y}_l^{(2)}, \ldots\}$; c) pre-quantization rate $\bar{R}_l$.
Initialization: Initial codebook $\mathcal{G}_l = \{\mathbf{g}_{l,1}, \ldots, \mathbf{g}_{l,|\mathcal{V}_l|}\}$.
Repeat
1) For given $\mathcal{G}_l$, find the optimal cells $\mathcal{S}^*_{l,v_l}$, $v_l \in \mathcal{V}_l$, by classifying the vectors $\{\mathbf{y}_l^{(1)}, \mathbf{y}_l^{(2)}, \ldots\}$ according to (96).
2) For a given $\mathcal{S}_{l,v_l}$, $v_l \in \mathcal{V}_l$, compute the optimal codepoints $\mathbf{g}^*_{l,v_l}$, $v_l \in \mathcal{V}_l$, as the conditional expectations in (97).
until convergence
Output: Pre-quantization codebook $\mathcal{G}_l = \{\mathbf{g}_{l,1}, \ldots, \mathbf{g}_{l,|\mathcal{V}_l|}\}$.
```

probabilities $p(i_l)$, $i_l \in \mathcal{I}_l$, $l = 1, 2$, are used to generate the desired binary source codebooks $\mathcal{H}_1$ and $\mathcal{H}_2$.

The requisite training data sets for Algorithms 7 and 8 are generated as follows. Source and noise samples $\{\mathbf{x}_l^{(1)}, \mathbf{x}_l^{(2)}, \ldots\}$ and $\{\mathbf{w}_l^{(1)}, \mathbf{w}_l^{(2)}, \ldots\}$ are created by sampling from their respective distributions. Then, the measurement training vectors $\{\mathbf{y}_l^{(1)}, \mathbf{y}_l^{(2)}, \ldots\}$ generated according to (86) are used to optimize each PQ_l, $l = 1, 2$. The MMSE estimate centroids $\mathbf{z}_{l,v_1,v_2}$, $v_1 \in \mathcal{V}_1$, $v_2 \in \mathcal{V}_2$, for Algorithm 8 are pre-computed as conditional expectations (99) using the source samples $\{\mathbf{x}_l^{(1)}, \mathbf{x}_l^{(2)}, \ldots\}$ and the indices $\{v_l^{(1)}, v_l^{(2)}, \ldots\}$ obtained from quantizing $\{\mathbf{y}_l^{(1)}, \mathbf{y}_l^{(2)}, \ldots\}$ via PQ_l, $l = 1, 2$. The multi-dimensional integrals in (97) and (104) are similarly evaluated via Monte Carlo integration techniques.

Convergence: The convergence of the iterative descent algorithms in Algorithms 7 and 8 rests on the rationale behind the Lloyd and LBG algorithms [211, 192, 188]; at each iteration step, the objective function value either decreases or remains the same, and thus, the algorithm converges. However, since only necessary but not the sufficient optimality conditions are met, the resulting quantization system is, at best, locally optimal [188, 66, 108, 294]. LBG type algorithms are known to be sensitive to initialization, making them susceptible to poor local minima. This accentuates the importance of proper initialization of system blocks (i.e., π_2, $\mathcal{C}_1$, and $\mathcal{C}_2$). One technique to mitigate this problem is *the splitting method* [188], which we use in the simulations

Algorithm 8 Practical training algorithm for E_1, E_2, and D (offline).

```
Inputs: a) CS matrices $\mathbf{\Phi}_l$; b) pre-quantization rates $\bar{R}_l$; c) codebook sizes $|\mathcal{I}_l|$; d) rate measures $r(p(i_l))$, $i_l \in \mathcal{I}_l$; e) weight parameter $\mu \in [0,1]$; f) MMSE estimate centroids $\mathbf{z}_{l,v_1,v_2}$, $v_1 \in \mathcal{V}_1$, $v_2 \in \mathcal{V}_2$, $l = 1,2$.
Initializations: a) Initial codebooks $\mathcal{C}_l = \{\mathbf{c}_{l,1,1}, \ldots, \mathbf{c}_{l,|\mathcal{I}_1|,|\mathcal{I}_2|}\}$, $l = 1,2$; b) initial message index mapping $\pi_2 = \{\pi_2(1), \ldots, \pi_2(|\mathcal{V}_2|)\}$.
Repeat
for $l = 1,2$ do
    1) For given $\mathcal{C}_1$, $\mathcal{C}_2$, $\pi_{l'}$, $l' \neq l$, and $r(p(i_l))$, $i_l \in \mathcal{I}_l$, find the optimal message index mapping $\pi_l^* = \{\pi_l^*(1), \ldots, \pi_l^*(|\mathcal{V}_l|)\}$ via (101).
    2) For given $\pi_l$, update the rate measures $r(p(i_l))$, $i_l \in \mathcal{I}_l$, according to (102).
    3) For given $\pi_1$ and $\pi_2$, compute the optimal codevectors $\mathbf{c}^*_{l,i_1,i_2}$, $i_1 \in \mathcal{I}_1$, $i_2 \in \mathcal{I}_2$, as the conditional expectations in (104).
end for
until convergence
Generate the binary source codebooks $\mathcal{H}_l$ using the index probabilities $p(i_l)$, $i_l \in \mathcal{I}_l$, $l = 1,2$.
Outputs: Message index mappings $\pi_l = \{\pi_l(1), \ldots, \pi_l(|\mathcal{V}_l|)\}$; source codebooks $\mathcal{H}_l = \{\mathbf{h}_{l,1}, \ldots, \mathbf{h}_{l,|\mathcal{H}_l|}\}$; reconstruction codebooks $\mathcal{C}_l = \{\mathbf{c}_{l,1,1}, \ldots, \mathbf{c}_{l,|\mathcal{I}_1|,|\mathcal{I}_2|}\}$; encoder mappings $\alpha_l$; decoder mappings $\alpha_l^{-1}$ and $\beta_l$, $l = 1,2$.
```

in Section 6. Other robust designs include a deterministic annealing method for fixed-rate VQ design in [250], and its extension as an entropy-constrained design in [251].

Remark 6.3. *The entropy-constrained design has a built-in tendency to reduce the codebook sizes, especially for large values of μ [66]. Namely, as Algorithm 8 proceeds, some message index, say $\breve{i}_1 \in \mathcal{I}_1$, may become unpopulated (i.e., $\pi_1^{-1}(\breve{i}_1) = \emptyset$) in Step 1). Consequently, the rate measure in Step 2) becomes $r(p(\breve{i}_1)) = \infty$, and the codevectors $\mathbf{c}_{1,\breve{i}_1,1}, \ldots, \mathbf{c}_{1,\breve{i}_1,|\mathcal{I}_2|}$ and $\mathbf{c}_{2,\breve{i}_1,1}, \ldots, \mathbf{c}_{2,\breve{i}_1,|\mathcal{I}_2|}$ in Step 3) become undefined. It was conjectured in [66] that, unlike for a fixed-rate quantizer, there is no rationale to re-include such unassigned indices in the subsequent iterations.*

Algorithm 9 Distributed variable-rate QCS method with complexity-constrained encoders (DQCS-PQ) (online).

```
Inputs: a) CS matrices $\mathbf{\Phi}_l$; b) PQ$_l$: codebook $\mathcal{G}_l$; c) E$_l$: mappings $\pi_l$ and $\alpha_l$, and codebook $\mathcal{H}_l$; d) D: mappings $\alpha^{-1}$ and $\beta_l$, and codebooks $\mathcal{H}_l$ and $\mathcal{C}_l$, $l = 1, 2$.
Separate encoding at sensor $l = 1, 2$:
(1) Acquisition of measurements $\mathbf{y}_l$ according to (86).
(2) PQ$_l$: Assignment of pre-quantization index $v_l^*$ via (105) using $\mathcal{G}_l$.
(3) E$_l$: a) Assignment of message index as $i_l^* = \pi_l(v_l^*)$;
b) Assignment of binary codeword as $\mathbf{h}_{l,i_l^*} = \alpha_l(i_l^*)$.
Joint decoding at the sink:
(4) D: a) Decoding of binary codewords as $(i_1^*, i_2^*) = (\alpha_1^{-1}(\mathbf{h}_{1,i_1^*}), \alpha_2^{-1}(\mathbf{h}_{2,i_2^*}))$; b) Joint reconstruction of source estimates as $\hat{\mathbf{x}}_l = \beta_l(i_1^*, i_2^*) = \mathbf{c}_{l,i_1^*,i_2^*}$ using $\mathcal{C}_l$, $l = 1, 2$.
```

Training Complexity: We assess the computational and memory requirements by estimating the number of involved training vectors, summands, and variables (i.e., indices) in one iteration loop of an algorithm. The complexity of optimizing the cells of each PQ$_l$, $l = 1, 2$, via (96) scales as the number of levels $|\mathcal{V}_l|$ and the size of the training data set; the complexity of optimizing the codepoints via (97) scales as $|\mathcal{V}_l|$. The complexity of computing the MMSE estimate centroids of (99) scales as $2|\mathcal{V}_1||\mathcal{V}_2|$ and the size of the training data set. Representing the most demanding training steps, the complexity of optimizing E$_1$ and E$_2$ via (101) scales as $|\mathcal{V}_1||\mathcal{V}_2||\mathcal{I}_1|$ and $|\mathcal{V}_1||\mathcal{V}_2||\mathcal{I}_2|$, respectively. The complexity of optimizing D via (104) scales as $2|\mathcal{V}_1||\mathcal{V}_2|$.

Online Communication Phase

The main operations executed for reconstructing a pair of source realizations $\mathbf{x}_1$ and $\mathbf{x}_2$ in the online phase of the DQCS-PQ method are summarized in Algorithm 9. At each sensor $l = 1, 2$, the measurement vector $\mathbf{y}_l$ is mapped to the (optimal) cell index $v_l^* \in \mathcal{V}_l$ (i.e., the output of PQ$_l$) using $\mathcal{G}_l = \{\mathbf{g}_{l,1}, \ldots, \mathbf{g}_{l,|\mathcal{V}_l|}\}$ as

$$v_l^* = \underset{v_l \in \mathcal{V}_l}{\operatorname{argmin}} \|\mathbf{y}_l - \mathbf{g}_{l,v_l}\|_2^2, \; l = 1, 2. \tag{105}$$

For a given $\pi_l = \{\pi_l(1), \ldots, \pi_l(|\mathcal{V}_l|)\}$, α_l, and $\mathcal{H}_l = \{\mathbf{h}_{l,1}, \ldots, \mathbf{h}_{l,|\mathcal{H}_l|}\}$, the index v_l^* is mapped to the message index $i_l^* = \pi_l(v_l^*)$ and further to the binary codeword $\mathbf{h}_{l,i_l^*} = \alpha_l(i_l^*)$ at E_l. The joint decoder D maps the received codewords to the index pair $(i_1^*, i_2^*) = \left(\alpha_1^{-1}(\mathbf{h}_{1,i_1^*}), \alpha_2^{-1}(\mathbf{h}_{2,i_2^*})\right)$ and reconstructs source estimates as $\hat{\mathbf{x}}_l = \beta_l(i_1^*, i_2^*) = \mathbf{c}_{l,i_1^*,i_2^*}$ using $\mathcal{C}_l = \{\mathbf{c}_{l,1,1}, \ldots, \mathbf{c}_{l,|\mathcal{I}_1|,|\mathcal{I}_2|}\}$, $l = 1, 2$.

Online Complexity: Thanks to the imposed NN constraint for each PQ_l, $l = 1, 2$, the encoding complexity is tolerable: a sensor only has to calculate $|\mathcal{V}_l|$ distances in (105), followed by the table look-ups associated with π_l, α_l, and $\mathcal{H}_l$. Adjustment of the pre-quantization rate $\bar{R}_l = \log_2|\mathcal{V}_l|$ allows DQCS-PQ to trade off between the encoding complexity and compression performance. Clearly, an achievable performance range depends on the scaling and operation point of the sensing setup through, e.g., the source statistics and parameters N, M_l, K, $\mathbf{\Phi}_l$, and $\sigma_{\mathbf{W}}^2$, $l = 1, 2$. Thus, the need to upscale $\bar{R}_l$ in a high-dimensional setup may expand the VQ look-up table size $2^{\bar{R}_l}$ intolerably large. In this regard, the DQCS-PQ method is mainly applicable to setups with moderate signal dimensions.

Remark 6.4. *Even though an MMSE estimate $\mathbb{E}[\mathbf{X}_l|\mathbf{Y}_1 = \mathbf{y}_1, \mathbf{Y}_2 = \mathbf{y}_2]$, $l = 1, 2$, is an integral quantity in defining the optimal encoders and decoder, the pre-quantization technique relieves the sensor of reconstructing this estimate in the online communication phase (see Algorithm 9). This is different from the related works [265, 266], where the necessity of forming these exponentially complex estimates [99, 235, 281] hinders the practical implementation. Recall that the pre-quantization obviates the need of computing the estimates also in the offline training phase (see Algorithm 8).*

Optimal Codeword Lengths

Recall that the rate approximation (92) merely steers the quantizer design in subsuming a desired source code. Thus, the respective codebooks $\mathcal{H}_1$ and $\mathcal{H}_2$ must eventually be generated in practice. For the separate coding, the optimal codeword lengths $\gamma(i_l)$, $i_l \in \mathcal{I}_l$, satisfying the Kraft inequality are known. For instance, (instantaneous) Huffman coding [149] and [74, Sect. 5.6] and arithmetic coding [247] and [74, Sect. 13.3] perform close to the entropy bound $R_l \geq H(I_l)$, $l = 1, 2$ [66, 108].

Remark 6.5. *Improved compression performance is achievable from DSC/SW coding of the message indices which takes the inter-sensor correlation of I_1 and I_2 into account. The SW rate region is defined by the rates satisfying $R_1 \geq H(I_1|I_2)$, $R_2 \geq H(I_2|I_1)$, and $R_1 + R_2 \geq H(I_1, I_2)$ [268] and [74, Theorem 15.4.1]. Consequently, the rate measure for the symmetric SW coding is $r(p(i_1, i_2)) = -\log_2 p(i_1, i_2)$, and thus, the average sum rate (cf. (92)) becomes $\underline{R} = H(I_1, I_2)$, where $H(I_1, I_2) = -\sum_{i_1=1}^{|\mathcal{I}_1|} \sum_{i_2=1}^{|\mathcal{I}_2|} p(i_1, i_2)\log_2 p(i_1, i_2)$ is the joint entropy of indices I_1 and I_2. An asymmetric SW design has been addressed in, e.g., [50]. The optimal codeword lengths, however, are unknown [108]. Typically, low-density parity-check codes, turbo codes, and syndrome based codes are employed to approach the SW boundary [311, 234, 191]. Unfortunately, as the coding involves (large) blocks of indices, the increased complexity and delay of such non-instantaneous codes may become an issue. Potential variants include the SW type finite-dimensional codes for lossless/near-lossless coding introduced in [320, 319].*

Numerical Results

Numerical results are presented to demonstrate the DR behavior of the DQCS-PQ method (Algorithm 9) and to compare it against several baseline QCS methods.

Simulation Setup

We consider a setup with equal numbers of measurements $M \triangleq M_1 = M_2$, and equal support probabilities $p(\mathcal{T}_s) = 1/\binom{N}{K}$, $\forall s = 1, \dots, \binom{N}{K}$. For a given $\mathcal{T}_s$, the non-zero parts of $\bar{\mathbf{X}}$ and $\mathbf{X}'_l$, denoted as $\bar{\mathbf{X}}_{\mathcal{T}_s}$ and $\mathbf{X}'_{l,\mathcal{T}_s}$, are defined as i.i.d. Gaussian random variables $\bar{\mathbf{X}}_{\mathcal{T}_s} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\bar{\mathbf{X}}}\mathbf{I}_K)$, and $\mathbf{X}'_{l,\mathcal{T}_s} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\mathbf{X}'}\mathbf{I}_K)$, $s = 1, \dots, \binom{N}{K}$, $l = 1, 2$. Thus, $\mathbf{X}_{l,\mathcal{T}_s} \sim \mathcal{N}(\mathbf{0}, (\sigma^2_{\bar{\mathbf{X}}} + \sigma^2_{\mathbf{X}'})\mathbf{I}_K)$. The spatial correlation between the sensors is adjusted by parameter $\rho^{\text{corr}} \triangleq \sigma^2_{\bar{\mathbf{X}}}/\sigma^2_{\mathbf{X}'}$ with $\sigma^2_{\bar{\mathbf{X}}} = 1$. Two types of measurement matrices $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are considered by 1) drawing the entries from Gaussian distribution $\mathcal{N}(0, 1/M)$, and normalizing the columns as $\|\cdot\|_2^2 = 1$, and 2) taking the first (last) M rows of an $N \times N$

DCT-matrix, and normalizing the columns as $\|\cdot\|_2^2 = 1$. Unless otherwise stated, $\bar{R}_1 = \bar{R}_2 = 10$ bits, and $\log_2|\mathcal{I}_1| = \log_2|\mathcal{I}_2| = \{1, \ldots, 8\}$ bits.

The splitting method [188, 133] is employed to optimize both the pre-quantizers (Algorithm 7), and the encoders and decoder (Algorithm 8). In the splitting procedure, a quantization system is first optimized for codebook sizes $|\mathcal{I}_l| = 2$, $l = 1, 2$ ($|\mathcal{V}_l|$ for PQ_l). Then, each resultant codevector is split through small perturbation, and the algorithm is run for $|\mathcal{I}_l| = 4$ using the new codevectors as initial ones (i.e., a "warm start"). This gradual bifurcation is repeated until the desired codebook sizes are reached.

For DQCS-PQ, the codebooks $\mathcal{H}_1$ and $\mathcal{H}_2$ are generated via the Huffman algorithm, and the average rates are defined via (91). The weight parameter is set by a rule $\mu \triangleq \mu_{\text{c}}/\log_2|\mathcal{I}_l|$, where $\mu_{\text{c}} \geq 0$ is a non-negative parameter. When E_1, E_2, and D are optimized for multiple (ascending) values of μ_{c}, the quantities π_2, $\mathcal{C}_1$, and $\mathcal{C}_2$ obtained for the preceding (smaller) μ_{c} are used to initialize Algorithm 8 for the next μ_{c}.

We consider two baseline methods which rely on a customary approach behind early QCS methods as by using a fixed rate $R_l = \log_2|\mathcal{I}_l|$, and relying on a VQ encoder that minimizes MSE quantization distortion:

1. VQ-E_S-D_S: a method with separate NN encoding, and separate decoding, where the VQ encoder of the lth sensor minimizes $\sum_{i_l \in \mathcal{I}_l} p(i_l)\mathbb{E}[\|\mathbf{Y}_l - \hat{\mathbf{y}}_{l,i_l}\|_2^2 | I_l = i_l]$ with encoder codepoints $\hat{\mathbf{y}}_{l,i_l} \in \mathbb{R}^M$, and the decoder consists of the MSE-optimal decoder codevectors $\mathbf{c}_{l,i_l}^{\text{E}_\text{S}\text{-}\text{D}_\text{S}} = \mathbb{E}[\mathbf{X}_l | I_l = i_l]$, $i_l \in \mathcal{I}_l$, for the associated Voronoi regions, $l = 1, 2$.

2. VQ-E_S-D_J: a method with separate encoding identical to VQ-E_S-D_S, but with joint decoding via decoder codevectors $\mathbf{c}_{l,i_1,i_2}^{\text{E}_\text{S}\text{-}\text{D}_\text{J}} = \mathbb{E}[\mathbf{X}_l | I_1 = i_1, I_2 = i_2]$, $i_1 \in \mathcal{I}_1$, $i_2 \in \mathcal{I}_2$, $l = 1, 2$. In summary, DQCS-PQ differs from VQ-E_S-D_J by using 1) pre-quantization, 2) variable-rate coding, and 3) encoders that minimize the signal reconstruction distortion.

The distortion is measured as $D_{\text{ave}} \triangleq 10\log_{10}(D_1/\mathbb{E}[\|\mathbf{X}_1\|_2^2] + D_2/\mathbb{E}[\|\mathbf{X}_2\|_2^2])$ (dB), where $\mathbb{E}[\|\mathbf{X}_l\|_2^2] = \sum_{s=1}^{\binom{N}{K}} p(\mathcal{T}_s)\mathbb{E}[\|\mathbf{X}_{l,\mathcal{T}_s}\|_2^2] = K(\sigma_{\bar{\mathbf{X}}}^2 + \sigma_{\mathbf{X}'}^2)$, $l = 1, 2$. The rate is measured as $R_{\text{ave}} \triangleq \underline{R}/2$ (bits). Due to the exponential complexity

Table 3: Huffman source codebooks $\mathcal{H}_1$ and $\mathcal{H}_2$ of DQCS-PQ used in Section 6.

Message index I_1/I_2	Codebook $\mathcal{H}_1$	Codebook $\mathcal{H}_2$
1	10010	–
2	101	101
3	10000	–
4	0	0
5	11	11
6	10011	–
7	–	1001
8	10001	1000

of quantizers in the offline training phase, the experiments are restricted to moderate quantization rates and signal dimensions. Data sets of size 5×10^5 are used for training and performance evaluation.

Illustration of Distributed Quantization

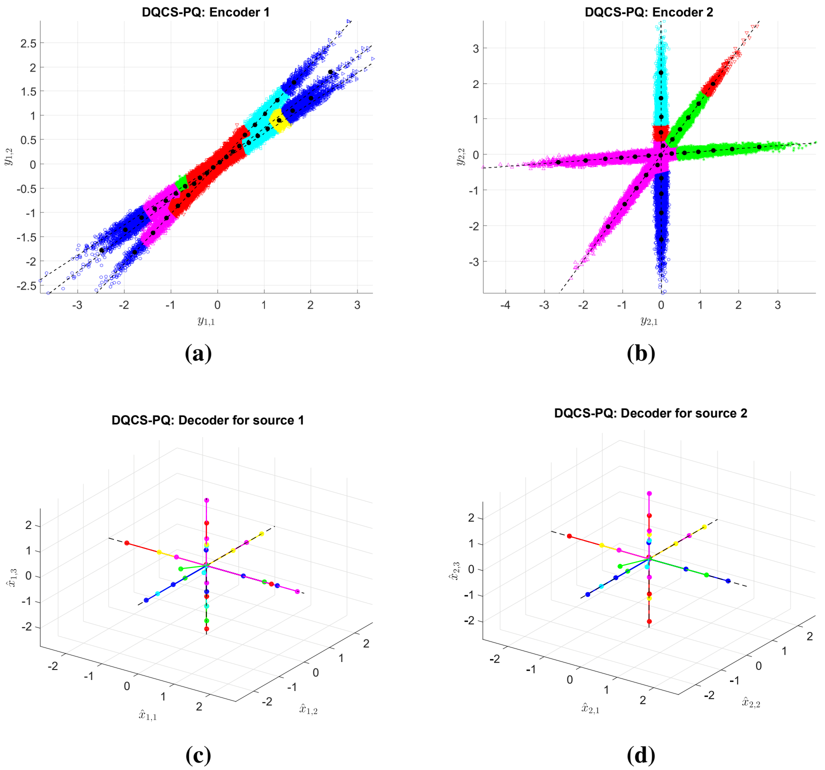
First, we visualize the general operation of DQCS-PQ by considering a low-dimensional setup with $N = 3$, $M = 2$, $K = 1$, $\sigma^2_{\mathbf{W}} = 0.003$, $\rho^{\text{corr}} = 10^2$, Gaussian measurement matrices

$$\begin{aligned} \mathbf{\Phi}_1 &= \begin{bmatrix} 0.8472 & 0.8044 & -0.6982 \\ 0.5312 & 0.5941 & -0.7159 \end{bmatrix}, \\ \mathbf{\Phi}_2 &= \begin{bmatrix} 0.9966 & -0.5589 & -0.0016 \\ 0.0827 & -0.8293 & 1.0000 \end{bmatrix}, \end{aligned} \tag{106}$$

and quantizer parameters $\mu_c = 0.5$, $|\mathcal{V}_l| = 32$, and $|\mathcal{I}_l| = 8$. The encoders/decoder optimized via Algorithms 7 and 8 are depicted in Fig. 24: (a) and (b) illustrate the outcomes of pre-quantization of $\mathbf{Y}_1$ and $\mathbf{Y}_2$, and the subsequent message index mappings π_1 and π_2, respectively; (c) and (d) depict the reconstruction codebooks $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively; (e) tabulates π_1 and π_2. The source codebooks $\mathcal{H}_1$ and $\mathcal{H}_2$ are presented in Table 3. For an ease of comparison, VQ-E$_\text{S}$-D$_\text{J}$ is similarly illustrated in Fig. 25.

The resulting DR performance for DQCS-PQ are $D_{\text{ave}} = -10.19$ dB and $R_{\text{ave}} = 1.83$ bits, and $D_{\text{ave}} = -4.475$ dB and $R_{\text{ave}} = 3$ bits for VQ-E$_\text{S}$-D$_\text{J}$. This implies a striking 39 $\%$ reduction in the rate, and a 5.7 dB reduction in the distortion in favor of the DQCS-PQ method.

Figs. 24(a) and (b) illustrate the key function of PQ$_1$ and PQ$_2$, "encoder shaping". Consider the message index $I_1 = 2$ and $I_2 = 7$: the 5 cells assigned



(a) **(b)**

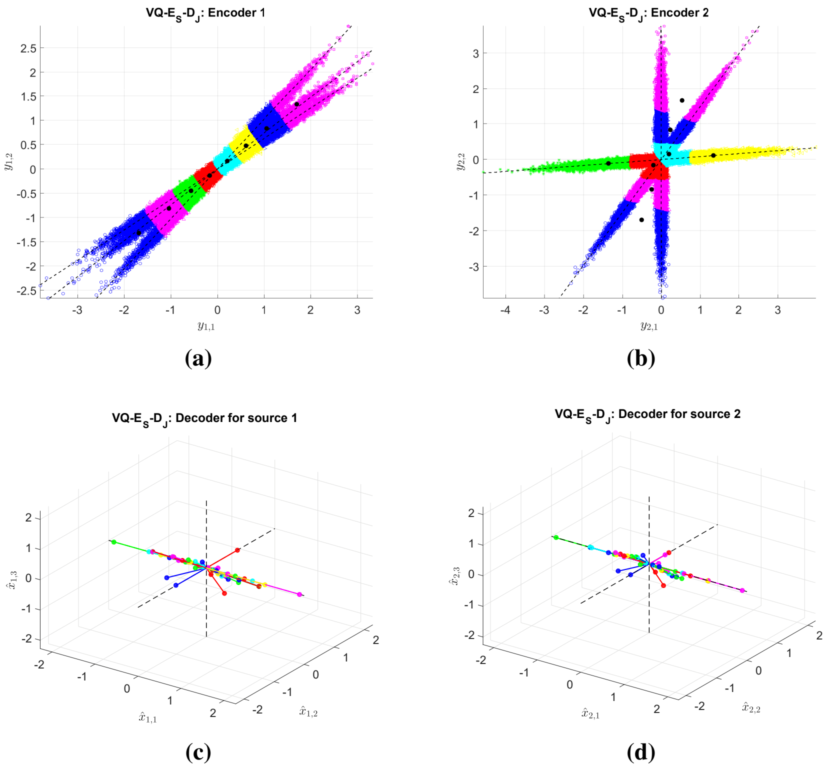
(c) **(d)**

Cell index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Message index mapping π_1	1	1	1	1	2	2	2	2	4	2	4	3	4	4	4	4	4	4	4	4	5	4	5	5	5	5	6	5	8	8	8	8
Message index mapping π_2	4	4	2	2	4	4	2	2	4	4	4	4	4	4	4	4	4	5	4	5	5	5	5	5	7	8	5	5	8	8	5	7

(e)

Figure 24: Visualization of DQCS-PQ in terms of (a) the pre-quantizer PQ_1 and encoder E_1, (b) the pre-quantizer PQ_2 and encoder E_2, (c) the reconstruction codebook $\mathcal{C}_1$, (d) reconstruction codebook $\mathcal{C}_2$, and (e) message index mappings π_1 and π_2. In (a) and (b), the black dots represent the codepoints $\mathbf{g}_{l,1}, \ldots, \mathbf{g}_{l,32}$ of PQ_1 and PQ_2; each dotted line represents the sparsity-dependent span $\underline{\phi}_{l,n} X_{l,n}$, $n = 1, 2, 3$; each color indicates the set $\pi_l^{-1}(i_l)$ for message index $i_l \in \mathcal{I}_l$, $l = 1, 2$ (the colors are equivalent in (e)). In (c) and (d), the dots represent the codevectors $\mathbf{c}_{l,1,1}, \ldots, \mathbf{c}_{l,8,8}$; the coordinate axes are shown as dashed lines.

to $I_1 = 2$, i.e., $\pi_1^{-1}(2) = \{5, 6, 7, 8, 10\}$, form an irregularly shaped region of measurement vectors, whereas the region for $I_2 = 7$ consists of 2 non-



(a) **(b)**

(c) **(d)**

Figure 25: Visualization of VQ-E$_\text{S}$-D$_\text{J}$ in terms of (a) the encoder at sensor 1, (b) the encoder at sensor 2, (c) the decoder codevectors for $\mathbf{X}_1$, and (d) the decoder codevectors for $\mathbf{X}_2$. The lines, dots, and colors have the same meanings as those in Fig. 24.

contiguous cells[16] by $\pi_2^{-1}(7) = \{25, 32\}$. This shaping feature allows the cells $\mathcal{S}_{l,1}, \ldots, \mathcal{S}_{l,32}$ to merge (i.e., classify measurement (training) vectors) into message indices $\mathcal{I}_l = \{1, \ldots, 8\}$, $l = 1, 2$, by taking the collective effect of the CS and DSC into account. As a consequence, to some extent the partitioning in Fig. 24(a) follows the sparsity-dependent linear projections $\underline{\phi}_{l,n} X_{l,n} + \mathbf{W}_l$, where $\underline{\phi}_{l,n} \in \mathbb{R}^M$ is the nth column of $\mathbf{\Phi}_l$, and $X_{l,n}$ is the nth element of $\mathbf{X}_l$, $n = 1, 2, 3$. Similarly, Figs. 24(c) and (d) show that nearly all reconstruction codevectors are (approximately) K-sparse vectors. Because they are also

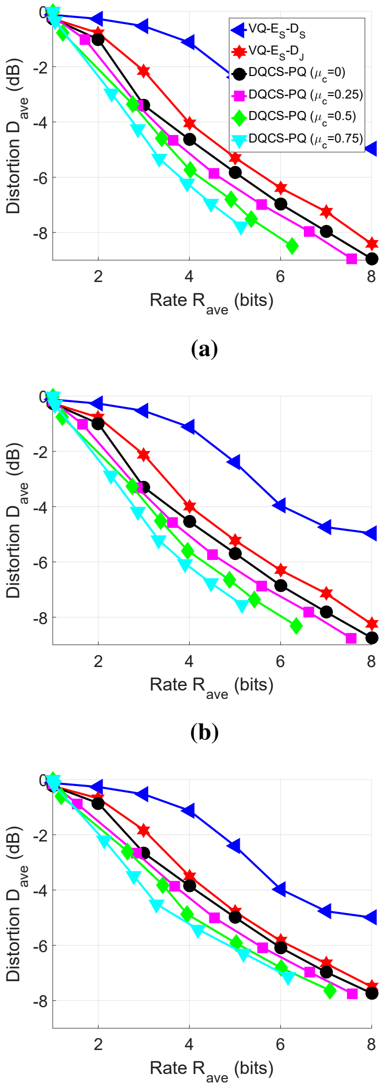
[16] This can be interpreted as an index reuse, characteristic to a distributed quantization setup [294, 110].

rather evenly distributed in $\mathbb{R}^3$, the codebooks $\mathcal{C}_1$ and $\mathcal{C}_2$ presumably contain accurate estimates for the K-sparse sources $\mathbf{X}_1$ and $\mathbf{X}_2$. On the contrary, the encoder regions of VQ-E$_\mathrm{S}$-D$_\mathrm{J}$ in Fig. 25(a) are solely based on the probability mass of $\mathbf{Y}_1$, and the codevectors in Figs. 25(c) and (d) are dispersed with no K-sparse structure. Such incognizance of the CS resulted in poor performance. Table 3 illustrates the power of entropy coding in DQCS-PQ: the codebooks $\mathcal{H}_1$ and $\mathcal{H}_2$ contain only 7 and 5 codewords (instead of 8), respectively, and the expected codeword length is minimized by assigning the shortest codeword ”0” to the most frequent indices $I_1 = I_2 = 4$, the codeword ”1 1” to the second most frequent indices $I_1 = I_2 = 5$, and so forth. Note that even though $\mathcal{H}_1$ contains 5-bit codewords, R_ave for DQCS-PQ is 39 % lower (and D_ave 5.7 dB lower) than for VQ-E$_\mathrm{S}$-D$_\mathrm{J}$ that uses fixed-length 3-bit codewords.

Distortion-Rate Performance

Fig. 26(a) depicts the average distortion D_ave versus the average rate R_ave for $N = 20$, $M = 8$, $K = 2$, $\sigma^2_\mathbf{W} = 0.01$, $\rho^\mathrm{corr} = 10^3$, and the Gaussian type $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$. By completely disregarding the signal correlation, the VQ-E$_\mathrm{S}$-D$_\mathrm{S}$ method performs poorly. Significant gains are achieved by using the VQ-E$_\mathrm{S}$-D$_\mathrm{J}$ for which the joint decoding effectively expands the codebook size from $|\mathcal{I}_l|$ to $|\mathcal{I}_l|^2$ (especially for high correlations). However, since the encoders of both methods are blind to the peculiarities of the CS, and they use fixed rates, the proposed variable-rate CS-aware DQCS-PQ invariably obtains the best compression performance for all values of μ_c, including the fixed-rate version ($\mu_\mathrm{c} = 0$). A similar performance trend is true for all the subsequent experiments as well.

It can be seen that VQ-E$_\mathrm{S}$-D$_\mathrm{S}$ becomes saturated with the distortion at around $D_\mathrm{ave} = -5$ dB. Such an unavoidable constant error level is caused by the remote source coding nature of CS. The saturation will also happen for the other methods for sufficiently high rates. Namely, regardless of the compression/coding method, the CS parts with additive measurement noise at the sensors prevent the perfect reconstruction of $\mathbf{X}_1$ and $\mathbf{X}_2$ at the decoder, even for the rates approaching infinity. For the VQ-E$_\mathrm{S}$-D$_\mathrm{S}$ method, the error level is defined by the rate-independent MMSE estimation error of the form $\sum_{l=1}^{2} \int_{\mathbf{y}_l} \mathbb{E}\left[\|\mathbf{X}_l - \mathbb{E}[\mathbf{X}_l|\mathbf{Y}_l = \mathbf{y}_l]\|_2^2\right] f(\mathbf{y}_l)\mathrm{d}\mathbf{y}_l$.



(a)

(b)

(c)

Figure 26: Distortion-rate performance for $N = 20$, $M = 8$, $K = 2$, the Gaussian type $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$, and signal correlation parameter (a) $\rho^{\text{corr}} = 10^3$, (b) $\rho^{\text{corr}} = 10^2$, and (c) $\rho^{\text{corr}} = 10^1$. The colors and markers of the curves in (b) and (c) are equivalent to those in (a).

For VQ-E$_\mathrm{S}$-D$_\mathrm{J}$ and DQCS-PQ, the level is given by $\sum_{l=1}^{2}\int_{\mathbf{y}_1}\int_{\mathbf{y}_2}\mathbb{E}\left[\|\mathbf{X}_l-\mathbb{E}[\mathbf{X}_l|\mathbf{Y}_1=\mathbf{y}_1,\mathbf{Y}_2=\mathbf{y}_2]\|_2^2\right]f(\mathbf{y}_1,\mathbf{y}_2)\mathrm{d}\mathbf{y}_1\mathrm{d}\mathbf{y}_2$.

Influence of the Weight Parameter

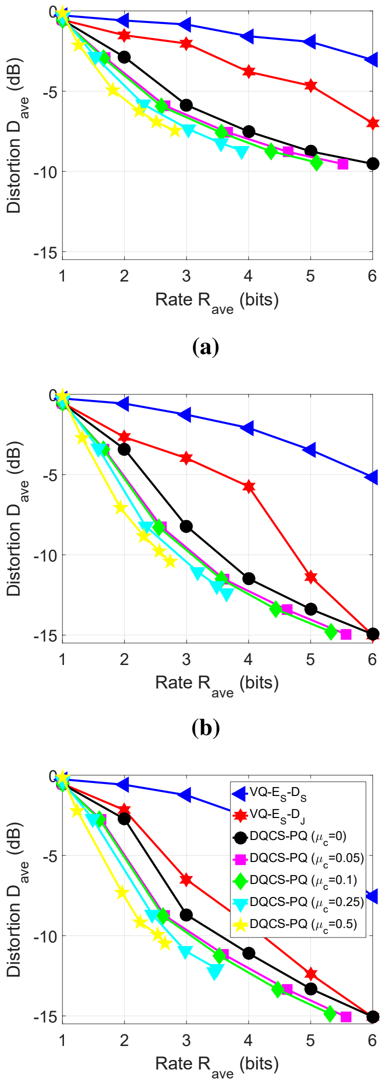
The impact of weight parameter μ in DQCS-PQ can be perceived from Fig. 26(a). By choosing an appropriate variable-rate code, DQCS-PQ can flexibly make a trade-off for the DR performance, and thus, lends itself to varying compression scenarios. For instance, consider a rate-limited WSN application, where the maximum (source) rate is restricted to, say, $R_\mathrm{ave}=5$ bits due to bad channel conditions, or congested data traffic. Using the entropy coding of the message indices, the reconstruction performance is improved from $D_\mathrm{ave}=-5.8$ dB ($\mu_\mathrm{c}=0$) to $D_\mathrm{ave}=-7.6$ dB ($\mu_\mathrm{c}=0.75$). Alternatively, consider an application with a minimum required reconstruction fidelity of, say, $D_\mathrm{ave}=-7$ dB. With the aid of variable-length coding, a significant rate reduction of 1.5 bits from $R_\mathrm{ave}=6.0$ bits ($\mu_\mathrm{c}=0$) to $R_\mathrm{ave}=4.5$ bits ($\mu_\mathrm{c}=0.75$) is achieved.

Influence of Source Correlation

Figs. 26(a) – (c) present the compression performance for different signal correlation levels $\rho^\mathrm{corr}=\{10^3,10^2,10^1\}$. As a first remark, the DQCS-PQ method utilizes the signal correlation efficiently for compressing sparse sources. The higher the correlation, the greater the compression gain is in favor of DQCS-PQ. For $\rho^\mathrm{corr}=10^3$ (high correlation), a distortion of $D_\mathrm{ave}=-7$ dB is achieved at rates $R_\mathrm{ave}=6.7$ bits for VQ-E$_\mathrm{S}$-D$_\mathrm{J}$, and 4.5 bits for DQCS-PQ ($\mu_\mathrm{c}=0.75$). For $\rho^\mathrm{corr}=10^1$ (low correlation), the respective rates are $R_\mathrm{ave}=7.4$ and 6.0 bits. Note that while VQ-E$_\mathrm{S}$-D$_\mathrm{J}$ and DQCS-PQ perform better with the increasing correlations, the indiscernible disparities of the curves of fully separate VQ-E$_\mathrm{S}$-D$_\mathrm{S}$ are solely attributed to different signal powers $\mathbb{E}[\|\mathbf{X}_l\|_2^2]$.

Influence of Number of Measurements

Fig. 27 demonstrates the influence of different number of measurements $M=\{2,3,4\}$ in the setup with $N=10$, $K=1$, $\sigma_\mathbf{W}^2=0.01$, $\rho^\mathrm{corr}=10^2$,



(a)

(b)

(c)

Figure 27: Distortion-rate performance for $N = 10$, $K = 1$, $\rho^{\text{corr}} = 10^2$, the DCT type $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$, and number of measurements (a) $M = 2$, (b) $M = 3$, and (c) $M = 4$. The colors and markers of the curves in (a) and (b) are equivalent to those in (c).

and the DCT type $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$. Unlike the other methods, DQCS-PQ achieves decent performance even in a very noisy scenario (i.e., for $M = 2$). By increasing the measurements to $M = 3$ and further to $M = 4$, the setup becomes less contaminated, allowing each QCS method to compress the sparse sources more reliably. However, whereas VQ-$\mathrm{E_S}$-$\mathrm{D_S}$ and VQ-$\mathrm{E_S}$-$\mathrm{D_J}$ make considerable gains when switching from $M = 3$ to $M = 4$, the respective improvement for DQCS-PQ is negligible. This is congruent with the CS philosophy: when M is sufficient for a successful support recovery, further increasing M does not provide significant gains. Accordingly, given that M is at a satisfactory level, it is more cost-effective to improve the reconstruction quality by increasing the rate. Since acquiring more measurements can be expensive – or even infeasible – in practice, the capability to operate at low signal-to-noise ratios is indisputably a great advantage of the DQCS-PQ method.

Conclusions

This section addressed lossy DSC for efficiently acquiring correlated sparse sources from quantized noisy compressive measurements in WSNs. A DR optimized complexity-constrained variable-rate distributed QCS method was developed to minimize a weighted sum of the average MSE signal reconstruction distortion and the average rate. In order to ameliorate the practical feasibility, the encoding complexity of each sensor was restrained by prediscretizing the measurement vector space by utilizing VQ. Conforming to an entropy-constrained design framework, the method incorporates a desired variable-rate code using the rate measures of an entropy bound. Alternating optimization was used to derive the necessary optimality conditions and propose practical training algorithms for a two-sensor system. The computational complexities in the training and communication phases were discussed.

The numerical results illustrated that the devised method has superior DR performance under varying signal correlation levels and signal-to-noise ratios. Moreover, the method is adaptable to various compression settings with stringent rate or distortion requirements. The results also demonstrated the improvement of compression performance by increasing either the number of measurements or the encoding rate in a QCS setup. Depending on the application and its restrictions, either option may be more beneficial, and, in the first place, feasible to realize. The key finding was that efficient finite-rate acquisition of correlated sparse sources calls for 1) a DSC design, 2) CS-awareness both at the encoder and decoder ends, and 3) the use of entropy coding.

The version of record is available at: http://dx.doi.org/10.1561/2000000107

7

Rate-Distortion Performance of Lossy Compressed Sensing

This section addresses the rate-distortion (RD) performance of lossy CS from an information-theoretic perspective. A single-sensor setup is assumed due to the analytical tractability. In this setup, a CS based sensor observes a sparse information source indirectly and communicates compressed noisy measurements to a decoder for signal reconstruction with the aim to minimize the MSE distortion. Thus, the obtained theoretical results shed light on, e.g., the compression performance limits of the single-sensor version of the QCS setup in Section 6. The minimum achievable rate for a given distortion fidelity in a QCS setup is represented by the remote rate-distortion function (RDF). We first present an analytically tractable lower bound to the remote RDF by providing support side information to the encoder and decoder. A variant of the Blahut-Arimoto algorithm is devised to numerically approximate the remote RDF. Furthermore, we present several practical symbol-by-symbol QCS algorithms relying on 1) compress-and-estimate, 2) estimate-and-compress, and 3) support-estimation-and-compress strategies. Numerical results illustrate the main RD characteristics of the lossy CS and compare the performance of practical QCS methods against the proposed limits. In particular, an entropy-constrained VQ based estimate-and-compress QCS method is numerically shown to approach the remote RDF.

Related Works

Since existing symbol-by-symbol quantizer based QCS algorithms are already discussed in Section 6, the following sections mainly review the works that focus on information theoretic perspectives of lossy CS.

Lossy CS

In a lossy compression system, the point of interest lies in encoding an information source at the minimum rate so that the source can be reproduced at the destination with a distortion not exceeding a tolerable level. This best achievable compression performance for a given distortion fidelity is given by the *rate-distortion function* (RDF) of a source [260]. However, the limit is achievable only via excessively complex encoder-decoder pairs with infinitely long block lengths [25, Sect. 3.3] and [134]. Thus, the RD theory is primarily applicable to performance analysis and benchmarking of practical lossy coding methods.
As discussed in Section 6, the compression task in a QCS setup falls into *remote source coding* [85][25, Sect. 3.5, 4.5.4][253, 304, 303, 110]. The compression limit for such a setup is given by the *remote RDF*. Information-theoretic analysis of QCS – termed *lossy CS* henceforth – is incomplete, i.e., the remote RDF for lossy CS is unknown. A problem overview along with initial results was first given in [129]. Kipnis *et al.* [165] analyzed lossy CS under a large system limit (i.e., $M, N \to \infty$) using the replica method, and derived the minimal achievable per-letter MSE in a general form. Coluccia *et al.* [69] derived a distortion-rate (DR) lower bound assuming support SI at the decoder, high-rate quantization, a large system regime $N \to \infty$, and noiseless measurements. As slightly different, yet related works, RD bounds for directly compressing sparse sources were derived in [57, 293, 230], and the compression of (sparse) Bernoulli generalized Gaussian sources via uniform SQ was studied in [113]. Other works on remote compression in (non-CS) setups include [276, 256, 163].

Computation of RDFs

Despite the well-known general definitions, deriving a RDF/remote RDF in closed form is, in general, elusive. To date, remote RDFs have been derived

The version of record is available at: http://dx.doi.org/10.1561/2000000107

only for few well-behaved source/observation distributions; see, e.g., the works in [85, 304], [25, Sect. 4.5.4] and [256, 276, 163], assuming Gaussian distributions, and [166] considering a remote binary source. When a closed-form solution is unattainable, one recourse is numerical approximation. The bulk of such iterative methods rely on the *Blahut-Arimoto* (BA) algorithm, which traces back to the pioneering works by Blahut [30] and Arimoto [11]. This algorithm has been adapted to remote sensing scenarios under joint compression and classification in [86], and the chief executive officer problem in [168]. The effect of discretizing a continuous source on the accuracy of the RDF evaluated via the BA algorithm for finite output alphabets was analyzed in [106]; similarly, an approximation of the capacity of a continuous channel was studied in [58]. Other computation methods include a mapping based method akin to deterministic annealing in [249], and Lagrange duality based convex optimization in [65].

Source Coding with Side Information

In some WSN applications, the encoder and/or decoder is intermittently reinforced by various types of prior knowledge, i.e., *side information* (SI), on the signal of interest. SI on, e.g., sparsity, occupied frequency bands, or magnitude variations of a signal can be obtained via temporal/spatial correlations or inter-sensor collaboration. Since SI reduces the necessary transmission rate, added SI can be used to derive RD lower bounds in closed form. Compression with shared SI at the encoder and decoder follows conditional RD theory introduced by Gray [131, 132]. The case with correlated, but not necessarily identical SI at the encoder and decoder was addressed in, e.g., [73, 62, 140]. Other (non-CS) SI aided compression variants can be found in, e.g., [25, Sect. 6.1], [174, 126, 189], [121, Sect. 11.1] and [124, Sect. 5.6].

Lossy CS via Remote Source Coding

In this work, the objective is to investigate the RD performance of the model depicted in Fig. 28, where the information source is observed via noisy compressed measurements, encoded with a lossy source code, and communicated to the decoder for signal reconstruction. The transmissions from encoder E to decoder D are assumed to be error-free. The compression task is classified as

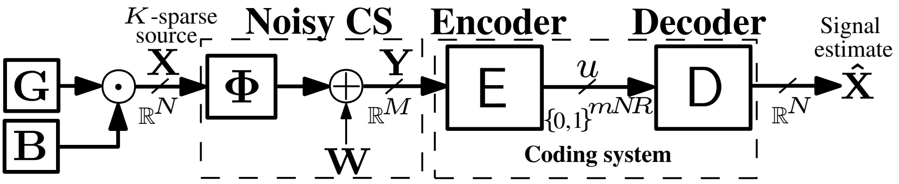


Figure 28: A single-sensor lossy compression of a sparse source from compressive noisy measurements.

remote source coding because the encoder accesses the source only through noisy measurements. The source and the measurement model are defined next, followed by the formal statement of the problem.

Source Model

Let $\{\mathbf{X}_n\}_{n=1}^{\infty}$ be a discrete-time memoryless vector source sequence[17]. Each random vector[18] $\mathbf{X}_n = [X_{n,1} \cdots X_{n,N}]^{\mathrm{T}}$ is K-sparse, where $K \leq N$, i.e., it takes on values in the continuous source alphabet $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^N : \|\mathbf{x}\|_0 = K\}$. The set $\mathcal{X}$ thus consists of the union of $\binom{N}{K}$ subspaces, i.e., the signal model is nonlinear [101, 81]. It is further assumed that the source sequence is generated from the memoryless sequence of tuples $\{(\mathbf{G}_n, \mathbf{B}_n)\}_{n=1}^{\infty}$ so that $\mathbf{X}_n = \mathbf{G}_n \odot \mathbf{B}_n$, where $\odot$ denotes the Hadamard product; $\mathbf{G}_n$ is a length-N zero mean Gaussian random vector $\mathbf{G}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{G}})$ with a covariance matrix $\boldsymbol{\Sigma}_{\mathbf{G}} \in \mathbb{S}_{++}^N$; $\mathbf{B}_n$ is a length-N binary support random vector, independent of $\mathbf{G}_n$, with the discrete alphabet $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_{|\mathcal{B}|}\}$, where $|\mathcal{B}| = \binom{N}{K}$ is the number of all possible *sparsity patterns*. Each $\mathbf{b}_s = [b_{s,1} \cdots b_{s,N}]^{\mathrm{T}} \in \mathcal{B}$ is unique, contains K ones and $N - K$ zeros, and is associated with the *a priori* probability $p(\mathbf{b}_s) \triangleq \Pr(\mathbf{B} = \mathbf{b}_s)$ with $p(\mathbf{b}_s) \in [0, 1]$ and $\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) = 1$.

[17] Due to the independence over time, the time index n will often be suppressed for brevity whenever not explicitly needed.

[18] In this section, with slight abuse of notation, we use capital bold letters (e.g., $\mathbf{X}$) to refer to *random* vectors, and small bold letters to refer to their corresponding realizations (e.g., $\mathbf{X} = \mathbf{x}$).

Noisy CS

Let $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ be a fixed and known measurement matrix, $K \leq M \leq N$. The sensor (i.e., the encoder) observes $\{\mathbf{X}_n\}_{n=1}^{\infty}$ indirectly [43, 87, 146, 49] as

$$\mathbf{Y}_n = \mathbf{\Phi}\mathbf{X}_n + \mathbf{W}_n,\ n = 1, 2, \dots \tag{107}$$

where $\mathbf{W}_n$, $n = 1, 2, \dots$, are length-M random measurement noise vectors independent of $\{\mathbf{X}_n\}_{n=1}^{\infty}$, and each $\mathbf{Y}_n$ is a length-M measurement random vector that takes values in the measurement vector space $\mathcal{Y}$. It is assumed that $\mathbf{W}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{W}})$ with a covariance matrix $\mathbf{\Sigma}_{\mathbf{W}} \in \mathbb{S}_{++}^{M}$. No restricting assumptions of $\mathbf{\Phi}$ are made in the derivations; the impact of $\mathbf{\Phi}$ on the CS recovery performance is discussed in Section 2.

Lossy CS Problem

Let $\mathbf{X}^m \triangleq \{\mathbf{X}_n\}_{n=1}^{m}$ and $\mathbf{x}^m \triangleq \{\mathbf{x}_n\}_{n=1}^{m}$ denote the blocks of m consecutive source random vectors and the corresponding realizations, respectively. Let $\mathcal{X}^m$ denote the m-fold Cartesian product of $\mathcal{X}$. Analogous notations are used for the other vectors. Let $\hat{\mathbf{X}}$ be the reproduction random vector at the decoder output, taking values in the reproduction alphabet $\hat{\mathcal{X}}$. Finally, define the average per-letter MSE distortion between vectors $\mathbf{x} = [x_1 \cdots x_N]^{\mathrm{T}} \in \mathcal{X}$ and $\hat{\mathbf{x}} = [\hat{x}_1 \cdots \hat{x}_N]^{\mathrm{T}} \in \hat{\mathcal{X}}$ as

$$d(\mathbf{x}, \hat{\mathbf{x}}) \triangleq N^{-1} \textstyle\sum_{k=1}^{N} (x_k - \hat{x}_k)^2 \tag{108}$$

and the average per-letter MSE distortion between blocks $\mathbf{x}^m \in \mathcal{X}^m$ and $\hat{\mathbf{x}}^m \in \hat{\mathcal{X}}^m$ as

$$d(\mathbf{x}^m, \hat{\mathbf{x}}^m) \triangleq (mN)^{-1} \textstyle\sum_{n=1}^{m} \sum_{k=1}^{N} (x_{n,k} - \hat{x}_{n,k})^2. \tag{109}$$

The lossy source coding system in Fig. 28 operates as follows [134, Sect. 2.1], [74, Sect. 10.2] and [121, Sect. 3.5, 3.6]. The encoder E observes a block of measurements $\mathbf{y}^m \in \mathcal{Y}^m$ and compresses it into a message represented by an index $u \in \mathcal{U}$ of rate mNR bits using an encoder mapping

$$g_{\mathsf{E}}^{m} : \mathcal{Y}^m \to \mathcal{U} \triangleq \{1, \dots, 2^{mNR}\} \tag{110}$$

where the rate R is defined as the bits/entry of $\mathbf{X}$. The decoder D uses the index to reconstruct an estimate of $\mathbf{x}^m \in \mathcal{X}^m$ via a decoder mapping

$$g_{\mathsf{D}}^{m} : \mathcal{U} \to \hat{\mathcal{X}}^m. \tag{111}$$

A pair (R, D) for distortion $D \geq 0$ is achievable if there exists a sequence of $(2^{mNR}, m)$-RD codes with mappings g_{E}^m and g_{D}^m so that $\lim_{m\to\infty} \mathbb{E}[d(\mathbf{X}^m, g_{\mathsf{D}}^m\{g_{\mathsf{E}}^m(\mathbf{Y}^m)\})] \leq D$. Let $\mathcal{R}$ be the closure of the set of achievable (R, D) pairs.

Definition 2. (Lossy CS source coding problem) *Amongst all the* E-D *pairs of mappings* (110) *and* (111)*, determine the infimum of (achievable) rates* R *such that* $\mathbf{X}$ *can be reproduced with the average distortion satisfying* $\mathbb{E}[d(\mathbf{X}^m, \hat{\mathbf{X}}^m)] \leq D + \epsilon$ *for any positive real number* ϵ*, i.e., define [74, Sect. 10.2]*

$$R_{\mathbf{X}}^{\mathrm{rem}}(D) = \inf_{(R,D)\in\mathcal{R}} R. \tag{112}$$

$R_{\mathbf{X}}^{\mathrm{rem}}(D)$ *is called* the remote RDF *of source* $\mathbf{X}$.

It is worth noting that the MSE distortion, in general, leads to non-sparse signal reconstruction, which might be undesirable in certain CS applications.

Remote RDF

The general expression of the remote RDF for a discrete memoryless source with discrete memoryless observations has been derived in [25, Eqs. (3.5.1) – (3.5.5)]. Adapting the result to continuous-valued signals $\mathbf{X}$ and $\mathbf{Y}$, $R_{\mathbf{X}}^{\mathrm{rem}}(D)$ in (112) can be expressed as

$$R_{\mathbf{X}}^{\mathrm{rem}}(D) = \min_{f(\hat{\mathbf{x}}|\mathbf{y}):\mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})]\leq D} \frac{1}{N} I(\mathbf{Y};\hat{\mathbf{X}}) \tag{113a}$$

where the optimization is over the conditional PDF $f(\hat{\mathbf{x}}|\mathbf{y})$, commonly referred to as *the test channel*, and $d(\mathbf{x}, \hat{\mathbf{x}})$ is the distortion in (108). The mutual information[19] between $\mathbf{Y}$ and $\hat{\mathbf{X}}$ is

$$I(\mathbf{Y};\hat{\mathbf{X}}) = \int_{\mathbf{y}}\int_{\hat{\mathbf{x}}} f(\mathbf{y}) f(\hat{\mathbf{x}}|\mathbf{y}) \log \frac{f(\hat{\mathbf{x}}|\mathbf{y})}{f(\hat{\mathbf{x}})} \mathrm{d}\mathbf{y}\mathrm{d}\hat{\mathbf{x}} \tag{113b}$$

and the average MSE distortion between $\mathbf{X}$ and $\hat{\mathbf{X}}$ is

$$\mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})] \overset{(a)}{=} \int_{\mathbf{x}}\int_{\mathbf{y}}\int_{\hat{\mathbf{x}}} f(\mathbf{x}) f(\mathbf{y}|\mathbf{x}) f(\hat{\mathbf{x}}|\mathbf{y}) d(\mathbf{x},\hat{\mathbf{x}}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\mathrm{d}\hat{\mathbf{x}} \tag{113c}$$

[19] The logarithms are in base 2 in this section.

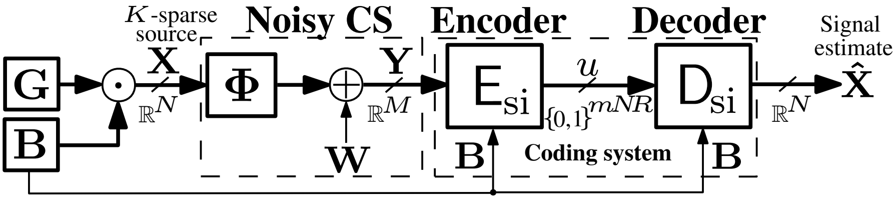


Figure 29: A single-sensor lossy CS of a sparse source with support side information.

where (a) follows from $f(\hat{\mathbf{x}}|\mathbf{y}) = f(\hat{\mathbf{x}}|\mathbf{y},\mathbf{x})$ because $\mathbf{X} \to \mathbf{Y} \to \hat{\mathbf{X}}$ forms a Markov chain. Note that the remote sensing mechanism is captured by the conditional PDF $f(\mathbf{y}|\mathbf{x})$, governed by the measurements in (107).
Due to the time-varying sparsity of $\{\mathbf{X}_n\}_{n=1}^{\infty}$ through $\{\mathbf{B}_n\}_{n=1}^{\infty}$, the PDFs of $\mathbf{X}$, and consequently, of $\mathbf{Y}$ are mixture distributions, which seems to make the optimization over $f(\hat{\mathbf{x}}|\mathbf{y})$ in (113a) difficult. Hence, we treat the lossy CS problem of Definition 2 with the following two approaches: 1) analytically tractable *lower bound* to $R_{\mathbf{X}}^{\mathrm{rem}}(D)$ is derived in Section 7, and 2) a method to numerically approximate $R_{\mathbf{X}}^{\mathrm{rem}}(D)$ is devised in Section 7. Note that similar difficulty resides also in the *direct* compression of $\mathbf{X}$, in which case only RD bounds have been derived [57, 293, 230].

Rate-Distortion Lower Bound for Lossy CS

Consider the compression setup of Fig. 29, where, compared to Fig. 28, the encoder E_{si} and decoder D_{si} possess SI on sequence $\{\mathbf{B}_n\}_{n=1}^{\infty}$. Such an SI aided setup can be used to derive a lower bound to $R_{\mathbf{X}}^{\mathrm{rem}}(D)$ in (113a). Having the support SI at the decoder is often optimistic in practice, but sometimes the encoder may acquire SI on $\mathbf{B}$ (i.e., an estimate $\hat{\mathbf{B}}$) from the measurements $\mathbf{Y}$ at a moderate cost via a sparse signal reconstruction algorithm (see Section 2). Nevertheless, the shared support SI allows to derive an analytically tractable lower bound to $R_{\mathbf{X}}^{\mathrm{rem}}(D)$ which sheds light on the RD behavior of the original setup in Fig. 28, and establishes a benchmark for practical coding methods.

Lossy CS Problem with Support SI

Owing to the support SI, an *informed* lossy source code is defined as follows [131, 174, 73] and [126, Sect. 2.3.1]. The encoder E_{si} observes a block of measurements $\mathbf{y}^m \in \mathcal{Y}^m$ along with the SI $\mathbf{b}^m \in \mathcal{B}^m$ and compresses it to a message index $u \in \mathcal{U}$ using an encoder mapping

$$g^m_{\mathsf{E}_{\mathrm{si}}} : \mathcal{Y}^m \times \mathcal{B}^m \to \mathcal{U}. \tag{114}$$

The decoder D_{si} uses the index and the common SI $\mathbf{b}^m$ to reconstruct an estimate of $\mathbf{x}^m \in \mathcal{X}^m$ via a decoder mapping

$$g^m_{\mathsf{D}_{\mathrm{si}}} : \mathcal{U} \times \mathcal{B}^m \to \hat{\mathcal{X}}^m. \tag{115}$$

A pair (R, D) for distortion $D \geq 0$ is achievable if there exists a sequence of informed $(2^{mNR}, m)$-RD codes with mappings $g^m_{\mathsf{E}_{\mathrm{si}}}$ and $g^m_{\mathsf{D}_{\mathrm{si}}}$ so that $\lim\limits_{m\to\infty} \mathbb{E}\big[d(\mathbf{X}^m, g^m_{\mathsf{D}_{\mathrm{si}}}\{g^m_{\mathsf{E}_{\mathrm{si}}}(\mathbf{Y}^m, \mathbf{B}^m), \mathbf{B}^m\})\big] \leq D$. Let $\mathcal{R}_{\mathrm{si}}$ be the closure of the set of such achievable (R, D) pairs.

Definition 3. (Lossy CS source coding problem with support SI) *Amongst all the* E_{si}-D_{si} *pairs of mappings* (114) *and* (115), *determine the infimum of (achievable) rates R so that $\mathbf{X}$ can be reproduced with the average distortion satisfying* $\mathbb{E}\big[d(\mathbf{X}^m, \hat{\mathbf{X}}^m)\big] \leq D + \epsilon$ *for any positive real number ϵ, i.e., define*

$$R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D) = \inf_{(R,D)\in\mathcal{R}_{\mathrm{si}}} R. \tag{116}$$

$R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ *is called* the conditional remote RDF *of source* $\mathbf{X}$. *Clearly,* $\mathcal{R} \subseteq \mathcal{R}_{\mathrm{si}}$*, and* $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ *establishes a lower bound to the best possible compression performance of the lossy CS as*

$$R^{\mathrm{rem}}_{\mathbf{X}}(D) \geq R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D). \tag{117}$$

Next, we present detailed derivation of $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$.

Conditional Remote RDF

The conditional RDF for a discrete source along with the respective coding theorems has been derived in [131]. Extending the results to a *remote* compression setup, the conditional remote RDF $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ can be expressed as

$$R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D) = \min_{\{f(\hat{\mathbf{x}}|\mathbf{y},\mathbf{b}_s)\}_{s=1}^{|\mathcal{B}|} : \mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})] \leq D} \frac{1}{N} I(\mathbf{Y}; \hat{\mathbf{X}}|\mathbf{B}) \tag{118a}$$

where the optimization is over the $|\mathcal{B}|$ different test channels $f(\hat{\mathbf{x}}|\mathbf{y},\mathbf{b}_s)$, $s=1,\ldots,|\mathcal{B}|$, the conditional mutual information between $\mathbf{Y}$ and $\hat{\mathbf{X}}$ given $\mathbf{B}$ is

$$I(\mathbf{Y};\hat{\mathbf{X}}|\mathbf{B}) = \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) I(\mathbf{Y};\hat{\mathbf{X}}|\mathbf{B}=\mathbf{b}_s) \tag{118b}$$

and the average MSE distortion between $\mathbf{X}$ and $\hat{\mathbf{X}}$ is

$$\mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})] = \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s)\mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})|\mathbf{B}=\mathbf{b}_s] \tag{118c}$$

where, compared to (113c), the expectation is also taken over $\mathbf{B}$. Since $\mathbf{B}$ is provided at no cost in the system shown in Fig. 29, $R^{\text{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ determines the complementary information rate that must be conveyed to the decoder D_{si} to reconstruct $\mathbf{X}$ with fidelity D.

We can observe that (118b) and (118c) decompose with respect to realizations $\mathbf{B}=\mathbf{b}_s$, $s=1,\ldots,|\mathcal{B}|$. The conditional remote RDF $R^{\text{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ in (118a) can thus be expressed as the weighted sum minimization [131, Theorem 5]

$$R^{\text{rem}}_{\mathbf{X}|\mathbf{B}}(D) = \min_{\substack{\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s)D_s = D \\ D_s \geq 0,\, s=1,\ldots,|\mathcal{B}|}} \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) R^{\text{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s) \tag{119}$$

with optimization variables D_s, $s=1,\ldots,|\mathcal{B}|$, where $R^{\text{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s)$ is *the conditional marginal remote RDF* of source $\mathbf{X}$ for a fixed realization $\mathbf{B}=\mathbf{b}_s$ and distortion $D_s \geq 0$, given as

$$R^{\text{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s) = \min_{f(\hat{\mathbf{x}}|\mathbf{y},\mathbf{b}_s):\mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})|\mathbf{B}=\mathbf{b}_s]\leq D_s} \frac{1}{N} I(\mathbf{Y};\hat{\mathbf{X}}|\mathbf{B}=\mathbf{b}_s) \tag{120a}$$

where the mutual information between $\mathbf{Y}$ and $\hat{\mathbf{X}}$, conditioned on $\mathbf{B}=\mathbf{b}_s$, is

$$I(\mathbf{Y};\hat{\mathbf{X}}|\mathbf{B}=\mathbf{b}_s) = \int_{\mathbf{y}}\int_{\hat{\mathbf{x}}} f(\mathbf{y}|\mathbf{b}_s) f(\hat{\mathbf{x}}|\mathbf{y},\mathbf{b}_s) \log\frac{f(\hat{\mathbf{x}}|\mathbf{y},\mathbf{b}_s)}{f(\hat{\mathbf{x}}|\mathbf{b}_s)} \mathrm{d}\mathbf{y}\mathrm{d}\hat{\mathbf{x}} \tag{120b}$$

and the average MSE distortion between $\mathbf{X}$ and $\hat{\mathbf{X}}$, conditioned on $\mathbf{B}=\mathbf{b}_s$, is

$$\mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})|\mathbf{B}=\mathbf{b}_s] \overset{(a)}{=} \int_{\mathbf{x}}\int_{\mathbf{y}}\int_{\hat{\mathbf{x}}} f(\mathbf{x}|\mathbf{b}_s) f(\mathbf{y}|\mathbf{x},\mathbf{b}_s) f(\hat{\mathbf{x}}|\mathbf{y},\mathbf{b}_s) \, d(\mathbf{x},\hat{\mathbf{x}})\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y}\mathrm{d}\hat{\mathbf{x}} \tag{120c}$$

where (a) follows from $f(\hat{\mathbf{x}}|\mathbf{y},\mathbf{x},\mathbf{b}_s) = f(\hat{\mathbf{x}}|\mathbf{y},\mathbf{b}_s)$ because $\mathbf{X}\to\mathbf{Y}\to\hat{\mathbf{X}}$ forms a Markov chain when conditioned on $\mathbf{B}$. Owing to the support SI, all

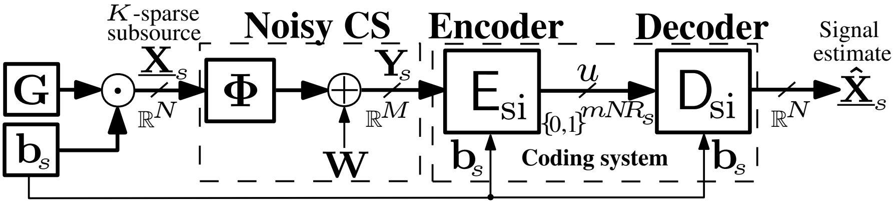


Figure 30: A single-sensor lossy CS of a sparse subsource with support side information.

the PDFs above are equivalent to those in (113), except they are conditioned on the realization $\mathbf{B} = \mathbf{b}_s$.

Based on the above formulations, the characterization of $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ in (119) boils down to deriving each $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s)$, $s = 1, \ldots, |\mathcal{B}|$, in (120a). This is carried out in the next section.

Conditional Marginal Remote RDF

The conditional marginal remote RDF $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s)$, $s = 1, \ldots, |\mathcal{B}|$, in (120a) determines the minimum (achievable) rate R_s so that $\mathbf{X}$ can be reproduced with the average distortion satisfying $\mathbb{E}\big[d(\mathbf{X}, \hat{\mathbf{X}})|\mathbf{B} = \mathbf{b}_s\big] \leq D_s$ in the setup depicted in Fig. 30, where $\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) R_s = R$. In order to derive $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s)$, we introduce the following three definitions.

Definition 4. (Subsource) *Let $\{\underline{\mathbf{X}}_{s,n}\}_{n=1}^{\infty} = \{\mathbf{G}_n \odot \mathbf{b}_s\}_{n=1}^{\infty}$ be the memoryless sequence of the sth* subsource, *consisting of K-sparse source vectors $\{\mathbf{X}_n\}_{n=1}^{\infty}$ restricted to a fixed realization $\mathbf{B} = \mathbf{b}_s$, $s = 1, \ldots, |\mathcal{B}|$. Each subsource $\underline{\mathbf{X}}_s$ comprises of two parts: 1) the length-K random vector*

$$\mathbf{X}_s \triangleq \mathbf{G}_{\mathrm{supp}(\mathbf{b}_s)} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{X}_s}) \tag{121}$$

that extracts the entries of $\underline{\mathbf{X}}_s$ (i.e., the entries of $\mathbf{G}$) restricted to the support of $\mathbf{b}_s$, where $\mathrm{supp}(\mathbf{b}_s) \triangleq \{k \in \{1, \ldots, N\} | b_{s,k} \neq 0\}$ denotes the support of vector $\mathbf{b}_s$, $\mathbf{G}_{\mathrm{supp}(\mathbf{b}_s)}$ extracts the entries G_k from $\mathbf{G}$ for indices $k \in \mathrm{supp}(\mathbf{b}_s)$, and the covariance matrix $\mathbf{\Sigma}_{\mathbf{X}_s} \in \mathbb{S}^K_{++}$ extracts the entries $\mathbf{\Sigma}_{\mathbf{G}}(k, k')$ from $\mathbf{\Sigma}_{\mathbf{G}}$ for indices $k, k' \in \mathrm{supp}(\mathbf{b}_s)$; 2) the all-zero vector $\mathbf{0}_{N-K}$ corresponding to the entries of $\underline{\mathbf{X}}_s$ for indices $k \in \mathrm{supp}(\mathbf{b}_s)^{\mathrm{c}}$, where $\mathrm{supp}(\mathbf{b}_s)^{\mathrm{c}} \triangleq \{\{1, \ldots, N\} \setminus \mathrm{supp}(\mathbf{b}_s)\}$ is the complement of $\mathrm{supp}(\mathbf{b}_s)$.

It is worth noting that the subsources $\underline{\mathbf{X}}_s$, $s = 1, \ldots, |\mathcal{B}|$, are virtual, i.e., not actually present in the system. However, they play an instructive role in the *derivations. Due to the decomposability, the subsources can be seen as a* composite source[20] *[25, Sect. 6.1.1] and [126].*

Definition 5. *(*Measurements of a subsource*) Let* $\{\mathbf{Y}_{s,n}\}_{n=1}^{\infty}$ *be the memoryless sequence of the measurements of form* (107) *restricted to a fixed realization* $\mathbf{B} = \mathbf{b}_s$*, i.e., the measurements of subsource* $\underline{\mathbf{X}}_s$*,* $s = 1, \dots, |\mathcal{B}|$*, in* (121)*, defined as*

$$\mathbf{Y}_s \triangleq \mathbf{\Phi}\underline{\mathbf{X}}_s + \mathbf{W} = \mathbf{\Phi}_s \mathbf{X}_s + \mathbf{W} \tag{122}$$

where matrix $\mathbf{\Phi}_s \in \mathbb{R}^{M \times K}$ *extracts the* K *columns of* $\mathbf{\Phi}$ *with indices* $k \in \mathrm{supp}(\mathbf{b}_s)$*, and thus,* $\mathbf{Y}_s \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{Y}_s})$ *with a covariance matrix* $\mathbf{\Sigma}_{\mathbf{Y}_s} = \mathbf{\Phi}_s \mathbf{\Sigma}_{\mathbf{X}_s} \mathbf{\Phi}_s^T + \mathbf{\Sigma}_{\mathbf{W}} \in \mathbb{S}_{++}^{M}$*.*

Definition 6. *(*MMSE estimator of a subsource*) Let* $\underline{\mathbf{Z}}_s$ *be a length-*N *random vector representing the MMSE estimator of source* $\mathbf{X}$ *given* $\mathbf{Y}$ *for a fixed realization* $\mathbf{B} = \mathbf{b}_s$*, i.e., the MMSE estimator of subsource* $\underline{\mathbf{X}}_s$ *in* (121) *given* $\mathbf{Y}_s$ *in* (122)*. Each* $\underline{\mathbf{Z}}_s$ *is given by the conditional expectation as [229, Sect. 8.2]*

$$\underline{\mathbf{Z}}_s \triangleq \mathbb{E}[\mathbf{X}|\mathbf{Y}, \mathbf{B} = \mathbf{b}_s], \; s = 1, \dots, |\mathcal{B}|, \tag{123}$$

which, owing to the sparsity of $\underline{\mathbf{X}}_s$ *(cf.* (121)*), splits into two parts: 1) the length-*K *random vector*

$$\mathbf{Z}_s \triangleq \mathbb{E}[\mathbf{X}_s|\mathbf{Y}, \mathbf{B} = \mathbf{b}_s] = \mathbf{\Sigma}_{\mathbf{X}_s\mathbf{Y}_s} \mathbf{\Sigma}_{\mathbf{Y}_s}^{-1} \mathbf{Y}_s = \mathbf{F}_s \mathbf{Y}_s \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{Z}_s}) \tag{124}$$

that represents the MMSE estimator of $\mathbf{X}_s$ *given* $\mathbf{Y}$ *and* $\mathbf{B} = \mathbf{b}_s$*; 2)* $\mathbf{0}_{N-K}$ *that corresponds to the MMSE estimator of the zero part of* $\underline{\mathbf{X}}_s$*. For jointly Gaussian random vectors,* $\mathbf{Z}_s$ *is linear [120, Sect. 10.2], where the cross-covariance matrix is* $\mathbf{\Sigma}_{\mathbf{X}_s\mathbf{Y}_s} = \mathbf{\Sigma}_{\mathbf{X}_s} \mathbf{\Phi}_s^T \in \mathbb{R}^{K \times M}$*, the MMSE estimation matrix is* $\mathbf{F}_s \triangleq \mathbf{\Sigma}_{\mathbf{X}_s\mathbf{Y}_s} \mathbf{\Sigma}_{\mathbf{Y}_s}^{-1} \in \mathbb{R}^{K \times M}$*, and* $\mathbf{Z}_s \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{Z}_s})$ *with covariance matrix* $\mathbf{\Sigma}_{\mathbf{Z}_s} = \mathbf{F}_s \mathbf{\Sigma}_{\mathbf{X}_s\mathbf{Y}_s}^T \in \mathbb{S}_{+}^{K}$*.*

$R_{\mathbf{X}|\mathbf{b}_s}^{\mathrm{rem}}(D_s)$ in (120a) can be characterized by a two-stage encoding structure, where the encoder first optimally estimates the subsource $\underline{\mathbf{X}}_s$ (see (121)) from measurements $\mathbf{Y}_s$ (see (122)), and then optimally encodes the constructed estimator $\underline{\mathbf{Z}}_s$ in (124). This is elaborated next.

[20] The sequence pair $\{(\mathbf{X}_n, \mathbf{B}_n)\}_{n=1}^{\infty}$ forms a jointly stationary and ergodic regenerative composite source with stationary memoryless subsource processes; $\{\mathbf{B}_n\}_{n=1}^{\infty}$ is the hidden switch sequence that controls the output process $\{\mathbf{X}_n\}_{n=1}^{\infty}$ by randomly activating the subsources $\{\underline{\mathbf{X}}_{s,n}\}_{n=1}^{\infty}$ according to probabilities $p(\mathbf{b}_s)$, $s = 1, \dots, |\mathcal{B}|$ [25, Sect. 6.1.1] and [126].

MMSE Distortion Separation

Let $\underline{\hat{\mathbf{X}}}_s$ be a length-N random vector representing the reproduction of subsource $\underline{\mathbf{X}}_s$ at the decoder output (see Fig. 30). Accordingly, the average conditional MSE distortion $\mathbb{E}\big[d(\underline{\mathbf{X}}_s,\underline{\hat{\mathbf{X}}}_s)\big] \triangleq \mathbb{E}\big[d(\mathbf{X},\hat{\mathbf{X}})|\mathbf{B}=\mathbf{b}_s\big]$ in (120c) separates as

$$\begin{aligned}\mathbb{E}\big[d(\underline{\mathbf{X}}_s,\underline{\hat{\mathbf{X}}}_s)\big] &= N^{-1}\mathbb{E}\big[\|\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s+\underline{\mathbf{Z}}_s-\underline{\hat{\mathbf{X}}}_s\|_2^2\big]\\ &\overset{(a)}{=} N^{-1}\mathbb{E}\big[\|\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s\|_2^2\big]+N^{-1}\mathbb{E}\big[\|\underline{\mathbf{Z}}_s-\underline{\hat{\mathbf{X}}}_s\|_2^2\big]\\ &\overset{(b)}{=} D_{\mathbf{Z}|\mathbf{b}_s}+\mathbb{E}\big[d(\underline{\mathbf{Z}}_s,\underline{\hat{\mathbf{X}}}_s)\big]\end{aligned} \tag{125}$$

where (a) follows from the MMSE orthogonality principle [229, Sect. 8.2.1] (see Appendix C); (b) follows by denoting the (rate-dependent) average MSE distortion between $\underline{\mathbf{Z}}_s$ and $\underline{\hat{\mathbf{X}}}_s$ as $\mathbb{E}\big[d(\underline{\mathbf{Z}}_s,\underline{\hat{\mathbf{X}}}_s)\big] = N^{-1}\mathbb{E}\big[\|\underline{\mathbf{Z}}_s-\underline{\hat{\mathbf{X}}}_s\|_2^2\big]$, and defining the (rate-independent) average MMSE estimation error with respect to subsource $\underline{\mathbf{X}}_s$ as [120, Sect. 10.2] (see Appendix D)

$$D_{\mathbf{Z}|\mathbf{b}_s} = N^{-1}\mathrm{Tr}\big(\mathbf{\Sigma}_{\mathbf{X}_s}-\mathbf{\Sigma}_{\mathbf{Z}_s}\big). \tag{126}$$

Similar separation appears also in, e.g., [85, 253, 304].

Reduced Distortion

Due to the decomposability of (124), the last term in (125) splits as

$$\mathbb{E}\big[d(\underline{\mathbf{Z}}_s,\underline{\hat{\mathbf{X}}}_s)\big] = \mathbb{E}\big[d(\mathbf{Z}_s,\hat{\mathbf{X}}_s)\big]+\mathbb{E}\big[d(\mathbf{0}_{N-K},\underline{\hat{\mathbf{X}}}_{\mathrm{supp}(\mathbf{b}_s)^{\mathrm{c}}})\big] \tag{127}$$

where $\hat{\mathbf{X}}_s$ is the length-K reproduction random vector associated with $\mathbf{X}_s$, and $\underline{\hat{\mathbf{X}}}_{\mathrm{supp}(\mathbf{b}_s)^{\mathrm{c}}}$ is the reproduction random vector associated with the zero part of $\underline{\mathbf{X}}_s$. Since an RDF is a monotonic nonincreasing function of the distortion [25, Sect. 2], it is optimal for $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s)$ to set $\underline{\hat{\mathbf{X}}}_{\mathrm{supp}(\mathbf{b}_s)^{\mathrm{c}}} = \mathbf{0}_{N-K}$, and thus the distortion in (125) reduces to

$$\mathbb{E}\big[d(\underline{\mathbf{X}}_s,\underline{\hat{\mathbf{X}}}_s)\big] = D_{\mathbf{Z}|\mathbf{b}_s}+\mathbb{E}\big[d(\mathbf{Z}_s,\hat{\mathbf{X}}_s)\big]. \tag{128}$$

Let $D'_s \geq 0$ be a *reduced* distortion criterion for the sth subsource as

$$D'_s \triangleq D_s - D_{\mathbf{Z}|\mathbf{b}_s} \geq 0,\ s=1,\ldots,|\mathcal{B}|, \tag{129}$$

where $D_s \geq 0$ is the distortion criterion in (120a), and $D_{\mathbf{Z}|\mathbf{b}_s}$ is given in (126). Note that according to (128), $\mathbb{E}\big[d(\mathbf{Z}_s,\hat{\mathbf{X}}_s)\big] \leq D'_s$ implies $\mathbb{E}\big[d(\underline{\mathbf{X}}_s,\underline{\hat{\mathbf{X}}}_s)\big] \leq D_s$.

Estimate-and-Compress Separation

Let $R^{\text{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s)$ denote the *direct* RDF of the MMSE estimator $\mathbf{Z}_s$ defined in (124) for the reduced distortion D'_s in (129), i.e., define

$$R^{\text{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s) = \min_{f(\hat{\mathbf{z}}_s|\mathbf{z}_s):\mathbb{E}[d(\mathbf{Z}_s,\hat{\mathbf{Z}}_s)]\le D'_s} \frac{1}{N} I(\mathbf{Z}_s;\hat{\mathbf{Z}}_s) \tag{130a}$$

where the minimization is over the test channel $f(\hat{\mathbf{z}}_s|\mathbf{z}_s)$, $\hat{\mathbf{Z}}_s$ is a length-K reproduction random vector for $\mathbf{Z}_s$, the average mutual information between $\mathbf{Z}_s$ and $\hat{\mathbf{Z}}_s$ is

$$I(\mathbf{Z}_s;\hat{\mathbf{Z}}_s) = \int_{\mathbf{z}_s}\int_{\hat{\mathbf{z}}_s} f(\mathbf{z}_s) f(\hat{\mathbf{z}}_s|\mathbf{z}_s) \log\frac{f(\hat{\mathbf{z}}_s|\mathbf{z}_s)}{f(\hat{\mathbf{z}}_s)} \mathrm{d}\mathbf{z}_s \mathrm{d}\hat{\mathbf{z}}_s \tag{130b}$$

and the average MSE distortion between $\mathbf{Z}_s$ and $\hat{\mathbf{Z}}_s$ is

$$\mathbb{E}[d(\mathbf{Z}_s,\hat{\mathbf{Z}}_s)] = \int_{\mathbf{z}_s}\int_{\hat{\mathbf{z}}_s} f(\mathbf{z}_s) f(\hat{\mathbf{z}}_s|\mathbf{z}_s) d(\mathbf{z}_s,\hat{\mathbf{z}}_s) \mathrm{d}\mathbf{z}_s \mathrm{d}\hat{\mathbf{z}}_s. \tag{130c}$$

The RDF $R^{\text{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s)$ can be derived by decorrelating the Gaussian (effective) source $\mathbf{Z}_s \sim \mathcal{N}(\mathbf{0},\mathbf{\Sigma}_{\mathbf{Z}_s})$ via the Karhunen-Loève transform, and applying reverse water-filling [74, Sect. 10.3.3]. Accordingly, let $\mathbf{\Sigma}_{\mathbf{Z}_s} = \mathbf{Q}_s\mathbf{\Lambda}_s\mathbf{Q}_s^{\mathrm{T}}$ be the eigendecomposition, where the diagonal matrix $\mathbf{\Lambda}_s \triangleq \operatorname{diag}(\lambda_{s,1},\ldots,\lambda_{s,K})$ contains the eigenvalues $\lambda_{s,1} \ge \ldots \ge \lambda_{s,K} \ge 0$ of $\mathbf{\Sigma}_{\mathbf{Z}_s} \in \mathbb{S}^K_+$, and the columns of $\mathbf{Q}_s \in \mathbb{R}^{K\times K}$ are the corresponding eigenvectors. Consequently, $R^{\text{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s)$ is given as

$$R^{\text{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s) = \min_{\substack{\sum_{k=1}^K D'_{s,k}=D'_s \\ D'_{s,k}\ge 0,\, k=1,\ldots,K}} \frac{1}{N}\textstyle\sum_{k=1}^K \max\left\{0, \frac{1}{2}\log\frac{\lambda_{s,k}}{D'_{s,k}}\right\} \tag{131}$$

where $D'_{s,k}$, $k=1,\ldots,K$, are the optimization variables.
The following proposition gives an expression for the conditional marginal remote RDF $R^{\text{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s)$.
Proposition 6: *The conditional marginal remote RDF of $\underline{\mathbf{X}}_s$ in (120a) is given as the (direct) RDF of the MMSE estimator $\mathbf{Z}_s$ in (131), i.e.,*

$$R^{\text{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s) = R^{\text{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s),\ s=1,\ldots,|\mathcal{B}|, \tag{132}$$

where $D'_s = D_s - D_{\mathbf{Z}|\mathbf{b}_s} \ge 0$ is the reduced distortion in (129), and $D_{\mathbf{Z}|\mathbf{b}_s}$ is given in (126).
Proof: The proposition follows from the proofs in [85, 304].

The version of record is available at: http://dx.doi.org/10.1561/2000000107

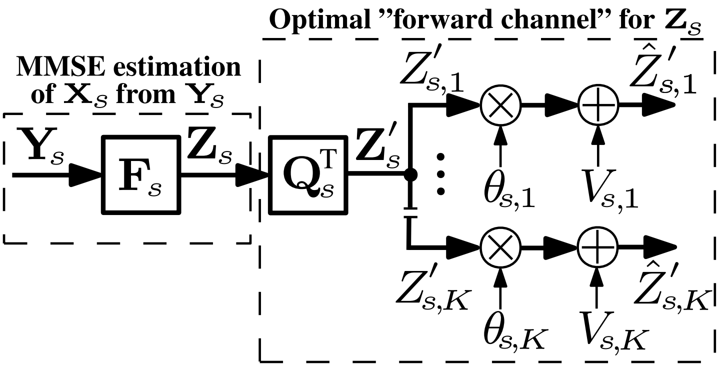


Figure 31: A forward channel to illustrate the optimal compression structure with respect to the conditional marginal remote RDF $R^{\text{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s)$ in (120a).

According to Proposition 6, the remote source coding problem of Definition 3 separates into 1) the MMSE estimation of $\underline{\mathbf{X}}_s$ given $\mathbf{Y}_s$, and 2) the derivation of the RDF of the resultant estimator. On this account, the best encoder E_{si} comprises of the MSE-optimal extraction of the subsources $\underline{\mathbf{X}}_s$ from the noisy linear measurements $\mathbf{Y}_s$ in (122), $s = 1,\ldots,|\mathcal{B}|$, followed by the optimal coding of the extracted messages. The estimate-and-compress separation is illustrated in Fig. 31.

Remark 7.1. *The two expressions* (130) *and* (131) *for* $R^{\text{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s)$ *are interrelated by Gaussian* "forward channels" *depicted in Fig. 31. Namely, the optimal conditional PDF* $f(\hat{\mathbf{z}}_s|\mathbf{z}_s)$ *can be described via* $\hat{Z}'_{s,k} \triangleq \theta_{s,k}Z'_{s,k} + V_{s,k}$, $k = 1,\ldots,K$, *with parameters* $\theta_{s,k} = \frac{\lambda_{s,k}-D'_{s,k}}{\lambda_{s,k}}$ *and* $\sigma^2_{V_{s,k}} = \theta_{s,k}D'_{s,k}$, *where* $D'_{s,k}$ *are optimal variables for* (131), $Z'_{s,k}$ *is the kth element of the decorrelated MMSE estimator* $\mathbf{Z}'_s = \mathbf{Q}_s^T\mathbf{Z}_s$, *and* $V_{s,k} \sim \mathcal{N}(0, \sigma^2_{V_{s,k}})$ *is a zero mean Gaussian random variable independent of* $Z'_{s,k}$ *[25, Theorem 4.3.2]. Thus, the forward channel model provides a practical way to realize the optimal conditional PDF and the respective reproduction random vectors.*

Remark 7.2. *If* $\operatorname{rank}(\mathbf{\Sigma}_{\mathbf{Z}_s}) < K$, *then the covariance matrix* $\mathbf{\Sigma}_{\mathbf{Z}_s}$ *has a nullspace. Consequently, the random vector* $\mathbf{Z}_s$ *is a degenerate Gaussian vector [33, Lecture 7], and* $\mathbf{Z}'_s$ *contains a deterministic zero part. This case is inherently handled in* (131) *by allocating* $D'_{s,k} = 0$ *for* $k = \operatorname{rank}(\mathbf{\Sigma}_{\mathbf{Z}_s}) + 1,\ldots,K$.

The null-space may be caused for example by the rank deficiency of matrix $\boldsymbol{\Phi}_s$.

Remark 7.3. *A proof of the optimality of the two-step coding structure is implicitly present in the seminal work by Dobrushin and Tsybakov [85, Sect. 5] for the case with frequency-weighted MSE distortion where the source and observable processes are jointly Gaussian and stationary. Furthermore, they proved such optimality explicitly for the MSE distortion in the case where observations are noisy versions of the signal (i.e., no dimension reduction) [85, Sect. 7]. Later, Wolf and Ziv [304] addressed a DR framework and proved that separation holds for MSE distortion under more general conditions (i.e., Gaussianity is not required). Consequently, the decomposition principle of Proposition 6 is also valid for non-Gaussian sources/observations; however, finding analytical expressions for* $R^{\mathrm{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s)$ *and* $D_{\mathbf{Z}|\mathbf{b}_s}$ *may be difficult. Similar separation results appear in, e.g., [303, 256, 163] and [25, Sect. 4.5.4].*

Remark 7.4. $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s)$ *is an upper bound to the conditional marginal remote RDF of a subsource* $\underline{\tilde{\mathbf{X}}}_s = \tilde{\mathbf{G}} \odot \mathbf{b}_s$, *where* $\tilde{\mathbf{G}}$ *is a non-Gaussian random vector with covariance matrix* $\boldsymbol{\Sigma}_{\tilde{\mathbf{G}}} = \boldsymbol{\Sigma}_{\mathbf{G}}$ *[25, p. 130].*

Characterization of the Conditional Remote RDF

Let $D_{\mathbf{Z}|\mathbf{B}} \geq 0$ denote the total average MMSE estimation error over all subsources $\underline{\mathbf{X}}_s$, $s = 1, \ldots, |\mathcal{B}|$, with the support SI, i.e.,

$$\begin{aligned} D_{\mathbf{Z}|\mathbf{B}} &\triangleq \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) D_{\mathbf{Z}|\mathbf{b}_s} \\ &\overset{(a)}{=} N^{-1} \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) \mathrm{Tr}\left(\boldsymbol{\Sigma}_{\mathbf{X}_s} - \boldsymbol{\Sigma}_{\mathbf{Z}_s}\right) \end{aligned} \tag{133}$$

where (a) follows from (126). The conditional remote RDF $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ is given by the following theorem.

Theorem 1. *For distortion range* $D_{\mathbf{Z}|\mathbf{B}} \leq D < \frac{1}{N} \sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) \mathrm{Tr}(\boldsymbol{\Sigma}_{\mathbf{X}_s})$, $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ *is positive and can be evaluated via the convex minimization problem as*

$$R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D) = \min_{\substack{\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) \sum_{k=1}^{K} D'_{s,k} = D - D_{\mathbf{Z}|\mathbf{B}} \\ D'_{s,k} \geq 0,\, k=1,\ldots,K,\, s=1,\ldots,|\mathcal{B}|}} N^{-1} \sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) \sum_{k=1}^{K} \max\left\{0, \frac{1}{2} \log \frac{\lambda_{s,k}}{D'_{s,k}}\right\} \tag{134}$$

where $\mathbf{\Sigma}_{\mathbf{X}_s}$ is the covariance matrix of $\mathbf{X}_s$ in (121); $\lambda_{s,1}, \ldots, \lambda_{s,K}$ are the eigenvalues of the covariance matrix $\mathbf{\Sigma}_{\mathbf{Z}_s}$ of $\mathbf{Z}_s$ in (124); $D'_{s,k}$ are the optimization variables, $k = 1, \ldots, K$, $s = 1, \ldots, |\mathcal{B}|$. If the distortion values satisfy $D \geq \frac{1}{N}\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s)\mathrm{Tr}(\mathbf{\Sigma}_{\mathbf{X}_s})$, then $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ is zero.

Proof: The proof is given in Appendix E.

Remark 7.5. *$R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ is an upper bound to the conditional remote RDF of a source $\tilde{\mathbf{X}} = \tilde{\mathbf{G}} \odot \mathbf{B}$, where $\tilde{\mathbf{G}}$ is a non-Gaussian random vector with covariance matrix $\mathbf{\Sigma}_{\tilde{\mathbf{G}}} = \mathbf{\Sigma}_{\mathbf{G}}$ [25, p. 130].*

In Theorem 1, $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ is determined by a weighted sum of the RDFs of the MMSE estimators $\mathbf{Z}_s$ under a reduced distortion criterion, where the weights, i.e., the prior probabilities of the sparsity patterns $p(\mathbf{b}_s)$, $s = 1, \ldots, |\mathcal{B}|$, represent the "appearance frequencies" of such estimators. In particular, (134) involves finding the optimal allocation of the distortion components not only for the $|\mathcal{B}|$ different sparsity patterns, but also for the K entries of each decorrelated random vector $\mathbf{Z}'_s$. This type of weighted minimization is discernibly a consequence of the composite source structure.

Furthermore, $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ reflects the remote sensing nature of the lossy CS: regardless of the rate, the lowest achievable distortion is ultimately dictated by $D_{\mathbf{Z}|\mathbf{B}}$ which is a constant term solely governed by the noisy measurement model in (122). This unavoidable degradation in compression performance, which is caused by the indirect observations of the source, distinguishes the lossy CS from directly compressing $\mathbf{X}$; see, e.g., the works in [57, 293, 230] which derive RD bounds for compressing sparse sources. Note that a constant distortion floor occurs whether or not the support SI is available and only the respective levels for $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ and $R^{\mathrm{rem}}_{\mathbf{X}}(D)$ are different. This is demonstrated by the numerical results in Section 7.

Numerical Approximation of the Remote RDF

Since finding an *analytical solution* for the lossy CS problem of Definition 2 turned out to be elusive, we develop a method based on the *Blahut-Arimoto* (BA) algorithm [30, 11] to *numerically approximate* the remote RDF $R^{\mathrm{rem}}_{\mathbf{X}}(D)$ in (113a). The standard BA algorithm is designed for direct (i.e., for intact measurements $\mathbf{Y} = \mathbf{X}$) source coding with discrete input/output alphabets.

Hence, the algorithm needs to be adapted to handle 1) continuous-valued signals $\mathbf{X}$ and $\mathbf{Y}$, and 2) the remote compression setup. The former is accomplished by a VQ-optimized alphabet discretization method, and the latter by appropriately modifying the distortion measure.

Discretization of Signal Alphabets

As the first step, the measurement vector space $\mathcal{Y}$ (i.e., the encoder input) and the reproduction alphabet $\hat{\mathcal{X}}$ (i.e., the decoder output) are discretized via a VQ. Let $\mathcal{V} \triangleq \{1, \ldots, |\mathcal{V}|\}$ be an index set. The $|\mathcal{V}|$-level VQ is determined by 1) the encoder regions $\mathcal{S}_v$, $v \in \mathcal{V}$, which partition the measurement space, i.e., $\mathcal{S}_v \subseteq \mathcal{Y}$, $\mathcal{S}_v \cap \mathcal{S}_{v'} = \emptyset$, for any $v \neq v'$, and $\bigcup_{v=1}^{|\mathcal{V}|} \mathcal{S}_v = \mathcal{Y}$; 2) the reconstruction codebook $\hat{\mathcal{X}}^{\mathrm{q}} \triangleq \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_{|\mathcal{V}|}\}$ with codevectors $\hat{\mathbf{x}}_v \in \mathbb{R}^N$, $v \in \mathcal{V}$. The VQ encoder is a mapping $\mathsf{E}^{\mathrm{vq}} : \mathcal{Y} \to \mathcal{V}$ such that for an input $\mathbf{y} \in \mathcal{S}_v$, it produces an index $\mathsf{E}^{\mathrm{vq}}(\mathbf{y}) = v \in \mathcal{V}$. The VQ decoder performs an inverse mapping $\mathsf{D}^{\mathrm{vq}} : \mathcal{V} \to \hat{\mathcal{X}}^{\mathrm{q}}$ as $\mathsf{D}^{\mathrm{vq}}(v) = \hat{\mathbf{x}}_v \in \hat{\mathcal{X}}^{\mathrm{q}}$. The random variable V represents the VQ output.

Since the next section elucidates the main role of the VQ in RD approximation, the optimization of the VQ is deferred until Section 7.

Modified Blahut-Arimoto Algorithm for Lossy CS

Consider now a VQ as described above with $p(v) \triangleq \Pr(V = v) = \int_{\mathbf{y}\in\mathcal{S}_v} f(\mathbf{y})\mathrm{d}\mathbf{y}$, $v \in \mathcal{V}$. Consequently, index $v \in \mathcal{V}$ represents all the measurement vectors that belong to VQ region $\mathcal{S}_v$. Similarly, let $\hat{\mathbf{X}}^{\mathrm{q}}$ be a discrete reproduction random vector at the output of decoder D with alphabet $\hat{\mathcal{X}}^{\mathrm{q}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_{|\mathcal{V}|}\}$ (i.e., the VQ codebook). Replacing $\mathbf{Y}$ with V and $\hat{\mathbf{X}}$ with $\hat{\mathbf{X}}^{\mathrm{q}}$ in (113a), $R_{\mathbf{X}}^{\mathrm{rem}}(D)$ can be approximated as

$$R_{\mathbf{X},\mathrm{ba}}^{\mathrm{rem}}(D) = \min_{p(\hat{\mathbf{x}}_j|v):\mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}}^{\mathrm{q}})]\leq D} \frac{1}{N} I(V;\hat{\mathbf{X}}^{\mathrm{q}}) \tag{135a}$$

where the optimization is over the conditional probabilities $p(\hat{\mathbf{x}}_j|v) \triangleq \Pr(\hat{\mathbf{X}}^{\mathrm{q}} = \hat{\mathbf{x}}_j | V = v)$, $v, j \in \mathcal{V}$. The mutual information between V and $\hat{\mathbf{X}}^{\mathrm{q}}$ is

$$I(V;\hat{\mathbf{X}}^{\mathrm{q}}) = \textstyle\sum_{v=1}^{|\mathcal{V}|} \sum_{j=1}^{|\mathcal{V}|} p(v)p(\hat{\mathbf{x}}_j|v)\log\dfrac{p(\hat{\mathbf{x}}_j|v)}{p(\hat{\mathbf{x}}_j)} \tag{135b}$$

and the average distortion between $\mathbf{X}$ and $\hat{\mathbf{X}}^{\mathrm{q}}$ is

$$\mathbb{E}\big[d(\mathbf{X},\hat{\mathbf{X}}^{\mathrm{q}})\big] = \textstyle\sum_{v=1}^{|\mathcal{V}|}\sum_{j=1}^{|\mathcal{V}|} p(v)p(\hat{\mathbf{x}}_j|v)\bar{d}(\mathbf{X},\hat{\mathbf{x}}_j|v) \tag{135c}$$

where $\bar{d}(\mathbf{X},\hat{\mathbf{x}}_j|v) \geq 0$ is the *modified* distortion measure, defined as the average per-letter MSE distortion between $\mathbf{X}$ and $\hat{\mathbf{x}}_j$ conditioned on $V=v$, i.e.,

$$\begin{aligned}
\bar{d}(\mathbf{X},\hat{\mathbf{x}}_j|v) &\triangleq \mathbb{E}\big[d(\mathbf{X},\hat{\mathbf{x}}_j)|V=v\big],\ v,j\in\mathcal{V}\\
&= \frac{1}{N}\int_{\mathbf{y}} \mathbb{E}\big[\|\mathbf{X}-\hat{\mathbf{x}}_j\|_2^2|V=v,\mathbf{Y}=\mathbf{y}\big] f(\mathbf{y}|v)\mathrm{d}\mathbf{y}\\
&= \frac{1}{N}\int_{\mathbf{y}} \frac{p(v|\mathbf{y})}{p(v)}\mathbb{E}\big[\|\mathbf{X}-\hat{\mathbf{x}}_j\|_2^2|V=v,\mathbf{Y}=\mathbf{y}\big] f(\mathbf{y})\mathrm{d}\mathbf{y}\\
&\overset{(a)}{=} \frac{1}{N}\frac{1}{p(v)}\int_{\mathbf{y}\in\mathcal{S}_v} \mathbb{E}\big[\|\mathbf{X}-\hat{\mathbf{x}}_j\|_2^2|\mathbf{Y}=\mathbf{y}\big] f(\mathbf{y})\mathrm{d}\mathbf{y}
\end{aligned} \tag{135d}$$

where (a) follows from the Markov chain $\mathbf{X}\to\mathbf{Y}\to V$, and from $p(v|\mathbf{y})=1$, if $\mathbf{y}\in\mathcal{S}_v$, $v\in\mathcal{V}$, and 0 otherwise. Note that pre-calculated $|\mathcal{V}|^2$ quantities $\bar{d}(\mathbf{X},\hat{\mathbf{x}}_j|v)$ remain fixed in the BA algorithm. In the context of *discrete* remote sources, a distortion measure similar to (135d) appears in, e.g., [25, Sect. 3.5] and [303, 86].

Consider a Lagrangian for (135a) as

$$\begin{aligned}
&\mathfrak{L}(\{p(\hat{\mathbf{x}}_j|v)\}_{v,j=1}^{|\mathcal{V}|},\lambda,\{\nu_v\}_{v=1}^{|\mathcal{V}|}) = \frac{1}{N}\textstyle\sum_{v=1}^{|\mathcal{V}|}\sum_{j=1}^{|\mathcal{V}|} p(v)p(\hat{\mathbf{x}}_j|v)\log\dfrac{p(\hat{\mathbf{x}}_j|v)}{p(\hat{\mathbf{x}}_j)}\\
&+\lambda\textstyle\sum_{v=1}^{|\mathcal{V}|}\sum_{j=1}^{|\mathcal{V}|} p(v)p(\hat{\mathbf{x}}_j|v)\bar{d}(\mathbf{X},\hat{\mathbf{x}}_j|v) + \sum_{v=1}^{|\mathcal{V}|}\nu_v\sum_{j=1}^{|\mathcal{V}|} p(\hat{\mathbf{x}}_j|v)
\end{aligned} \tag{136}$$

where $\lambda>0$ is the Lagrange multiplier associated with the sum distortion constraint, and ν_v, $v\in\mathcal{V}$, are the Lagrange multipliers associated with the valid conditional probability constraints $\sum_{j=1}^{|\mathcal{V}|} p(\hat{\mathbf{x}}_j|v)=1$, $\forall v\in\mathcal{V}$. Following a standard BA procedure, an (R,D) point of $R_{\mathbf{X},\mathrm{ba}}^{\mathrm{rem}}(D)$ in (135a) is obtained by sequentially updating the conditional probabilities $p(\hat{\mathbf{x}}_j|v)$ and the reproduction probabilities $p(\hat{\mathbf{x}}_j)$ for a fixed λ at each iteration $t=1,2,\ldots$ as [74, Sect. 10.8]

$$p(\hat{\mathbf{x}}_j|v)^{t+1} := \frac{p(\hat{\mathbf{x}}_j)^t\exp\big[-\lambda\bar{d}(\mathbf{X},\hat{\mathbf{x}}_j|v)\big]}{\sum_{j'=1}^{|\mathcal{V}|} p(\hat{\mathbf{x}}_{j'})^t\exp\big[-\lambda\bar{d}(\mathbf{X},\hat{\mathbf{x}}_{j'}|v)\big]},\ \forall v,j\in\mathcal{V} \tag{137a}$$

$$p(\hat{\mathbf{x}}_j)^{t+1} := \textstyle\sum_{v=1}^{|\mathcal{V}|} p(\hat{\mathbf{x}}_j|v)^{t+1}p(v),\ \forall j\in\mathcal{V}, \tag{137b}$$

until convergence, and by evaluating the rate according to (135b) and the distortion according to (135c). Hence, different values for λ sweep the curve for $R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$, which approximates the remote RDF $R^{\text{rem}}_{\mathbf{X}}(D)$ in (113a) with an accuracy that increases with the number $|\mathcal{V}|$.

The proposed BA method is summarized in Algorithm 10. The algorithm can be terminated when the quantities do not significantly change (e.g., when $\sum_{j=1}^{|\mathcal{V}|}\left(p(\hat{\mathbf{x}}_j)^t - p(\hat{\mathbf{x}}_j)^{t-1}\right)^2 < \epsilon_{\text{ba}}$ for a pre-defined positive constant $\epsilon_{\text{ba}} > 0$). The algorithm inputs $p(v)$, $\hat{\mathbf{x}}_v$, and $\bar{d}(\mathbf{X}, \hat{\mathbf{x}}_j|v)$, $v, j \in \mathcal{V}$, are the outcomes of the VQ optimization. The optimization of the VQ is addressed next.

Vector Quantization Optimization

In Algorithm 10, the accuracy of distortion evaluation through (135c) is ultimately limited by the $|\mathcal{V}|^2$ fixed quantities $\bar{d}(\mathbf{X}, \hat{\mathbf{x}}_j|v)$, $v, j \in \mathcal{V}$, of (135d). Taking this into account, we optimize the E^{vq}-D^{vq} pair such that it minimizes the average MSE distortion between the source $\mathbf{X}$ and its $|\mathcal{V}|$-level reproduction $\hat{\mathbf{X}}^{\text{q}}$. Thus, the VQ design aims at finding the encoder regions $\mathcal{S}_v$ and codevectors $\hat{\mathbf{x}}_v$, $v \in \mathcal{V}$, as

$$\begin{aligned}
\{\mathcal{S}_v^*, \hat{\mathbf{x}}_v^*\}_{v\in\mathcal{V}} &:= \underset{\{\mathcal{S}_v, \hat{\mathbf{x}}_v\}_{v\in\mathcal{V}}}{\text{argmin}} \frac{1}{N}\mathbb{E}[\|\mathbf{X} - \hat{\mathbf{X}}^{\text{q}}\|_2^2] \\
&:= \underset{\{\mathcal{S}_v, \hat{\mathbf{x}}_v\}_{v\in\mathcal{V}}}{\text{argmin}} \frac{1}{N} \textstyle\sum_{v\in\mathcal{V}} \displaystyle\int_{\mathbf{y}} p(v|\mathbf{y}) \\
&\qquad\qquad \mathbb{E}[\|\mathbf{X} - \hat{\mathbf{x}}_v\|_2^2 | V = v, \mathbf{Y} = \mathbf{y}] f(\mathbf{y})\text{d}\mathbf{y} \\
&\overset{(a)}{:=} \underset{\{\mathcal{S}_v, \hat{\mathbf{x}}_v\}_{v\in\mathcal{V}}}{\text{argmin}} \frac{1}{N} \textstyle\sum_{v\in\mathcal{V}} \displaystyle\int_{\mathbf{y}\in\mathcal{S}_v} \mathbb{E}[\|\mathbf{X} - \hat{\mathbf{x}}_v\|_2^2 | \mathbf{Y} = \mathbf{y}] f(\mathbf{y})\text{d}\mathbf{y} \\
&\overset{(b)}{:=} \underset{\{\mathcal{S}_v, \hat{\mathbf{x}}_v\}_{v\in\mathcal{V}}}{\text{argmin}} \textstyle\sum_{v\in\mathcal{V}} p(v)\bar{d}(\mathbf{X}, \hat{\mathbf{x}}_v|v)
\end{aligned} \tag{138}$$

where (a) follows from the Markov chain $\mathbf{X} \to \mathbf{Y} \to V$, and $p(v|\mathbf{y}) = 1$, if $\mathbf{y} \in \mathcal{S}_v$, $v \in \mathcal{V}$, and 0 otherwise; (b) follows from (135d).

Remark 7.6. *Besides $\bar{d}(\mathbf{X}, \hat{\mathbf{x}}_j|v)$, $v, j \in \mathcal{V}$, the VQ affects the final distortion in* (135c) *through the conditional probabilities $p(\hat{\mathbf{x}}_j|v)$, $v, j \in \mathcal{V}$ – the variables to be optimized in iterative steps* (137a) *and* (137b)*. In addition, the VQ affects the rate approximation in* (135b) *through the index probabilities $p(v)$, $v \in \mathcal{V}$. Therefore, a better approximation of $R^{\text{rem}}_{\mathbf{X}}(D)$ is achievable by*

Algorithm 10 A modified Blahut-Arimoto algorithm for approximating the remote RDF $R_{\mathbf{X}}^{\text{rem}}(D)$

```
Inputs: a) Lagrange multipliers $\lambda > 0$; b) codevectors $\hat{\mathbf{x}}_v$, and index probabilities $p(v)$, $v \in \mathcal{V}$, obtained as described in Section 7; c) modified distortion measures $\bar{d}(\mathbf{X}, \hat{\mathbf{x}}_j|v)$, $v, j \in \mathcal{V}$, of form (135d).
Initializations: a) Set $t := 1$; b) set $p(\hat{\mathbf{x}}_j)^t := 1/|\mathcal{V}|$, $j \in \mathcal{V}$.
for a given $\lambda$
repeat
    1) Update the conditional probabilities $p(\hat{\mathbf{x}}_j|v)^{t+1}$, $v, j \in \mathcal{V}$, according to (137a).
    2) Update the reproduction probabilities $p(\hat{\mathbf{x}}_j)^{t+1}$, $j \in \mathcal{V}$, according to (137b).
    3) Set $t := t + 1$.
until a pre-defined stopping criterion is met.
4) Compute the rate $R_\lambda$ according to (135b), and the distortion $D_\lambda$ according to (135c).
end for
Output: $R_{\mathbf{X},\text{ba}}^{\text{rem}}(D)$ curve determined by the $(R_\lambda, D_\lambda)$ pairs.
```

incorporating the VQ optimization in the iterative loop of Algorithm 10, and, thus, generating a unique VQ for each λ. For example, the mapping approach in [249] adapts the reproduction points within the optimization loop. Nevertheless, the presented non-adaptive discretization yields decent accuracy for all distortion values D, as demonstrated in Section 7.

Remark 7.7. *Note that if one sets the conditional probabilities* (137a) *as $p(\hat{\mathbf{x}}_j|v) = 1$ for $v = j$, and 0 otherwise, the reproduction probabilities become $p(\hat{\mathbf{x}}_j) = p(v)$ for $v = j$, and 0 otherwise in* (137b)*. Thus, the distortion* (135c) *for $R_{\mathbf{X},\text{ba}}^{\text{rem}}(D)$ becomes equal to the VQ distortion in* (138)*, and the rate* (135b) *becomes $R = -\sum_{v=1}^{|\mathcal{V}|} p(v)\log(p(v)) = H(V)$, i.e., the entropy of quantization index V. In this sense, $R_{\mathbf{X},\text{ba}}^{\text{rem}}(D)$ can be seen as a "noisy VQ" that randomizes the mapping $V \to \hat{\mathcal{X}}^{\text{q}}$ via conditional probabilities $p(\hat{\mathbf{x}}_j|v)$ that determine a noisy channel between the encoder output and decoder input.*

The joint optimization over $\mathcal{S}_v$ and $\hat{\mathbf{x}}_v$, $v \in \mathcal{V}$, in (138) is intractable, and thus, a common alternating minimization is used to derive necessary optimality conditions [211, 107, 192, 188]. Then, the proposed VQ is equivalent to the VQ in [265] designed for noiseless channels. The optimal encoder regions for fixed codevectors satisfy a generalized nearest-neighbor condition

$$\mathcal{S}_v^* = \Big\{ \mathbf{y} : \|\mathbf{z} - \hat{\mathbf{x}}_v\|_2^2 \leq \|\mathbf{z} - \hat{\mathbf{x}}_{v'}\|_2^2,\ \forall v' \neq v \Big\},\ \forall v \in \mathcal{V}, \tag{139}$$

where $\mathbf{z} \in \mathbb{R}^N$ is the MMSE estimate of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$ (i.e., $\mathbf{Z}$ is the MMSE estimator of $\mathbf{X}$ given $\mathbf{Y}$), defined as [99, 235, 281] and [98, Sect. 11.5]

$$\begin{aligned} \mathbf{z} &\triangleq \mathbb{E}[\mathbf{X}|\mathbf{Y} = \mathbf{y}] \\ &= \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s|\mathbf{y}) \mathbb{E}[\mathbf{X}|\mathbf{Y} = \mathbf{y}, \mathbf{B} = \mathbf{b}_s] \\ &= \textstyle\sum_{s=1}^{|\mathcal{B}|} \dfrac{p(\mathbf{b}_s) f(\mathbf{y}|\mathbf{b}_s)}{\sum_{s'=1}^{|\mathcal{B}|} p(\mathbf{b}_{s'}) f(\mathbf{y}|\mathbf{b}_{s'})} \underline{\mathbf{z}}_s \end{aligned} \tag{140}$$

where the conditional PDF $f(\mathbf{y}|\mathbf{b}_s)$ is Gaussian as $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{Y}_s})$ (see Definition 5), and $\underline{\mathbf{z}}_s \triangleq \mathbb{E}[\mathbf{X}|\mathbf{Y} = \mathbf{y}, \mathbf{B} = \mathbf{b}_s]$ is the MMSE estimate of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$ and $\mathbf{B} = \mathbf{b}_s$, which, according to Definition 6, comprises of vectors $\mathbf{z}_s = \mathbf{F}_s \mathbf{y}_s \in \mathbb{R}^K$ and $\mathbf{0}_{N-K}$. Similarly, the optimal codevectors for fixed encoder regions satisfy a generalized centroid condition

$$\hat{\mathbf{x}}_v^* = \frac{1}{p(v)} \int_{\mathbf{y} \in \mathcal{S}_v} \mathbb{E}[\mathbf{X}|\mathbf{Y} = \mathbf{y}] f(\mathbf{y}) \mathrm{d}\mathbf{y},\ \forall v \in \mathcal{V}. \tag{141}$$

The VQ can be trained offline via the iterative Lloyd algorithm [107, 192, 188] by successively applying the necessary optimality conditions (139) and (141) for training data sets; see Section 6 for training principles.

Practical Symbol-by-Symbol QCS Methods

Recall that approaching $R_{\mathbf{X}}^{\mathrm{rem}}(D)$ in (113a) requires encoding (large) blocks of vectors, which is infeasible in practice. To this end, this section presents the design of practical QCS methods relying on symbol-by-symbol quantization under three different compression strategies: 1) compress-and-estimate, 2) estimate-and-compress, and 3) support-estimation-and-compress strategies. The schemes are depicted in Fig. 32. The methods use different quantization schemes, which are introduced in Section 7. The details of each method are presented in Sections 7 – 7.

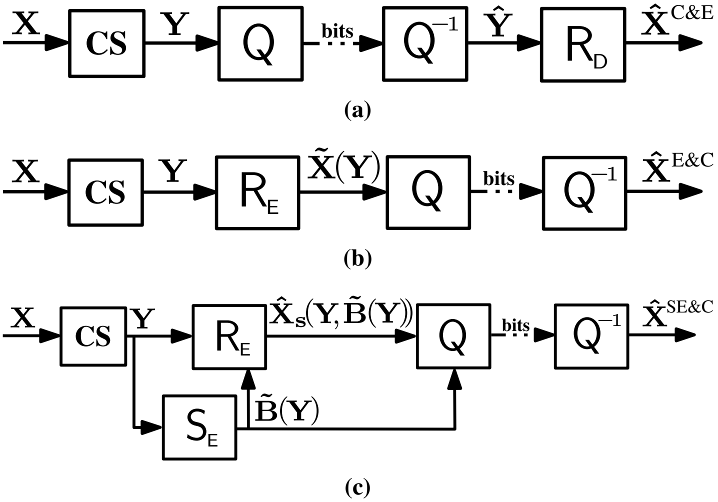


Figure 32: A single-source QCS setups relying on (a) compress-and-estimate, (b) estimate-and-compress, and (c) support-estimation-and-compress strategies.

Quantization Types

Let $\mathbf{U}$ be a length-L random vector with PDF $f(\mathbf{u})$. $\mathbf{U}$ is the input of a quantizer Q, determined by 1) encoder regions $\mathcal{S}_i$, $i \in \mathcal{I} = \{1, \ldots, |\mathcal{I}|\}$, which partition the input as $\mathcal{S}_i \subseteq \mathbb{R}^L$, $\mathcal{S}_i \cap \mathcal{S}_{i'} = \emptyset$, for any $i \neq i'$, and $\bigcup_{i=1}^{|\mathcal{I}|} \mathcal{S}_i = \mathbb{R}^L$, and 2) a reconstruction codebook $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{|\mathcal{I}|}\}$ with codevectors $\mathbf{c}_i \in \mathbb{R}^L$. For a realization $\mathbf{u}$, quantizer encoder $\mathsf{Q} : \mathbb{R}^L \to \mathcal{I}$ produces an index $i = \mathsf{Q}(\mathbf{u})$ if $\mathbf{u} \in S_i$. For a received index i, quantizer decoder $\mathsf{Q}^{-1} : \mathcal{I} \to \mathcal{C}$ produces an estimate of $\mathbf{u}$ as $\hat{\mathbf{u}} = \mathsf{Q}^{-1}(i) = \mathbf{c}_i \in \mathcal{C}$.

Uniform Scalar Quantization

A uniform scalar quantizer (USQ) Q^{u} of rate R' bits/U consists of fixed-length intervals as presented in Algorithm 11. The online compression phase is presented in Algorithm 12. Saturation effects are omitted herein.

Lloyd-Max Quantization

A fixed-rate *Lloyd-Max quantizer* [211, 192] Q^{lm} of rate R' bits/$\mathbf{U}$ optimizes the $|\mathcal{I}| = 2^{R'}$ quantization regions and the codebook to minimize the MSE

Algorithm 11 Uniform scalar quantization (offline).

```
Input: a) Quantization rate $R'$; b) input range parameters $q^{\mathrm{hi}} \geq q^{\mathrm{lo}}$.
Output: Reconstruction codebook $\mathcal{C}^{\mathrm{u}} = \{c_1^{\mathrm{u}}, \dots, c_{2^{R'}}^{\mathrm{u}}\}$ generated as $c_1^{\mathrm{u}} = q^{\mathrm{lo}}$; $c_i^{\mathrm{u}} = c_{i-1}^{\mathrm{u}} + (q^{\mathrm{hi}} - q^{\mathrm{lo}})/(2^{R'} - 1)$, for all $2 \leq i \leq 2^{R'} - 1$; and $c_{2^{R'}}^{\mathrm{u}} = q^{\mathrm{hi}}$.
```

Algorithm 12 Fixed-rate quantization of an input realization **u** (online).

```
Input: Reconstruction codebook $\mathcal{C}^{\mathrm{fix}}$ ($\mathcal{C}^{\mathrm{lm}}$ or $\mathcal{C}^{\mathrm{u}}$).
Encoding: 1) Find the encoding index $i^* = \mathrm{argmin}_{i \in \mathcal{I}} \|\mathbf{u} - \mathbf{c}_i^{\mathrm{fix}}\|_2^2$; 2) communicate the $R'$-bit representation of $i^*$ to the decoder.
Decoding: Given the received index $i^*$, obtain an estimate of $\mathbf{u}$ as $\hat{\mathbf{u}} = \mathbf{c}_{i^*}^{\mathrm{fix}} \in \mathcal{C}^{\mathrm{fix}}$.
```

distortion

$$D' = \sum_{i=1}^{2^{R'}} p(i)\mathbb{E}[\|\mathbf{U} - \mathbf{c}_i^{\mathrm{lm}}\|_2^2 | I = i], \tag{142}$$

where I represents the quantization index, and $p(i) \triangleq \Pr(I = i)$. Q$^{\mathrm{lm}}$ can be trained by the iterative Lloyd and LBG algorithms [211, 192, 188], as presented in Algorithm 13. The online compression phase is presented in Algorithm 12.

Variable-Rate Quantization

A *variable-rate quantizer* Q$^{\mathrm{vr}}$ minimizes a distortion-rate cost function (cf. (94))

$$(1 - \mu')D' - \mu' \sum_{i \in \mathcal{I}} p(i)\log(p(i)) \tag{145}$$

where D' is given in (142), and $\mu' \in [0, 1]$ is a weighting parameter. Following entropy-constrained scalar/vector quantization (ECSQ/ECVQ) [306, 66], Q$^{\mathrm{vr}}$ can be trained as presented in Algorithm 14 (see also Algorithm 9 in Section 6). The online phase is presented in Algorithm 15. The source codebook $\mathcal{H} \triangleq \{\mathbf{h}_1, \dots, \mathbf{h}_{|\mathcal{H}|}\}$,contains in general variable length binary codewords and it can be generated by, e.g., the Huffman coding [149].

Algorithm 13 Training of a Lloyd-Max quantizer (offline).

```
Input: a) Sequence $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots\}$ sampled from $f(\mathbf{u})$; b) quantization rate $R'$.
Initialization: Reconstruction codebook $\mathcal{C}^{\mathrm{lm}}$.
Repeat until convergence
    1) For given codebook, find the optimal regions by classifying $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots\}$ as
        $\mathcal{S}_i^{\mathrm{lm}} = \Big\{\mathbf{u} : \|\mathbf{u} - \mathbf{c}_i^{\mathrm{lm}}\|_2^2 \leq \|\mathbf{u} - \mathbf{c}_{i'}^{\mathrm{lm}}\|_2^2,\ \forall i' \neq i\Big\},\ i \in \mathcal{I}. \quad (143)$
    2) For given regions, find the optimal codevectors as the conditional expectations
        $\mathbf{c}_i^{\mathrm{lm}} = \mathbb{E}[\mathbf{U}|I = i] = \frac{1}{p(i)} \int_{\mathbf{u} \in \mathcal{S}_i^{\mathrm{lm}}} \mathbf{u} f(\mathbf{u}) \mathrm{d}\mathbf{u},\ i \in \mathcal{I}. \quad (144)$
Output: Reconstruction codebook $\mathcal{C}^{\mathrm{lm}} = \{\mathbf{c}_1^{\mathrm{lm}}, \ldots, \mathbf{c}_{2^{R'}}^{\mathrm{lm}}\}$.
```

Algorithm 14 Training of an entropy-constrained quantizer (offline).

```
Input: a) Sequence $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots\}$ sampled from $f(\mathbf{u})$; b) quantization levels $2^{R'}$; c) weight parameter $\mu' \in [0, 1]$.
Initialization: i) Reconstruction codebook $\mathcal{C}^{\mathrm{vr}}$, and ii) index probabilities $p(i) = 1/2^{R'}$, $i \in \mathcal{I}$.
Repeat until convergence
    1) For given codebook and rate measures, find the optimal regions by classifying $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots\}$ as
        $\mathcal{S}_i^{\mathrm{vr}} = \Big\{\mathbf{u} :\ (1-\mu')\|\mathbf{u} - \mathbf{c}_i^{\mathrm{vr}}\|_2^2 - \mu'\log(p(i)) \leq$
            $(1-\mu')\|\mathbf{u} - \mathbf{c}_i^{\mathrm{vr}}\|_2^2 - \mu'\log(p(i')),\ \forall i' \neq i\Big\},\ i \in \mathcal{I}. \quad (146)$
    2) Update the rate measures $-\log(p(i))$, $i \in \mathcal{I}$, given the new regions.
    3) For given regions, find the optimal codevectors $\mathbf{c}_i^{\mathrm{vr}}$, $i \in \mathcal{I}$, equivalently as in (144).
Output: a) Reconstruction codebook $\mathcal{C}^{\mathrm{vr}} = \{\mathbf{c}_1^{\mathrm{vr}}, \ldots, \mathbf{c}_{2^{R'}}^{\mathrm{vr}}\}$; b) index probabilities $p(i)$, $i \in \mathcal{I}$; c) source codebook $\mathcal{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_{|\mathcal{H}|}\}$ generated using $p(i)$, $i \in \mathcal{I}$.
```

Algorithm 15 Variable-rate quantization of an input realization $\mathbf{u}$ (online).

```
Input: a) Reconstruction codebook $\mathcal{C}^{\mathrm{vr}}$; b) index probabilities $p(i)$, $i \in \mathcal{I}$; c) weight parameter $\mu' \in [0,1]$; d) source codebook $\mathcal{H}$.
Encoding: 1) Find the encoding index $i^* = \mathrm{argmin}_{i\in\mathcal{I}}\,(1-\mu')\|\mathbf{u}-\mathbf{c}_i^{\mathrm{vr}}\|_2^2 - \mu'\log(p(i))$; 2) communicate the binary codeword $\mathbf{h}_{i^*}$ to the decoder.
Decoding: Given the received codeword $\mathbf{h}_{i^*}$, find the corresponding index $i^*$ and obtain an estimate of $\mathbf{u}$ as $\hat{\mathbf{u}} = \mathbf{c}_{i^*}^{\mathrm{vr}} \in \mathcal{C}^{\mathrm{vr}}$.
```

Compress-And-Estimate QCS Methods

A *compress-and-estimate* (C&E) scheme depicted in Fig. 32(a) consists of two stages: 1) a compression stage where the encoder quantizes $\mathbf{Y}$ under an MSE distortion criterion that depends only on $\mathbf{Y}$ (not on $\mathbf{X}$), and 2) an estimation stage where the decoder estimates $\mathbf{X}$ from the decoded quantized measurements $\hat{\mathbf{Y}} = \mathsf{Q}^{-1}(\mathsf{Q}(\mathbf{Y}))$. Let $\hat{\mathbf{X}}^{\mathrm{C\&E}}$ denote a length-N reproduction random vector at the decoder output. The end-to-end MSE distortion can be written as

$$D^{\mathrm{C\&E}} = \mathbb{E}[\|\mathbf{X} - \hat{\mathbf{X}}^{\mathrm{C\&E}}\|_2^2] = \mathbb{E}[\|\mathbf{X} - \mathsf{R}_{\mathsf{D}}\left[\mathsf{Q}^{-1}(\mathsf{Q}(\mathbf{Y}))\right]\|_2^2], \tag{147}$$

where $\mathsf{R}_{\mathsf{D}} : \hat{\mathbf{Y}} \to \hat{\mathbf{X}}^{\mathrm{C\&E}}$ is a signal reconstruction algorithm at D. A C&E principle underlies many early QCS algorithms in, e.g., [151, 78]. An information-theoretic study of QCS under C&E scheme is addressed in [164]. Four C&E algorithms with 1) USQ, 2) Lloyd-Max SQ (referred to as "SQ" in the QCS method names henceforth), 3) ECSQ, and 4) VQ are considered. The details of each method are summarized in Table 4. The table lists the used offline training algorithms with the key parameters, i.e., the quantizer input vector $\mathbf{U}$ of length L, and the total average rate $R_{\mathrm{tot}} = NR$ bits/$\mathbf{X}$, where the rate R is defined as the bits/entry of $\mathbf{X}$.

To summarize the main operations of SQ based methods 1) – 3), the decoder receives M indices, forms $\hat{\mathbf{y}} = [c_{1,i_1^*}^{\mathrm{C\&E}} \cdots c_{M,i_M^*}^{\mathrm{C\&E}}]^{\mathrm{T}}$, and estimates the source as $\hat{\mathbf{x}} = \mathsf{R}_{\mathsf{D}}(\hat{\mathbf{y}})$. In particular, the MSE-optimal R_{D} for C&E$_{\mathrm{vq}}$ can be defined as follows. For C&E$_{\mathrm{vq}}$, the MSE-optimal output vectors of $\mathsf{R}_{\mathsf{D}}(\cdot)$ that minimize the distortion in (147) are given as

$$\hat{\mathbf{x}}(\mathbf{c}_i^{\mathrm{C\&E_{vq}}}) = \mathbb{E}[\mathbf{X}|I = i] = \frac{1}{p(i)} \int_{\mathbf{y}\in\mathcal{S}_i^{\mathrm{C\&E_{vq}}}} \mathbf{z} f(\mathbf{y}) \mathrm{d}\mathbf{y}, \ i \in \mathcal{I}, \tag{148}$$

where $\mathbf{z} \in \mathbb{R}^N$ is the MMSE estimate of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$ (see (140)).

Estimate-and-Compress QCS Methods

An *estimate-and-compress* (E&C) scheme depicted in Fig. 32(b) consists of two stages: 1) an estimation stage of $\mathbf{X}$ from $\mathbf{Y}$ at the encoder, and 2) a compression stage to quantize the resulting signal estimate under an MSE distortion criterion. Let $\hat{\mathbf{X}}^{\mathrm{E\&C}}$ denote a length-N reproduction random vector at the decoder output. The end-to-end MSE distortion reads as

$$D^{\mathrm{E\&C}} = \mathbb{E}[\|\mathbf{X} - \hat{\mathbf{X}}^{\mathrm{E\&C}}\|_2^2] = \mathbb{E}[\|\mathbf{X} - \mathsf{Q}^{-1}\left[\mathsf{Q}(\mathsf{R}_{\mathrm{E}}(\mathbf{Y}))\right]\|_2^2], \tag{149}$$

where $\mathsf{R}_{\mathrm{E}} : \mathbf{Y} \rightarrow \tilde{\mathbf{X}}(\mathbf{Y})$ is a reconstruction algorithm at E, and $\tilde{\mathbf{X}}(\mathbf{Y})$ denotes the estimator of $\mathbf{X}$ from $\mathbf{Y}$. Recall that an E&C scheme adheres to the optimal encoding structure for remote source coding [304]. E&C based QCS algorithms have been devised in, e.g., [265, 266, 177, 179, 180, 182, 181] and [184, Sect. V]. The related information-theoretic studies include [165, 183, 184].

Five E&C algorithms with 1) USQ, 2) SQ, 3) ECSQ, 4) VQ, and 5) ECVQ are considered with the details listed in Table 4. For SQ based methods 1) – 3), the decoder uses the N received indices to estimate the source as $\hat{\mathbf{x}} = [c^{\mathrm{E\&C}}_{1,i_1^*} \cdots c^{\mathrm{E\&C}}_{N,i_N^*}]^{\mathrm{T}}$; for VQ based methods 4) – 5), the decoder operates as $\hat{\mathbf{x}} = \mathbf{c}^{\mathrm{E\&C}}_{i^*}$.

Support-Estimation-and-Compress QCS Methods

A *support-estimation-and-compress* (SE&C) scheme depicted in Fig. 32(c) consists of three stages: 1) a support estimation stage where the encoder estimates $\mathbf{B}$ from $\mathbf{Y}$, 2) an estimation stage of $\mathbf{X}$ given $\mathbf{Y}$ and the support estimator at E, and 3) a two-phase compression stage of the resulting source estimate by a) lossless compression of the support, and b) lossy compression of the non-zero part. Let $\hat{\mathbf{X}}^{\mathrm{SE\&C}}$ denote a length-N reproduction random vector at the decoder output. The end-to-end MSE distortion can be expressed as

$$\begin{aligned} D^{\mathrm{SE\&C}} &= \mathbb{E}[\|\mathbf{X} - \hat{\mathbf{X}}^{\mathrm{SE\&C}}\|_2^2] \\ &= \mathbb{E}[\|\mathbf{X} - \mathsf{Q}^{-1}[\mathsf{Q}\left(\mathsf{S}_{\mathrm{E}}(\mathbf{Y}) + \mathsf{R}_{\mathrm{E}}(\mathbf{Y}|\mathsf{S}_{\mathrm{E}}(\mathbf{Y}))\right)]\|_2^2], \end{aligned} \tag{150}$$

Table 4: QCS methods relying on different compression strategies

Class	Method	Training/ Online	$\mathbf{U}$	L	$R_{\mathrm{tot}} = NR$
Compress-and-Estimate	$\mathrm{C\&E}_{\mathrm{usq}}$	Alg. 11 / 12	Y_m	1	MR'
	$\mathrm{C\&E}_{\mathrm{sq}}$	Alg. 13 / 12	Y_m	1	MR'
	$\mathrm{C\&E}_{\mathrm{ecsq}}$	Alg. 14 / 15	Y_m	1	$\sum_{m=1}^{M} \gamma(\mathcal{H}_m^{\mathrm{C\&E}})$
	$\mathrm{C\&E}_{\mathrm{vq}}$	Alg. 13 / 12	$\mathbf{Y}$	M	R'
Estimate-and-Compress	$\mathrm{E\&C}_{\mathrm{usq}}$	Alg. 11 / 12	$[\tilde{\mathbf{X}}(\mathbf{Y})]_n$	1	NR'
	$\mathrm{E\&C}_{\mathrm{sq}}$	Alg. 13 / 12	$[\tilde{\mathbf{X}}(\mathbf{Y})]_n$	1	NR'
	$\mathrm{E\&C}_{\mathrm{ecsq}}$	Alg. 14 / 15	$[\tilde{\mathbf{X}}(\mathbf{Y})]_n$	1	$\sum_{n=1}^{N} \gamma(\mathcal{H}_n^{\mathrm{E\&C}})$
	$\mathrm{E\&C}_{\mathrm{vq}}$	Alg. 13 / 12	$\tilde{\mathbf{X}}(\mathbf{Y})$	N	R'
	$\mathrm{E\&C}_{\mathrm{ecvq}}$	Alg. 14 / 15	$\tilde{\mathbf{X}}(\mathbf{Y})$	N	$\gamma(\mathcal{H}^{\mathrm{E\&C}})$
Support-Estimation-and-Compress	$\mathrm{SE\&C}_{\mathrm{usq}}$	Alg. 11 / 12	$[\mathbf{Z}_s(\mathbf{Y}, \tilde{\mathbf{B}}(\mathbf{Y}))]_n$	1	$\gamma(\mathcal{H}_{\mathrm{supp}}^{\mathrm{SE\&C}}) + KR'$
	$\mathrm{SE\&C}_{\mathrm{sq}}$	Alg. 13 / 12	$[\mathbf{Z}_s(\mathbf{Y}, \tilde{\mathbf{B}}(\mathbf{Y}))]_n$	1	$\gamma(\mathcal{H}_{\mathrm{supp}}^{\mathrm{SE\&C}}) + KR'$
	$\mathrm{SE\&C}_{\mathrm{ecsq}}$	Alg. 14 / 15	$[\mathbf{Z}_s(\mathbf{Y}, \tilde{\mathbf{B}}(\mathbf{Y}))]_n$	1	$\gamma(\mathcal{H}_{\mathrm{supp}}^{\mathrm{SE\&C}}) + \sum_{k=1}^{K} \gamma(\mathcal{H}_k^{\mathrm{SE\&C}})$
	$\mathrm{SE\&C}_{\mathrm{vq}}$	Alg. 13 / 12	$\mathbf{Z}_s(\mathbf{Y}, \tilde{\mathbf{B}}(\mathbf{Y}))$	K	$\gamma(\mathcal{H}_{\mathrm{supp}}^{\mathrm{SE\&C}}) + R'$

where $\mathsf{S}_{\mathrm{E}} : \mathbf{Y} \rightarrow \tilde{\mathbf{B}}(\mathbf{Y})$ is a support estimation algorithm at E, and $\tilde{\mathbf{B}}(\mathbf{Y})$ denotes the estimator of $\mathbf{B}$ from $\mathbf{Y}$. The sparsity K is assumed to be known by $\mathsf{S}_{\mathrm{E}}(\cdot)$. QCS schemes akin to SE&C have been considered in, e.g., [129, 265].

The optimal $\mathsf{R}_{\mathrm{E}}(\cdot)$ in (150) is the one that minimizes the MSE distortion $\mathbb{E}[\|\mathbf{X}_s - \mathsf{R}_{\mathrm{E}}(\cdot)\|_2^2]$, where $\mathbf{X}_s$ is the sth subsource in Definition 4. It can be shown that $\mathsf{R}_{\mathrm{E}}(\mathbf{Y}|\mathsf{S}_{\mathrm{E}}(\mathbf{Y})) = \mathbf{Z}_s(\mathbf{Y}, \tilde{\mathbf{B}}(\mathbf{Y}))$, where $\mathbf{Z}_s(\mathbf{Y}, \tilde{\mathbf{B}}(\mathbf{Y}))$ is the MMSE estimator of $\mathbf{X}_s$ given $\mathbf{Y}$ and $\tilde{\mathbf{B}}(\mathbf{Y})$ so that (cf. (124))

$$\begin{aligned} \mathbf{z}_s(\mathbf{y}, \tilde{\mathbf{b}}_s) &\triangleq \mathbb{E}[\mathbf{X}_s | \mathbf{Y} = \mathbf{y}, \tilde{\mathbf{B}}(\mathbf{Y}) = \tilde{\mathbf{b}}_s] \\ &= \mathbf{\Sigma}_{\mathbf{X}_s} \mathbf{\Phi}_s^{\mathrm{T}} \left(\mathbf{\Phi}_s \mathbf{\Sigma}_{\mathbf{X}_s} \mathbf{\Phi}_s^{\mathrm{T}} + \mathbf{\Sigma}_{\mathbf{W}} \right)^{-1} \mathbf{y}. \end{aligned} \tag{151}$$

Four SE&C algorithms with 1) USQ, 2) SQ, 3) ECSQ, and 4) VQ are considered; see the details in Table 4. Quantizers operate equivalently as in the corresponding E&C schemes. The support estimator $\tilde{\mathbf{B}}(\mathbf{Y})$ is communicated losslessly using a binary source codebook $\mathcal{H}_{\mathrm{supp}}^{\mathrm{SE\&C}}$, which can be generated by, e.g., the Huffman coding.

Numerical Results

Numerical results are presented to illustrate the RD behavior of lossy CS, assess the tightness of the lower bound, and compare the performance of the QCS methods of Section 7 against the derived limits.

Simulation Setup

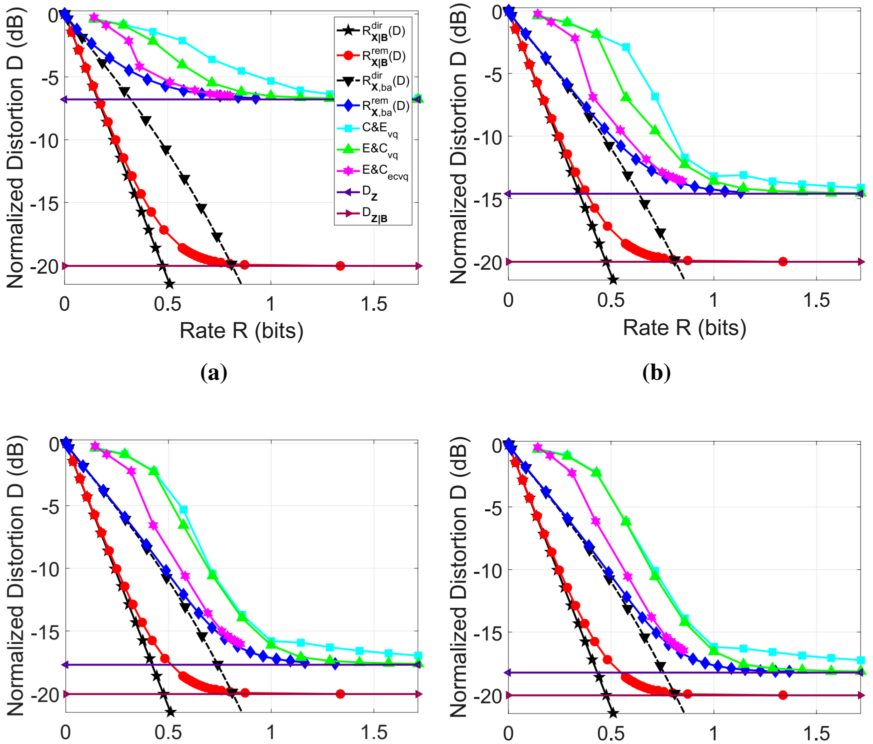
Consider setups with $\mathbf{\Sigma_G} = \sigma_G^2 \mathbf{I}_N$ with $\sigma_G^2 = 1$, and $\mathbf{\Sigma_W} = \sigma_W^2 \mathbf{I}_M$ with $\sigma_W^2 = 0.01$. The following curves are evaluated:
1) $R^{\text{rem}}_{\mathbf{X}|\mathbf{B}}(D)$: the conditional remote RDF of Theorem 1.
2) $R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$: a numerically approximated remote RDF of Algorithm 10.
3) $R^{\text{dir}}_{\mathbf{X}|\mathbf{B}}(D)$: *the conditional direct RDF of* $\mathbf{X}$, corresponding to the lossy compression of $\mathbf{X}$ with $\mathbf{B}$ available as the SI at the encoder and decoder, derived in Appendix F (see [293, Sect. VII-A]). Note that $R^{\text{dir}}_{\mathbf{X}|\mathbf{B}}(D) \leq R^{\text{rem}}_{\mathbf{X}|\mathbf{B}}(D)$.
4) $R^{\text{dir}}_{\mathbf{X},\text{ba}}(D)$: a numerically approximated *direct RDF of* $\mathbf{X}$ which represents lossy compression of $\mathbf{X}$ (with no support SI), and is obtained by applying the discretization of Section 7 and Algorithm 10 using $\mathbf{Y} = \mathbf{X}$. Note that $R^{\text{dir}}_{\mathbf{X}|\mathbf{B}}(D) \leq R^{\text{dir}}_{\mathbf{X},\text{ba}}(D) \leq R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$.
5) $D_{\mathbf{Z}|\mathbf{B}}$: the average MMSE estimation error in (133) (known $\mathbf{B}$).
6) $D_{\mathbf{Z}}$: numerically evaluated average MMSE estimation error of $\mathbf{X}$ given $\mathbf{Y}$, i.e., $D_{\mathbf{Z}} \triangleq N^{-1}\mathbb{E}[\|\mathbf{X} - \mathbf{Z}\|_2^2]$; estimator $\mathbf{Z} \triangleq \mathbb{E}[\mathbf{X}|\mathbf{Y}]$ takes values according to (140) (unknown $\mathbf{B}$).
7) Various QCS methods of Section 7 which will be specified for each simulation case later.
The measurement matrix $\mathbf{\Phi}$ is generated by taking the first M rows of an $N \times N$ discrete cosine transform matrix, and normalizing the columns as $\|\cdot\|_2 = 1$. ECSQs/ECVQs are run with $\mu' = 0.1/\log(|\mathcal{I}|)$. All binary source codebooks $\mathcal{H}$ are generated via the Huffman coding. The distortion is measured as $10\log_{10}(\mathbb{E}[d(\mathbf{X}, \mathbf{X}^{\text{est}})]/N^{-1}\mathbb{E}[\|\mathbf{X}\|_2^2])$ dB, where $\mathbf{X}^{\text{est}}$ is the method-dependent decoded estimate of $\mathbf{X}$, and $\mathbb{E}[\|\mathbf{X}\|_2^2] = \sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s)\text{Tr}(\mathbf{\Sigma}_{\mathbf{X}_s})$. The rate is measured as R bits/entry of $\mathbf{X}$. The convex minimization problems are solved via CVX [130].
Complexity: As the complexities of Algorithm 10 and the QCS methods in Section 7 increase exponentially with the number of quantization levels and $|\mathcal{B}|$, the experiments are confined to moderate signal dimensions and

quantization rates. It is worth remarking that in respect to any discretized BA algorithm, there is similar complexity issue due to a large number of variables, regardless of the quantization method. In fact, due to the VQ advantages [194], the proposed algorithm enjoys a superior trade-off between the approximation accuracy and the complexity compared to SQ. The complexities of the VQ based QCS methods can be reduced by low-complexity VQ variants such as tree-structured, multi-step, and lattice VQs [133]. Since $\mathbf{Y}$ has a Gaussian mixture density, computationally efficient VQs designed in [272] are also potential candidates.

Rate-Distortion Behavior of Lossy CS

Consider a setup with $N = 7$, $M = 5$, $K = 1$, and equal support probabilities $p(\mathbf{b}_s) = 1/|\mathcal{B}|$, $\forall s = 1, \ldots, |\mathcal{B}|$. Fig. 33(d) depicts the average distortion versus the average rate for different compression schemes. The quantization algorithms are run with $R' = \{2^1, \ldots, 2^{12}\}$, and the BA Algorithm 10 with $\log|\mathcal{V}| = 12$. Only few VQ based QCS methods of Section 7 are reported here; experiments for all practical QCS methods are presented in Section 7. In order to obtain high performance, the MMSE estimator in (140) – which is known to have exponential complexity [99] – is used as $\mathsf{R}_{\mathrm{E}}(\cdot)$ for E&C$_{\mathrm{vq}}$ and E&C$_{\mathrm{ecvq}}$. Lower complexity alternatives are investigated in Section 7.

Let us first investigate the SI aided lower bounds $R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D)$, $R^{\mathrm{dir}}_{\mathbf{X},\mathrm{ba}}(D)$, and $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ to the remote RDF $R^{\mathrm{rem}}_{\mathbf{X}}(D)$ in (113a). Owing to the direct observations with support SI, $R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D)$ appears as the line (in log scale) with slope $-6\frac{N}{K} = -42$ dB/bit [293] and yields the lowest R for all values of D, as expected. The substantially increased rate for $R^{\mathrm{dir}}_{\mathbf{X},\mathrm{ba}}(D)$ compared to $R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D)$ is caused by the necessity of conveying the support of $\mathbf{X}$ to the decoder. While $R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D)$ and $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ nearly coincide at high distortion, the curves diverge for moderate to low distortion values. The gradually increasing gap between $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ and $R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D)$ for low values of D is a consequence of the remote sensing. Note that whereas an arbitrarily small distortion is achievable at asymptotically high rates for $R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D)$ and $R^{\mathrm{dir}}_{\mathbf{X},\mathrm{ba}}(D)$ (i.e., $\lim_{D\to 0} R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D) = \infty$ and $\lim_{D\to 0} R^{\mathrm{dir}}_{\mathbf{X},\mathrm{ba}}(D) = \infty$), the lowest achievable distortion for $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ is the MMSE estimation error floor $D_{\mathbf{Z}|\mathbf{B}}$ (i.e., $\lim_{D\to D_{\mathbf{Z}|\mathbf{B}}} R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D) = \infty$).



(a)

(b)

(c)

(d)

Figure 33: Rate-distortion performance of lossy CS schemes with equal support probabilities for $N = 7$, $K = 1$, and the number of measurements (a) $M = 2$, (b) $M = 3$, (c) $M = 4$, and (d) $M = 5$. The colors and markers of the curves in (b), (c), and (d) are equivalent to those in (a).

Next, focus on the approximate remote RDF $R^{\mathrm{rem}}_{\mathbf{X},\mathrm{ba}}(D)$, i.e., the best achievable performance of any QCS method. The gap between $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ and $R^{\mathrm{rem}}_{\mathbf{X},\mathrm{ba}}(D)$ represents the compression loss induced by the random measurements taken without knowing the sparse support in a QCS setup [129]. The tightness of the lower bound is heavily influenced by the signal setup parameters, as will be exemplified in the subsequent experiments. Despite the gap, the lower bound $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ captures the main peculiarities of the lossy CS: the curve has an almost linear distortion region at low rates, whereas for

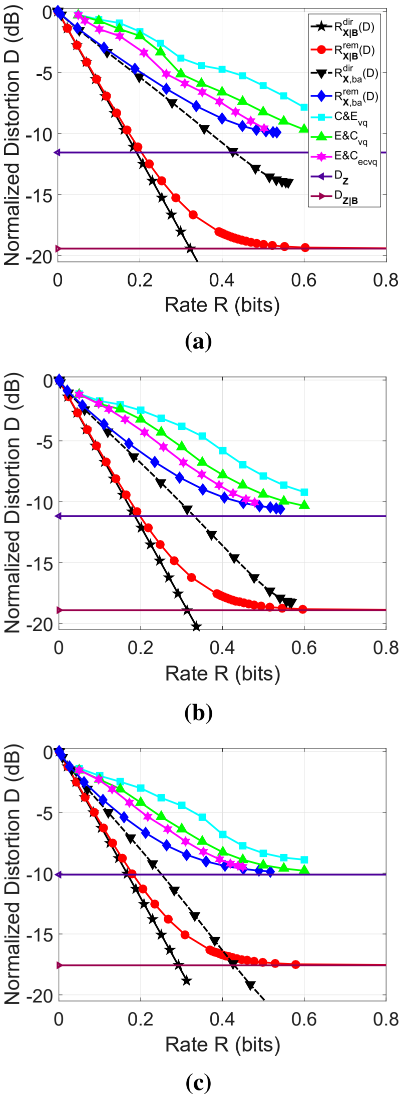
high rates, the distortion saturates to the MMSE estimation error floor $D_{\mathbf{Z}}$. As a special remark, the slope of $R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$ at low rates is steeper than the conventional -6 dB/bit due to the sparsity. Note that for a small R, the rate is the most limiting factor to achievable distortion, and, thus, $R^{\text{dir}}_{\mathbf{X},\text{ba}}(D)$ nearly coincides with $R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$; for higher rates, the impact of noisy compressive measurements increases, thereby degrading the performance of $R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$. Regarding the approximation accuracy of $R^{\text{dir}}_{\mathbf{X},\text{ba}}(D)$ and $R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$, observe that the highest obtained rate is $R \approx 1.3$ bits, so the "over-sampling ratio" of the VQ discretization is at least $|\mathcal{V}|/2^{NR} \approx 7.5$.
Because the encoder of $\text{C\&E}_{\text{vq}}$ is incognizant of CS, its performance is the worst amongst the QCS methods. The advantages of entropy coding are shown by the $\text{E\&C}_{\text{ecvq}}$ curves which, for moderate rates, approach the compression limit $R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$. As a proof of validity, the $\text{E\&C}_{\text{vq}}$ method eventually saturates to $D_{\mathbf{Z}}$, which is expected to also happen for the other QCS methods at sufficiently high rates.

Effect of the Number of Measurements

For the setup of Section 7, Figs. 33(a) – (d) illustrate the influence of different numbers of measurements $M = \{2, 3, 4, 5\}$ on the compression performance. As M increases, i.e., the signal-to-noise ratio increases, the level of $D_{\mathbf{Z}}$ decreases, and the performance of each method that has no support SI moves closer to the lower bound $R^{\text{rem}}_{\mathbf{X}|\mathbf{B}}(D)$. The largest gain is achieved when M is increased from $M = 2$ to $M = 3$, whereas the difference between $M = 4$ and $M = 5$ is almost negligible. This matches the CS philosophy: increasing M beyond the value that suffices for accurate CS signal recovery does not bring significant gains. In this respect, provided that M is already at this satisfactory level, it pays off to primarily invest in rate R to meet the given distortion fidelity D. Note that the convergence of the curves to their respective distortion floors is rather similar for all M, and that $R^{\text{dir}}_{\mathbf{X}|\mathbf{B}}(D)$, $R^{\text{dir}}_{\mathbf{X},\text{ba}}(D)$, $R^{\text{rem}}_{\mathbf{X}|\mathbf{B}}(D)$, and $D_{\mathbf{Z}|\mathbf{B}}$ remain unaltered.

Effect of Support Probabilities

Consider a setup with $N = 20$, $M = 8$, $K = 2$, and unequal support probabilities as $p(\mathbf{b}_s) = \alpha^{s}_{\text{pl}}/\sum_{s'=1}^{|\mathcal{B}|} \alpha^{s'}_{\text{pl}}$, $s = 1, \ldots, |\mathcal{B}|$, where $0 < \alpha_{\text{pl}} \leq 1$ is a



(a)

(b)

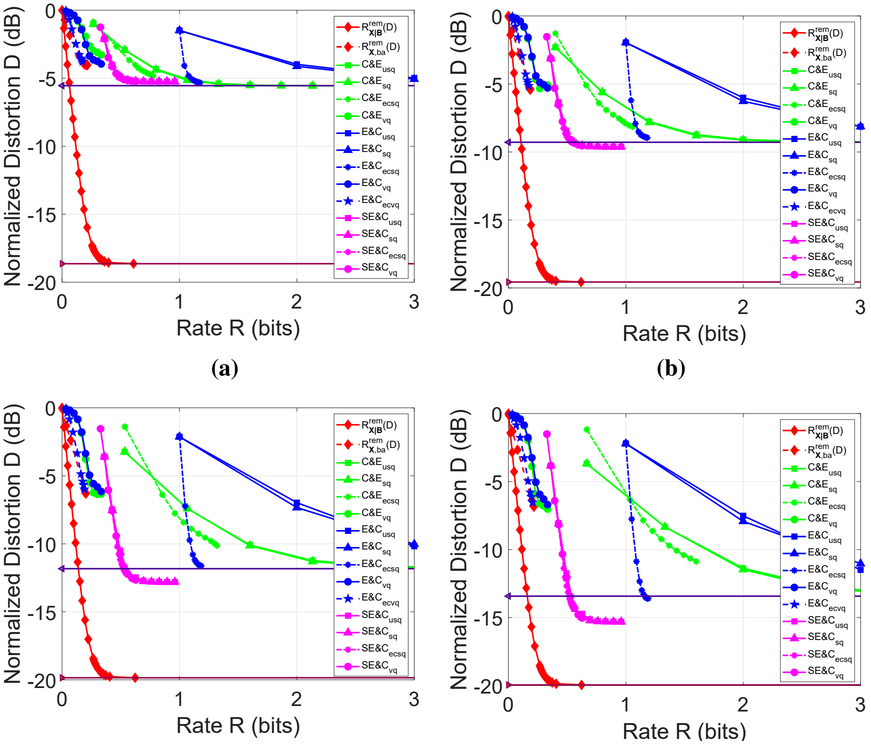
(c)

Figure 34: Rate-distortion performance of lossy CS schemes for $N = 20$, $M = 8$, $K = 2$, and power law type support probabilities with parameters (a) $\alpha_{\text{pl}} = 0.98$, (b) $\alpha_{\text{pl}} = 0.90$, and (c) $\alpha_{\text{pl}} = 0.72$. The colors and markers of the curves in (b) and (c) are equivalent to those in (a).

parameter that adjusts the concentration of the probability mass function of $\mathbf{B}$, and $1 > p(\mathbf{b}_1) \geq \cdots \geq p(\mathbf{b}_{|\mathcal{B}|}) > 0$. For small values of α_{pl}, the probability mass concentrates around a fraction of elements in $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_{|\mathcal{B}|}\}$, and vice versa. $\alpha_{\text{pl}} = 1$ corresponds to the uniform distribution, whereas $\alpha_{\text{pl}} \to 0$ approaches remote compression of only a single sparse vector. The vectors $\mathbf{b}_s$ in alphabet $\mathcal{B}$ are ordered so that the decimal number of a binary string represented by $\mathbf{b}_{s+1}$ is greater than that of $\mathbf{b}_s$, $s = 1, \ldots, |\mathcal{B}| - 1$ ($|\mathcal{B}| = 190$). Fig. 34 shows the average distortion versus the average rate for $\alpha_{\text{pl}} = \{0.98, 0.90, 0.72\}$. Decreasing α_{pl} reduces the uncertainty of the signal support which improves the compression efficiency. This is seen in the increased decay rate of D for the non-SI schemes, the shift of $R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$ towards $R^{\text{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ and $R^{\text{dir}}_{\mathbf{X},\text{ba}}(D)$ towards $R^{\text{dir}}_{\mathbf{X}|\mathbf{B}}(D)$, and the diminution of the gap between $D_{\mathbf{Z}}$ and $D_{\mathbf{Z}|\mathbf{B}}$, which is related to the best possible support recovery for a given setup. This exemplifies that, for a sufficiently concentrated probability mass of $\mathbf{B}$, the E&C$_{\text{ecvq}}$ method efficiently encodes sparse vectors from noisy compressive measurements: its performance approaches the best achievable performance of a support unaware QCS method (i.e., $R^{\text{rem}}_{\mathbf{X},\text{ba}}(D)$). The result illustrates the MSE separation principle (cf. (125)): an efficient QCS method implicitly (successfully) recovers $\mathbf{X}$ from $\mathbf{Y}$ and encodes the resulting estimates optimally.

Performance of QCS Methods

Let us now focus on the practical QCS methods. Consider a setup with $N = 30$, $M = \{8, 12, 16, 20\}$, $K = 2$, and equal support probabilities. Basis pursuit denoising (BPDN) [61] is used as a moderate complexity reconstruction algorithm for R_{D} in C&E, R_{E} in E&C, and S_{E} in SE&C; S_{E} forms the support estimate from the indices of the K largest magnitudes of the BPDN output. In other words, the BPDN approximates the MMSE estimate (140) for R_{E}; another alternative is the randomized OMP [99]. Note that more sophisticated support recovery algorithms S_{E} could be considered as well. For C&E$_{\text{usq}}$ and SE&C$_{\text{usq}}$, q^{lo} and q^{hi} are set as the minimum and maximum codepoints of the corresponding Lloyd-Max SQs. The same procedure is first used for E&C$_{\text{usq}}$, after which each codepoint is added a constant shift so that E&C$_{\text{usq}}$ contains the codepoint of E&C$_{\text{sq}}$ that is closest to zero. The quantization algorithms are run with $R' = \{2^1, \ldots, 2^{10}\}$, and the BA Algorithm 10 with $\log|\mathcal{V}| = 10$.



(a) **(b)** **(c)** **(d)**

Figure 35: Rate-distortion performance of QCS schemes with equal support probabilities for $N = 30$, $K = 2$, and the number of measurements (a) $M = 8$, (b) $M = 12$, (c) $M = 16$, and (d) $M = 20$.

Fig. 35 depicts the average distortion versus the average rate for different QCS schemes for $M = \{8, 12, 16, 20\}$. The performances of the SQ based methods can be ranked as E&C $<$ C&E $<$ SE&C, and the VQ based methods as SE&C $<$ C&E $<$ E&C. Next, we elaborate each of the three classes in more detail.

C&E: $\mathrm{C\&E_{usq}}$ and $\mathrm{C\&E_{sq}}$ perform nearly identically. The entropy coding in $\mathrm{C\&E_{ecsq}}$ can only slightly improve the performance. For large values of M, $\mathrm{C\&E_{vq}}$ slightly outperforms $\mathrm{E\&C_{vq}}$ even though $\mathrm{E\&C_{vq}}$ uses the optimal E&C structure. This is caused by the fact that $\mathrm{R_E}$ in E&C is approximated as the BPDN instead of the MMSE estimator in (140). Recall that while $\mathrm{C\&E_{vq}}$

uses a suboptimal C&E strategy, it uses the MSE-optimal outputs in (148) for a given VQ.

E&C: Among all QCS methods, entropy coding has the most significant impact on $\mathrm{E\&C_{sq}} \rightarrow \mathrm{E\&C_{ecsq}}$, which stems from the fact that $\mathrm{E\&C_{sq}}$ quantizes the zero inputs inefficiently. The best practical QCS method is the $\mathrm{E\&C_{ecvq}}$ method which approaches the compression limit $R^{\mathrm{rem}}_{\mathbf{X},\mathrm{ba}}(D)$.

SE&C: It can be seen that the adaptive compression of the support set and estimated magnitudes is an effective strategy as the SE&C methods outperform all SQ based C&E and E&C methods. Because all SE&C curves nearly coincide, the support recovery performance of S_{E} seems to be the limiting factor in SE&C.

In general, as M increases, the QCS methods perform better and the gap between the analytical MMSE estimation floor with support side information (the lower horizontal line $D_{\mathbf{Z}|\mathbf{B}}$) and the error floor of the BPDN reconstruction (the upper horizontal line) decreases. Note that only C&E and E&C methods relying plainly on the BPDN outputs saturate to the upper floor; because SE&C removes the noise outside the (estimated) support of each BPDN output, it can surpass the floor, as illustrated by Figs. 35(b)–(d).

Conclusions

This section addressed lossy compression of single-sensor CS from the remote source coding perspective. By providing support SI to the encoder and decoder, a conditional remote RDF that establishes a compression lower bound for a finite-rate CS setup was derived. The best such encoder separates into an MMSE estimation step and an optimal transmission step. A modified BA algorithm was developed to numerically approximate the remote RDF, and, thus, to assess the best attainable compression performance of any practical QCS method. The main RD characteristics of the lossy CS were demonstrated by comparing the performance of various practical QCS methods against the derived limits.

Simulation results showed that when SQ is used, an adaptive compression of the support set and estimated magnitudes is an effective strategy. When VQ/ECVQ is used, the estimate-and-compress strategy, as supported by the theory, is the best one. Accordingly, the ECVQ based estimate-and-compress method was numerically shown to approach the remote RDF.

As illustrated by the numerical experiments, the sparsity of a signal is a feature that provides substantial compression gains, and, thus, energy savings in a CS based acquisition setup. Accordingly, finding an appropriate trade-off point in terms of the compression rate and tolerable distortion in a low-power sensor application with sparse signals is worth striving for.

Part III

Sparsity-Enabled Cognitive and Cellular Communications

8

Channel Gain Cartography for Cognitive Radios Leveraging Low Rank and Sparsity

This section focuses on channel gain cartography which aims at inferring the channel gains between two arbitrary points in space based on the measurements (samples) of the gains collected by a set of radios deployed in the area. Channel gain maps are useful for various sensing and resource allocation tasks essential for the operation of cognitive radio (CR) networks. In the considered framework, the channel gains are modeled as the tomographic accumulations of an underlying spatial loss field (SLF), which captures the attenuation in the signal strength due to the obstacles in the propagation path. In order to estimate the map accurately with a relatively small number of measurements, the SLF is postulated to have a low-rank structure possibly with sparse deviations. Efficient batch and online algorithms are developed for the resulting map reconstruction problem. Comprehensive tests with both synthetic and real datasets corroborate that the algorithms can accurately reveal the structure of the propagation medium and produce the desired channel gain maps.

Related Works

RF cartography is an instrumental concept for CR tasks [160], motivated by the under-utilization of the licensed RF spectrum [105]. Based on the measurements collected by spatially distributed CR sensors, RF cartography constructs the maps over the space, time, and frequency, portraying the RF landscape in which the CR network is deployed. Notable RF maps that have been proposed include the power spectral density (PSD) maps, which acquire the ambient interference power distribution, revealing the crowded regions that CR transceivers need to avoid [21]; and the channel gain (CG) maps, which capture the channel gains between any two points in space, allowing CR networks to perform accurate spectrum sensing and aggressive spatial reuse [161].

Prior works on channel gain cartography capitalized on experimentally validated notion of a spatial loss field (SLF) [1], which expresses the shadow fading over an arbitrary link as the weighted integral of the underlying attenuation that the RF propagation experiences due to the blocking objects in the path. Linear interpolation techniques such as kriging were employed to estimate the shadow fading based on spatially correlated measurements, and the spatio-temporal dynamics were tracked using Kalman filtering approaches [161, 79]. It is worth noting that SLF reconstruction is tantamount to the radio tomographic imaging (RTI), useful in a wide range of applications, from locating survivors in rescue operations to environmental monitoring [297, 298, 141]. The method in [297] captures the variation of the propagation medium by taking SLF differences at consecutive time slots into consideration. To cope with multipath fading in a cluttered environment, multiple channel measurements were utilized to enhance localization accuracy in [155]. However, the methods in [297, 155] do not reveal static objects in the imaging area. In contrast, a method to track moving objects using a dynamic SLF model, as well as identifying the static ones, was reported in [141]. Exploiting the sparse occupancy of the monitored area by the target objects, sparsity-leveraging algorithms for constructing obstacle maps were developed [220, 158, 219]. This work adopts a related data model, but mainly focuses on the channel gain map construction for CR applications.

Although more sophisticated methodologies for channel modeling do exist [282, 305], the computational cost and requirements on various structural/geometric prior information may hinder their use in CR applications. On the other hand, the SLF model has been experimentally validated [1], as well as in our work through a real tomographic imaging example. The proposed approach in this section provides a computationally efficient solution, by capitalizing on the inherent structure of measurement data, rather than relying heavily on the physics of RF propagation.

System Model and Problem Statement

Notation. Bold uppercase (lowercase) letters denote matrices (column vectors). Calligraphic letters are used for sets; $\mathbf{I}_n$ is the $n \times n$ identity matrix. $\mathbf{0}_n$ denotes an $n \times 1$ vector of all zeros, and $\mathbf{0}_{n\times n}$ an $n \times n$ matrix of all zeros. Operators $(\cdot)^{\mathcal{T}}$, $\operatorname{tr}(\cdot)$, and $\sigma_i(\cdot)$ represent the transposition, trace, and the i-th largest singular value of a matrix, respectively; $|\cdot|$ is used for the cardinality of a set, and the magnitude of a scalar. $\mathbf{R} \succeq \mathbf{0}$ signifies that $\mathbf{R}$ is positive semidefinite. The ℓ_1-norm of $\mathbf{X} \in \mathbb{R}^{n\times n}$ is $||\mathbf{X}||_1 := \sum_{i,j=1}^{n} |X_{ij}|$. The ℓ_∞-norm of $\mathbf{X} \in \mathbb{R}^{n\times n}$ is represented by $||\mathbf{X}||_\infty := \max\{|X_{ij}| : i, j = 1, \ldots, n\}$. For two matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times n}$, the matrix inner product $\langle \mathbf{X}, \mathbf{Y}\rangle := \operatorname{tr}(\mathbf{X}^{\mathcal{T}}\mathbf{Y})$. The Frobenius norm of matrix $\mathbf{Y}$ is $||\mathbf{Y}||_F := \sqrt{\operatorname{tr}(\mathbf{Y}\mathbf{Y}^{\mathcal{T}})}$. The spectral norm of $\mathbf{Y}$ is $||\mathbf{Y}|| := \max_{||\mathbf{x}||_2=1} ||\mathbf{Y}\mathbf{x}||_2$, and $||\mathbf{Y}||_* := \sum_i \sigma_i(\mathbf{Y})$ is the nuclear norm of $\mathbf{Y}$. For a function $h : \mathbb{R}^{m\times n} \to \mathbb{R}$, the directional derivative of h at $\mathbf{X} \in \mathbb{R}^{m\times n}$ along a direction $\mathbf{D} \in \mathbb{R}^{m\times n}$ is denoted as $h'(\mathbf{X}; \mathbf{D}) := \lim_{t\to 0+}[h(\mathbf{X} + t\mathbf{D}) - h(\mathbf{X})]/t$.

Consider a set of N CRs deployed over a geographical area represented by a two-dimensional plane $\mathcal{A} \subset \mathbb{R}^2$. Let $\mathbf{x}_n^{(t)} \in \mathcal{A}$ denote the position of CR $n \in \{1, 2, \ldots, N\}$ at time t. By exchanging pilot sequences, the CR nodes can estimate the channel gains among them. A typical channel gain between nodes n and n' can be modeled as the product of pathloss, shadowing, and small-scale fading. By averaging out the effect of the small-scale fading, the (averaged) channel gain measurement over a link (n, n') at time t, denoted by $G(\mathbf{x}_n^{(t)}, \mathbf{x}_{n'}^{(t)})$, can be represented (in dB) as

$$G(\mathbf{x}_n^{(t)}, \mathbf{x}_{n'}^{(t)}) = G_0 - \gamma 10 \log_{10} ||\mathbf{x}_n^{(t)} - \mathbf{x}_{n'}^{(t)}|| + s(\mathbf{x}_n^{(t)}, \mathbf{x}_{n'}^{(t)}) \tag{152}$$

where G_0 is the path gain at unit distance; $||\mathbf{x}_n^{(t)} - \mathbf{x}_{n'}^{(t)}||$ is the distance between nodes n and n'; γ is the pathloss exponent; and $s(\mathbf{x}_n^{(t)}, \mathbf{x}_{n'}^{(t)})$ is the attenuation due to the shadow fading. By subtracting the known pathloss component in (152), the noisy shadowing measurement

$$\check{s}(\mathbf{x}_n^{(t)}, \mathbf{x}_{n'}^{(t)}) = s(\mathbf{x}_n^{(t)}, \mathbf{x}_{n'}^{(t)}) + \epsilon(\mathbf{x}_n^{(t)}, \mathbf{x}_{n'}^{(t)}) \tag{153}$$

is obtained, where $\epsilon(\mathbf{x}_n^{(t)}, \mathbf{x}_{n'}^{(t)})$ denotes the measurement noise. Let $\mathcal{M}^{(t)}$ be the set of links, for which channel gain measurements are made at time t, and collect those measurements in vector $\check{\mathbf{s}}^{(t)} \in \mathbb{R}^{|\mathcal{M}^{(t)}|}$. The goal of channel gain cartography is to predict the channel gain between arbitrary points $\mathbf{x}$, $\mathbf{x}' \in \mathcal{A}$ at time t, based on the known nodal positions $\{\mathbf{x}_n^{(t)}\}$ and the channel gain measurements collected up to time t, that is, $\{\check{\mathbf{s}}^{(\tau)}\}_{\tau=1}^{t}$ [161, 79].

In order to achieve this interpolation, the structure of shadow fading experienced by co-located radio links will be leveraged. To this end, a variety of correlation models for shadow fading have been proposed [1, 139, 137]. In particular, the models in [161, 1, 298, 141] rely on the so-termed *spatial loss field* (SLF), which captures the attenuation due to obstacles in the line-of-sight propagation.

Let $f : \mathcal{A} \to \mathbb{R}$ denote the SLF, which captures the attenuation at location $\tilde{\mathbf{x}} \in \mathcal{A}$, and $w(\mathbf{x}, \mathbf{x}', \tilde{\mathbf{x}})$ is the weight function modeling the influence of the SLF at $\tilde{\mathbf{x}}$ to the shadowing experienced by link $\mathbf{x}$–$\mathbf{x}'$. Then, $s(\mathbf{x}, \mathbf{x}')$ is expressed as [142]

$$s(\mathbf{x}, \mathbf{x}') = \int_{\mathcal{A}} w(\mathbf{x}, \mathbf{x}', \tilde{\mathbf{x}}) f(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}}. \tag{154}$$

The *normalized ellipse model* is often used for the weight function, with w taking the form [297]

$$w(\mathbf{x}, \mathbf{x}', \tilde{\mathbf{x}}) := \begin{cases} 1/\sqrt{d(\mathbf{x}, \mathbf{x}')}, & \text{if } d(\mathbf{x}, \tilde{\mathbf{x}}) + d(\mathbf{x}', \tilde{\mathbf{x}}) \\ & < d(\mathbf{x}, \mathbf{x}') + \delta \\ 0, & \text{otherwise} \end{cases} \tag{155}$$

where $d(\mathbf{x}, \mathbf{x}') := \|\mathbf{x} - \mathbf{x}'\|$ is the distance between positions $\mathbf{x}$ and $\mathbf{x}'$, and $\delta > 0$ is a tunable parameter. The value of δ is commonly set to half the wavelength to assign non-zero weights only within the first Fresnel zone. The integral in (154) can be approximated by

$$s(\mathbf{x}, \mathbf{x}') \simeq \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} w(\mathbf{x}, \mathbf{x}', \tilde{\mathbf{x}}_{i,j}) f(\tilde{\mathbf{x}}_{i,j}) \tag{156}$$

where $\{\tilde{\mathbf{x}}_{i,j}\}_{i=1,j=1}^{N_x,N_y}$ are the pre-specified grid points over $\mathcal{A}$. Let matrix $\mathbf{F} \in \mathbb{R}^{N_x \times N_y}$ denote the SLF, sampled by the N_x-by-N_y grid. Similarly, the weight matrix $\mathbf{W}_{\mathbf{x}\mathbf{x}'}$ corresponding to link $\mathbf{x}$–$\mathbf{x}'$ is constructed. The shadow fading over link $\mathbf{x}$–$\mathbf{x}'$ in (156) can then be expressed as a linear projection of the SLF given by

$$s(\mathbf{x}, \mathbf{x}') \simeq \langle \mathbf{W}_{\mathbf{x}\mathbf{x}'}, \mathbf{F} \rangle = \text{tr}(\mathbf{W}_{\mathbf{x}\mathbf{x}'}^{\mathcal{T}} \mathbf{F}). \tag{157}$$

The goal is to form an estimate $\hat{\mathbf{F}}^{(t)}$ of $\mathbf{F}^{(t)}$ at time t, based on $\{\mathbf{x}_n^{(t)}\}$ and $\{\check{\mathbf{s}}^{(\tau)}\}_{\tau=1}^{t}$. Once $\hat{\mathbf{F}}^{(t)}$ is obtained, the shadowing and the overall channel gain across any link $\mathbf{x}$–$\mathbf{x}'$ at time t can be estimated via (157) and (152) as

$$\hat{s}(\mathbf{x}^{(t)}, \mathbf{x}'^{(t)}) = \langle \mathbf{W}_{\mathbf{x}\mathbf{x}'}^{(t)}, \hat{\mathbf{F}}^{(t)} \rangle \tag{158}$$

$$\hat{G}(\mathbf{x}^{(t)}, \mathbf{x}'^{(t)}) = G_0 - \gamma 10 \log_{10} ||\mathbf{x}^{(t)} - \mathbf{x}'^{(t)}|| + \hat{s}(\mathbf{x}^{(t)}, \mathbf{x}'^{(t)}). \tag{159}$$

The number of unknown $\mathbf{F}^{(t)}$ entries is less than $N_x N_y$, while the number of measurements is $\mathcal{O}(tN^2)$, provided that the SLF remains invariant for t slots. If the number of entries to be estimated in $\mathbf{F}^{(t)}$ is larger than the number of measurements, the problem is underdetermined and cannot be solved uniquely. To overcome this and further improve the performance even in the overdetermined cases, a priori knowledge on the structure of $\mathbf{F}^{(t)}$ will be exploited next to regularize the problem.

Channel Gain Prediction Using Low Rank and Sparsity

Problem Formulation

The low-rank plus sparse structure has been advocated in various problems in machine learning and signal processing [40, 102, 207]. Low-rank matrices are effective in capturing slow variation or regular patterns, and sparsity is instrumental for incorporating robustness against outliers. Inspired by these, we postulate that $\mathbf{F}$ has a low-rank-plus-sparse structure as

$$\mathbf{F} = \mathbf{L} + \mathbf{E} \tag{160}$$

where matrix $\mathbf{L}$ is low-rank, and $\mathbf{E}$ is sparse. This model is particularly attractive in urban or indoor scenarios where the obstacles often possess regular patterns, while the sparse term can capture irregularities that do not conform to the low-rank model.

Redefine $\mathbf{W}_{nn'}^{(t)} := \mathbf{W}_{\mathbf{x}_n\mathbf{x}_{n'}}^{(t)}$ and $\check{s}_{nn'}^{(t)} := \check{s}(\mathbf{x}_n^{(t)}, \mathbf{x}_{n'}^{(t)})$ for brevity.[21]Toward estimating $\mathbf{F}^{(t)}$ that obeys (160), consider the cost

$$c^{(t)}(\mathbf{L}, \mathbf{E}) := \frac{1}{2} \sum_{(n,n') \in \mathcal{M}^{(t)}} \left(\langle \mathbf{W}_{nn'}^{(t)}, \mathbf{L} + \mathbf{E} \rangle - \check{s}_{nn'}^{(t)} \right)^2 \tag{161}$$

which fits the shadowing measurements to the model. Then, with T denoting the total number of time slots taking measurements, we adopt the following optimization criterion

$$\text{(P1)} \quad \min_{\mathbf{L}, \mathbf{E} \in \mathbb{R}^{N_x \times N_y}} \sum_{\tau=1}^{T} \beta^{T-\tau} \left[c^{(\tau)}(\mathbf{L}, \mathbf{E}) + \lambda ||\mathbf{L}||_* + \mu ||\mathbf{E}||_1 \right] \tag{162}$$

where $\beta \in (0, 1]$ is the forgetting factor that can be optionally put in to weigh the recent observations more heavily. The nuclear norm regularization term promotes a low-rank $\mathbf{L}$, while the ℓ_1-norm encourages sparsity in $\mathbf{E}$. Parameters λ and μ are appropriately chosen to control the effect of these regularizers. Conditions for exact recovery through a related convex formulation in the absence of measurement noise can be found in [307].

Problem (162) is convex, and can be tackled using existing efficient solvers, such as the interior-point method. Once the optimal $\hat{\mathbf{L}}$ and $\hat{\mathbf{E}}$ are found, the desired $\hat{\mathbf{F}}$ is obtained as $\hat{\mathbf{F}} = \hat{\mathbf{L}} + \hat{\mathbf{E}}$. However, the general-purpose optimization packages tend to scale poorly as the problem size grows. Specialized algorithms developed for related problems often employ costly SVD operations iteratively [307]. Furthermore, such an algorithm might not be amenable for an online implementation. Building on [206] and [245], an efficient solution is proposed next with reduced complexity.

[21] Prompted by [155], the benefit of multi-channel diversity for RTI may be incorporated in the presented framework. Suppose K channels $\mathcal{K}_{nn'}^{(t)}$ are available to sensors n and n' at time t, and let $\acute{s}_{nn',k}^{(t)}$ denote the noisy measurement including fading over link $\mathbf{x}_n$–$\mathbf{x}_{n'}$ at t in channel $k \in \mathcal{K}_{nn'}^{(t)}$. Construct a new measurement as $\bar{s}_{nn'}^{(t)} = \phi(\acute{s}_{nn',1}^{(t)}, \acute{s}_{nn',2}^{(t)}, \cdots, \acute{s}_{nn',K}^{(t)})$, where $\phi(\cdot)$ is a channel selection function [155]. By replacing $\check{s}_{nn'}^{(t)}$ in (161) with $\bar{s}_{nn'}^{(t)}$, the multiple channel measurements can be incorporated without altering the method. However, the dynamic channel availability and multi-channel measurements will increase algorithm complexity. On the other hand, it is not clear whether such a multi-channel approach can be adopted for estimating *any channel gain* over multiple frequency bands.

Efficient Batch Solution

Without loss of generality, consider replacing $\mathbf{L}$ with the low-rank product $\mathbf{P}\mathbf{Q}^{\mathcal{T}}$, where $\mathbf{P} \in \mathbb{R}^{N_x \times \rho}$ and $\mathbf{Q} \in \mathbb{R}^{N_y \times \rho}$, and ρ is a pre-specified overestimate of the rank of $\mathbf{L}$. It is known that (e.g., [245])

$$||\mathbf{L}||_* = \min_{\mathbf{P},\mathbf{Q}} \frac{1}{2}\left(||\mathbf{P}||_F^2 + ||\mathbf{Q}||_F^2\right)$$
$$\text{subject to } \mathbf{L} = \mathbf{P}\mathbf{Q}^{\mathcal{T}}. \tag{163}$$

Thus, a natural re-formulation of (162) is (see also [206])

$$\text{(P2)} \quad \min_{\mathbf{P},\mathbf{Q},\mathbf{E}} f(\mathbf{P},\mathbf{Q},\mathbf{E}) := \tag{164}$$
$$\sum_{\tau=1}^{T} \beta^{T-\tau}\left[c^{(\tau)}(\mathbf{P}\mathbf{Q}^{\mathcal{T}},\mathbf{E}) + \frac{\lambda}{2}\left(||\mathbf{P}||_F^2 + ||\mathbf{Q}||_F^2\right) + \mu||\mathbf{E}||_1\right].$$

Instead of seeking the N_xN_y entries of $\mathbf{L}$, the factorization approach (164) entails only $(N_x + N_y)\rho$ unknowns, thus reducing complexity and memory requirements significantly when $\rho \ll \min\{N_x, N_y\}$. Furthermore, adoption of the separable Frobenius norm regularizer in (P2) comes with no loss of optimality as asserted in the following lemma.

Lemma 3: *If* $\{\hat{\mathbf{L}}, \hat{\mathbf{E}}\}$ *minimize (P1) and we choose* $\rho \geq$ *rank*$(\hat{\mathbf{L}})$*, then, (P2) is equivalent to (P1) at the minimum.*

Proof: It is clear that the minimum of (P1) is no larger than that of

$$\min_{\mathbf{P},\mathbf{Q},\mathbf{E}} \sum_{\tau=1}^{T} \beta^{T-\tau}\left[c^{(\tau)}(\mathbf{P}\mathbf{Q}^{\mathcal{T}},\mathbf{E}) + \lambda||\mathbf{P}\mathbf{Q}^{\mathcal{T}}||_* + \mu||\mathbf{E}||_1\right] \tag{165}$$

since the search space is reduced by the re-parametrization $\mathbf{L} = \mathbf{P}\mathbf{Q}^{\mathcal{T}}$ with $\rho \leq \min\{N_x, N_y\}$. Now (163) implies that the minimum of (165) is no larger than that of (P2). However, the inequality is tight since the objectives of (P1) and (P2) are identical for $\mathbf{E} := \hat{\mathbf{E}}$, $\mathbf{P} := \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}^{1/2}$, and $\mathbf{Q} := \hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{1/2}$, where $\hat{\mathbf{L}} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}^{\mathcal{T}}$ is the SVD. Consequently, (P1) and (P2) have identical costs at the minimum. ■

Although (P1) is a convex optimization problem, (P2) is not. Thus, in general, one can obtain only a *locally* optimal solution of (P2), which may not be the globally optimal solution of (P1). Interestingly, under appropriate conditions, global optimality can be guaranteed for the local optima of (P2), as claimed in the following proposition.

Proposition 7: *If* $\{\bar{\mathbf{P}}, \bar{\mathbf{Q}}, \bar{\mathbf{E}}\}$ *is a stationary point of (P2),* $\bar{\beta} := \sum_{\tau=1}^{T} \beta^{T-\tau}$*, and* $||\tilde{f}(\bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}}, \bar{\mathbf{E}})|| \leq \lambda\bar{\beta}$ *with*

$$\tilde{f}(\hat{\mathbf{L}}, \hat{\mathbf{E}}) := \sum_{\tau=1}^{T} \beta^{T-\tau} \left[\sum_{(n,n')\in\mathcal{M}^{(\tau)}} \left(\langle \mathbf{W}_{nn'}^{(\tau)}, \hat{\mathbf{L}} + \hat{\mathbf{E}} \rangle - \check{s}_{nn'}^{(\tau)} \right) \mathbf{W}_{nn'}^{(\tau)} \right] \tag{166}$$

then $\{\hat{\mathbf{L}} := \bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}}, \hat{\mathbf{E}} := \bar{\mathbf{E}}\}$ *is a globally optimal solution to (P1).*

Proof: See Appendix G.

A stationary point of (P2) can be obtained through a block coordinate-descent (BCD) algorithm, where the optimization is performed in a cyclic fashion over one of $\{\mathbf{E}, \mathbf{P}, \mathbf{Q}\}$ with the remaining two variables fixed. In fact, since the term $\mu||\mathbf{E}||_1$ is separable in the individual entries as well, the cyclic update can be stretched all the way up to the individual entries of $\mathbf{E}$ without affecting convergence [280]. The proposed solver entails an iterative procedure comprising three steps per iteration $k = 1, 2, \ldots$

[S1] Update E:

$$\mathbf{E}[k+1] = \arg\min_{\mathbf{E}} \sum_{\tau=1}^{T} \beta^{T-\tau} \left[c^{(\tau)}(\mathbf{P}[k]\mathbf{Q}^{\mathcal{T}}[k], \mathbf{E}) + \mu||\mathbf{E}||_1 \right]$$

[S2] Update P:

$$\mathbf{P}[k+1] = \arg\min_{\mathbf{P}} \sum_{\tau=1}^{T} \beta^{T-\tau} \left[c^{(\tau)}(\mathbf{P}\mathbf{Q}^{\mathcal{T}}[k], \mathbf{E}[k+1]) + \frac{\lambda}{2}||\mathbf{P}||_F^2 \right]$$

[S3] Update Q:

$$\mathbf{Q}[k+1] = \arg\min_{\mathbf{Q}} \sum_{\tau=1}^{T} \beta^{T-\tau} \left[c^{(\tau)}(\mathbf{P}[k+1]\mathbf{Q}^{\mathcal{T}}, \mathbf{E}[k+1]) + \frac{\lambda}{2}||\mathbf{Q}||_F^2 \right].$$

To update each block variable, the cost in (P2) is minimized while fixing the other block variables to their up-to-date iterates.

To detail the update rules, let $\boldsymbol{\mathcal{W}}^{(t)} \in \mathbb{R}^{N_x N_y \times |\mathcal{M}(t)|}$ be a matrix with columns equal to $\mathrm{vec}\left(\mathbf{W}_{nn'}^{(t)}\right)$ for $(n, n') \in \mathcal{M}(t)$, where $\mathrm{vec}(\cdot)$ produces a column vector by stacking the columns of a matrix one below the other ($\mathrm{unvec}(\cdot)$ denotes the reverse process). Define $\boldsymbol{\mathcal{W}} := [\sqrt{\beta^{T-1}}\boldsymbol{\mathcal{W}}^{(1)} \ldots \sqrt{\beta^{0}}\boldsymbol{\mathcal{W}}^{(T)}]$, $\check{\mathbf{s}} := [\sqrt{\beta^{T-1}}\check{\mathbf{s}}^{(1)\mathcal{T}} \ldots \sqrt{\beta^{0}}\check{\mathbf{s}}^{(T)\mathcal{T}}]^{\mathcal{T}}$, and $\mathbf{e} := \mathrm{vec}(\mathbf{E})$. Then, one can write $\sum_{\tau=1}^{T} \beta^{T-\tau} c^{(\tau)}(\mathbf{P}\mathbf{Q}^{\mathcal{T}}, \mathbf{E}) = \|\boldsymbol{\mathcal{W}}^{\mathcal{T}}\mathrm{vec}(\mathbf{P}\mathbf{Q}^{\mathcal{T}} + \mathbf{E}) - \check{\mathbf{s}}\|_2^2$. Let e_l denote the l-th entry of $\mathbf{e}$, and $\mathbf{e}_{-l}$ represent the replica of $\mathbf{e}$ without its

l-th entry. Similarly, let $\boldsymbol{\omega}_l^{\mathcal{T}}$ denote the l-th row of the matrix $\boldsymbol{\mathcal{W}}$, and $\boldsymbol{\mathcal{W}}_{-l}$ denote the matrix $\boldsymbol{\mathcal{W}}$ with its l-th row removed. The soft-thresholding function soft_th$(\cdot;\mu)$ is defined as

$$\text{soft\_th}(x;\mu) := \text{sgn}(x)\max\{0, |x| - \mu\}. \tag{167}$$

Minimization in **[S1]** proceeds sequentially over the individual entries of $\mathbf{e}$. At iteration k, each entry is updated via

$$e_l[k+1] = \arg\min_{e_l} \frac{1}{2}||e_l\boldsymbol{\omega}_l - \check{\check{\mathbf{s}}}||_2^2 + \mu\bar{\beta}|e_l|,\ l = 1,\ldots,N_xN_y \tag{168}$$

where $\check{\check{\mathbf{s}}}_l[k] := \check{\mathbf{s}} - \boldsymbol{\mathcal{W}}^{\mathcal{T}}\text{vec}(\mathbf{P}[k]\mathbf{Q}^{\mathcal{T}}[k]) - \boldsymbol{\mathcal{W}}_{-l}^{\mathcal{T}}\mathbf{e}_{-l}$. The closed-form solution for e_l is obtained as

$$e_l[k+1] = \frac{\text{soft\_th}(\boldsymbol{\omega}_l^{\mathcal{T}}\check{\check{\mathbf{s}}}_l[k];\mu\bar{\beta})}{||\boldsymbol{\omega}_l||_2^2}. \tag{169}$$

Matrices $\mathbf{P}$ and $\mathbf{Q}$ are similarly updated over their rows through **[S2]** and **[S3]**. Let $\mathbf{p}_i$ be the i-th row of $\mathbf{P}$, transposed to a column vector; i.e., $\mathbf{P} := [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{N_x}]^{\mathcal{T}}$. Define $\tilde{\boldsymbol{\mathcal{W}}}_i^{(t)} \in \mathbb{R}^{|\mathcal{M}(t)|\times N_y}$ to be the matrix whose rows are the i-th rows of $\{\mathbf{W}_{nn'}^{(t)}\}_{(n,n')\in\mathcal{M}(t)}$ denoted as $\tilde{\mathbf{w}}_{nn',i}^{(t)\mathcal{T}}$, and $\tilde{\mathbf{s}}_i^{(t)} \in \mathbb{R}^{|\mathcal{M}(t)|}$ a vector with entries equal to

$$\tilde{s}_{nn',i}^{(t)} := \check{s}_{nn'}^{(t)} - \langle\mathbf{W}_{nn'}^{(t)}, \mathbf{E}[k+1]\rangle - \sum_{j\neq i}^{N_x}\tilde{\mathbf{w}}_{nn',j}^{(t)\mathcal{T}}\mathbf{Q}[k]\mathbf{p}_j \tag{170}$$

for $(n,n') \in \mathcal{M}(t)$. Define also $\tilde{\boldsymbol{\mathcal{W}}}_i := [\sqrt{\beta^{T-1}}\tilde{\boldsymbol{\mathcal{W}}}_i^{(1)\mathcal{T}} \ldots \sqrt{\beta^0}\tilde{\boldsymbol{\mathcal{W}}}_i^{(T)\mathcal{T}}]^{\mathcal{T}}$ and $\tilde{\mathbf{s}}_i := [\sqrt{\beta^{T-1}}\tilde{\mathbf{s}}_i^{(1)\mathcal{T}} \ldots \sqrt{\beta^0}\tilde{\mathbf{s}}_i^{(T)\mathcal{T}}]^{\mathcal{T}}$. Then, $\mathbf{p}_i$ is updated by solving a ridge-regression problem as

$$\mathbf{p}_i[k+1] = \arg\min_{\mathbf{p}_i}\left[\frac{1}{2}||\tilde{\boldsymbol{\mathcal{W}}}_i\mathbf{Q}[k]\mathbf{p}_i - \tilde{\mathbf{s}}_i||_2^2 + \frac{\lambda\bar{\beta}}{2}||\mathbf{p}_i||_2^2\right]$$

whose solution is given in closed form by

$$\mathbf{p}_i[k+1] = \left[\mathbf{Q}^{\mathcal{T}}[k]\tilde{\boldsymbol{\mathcal{W}}}_i^{\mathcal{T}}\tilde{\boldsymbol{\mathcal{W}}}_i\mathbf{Q}[k] + \lambda\bar{\beta}\mathbf{I}_\rho\right]^{-1}\mathbf{Q}^{\mathcal{T}}[k]\tilde{\boldsymbol{\mathcal{W}}}_i^{\mathcal{T}}\tilde{\mathbf{s}}_i \tag{171}$$

which involves matrix inversion of dimension only ρ-by-ρ. Likewise, let $\mathbf{q}_i$ denote the i-th row of $\mathbf{Q}$, transposed to a column vector; i.e., $\mathbf{Q}$:=

$[\mathbf{q}_1, \ldots, \mathbf{q}_{N_y}]^{\mathcal{T}}$. Define also $\breve{\boldsymbol{\mathcal{W}}}_i := [\sqrt{\beta^{T-1}}\breve{\boldsymbol{\mathcal{W}}}_i^{(1)\mathcal{T}} \ldots \sqrt{\beta^0}\breve{\boldsymbol{\mathcal{W}}}_i^{(T)\mathcal{T}}]^{\mathcal{T}}$ and $\breve{\mathbf{s}}_i := [\sqrt{\beta^{T-1}}\breve{\mathbf{s}}_i^{(1)\mathcal{T}} \ldots \sqrt{\beta^0}\breve{\mathbf{s}}_i^{(T)\mathcal{T}}]^{\mathcal{T}}$, where $\breve{\boldsymbol{\mathcal{W}}}_i^{(t)} \in \mathbb{R}^{|\mathcal{M}(t)| \times N_x}$ is the matrix whose rows are the transpositions of the i-th columns of $\{\mathbf{W}_{nn'}^{(t)}\}_{(n,n') \in \mathcal{M}(t)}$, denoted as $\breve{\mathbf{w}}_{nn',i}^{(t)}$, and $\breve{\mathbf{s}}_i^{(t)} \in \mathbb{R}^{|\mathcal{M}(t)|}$ has entries

$$\breve{s}_{nn',i}^{(t)} := \breve{s}_{nn'}^{(t)} - \langle \mathbf{W}_{nn'}^{(t)}, \mathbf{E}[k+1] \rangle - \sum_{j \neq i}^{N_y} \breve{\mathbf{w}}_{nn',j}^{(t)\mathcal{T}} \mathbf{P}[k+1]\mathbf{q}_j \tag{172}$$

for $(n, n') \in \mathcal{M}(t)$. The update for $\mathbf{q}_i$ is then given by solving another ridge regression problem to obtain

$$\mathbf{q}_i[k+1] = \arg\min_{\mathbf{q}_i} \left[\frac{1}{2} ||\breve{\boldsymbol{\mathcal{W}}}_i \mathbf{P}[k+1]\mathbf{q}_i - \breve{\mathbf{s}}_i||_2^2 + \frac{\lambda\bar{\beta}}{2} ||\mathbf{q}_i||_2^2 \right]$$

whose solution is given also in closed form by

$$\mathbf{q}_i[k+1] = \left[\mathbf{P}^{\mathcal{T}}[k+1] \breve{\boldsymbol{\mathcal{W}}}_i^{\mathcal{T}} \breve{\boldsymbol{\mathcal{W}}}_i \mathbf{P}[k+1] + \lambda\bar{\beta}\mathbf{I}_\rho \right]^{-1} \mathbf{P}^{\mathcal{T}}[k+1] \breve{\boldsymbol{\mathcal{W}}}_i^{\mathcal{T}} \breve{\mathbf{s}}_i \tag{173}$$

which again involves matrix inversion of dimension ρ-by-ρ. The overall algorithm is tabulated in Table 5.

Although the proposed batch algorithm exhibits low computational and memory requirements, it is not suitable for online processing, since (164) must be re-solved every time a new set of measurements arrive, incurring major computational burden. Thus, the development of an online recursive algorithm is well motivated.

Online Algorithm

Stochastic Approximation Approach

In practice, it is often the case that a new set of data becomes available *sequentially* in time. Then, it is desirable to have an algorithm that can process the newly acquired data incrementally and refine the previous estimates, rather than re-computing the batch solution, which may incur prohibitively growing computational burden. Furthermore, when the channel is time-varying due to, e.g., mobile obstacles, online algorithms can readily track such variations.

Stochastic approximation (SA) is an appealing strategy for deriving online algorithms [267, 169]. Moreover, techniques involving minimizing majorized

Table 5: Batch solver of (P2)

```
**Initialize** $\mathbf{E}[1] := \mathbf{0}_{N_x \times N_y}$, $\mathbf{P}[1]$ and $\mathbf{Q}[1]$ at random.
**For** $k = 1, 2, \ldots$
    **[S1]** *Update* $\mathbf{E}$*:*
    **Set** $\mathbf{e} = \mathrm{vec}(\mathbf{E}[k])$
    **For** $l = 1, 2, \ldots, N_x N_y$
        **Set** $\check{\mathbf{s}}_l[k] := \check{\mathbf{s}} - \boldsymbol{\mathcal{W}}^{\mathcal{T}}\mathrm{vec}(\mathbf{P}[k]\mathbf{Q}^{\mathcal{T}}[k]) - \boldsymbol{\mathcal{W}}_{-l}^{\mathcal{T}}\mathbf{e}_{-l}$
        $e_l[k+1] = \mathrm{soft\_th}(\boldsymbol{\omega}_l^{\mathcal{T}}\check{\mathbf{s}}_l[k]; \mu\bar{\beta})/||\boldsymbol{\omega}_l||_2^2$
    **Next** $l$
    **Set** $\mathbf{E}[k+1] = \mathrm{unvec}(\mathbf{e}[k+1])$
    **[S2]** *Update* $\mathbf{P}$*:*
    **For** $i = 1, 2, \ldots, N_x$
        **Set** $\tilde{\boldsymbol{\mathcal{W}}}_i$ and $\tilde{\mathbf{s}}_i$
        $\mathbf{p}_i[k+1] = \left[\mathbf{Q}^{\mathcal{T}}[k]\tilde{\boldsymbol{\mathcal{W}}}_i^{\mathcal{T}}\tilde{\boldsymbol{\mathcal{W}}}_i\mathbf{Q}[k] + \lambda\bar{\beta}\mathbf{I}_\rho\right]^{-1}(\mathbf{Q}^{\mathcal{T}}[k]\tilde{\boldsymbol{\mathcal{W}}}_i^{\mathcal{T}}\tilde{\mathbf{s}}_i)$
    **Next** $i$
    $\mathbf{P}[k+1] = [\mathbf{p}_1[k+1], \mathbf{p}_2[k+1], \ldots, \mathbf{p}_{N_x}[k+1]]^{\mathcal{T}}$
    **[S3]** *Update* $\mathbf{Q}$*:*
    **For** $i = 1, 2, \ldots, N_y$
        **Set** $\check{\boldsymbol{\mathcal{W}}}_i$ and $\check{\mathbf{s}}_i$
        $\mathbf{q}_i[k+1] = \left[\mathbf{P}^{\mathcal{T}}[k+1]\check{\boldsymbol{\mathcal{W}}}_i^{\mathcal{T}}\check{\boldsymbol{\mathcal{W}}}_i\mathbf{P}[k+1] + \lambda\bar{\beta}\mathbf{I}_\rho\right]^{-1}$
                                   $\times(\mathbf{P}^{\mathcal{T}}[k+1]\check{\boldsymbol{\mathcal{W}}}_i^{\mathcal{T}}\check{\mathbf{s}}_i)$
    **Next** $i$
    $\mathbf{Q}[k+1] = [\mathbf{q}_1[k+1], \mathbf{q}_2[k+1], \ldots, \mathbf{q}_{N_y}[k+1]]^{\mathcal{T}}$
**Next** $k$
```

surrogate functions were developed to handle nonconvex cost functions in online settings [206, 208, 199, 242]. An online algorithm to solve a dictionary learning problem was proposed in [199]. A stochastic gradient descent algorithm was derived for subspace tracking and anomaly detection in [206]. Next, an online algorithm for the CPCP problem is developed. The proposed approach employs quadratic surrogate functions with diagonal weighting to capture disparate curvatures in the directions of different block variables.

For simplicity, let the number of measurements per time slot t be constant $M := |\mathcal{M}^{(t)}|$ for all t. Define $\mathbf{X} := (\mathbf{P}, \mathbf{Q}, \mathbf{E}) \in \mathcal{X} \subset \mathcal{X}' := \mathbb{R}^{(N_x \times \rho)} \times \mathbb{R}^{(N_y \times \rho)} \times \mathbb{R}^{(N_x \times N_y)}$, where $\mathcal{X}$ is a compact convex set, and $\mathcal{X}'$ a bounded open set, and $\boldsymbol{\xi}^{(t)} := [\{\check{s}_m^{(t)}\}_{m=1}^M, \{\mathbf{W}_m^{(t)}\}_{m=1}^M] \in \varXi$, where $\varXi$ is assumed to be bounded. Define with slight abuse of notation

$$\begin{aligned} g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}) &= g_1(\mathbf{P}, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)}) \\ &:= \frac{1}{2} \sum_{m=1}^{M} \left(\langle \mathbf{W}_m^{(t)}, \mathbf{P}\mathbf{Q}^{\mathcal{T}} + \mathbf{E} \rangle - \check{s}_m^{(t)} \right)^2 \end{aligned} \tag{174}$$

$$g_2(\mathbf{X}) = g_2(\mathbf{P}, \mathbf{Q}, \mathbf{E}) := \frac{\lambda}{2} \left(||\mathbf{P}||_F^2 + ||\mathbf{Q}||_F^2 \right) + \mu ||\mathbf{E}||_1. \tag{175}$$

A quadratic surrogate function for $g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)})$ is then constructed as

$$\begin{aligned} &\check{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) := g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \\ &\quad + \langle \mathbf{P} - \mathbf{P}^{(t-1)}, \nabla_{\mathbf{P}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \rangle + \frac{\eta_{\mathbf{P}}^{(t)}}{2} \|\mathbf{P} - \mathbf{P}^{(t-1)}\|_F^2 \\ &\quad + \langle \mathbf{Q} - \mathbf{Q}^{(t-1)}, \nabla_{\mathbf{Q}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \rangle + \frac{\eta_{\mathbf{Q}}^{(t)}}{2} \|\mathbf{Q} - \mathbf{Q}^{(t-1)}\|_F^2 \\ &\quad + \langle \mathbf{E} - \mathbf{E}^{(t-1)}, \nabla_{\mathbf{E}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \rangle + \frac{\eta_{\mathbf{E}}^{(t)}}{2} \|\mathbf{E} - \mathbf{E}^{(t-1)}\|_F^2 \end{aligned} \tag{176}$$

where $\eta_{\mathbf{P}}^{(t)}$, $\eta_{\mathbf{Q}}^{(t)}$, and $\eta_{\mathbf{E}}^{(t)}$ are positive constants, and with $\tilde{\tilde{f}}_m^{(t)}(\mathbf{P}, \mathbf{Q}, \mathbf{E}) := \langle \mathbf{W}_m^{(t)}, \mathbf{P}\mathbf{Q}^{\mathcal{T}} + \mathbf{E} \rangle - \check{s}_m^{(t)}$ it can be readily verified that

$$\nabla_{\mathbf{P}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) = \sum_{m=1}^{M} \tilde{\tilde{f}}_m^{(t)}(\mathbf{P}^{(t-1)}, \mathbf{Q}^{(t-1)}, \mathbf{E}^{(t-1)}) \mathbf{W}_m^{(t)} \mathbf{Q}^{(t-1)} \tag{177}$$

$$\nabla_{\mathbf{Q}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) = \sum_{m=1}^{M} \tilde{\tilde{f}}_m^{(t)}(\mathbf{P}^{(t-1)}, \mathbf{Q}^{(t-1)}, \mathbf{E}^{(t-1)}) {\mathbf{W}_m^{(t)}}^{\mathcal{T}} \mathbf{P}^{(t-1)} \tag{178}$$

$$\nabla_{\mathbf{E}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) = \sum_{m=1}^{M} \tilde{\tilde{f}}_m^{(t)}(\mathbf{P}^{(t-1)}, \mathbf{Q}^{(t-1)}, \mathbf{E}^{(t-1)}) \mathbf{W}_m^{(t)}. \tag{179}$$

Let us focus on the case without the forgetting factor, i.e., $\beta = 1$. A convergent SA algorithm for (P2) is obtained by considering the following surrogate problem

$$\text{(P3)} \quad \min_{\mathbf{X}} \frac{1}{t} \sum_{\tau=1}^{t} \left[\check{g}_1(\mathbf{X}, \mathbf{X}^{(\tau-1)}, \boldsymbol{\xi}^{(\tau)}) + g_2(\mathbf{X}) \right]. \tag{180}$$

In fact, solving (P3) yields a stochastic gradient descent (SGD) algorithm. In particular, since variables $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{E}$ can be separately optimized in (P3), the proposed algorithm updates the variables in parallel in each time slot t as

$$\begin{aligned} \mathbf{P}^{(t)} = \arg\min_{\mathbf{P}} \sum_{\tau=1}^{t} \Big[& \langle \mathbf{P} - \mathbf{P}^{(\tau-1)}, \nabla_{\mathbf{P}} g_1(\mathbf{X}^{(\tau-1)}, \boldsymbol{\xi}^{(\tau)}) \rangle \\ & + \frac{\eta_{\mathbf{P}}^{(\tau)}}{2} ||\mathbf{P} - \mathbf{P}^{(\tau-1)}||_F^2 + \frac{\lambda}{2} ||\mathbf{P}||_F^2 \Big] \end{aligned} \tag{181}$$

$$\begin{aligned} \mathbf{Q}^{(t)} = \arg\min_{\mathbf{Q}} \sum_{\tau=1}^{t} \Big[& \langle \mathbf{Q} - \mathbf{Q}^{(\tau-1)}, \nabla_{\mathbf{Q}} g_1(\mathbf{X}^{(\tau-1)}, \boldsymbol{\xi}^{(\tau)}) \rangle \\ & + \frac{\eta_{\mathbf{Q}}^{(\tau)}}{2} ||\mathbf{Q} - \mathbf{Q}^{(\tau-1)}||_F^2 + \frac{\lambda}{2} ||\mathbf{Q}||_F^2 \Big] \end{aligned} \tag{182}$$

$$\begin{aligned} \mathbf{E}^{(t)} = \arg\min_{\mathbf{E}} \sum_{\tau=1}^{t} \Big[& \langle \mathbf{E} - \mathbf{E}^{(\tau-1)}, \nabla_{\mathbf{E}} g_1(\mathbf{X}^{(\tau-1)}, \boldsymbol{\xi}^{(\tau)}) \rangle \\ & + \frac{\eta_{\mathbf{E}}^{(\tau)}}{2} ||\mathbf{E} - \mathbf{E}^{(\tau-1)}||_F^2 + \mu ||\mathbf{E}||_1 \Big]. \end{aligned} \tag{183}$$

By checking the first-order optimality conditions, and defining $\bar{\eta}_{\mathbf{P}}^{(t)} := \sum_{\tau=1}^{t} \eta_{\mathbf{P}}^{(\tau)}$ and $\bar{\eta}_{\mathbf{Q}}^{(t)} := \sum_{\tau=1}^{t} \eta_{\mathbf{Q}}^{(\tau)}$, the update rules for $\mathbf{P}$ and $\mathbf{Q}$ are obtained as

$$\mathbf{P}^{(t)} = \frac{1}{\bar{\eta}_{\mathbf{P}}^{(t)} + \lambda t} \sum_{\tau=1}^{t} \left[\eta_{\mathbf{P}}^{(\tau)} \mathbf{P}^{(\tau-1)} - \nabla_{\mathbf{P}} g_1(\mathbf{X}^{(\tau-1)}, \boldsymbol{\xi}^{(\tau)}) \right] \tag{184}$$

$$\mathbf{Q}^{(t)} = \frac{1}{\bar{\eta}_{\mathbf{Q}}^{(t)} + \lambda t} \sum_{\tau=1}^{t} \left[\eta_{\mathbf{Q}}^{(\tau)} \mathbf{Q}^{(\tau-1)} - \nabla_{\mathbf{Q}} g_1(\mathbf{X}^{(\tau-1)}, \boldsymbol{\xi}^{(\tau)}) \right] \tag{185}$$

which can be written in recursive forms as

$$\mathbf{P}^{(t)} = \mathbf{P}^{(t-1)} - \frac{1}{\bar{\eta}_{\mathbf{P}}^{(t)} + \lambda t} \left(\nabla_{\mathbf{P}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) + \lambda \mathbf{P}^{(t-1)} \right) \tag{186}$$

$$\mathbf{Q}^{(t)} = \mathbf{Q}^{(t-1)} - \frac{1}{\bar{\eta}_{\mathbf{Q}}^{(t)} + \lambda t} \left(\nabla_{\mathbf{Q}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) + \lambda \mathbf{Q}^{(t-1)} \right). \tag{187}$$

Table 6: Online SGD solver of (P2)

```
**Initialize** $\mathbf{E}^{(0)} := \mathbf{0}_{N_x \times N_y}$, $\mathbf{P}^{(0)}$ and $\mathbf{Q}^{(0)}$ at random.
**For** $t = 1, 2, \ldots$
    **Set** $L_{\mathbf{P}} = \sum_{m=1}^{M} \left\| \mathbf{W}_m^{(t)} \mathbf{Q}^{(t-1)} \right\|_F^2$,
        $L_{\mathbf{Q}} = \sum_{m=1}^{M} \left\| \mathbf{W}_m^{(t)\mathcal{T}} \mathbf{P}^{(t-1)} \right\|_F^2$,
        $L_{\mathbf{E}} = \sum_{m=1}^{M} \left\| \mathbf{W}_m^{(t)} \right\|_F^2$, and $L_{\min} = \min\{L_{\mathbf{P}}, L_{\mathbf{Q}}, L_{\mathbf{E}}\}$.
    **Set** $\eta_{\mathbf{P}}^{(t)} \geq \frac{L_{\mathbf{P}}^2}{L_{\min}}$, $\eta_{\mathbf{Q}}^{(t)} \geq \frac{L_{\mathbf{Q}}^2}{L_{\min}}$, and $\eta_{\mathbf{E}}^{(t)} \geq \frac{L_{\mathbf{E}}^2}{L_{\min}}$.
    **Set** $\bar{\eta}_{\mathbf{P}}^{(t)} = \sum_{\tau=1}^{t} \eta_{\mathbf{P}}^{(\tau)}$, $\bar{\eta}_{\mathbf{Q}}^{(t)} = \sum_{\tau=1}^{t} \eta_{\mathbf{Q}}^{(\tau)}$, and $\bar{\eta}_{\mathbf{E}}^{(t)} = \sum_{\tau=1}^{t} \eta_{\mathbf{E}}^{(\tau)}$.
    $\mathbf{P}^{(t)} = \mathbf{P}^{(t-1)} - \frac{1}{\bar{\eta}_{\mathbf{P}}^{(t)} + \lambda t} \left( \nabla_{\mathbf{P}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) + \lambda \mathbf{P}^{(t-1)} \right)$
    $\mathbf{Q}^{(t)} = \mathbf{Q}^{(t-1)} - \frac{1}{\bar{\eta}_{\mathbf{Q}}^{(t)} + \lambda t} \left( \nabla_{\mathbf{Q}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) + \lambda \mathbf{Q}^{(t-1)} \right)$
    $\mathbf{Z}^{(t)} = \frac{1}{\bar{\eta}_{\mathbf{E}}^{(t)}} \left[ \eta_{\mathbf{E}}^{(t)} \mathbf{E}^{(t-1)} + \bar{\eta}_{\mathbf{E}}^{(t-1)} \mathbf{Z}^{(t-1)} - \nabla_{\mathbf{E}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \right]$
    $\mathbf{E}^{(t)} = \text{soft\_th}(\mathbf{Z}^{(t)}; \mu t / \bar{\eta}_{\mathbf{E}}^{(t)})$
**Next** $t$
```

Due to the non-smoothness of $||\mathbf{E}||_1$, the update for $\mathbf{E}$ proceeds in two steps. First, an auxiliary variable $\mathbf{Z}^{(t)}$ is introduced, which is computed as

$$\mathbf{Z}^{(t)} = \frac{1}{\bar{\eta}_{\mathbf{E}}^{(t)}} \left[\sum_{k=1}^{t} \eta_{\mathbf{E}}^{(k)} \mathbf{E}^{(k-1)} - \nabla_{\mathbf{E}} g_1(\mathbf{X}^{(k-1)}, \boldsymbol{\xi}^{(k)}) \right]. \tag{188}$$

Again defining $\bar{\eta}_{\mathbf{E}}^{(t)} := \sum_{\tau=1}^{t} \eta_{\mathbf{E}}^{(\tau)}$, matrix $\mathbf{Z}^{(t)}$ can be obtained recursively as

$$\mathbf{Z}^{(t)} = \frac{1}{\bar{\eta}_{\mathbf{E}}^{(t)}} \left[\eta_{\mathbf{E}}^{(t)} \mathbf{E}^{(t-1)} + \bar{\eta}_{\mathbf{E}}^{(t-1)} \mathbf{Z}^{(t-1)} - \nabla_{\mathbf{E}} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \right]. \tag{189}$$

Then, $\mathbf{E}^{(t)}$ is updated as

$$\mathbf{E}^{(t)} = \text{soft\_th}(\mathbf{Z}^{(t)}; \mu t / \bar{\eta}_{\mathbf{E}}^{(t)}). \tag{190}$$

The overall online algorithm is listed in Table 6.

Remark 1 (Computational complexity). In the batch algorithm of Table 5, the complexity orders for computing the updates for each of $\mathbf{p}_i$ and $\mathbf{q}_i$ are $\mathcal{O}(N_y M T)$ and $\mathcal{O}(N_x M T)$, respectively, due to the computation of $\tilde{\boldsymbol{\mathcal{W}}}^{\mathcal{T}} \tilde{\mathbf{s}}_i$

and $\breve{\boldsymbol{\mathcal{W}}}_i^{\mathcal{T}} \breve{\mathrm{s}}_i$. Thus, the complexity orders for updating $\mathbf{P}$ and $\mathbf{Q}$ per iteration k are both $\mathcal{O}(N_x N_y MT)$. The update of e_l incurs complexity $\mathcal{O}(MT)$ for computing $\boldsymbol{\omega}_l^{\mathcal{T}} \check{\breve{\mathrm{s}}}_l$. Thus, the complexity order for updating $\mathbf{E}$ per iteration k is $\mathcal{O}(N_x N_y MT)$. Accordingly, the overall per-iteration complexity of the batch algorithm becomes $\mathcal{O}(N_x N_y MT)$. On the other hand, the complexity of the online algorithm in Table 6 is dominated by the gradient computations, which require $\mathcal{O}(\rho N_x N_y M)$. Since ρ is smaller than N_x and N_y, and the per-iteration complexity does not grow with T, the online algorithm has a much more affordable complexity than its batch counterpart, and it is scalable for large network scenarios.

Convergence

The iterates $\{\mathbf{X}^{(t)}\}_{t=1}^{\infty}$ generated from the algorithm in Table 6 converge to a stationary point of (P2), as asserted in the following proposition. First define

$$C_t(\mathbf{X}) := \frac{1}{t} \sum_{\tau=1}^{t} \left[g_1(\mathbf{X}, \boldsymbol{\xi}^{(\tau)}) + g_2(\mathbf{X}) \right] \tag{191}$$

$$\check{C}_t(\mathbf{X}) := \frac{1}{t} \sum_{\tau=1}^{t} \left[\check{g}_1(\mathbf{X}, \mathbf{X}^{(\tau-1)}, \boldsymbol{\xi}^{(\tau)}) + g_2(\mathbf{X}) \right] \tag{192}$$

$$C(\mathbf{X}) := \mathbb{E}_{\boldsymbol{\xi}} \left[g_1(\mathbf{X}, \boldsymbol{\xi}) + g_2(\mathbf{X}) \right]. \tag{193}$$

Note that $C_t(\mathbf{X})$ is essentially identical to the cost of (P2). Furthermore, the minimizer of $C_t(\mathbf{X})$ approaches that of $C(\mathbf{X})$ when $t \to \infty$, provided $\boldsymbol{\xi}$ obeys the law of large numbers, which is clearly the case when e.g., $\{\boldsymbol{\xi}^{(t)}\}$ is i.i.d. Assume that $\nabla_{\mathbf{P}} g_1(\cdot, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi})$, $\nabla_{\mathbf{Q}}(\mathbf{P}, \cdot, \mathbf{E}, \boldsymbol{\xi})$ and $\nabla_{\mathbf{E}}(\mathbf{P}, \mathbf{Q}, \cdot, \boldsymbol{\xi})$ are Lipschitz with respect to $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{E}$, respectively, with constants $L_{\mathbf{P}}$, $L_{\mathbf{Q}}$, and $L_{\mathbf{E}}$, respectively (which will be shown in Appendix H). Furthermore, let $\bar{\alpha}_i^{(t)} := (\sum_{\tau=1}^{t} (\eta_i^{(\tau)} + \lambda))^{-1}$ for $i \in \{\mathbf{P}, \mathbf{Q}\}$, and $\bar{\alpha}_{\mathbf{E}}^{(t)} := (\bar{\eta}_{\mathbf{E}}^{(t)})^{-1}$ denote step sizes.

Proposition 8: *If* (a1) $\{\boldsymbol{\xi}^{(t)}\}_{t=1}^{\infty}$ *is an independent and identically distributed (i.i.d) random sequence;* (a2) $\{\mathbf{X}^{(t)}\}_{t=1}^{\infty}$ *are in a compact set* $\mathcal{X}$*;* (a3) $\mathit{\Xi}$ *is bounded;* (a4) *For* $i \in \{\mathbf{P}, \mathbf{Q}, \mathbf{E}\}$, $\bar{\eta}_i^{(t)} \geq ct \; \forall t$ *for some* $c \geq 0$*; and* (a5) $c' \geq \eta_i^{(t)} \geq L_i^2 / L_{\min} \; \forall t$ *for some* $c' > 0$ *and* $L_{\min} := \min\{L_{\mathbf{P}}, L_{\mathbf{Q}}, L_{\mathbf{E}}\}$, *then the iterates* $\{\mathbf{X}^{(t)}\}_{t=1}^{\infty}$ *generated by the algorithm in Table 6 converge to* *the set of stationary points of* (P2) *with* $\beta = 1$, *i.e.,*

$$\lim_{t\to\infty} \inf_{\bar{\mathbf{X}}\in\bar{\mathcal{X}}} \|\mathbf{X}^{(t)} - \bar{\mathbf{X}}\|_F = 0 \qquad \text{a.s.} \tag{194}$$

where $\bar{\mathcal{X}}$ *is the set of stationary points of* $C(\mathbf{X})$.
Proof: See Appendix H.

Numerical Tests

Performance of the presented batch and online algorithms is assessed through numerical tests using both synthetic and real datasets. A few existing methods are also tested for comparison. The ridge-regularized least-squares (LS) scheme estimates the SLF as $\text{vec}(\hat{\mathbf{F}}) = (\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{W}}^{\mathcal{T}} + \omega\mathbf{C}_{\mathbf{f}}^{-1})^{-1}\boldsymbol{\mathcal{W}}\check{\mathbf{s}}$, where $\mathbf{C}_{\mathbf{f}}$ is the spatial covariance matrix of the SLF, and ω is a regularization parameter [297, 155, 142]. The total variation (TV)-regularized LS scheme in [236] is also tested, which solves $\min_{\mathbf{f}} \|\check{\mathbf{s}} - \boldsymbol{\mathcal{W}}^{\mathcal{T}}\mathbf{f}\|_2^2 + \omega\big(\sum_{i=1}^{N_x-1}\sum_{j=1}^{N_y}|f_{i+1,j} - f_{i,j}| + \sum_{i=1}^{N_x}\sum_{j=1}^{N_y-1}|f_{i,j+1} - f_{i,j}|\big)$ where $\mathbf{f} := \text{vec}(\mathbf{F})$ and $f_{i,j}$ corresponds to the (i,j)-th element of $\mathbf{F}$. Finally, the LASSO estimator is obtained by solving (P1) with $\lambda = 0$.

Test with Synthetic Data

Random tomographic measurements were taken by sensors deployed uniformly over $\mathcal{A} := [0.5, 40.5] \times [0.5, 40.5]$, from which the SLF with $N_x = N_y = 40$ was reconstructed. Per-time slot, 10 measurements were taken, corrupted by zero-mean white Gaussian noise with variance $\sigma^2 = 0.1$. The regularization parameters were set to $\lambda = 0.05$ and $\mu = 0.01$ through cross-validation by minimizing the normalized error $\|\hat{\mathbf{F}} - \mathbf{F}_0\|_F/\|\mathbf{F}_0\|_F$, where $\mathbf{F}_0$ is the ground-truth SLF depicted in Fig. 36. Other parameters were set to $\rho = 13$, $\beta = 1$, and $\delta = 0.06$; while $\mathbf{C}_{\mathbf{f}} = \mathbf{I}_{N_xN_y}$ and $\omega = 0.13$ were used for the ridge-regularized LS.

To validate the batch algorithm in Table 5, two cases were tested. In the first case, the measurements were generated for $T = 130$ time slots using $N = 52$ sensors, while in the second case, $T = 260$ and $N = 73$ were used. As a comparison, the accelerated proximal gradient (APG) algorithm was also derived for (P1) [187]. Note that the APG requires the costly SVD operation of an N_x-by-N_y matrix per iteration, while only the inversion of a ρ-by-ρ

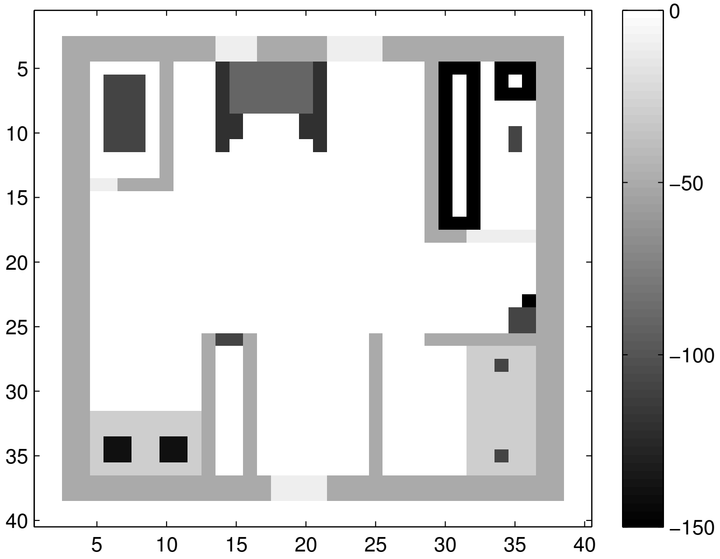
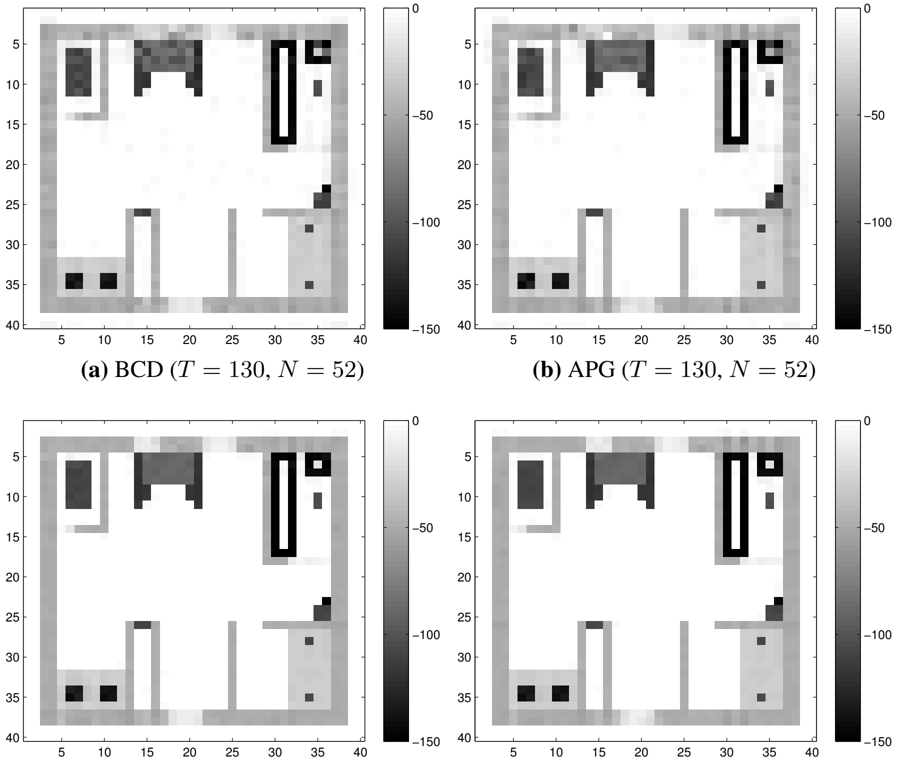


Figure 36: True SLF.

matrix is necessary in the proposed BCD algorithm. Fig. 37 shows the SLFs reconstructed by APG and BCD algorithms for the two cases. Apparently, the reconstructed SLFs capture well the features of the ground-truth SLF in Fig. 36. Note that (P2) is underdetermined when $T = 130$ since the total number of unknowns in (P2) is $2,640$ while the total number of measurements is only $1,300$. This verifies that the channel gain maps can be accurately interpolated with a small number of measurements by leveraging the attributes of the low rank and sparsity. Fig. 38a shows the convergence of the BCD and APG algorithms. The cost of (P2) from the BCD algorithm converges to that of (P1) from APG after $k = 550$ iterations, showing that the performance of solving (P1) directly is achievable by the proposed algorithm solving (P2) instead. This can also be corroborated from the reconstructed SLFs in Fig. 37 as well.

Table 7 lists the reconstruction error when $T = 130$ and the per-iteration complexity of the batch algorithms. It is seen that the proposed method outperforms benchmark algorithms in terms of the reconstruction error. Note that the ridge-regularized LS has a one-shot (non-iterative) complexity of



(a) BCD ($T = 130$, $N = 52$)

(b) APG ($T = 130$, $N = 52$)

(c) BCD ($T = 260$, $N = 73$)

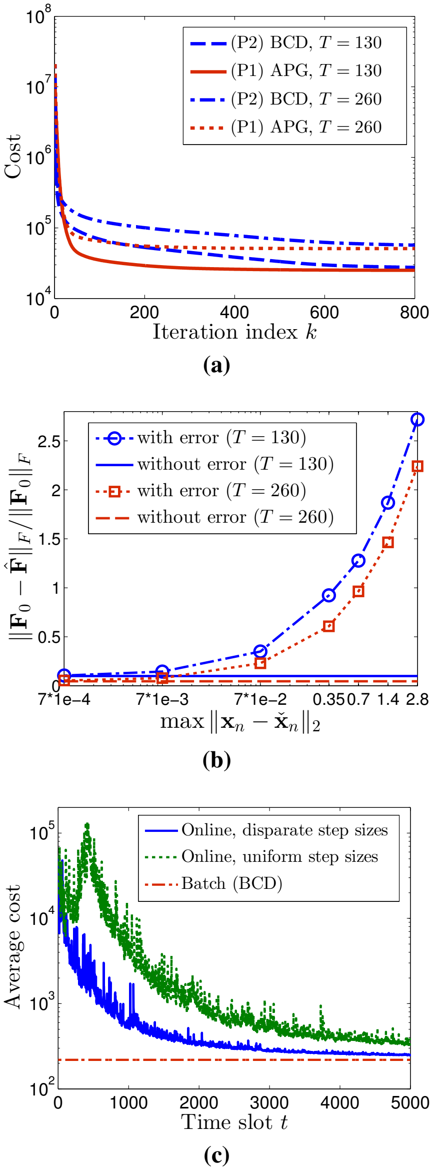
(d) APG ($T = 260$, $N = 73$)

Figure 37: SLFs reconstructed by the batch algorithms.

$\mathcal{O}((N_x N_y)^3)$, but its reconstruction capability is worse than the proposed algorithm as the true SLF is not smooth.

To test robustness of the proposed algorithm against imprecise CR location estimates, the reconstruction error versus the maximum sensor location error is depicted in Fig. 38b. To reconstruct $\mathbf{F}$ matrix, $\boldsymbol{\mathcal{W}}$ was computed via a set of erroneous sensor locations $\check{\mathbf{x}}_n^{(t)}$ obtained by adding uniformly random perturbations to true locations $\mathbf{x}_n^{(t)}$. It is seen that the SLF could be accurately reconstructed when the location error was small.

The numerical tests for the online algorithm were carried out with the same parameter setting as the batch experiments with $N = 317$. Fig. 38c depicts the evolution of the average cost in (191) for two sets of values for $(\bar{\eta}_{\mathbf{P}}^{(t)}, \bar{\eta}_{\mathbf{Q}}^{(t)}, \bar{\eta}_{\mathbf{E}}^{(t)})$.



(a)

(b)

(c)

Figure 38: SLF reconstruction using the batch and online algorithms. (a) Cost versus iterations (batch). (b) Reconstruction error versus CR location error (batch). (c) Average cost over time slots (online).

Table 7: Reconstruction error at $T = 130$ and computational complexity per iteration.

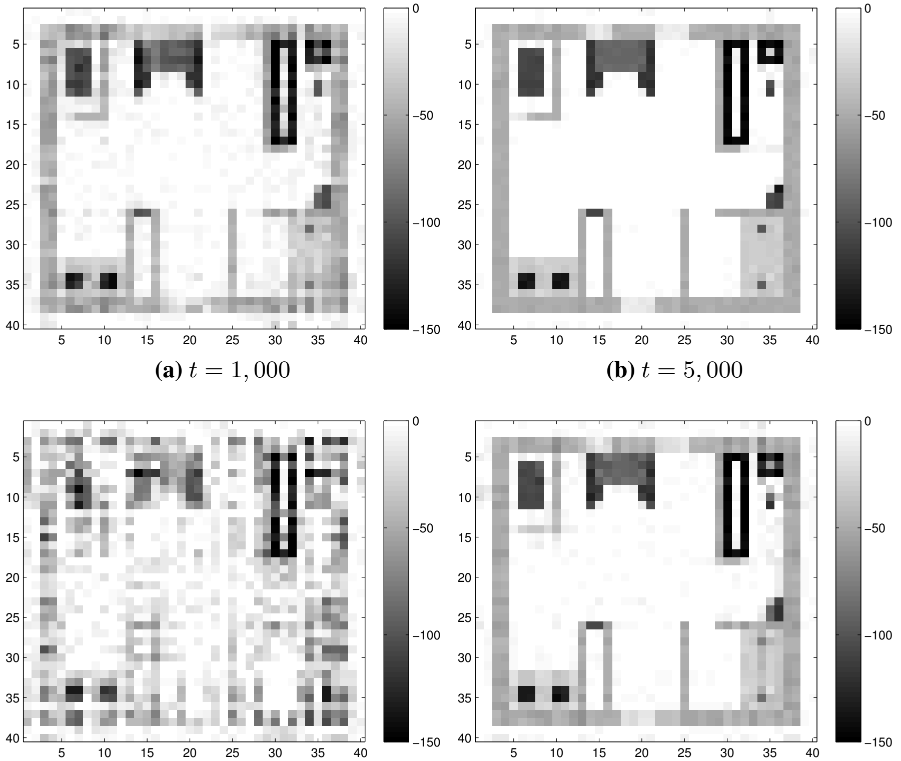
Algorithm	Proposed (BCD)	Ridge-regularized LS	Total variation (ADMM)	LASSO
$\|\mathbf{F}_0 - \hat{\mathbf{F}}\|_F / \|\mathbf{F}_0\|_F$	0.1064	0.1796	0.1196	0.1828
Complexity per iteration	$\mathcal{O}(N_x N_y MT)$	N/A	$\mathcal{O}((N_x N_y)^3 + (N_x N_y)^2 MT)$	$\mathcal{O}(N_x N_y MT)$

The green dotted curve corresponds to using $\bar{\eta}_{\mathbf{P}}^{(t)} = \bar{\eta}_{\mathbf{Q}}^{(t)} = \bar{\eta}_{\mathbf{E}}^{(t)} = 300$, while the blue solid curve is for $\bar{\eta}_{\mathbf{P}}^{(t)} = \bar{\eta}_{\mathbf{Q}}^{(t)} = 300$, and $\bar{\eta}_{\mathbf{E}}^{(t)} = 10$. It can be seen that the uniform step sizes for all variables result in convergence rate that is slower than that with the disparate step sizes. Fig. 39 shows the SLFs reconstructed via the online algorithm at $t = 1,000$ and $t = 5,000$ using the two choices of step sizes. It can be seen that for a given time slot t, flexibly choosing the step sizes yields much more accurate reconstruction. As far as reconstruction error, the online algorithm with disparate step sizes yields 6.3×10^{-2} at $t = 5,000$, while its batch counterpart has 2.4×10^{-2}. Although slightly less accurate SLF is obtained by the online algorithm, it comes with greater computational efficiency.

To assess the tracking ability of the online algorithm, the slow channel variation was simulated. The measurements were generated using the SLF in Fig. 36 with three additional objects slowly moving in the rate of unit pixel width per 70 time slots. Fig. 40 depicts instances of the true and reconstructed SLFs at $t = 2,400$ and $t = 3,200$, respectively, obtained by the online algorithm. The moving objects are marked by the red circles. It is seen that the reconstructed SLFs correctly capture the moving objects, while the stationary objects are estimated more clearly as t increases.

Test with Real Data

To validate the performance of the proposed framework for SLF and channel gain map estimation in realistic scenarios, real received signal strength (RSS) measurements were also processed. The data were collected by a set of $N = 20$ sensors deployed in the perimeter of a square-shaped testbed as shown in Fig. 41, where the crosses indicate the sensor positions. Data collection was performed in two steps [142]. First, free-space measurements were taken to



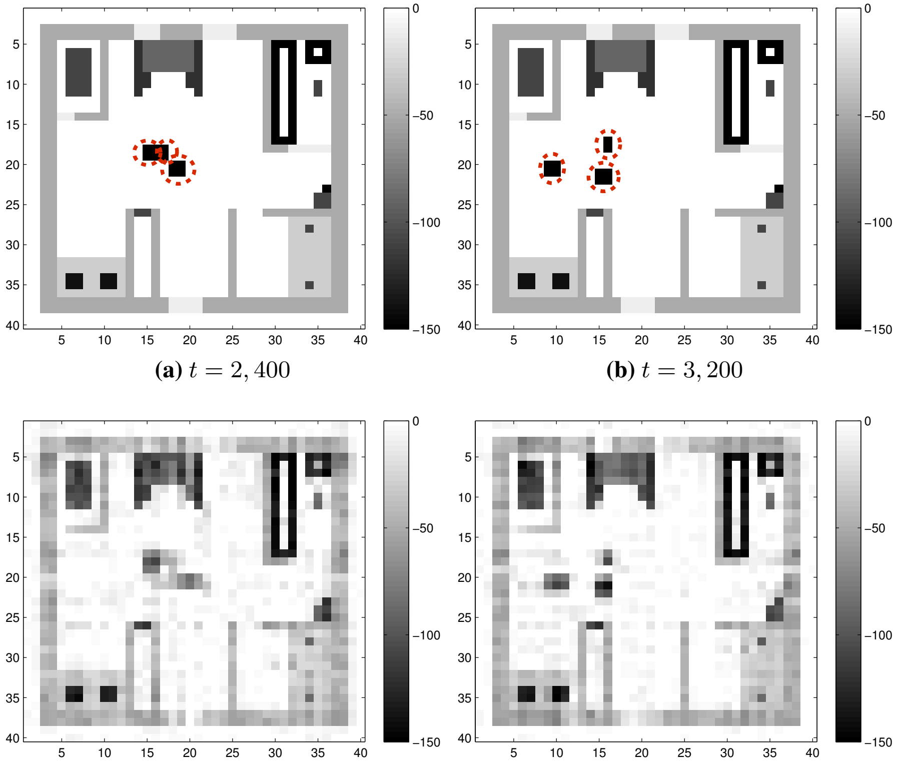
(a) $t = 1,000$ **(b)** $t = 5,000$

(c) $t = 1,000$ **(d)** $t = 5,000$

Figure 39: SLFs reconstructed by the online algorithm. (a) and (b) correspond to using $\bar{\eta}_{\mathbf{P}}^{(t)} = \bar{\eta}_{\mathbf{Q}}^{(t)} = 300$ and $\bar{\eta}_{\mathbf{E}}^{(t)} = 10$. (c) and (d) use $\bar{\eta}_{\mathbf{P}}^{(t)} = \bar{\eta}_{\mathbf{Q}}^{(t)} = \bar{\eta}_{\mathbf{E}}^{(t)} = 300$.

obtain estimates of the path gain G_0 and the pathloss exponent γ via least-squares. The estimated γ was approximately 2, and G_0 was found to be 75. Then, tomographic measurements were formed with the artificial structure shown in Fig. 41. For the both measurements, 100 measurements were taken per time slot, in the 2.425 GHz frequency band, across 24 time slots. The shadowing measurements were obtained by subtracting the estimated pathloss from the RSS measurements.

The SLFs of size $N_x = N_y = 61$ were reconstructed by the proposed batch algorithm. The regularization parameters were set to $\lambda = 4.5$ and $\mu = 3.44$, which were determined by cross-validation. The parameter δ in (155) was set



(a) $t = 2,400$ **(b)** $t = 3,200$

(c) $t = 2,400$ **(d)** $t = 3,200$

Figure 40: (a) and (b) are true SLFs; (c) and (d) show reconstructed SLFs at different time slots.

to 0.2 feet to capture the non-zero weights within the first Fresnel zone, and $\rho = 10$ and $\beta = 1$ were used.

For comparison, the ridge-regularized LS estimator was also tested. To construct $\mathbf{C_f}$, the exponential decay model in [1] was used, which models the covariance between points $\mathbf{x}$ and $\mathbf{x}'$ as $\mathbf{C_f}(\mathbf{x}, \mathbf{x}') = \sigma_s^2 e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2}{\kappa}}$, where σ_s^2 and $\kappa > 0$ are model parameters. In our tests, $\sigma_s^2 = \kappa = 1$, and $\omega = 79.9$ were used.

The SLF, shadow fading map, and channel gain map reconstructed by the proposed BCD algorithm are depicted in Fig. 42. The shadow fading and channel gain maps portray the gains in dB between any point in the map and the

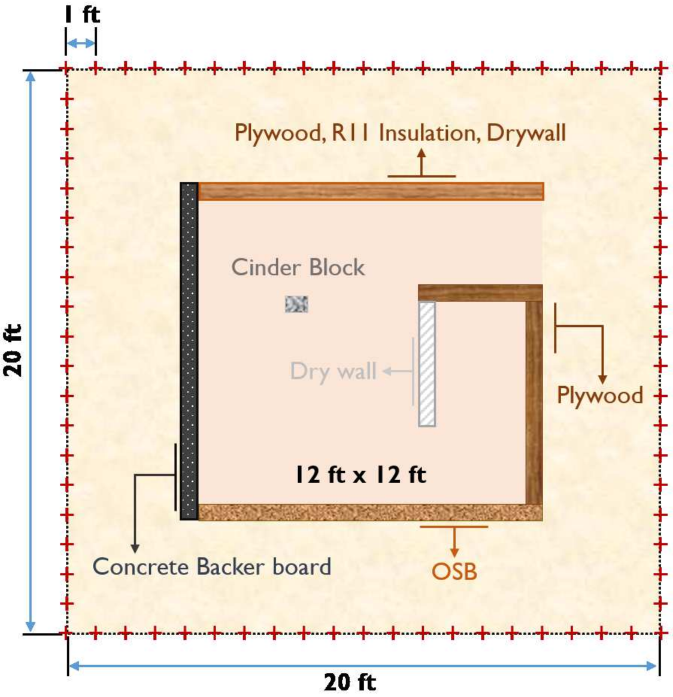
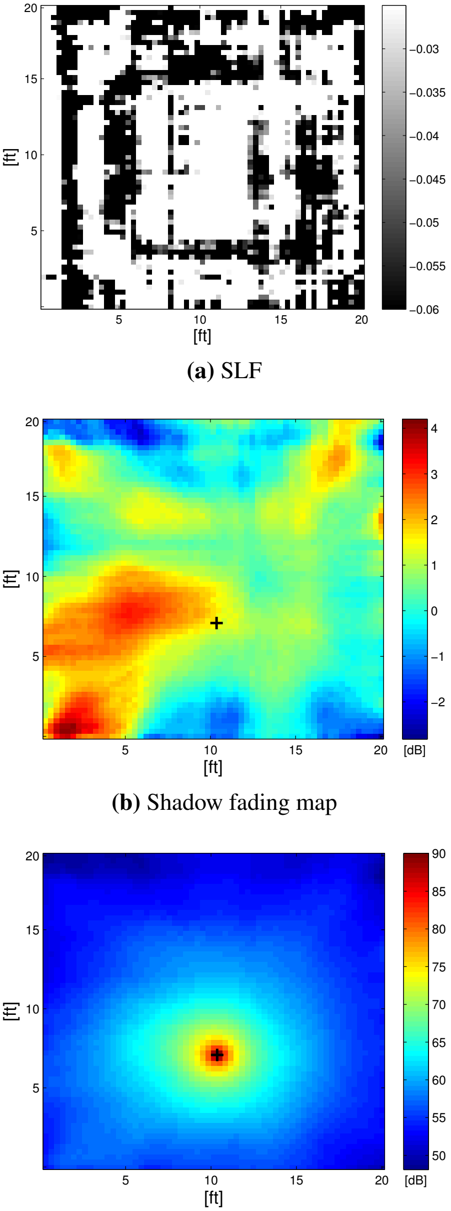


Figure 41: Configuration of the testbed.

fixed CR location at $(10.2, 7.2)$ (marked by the cross). Fig. 43 shows the results from the ridge-regularized LS estimation. It can be seen from Fig. 42(a) and Fig. 43(b) that the proposed low-rank plus sparse model produces a somewhat sharper SLF image than the ridge-regularized LS approach. Although the latter yields a smooth SLF image, it produces more artifacts near the isolated block and the boundary of the SLF. Such artifacts may lead to less accurate shadowing and channel gain maps. For instance, Fig. 42(b) and Fig. 43(b) both show that the shadow fading is stronger as more building material is crossed in the communication path. However, somewhat strong attenuations are observed near the cinder block location and the interior of the oriented strand board (OSB) wall only in Fig. 43(b), which seems anomalous.

The online algorithm was also tested with the real data. Parameters $\bar{\eta}_{\mathbf{P}}^{(t)} = \bar{\eta}_{\mathbf{Q}}^{(t)} = 620$ and $\bar{\eta}_{\mathbf{E}}^{(t)} = 200$ were selected, and 6×10^5 measurements were

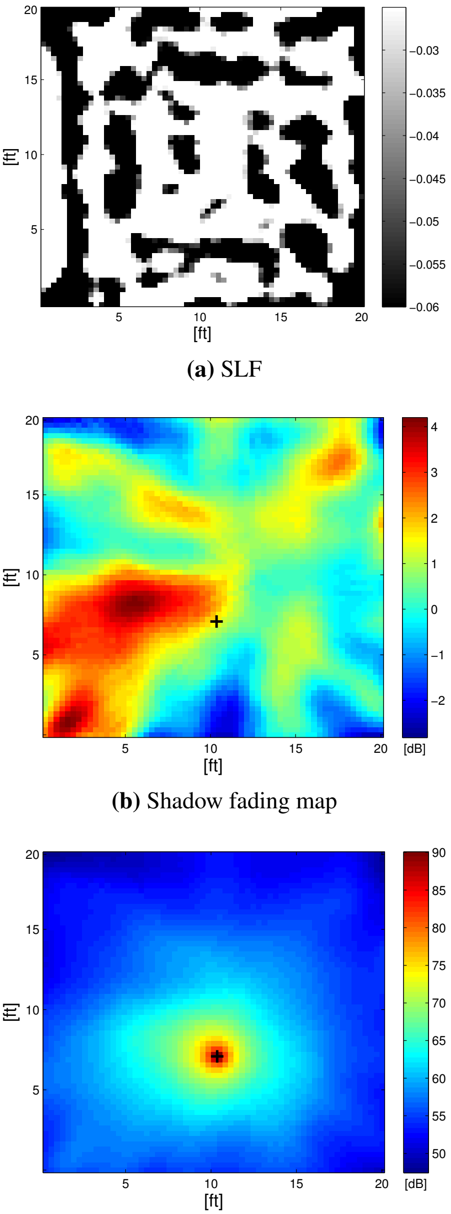


(a) SLF

(b) Shadow fading map

(c) Channel gain map

Figure 42: Reconstructions by the proposed batch algorithm.

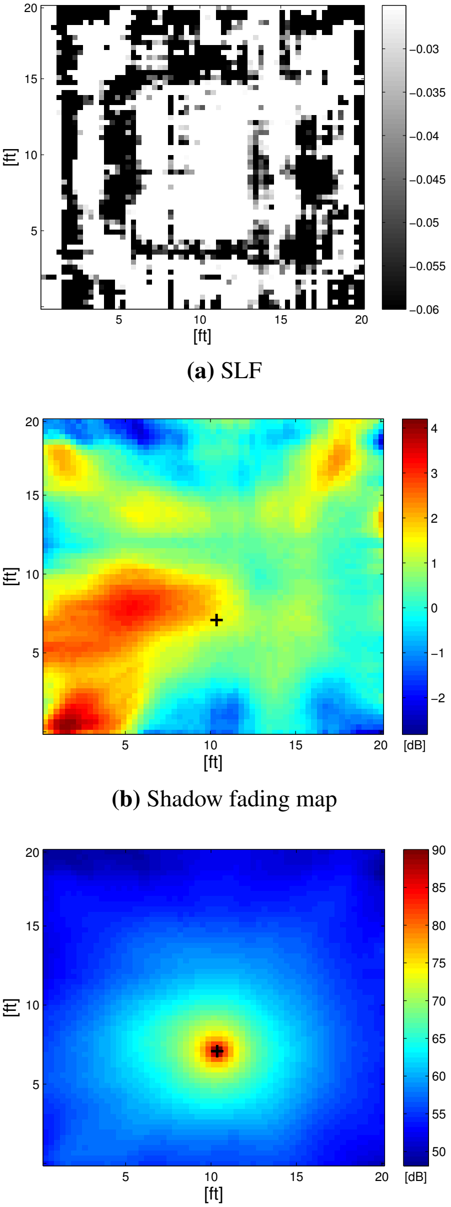


(a) SLF

(b) Shadow fading map

(c) Channel gain map

Figure 43: Reconstructions by the ridge-regularized LS.

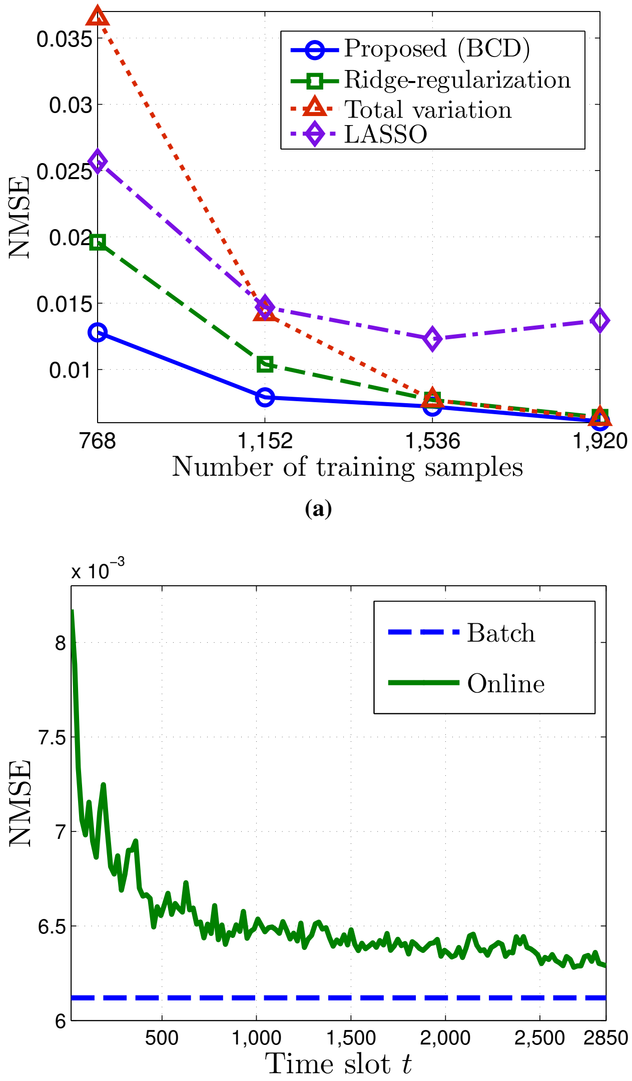


(a) SLF

(b) Shadow fading map

(c) Channel gain map

Figure 44: Reconstructions by the proposed online algorithm.



(a)

(b)

Figure 45: NMSE of channel gain prediction by (a) the batch; and (b) online algorithms.

uniformly drawn from the original dataset with replacement to demonstrate the asymptotic performance. Fig. 44 depicts the reconstructed SLF, shadow fading and channel gain maps obtained from the online algorithm. It can be seen that the SLF shown in Fig. 44a is close to that depicted in Fig. 42a. Similar observations can be made for the shadow fading and channel gain maps as well. Thus, the online algorithm is a viable alternative to the batch algorithm with reduced computational complexity, and affordable memory requirement. Channel gain estimation performance of the proposed algorithms was assessed via 5-fold cross-validation. Let $\check{\mathbf{g}}_{\text{test}}$ and $\hat{\mathbf{g}}_{\text{test}}$ denote RSS measurement vectors in the test set and its estimate, respectively. Prediction performance is measured by the normalized mean-square error (NMSE) $\|\check{\mathbf{g}}_{\text{test}} - \hat{\mathbf{g}}_{\text{test}}\|^2/\|\check{\mathbf{g}}_{\text{test}}\|^2$. Fig. 45a displays the NMSE of batch algorithms with 480 test samples versus the number of training samples. It is shown that the proposed algorithm outperforms competing alternatives, particularly when a small number of training samples are available, validating the usefulness of the proposed model. The online algorithm was also tested with 2.85×10^5 measurements uniformly drawn from $1,920$ training samples with replacement. Fig. 45b depicts the evolution of the NMSE measured on 480 test samples at every t. It is observed that the online algorithm attains the batch performance as t increases.

Conclusions

A low-rank plus sparse matrix model was proposed for channel gain cartography, which is instrumental for various CR spectrum sensing and resource allocation tasks. The channel gains were modeled as the sum of the distance-based pathloss and the tomographic accumulation of shadowing due to the underlying SLF. The SLF was postulated to have a low-rank structure corrupted by sparse outliers. Efficient batch and online algorithms were developed by leveraging a bifactor-based characterization of the matrix nuclear norm. The algorithms enjoy low computational complexity and a reduced memory requirement, without sacrificing the optimality, with provable convergence properties. Tests with both synthetic and real measurement datasets corroborated the claims and showed that the algorithms could accurately reveal the structure of the propagation medium.

9

Exploiting Sparse User Activity in Multiuser Detection

This section focuses on efficient sparsity-utilizing multiuser detectors in code-division multiple access (CDMA) systems. Relying on the fact that the number of *active* users in CDMA systems is often much lower than the spreading gain, the presented approach fruitfully exploits this *a priori* information to improve performance of multiuser detectors. A low-activity factor manifests itself in a sparse symbol vector with entries drawn from a finite alphabet that is augmented by the zero symbol to capture user inactivity. The non-equiprobable symbols of the augmented alphabet motivate a sparsity-exploiting maximum *a posteriori* probability (S-MAP) criterion, which is shown to yield a cost comprising the ℓ_2 least-squares error penalized by the p-th norm of the wanted symbol vector ($p = 0, 1, 2$). Related optimization problems appear in variable selection (shrinkage) schemes developed for linear regression, as well as in the emerging field of CS. The contribution of this work to such *sparse* CDMA systems is a gamut of sparsity-exploiting multiuser detectors trading off performance for complexity requirements. From the vantage point of CS and the Lasso spectrum of applications, the contribution amounts to sparsity-exploiting algorithms when the entries of the wanted signal vector adhere to finite-alphabet constraints.

Related Works

Multiuser detection (MUD) algorithms play a major role for mitigating multi-access interference present in CDMA systems; see e.g., [288] and references therein. These well-appreciated MUD algorithms simultaneously detect the transmitted symbols of all active user terminals. However, they require knowledge of which terminals are active, and exploit no possible user (in)activity. In this work, MUD algorithms are developed when the active terminals are unknown and the activity factor (probability of each user being active) is low – a typical scenario in tactical or commercial CDMA systems deployed. The inactivity per user can be naturally incorporated by augmenting the underlying alphabet with an extra zero constellation point. Low activity thus implies a sparse symbol vector to be detected. With non-equiprobable symbols in the augmented alphabet, the optimal sparsity-embracing MUD naturally suggests a maximum *a posteriori* (MAP) criterion for detection. Sparse sphere decoding has been considered in, e.g., [277]. Sparsity has been used for estimating parameters of communication systems in, e.g., [53, 23, 9, 109], but not for multiuser detection.

Modeling and Problem Statement

Consider the uplink of a CDMA system with K user terminals and spreading gain N. Assume first that $N \geq K$. The under-determined case ($N < K$) will also be addressed later on in Section 9. Suppose the system has a relatively low activity factor, which analytically means that each terminal is active with probability (w.p.) $p_a < 1/2$ per symbol, and the events "*active*" are independent across symbols and across users. The case of correlated (in)activity of users across symbols will be dealt with in Section 9. Let $b_k \in \mathcal{A}$ denote the symbol, drawn from a finite alphabet by the k-th user, when active; otherwise, $b_k = 0$. Incorporating possible (in)activity per user is equivalent to having b_k take values from an *augmented* alphabet $\mathcal{A}_a := \mathcal{A} \bigcup \{0\}$.

The access point (AP) receives the superimposed modulated (quasi-) synchronous signature waveforms through (possibly frequency-selective) fading channels in the presence of additive white Gaussian noise (AWGN); and projects on the orthonormal space spanned by the aggregate waveforms to obtain the received chip samples collected in the $N \times 1$ vector $\mathbf{y}$. With the

$K \times 1$ vector $\mathbf{b}$ containing the symbols of all (active and inactive) users, the canonical input-output relationship is, see e.g., [288, Sec. 2.9]

$$\mathbf{y} = \mathbf{H}\mathbf{b} + \mathbf{w} \tag{195}$$

where $\mathbf{H}$ is an $N \times K$ matrix capturing transmit-receive filters, spreading sequences, channel impulse responses, and timing offsets; and the $N \times 1$ vector $\mathbf{w}$ is the AWGN. Without loss of generality (w.l.o.g.), $\mathbf{w}$ can be scaled to have unit variance. Note that (195) holds for quasi-synchronous systems too, where relative user asynchronism is bounded to a few chips per symbol, provided that: either i) user transmissions include guard bands to eliminate inter-symbol interference (ISI); or ii) the received vector $\mathbf{y}$ collects only the chips of each user that belong to the part of the common symbol interval under consideration.

The low activity factor implies that $\mathbf{b}$ is a sparse vector. However, the AP is neither aware of the positions nor the number of zero entries in $\mathbf{b}$. In order to perform multiuser detection (MUD) needed to determine the optimal $\hat{\mathbf{b}}$, the AP must account for the augmented alphabet $\mathcal{A}_a$, i.e., consider all the possible candidate vectors $\mathbf{b} \in \mathcal{A}_a^K$. This way, the MUD also determines the k-th user's activity captured by the extra constellation point $b_k = 0$. Supposing that the AP has the channel matrix $\mathbf{H}$ available (e.g., via training), the goal of this paper is to detect the optimal $\hat{\mathbf{b}}$ given the received vector $\mathbf{y}$ by exploiting the sparsity of active users.

To motivate this sparsity-exploiting MUD setup in the CDMA context, consider a set of terminals wishing to link with a common AP. Suppose that the AP acquires the full matrix $\mathbf{H}$ (with all terminals active) during a training phase. Those channels may include either non-dispersive or multipath fading, and are assumed invariant during the coherence time, which is typically larger compared to the symbol period. Each terminal accesses the channel randomly, and the AP receives the superposition of signals from the active terminals only. The AP is interested in determining both the active terminals and the symbols transmitted.

Another scenario where $\mathbf{H}$ is known and sparsity-exploiting MUD is well motivated, entails an unmanned aerial vehicle (UAV) collecting information from a ground wireless sensor network (WSN) placed over a grid, as depicted in Fig. 46. As the UAV flies over the grid of sensors, it collects the signals from a random subset of them. If the channel fading is predominantly affected by

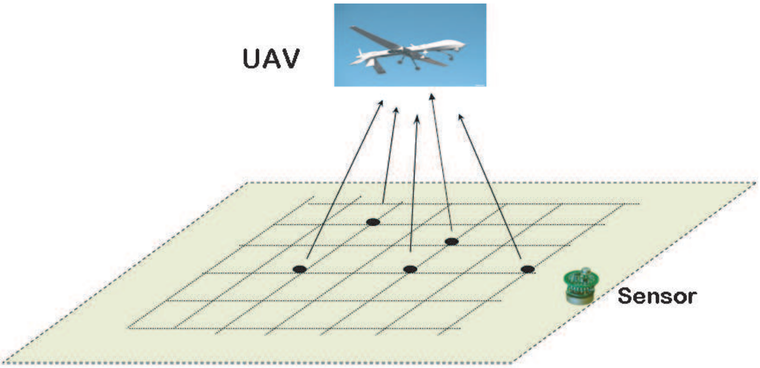


Figure 46: Wireless sensors access a UAV.

path loss, the UAV can acquire $\mathbf{H}$ based on the relative AP-to-sensor distances. Again, the UAV faces the problem of determining both identities of active sensors, and the symbols each active sensor transmits.

With this problem setup in mind, we will develop different MUD strategies, which account for the low activity factor. First, we will look at the maximum *a posteriori* probability (MAP) optimum MUD that exploits the sparsity present.

Sparsity-Exploiting MAP Detector

The goal is to detect $\mathbf{b}$ in (195), given a prescribed activity factor, the received vector $\mathbf{y}$, and the matrix $\mathbf{H}$. Recall though that the low activity factor leads to a sparse $\mathbf{b}$, i.e., each entry b_k is more likely to take the value 0 from the alphabet. Because entries $\{b_k\}_{k=1}^K$ are non-equiprobable, the optimal detector in the sense of minimizing the detection error rate is the MAP one.

Aiming at a sparsity-aware MAP criterion, consider first the prior probability for $\mathbf{b}$. For simplicity in exposition, suppose for now that each terminal transmits binary symbols when active, i.e., $\mathcal{A} = \{\pm 1\}$. (It will become clear later on that all sparsity-cognizant MUD schemes are applicable to finite alphabets of general constellations, not necessarily binary.) If b_k takes values from $\{-1, 0, 1\}$, with corresponding probabilities $\{p_a/2, 1 - p_a, p_a/2\}$, and since each entry b_k is independent from $b_{k'}$ for $k \neq k'$, the prior probability for $\mathbf{b}$

can be expressed as

$$\Pr(\mathbf{b}) = \prod_{k=1}^{K} \Pr(b_k) = (1 - p_a)^{K - \|\mathbf{b}\|_0} (p_a/2)^{\|\mathbf{b}\|_0} \tag{196}$$

where $\|\mathbf{b}\|_0$ denotes the ℓ_0 (pseudo) norm that is by definition equal to the number of non-zero entries in the vector $\mathbf{b}$. Upon taking logarithms, (196) yields

$$\ln \Pr(\mathbf{b}) = -\lambda \|\mathbf{b}\|_0 + K \ln(1 - p_a) \tag{197}$$

where

$$\lambda := \ln \frac{1 - p_a}{p_a/2}. \tag{198}$$

Since low activity factor means $p_a < 1/2$, it follows readily from (198) that $\lambda > 0$. With the prior distribution of $\mathbf{b}$ in (197), the sparsity-aware MAP (S-MAP) detector is

$$\begin{aligned} \hat{\mathbf{b}}^{\text{MAP}} &= \arg \max_{\mathbf{b} \in \mathcal{A}_a^K} \Pr(\mathbf{b}|\mathbf{y}) = \arg \min_{\mathbf{b} \in \mathcal{A}_a^K} -\ln p(\mathbf{y}|\mathbf{b}) - \ln \Pr(\mathbf{b}) \\ &= \arg \min_{\mathbf{b} \in \mathcal{A}_a^K} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_0 \end{aligned} \tag{199}$$

where the last equality follows from (197) and the Gaussianity of $\mathbf{w}$. Hence, the S-MAP detection task amounts to finding the vector in the constraint set $\mathcal{A}_a^K$, which minimizes the cost in (199).

Two remarks are now in order.

Remark 4: *(General constellations).* Beyond binary alphabets, it is easy to see why the S-MAP detector in (199) applies to more general constellations, including pulse amplitude modulation (PAM), phase-shift keying (PSK), and quadrature amplitude modulation (QAM). Specifically, for general M-ary constellations with $M \geq 2$ it suffices to adjust accordingly the prior probability as a function of $\|\mathbf{b}\|_0$ in (196). This will render the S-MAP MUD in (199) applicable to general M-ary constellations, provided that λ in (198) is replaced by $\lambda := \ln \frac{1-p_a}{p_a/M}$.

Remark 5: *(Scale λ as a function of p_a).* The definition in (198) reveals the explicit relationship of λ with the activity factor p_a. Different from CS and VS approaches, where λ is a tuning parameter often chosen with cross-validation techniques as a function of the data size N and K, here it is directly coupled with the user activity factor p_a. Such a coupling carries over even when users have distinct activity factors. Specifically, if the user k is active w.p. $p_{a,k}$, then the term $\lambda\|\mathbf{b}\|_0 := \lambda\sum_{k=1}^{K}|b_k|$ in (199) should change to $\sum_{k=1}^{K}\lambda_k|b_k|$, where λ_k is defined as in (198) with $p_{a,k}$ substituting p_a. This user-specific regularization will be used in Section 9 to adaptively estimate user activity factors on-the-fly, and thus enable sparsity-aware MUD which accounts for correlated user (in)activity across the time slots.

With the variable b_k only taking values from $\{\pm 1, 0\}$, it holds for $p \geq 1$ that

$$\|\mathbf{b}\|_0 = \sum_{k=1}^{K}|b_k|^p = \|\mathbf{b}\|_p^p, \ \ \forall\, \mathbf{b} \in \mathcal{A}_a^K. \tag{200}$$

Hence, the S-MAP detector (199) for binary transmissions is equivalent to

$$\hat{\mathbf{b}}^{\text{MAP}} = \arg\min_{\mathbf{b}\in\mathcal{A}_a^K} \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{b}\|_2^2 + \lambda\|\mathbf{b}\|_p^p\,, \ \ \forall\, p \geq 1. \tag{201}$$

Notice that the equivalence between (199) and (201) is based on the norm equivalence in (200), which holds only for constant modulus constellations. Although the cost in (201) will turn out to be of interest on its own, it is not an S-MAP detector for non-constant modulus constellations.

Interestingly, since low-activity factor implies a positive λ, the problem (201) entails a convex cost function, compared to the non-convex one in (199). In lieu of the finite-alphabet constraint, the criterion of (201) consists of the least-squares (LS) cost regularized by the ℓ_p norm, which in the context of linear regression has been adopted to mitigate data overfitting. In contrast, the LS estimator only considers the goodness-of-fit, and thus tends to overfit the data. Shrinking the LS estimator, by penalizing its size through the ℓ_p norm, typically outperforms LS in practice. For example, the Lasso adopts the ℓ_1 norm through which it effects sparsity[278]. In the MUD context for CDMA systems with low-activity factor, the vector $\mathbf{b}$ has a sparse structure, which motivates well this regularizing strategy. What is distinct and interesting here is that this penalty-augmented LS approach under finite-alphabet constraints emerges naturally as the logarithm of the prior in the S-MAP detector.

However, the finite-alphabet constraint renders the solution of (201) combinatorially complex. For general $\mathbf{H}$ and $\mathbf{y}$, the solution of (201) requires exhaustive search over all the 3^K feasible points, with the complexity growing exponentially in the problem dimension K. Likewise, for general M-ary alphabets the complexity incurred by (199) is $\mathcal{O}((M+1)^K)$. On the other hand, many (sub-) optimal alternatives are available in the MUD literature; see e.g., [288, Ch. 5-7]. Similarly here, we will develop different (sub-) optimal algorithms to trade off complexity for probability of error performance in sparsity-exploiting MUD alternatives.

Since the exponential complexity of MUD stems from the finite-alphabet constraint, one reduced-complexity approach is to solve the unconstrained convex problem, and then quantize the resultant soft decision to the nearest point in the alphabet. This approach includes the sub-optimal linear MUD algorithms (decorrelating and minimum mean-square error (MMSE) detectors). Another approach is to search over (possibly a subset of) the alphabet lattice directly as in the decision-directed detectors or the sphere decoders [97]. Likewise, it is possible to devise (sub-) optimal algorithms for solving the S-MAP MUD problem (201) along these two categories. First, we will present the sparsity-exploiting MUD algorithms by relaxing the finite-alphabet constraint.

Relaxed S-MAP Detectors

In addition to offering an S-MAP detector for constant modulus constellations, the cost in (201) is convex. Thus, by relaxing the combinatorial constraint, the optimization problem (201) can be solved efficiently by capitalizing on convexity. As mentioned earlier, this problem is similar to the penalty-augmented LS criterion that is used for VS in linear regression, where the choice of p is important for controlling the shrinking effect, that is the degree of sparsity in the solution. Next, we will develop detectors for two choices of p, and compare them in terms of complexity and performance.

Linear Ridge MUD

The choice $p = 2$ is a popular one in statistics, well-known as *Ridge regression*. Its popularity is mainly due to the fact that it can regularize the LS solution while retaining its closed-form expression as a linear function of the data $\mathbf{y}$. A relaxed detection algorithm for S-MAP MUD can be developed accordingly with $p = 2$, what we term Ridge detector (RD). Ignoring the finite-alphabet constraint, the optimal solution of (201) for $p = 2$ takes a linear form

$$\begin{aligned}\mathbf{b}^{\mathrm{RD}} &= \arg\min_{\mathbf{b}} \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{b}\|_2^2 + \lambda\|\mathbf{b}\|_2^2 \\ &= (\mathbf{H}^T\mathbf{H} + 2\lambda\mathbf{I})^{-1}\mathbf{H}^T\mathbf{y}. \end{aligned} \tag{202}$$

In addition to its simplicity, and different from LS, the existence of the inverse in (202) is ensured even for ill-posed or under-determined problems; i.e., when $\mathbf{H}$ is rank deficient or fat ($N < K$). Notice that $\mathbf{b}^{\mathrm{RD}}$ takes a form similar to the linear MMSE multiuser detector, with the parameter λ replacing the noise variance, and connecting the activity factor with the degree of regularization applied to the matrix $\mathbf{H}^T\mathbf{H}$.

Upon quantizing each entry of the soft decision $\mathbf{b}^{\mathrm{RD}}$ with the operator

$$\mathcal{Q}_\theta(x) := \mathrm{sign}(x)\mathbb{1}(|x| \geq \theta) \tag{203}$$

where $\theta > 0$, $\mathrm{sign}(x) = 1(-1)$ with $x > (<)0$, and $\mathbb{1}$ denoting the indicator function, the hard RD is

$$\hat{\mathbf{b}}^{\mathrm{RD}} = \mathcal{Q}_\theta(\mathbf{b}^{\mathrm{RD}}) = \mathcal{Q}_\theta\left((\mathbf{H}^T\mathbf{H} + 2\lambda\mathbf{I})^{-1}\mathbf{H}^T\mathbf{y}\right). \tag{204}$$

Because the detector in (202) is linear, it is possible to express linearly its soft output $\mathbf{b}^{\mathrm{RD}}$ with respect to (w.r.t.) the input symbol vector $\mathbf{b}$. Based on this relationship, one can subsequently derive the symbol error rate (SER) of the hard detected symbols in $\hat{\mathbf{b}}^{\mathrm{RD}}$ as a function of the quantization threshold θ. These steps will be followed next to obtain the performance of the RD.

Performance Analysis

Letting $\check{\mathbf{b}}$ denote the vector transmitted, and substituting $\mathbf{y} = \mathbf{H}\check{\mathbf{b}} + \mathbf{w}$ into (202) yields

$$\mathbf{b}^{\mathrm{RD}} = (\mathbf{H}^T\mathbf{H} + 2\lambda\mathbf{I})^{-1}\mathbf{H}^T(\mathbf{H}\check{\mathbf{b}} + \mathbf{w}) = \mathbf{G}\check{\mathbf{b}} + \mathbf{w}' \tag{205}$$

where $\mathbf{G} := \mathbf{I} - 2\lambda(\mathbf{H}^T\mathbf{H} + 2\lambda\mathbf{I})^{-1}$, and the colored noise $\mathbf{w}' := (\mathbf{H}^T\mathbf{H} + 2\lambda\mathbf{I})^{-1}\mathbf{H}^T\mathbf{w}$ is zero-mean Gaussian with covariance matrix $\mathbf{\Sigma}_{w'} := E\{\mathbf{w}'(\mathbf{w}')^T\} = (\mathbf{H}^T\mathbf{H} + 2\lambda\mathbf{I})^{-2}\mathbf{H}^T\mathbf{H}$.

It follows readily from (205) that the k-th entry of $\mathbf{b}^{\mathrm{RD}}$ satisfies

$$b_k^{\mathrm{RD}} = G_{kk}\check{b}_k + \sum_{\ell \neq k} G_{k\ell}\check{b}_\ell + w'_k \tag{206}$$

where $G_{k\ell}$ and w'_k are the (k,ℓ)-th and k-th entries of $\mathbf{G}$ and $\mathbf{w}'$, respectively. The last two terms in the right-hand side of (206) capture the multiuser interference-plus-noise effect, which has variance

$$\sigma_k^2 = \text{var}\left\{\sum_{\ell \neq k} G_{k\ell}\check{b}_\ell + w'_k\right\} = \sum_{\ell \neq k} G_{k\ell}^2 p_a + \Sigma_{w',kk} \tag{207}$$

where $\Sigma_{w',kk}$ denotes the (k,k)-th entry of $\mathbf{\Sigma}_{w'}$.

With the interference-plus-noise term being approximately Gaussian distributed, deciphering $\check{b}_k$ from (206) amounts to detecting a ternary deterministic signal in the presence of zero-mean, Gaussian noise of known variance. Hence, the symbol error rate (SER) for the k-th entry using the quantization rule in (204) entry-wise can be analytically obtained as

$$P_{e,k}^{\text{RD}} = 2(1-p_a)Q\left(\frac{\theta}{\sigma_k}\right) + p_a Q\left(\frac{G_{kk}-\theta}{\sigma_k}\right) \tag{208}$$

where $Q(\mu) := (1/\sqrt{2\pi})\int_\mu^\infty \exp(-\nu^2/2)d\nu$ denotes the Gaussian tail function.

The SER in (208) is a convex function of the threshold θ. Thus, taking the first-order derivative w.r.t. θ and setting it equal to zero yields the optimal threshold for the k-th entry as

$$\hat{\theta}_k = \frac{G_{kk}}{2} + \frac{\sigma_k^2}{G_{kk}}\lambda\ . \tag{209}$$

The corresponding minimum SER becomes [cf. (208) and (209)]

$$\hat{P}_{e,k}^{\text{RD}} = 2(1-p_a)Q\left(\frac{G_{kk}}{2\sigma_k} + \frac{\lambda\sigma_k}{G_{kk}}\right) + p_a Q\left(\frac{G_{kk}}{2\sigma_k} - \frac{\lambda\sigma_k}{G_{kk}}\right). \tag{210}$$

As the CDMA system signal-to-noise ratio (SNR) goes to infinity, asymptotically we have $\mathbf{G} \to \mathbf{I}$ and $\mathbf{\Sigma}_{w'} \to \mathbf{0}$, so the optimal threshold in (209) approaches 0.5. The numerical tests in Section 9 will also confirm that selecting $\theta_k = 0.5$ approaches the minimum SER $\hat{P}_{e,k}^{\text{RD}}$ over the range of SNR values encountered in most practical settings.

The clear advantage of RD-MUD is its simplicity as a linear detector. However, using the ℓ_2 norm for regularization, the RD-MUD inherently introduces a Gaussian prior for the unconstrained symbol vector and is thus not affecting sparsity in $\mathbf{b}^{\text{RD}}$; see also [278]. This renders the performance of RD dependent on the quantization threshold θ – a fact also corroborated by the simulations in Section 9. These considerations motivate the ensuing development of an alternative relaxed S-MAP algorithm, which accounts for the sparsity present in $\mathbf{b}$.

Lasso-Based MUD

Another popular regression method is the Lasso one, which regularizes the LS cost with the ℓ_1 norm. In the Bayesian formulation, regularization with the ℓ_1 norm corresponds to adopting a Laplacian prior for $\mathbf{b}$ [278]. The nice feature of Lasso-based regression is that it ensures sparsity in the resultant estimates. The degree of sparsity depends on the value of λ, which is selected here using the *a priori* information available on the activity factor [cf. (198)]. The optimal solution of (201) for $p = 1$ without the finite-alphabet constraint yields the Lasso detector (LD) as

$$\mathbf{b}^{\mathrm{LD}} = \arg\min_{\mathbf{b}} \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{b}\|_2^2 + \lambda\|\mathbf{b}\|_1. \tag{211}$$

While a closed-form solution is impossible for general $\mathbf{H}$, the minimization in (211) is a quadratic programming (QP) problem that can be readily accomplished using available QP solvers, such as SeDuMi [271]. Upon slicing the solution in (211), we obtain the detection result as

$$\hat{\mathbf{b}}^{\mathrm{LD}} = \mathcal{Q}_\theta(\mathbf{b}^{\mathrm{LD}}). \tag{212}$$

The larger λ is, the more sparse $\mathbf{b}^{\mathrm{LD}}$ becomes [cf. (198)]. This is intuitively reasonable, because λ is inversely proportional to the activity factor p_a. Since the Lasso approach (211) yields sparse estimates systematically, and can be obtained via QP solvers in polynomial time, LD is a competitive MUD alternative. Lack of a closed-form solution prevents analytical evaluation of the SER, which will be tested using simulations in Section IV.

Remark 6: So far, we assumed $\mathcal{A} = \{\pm 1\}$ to ensure equivalence of the ℓ_p-norm regularized S-MAP detector (201) with the more general one in (199). However, the sub-optimal algorithms of this section ignore the finite-alphabet constraint, and just rely on the convexity of the cost function in (201) to offer MUD schemes that can be implemented efficiently, either in linear closed-form or through quadratic programming. In fact, starting from (201) and forgoing its equivalence with (199), the RD and LD relaxations of (201) apply also for general M-ary alphabets for any $M > 2$. Of course, the quantization thresholds required for slicing the soft symbol estimates in order to obtain hard symbol estimates must be modified in accordance with the corresponding constellation. For non-constant modulus transmissions, the cost in (201) favors low-energy (close to the origin) constellation points, but this effect is mitigated by the judicious selection of the quantization thresholds.

Note that forgoing the equivalence of (201) with (199) is less of an issue for RD and LD because the major limitation of these simple relaxation-based algorithms is their sub-optimality, which emerges because they do not account for the finite-alphabet symbol constraints. Next, MUD algorithms are developed to minimize the S-MAP cost while adhering to the constraint in (199) explicitly.

S-MAP Detectors with Lattice Search

User symbols in this section are drawn from an M-ary PAM alphabet $\mathcal{A} = \{\pm 1, \pm 3, \ldots, \pm(M-1)\}$, with M even. Consider also reformulating the S-MAP problem in (199) using the QR decomposition of the matrix $\mathbf{H}$ (assumed here to be square or tall with full column rank) as $\mathbf{H} = \mathbf{QR}$, where $\mathbf{R}$ is a $K \times K$ upper triangular matrix, and $\mathbf{Q}$ is an $N \times K$ unitary matrix. Substituting this QR decomposition into (199), and left multiplying with the unitary $\mathbf{Q}$ inside the LS cost, the S-MAP detector becomes: $\hat{\mathbf{b}}^{\text{MAP}} = \arg\min_{\mathbf{b}\in\mathcal{A}_a^k} \frac{1}{2}\left\|\mathbf{Q}^T\mathbf{y} - \mathbf{Q}^T(\mathbf{QR})\mathbf{b}\right\|_2^2 + \lambda\|\mathbf{b}\|_0 = \arg\min_{\mathbf{b}\in\mathcal{A}_a^K} \frac{1}{2}\|\mathbf{y}' - \mathbf{Rb}\|_2^2 + \lambda\|\mathbf{b}\|_0$, where $\mathbf{y}' := \mathbf{Q}^T\mathbf{y}$; or, after using the definitions of the norms,

$$\hat{\mathbf{b}}^{\text{MAP}} = \arg\min_{\mathbf{b}\in\mathcal{A}_a^K} \sum_{k=1}^{K} \left\{ \frac{1}{2}\left(y'_k - \sum_{\ell=k}^{K} R_{k\ell} b_\ell \right)^2 + \lambda |b_k|_0 \right\} . \tag{213}$$

Although the optimal solution of (213) still incurs exponential complexity, the upper triangular form of $\mathbf{R}$ enables decomposition of (213) into sub-problems involving only scalar variables. As it will be seen later in Section 9, the S-MAP problem accepts a neat closed-form solution in the scalar case ($K = 1$). This is instrumental for the development of efficient (near-) optimal algorithms searching over the finite-alphabet induced lattice. One such sub-optimal MUD algorithm is described next.

Algorithm 16 (DDD): Input $\mathbf{y}'$, $\mathbf{R}$, λ, and M even. Output $\hat{\mathbf{b}}^{\mathrm{DDD}}$.

```
for $k = K, K-1, \ldots, 1$ do
    (Back substitution) Compute the unconstrained LS solution $b_k^{\mathrm{LS}} := (y'_k - \sum_{\ell=k+1}^{K} R_{k\ell}\hat{b}_\ell^{\mathrm{DDD}})/R_{kk}$.
    (Quantize to $\mathcal{A}$) Set $\hat{b}_k^{\mathrm{DDD}} := \lfloor b_k^{\mathrm{LS}} \rceil$.
    (Compare with 0) If $2b_k^{\mathrm{LS}}\lfloor b_k^{\mathrm{LS}} \rceil - \lfloor b_k^{\mathrm{LS}} \rceil^2 - 2\lambda/R_{kk}^2 \leq 0$, then set $\hat{b}_k^{\mathrm{DDD}} := 0$.
end for
```

Sparsity-Exploiting Decision-Directed MUD

Close look at (213) reveals that once the estimates $\{\hat{b}_\ell\}_{\ell=k+1}^{K}$ are available, the optimal $\hat{b}_k$ can be obtained by minimizing the cost corresponding to the k-th summand of (213). This leads to the per-symbol optimal decision-directed detector (DDD), which following related schemes in different contexts, could be also called *successive interference cancellation* or *decision-feedback decoding*, see e.g., [123, Sec. 9.4] and [288, Ch. 7]. The main difference here is that $\mathbf{b}$ is sparse.

The DDD algorithm relies on back substitution to decompose the overall S-MAP cost into K sub-costs each dependent on a single scalar variable and accepting a closed-form solution. Specifically, supposing the symbols $\{\hat{b}_\ell^{\mathrm{DDD}}\}_{\ell=k+1}^{K}$ have been already detected, the DDD algorithm detects the k-th symbol as

$$\hat{b}_k^{\mathrm{DDD}} = \arg\min_{b_k \in \mathcal{A}_a} \frac{1}{2}\left(y'_k - \sum_{\ell=k+1}^{K} R_{k\ell}\hat{b}_\ell^{\mathrm{DDD}} - R_{kk}b_k\right)^2 + \lambda|b_k|_0 \,. \quad (214)$$

This minimization problem entails only one scalar variable taking one of $(M+1)$ possible points in $\mathcal{A}_a$. Thus, the minimum is found after comparing the costs corresponding to these $(M+1)$ values. Appendix I proves that this optimal solution can be found in closed form as

$$\hat{b}_k^{\mathrm{DDD}} = \lfloor b_k^{\mathrm{LS}} \rceil \mathbb{1}\left(2b_k^{\mathrm{LS}}\lfloor b_k^{\mathrm{LS}} \rceil - \lfloor b_k^{\mathrm{LS}} \rceil^2 - 2\lambda/R_{kk}^2 > 0\right) \quad (215)$$

where $b_k^{\mathrm{LS}} := (y'_k - \sum_{\ell=k+1}^{K} R_{k\ell}\hat{b}_\ell^{\mathrm{DDD}})/R_{kk}$, and $\lfloor \cdot \rceil$ quantizes to the nearest point in $\mathcal{A}$. The simple implementation steps are tabulated as Algorithm 16.

When $\mathbf{R}$ is diagonal, Algorithm 1 yields the optimal S-MAP detection result; i.e., $\hat{\mathbf{b}}^{\text{MAP}} = \hat{\mathbf{b}}^{\text{DDD}}$ for this case. However, since the DDD detects symbols sequentially, it is prone to error propagation, especially at low SNR values. The error propagation can be mitigated by preprocessing and ordering methods [288, Ch. 7]. Also similar to all related detectors that rely on back substitution, error performance of the sparse DDD will be analyzed assuming there is no error propagation. For the special case of $M = 2$, Appendix J shows that under this assumption, the SER for (215) becomes

$$P_{e,k}^{\text{DDD}} = 2(1 - p_a)Q\left(\frac{|R_{kk}|}{2} + \frac{\lambda}{|R_{kk}|}\right) + p_a Q\left(\frac{|R_{kk}|}{2} - \frac{\lambda}{|R_{kk}|}\right) . \quad (216)$$

For a general M-ary constellation, it is also possible to approximate the SER using the union bound.

Because it accounts for the finite-alphabet constraint, sparse DDD outperforms the relaxed detectors of the previous section – a fact that will be confirmed also by simulations. However, the error propagation emerging at medium-low SNR degrades the sparse DDD performance when compared to the optimal but computationally complex S-MAP detector. As a compromise, a branch-and-bound type of MUD algorithm is developed next, to attain (near-) optimal performance by exploiting the finite-alphabet and sparsity constraints, at the price of increased complexity compared to DDD.

Sparsity-Exploiting Sphere Decoding-based MUD

Sphere decoding (SD) algorithms have been widely used for maximum-likelihood (ML) demodulation of multiuser and/or multiple-input multiple-output (MIMO) transmissions. Given a PAM or QAM alphabet, SD yields (near-) ML performance at polynomial average complexity; see e.g., [123, Sec. 5.2]. However, different from the ML-optimal SD that minimizes an LS cost, the S-MAP problem (213) entails also a regularization term to account for sparsity in $\mathbf{b}$. Although the resultant algorithm will be termed sparse sphere decoder (SSD), it searches in fact within an "ℓ_0-*norm regularized* sphere," which is not a sphere but a hyper-solid.

The goal is to find the unknown $K \times 1$ vector $\mathbf{b} \in \mathcal{A}_a^K$, which minimizes the distance metric [cf. (213)]

$$D_1^K(\mathbf{b}) := \sum_{k=1}^{K} \left\{ \frac{1}{2} \left(y'_k - \sum_{\ell=k}^{K} R_{k\ell} b_\ell \right)^2 + \lambda |b_k|_0 \right\}. \tag{217}$$

For a large enough threshold τ, candidate vectors (and thus the minimizer of D_1^K too) satisfy[22]

$$D_1^K(\mathbf{b}) < \tau \tag{218}$$

that specifies a hyper-solid inside which the wanted minimizer must lie. Define now [cf. (215)]

$$\rho_k := \left(y'_k - \sum_{\ell=k+1}^{K} R_{k\ell} b_\ell \right) \Big/ R_{kk}, \quad k = K, K-1 \cdots 1 \tag{219}$$

where $\rho_K := y'_K / R_{KK}$. Note that ρ_k depends on $\{b_\ell\}_{\ell=k+1}^K$.
Using (219), the hyper-solid in (218) can be expressed as

$$D_1^K(\mathbf{b}) = \sum_{k=1}^{K} \left\{ \frac{R_{kk}^2}{2} (\rho_k - b_k)^2 + \lambda |b_k|_0 \right\} < \tau \tag{220}$$

or, in a more compact form as $D_1^K(\mathbf{b}) = \sum_{k=1}^K d_k(b_k) < \tau$, where $d_k(b_k) := (R_{kk}^2/2)(\rho_k - b_k)^2 + \lambda |b_k|_0$. In addition to the overall metric D_1^K assessing a candidate vector $\mathbf{b} \in \mathcal{A}_a^K$, as well as the per entry metric d_k for each candidate symbol $b_k \in \mathcal{A}_a$, it will be useful to define the *accumulated* metric $D_k^K := \sum_{\ell=k}^K d_\ell(b_\ell)$ corresponding to the $K - k + 1$ candidate symbols from entry K down to entry k.
Per entry k, which subsequently will be referred to as level k, eq. (220) implies a set of inequalities:

$$\text{Level } k: \quad d_k(b_k) < \tau - D_{k+1}^K, \quad \text{for } k = K, K-1 \ldots 1, \tag{221}$$

with $D_{K+1}^K := 0$. SSD relies on the Schnorr-Euchner (SE) enumeration, see e.g., [80], properly adapted here to account for the ℓ_0-norm regularization. SE capitalizes on the inequalities (221) to search efficiently over all possible

[22] At initialization, τ is set equal to ∞ so that (218) is always satisfied.

vectors $\mathbf{b}$ with entries belonging to $\mathcal{A}_a$. Any candidate $\mathbf{b} \in \mathcal{A}_a^K$ obeying the K inequalities in (221) for a given τ, will be termed *admissible*. Threshold τ is reduced after each admissible $\mathbf{b}$ is identified, as will be detailed soon. The SE-based SSD amounts to a *depth-first* tree search, which seeks and checks candidate vectors starting from entry (level) K and working downwards to entry 1 per candidate vector.

At level K, SE search chooses $b_K = \lfloor \rho_K \rceil \mathbb{1}(2\rho_K \lfloor \rho_K \rceil - \lfloor \rho_K \rceil^2 - 2\lambda / R_{KK}^2 > 0)$, which we know from (215) is the constellation point yielding the smallest d_K. If this choice of b_K does not satisfy the inequality (221) with $k = K$, no other constellation point will satisfy it either, and the minimizer of D_1^K in (217) must lie outside[23] the hyper-solid postulated by (218). If this choice of b_K satisfies (221), SE proceeds to level $K - 1$ in which (219) is used first with $k = K - 1$ to find ρ_{K-1} that depends on the chosen b_K from level K; subsequently, b_{K-1} is selected as $b_{K-1} = \lfloor \rho_{K-1} \rceil \mathbb{1}(2\rho_{K-1} \lfloor \rho_{K-1} \rceil - \lfloor \rho_{K-1} \rceil^2 - 2\lambda / R_{K-1,K-1}^2 > 0)$. If this choice of b_{K-1} does not satisfy (221) with $k = K - 1$, then we move back to level K, and select b_K equal to the constellation point yielding the *second* smallest d_K, and so on; otherwise, we proceed to level $K - 2$. Continuing this procedure down to level 1, yields the first candidate vector $\hat{\mathbf{b}}$, which is deemed admissible since it has entries belonging to $\mathcal{A}_a$ and also satisfying (218). This candidate is stored, and the threshold is updated to $\tau := D_1^K(\hat{\mathbf{b}})$.

Then, the search proceeds looking for a better candidate. Now at level 1, we move up to level 2 and choose b_2 equal to the constellation point yielding the *second* smallest cost d_2. If this b_2 satisfies (221) at level 2 with the current τ, we move down to level 1 to update the value of b_1 (note that b_2 has just been updated and $\{b_\ell\}_{\ell=3}^K$ are equal to the corresponding entries in $\hat{\mathbf{b}}$). If (221) at level 2 is not satisfied with the current τ, we move up to level 3 to update the value of b_3, and so on.

Finally, when it fails to find any other admissible candidate satisfying (221), the search stops, and the latest admissible candidate $\hat{\mathbf{b}}$ is the optimal $\hat{\mathbf{b}}^{\mathrm{MAP}}$ solution sought. With $\tau = \infty$, the first found admissible candidate $\hat{\mathbf{b}}$ is the $\hat{\mathbf{b}}^{\mathrm{DDD}}$ solution of Section V-A.

Before summarizing the SSD steps, it is prudent to elaborate on the *ordered* enumeration of the constellation points *per level*, which in fact constitutes

[23] This will never happen with $\tau = \infty$ in (218).

the main difference of SSD relative to SD. In lieu of the 0 constellation point and the ℓ_0 norm, SE in SD enumerates the PAM symbols per level k in the order of increasing cost as: $\{b_k, b_k + 2\Delta_k, b_k - 2\Delta_k, b_k + 4\Delta_k, b_k - 4\Delta_k, \cdots\} \bigcap \mathcal{A}$, with $b_k := \lfloor \rho_k \rceil$ and $\Delta_k := \text{sign}(\rho_k - b_k)$. (With $\lfloor \rho_k \rceil$ yielding the smallest d_k, if $\Delta_k = 1$, then $\lfloor \rho_k \rceil + 2$ yields the second smallest d_k and $\lfloor \rho_k \rceil - 2$ the third; and the other way around, if $\Delta_k = -1$.) SD effects such an ordered enumeration by alternately updating $b_k = b_k + 2\Delta_k$ and $\Delta_k = -\Delta_k - \text{sign}(\Delta_k)$, [80]. To demonstrate how SSD further accounts for the ℓ_0-norm regularization and the augmented alphabet of S-MAP which includes 0, let $b_k^{(i)} \in \mathcal{A}_a$ denote the symbol for level k incurring the i-th smallest ($i = 1, 2, \ldots, M + 1$) cost d_k. If $b_k^{(i)} \in \mathcal{A}$, then $b_k^{(i+1)}$ will be either 0 or $b_k^{(i)} + 2\Delta_k$. If $d_k(0) < d_k(b_k^{(i)} + 2\Delta_k)$, then the next symbol in the ordered enumeration should be $b_k^{(i+1)} = 0$, and an auxiliary variable $b_k^{(c)}$ is used to cache the subsequent symbol in the order as $b_k^{(i+2)} = b_k^{(i)} + 2\Delta_k$. With $b_k^{(i+1)} = 0$, the auxiliary variable allows the wanted $b_k^{(i+2)}$ at the next enumeration step to be retrieved from $b_k^{(c)}$.

Similar to SD, the ordered enumeration pursued by SSD per level implies a corresponding order in considering all $\mathbf{b} \in \mathcal{A}_a^K$, which leads to a repetition-free and exhaustive search of all admissible candidate vectors. At the same time, the hyper-solid postulated by (218) shrinks as τ decreases, until no other admissible vector can be found. This guarantees that the SSD outputs the vector with the smallest D_1^K, and thus the optimal solution to (213). The SSD algorithm can be summarized in the following six steps 1–6 tabulated as Algorithm 17.

Remark 7: SSD inherits all the attractive features of SD [80]. Specifically, during the search one basically needs to store D_k^K per level k. Its *in place update* for each b_k candidate implies that SSD *memory requirements* are only linear in K. In addition, the *computational efficiency* of SSD (relative to that of ML which is $\mathcal{O}((M + 1)^K)$) stems from four factors: (i) the DDD solution provides an admissible initialization reducing the search space at the outset; (ii) the recursive search enabled by the QR decomposition gives rise to the causally dependent inequalities (221), which restrict admissible candidate entries to choices that even decrease over successive depth-first passes of the search; (iii) ordering per level increases the likelihood of finding "near-optimal admissible" candidates early, which means quick and sizeable shrinkage of

Algorithm 17 (SSD): Input $\tau = \infty$, $\mathbf{y}'$, $\mathbf{R}$, λ, and M even. Output the solution $\hat{\mathbf{b}}^{\mathrm{MAP}} := \hat{\mathbf{b}}^{\mathrm{SSD}}$ to (213).

```
1: (Initialization) Set $k := K$, $D^K_{k+1} := 0$.
2: Compute $\rho_k$ as in (219), set $b_k := \lfloor \rho_k \rceil$, $b_k^{(c)} := 0$, $\Delta_k := \mathrm{sign}(\rho_k - b_k)$.
   If $2\rho_k b_k - b_k^2 - 2\lambda / R_{kk}^2 < 0$, then
                              // symbol 0 yields smaller $d_k$ than $\lfloor \rho_k \rceil$
       Set $b_k^{(c)} := b_k$, and $b_k := 0$.
   End if and go to Step 3.
3: If $d_k(b_k) := (R_{kk}^2/2)(\rho_k - b_k)^2 + \lambda |b_k|_0 \geq \tau - D^K_{k+1}$, then
       go to Step 4.                  // outside hyper-solid in (218)
   Else if $|b_k| > M - 1$, then
         go to Step 6.                // inside hyper-solid in (218),
                                         but outside $\mathcal{A}_a$
   Else if $k > 1$, then
         compute $D_k^K := D^K_{k+1} + d_k(b_k)$; set $k := k - 1$, and go to Step 2.
         // go the next level (deeper in the tree)
   Else  go to Step 5.                 // $k = 1$, at the tree's bottom
   End if
4: If $k = K$, then terminate
   Else set $k := k + 1$, go to Step 6.
   End if
5: (An admissible b is found)
   Set $\tau := D_2^K + d_1(b_1)$, $\hat{\mathbf{b}}^{\mathrm{SSD}} := \mathbf{b}$, and $k := k + 1$; then go to Step 6.
6: (Enumeration at level $k$ proceeds to the candidate symbol next in the order)
   If $b_k = 0$, then
         Retrieve the next (based on cost $d_k$ ordering) symbol $b_k := b_k^{(c)}$,
         and set $b_k^{(c)} := FLAG$.
   Else set $b_k := b_k + 2\Delta_k$, and $\Delta_k := -\Delta_k - \mathrm{sign}(\Delta_k)$.
         If $b_k^{(c)} \neq FLAG$ and $2\rho_k b_k - b_k^2 - 2\lambda / R_{kk}^2 < 0$, then
                                 // 0 yields smaller $d_k$ than $b_k$
            Set $b_k^{(c)} := b_k$, and $b_k := 0$.
         End if
   End if and go to Step 3.
```

the hyper-solid, and thus fast convergence to the S-MAP optimal solution; and (iv) metrics involved in the search can be efficiently reused since children of the same level in the tree share the already computed accumulated metric of the "partial path" from this level to the root.

Compared to other sub-optimal detection schemes presented in previous sections, the SSD algorithm can return the S-MAP optimal solution possibly at exponential complexity, unless one stops the search at the affordable complexity – case in which the solution is only ensured to be near-optimal. Fortunately, at medium-high SNR, both SD and SSD return the optimal solution at average complexity which is polynomial (typically cubic). Moreover, SSD can be generalized to provide symbol-by-symbol soft output with approximate *a posteriori* probabilities, as is the case with the SD; see e.g., [123, Chapter 5].

Generalizations of S-MAP Detectors

Up to now, four sparsity-exploiting MUD algorithms have been developed to solve the integer program associated with the linear model in (195). The present section will present interesting generalizations to account for correlated user (in)activity across symbols, and under-determined CDMA systems.

Exploiting User (In)Activity Across Symbols

Sparsity-aware detectors for the linear model in (195) were so far developed on a symbol-by-symbol basis, which does not account for the fact that user (in)activity typically persists across multiple symbols. To this end, user activity across time can for instance be thought of as a Markov chain with two states (active-inactive). Once a user terminal starts transmitting to the AP, it becomes more likely to stay active for the next symbol slot too; and likewise, inactive once it stops transmitting. In this model, the state transition probability from either one state to the other is relatively much smaller than that of staying unchanged, and this manifests itself to the said dependence of user (in)activities across time.

Admittedly, MUD schemes accounting for this dependence must process the aggregation of data vectors $\mathbf{y}$ obeying (195) across slots. With N_s denoting the number of slots, the number of unknowns (KN_s) can grow prohibitively with N_s. One approach to cope with this "curse of dimensionality" is via dynamic programming, which can take advantage of the fact that correlation is only present between two consecutive slots; see e.g., [288, Sec. 4.2]. However, for M-ary alphabets, the resultant sequential detector requires evaluating per symbol slot the path weights of all $(M+1)^K$ possible symbol vectors. This high computational burden is impractical for real-time implementations.

The proposed alternative to bypass this challenge stems from the observation that for a given slot t the influence of all previous slots $\{t'\}_{t'=0}^{t-1}$ on the S-MAP detection rule is reflected only in the prior probability of each user being active at time t; i.e., user k's current (and time-varying) activity factor $p_{a,k}(t)$. The natural means to capture this influence online is to track each user's activity factor using the recursive LS (RLS) estimator [254, Ch. 12], based on activity factors from previous slots; that is,

$$\hat{p}_{a,k}(t) = \arg\min_{p} \sum_{t'=0}^{t-1} \beta_k^{t,t'} (p - |\hat{b}_k(t')|_0)^2, \quad t = 1, \ldots \tag{222}$$

where $\hat{b}_k(t')$ denotes user k's detected symbol at time t', and the so-called "forgetting-factor" $\beta_k^{t,t'}$ describes the effective memory (data windowing). A popular choice is the exponentially decaying window, for which $\beta_k^{t,t'} := \beta_k^{t-t'}$ for some $0 \ll \beta_k < 1$. Accordingly, (222) can expressed in closed form, recursively as

$$\begin{aligned}\hat{p}_{a,k}(t) &= \frac{1-\beta_k}{\beta_k}\left(\sum_{t'=0}^{t-1} \beta_k^{t-t'} |\hat{b}_k(t')|_0\right) / (1-\beta_k^t) \\ &= \frac{\beta_k - \beta_k^t}{1-\beta_k^t}\hat{p}_{a,k}(t-1) + \frac{1-\beta_k}{1-\beta_k^t}|\hat{b}_k(t-1)|_0, \quad t = 1, \ldots \end{aligned} \tag{223}$$

where the last equality comes from back substitution of $\hat{p}_{a,k}(t-1)$. The choice of β_k critically depends on the (in)activity correlation between consecutive slots. In the extreme case where user (in)activities across slots are independent, the infinite-memory window ($\beta_k = 1$) is optimal, and (223) reduces to the simple online time-averaging estimate $\hat{p}_{a,k}(t) = \frac{1}{t}\sum_{t'=0}^{t-1} |\hat{b}_k(t')|_0$.

Adapting the user activity factors allows one to weigh entries of ℓ_0-norm regularization which in turn affects the prior probability in the S-MAP detector (199) through the coefficient $\lambda_k(t)$ corresponding to $\hat{p}_{a,k}(t)$ (cf. Remark 5). Note that when the correlation across time is strong, it is possible that $\hat{p}_{a,k}(t)$ can approach 1, case in which $\lambda_k(t)$ is not guaranteed to stay positive. This will cause problems to the relaxed S-MAP detectors of Section 9, as those schemes rely on the convexity of the cost function in (201). However, the DDD and SSD algorithms will remain operational, because they rely on enumeration per symbol (in DDD) or per group of symbols within a sphere (in SSD). Note also that with $\lambda < 0$ the regularization term in the minimization of (213) is non-positive; hence, $\lambda < 0$ encourages searching over the non-zero constellation points (and thus discourages sparsity), whereas $\lambda > 0$ promotes sparsity.

Under-Determined CDMA Systems

Minimal-size spreading sequences, even smaller than the number of users, is well motivated for bandwidth and power savings. Without finite-alphabet constraints on the wanted vector, results available in the CS literature guarantee recovery of sparse signals from a few observations; see e.g., [46, 49] and references therein. Specifically, [46] shows that if the vector of interest is sparse (or compressible) over a known basis, then it is possible to reconstruct it with very high accuracy from a small number of random linear projections at least in the ideal noise-free case. For non-ideal observations corrupted with unknown noise of bounded perturbation, [49] provides an upper bound on the reconstruction error, which is proportional to the noise variance for a sufficiently sparse signal. However, CS theory pertains to sparse analog-valued signals. Moreover, the noise considered in a practical communication system is typically Gaussian, or generally drawn from a distribution having possibly unbounded support. Therefore, existing results from the CS literature do not carry over to the present context.

Nevertheless, it is still interesting to consider extensions of all the (sub-)optimal sparsity-exploiting MUD algorithms to an under-determined CDMA system with $N < K$, where the observation matrix $\mathbf{H}$ becomes fat. Consider first the two types of relaxed S-MAP detectors. The RD-MUD in (204) clearly works when $N < K$, since the $2\lambda\mathbf{I}$ term inside the inversion renders the overall matrix full rank. However, since the RD is a linear detector, it is expected to lose identifiability in the under-determined case, similar to the MMSE detectors for the sparsity-agnostic MUD schemes. Similarly, the LD problem (211) is also solvable for a fat $\mathbf{H}$ matrix, as the Lasso problem in CS. However, neither of them accounts for the augmented finite-alphabet constraint present in the original S-MAP problem (201).

The S-MAP detectors with lattice search are challenging to implement when $N < K$. The main obstacle is the QR decomposition of the fat matrix $\mathbf{H}$, which yields the upper triangular matrix $\mathbf{R}$ of the same dimension $N \times K$. Instead of a single unknown symbol, now the sparse DDD must optimize over the last $(N - K + 1)$ symbols in $\mathbf{b}$. However, apart from exhaustive search there is no low-complexity method to solve the aforementioned problem involving $(N - K + 1)$ variables, because sub-optimal alternatives introduce severe error propagation.

The same problem appears also with the SSD. To tackle the under-determined case, the generalized SD in [80] fixes the last $(N - K)$ symbols of $\mathbf{b}$ and relies on the standard SSD to detect the remaining K symbols that minimize a cost similar to the one in (217). Repeating this search for every choice of the last $(N - K)$ symbols, yields eventually the overall optimum vector. The complexity of the latter is exponential in $(N - K)$, regardless of the SNR. Recently, an alternative SD approach to avoid this exponential complexity has been developed for the under-determined case [75]. This algorithm takes advantage of the fact that for constant-modulus constellations the usual LS cost can be modified without affecting optimality, by adding the ℓ_2-norm $\mathbf{b}^T\mathbf{b}$ constant for every vector in the alphabet. This extra term allows one to obtain an equivalent full-rank system on which the standard SD algorithm can be applied. This efficient method can be readily extended to handle non-constant modulus constellations.

Interestingly, for our S-MAP detectors in (201) with lattice search of binary transmitted symbols, the norm term needed for regularization comes naturally from the Bernoulli prior. Specifically, with $p = 2$ the reformulated S-MAP detectors in (201) can be equivalently written as

$$\begin{aligned}\hat{\mathbf{b}}^{\text{MAP}} &= \arg\min_{\mathbf{b}\in\mathcal{A}_a^K} \frac{1}{2}[\mathbf{b}^T(\mathbf{H}^T\mathbf{H} + 2\lambda\mathbf{I})\mathbf{b} - 2\mathbf{y}^T\mathbf{H}\mathbf{b}] \\ &= \arg\min_{\mathbf{b}\in\mathcal{A}_a^K} \frac{1}{2}\|\tilde{\mathbf{y}}' - \tilde{\mathbf{R}}\mathbf{b}\|_2^2 \qquad (224)\end{aligned}$$

where $\tilde{\mathbf{R}}$ is the full rank $K \times K$ upper triangular matrix such that $\tilde{\mathbf{R}}^T\tilde{\mathbf{R}} = \mathbf{H}^T\mathbf{H} + 2\lambda\mathbf{I}$, and $\tilde{\mathbf{y}}' := \tilde{\mathbf{R}}^{-T}\mathbf{H}^T\mathbf{y}$. Utilizing the metric of (224), the back substitution of DDD and the lattice point search of SSD can be implemented easily. Hence, these S-MAP detectors can be readily extended to under-determined systems. In this way, all the (sub-)optimal S-MAP detectors are applicable with less observations than unknowns in a CDMA system with low activity factor, but their SER performance will certainly be affected. In Section 9, we will provide simulated performance comparisons of the different optimal and sub-optimal S-MAP algorithms proposed, for a variable number of observations.

Group Lassoing Block Activity

The last generalization considered pertains to user (in)activity in a (quasi-)synchronous block fashion, where during a block of N_s symbol slots, user k remains (in)active independently from other users and across blocks. Concatenate the K user symbols across N_s time slots in the $K \times N_s$ matrix $\mathbf{B} := [\mathbf{b}(1) \dots \mathbf{b}(N_s)]$, where $\mathbf{b}(t)$ collects the symbols of all K users at slot t, and likewise for the receive-data matrix $\mathbf{Y}$ as well as the noise matrix $\mathbf{W}$, both of size $N \times N_s$. With these definitions, the counterpart of (195) for this block model is $\mathbf{Y} = \mathbf{HB} + \mathbf{W}$. Letting the $N_s \times 1$ vector $\breve{\mathbf{b}}_k := [b_k(1), \dots, b_k(N_s)]^T$ collect the N_s symbols of user k, it is useful to consider it drawn from an augmented (due to possible inactivity) block alphabet $\mathcal{A}_{a,N_s} := \mathcal{A}^{N_s} \bigcup \{\mathbf{0}_{N_s}\}$. Assuming again binary transmissions, the S-MAP *block* detector will now yield

$$\begin{aligned}\hat{\mathbf{B}}^{\text{MAP}} &= \arg \min_{\breve{\mathbf{b}}_k \in \mathcal{A}_{a,N_s}} \frac{1}{2}\|\mathbf{Y} - \mathbf{HB}\|_F^2 + \sum_{k=1}^{K} \frac{\lambda_b}{\sqrt{N_s}} \|\breve{\mathbf{b}}_k\|_0 \\ &= \arg \min_{\breve{\mathbf{b}}_k \in \mathcal{A}_{a,N_s}} \frac{1}{2}\|\mathbf{Y} - \mathbf{HB}\|_F^2 + \sum_{k=1}^{K} \lambda_b \|\breve{\mathbf{b}}_k\|_2 \qquad (225)\end{aligned}$$

where $\lambda_b := \frac{1}{\sqrt{N_s}} \ln\left(\frac{1-p_a}{p_a/2^{N_s}}\right)$.
Similar to (201), the convex reformulation of the cost in (225) will lead to what is referred to in statistics as *Group Lasso* [315], which effects group sparsity on a block of symbols ($\breve{\mathbf{b}}_k$ in our case). This Group-Lasso based formulation is particularly handy for under-determined CDMA systems. In fact, the unconstrained version of (225) can be solved first to unveil the nonzero rows (i.e., the support) of $\mathbf{B}$, with improved reliability as N_s increases. Subsequently, standard sparsity-agnostic MUD schemes can be run on the estimated set of active users. Note that such a two-step approach works in the

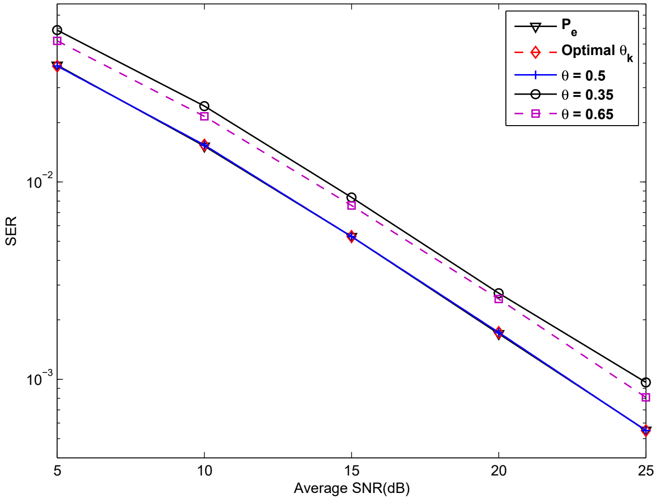


Figure 47: SER vs. average SNR (in dB) for RD-MUD with $N = 32$ and $K = 20$ and different quantization threshold θ's.

under-determined case, and also reduces number of symbols to be detected per time slot.

Simulations

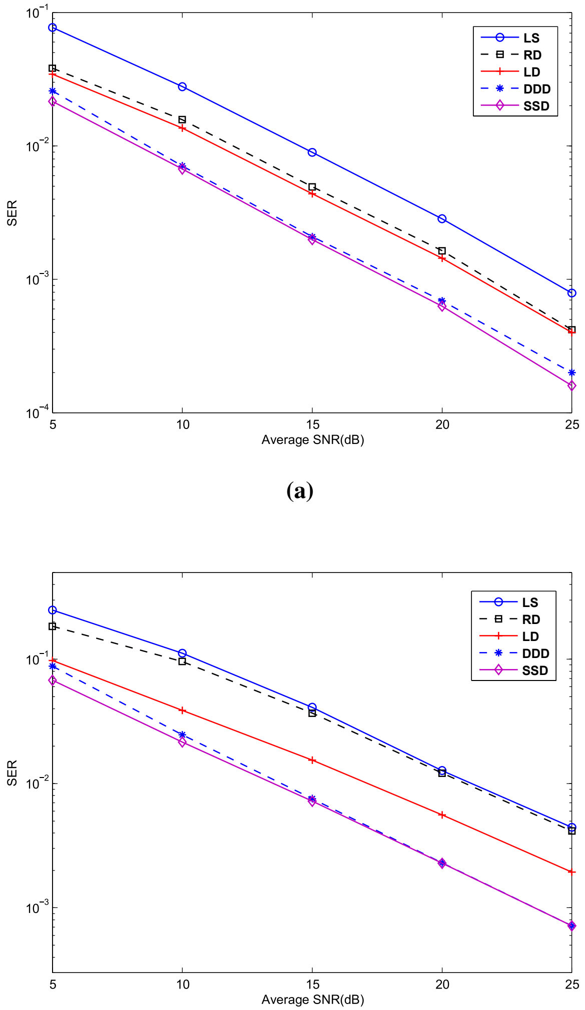
We simulate a CDMA system with $K = 20$ users, each with activity factor $p_a = 0.3$. Random sequences with length $N = 32$ are used for spreading. We consider non-dispersive independent Rayleigh fading channels between AP and users, where the channel gain g_k of the k-th user is Rayleigh distributed with variance $E[g_k^2] = \sigma^2$. Thus, the average system SNR is set to be σ^2 since the AWGN $\mathbf{w}$ has unit variance.

Test Case 1 (Quantization thresholds for RD). First, we test the RD scheme with different quantization thresholds θ in (203). The optimum threshold for the k-th symbol is obtained as in (209) per channel realization $\mathbf{H}$. The resulting SER is compared for four choices of θ: $0.5,\ 0.35,\ 0.65$, and $\hat{\theta}_k$. The theoretical minimum SER $\hat{P}_{e,k}^{\text{RD}}$ in (210) using the optimum θ is also added for comparison. Fig. 47 shows that the SER curve with $\theta = 0.5$ comes very close to the one of the optimal $\hat{\theta}_k$, and thus constitutes a near-optimal choice in practice. Moreover, those two curves also coincide with the analytical SER formulation corresponding to $\hat{P}^{\mathrm{RD}}_{e,k}$, thus corroborating the closed-form expression in (210).

Test Case 2 (S-MAP MUD algorithms). Next, the RD, LD, DDD, and SSD MUD algorithms are all tested for both BPSK and 4-PAM constellations, and their SER performance is compared. For LD, the quadratic program in (211) is solved using the general-purpose SeDuMi toolbox [271]. The quantization rule chooses the nearest point in $\mathcal{A}_a$ for both RD and LD. For comparison, we also include the ordinary LS detector, which corresponds to the RD solution in (204) with $\lambda = 0$.

Fig. 48(a) shows that the LS detector exhibits the worst performance. This is intuitive since it neither exploits the finite alphabet nor the sparsity present in $\mathbf{b}$. The SSD exhibits the best performance at the price of highest complexity. The LD outperforms the RD algorithm, as predicted. It is interesting to observe that even at low SNR region the DDD algorithm is surprisingly competitive, especially in view of its low complexity that grows only linearly in the number of symbols K. The diversity orders for those detectors are basically the same. This is reasonable since independent Rayleigh fading channels between AP and users were simulated here. The corresponding curves for 4-PAM depicted in Fig. 48(b) follow the same trend. However, compared to Fig. 48(a), the RD algorithm degrades noticeably as its SER approaches the LS one. This is because choices of quantization thresholds become more influential as the constellation size increases. As expected, the LD exhibits resilience to this influence. The DDD and SSD algorithms have almost identical performance in high-SNR region.

Test Case 3 ((In)activity across symbols). In this case, the user (in)activity is correlated across time slots. We model this random (in)activity process as a two-state (active-inactive) stationary Markov chain. The state transition matrix is $\mathbf{P} = [a\ (1-a);\ b\ (1-b)]$, where a is uniformly distributed over $[0.8\ 0.85]$, and b over $[0.05\ 0.1]$, for each user. For this model, the expected number of successive active slots is $1/(1-a)$, and $1/b$ for the inactive ones. Also, the limiting probability for the "active" state becomes $b/(1-a+b)$, taking values from the interval $[0.2\ 0.4]$. Note that the activity factor over time is still quite low. We use the RLS approach to estimate $\hat{p}_{a,k}(t)$ as in (223) using $\beta = 0.5$, and test both the DDD and SSD algorithms in solving the resultant S-MAP



(a)

(b)

Figure 48: SER vs. average SNR (in dB) of sparsity-exploiting MUD algorithms with $N = 32$ and $K = 20$ for (a) BPSK, and (b) 4-PAM alphabets.

detection problem. The empirical SER is plotted in Fig. 49 across time for different SNR values. Clearly, the proposed scheme is effective in tracking

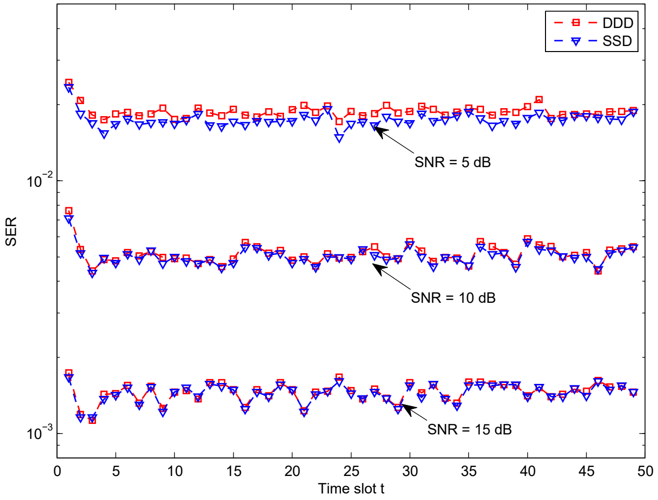


Figure 49: SER vs. time t of sparsity-exploiting MUD algorithms with RLS estimation of the activity factors.

the evolving time-correlated activity. For the same SNR value, it yields SER performance similar to the independent case in Fig. 48(a).

Test Case 4 (Under-determined CDMA systems). We also test the S-MAP MUD algorithms for under-determined systems, by varying N from 32 to 16 and 8. The results are depicted in Fig. 50. Since the RD is a simple linear detector, it is expected that once $N < K$, it will lose identifiability, and exhibits a considerably flat SER curve. At the same time, DDD still enjoys almost full diversity with a moderate choice of $N = 16$. Being the optimum detector, the SSD collects the full diversity even if $N = 8$; however, the other two kinds of detectors exhibit flat SER curves, as expected.

The Group Lasso scheme for recovering block activity is also included for the under-determined case. Fig. 51 illustrates the activity recovery error rate for different values of N and N_s. The number of observations N affects the diversity order, while the block size N_s influences the recovery accuracy.

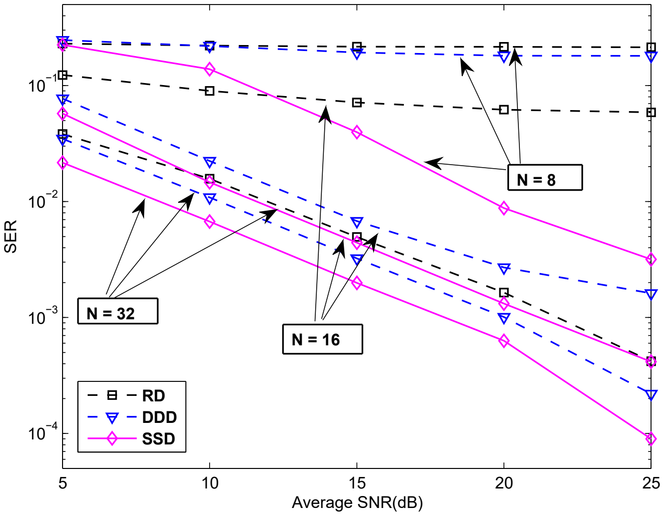


Figure 50: SER vs. average SNR (in dB) of sparsity-exploiting MUD algorithms with $N = 32, 16$, or 8 and $K = 20$.

Conclusions

The MUD problem of sparse symbol vectors emerging with CDMA systems having low-activity factor was considered. Viewing user inactivity as augmenting the underlying alphabet, the *a priori* available sparsity information was exploited in the optimal S-MAP detector. The exponential complexity associated with the S-MAP detector was reduced by resorting to (sub-) optimal algorithms. Relaxed S-MAP detectors (RD and LD) come with low complexity but sacrifice optimality, because they ignore the finite-alphabet constraint. The second kind of detectors (DDD and SSD), searches over a subset of the alphabet and exhibits improved performance at increased complexity. The performance was analyzed for the RD and DDD algorithms, and closed-form expressions were derived for the corresponding symbol error rates.

S-MAP detectors were further generalized to deal with correlated user (in)activity across symbols by recursively estimating each user's activity factor online; and also with under-determined (a.k.a. over-saturated CDMA) settings emerging when the spreading gain is smaller than the potential number of

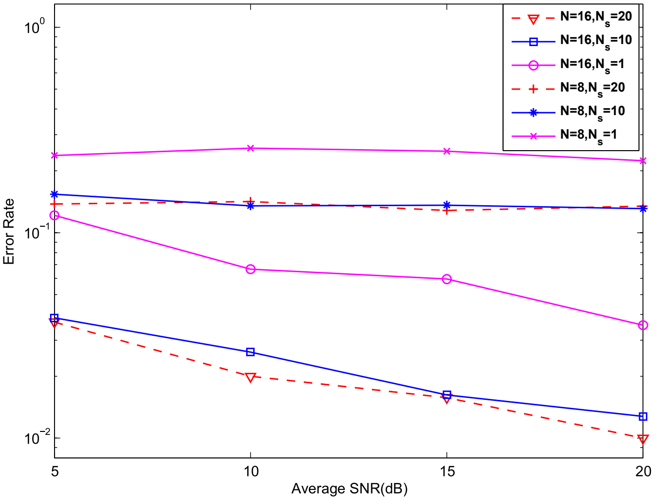


Figure 51: Error rate for block activity vs. average SNR (in dB) of Group Lasso algorithm with $N = 16, 8$ and $N_s = 20, 10, 1$.

users. Coping with the latter becomes possible through regularization with the ℓ_0 norm prior, or, with the Group Lasso-based recovery of the active user set. The numerical tests corroborated our analytical findings and quantified the relative performance of the developed sparsity-exploiting MUD algorithms.

10

Summary

As it was manifested, sparsity is present – either originally or after projecting the signal onto an appropriate basis – in a plethora of natural signals and practical communication systems. Accordingly, this monograph reviewed various CS frameworks where the signal sparsity is used as the key attribute to ameliorate the signal reconstruction/detection performance while conserving the radio and energy resources by reducing the sampling rate, transmission rate, and computational load. In the first part, advanced CS techniques based on least-squares and Lasso were presented for sparsity-aware recursive real-time estimation and signal reconstruction in a perturbed sensing setup. The second part was dedicated to CS based environmental monitoring and lossy compression techniques in wireless sensor networks. The third part addressed spectrum sensing in cognitive radio communications and multi-user detection in wireless networks. The numerical experiments of each framework demonstrated that exploiting an inherent sparsity of the underlying signal provides significant system performance gains. Thus, the principles of the presented sparse signal processing techniques are worth taking into consideration when striving for high performance in an application involving signal sparsity.

Acknowledgements

The work of M. Leinonen and M. Codreanu has been financially supported in part by Infotech Oulu, the Academy of Finland (grant 323698), and Academy of Finland 6Genesis Flagship (grant 318927). M. Codreanu would like to acknowledge the support of the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 793402 (COMPRESS NETS). G. B. Giannakis' work was supported in part by NSF grants NSF 1508993, 1514056, 1711471, and 1901134.

Appendices

A

Proof of Proposition 4

Define the vector $\boldsymbol{\chi}_k := \widehat{\mathbf{x}}_N^{\mathrm{OCD}}$ for $N = kP$, that is, the one containing the iterates at the end of the kth cycle when each variable has been updated k times. The proof that $\boldsymbol{\chi}_k$ converges to $\mathbf{x}_o$ as $k \to \infty$ will proceed in five stages. In the first one, Algorithm 1 is put in the form of a noisy vector-matrix difference equation. The second and third stages prove that the corresponding discrete-time dynamical system is exponentially stable, and that the sequence $\{\boldsymbol{\chi}_k\}_{k=1}^{\infty}$ is bounded. In the fourth stage, convergence to a limit point $\boldsymbol{\chi}_\infty$ is proved. The proof concludes by showing that $\boldsymbol{\chi}_\infty = \mathbf{x}_o$.

Dynamical System

Let $\bar{\mathbf{R}}_k$ denote the matrix with entries $\bar{R}_k(p,q) := R_{kP+p}(p,q)/(kP+p)$, and $\bar{\mathbf{r}}_k$ the vector with entries $\bar{r}_k(p) := r_{kP+p}(p)/(kP+p)$. Conditions (r1) and (r2) guarantee that $\bar{\mathbf{R}}_k \underset{k\to\infty}{\longrightarrow} \mathbf{R}_\infty$ and $\bar{\mathbf{r}}_k \underset{k\to\infty}{\longrightarrow} \mathbf{r}_\infty$ w.p. 1. Consider the decomposition $\bar{\mathbf{R}}_k = \bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k + \bar{\mathbf{U}}_k$ where $\bar{\mathbf{D}}_k$ is diagonal, and $\bar{\mathbf{L}}_k$ ($\bar{\mathbf{U}}_k$) is strictly lower (upper) triangular. Observe that $\bar{\mathbf{L}}_k \neq \bar{\mathbf{U}}_k^T$.
Dividing the cost function of the problem in (17) by $kP + p$ yields

$$\chi_{k+1}(p) = \arg\min_{\chi}\left[\frac{1}{2}\left(\frac{R_{kP+p}(p,p)}{kP+p}\right)\chi^2 - \left(\frac{r_{kP+p,p}}{kP+p}\right)\chi + \left(\frac{\lambda}{kP+p}\right)|\chi|\right]. \tag{226}$$

The solution of this scalar minimization problem can be obtained in two steps. First, solve the differentiable linear-quadratic part of (226) using the auxiliary vector $\mathbf{z}_{k+1}$ to obtain

$$\begin{aligned} z_{k+1}(p) = \arg\min_{z}\Bigg[\frac{1}{2}\left(\frac{R_{kP+p}(p,p)}{kP+p}\right)z^2 - \Bigg(\frac{r_{kP+p}(p)}{kP+p} - \\ \sum_{q<p}\frac{R_{kP+p}(p,q)}{kP+p}\chi_{k+1}(q) - \sum_{q>p}\frac{R_{kP+p}(p,q)}{kP+p}\chi_k(q)\Bigg)z\Bigg] \end{aligned} \tag{227}$$

where $r_{kP+p,p}$ was expanded according to its definition in (21) and divided in two sums: the one already updated in the cycle $k+1$ $(q<p)$, and the second one updated in the kth cycle $(q>p)$. The second step to solve (226) is to pass $z_{k+1}(p)$ through the soft-threshold operator

$$\chi_{k+1}(p) = \operatorname{sgn}\left(z_{k+1}(p)\right)\left[|z_{k+1}(p)| - \frac{\lambda_{kP+p}}{kP+p}\right]_+ . \tag{228}$$

Using the decomposition of $\bar{\mathbf{R}}_k$, (227) can be rewritten as

$$z_{k+1}(p) = \arg\min_z \frac{1}{2}\bar{D}_k(p,p)\, z^2 - \\ \left(\bar{r}_k(p) - \sum_{q<p}\bar{L}_k(p,q)\chi_{k+1}(q) - \sum_{q>p}\bar{U}_k(p,q)\chi_k(q)\right) z$$

whose solution is obtained (after equating the derivative to zero) as

$$\bar{D}_k(p,p)\; z_{k+1}(p) = \bar{r}_k(p) - \sum_{q<p}\bar{L}_k(p,q)\chi_{k+1}(q) - \sum_{q>p}\bar{U}_k(p,q)\chi_k(q). \tag{229}$$

Concatenating the latter with $p=1,\ldots,P$ yields the matrix-vector difference equation

$$\bar{\mathbf{D}}_k \boldsymbol{z}_{k+1} = \bar{\mathbf{r}}_k - \bar{\mathbf{L}}_k \boldsymbol{\chi}_{k+1} - \bar{\mathbf{U}}_k \boldsymbol{\chi}_k. \tag{230}$$

The soft-thresholding operation in (228) can be accounted for by defining the error vector $\boldsymbol{\epsilon}_k := \boldsymbol{\chi}_{k+1} - \mathbf{z}_{k+1}$ in which case (230) can be re-written as

$$\bar{\mathbf{D}}_k(\boldsymbol{\chi}_{k+1} - \boldsymbol{\epsilon}_k) = \bar{\mathbf{r}}_k - \bar{\mathbf{L}}_k \boldsymbol{\chi}_{k+1} - \bar{\mathbf{U}}_k \boldsymbol{\chi}_k. \tag{231}$$

Assuming that there exists a $k^\star$ such that $\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k$ is invertible for each $k > k^\star$, (231) can be written as

$$\boldsymbol{\chi}_{k+1} = \bar{\mathbf{G}}_k \boldsymbol{\chi}_k + (\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k)^{-1}\bar{\mathbf{r}}_k + (\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k)^{-1}\bar{\mathbf{D}}_k \boldsymbol{\epsilon}_k \tag{232}$$

with $\bar{\mathbf{G}}_k := (\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k)^{-1}\bar{\mathbf{U}}_k$. The key point to be used subsequently is that (228) guarantees that the entries of $\boldsymbol{\epsilon}_k$ are bounded by a vanishing sequence. Specifically, it holds that

$$|\epsilon_k(p)| \leq \frac{\lambda_{kP+p}}{kP+p}, \text{ for } p = 1,\ldots,P \tag{233}$$

since the input-output variables of the soft-threshold operator $\chi = \mathrm{sgn(z)}[|\mathrm{z}| - \lambda]_+$ obey $|\chi - z| \leq \lambda$.

Exponential Stability

Let $\bar{\mathbf{G}}(l:k) := \prod_{i=l}^{k} \bar{\mathbf{G}}_i$ in (232) denote the product of the transition matrices $\bar{\mathbf{G}}_k$. The goal of this stage is to prove that $||\bar{\mathbf{G}}(l:k)|| \leq c\rho^{k-l+1}$, with $\rho < 1$. The convergence of $\bar{\mathbf{R}}_k$ to $\mathbf{R}_\infty$ implies convergence of $\bar{\mathbf{G}}_k$ to $\mathbf{G}_\infty := -(\mathbf{D}_\infty + \mathbf{L}_\infty)^{-\mathbf{1}}\mathbf{U}_\infty$, where $\mathbf{D}_\infty$, $\mathbf{L}_\infty$ and $\mathbf{U}_\infty$ are the diagonal, lower triangular, and upper triangular parts of $\mathbf{R}_\infty$, respectively. Since $\mathbf{R}_\infty$ is positive definite, the spectral radius of $\mathbf{G}_\infty$ is strictly less than one, i.e., $\varrho(G_\infty) < 1$ [127, p. 512]. Furthermore, for every $\delta > 0$ there exists a $c(\delta)$ constant w.r.t. k, such that $||\mathbf{G}_\infty^k|| < c(\delta)[\varrho(\mathbf{G}_\infty) + \delta]^k$ [127, p. 336]. Then, by selecting $\delta = (1 - \varrho(\mathbf{G}_\infty))/2$, and defining $\rho_\infty := (1 + \varrho(\mathbf{G}_\infty))/2$, it holds that

$$||\mathbf{G}_\infty^k|| < c\rho_\infty^k, \qquad \text{with } \rho_\infty < 1. \tag{234}$$

Upon defining $\tilde{\mathbf{G}}_k := \bar{\mathbf{G}}_k - \mathbf{G}_\infty$, the following recursion is obtained

$$\begin{aligned} \bar{\mathbf{G}}(1:k) &= \bar{\mathbf{G}}_k \bar{\mathbf{G}}(1:k-1) \\ &= \mathbf{G}_\infty \bar{\mathbf{G}}(1:k-1) + \tilde{\mathbf{G}}_k \bar{\mathbf{G}}(1:k-1) \\ &= \mathbf{G}_\infty^k + \sum_{i=1}^{k} \mathbf{G}_\infty^{k-i} \tilde{\mathbf{G}}_i \bar{\mathbf{G}}(1:i-1), \quad \bar{\mathbf{G}}(1:0) := \mathbf{I}. \end{aligned}$$

Using (234), the latter can be bounded as

$$||\bar{\mathbf{G}}(1:k)|| \leq c\rho_\infty^k + c\sum_{i=1}^{k} \rho_\infty^{k-i} ||\tilde{\mathbf{G}}_i|| ||\bar{\mathbf{G}}(1:i-1)||$$

which after multiplying both sides by ρ_∞^{-k} yields

$$\rho_\infty^{-k} ||\bar{\mathbf{G}}(1:k)|| \leq c + \sum_{i=1}^{k} c\rho_\infty^{-1} ||\tilde{\mathbf{G}}_i|| ||\bar{\mathbf{G}}(1:i-1)|| \rho_\infty^{-(i-1)} \tag{235}$$

and allows one to apply the discrete Bellman-Gronwall lemma (see e.g., [269, p. 315]).

Lemma 5 (Bellman-Gronwall)**.** *If $c, \xi_k, h_k \geq 0$ $\forall k$ satisfy the recursive inequality*

$$\xi_k \leq c + \sum_{i=1}^{k} h_{i-1} \xi_{i-1} \tag{236}$$

then ξ_k obeys the non-recursive inequality

$$\xi_k \leq c \prod_{i=1}^{k} (1 + h_{i-1}). \tag{237}$$

For $\xi_k = \rho_\infty^{-k} ||\bar{\mathbf{G}}(1:k)||$ and $h_k = c\rho_\infty^{-1} ||\tilde{\mathbf{G}}_{k+1}||$, (235) takes the form of (236), so that (237) holds and (after multiplying both sides by ρ_∞^k) results in

$$||\bar{\mathbf{G}}(1:k)|| \leq \rho_\infty^k c \prod_{i=1}^{k} \left(1 + c\rho_\infty^{-1} ||\tilde{\mathbf{G}}_i||\right) = c \prod_{i=1}^{k} \left(\rho_\infty + c||\tilde{\mathbf{G}}_i||\right). \tag{238}$$

Raising both sides of (238) to the power of $1/k$ and applying the geometric-arithmetic mean inequality, it follows that

$$||\bar{\mathbf{G}}(1:k)||^{1/k} \leq c^{1/k} \frac{1}{k} \sum_{i=1}^{k} \left(\rho_\infty + c||\tilde{\mathbf{G}}_i||\right)$$

which is readily rewritten as

$$||\bar{\mathbf{G}}(1:k)|| \leq c \left(\rho_\infty + c\frac{1}{k} \sum_{i=1}^{k} ||\tilde{\mathbf{G}}_i||\right)^k .$$

Since $\bar{\mathbf{G}}_k \underset{k\to\infty}{\longrightarrow} \mathbf{G}_\infty$ and $||\tilde{\mathbf{G}}_k|| \underset{k\to\infty}{\longrightarrow} \mathbf{0}$ w.p. 1, for every $\delta > 0$ there exists an integer k_0 such that if $k \geq k_0$, then $\frac{1}{k}\sum_{i=1}^{k} ||\tilde{\mathbf{G}}_i|| \leq \delta$ w.p 1. Thus, if δ is selected as $(1-\rho_\infty)/(2c)$, and ρ as $(1+\rho_\infty)/2$, the following bound is obtained

$$||\bar{\mathbf{G}}(1:k)|| \leq c\rho^k, \qquad \rho < 1,\ k \geq k_0, \qquad \text{w.p. } 1.$$

It is clear, by inspection, that the proof so far carries over even if the product of transition matrices starts at $l > 1$; that is

$$||\bar{\mathbf{G}}(l:k)|| \leq c\rho^{k-l+1}, \qquad \rho < 1,\ k \geq k_0, \qquad \text{w.p. } 1. \tag{239}$$

Certainly, c, ρ, and k_0 do not depend on l. However, k_0 does depend on the realization of the random sequence $\bar{\mathbf{G}}_k$, and its existence and finiteness are guaranteed w.p. 1.

Boundedness

Define $\mathbf{v}_k := (\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k)^{-1}(\bar{\mathbf{r}}_k + \bar{\mathbf{D}}_k \boldsymbol{\epsilon}_k)$, and rewrite (232) as

$$\boldsymbol{\chi}_{k+1} = \bar{\mathbf{G}}_k \boldsymbol{\chi}_k + \mathbf{v}_k.$$

Using back substitution, $\boldsymbol{\chi}_{k+1}$ can then be expressed as

$$\boldsymbol{\chi}_{k+1} = \bar{\mathbf{G}}(k_0 : k)\boldsymbol{\chi}_{k_0} + \sum_{i=1}^{k-k_0} \bar{\mathbf{G}}(k-i+1 : k)\mathbf{v}_{k-i} + \mathbf{v}_k.$$

Since $\bar{\mathbf{D}}_k$, $\bar{\mathbf{L}}_k$, $\bar{\mathbf{r}}_k$, and $\boldsymbol{\epsilon}_k$ converge, the random sequence $\mathbf{v}_k$ converges too w.p. 1; hence, it can be stochastically bounded by a random variable v; that is, $||\mathbf{v}_k|| < v$, $\forall k$, w.p. 1. This, combined with the exponential stability ensured by (239), guarantees that the realizations of the random sequence $\boldsymbol{\chi}_k$ are (stochastically) bounded; thus

$$\begin{aligned} ||\boldsymbol{\chi}_{k+1}|| &\leq c\rho^{-k+k_0-1}||\boldsymbol{\chi}_{k_0}|| + \sum_{i=1}^{k-k_0} c\rho^{-i} v + v \\ &\leq c||\boldsymbol{\chi}_{k_0}|| + cv\left(\frac{1}{1-\rho} + 1\right), \quad \text{w.p. } 1. \end{aligned} \tag{240}$$

Convergence

Define the error $\tilde{\mathbf{D}}_k := \bar{\mathbf{D}}_k - \mathbf{D}_\infty$, and similarly $\tilde{\mathbf{L}}_k := \bar{\mathbf{L}}_k - \mathbf{L}_\infty$, $\tilde{\mathbf{U}}_k := \bar{\mathbf{U}}_k - \mathbf{U}_\infty$, and $\tilde{\mathbf{r}}_k := \bar{\mathbf{r}}_k - \mathbf{r}_\infty$. Using these new variables, (231) can be rewritten in error form as

$$(\mathbf{D}_\infty + \tilde{\mathbf{D}}_k)(\boldsymbol{\chi}_{k+1} - \boldsymbol{\epsilon}_k) = (\mathbf{r}_\infty + \tilde{\mathbf{r}}_k) - (\mathbf{L}_\infty + \tilde{\mathbf{L}}_k)\boldsymbol{\chi}_{k+1} - (\mathbf{U}_\infty + \tilde{\mathbf{U}}_k)\boldsymbol{\chi}_k \tag{241}$$

and, after regrouping terms, as

$$\begin{aligned} &\boldsymbol{\chi}_{k+1} = -(\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}\mathbf{U}_\infty \boldsymbol{\chi}_k + (\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1} \\ &\times\Big(\mathbf{r}_\infty + \tilde{\mathbf{r}}_k - (\tilde{\mathbf{D}}_k + \tilde{\mathbf{L}}_k)\boldsymbol{\chi}_{k+1} + (\mathbf{D}_\infty + \tilde{\mathbf{D}}_k)\boldsymbol{\epsilon}_k - \tilde{\mathbf{U}}_k \boldsymbol{\chi}_k\Big). \end{aligned} \tag{242}$$

Equation (242) describes an exponentially stable linear time-invariant system with transition matrix $-(\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}\mathbf{U}_\infty$, and input $\mathbf{u}_k := (\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}[\mathbf{r}_\infty + \tilde{\mathbf{r}}_k - (\tilde{\mathbf{D}}_k + \tilde{\mathbf{L}}_k)\boldsymbol{\chi}_{k+1} + (\mathbf{D}_\infty + \tilde{\mathbf{D}}_k)\boldsymbol{\epsilon}_k - \tilde{\mathbf{U}}_k \boldsymbol{\chi}_k]$. The input can be divided into its limiting point $\mathbf{u}_\infty := (\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}\mathbf{r}_\infty$, and the error

$\tilde{\mathbf{u}}_k := (\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}[\tilde{\mathbf{r}}_k - (\tilde{\mathbf{D}}_k + \tilde{\mathbf{L}}_k)\boldsymbol{\chi}_{k+1} + (\mathbf{D}_\infty + \tilde{\mathbf{D}}_k)\boldsymbol{\epsilon}_k - \tilde{\mathbf{U}}_k\boldsymbol{\chi}_k]$. As $k \to \infty$, the vector $\tilde{\mathbf{u}}_k$ goes to zero almost surely because the sequence $\boldsymbol{\chi}_k$ is bounded, and the error $\tilde{\mathbf{r}}_k$, $\tilde{\mathbf{D}}_k$, $\tilde{\mathbf{L}}_k$ and $\tilde{\mathbf{U}}_k$ as well as $\boldsymbol{\epsilon}_k$, all go to zero w.p. 1. With this notation and recalling the definition $\mathbf{G}_\infty := -(\mathbf{D}_\infty + \mathbf{L}_\infty)^{-\mathbf{1}}\mathbf{U}_\infty$, (242) is rewritten as

$$\boldsymbol{\chi}_{k+1} = \mathbf{G}_\infty\boldsymbol{\chi}_k + \mathbf{u}_\infty + \tilde{\mathbf{u}}_k$$

and back-substituting again the new expression for $\boldsymbol{\chi}_{k+1}$ yields

$$\boldsymbol{\chi}_{k+1} = \mathbf{G}_\infty^k\boldsymbol{\chi}_1 + \sum_{i=1}^{k}\mathbf{G}_\infty^{k-i}\mathbf{u}_\infty + \sum_{i=1}^{k}\mathbf{G}_\infty^{k-i}\tilde{\mathbf{u}}_i. \tag{243}$$

Convergence of this recursion will be established by showing that the first and third terms in the right-hand side vanish as $k \to \infty$, while the surviving one corresponds to a stable geometric series. Given that $\exists\ c > 0, \rho_\infty < 1$ such that $\forall n\ ||\boldsymbol{G}_\infty^n|| \leq c\rho_\infty^n$, convergence of the first term to zero follows readily from (234). The third term represents the limiting output value of a multiple input-multiple output stable linear time-invariant system with vanishing input; that is $\lim_{i\to\infty}\tilde{\boldsymbol{u}}_i = 0$. As $\forall k\ ||\boldsymbol{G}_\infty^k|| \leq c\rho_\infty^k$, (234) implies that it is possible to bound the sum under consideration as

$$\left\|\sum_{i=1}^{k}\mathbf{G}_\infty^{k-i}\tilde{\mathbf{u}}_i\right\| \leq c\sum_{i=1}^{k}\rho_\infty^{k-i}||\tilde{\mathbf{u}}_i||\ . \tag{244}$$

Since $\lim_{i\to\infty}\tilde{\mathbf{u}}_i = 0$, it holds by the definition of the limit that for any $\epsilon > 0\ \exists\ N \in \mathbb{N}$ so that $||\tilde{\mathbf{u}}_i|| \leq \epsilon, \forall\ i \geq N$. Using the latter along with (244), it follows that for $k \geq N$

$$\begin{aligned}\left\|\sum_{i=1}^{k}\mathbf{G}_\infty^{k-i}\tilde{\mathbf{u}}_i\right\| &\leq c\sum_{i=1}^{N-1}\rho_\infty^{k-i}||\tilde{\mathbf{u}}_i|| + c\epsilon\sum_{i=N}^{k}\rho_\infty^{k-i} \\ &= c\rho_\infty^{k-N}\sum_{i=1}^{N-1}\rho_\infty^{N-i}||\tilde{\mathbf{u}}_i|| + c\epsilon\sum_{i=N}^{k}\rho_\infty^{k-i}\ . \end{aligned} \tag{245}$$

Because $\sum_{i=1}^{N-1}\rho_\infty^{N-i}||\tilde{\mathbf{u}}_i||$ does not depend on k, the limit of the first summand in (245) goes to zero; hence,

$$\lim_{k\to\infty}\left\|\sum_{i=1}^{k}\mathbf{G}_\infty^{k-i}\tilde{\mathbf{u}}_i\right\| \leq c\epsilon/(1-\rho_\infty)\ .$$

The last inequality holds $\forall\epsilon > 0$; thus,

$$\lim_{k\to\infty} \left\| \sum_{i=1}^{k} \mathbf{G}_{\infty}^{k-i} \tilde{\mathbf{u}}_i \right\| = 0$$

which establishes convergence to zero of the third sum in the right-hand side of (243).

Limit Point

Once convergence is established, it is possible to take the limit as $k \to \infty$ in (241) to obtain

$$\mathbf{D}_{\infty}(\boldsymbol{\chi}_{\infty} - \boldsymbol{\epsilon}_{\infty}) = \mathbf{r}_{\infty} - \mathbf{L}_{\infty}\boldsymbol{\chi}_{\infty} - \mathbf{U}_{\infty}\boldsymbol{\chi}_{\infty}, \qquad \text{w.p. } 1. \tag{246}$$

Recalling that $||\boldsymbol{\epsilon}_{\infty}|| \leq \lim_{N\to\infty} \frac{\lambda_N}{N} = 0$, (246) reduces to

$$(\mathbf{D}_{\infty} + \mathbf{L}_{\infty} + \mathbf{U}_{\infty})\boldsymbol{\chi}_{\infty} = \mathbf{r}_{\infty}, \qquad \text{w.p. } 1 \tag{247}$$

and since $\mathbf{D}_{\infty} + \mathbf{L}_{\infty} + \mathbf{U}_{\infty} = \mathbf{R}_{\infty}$, it holds that

$$\boldsymbol{\chi}_{\infty} = (\mathbf{R}_{\infty})^{-1}\mathbf{r}_{\infty} = \mathbf{x}_o, \qquad \text{w.p. } 1 \tag{248}$$

which concludes the proof.

B

Construction of Matrix Ψ' in (76)

The matrix $\boldsymbol{\Psi}'$ in (76) can be expressed as

$$\begin{aligned}\boldsymbol{\Psi}' &= \boldsymbol{\Psi}'_{\mathrm{T}} \otimes \boldsymbol{\Psi}_{\mathrm{S}} \\ &\overset{(a)}{=} (\mathbf{I}'_{\mathrm{T}} \boldsymbol{\Psi}_{\mathrm{T}}) \otimes \boldsymbol{\Psi}_{\mathrm{S}} \\ &\overset{(b)}{=} (\mathbf{I}'_{\mathrm{T}} \boldsymbol{\Psi}_{\mathrm{T}}) \otimes (\mathbf{I}_N \boldsymbol{\Psi}_{\mathrm{S}}) \\ &\overset{(c)}{=} (\mathbf{I}'_{\mathrm{T}} \otimes \mathbf{I}_N)(\boldsymbol{\Psi}_{\mathrm{T}} \otimes \boldsymbol{\Psi}_{\mathrm{S}}) \\ &\overset{(d)}{=} (\mathbf{I}'_{\mathrm{T}} \otimes \mathbf{I}_N)\boldsymbol{\Psi} \\ &\overset{(e)}{=} [\boldsymbol{\psi}_1 \ \cdots \ \boldsymbol{\psi}_{N(W-1)}]^{\mathrm{T}}\end{aligned} \tag{249}$$

where (a) follows from defining a binary sparse matrix $\mathbf{I}'_{\mathrm{T}} \triangleq [\mathbf{I}_{W-1} \ \mathbf{0}_{(W-1)\times 1}] \in \mathbb{B}^{(W-1)\times W}$ that performs a row selection of $\boldsymbol{\Psi}_{\mathrm{T}}$ as $\boldsymbol{\Psi}'_{\mathrm{T}} = \mathbf{I}'_{\mathrm{T}} \boldsymbol{\Psi}_{\mathrm{T}}$; (b) follows because $\boldsymbol{\Psi}_{\mathrm{S}} = \mathbf{I}_N \boldsymbol{\Psi}_{\mathrm{S}}$; (c) follows from $\mathbf{AB} \otimes \mathbf{CD} = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \otimes \mathbf{D})$; (d) follows from (69); (e) follows from performing a row selection of $\boldsymbol{\Psi}$ via a binary sparse matrix $\mathbf{I}'_{\mathrm{T}} \otimes \mathbf{I}_N = [\mathbf{I}_{N(W-1)} \ \mathbf{0}_{N(W-1)\times N}] \in \mathbb{B}^{N(W-1)\times NW}$ as $(\mathbf{I}'_{\mathrm{T}} \otimes \mathbf{I}_N)\boldsymbol{\Psi} = [\boldsymbol{\psi}_1 \ \cdots \ \boldsymbol{\psi}_{N(W-1)}]^{\mathrm{T}}$, which is the desired result.

C

MMSE Orthogonality Principle for (125)

In order to show the equality $\mathbb{E}\left[\|\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s+\underline{\mathbf{Z}}_s-\hat{\underline{\mathbf{X}}}_s\|_2^2\right] = \mathbb{E}\left[\|\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s\|_2^2\right]+\mathbb{E}\left[\|\underline{\mathbf{Z}}_s-\hat{\underline{\mathbf{X}}}_s\|_2^2\right]$ in step (a) of (125), we show that the associated cross-term $\mathbb{E}\left[(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)^{\mathrm{T}}(\underline{\mathbf{Z}}_s-\hat{\underline{\mathbf{X}}}_s)\right]$ is zero. This term can be written as

$$\mathbb{E}\left[(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)^{\mathrm{T}}(\underline{\mathbf{Z}}_s-\hat{\underline{\mathbf{X}}}_s)\right]=\mathbb{E}\left[\underline{\mathbf{Z}}_s^{\mathrm{T}}(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)\right]-\mathbb{E}\left[\hat{\underline{\mathbf{X}}}_s^{\mathrm{T}}(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)\right]. \tag{250}$$

By the law of total expectation, the first term of (250) can be written as

$$\begin{aligned}\mathbb{E}\left[\underline{\mathbf{Z}}_s^{\mathrm{T}}(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)\right] &= \mathbb{E}\Big\{\mathbb{E}\left[\underline{\mathbf{Z}}_s^{\mathrm{T}}(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)|\mathbf{Y}_s\right]\Big\}\\ &= \mathbb{E}\Big\{\underline{\mathbf{Z}}_s^{\mathrm{T}}(\mathbb{E}[\underline{\mathbf{X}}_s|\mathbf{Y}_s]-\underline{\mathbf{Z}}_s)\Big\}\\ &\overset{(a)}{=} \mathbb{E}\Big\{\underline{\mathbf{Z}}_s^{\mathrm{T}}(\underline{\mathbf{Z}}_s-\underline{\mathbf{Z}}_s)\Big\}\\ &= 0\end{aligned} \tag{251}$$

where (a) follows from (123). Similarly, the second term of (250) can be written as

$$\begin{aligned}\mathbb{E}\left[\hat{\underline{\mathbf{X}}}_s^{\mathrm{T}}(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)\right] &= \mathbb{E}\Bigg(\mathbb{E}\Big\{\mathbb{E}\left[\hat{\underline{\mathbf{X}}}_s^{\mathrm{T}}(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)|\underline{\mathbf{X}}_s,\mathbf{Y}_s\right]|\mathbf{Y}_s\Big\}\Bigg)\\ &= \mathbb{E}\Bigg(\mathbb{E}\Big\{\mathbb{E}\left[\hat{\underline{\mathbf{X}}}_s|\underline{\mathbf{X}}_s,\mathbf{Y}_s\right]^{\mathrm{T}}(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)|\mathbf{Y}_s\Big\}\Bigg)\\ &\overset{(a)}{=} \mathbb{E}\Bigg(\mathbb{E}\Big\{\mathbb{E}\left[\hat{\underline{\mathbf{X}}}_s|\mathbf{Y}_s\right]^{\mathrm{T}}(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)|\mathbf{Y}_s\Big\}\Bigg)\\ &= \mathbb{E}\Bigg(\mathbb{E}\left[\hat{\underline{\mathbf{X}}}_s|\mathbf{Y}_s\right]^{\mathrm{T}}(\mathbb{E}\{\underline{\mathbf{X}}_s|\mathbf{Y}_s\}-\underline{\mathbf{Z}}_s)\Bigg)\\ &\overset{(b)}{=} \mathbb{E}\Bigg(\mathbb{E}\left[\hat{\underline{\mathbf{X}}}_s|\mathbf{Y}_s\right]^{\mathrm{T}}(\underline{\mathbf{Z}}_s-\underline{\mathbf{Z}}_s)\Bigg)\\ &= 0\end{aligned} \tag{252}$$

where (a) follows because $\underline{\mathbf{X}}_s \rightarrow \mathbf{Y}_s \rightarrow \hat{\underline{\mathbf{X}}}_s$ forms a Markov chain, and (b) follows from (123). By (251) and (252), $\mathbb{E}\left[(\underline{\mathbf{X}}_s-\underline{\mathbf{Z}}_s)^{\mathrm{T}}(\underline{\mathbf{Z}}_s-\hat{\underline{\mathbf{X}}}_s)\right]=0$.

D

MMSE Estimation Error in (126)

The average MMSE estimation error in (125) can be expressed as

$$\begin{aligned} D_{\mathbf{Z}|\mathbf{b}_s} &= N^{-1}\mathbb{E}[\|\underline{\mathbf{X}}_s - \underline{\mathbf{Z}}_s\|_2^2] \\ &\overset{(a)}{=} N^{-1}\mathbb{E}[\|\mathbf{X}_s - \mathbf{Z}_s\|_2^2] \\ &= N^{-1}\mathrm{Tr}\left\{\mathbb{E}[(\mathbf{X}_s - \mathbf{Z}_s)(\mathbf{X}_s - \mathbf{Z}_s)^{\mathrm{T}}]\right\} \\ &= N^{-1}\mathrm{Tr}\left\{\mathbb{E}[\mathbf{X}_s\mathbf{X}_s^{\mathrm{T}} - \mathbf{X}_s\mathbf{Z}_s^{\mathrm{T}} - \mathbf{Z}_s\mathbf{X}_s^{\mathrm{T}} + \mathbf{Z}_s\mathbf{Z}_s^{\mathrm{T}}]\right\} \\ &= N^{-1}\mathrm{Tr}\left\{\boldsymbol{\Sigma}_{\mathbf{X}_s} - \boldsymbol{\Sigma}_{\mathbf{X}_s\mathbf{Z}_s} - \boldsymbol{\Sigma}_{\mathbf{X}_s\mathbf{Z}_s}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{Z}_s}\right\} \end{aligned} \tag{253}$$

where (a) follows from removing the zero parts of $\underline{\mathbf{X}}_s$ and $\underline{\mathbf{Z}}_s$ (see (121) and (124)), and the cross-covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}_s\mathbf{Z}_s} \in \mathbb{R}^{K\times K}$ is

$$\begin{aligned} \boldsymbol{\Sigma}_{\mathbf{X}_s\mathbf{Z}_s} &= \mathbb{E}[\mathbf{X}_s(\mathbf{F}_s\mathbf{Y}_s)^{\mathrm{T}}] = \boldsymbol{\Sigma}_{\mathbf{X}_s\mathbf{Y}_s}\mathbf{F}_s^{\mathrm{T}} = \boldsymbol{\Sigma}_{\mathbf{X}_s\mathbf{Y}_s}\boldsymbol{\Sigma}_{\mathbf{Y}_s}^{-1}\boldsymbol{\Sigma}_{\mathbf{X}_s\mathbf{Y}_s}^{\mathrm{T}} \\ &= \boldsymbol{\Sigma}_{\mathbf{Z}_s}. \end{aligned} \tag{254}$$

Substituting (254) into (253) results in $D_{\mathbf{Z}|\mathbf{b}_s} = N^{-1}\mathrm{Tr}(\boldsymbol{\Sigma}_{\mathbf{X}_s} - \boldsymbol{\Sigma}_{\mathbf{Z}_s})$.

E

The Proof of Theorem 1

Using the equivalence $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{b}_s}(D_s) = R^{\mathrm{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s)$ of Proposition 6, and $D'_s = D_s - D_{\mathbf{Z}|\mathbf{b}_s} \geq 0$ in (129), the conditional remote RDF in (119) can be recast as

$$R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D) = \min_{\substack{\sum_{s=1}^{\binom{N}{K}} p(\mathbf{b}_s)[D'_s + D_{\mathbf{Z}|\mathbf{b}_s}] = D \\ D'_s \geq 0,\, s=1,\ldots,\binom{N}{K}}} \sum_{s=1}^{\binom{N}{K}} p(\mathbf{b}_s) R^{\mathrm{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s) \tag{255}$$

with optimization variables D'_s, $s = 1, \ldots, \binom{N}{K}$. Let $D'_{s,k} \geq 0$, $k = 1, \ldots, K$, $s = 1, \ldots, \binom{N}{K}$, be auxiliary variables for (255). Inserting the new variables with constraint $\sum_{k=1} D'_{s,k} = D'_s$, and substituting $D_{\mathbf{Z}|\mathbf{B}} = \sum_{s=1}^{\binom{N}{K}} p(\mathbf{b}_s) D_{\mathbf{Z}|\mathbf{b}_s}$ in (133) and the expression of $R^{\mathrm{dir}}_{\mathbf{Z}|\mathbf{b}_s}(D'_s)$ in (131), $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ in (255) can be equivalently expressed as

$$R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D) = \min_{\substack{\sum_{s=1}^{\binom{N}{K}} p(\mathbf{b}_s) D'_s = D - D_{\mathbf{Z}|\mathbf{B}} \\ \sum_{k=1}^{K} D'_{s,k} = D'_s,\, s=1,\ldots,\binom{N}{K} \\ D'_s \geq 0,\, s=1,\ldots,\binom{N}{K} \\ D'_{s,k} \geq 0,\, k=1,\ldots,K,\, s=1,\ldots,\binom{N}{K}}} N^{-1} \sum_{s=1}^{\binom{N}{K}} p(\mathbf{b}_s) \sum_{k=1}^{K} \max\left\{0, \frac{1}{2}\log\frac{\lambda_{s,k}}{D'_{s,k}}\right\} \tag{256}$$

with optimization variables D'_s, and $D'_{s,k}$, $k = 1, \ldots, K$, $s = 1, \ldots, \binom{N}{K}$. Finally, eliminating the variables D'_s, $s = 1, \ldots, \binom{N}{K}$, by substituting the second set of equality constraints into the first one yields the expression for $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ in (134).

Remark E.1. *A valid distortion is $D \geq D_{\mathbf{Z}|\mathbf{B}} \geq 0$ because the distortion criterion must be non-negative.*

Remark E.2. *For all distortion criteria $D \geq \frac{1}{N} \sum_{s=1}^{\binom{N}{K}} p(\mathbf{b}_s) \mathrm{Tr}(\mathbf{\Sigma}_{\mathbf{X}_s})$, $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D) = 0$; if the encoder sends no information (i.e., $R = 0$),* *then the decoder can set* $\hat{\mathbf{X}} = \mathbf{0}_N$*, resulting in an admissible distortion because* $\mathbb{E}[d(\mathbf{X}, \hat{\mathbf{X}})] = \frac{1}{N}\mathbb{E}[\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2] = \frac{1}{N}\mathbb{E}[\|\mathbf{X}\|_2^2] = \frac{1}{N}\sum_{s=1}^{\binom{N}{K}} p(\mathbf{b}_s)\mathrm{Tr}(\mathbf{\Sigma}_{\mathbf{X}_s}) \leq D$*.*

Combining the above derivations with Remarks E.1 and E.2, $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$ has the characterization of Theorem 1, which concludes the proof. ■

F

The Conditional Direct RDF

The conditional direct RDF $R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D)$ determines the minimum achievable rate R for distortion $D \geq 0$ in the compression scheme, where the encoder observes $\mathbf{X}$ directly (i.e., without CS), and $\mathbf{B}$ is available as SI at the encoder and decoder. It is defined as (cf. (118))

$$R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D) = \min_{\{f(\hat{\mathbf{x}}|\mathbf{x},\mathbf{b}_s)\}_{s=1}^{|\mathcal{B}|}:\mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})]\leq D} \frac{1}{N} I(\mathbf{X};\hat{\mathbf{X}}|\mathbf{B}) \tag{257a}$$

where the optimization is over the conditional PDFs $f(\hat{\mathbf{x}}|\mathbf{x},\mathbf{b}_s)$, $s = 1,\ldots,|\mathcal{B}|$. The conditional mutual information between $\mathbf{X}$ and $\hat{\mathbf{X}}$ given $\mathbf{B}$ is

$$\begin{aligned} I(\mathbf{X};\hat{\mathbf{X}}|\mathbf{B}) &= \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) I(\mathbf{X};\hat{\mathbf{X}}|\mathbf{B}=\mathbf{b}_s) \\ &= \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) \displaystyle\int_{\mathbf{x}}\int_{\hat{\mathbf{x}}} f(\mathbf{x}|\mathbf{b}_s) f(\hat{\mathbf{x}}|\mathbf{x},\mathbf{b}_s) \log\frac{f(\hat{\mathbf{x}}|\mathbf{x},\mathbf{b}_s)}{f(\hat{\mathbf{x}}|\mathbf{b}_s)} \mathrm{d}\mathbf{x}\mathrm{d}\hat{\mathbf{x}} \end{aligned} \tag{257b}$$

and the average MSE distortion between $\mathbf{X}$ and $\hat{\mathbf{X}}$ is

$$\begin{aligned} \mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})] &= \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) \mathbb{E}[d(\mathbf{X},\hat{\mathbf{X}})|\mathbf{B}=\mathbf{b}_s] \\ &= \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) \displaystyle\int_{\mathbf{x}}\int_{\hat{\mathbf{x}}} f(\mathbf{x}|\mathbf{b}_s) f(\hat{\mathbf{x}}|\mathbf{x},\mathbf{b}_s) d(\mathbf{x},\hat{\mathbf{x}}) \mathrm{d}\mathbf{x}\mathrm{d}\hat{\mathbf{x}}. \end{aligned} \tag{257c}$$

Following the steps analogous to those for $R^{\mathrm{rem}}_{\mathbf{X}|\mathbf{B}}(D)$, the conditional direct RDF is given as (cf. (134))

$$R^{\mathrm{dir}}_{\mathbf{X}|\mathbf{B}}(D) = \min_{\substack{\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) \sum_{k=1}^{K} D_{s,k} = D \\ D_{s,k}\geq 0,\, k=1,\ldots,K,\, s=1,\ldots,|\mathcal{B}|}} N^{-1} \textstyle\sum_{s=1}^{|\mathcal{B}|} p(\mathbf{b}_s) \sum_{k=1}^{K} \max\left\{0, \dfrac{1}{2}\log\dfrac{\bar{\lambda}_{s,k}}{D_{s,k}}\right\} \tag{258}$$

where $D_{s,k}$, $k = 1,\ldots,K$, $s = 1,\ldots,|\mathcal{B}|$, are the optimization variables; $\bar{\lambda}_{s,1} \geq \ldots \geq \bar{\lambda}_{s,K} > 0$ are the eigenvalues of covariance matrix $\mathbf{\Sigma}_{\mathbf{X}_s} = \bar{\mathbf{Q}}_s \bar{\mathbf{\Lambda}}_s \bar{\mathbf{Q}}_s^{\mathrm{T}}$, where the columns of $\bar{\mathbf{Q}}_s \in \mathbb{R}^{K\times K}$ are the eigenvectors of $\mathbf{\Sigma}_{\mathbf{X}_s} \in \mathbb{S}^K_{++}$, and $\bar{\mathbf{\Lambda}}_s \triangleq \mathrm{diag}(\bar{\lambda}_{s,1},\ldots,\bar{\lambda}_{s,K})$.

G

Proof of Proposition 7

A stationary point $\bar{\mathbf{P}}$, $\bar{\mathbf{Q}}$ and $\bar{\mathbf{E}}$ of (P2) must satisfy the following first-order optimality conditions [34]

$$\begin{aligned}
&\mathbf{0}_{N_x\times N_y} \in \partial_{\mathbf{E}} f(\bar{\mathbf{P}},\bar{\mathbf{Q}},\bar{\mathbf{E}}) = \\
&\left\{\tilde{f}(\bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}},\bar{\mathbf{E}})\ + \mu\bar{\beta}\left[\mathrm{sgn}(\bar{\mathbf{E}})+\tilde{\mathbf{E}}\right] \middle| \bar{\mathbf{E}}\odot\tilde{\mathbf{E}}=\mathbf{0},\ \|\tilde{\mathbf{E}}\|_\infty \le 1\right\}
\end{aligned} \tag{259}$$

$$\nabla_{\mathbf{P}} f(\bar{\mathbf{P}},\bar{\mathbf{Q}},\bar{\mathbf{E}}) = \tilde{f}(\bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}},\bar{\mathbf{E}})\bar{\mathbf{Q}} + \lambda\bar{\beta}\bar{\mathbf{P}} = \mathbf{0}_{N_x\times\rho} \tag{260}$$

$$\nabla_{\mathbf{Q}^{\mathcal{T}}} f(\bar{\mathbf{P}},\bar{\mathbf{Q}},\bar{\mathbf{E}}) = \bar{\mathbf{P}}^{\mathcal{T}}\tilde{f}(\bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}},\bar{\mathbf{E}}) + \lambda\bar{\beta}\bar{\mathbf{Q}}^{\mathcal{T}} = \mathbf{0}_{\rho\times N_y} \tag{261}$$

where $\odot$ denotes the element-wise (Hadamard) product. Through post-multiplying (260) by $\bar{\mathbf{P}}^{\mathcal{T}}$ and pre-multiplying (261) by $\bar{\mathbf{Q}}$, one can see that

$$\begin{aligned}
\tilde{f}(\bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}},\bar{\mathbf{E}}) &= -\mu\bar{\beta}(\mathrm{sgn}(\bar{\mathbf{E}})+\tilde{\mathbf{E}}) \\
\mathrm{tr}\left(\tilde{f}(\bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}},\bar{\mathbf{E}})\bar{\mathbf{Q}}\bar{\mathbf{P}}^{\mathcal{T}}\right) &= -\lambda\bar{\beta}\mathrm{tr}(\bar{\mathbf{P}}\bar{\mathbf{P}}^{\mathcal{T}}) = -\lambda\bar{\beta}\mathrm{tr}(\bar{\mathbf{Q}}\bar{\mathbf{Q}}^{\mathcal{T}}).
\end{aligned} \tag{262}$$

Define now $\kappa(\mathbf{R}_1,\mathbf{R}_2) := \frac{1}{2}\left(\mathrm{tr}(\mathbf{R}_1)+\mathrm{tr}(\mathbf{R}_2)\right)$, and consider the following *convex* problem

$$\begin{aligned}
\text{(P4)}\quad &\min_{\substack{\mathbf{L},\mathbf{E}\in\mathbb{R}^{N_x\times N_y},\\ \mathbf{R}_1\in\mathbb{R}^{N_x\times N_x},\\ \mathbf{R}_2\in\mathbb{R}^{N_y\times N_y}}} \sum_{\tau=1}^{T}\beta^{T-\tau}c^{(\tau)}(\mathbf{L},\mathbf{E}) + \lambda\bar{\beta}\,\kappa(\mathbf{R}_1,\mathbf{R}_2) + \mu\bar{\beta}\,||\mathbf{E}||_1 \\
&\text{subject to } \mathbf{R} := \begin{pmatrix}\mathbf{R}_1 & \mathbf{L}\\ \mathbf{L}^{\mathcal{T}} & \mathbf{R}_2\end{pmatrix} \succeq \mathbf{0}
\end{aligned} \tag{263}$$

which is *equivalent* to (P1). Equivalence can be easily inferred by minimizing (P4) with respect to $\{\mathbf{R}_1,\mathbf{R}_2\}$ and noting an alternative characterization of the nuclear norm given by [245]

$$\begin{aligned}
\|\mathbf{L}\|_* = &\min_{\mathbf{R}_1,\mathbf{R}_2} \kappa(\mathbf{R}_1,\mathbf{R}_2) \\
&\text{subject to } \mathbf{R}\succeq\mathbf{0}.
\end{aligned} \tag{264}$$

In what follows, the optimality conditions of the conic program (P4) are explored. Introducing a Lagrange multiplier matrix $\mathbf{M} \in \mathbb{R}^{(N_x+N_y)\times(N_x+N_y)}$ associated with the conic constraint in (263), the Lagrangian is first formed as

$$\begin{aligned}\mathcal{L}(\mathbf{L},\mathbf{E},\mathbf{R}_1,\mathbf{R}_2;\mathbf{M}) = \sum_{\tau=1}^{T} \beta^{T-\tau} c^{(\tau)}(\mathbf{L},\mathbf{E})\\ + \lambda\bar{\beta}\,\kappa(\mathbf{R}_1,\mathbf{R}_2) + \mu\bar{\beta}\,||\mathbf{E}||_1 - \langle \mathbf{M},\mathbf{R}\rangle. \end{aligned} \tag{265}$$

For notational convenience, partition $\mathbf{M}$ as

$$\mathbf{M} := \begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_4 & \mathbf{M}_3 \end{pmatrix} \tag{266}$$

in accordance with the block structure of $\mathbf{R}$ in (263), where $\mathbf{M}_1 \in \mathbb{R}^{N_x\times N_x}$ and $\mathbf{M}_3 \in \mathbb{R}^{N_y\times N_y}$. The optimal solution to (P4) must satisfy: (i) the stationarity conditions

$$\nabla_{\mathbf{L}}\mathcal{L}(\mathbf{L},\mathbf{E},\mathbf{R}_1,\mathbf{R}_2;\mathbf{M}) = \tilde{f}(\mathbf{L},\mathbf{E}) - \mathbf{M}_2 - \mathbf{M}_4^{\mathcal{T}} = \mathbf{0} \tag{267}$$

$$\begin{aligned}&\mathbf{0} \in \partial_{\mathbf{E}}\mathcal{L}(\mathbf{L},\mathbf{E},\mathbf{R}_1,\mathbf{R}_2;\mathbf{M}) =\\ &\quad\left\{\tilde{f}(\mathbf{L},\mathbf{E}) + \mu\bar{\beta}\left[\mathrm{sgn}(\mathbf{E}) + \tilde{\mathbf{E}}\right] \middle| \mathbf{E}\odot\tilde{\mathbf{E}} = \mathbf{0},\ \|\tilde{\mathbf{E}}\|_\infty \le 1\right\}\end{aligned} \tag{268}$$

$$\nabla_{\mathbf{R}_1}\mathcal{L}(\mathbf{L},\mathbf{E},\mathbf{R}_1,\mathbf{R}_2;\mathbf{M}) = \frac{\lambda\bar{\beta}}{2}\mathbf{I}_{N_x} - \mathbf{M}_1 = \mathbf{0} \tag{269}$$

$$\nabla_{\mathbf{R}_2}\mathcal{L}(\mathbf{L},\mathbf{E},\mathbf{R}_1,\mathbf{R}_2;\mathbf{M}) = \frac{\lambda\bar{\beta}}{2}\mathbf{I}_{N_y} - \mathbf{M}_3 = \mathbf{0} \tag{270}$$

(ii) complementary slackness condition $\langle\mathbf{M},\mathbf{R}\rangle = 0$; (iii) primal feasibility $\mathbf{R} \succeq \mathbf{0}$; and (iv) dual feasibility $\mathbf{M} \succeq \mathbf{0}$.

Using the stationary point $\bar{\mathbf{P}}$, $\bar{\mathbf{Q}}$ and $\bar{\mathbf{E}}$ of (P2), construct a candidate solution for (P4) as $\hat{\mathbf{L}} := \bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}}$, $\hat{\mathbf{E}} := \bar{\mathbf{E}}$, $\hat{\mathbf{R}}_1 := \bar{\mathbf{P}}\bar{\mathbf{P}}^{\mathcal{T}}$, and $\hat{\mathbf{R}}_2 := \bar{\mathbf{Q}}\bar{\mathbf{Q}}^{\mathcal{T}}$, as well as $\hat{\mathbf{M}}_1 := \frac{\lambda\bar{\beta}}{2}\mathbf{I}_{N_x}$, $\hat{\mathbf{M}}_2 := \frac{1}{2}\tilde{f}(\bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}},\bar{\mathbf{E}})$, $\hat{\mathbf{M}}_3 := \frac{\lambda\bar{\beta}}{2}\mathbf{I}_{N_y}$, and $\hat{\mathbf{M}}_4 := \hat{\mathbf{M}}_2^{\mathcal{T}}$. After substituting these into (267)–(270), it can be readily verified that condition (i) holds. Condition (ii) also holds since

$$\begin{aligned}\langle\hat{\mathbf{M}},\hat{\mathbf{R}}\rangle &= \langle\hat{\mathbf{M}}_1,\hat{\mathbf{R}}_1\rangle + \langle\hat{\mathbf{M}}_2,\hat{\mathbf{L}}\rangle + \langle\hat{\mathbf{M}}_3,\hat{\mathbf{R}}_2\rangle + \langle\hat{\mathbf{M}}_4,\hat{\mathbf{L}}^{\mathcal{T}}\rangle\\ &= \frac{\lambda\bar{\beta}}{2}\mathrm{tr}(\bar{\mathbf{P}}\bar{\mathbf{P}}^{\mathcal{T}} + \bar{\mathbf{Q}}\bar{\mathbf{Q}}^{\mathcal{T}}) + \mathrm{tr}\left(\tilde{f}(\bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}},\bar{\mathbf{E}})\bar{\mathbf{Q}}\bar{\mathbf{P}}^{\mathcal{T}}\right)\\ &= \mathbf{0}\end{aligned} \tag{271}$$

where the last equality follows from (262). Condition (iii) is met since $\mathbf{R}$ can be rewritten as

$$\mathbf{R} = \begin{pmatrix} \bar{\mathbf{P}}\bar{\mathbf{P}}^{\mathcal{T}} & \bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}} \\ \bar{\mathbf{Q}}\bar{\mathbf{P}}^{\mathcal{T}} & \bar{\mathbf{Q}}\bar{\mathbf{Q}}^{\mathcal{T}} \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{P}} \\ \bar{\mathbf{Q}} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{P}} \\ \bar{\mathbf{Q}} \end{pmatrix}^{\mathcal{T}} \succeq \mathbf{0}. \tag{272}$$

For (iv), according to the Schur complement condition for positive semidefinite matrices, $\mathbf{M} \succeq \mathbf{0}$ holds if and only if

$$\hat{\mathbf{M}}_3 - \hat{\mathbf{M}}_4 \hat{\mathbf{M}}_1^{-1} \hat{\mathbf{M}}_2 \succeq \mathbf{0} \tag{273}$$

which is equivalent to $\lambda_{\max}(\hat{\mathbf{M}}_2^{\mathcal{T}} \hat{\mathbf{M}}_2) \leq (\lambda\bar{\beta}/2)^2$, or $||\tilde{f}(\bar{\mathbf{P}}\bar{\mathbf{Q}}^{\mathcal{T}}, \bar{\mathbf{E}})|| \leq \lambda\bar{\beta}$. ■

H

Proof of Proposition 8

The proof uses the technique similar to the one employed in [199], where the convergence of online algorithms for optimizing objectives involving non-convex bilinear terms and sparse matrices was established in the context of dictionary learning.

In order to proceed with the proof, three lemmata are first established. The first lemma concerns some properties of $g(\mathbf{X}, \boldsymbol{\xi}^{(t)}) := g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}) + g_2(\mathbf{X})$, and $\breve{g}(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) := \breve{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) + g_2(\mathbf{X})$.

Lemma 4: *If the assumptions* (a1)–(a5) *in Proposition 8 hold, then*

(p1) $\breve{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)})$ *majorizes* $g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)})$*, i.e.,* $\breve{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \geq g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}) \; \forall \, \mathbf{X} \in \mathcal{X}'$*;*

(p2) $\breve{g}_1$ *is locally tight, i.e.,* $\breve{g}_1(\mathbf{X}^{(t-1)}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) = g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)})$*;*

(p3) $\nabla \breve{g}_1(\mathbf{X}^{(t-1)}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) = \nabla g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)})$*;*

(p4) $\breve{g}(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) := \breve{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) + g_2(\mathbf{X})$ *is uniformly strongly convex in* $\mathbf{X}$*, i.e.,* $\forall (\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \in \mathcal{X} \times \mathcal{X} \times \Xi$*, it holds that*

$$\begin{aligned} &\breve{g}(\mathbf{X} + \mathbf{D}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) - \breve{g}(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \\ &\qquad\qquad \geq \breve{g}'(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}; \mathbf{D}) + \frac{\zeta}{2} ||\mathbf{D}||_F^2 \end{aligned}$$

where $\zeta > 0$ *is a constant and* $\breve{g}'(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}; \mathbf{D})$ *is a directional derivative of* $\breve{g}$ *at* $\mathbf{X}$ *along the direction* $\mathbf{D}$*;*

(p5) g_1 *and* $\breve{g}_1$*, their derivatives, and their Hessians are uniformly bounded;*

(p6) g_2 *and its directional derivative* g_2' *are uniformly bounded; and*

(p7) *there exists* $\bar{g} \in \mathbb{R}$ *such that* $|\breve{g}(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)})| \leq \bar{g}$*.*

Proof: For (p1), let us first show that $\nabla_{\mathbf{P}} g_1(\mathbf{P}, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)})$, $\nabla_{\mathbf{Q}} g_1(\mathbf{P}, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)})$, and $\nabla_{\mathbf{E}} g_1(\mathbf{P}, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)})$ are Lipschitz continuous for $\mathbf{X} := (\mathbf{P}, \mathbf{Q}, \mathbf{E}) \in \mathcal{X}'$ and $\boldsymbol{\xi}^{(t)} \in \Xi$. For arbitrary $\mathbf{X}_1 := (\mathbf{P}_1, \mathbf{Q}_1, \mathbf{E}_1)$, $\mathbf{X}_2 := (\mathbf{P}_2, \mathbf{Q}_2, \mathbf{E}_2) \in \mathcal{X}'$, the variation of ∇g_1 in (177) can be bounded as

$$\begin{aligned}
&\|\nabla_{\mathbf{P}} g_1(\mathbf{P}_1, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)}) - \nabla_{\mathbf{P}} g_1(\mathbf{P}_2, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)})\|_F \\
&\qquad\qquad = \left\| \sum_{m=1}^{M} \langle \mathbf{W}_m^{(t)}, (\mathbf{P}_1 - \mathbf{P}_2)\mathbf{Q}^{\mathcal{T}} \rangle \mathbf{W}_m^{(t)} \mathbf{Q} \right\|_F \\
&\qquad\qquad \overset{\text{(i1)}}{\leq} \sum_{m=1}^{M} |\langle \mathbf{W}_m^{(t)}, (\mathbf{P}_1 - \mathbf{P}_2)\mathbf{Q}^{\mathcal{T}} \rangle| \|\mathbf{W}_m^{(t)} \mathbf{Q}\|_F \\
&\qquad\qquad \overset{\text{(i2)}}{\leq} \sum_{m=1}^{M} \|\mathbf{P}_1 - \mathbf{P}_2\|_F \, \|\mathbf{W}_m^{(t)} \mathbf{Q}\|_F^2
\end{aligned}$$

where (i1) and (i2) are due to the triangle and Cauchy-Schwarz inequalities, respectively. Since $\mathcal{X}'$ and Ξ are assumed to be bounded, $\sum_{m=1}^{M} \|\mathbf{W}_m^{(t)} \mathbf{Q}\|_F^2$ is bounded. Therefore, there exists a positive constant $L_{\mathbf{P}}$ such that

$$\|\nabla_{\mathbf{P}} g_1(\mathbf{P}_1, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)}) - \nabla_{\mathbf{P}} g_1(\mathbf{P}_2, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)})\|_F \leq L_{\mathbf{P}} \|\mathbf{P}_1 - \mathbf{P}_2\|_F \tag{274}$$

meaning that $\nabla_{\mathbf{P}} g_1(\mathbf{P}, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)})$ is Lipschitz continuous with constant $L_{\mathbf{P}}$. Similar arguments hold for $\nabla_{\mathbf{Q}} g_1(\mathbf{P}, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)})$ and $\nabla_{\mathbf{E}} g_1(\mathbf{P}, \mathbf{Q}, \mathbf{E}, \boldsymbol{\xi}^{(t)})$ as well, with Lipschitz constants $L_{\mathbf{Q}}$ and $L_{\mathbf{E}}$, respectively. Then, upon defining $\|\mathbf{X}\|_\Delta := \sqrt{L_{\mathbf{P}}^2 \|\mathbf{P}\|_F^2 + L_{\mathbf{Q}}^2 \|\mathbf{Q}\|_F^2 + L_{\mathbf{E}}^2 \|\mathbf{E}\|_F^2}$, it is easy to verify

$$\|\nabla g_1(\mathbf{X}_1, \boldsymbol{\xi}^{(t)}) - \nabla g_1(\mathbf{X}_2, \boldsymbol{\xi}^{(t)})\|_F \leq \|\mathbf{X}_1 - \mathbf{X}_2\|_\Delta. \tag{275}$$

On the other hand, the proof of the Descent Lemma [29] can be adopted to show

$$\begin{aligned}
&g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}) - g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \\
&\leq \langle \mathbf{X} - \mathbf{X}^{(t-1)}, \nabla g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \rangle + \int_0^1 \|\mathbf{X} - \mathbf{X}^{(t-1)}\|_F \\
&\times \|\nabla g_1(\mathbf{X}^{(t-1)} + \alpha(\mathbf{X} - \mathbf{X}^{(t-1)}), \boldsymbol{\xi}^{(t)}) \\
&\qquad\qquad\qquad - \nabla g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)})\|_F d\alpha.
\end{aligned} \tag{276}$$

Note that

$$\|\mathbf{X}\|_F \leq \frac{1}{L_{\min}} \|\mathbf{X}\|_\Delta \tag{277}$$

where $L_{\min} := \min\{L_{\mathbf{P}}, L_{\mathbf{Q}}, L_{\mathbf{E}}\}$. Then, substitution of (275) into (276) with $\mathbf{X}_1 = \mathbf{X}^{(t-1)} + \alpha(\mathbf{X} - \mathbf{X}^{(t-1)})$ and $\mathbf{X}_2 = \mathbf{X}^{(t-1)}$ yields

$$\begin{aligned} g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) &+ \langle \mathbf{X} - \mathbf{X}^{(t-1)}, \nabla g_1(\mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \rangle \\ &+ \frac{1}{2L_{\min}} \|\mathbf{X} - \mathbf{X}^{(t-1)}\|_{\Delta}^2 \geq g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}) \end{aligned} \tag{278}$$

which completes the proof by the construction of $\breve{g}_1$, provided that $\eta_i^{(t)} \geq L_i^2 / L_{\min}$ for all $i \in \{\mathbf{P}, \mathbf{Q}, \mathbf{E}\}$.

To show (p2) and (p3), let us first denote

$$\nabla g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}) = (\nabla_{\mathbf{P}} g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}), \nabla_{\mathbf{Q}} g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}), \nabla_{\mathbf{E}} g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)})) \tag{279}$$

$$\begin{aligned} \nabla \breve{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) = (&\nabla_{\mathbf{P}} g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}) + \eta_{\mathbf{P}}^{(t)} (\mathbf{P} - \mathbf{P}^{(t-1)}), \\ &\nabla_{\mathbf{Q}} g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}) + \eta_{\mathbf{Q}}^{(t)} (\mathbf{Q} - \mathbf{Q}^{(t-1)}), \\ &\nabla_{\mathbf{E}} g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)}) + \eta_{\mathbf{E}}^{(t)} (\mathbf{E} - \mathbf{E}^{(t-1)})). \end{aligned} \tag{280}$$

Then, it suffices to evaluate $\breve{g}_1(\mathbf{X}, \boldsymbol{\xi}^{(t)})$ and $\nabla \breve{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)})$ at $\mathbf{X}^{(t-1)}$ to see that (p2) and (p3) hold.

To show (p4), let us first find $\breve{g}_1'$ and g_2'. Along a direction $\mathbf{D} := (\mathbf{D}_{\mathbf{P}}, \mathbf{D}_{\mathbf{Q}}, \mathbf{D}_{\mathbf{E}}) \in \mathcal{X}'$, it holds that $\breve{g}_1'(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}; \mathbf{D}) = \langle \nabla \breve{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}), \mathbf{D} \rangle$ since $\breve{g}_1$ is differentiable. Similarly, $g_2'(\mathbf{X}; \mathbf{D}) = \lambda(\langle \mathbf{P}, \mathbf{D}_{\mathbf{P}} \rangle + \langle \mathbf{Q}, \mathbf{D}_{\mathbf{Q}} \rangle) + \mu h'(\mathbf{E}; \mathbf{D}_{\mathbf{E}})$ where $h(\mathbf{E}) := \|\mathbf{E}\|_1$, $\mathbf{d}_{\mathbf{E}} := \mathrm{vec}(\mathbf{D}_{\mathbf{E}})$ with its l-th entry being $d_{\mathbf{E},l}$, and

$$\begin{aligned} h'(\mathbf{E}; \mathbf{D}_{\mathbf{E}}) &:= \lim_{t \to 0+} \frac{h(\mathbf{E} + t\mathbf{D}_{\mathbf{E}}) - h(\mathbf{E})}{t} \\ &= \lim_{t \to 0+} \frac{\sum_{l, e_l \neq 0} (|e_l + t d_{\mathbf{E},l}| - |e_l|) + \sum_{l, e_l = 0} |t d_{\mathbf{E},l}|}{t} \\ &= \sum_{l, e_l \neq 0} \mathrm{sgn}(e_l) d_{\mathbf{E},l} + \sum_{l, e_l = 0} |d_{\mathbf{E},l}|. \end{aligned} \tag{281}$$

On the other hand, the variation of $\breve{g}$ can be written as

$$\begin{aligned} &\breve{g}(\mathbf{X} + \mathbf{D}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) - \breve{g}(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \\ &= \breve{g}_1'(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}; \mathbf{D}) + \sum_{i \in \{\mathbf{P}, \mathbf{Q}, \mathbf{E}\}} \frac{\eta_i^{(t)}}{2} \|\mathbf{D}_i\|_F^2 + g_2(\mathbf{X} + \mathbf{D}) - g_2(\mathbf{X}). \end{aligned} \tag{282}$$

Note that $\sum_i \frac{\eta_i^{(t)}}{2}\|\mathbf{D}_i\|_F^2 \geq \frac{L_{\min}}{2}||\mathbf{D}||_F^2$ since $\eta_i^{(t)} \geq L_i^2/L_{\min}$ by algorithmic construction. Furthermore, $g_2(\mathbf{X}+\mathbf{D}) - g_2(\mathbf{X}) \geq g_2'(\mathbf{X};\mathbf{D})$ since g_2 is convex [226]. Then, the variation of $\breve{g}$ in (282) can be lower-bounded as

$$\begin{aligned} &\breve{g}(\mathbf{X}+\mathbf{D}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) - \breve{g}(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \\ &\qquad \geq \breve{g}'(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}; \mathbf{D}) + \frac{L_{\min}}{2}||\mathbf{D}||_F^2 \end{aligned} \tag{283}$$

where $\breve{g}'(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}; \mathbf{D}) = \breve{g}_1'(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}; \mathbf{D}) + g_2'(\mathbf{X};\mathbf{D})$. Therefore, (p4) holds for a positive constant $\zeta \leq L_{\min}$.

By the compactness of $\mathcal{X}$ and boundedness of $\varXi$ by (a3), (p5) is automatically satisfied since g_1 and $\breve{g}_1$ are continuously twice differentiable in $\mathbf{X}$ [242]. In addition, one can easily show (p6) since g_2 and g_2' are also uniformly bounded by the compactness of $\mathcal{X}$.

Let K_1 and K_2 denote constants where $|\breve{g}_1| \leq K_1$ and $|g_2| \leq K_2$, respectively, by (p5) and (p6). Then, (p7) readily follows since

$$\begin{aligned} |\breve{g}(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)})| &= |\breve{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) + g_2(\mathbf{X})| \\ &\leq |\breve{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)})| + |g_2(\mathbf{X})| \\ &\leq K_1 + K_2 =: \bar{g}. \blacksquare \end{aligned} \tag{284}$$

The next lemma asserts that a distance between two subsequent estimates asymptotically goes to zero, which will be used to show $\lim_{t\to\infty} \check{C}_{1,t}(\mathbf{X}^{(t)}) - C_{1,t}(\mathbf{X}^{(t)}) = 0$, almost surely.

Lemma 5: *If* (a2)–(a5) *hold, then* $||\mathbf{X}^{(t+1)} - \mathbf{X}^{(t)}||_F = \mathcal{O}(1/t)$.

Proof: See [242, Lemma 2]. A proof of Lemma 5 is omitted to avoid duplication of the proof of [242, Lemma 2]. Hence, it suffices to mention that Lemma 4 guarantees the formulation of the proposed work satisfying the general assumptions on the formulation in [242]. ■

Lemma 5 does not guarantee convergence of the iterates to the stationary point of (P2). However, the final lemma asserts that the overestimated cost sequence converges to the cost of (P2), almost surely. Before proceeding to the next lemma, let us first define

$$C_{1,t}(\mathbf{X}) := \frac{1}{t}\sum_{\tau=1}^{t} g_1(\mathbf{X}, \boldsymbol{\xi}^{(\tau)}) \tag{285}$$

$$\check{C}_{1,t}(\mathbf{X}) := \frac{1}{t}\sum_{\tau=1}^{t} \check{g}_1(\mathbf{X}, \mathbf{X}^{(\tau-1)}, \boldsymbol{\xi}^{(\tau)}) \tag{286}$$

and $C_2(\mathbf{X}) := g_2(\mathbf{X})$. Note also that $\check{C}_t(\mathbf{X}) - C_t(\mathbf{X}) = \check{C}_{1,t}(\mathbf{X}) - C_{1,t}(\mathbf{X})$.

Lemma 6: *If* (a1)–(a5) *hold,* $\check{C}_t(\mathbf{X}^{(t)})$ *converges almost surely, and* $\lim_{t\to\infty} \check{C}_{1,t}(\mathbf{X}^{(t)}) - C_{1,t}(\mathbf{X}^{(t)}) = 0$, *almost surely.*

Proof: See [242, Lemma 1]. A proof of Lemma 6 is omitted to avoid duplication of the proof of [242, Lemma 1]. Instead, a sketch of the proof is following. It is firstly shown that the sequence $\{\check{C}_t(\mathbf{X}^{(t)})\}_{t=1}^{\infty}$ follows a quasi-martingale process and converges almost surely. Then, the lemma on positive converging sums (see [199, Lemma 8]) and Lemma 3 are used to claim that $\lim_{t\to\infty} \check{C}_{1,t}(\mathbf{X}^{(t)}) - C_{1,t}(\mathbf{X}^{(t)}) = 0$, almost surely. ■

The last step of the proof for Proposition 8 is inspired by [242]. Based on Lemma 6, it will be shown that the sequence $\{\nabla\check{C}_{1,t}(\mathbf{X}^{(t)}) - \nabla C_{1,t}(\mathbf{X}^{(t)})\}_{t=1}^{\infty}$ goes to zero, almost surely. Together with C_2', it follows that $\lim_{t\to\infty} C_t'(\mathbf{X}^{(t)}; \mathbf{D}) \geq 0\ \forall \mathbf{D},$ a.s. by algorithmic construction, implying convergence of a sequence $\{\mathbf{X}^{(t)}\}_{t=1}^{\infty}$ to the set of stationary points of $C(\mathbf{X})$.

By the compactness of $\mathcal{X}$, it is always possible to find a convergent subsequence $\{\mathbf{X}^{(t)}\}_{t=1}^{\infty}$ to a limit point $\bar{\mathbf{X}} \in \mathcal{X}$. Then, by the strong law of large numbers [116] under (a1) and equicontinuity of a family of functions $\{C_{1,t}(\cdot)\}_{t=1}^{\infty}$ due to the uniform boundedness of ∇g_1 in (p5) [35], upon restricting to the subsequence, one can have

$$\lim_{t\to\infty} C_{1,t}(\mathbf{X}^{(t)}) = \mathbb{E}_{\boldsymbol{\xi}}[g_1(\bar{\mathbf{X}}, \boldsymbol{\xi})] =: C_1(\mathbf{X}). \tag{287}$$

Similarly, a family of functions $\{\check{C}_{1,t}(\cdot)\}_{t=1}^{\infty}$ is equicontinuous due to the uniform boundedness of $\nabla\check{g}_1$ in (p5). Furthermore, $\{\check{C}_{1,t}(\cdot)\}_{t=1}^{\infty}$ is pointwisely bounded by (a1)–(a3). Thus, Arzelá-Ascoli theorem (see [35, Cor. 2.5] and [96]) implies that there exists a uniformly continuous function $\check{C}_1(\mathbf{X})$ such that $\lim_{t\to\infty} \check{C}_{1,t}(\mathbf{X}) = \check{C}_1(\mathbf{X})\ \forall\ \mathbf{X} \in \mathcal{X}$ and after restricting to the subsequence

$$\lim_{t\to\infty} \check{C}_{1,t}(\mathbf{X}^{(t)}) = \check{C}_1(\bar{\mathbf{X}}). \tag{288}$$

Furthermore, since $\check{g}_1(\mathbf{X}, \mathbf{X}^{(t-1)}, \boldsymbol{\xi}^{(t)}) \geq g_1(\mathbf{X}, \boldsymbol{\xi}^{(t)})$ as in $(\mathrm{p}1)$, it follows that

$$\check{C}_{1,t}(\mathbf{X}) - C_{1,t}(\mathbf{X}) \geq 0 \, \forall \mathbf{X}. \tag{289}$$

By letting $t \to \infty$ on (289) and combining Lemma 6 with (287) and (288), one deduces that

$$\check{C}_1(\bar{\mathbf{X}}) - C_1(\bar{\mathbf{X}}) = 0, \text{ a.s.} \tag{290}$$

meaning that $\check{C}_{1,t}(\mathbf{X}) - C_{1,t}(\mathbf{X})$ takes its minimum at $\bar{\mathbf{X}}$ and

$$\nabla \check{C}_1(\bar{\mathbf{X}}) - \nabla C_1(\bar{\mathbf{X}}) = 0, \text{ a.s.} \tag{291}$$

by the first-order optimality condition.
On the other hand, the fact that $\mathbf{X}^{(t)}$ minimizes $\check{C}_t(\mathbf{X})$ by algorithmic construction and g_2' exists (so does C_2'), yields

$$\check{C}_{1,t}(\mathbf{X}^{(t)}) + C_2(\mathbf{X}^{(t)}) \leq \check{C}_{1,t}(\mathbf{X}) + C_2(\mathbf{X}) \quad \forall \mathbf{X} \in \mathcal{X} \tag{292}$$

and $\lim_{t\to\infty} \check{C}_{1,t}(\mathbf{X}^{(t)}) + C_2(\mathbf{X}^{(t)}) \leq \lim_{t\to\infty} \check{C}_{1,t}(\mathbf{X}) + C_2(\mathbf{X}) \, \forall \mathbf{X} \in \mathcal{X}$, which implies

$$\lim_{t\to\infty} \langle \nabla \check{C}_{1,t}(\mathbf{X}^{(t)}), \mathbf{D} \rangle + C_2'(\mathbf{X}^{(t)}; \mathbf{D}) \geq 0 \quad \forall \mathbf{D}. \tag{293}$$

Using the result in (291), (293) can be re-written as $\langle \nabla C_1(\bar{\mathbf{X}}), \mathbf{D} \rangle + C_2'(\bar{\mathbf{X}}; \mathbf{D}) \geq 0 \, \forall \mathbf{D}, \text{ a.s.}$ or

$$C'(\bar{\mathbf{X}}; \mathbf{D}) \geq 0 \quad \forall \mathbf{D}, \text{ a.s.} \tag{294}$$

Thus, the subsequence $\{\mathbf{X}^{(t)}\}_{t=1}^{\infty}$ asymptotically coincides with the set of stationary points of $C(\mathbf{X})$. ■

I

Proof of (215)

To solve the optimization problem (214), consider first the unconstrained solution to the LS part of the cost, which can be written as

$$b_k^{\mathrm{LS}} := \left(y'_k - \textstyle\sum_{\ell=k+1}^{K} R_{k\ell}\hat{b}_\ell^{\mathrm{DDD}}\right)/R_{kk}. \tag{295}$$

The detected symbol in (214) can be equivalently expressed as

$$\begin{aligned}\hat{b}_k^{\mathrm{DDD}} &= \arg\min_{b_k\in\mathcal{A}_a} f(b_k),\\ f(b_k) &:= \left(b_k^{\mathrm{LS}} - b_k\right)^2 + (2\lambda/R_{kk}^2)|b_k|_0.\end{aligned} \tag{296}$$

The solution $\hat{b}_k^{\mathrm{DDD}}$ can be obtained by comparing $f(0)$ with $\min_{b_k\in\mathcal{A}} f(b_k)$. Specifically, as the cost $f(b_k)$ is quadratic for $b_k \in \mathcal{A}$, the minimum is achieved at $f(\lfloor b_k^{\mathrm{LS}}\rceil)$, by quantizing b_k^{LS} to the nearest point in $\mathcal{A}$. Thus, $\hat{b}_k^{\mathrm{DDD}} = 0$ only if $f(0) \le f(\lfloor b_k^{\mathrm{LS}}\rceil)$, or equivalently, after using the definition of $f(\cdot)$, if $2b_k^{\mathrm{LS}}\lfloor b_k^{\mathrm{LS}}\rceil - \lfloor b_k^{\mathrm{LS}}\rceil^2 - 2\lambda/R_{kk}^2 \le 0$. This completes the proof of (215).

J

Proof of (216)

When $M = 2$, the DDD solution (215) reduces to

$$\hat{b}_k^{\mathrm{DDD}} = \mathrm{sign}(b_k^{\mathrm{LS}})\mathbb{1}(2|b_k^{\mathrm{LS}}| - 1 - 2\lambda/R_{kk}^2 > 0) \tag{297}$$

due to the fact that $\lfloor b_k^{\mathrm{LS}} \rceil = \mathrm{sign}(b_k^{\mathrm{LS}}) \in \{\pm 1\}$.
Recalling that $\check{\mathbf{b}}$ denotes the transmitted vector, and substituting $\mathbf{y} = \mathbf{H}\check{\mathbf{b}} + \mathbf{w}$ yields

$$\mathbf{y}' := \mathbf{Q}^T\mathbf{y} = \mathbf{R}\check{\mathbf{b}} + \mathbf{u} \tag{298}$$

where $\mathbf{u} := \mathbf{Q}^T\mathbf{w}$ is zero-mean Gaussian with identity covariance matrix. Supposing that there is no error propagation, the b_k^{LS} term in (297) becomes [cf. (295)]

$$\begin{aligned} b_k^{\mathrm{LS}} &= \left(\sum_{\ell=k}^{K} R_{k\ell}\check{b}_\ell + u_k - \sum_{\ell=k+1}^{K} R_{k\ell}\hat{b}_k^{\mathrm{DDD}}\right) \Big/ R_{kk} \\ &= \check{b}_\ell + u_k/R_{kk} \end{aligned} \tag{299}$$

where u_k denotes the k-th entry of $\mathbf{u}$.
To analyze the error probability for the detector in (297) consider the following three cases.
a) $\check{b}_k = 0$ is sent: under this case, $b_k^{\mathrm{LS}} = u_k/R_{kk}$ and a detection error emerges when $\hat{b}_k^{\mathrm{DDD}} = 1$ or -1. With the closed-form DDD detector (297) in mind, such an error occurs only if both $b_k^{\mathrm{LS}} \neq 0$ and $2|b_k^{\mathrm{LS}}| - 2\lambda/R_{kk}^2 - 1 > 0$ hold. The first case corresponds to $u_k \neq 0$ and the second one is equivalent to $|u_k| > |R_{kk}|/2 + \lambda/|R_{kk}|$, which is included in the event $u_k \neq 0$. Hence, to evaluate the error probability it suffices to consider only the case $|u_k| > |R_{kk}|/2 + \lambda/|R_{kk}|$.
b) $\check{b}_k = 1$ is sent: following the analysis in a), an error occurs if $\mathrm{sign}(R_{kk})u_k \leq -|R_{kk}|/2 + \lambda/|R_{kk}|$.

c) $\check{b}_k = -1$ is sent: following the analysis in a), an error occurs if $\operatorname{sign}(R_{kk})u_k \geq |R_{kk}|/2 - \lambda/|R_{kk}|$.

Given that the Gaussian distributed u_k has variance 1, the overall SER for the k-th entry b_k becomes

$$\begin{aligned} P_{e,k}^{\mathrm{DDD}} &= \textstyle\sum_{i=0,\pm 1} P(\text{error}|\check{b}_k = i)P(\check{b}_k = i) \\ &= 2(1-p_a)Q\left(\frac{|R_{kk}|}{2} + \frac{\lambda}{|R_{kk}|}\right) \\ &\qquad + p_a Q\left(\frac{|R_{kk}|}{2} - \frac{\lambda}{|R_{kk}|}\right) \ . \end{aligned} \tag{300}$$

References

[1] Agrawal, P. and N. Patwari. 2009. “Correlated Link Shadow Fading in Multi-Hop Wireless Networks”. *IEEE Trans. Wireless Commun.* 8(9): 4024–4036.

[2] Akaike, H. 1974. “A New Look at the Statistical Model Identification”. *IEEE Trans. Automat. Contr.*: 716–723.

[3] Akçakaya, M. and V. Tarokh. 2010. “Shannon-Theoretic Limits on Noisy Compressive Sampling”. *IEEE Trans. Inform. Theory*. 56(1): 492–504.

[4] Akyildiz, I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. “Wireless sensor networks: a survey”. *Computer Networks*. 38: 393–422.

[5] Akyildiz, I. F., M. C. Vuran, and Ö. B. Akan. 2004. “On Exploiting Spatial and Temporal Correlation in Wireless Sensor Networks”. In: *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*. Cambridge, UK. 71–80.

[6] Anastasi, G., M. Conti, M. D. Francesco, and A. Passarella. 2009. “Energy Conservation in Wireless Sensor Networks: A Survey”. *Ad Hoc Netw.* 7(3): 537–568.

[7] Angelosante, D., J. A. Bazerque, and G. B. Giannakis. 2010a. “Online Adaptive Estimation of Sparse Signals: Where RLS Meets the ℓ_1-Norm”. *IEEE Trans. Signal Processing*. 58(7): 3436–3447.

[8] Angelosante, D. and G. B. Giannakis. 2009. "RLS-Weighted Lasso for Adaptive Estimation of Sparse Signals". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Taipei, Taiwan.

[9] Angelosante, D., E. Grossi, G. B. Giannakis, and M. Lops. 2010b. "Sparsity-Aware Estimation of CDMA System Parameters". *EURASIP J. Advances Signal Processing*. 2010(June). Article ID 417981.

[10] Ariananda, D. D. and G. Leus. 2012. "Compressive Wideband Power Spectrum Estimation". *IEEE Trans. Signal Processing*. 60(9): 4775–4789.

[11] Arimoto, S. 1972. "An Algorithm for Computing the Capacity of Arbitrary Discrete Memoryless Channels". *IEEE Trans. Inform. Theory*. 18(1): 14–20.

[12] Asif, M. S. and J. Romberg. 2013. "Fast and Accurate Algorithms for Re-Weighted l_1-Norm Minimization". *IEEE Trans. Signal Processing*. 61(23): 5905–5916.

[13] Asif, M. S. and J. Romberg. 2014. "Sparse Recovery of Streaming Signals Using ℓ_1-Homotopy". *IEEE Trans. Signal Processing*. 62(16): 4209–4223.

[14] Bajwa, W., J. Haupt, A. Sayeed, and R. Nowak. 2007. "Joint Source-Channel Communication for Distributed Estimation in Sensor Networks". *IEEE Trans. Inform. Theory*. 53(10): 3629–3653.

[15] Baraniuk, R. G., V. Cevher, M. F. Duarte, and C. Hegde. 2010. "Model-Based Compressive Sensing". *IEEE Trans. Inform. Theory*. 56(4): 1982–2001.

[16] Baraniuk, R., M. Davenport, R. DeVore, and M. Wakin. 2008. "A Simple Proof of the Restricted Isometry Property for Random Matrices". *Constructive Approximation*. 28(3): 253–263.

[17] Baraniuk, R. 2007. "Compressive Sensing [Lecture Notes]". *IEEE Signal Processing Mag.* 24(4): 118–121.

[18] Baron, D., S. Sarvotham, and R. G. Baraniuk. 2010. "Bayesian Compressive Sensing Via Belief Propagation". *IEEE Trans. Signal Processing*. 58(1): 269–280.

[19] Baron, D., M. Wakin, M. Duarte, S. Sarvotham, and R. Baraniuk. 2005. "Distributed Compressed Sensing". *Tech. rep.* No. ECE-0612. Rice University, Houston, TX: Dept. Elect. Comput. Eng. URL: http://webee.technion.ac.il/people/drorb/pdf/DCS112005.pdf.

[20] Bazerque, J. A. and G. B. Giannakis. 2010. “Distributed Spectrum Sensing for Cognitive Radio Networks by Exploiting Sparsity”. *IEEE Trans. Signal Processing*. 58(3): 1847–1862.

[21] Bazerque, J. A., G. Mateos, and G. B. Giannakis. 2011. “Group-Lasso on Splines for spectrum Cartography”. *IEEE Trans. Signal Processing*. 59(10): 4648–4663.

[22] Beck, A., A. Ben-Tal, and M. Teboulle. 2006. “Finding a Global Optimal Solution for a Quadratically Constrained Fractional Quadratic Problem with Applications to the Regularized Total Least Squares”. *SIAM J. on Matrix Anal. and Appl.* 28(2): 425–445.

[23] Berger, C. R., S. Zhou, J. C. Preisig, and P. Willett. 2010. “Sparse Channel Estimation for Multicarrier Underwater Acoustic Communication: From Subspace Methods to Compressed Sensing”. *IEEE Trans. Signal Processing*. 58(3): 1708–1721.

[24] Berger, J., V. D. Oliveira, and B. Sanso. 2000. “Objective Bayesian Analysis of Spatially Correlated Data”. *Journal of the American Statistical Association*. 96(Dec.): 1361–1374.

[25] Berger, T. 1971. *Rate-Distortion Theory: A Mathematical Basis for Data Compression. Prentice-Hall Series in Information and System Sciences*. Prentice Hall. 352.

[26] Berger, T. and J. Gibson. 1998. “Lossy Source Coding”. *IEEE Trans. Inform. Theory*. 44(6): 2693–2723.

[27] Berinde, R., A. Gilbert, P. Indyk, H. Karloff, and M. Strauss. 2008. “Combining Geometry and Combinatorics: A Unified Approach to Sparse Signal Recovery”. In: *Proc. Annual Allerton Conf. Commun., Contr., Computing*. Illinois, USA. 798–805.

[28] Berinde, R. and P. Indyk. 2008. “Sparse Recovery Using Sparse Random Matrices”. *Tech. rep.* MIT.

[29] Bertsekas, D. P. 1999. *Nonlinear Programming*. 2nd. Belmont, Massachusetts, USA: Athena Scientific.

[30] Blahut, R. 1972. “Computation of Channel Capacity and Rate-Distortion Functions”. *IEEE Trans. Inform. Theory*. 18(4): 460–473.

[31] Blumensath, T. and M. E. Davies. 2008. “Iterative Thresholding for Sparse Approximations”. *Journal of Fourier Analysis and Applications*. 14(5): 629–654.

[32] Boufounos, P. T. 2012. “Universal Rate-Efficient Scalar Quantization”. *IEEE Trans. Inform. Theory*. 58(3): 1861–1872.

[33] Boyd, S. 2008. "EE363: Linear Dynamical Systems [lecture material 2008 – 2009], Stanford University". http://stanford.edu/class/ee363/index.html.

[34] Boyd, S. and L. Vandenberghe. 2004. *Convex Optimization*. Cambridge, UK: Cambridge University Press.

[35] Brown, R. F. 2004. *A Topological Introduction to Nonlinear Analysis*. Boston, MA: Birkhäuser.

[36] Caione, C., D. Brunelli, and L. Benini. 2012. "Distributed Compressive Sampling for Lifetime Optimization in Dense Wireless Sensor Networks". *IEEE Trans. Ind. Informat.* 8(1): 30–40.

[37] Caione, C., D. Brunelli, and L. Benini. 2014. "Compressive Sensing Optimization for Signal Ensembles in WSNs". *IEEE Trans. Ind. Informat.* 10(1): 382–392.

[38] Candés, E. J. 2006. "Compressive Sampling". In: *Proceedings of the International Congress of Mathematicians*. Vol. 3. Madrid, Spain. 1433–1452.

[39] Candés, E. J. 2008. "The Restricted Isometry Property and Its Implications for Compressed Sensing". *C.R. Acad. Sci. Paris, Ser. I.* 346(9-10): 589–592.

[40] Candès, E. J., X. Li, Y. Ma, and J. Wright. 2011. "Robust Principal Component Analysis?" *J. of the ACM*. 58(3, article no. 11).

[41] Candés, E. J. and J. Romberg. 2005. "l_1-MAGIC: Recovery of Sparse Signals via Convex Programming". http://users.ece.gatech.edu/~justin/l1magic/.

[42] Candés, E. J. and J. Romberg. 2007. "Sparsity and Incoherence in Compressive Sampling". *Inverse Problems*. 23(3): 969–985.

[43] Candés, E. J., J. Romberg, and T. Tao. 2006a. "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information". *IEEE Trans. Inform. Theory*. 52(2): 489–509.

[44] Candés, E. J., J. Romberg, and T. Tao. 2006b. "Stable Signal Recovery from Incomplete and Inaccurate Measurements". *Communications on Pure and Applied Mathematics*. 59(8): 1207–1223.

[45] Candés, E. J. and T. Tao. 2005. "Decoding by Linear Programming". *IEEE Trans. Inform. Theory*. 51(12): 4203–4215.

[46] Candés, E. J. and T. Tao. 2006. "Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?" *IEEE Trans. Inform. Theory*. 52(12): 5406–5425.

[47] Candés, E. J. and T. Tao. 2007. "The Dantzig Selector: Statistical Estimation When p Is Much Larger than n". *The Annals of Statistics*. 35(6): 2313–2351.

[48] Candés, E. J., M. Wakin, and S. Boyd. 2008. "Enhancing Sparsity by Reweighted l_1 Minimization". *Journal of Fourier Analysis and Applications*. 14(5-6): 877–905.

[49] Candés, E. J. and M. Wakin. 2008. "An Introduction To Compressive Sampling". *IEEE Signal Processing Mag.* 25(2): 21–30.

[50] Cardinal, J. and G. V. Assche. 2002. "Joint Entropy-Constrained Multiterminal Quantization". In: *Proc. IEEE Int. Symp. Inform. Theory*. Lausanne, Switzerland. 63.

[51] Carmi, A. Y., L. S. Mihaylova, and S. J. Godsill. 2014. *Compressed Sensing & Sparse Filtering*. first. *Signals and Communication Technology*. Springer-Verlag Berlin Heidelberg. 502.

[52] Carrillo, R. E. and K. E. Barner. 2009. "Iteratively Re-Weighted Least Squares for Sparse Signal Reconstruction from Noisy Measurements". In: *Proc. Conf. Inform. Sciences Syst. (CISS)*. Baltimore, MD. 448–453.

[53] Cetin, M. and B. M. Sadler. 2005. "Semi-Blind Sparse Channel Estimation with Constant Modulus Symbols". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Philadelphia, PN.

[54] Cevher, V., M. F. Duarte, and R. G. Baraniuk. 2008. "Distributed Target Localization via Spatial Sparsity". In: *Proc. European Sign. Proc. Conf.* Lausanne, Switzerland.

[55] Chae, D. H., P. Sadeghi, and R. A. Kennedy. 2010. "Effects of Basis-Mismatch in Compressive Sampling of Continuous Sinusoidal Signals". In: *Proc. of 2nd Intl. Conf. on Future Computer and Communication*. Vol. 2. Wuhan, China. 739–743.

[56] Chandrasekaran, V., B. Recht, P. A. Parrilo, and A. S. Willsky. 2012. "The Convex Geometry of Linear Inverse Problems". *Found. Comput. Math.* 12(6): 805–849.

[57] Chang, C. 2010. "On the Rate Distortion Function of Bernoulli Gaussian Sequences". In: *Proc. IEEE Int. Symp. Inform. Theory*. Austin, TX. 66–70.

[58] Chang, C.-I. and L. D. Davisson. 1988. "On Calculating the Capacity of an Infinite-Input Finite (Infinite)-Output Channel". *IEEE Trans. Inform. Theory*. 34(5): 1004–1010.

[59] Chartrand, R. 2007. "Exact Reconstruction of Sparse Signals via Nonconvex Minimization". *IEEE Signal Processing Lett.* 14(10): 707–710.

[60] Chartrand, R. and Y. Wotao. 2008. "Iteratively Reweighted Algorithms for Compressive Sensing". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Las Vegas, NV. 3869–3872.

[61] Chen, S., D. Donoho, and M. Saunders. 1998. "Atomic Decomposition by Basis Pursuit". *SIAM Journal on Scientific Computing*. 20(1): 33–61.

[62] Cheng, S., V. Stankovic, and Z. Xiong. 2005. "Computing the Channel Capacity and Rate-Distortion Function with Two-Sided State Information". *IEEE Trans. Inform. Theory*. 51(12): 4418–4425.

[63] Chepuri, S. P. and G. Leus. 2016. "Sparse Sensing for Statistical Inference". *Found. Trends Signal Proc.* 9(3–4): 233–368.

[64] Chi, Y., A. Pezeshki, L. Scharf, and R. Calderbank. 2010. "Sensitivity to Basis Mismatch in Compressed Sensing". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Dallas, TX.

[65] Chiang, M. and S. Boyd. 2004. "Geometric Programming Duals of Channel Capacity and Rate Distortion". *IEEE Trans. Inform. Theory*. 50(2): 245–258.

[66] Chou, P., T. Lookabaugh, and R. Gray. 1989. "Entropy-Constrained Vector Quantization". *IEEE Trans. Acoust., Speech, Signal Processing*. 37(1): 31–42.

[67] Coluccia, G., S. Kuiteing, A. Abrardo, M. Barni, and E. Magli. 2012. "Progressive Compressed Sensing and Reconstruction of Multidimensional Signals Using Hybrid Transform/Prediction Sparsity Model". *IEEE J. Emerg. Select. Topics Circuits Syst.* 2(3): 340–352.

[68] Coluccia, G., E. Magli, A. Roumy, and V. Toto-Zarasoa. 2011. "Lossy Compression of Distributed Sparse Sources: A Practical Scheme". In: *Proc. European Sign. Proc. Conf.* Barcelona, Spain. 422–426.

[69] Coluccia, G., A. Roumy, and E. Magli. 2014. "Operational Rate-Distortion Performance of Single-Source and Distributed Compressed Sensing". *IEEE Trans. Commun.* 62(6): 2022–2033.

[70] Cotter, S. F. and B. D. Rao. 2002a. "Application Of Total Least Squares (TLS) to the Design of Sparse Signal Representation Dictionaries". In: *Proc. Annual Asilomar Conf. Signals, Syst., Comp.* Pacific Grove, CA.

[71] Cotter, S. F. and B. D. Rao. 2002b. "Sparse Channel Estimation via Matching Pursuit with Application to Equalization". *IEEE Trans. Commun.* 50(3): 374–377.

[72] Cotter, S. F., B. D. Rao, K. Engan, and K. Kreutz-Delgado. 2005. "Sparse Solutions to Linear Inverse Problems with Multiple Measurement Vectors". *IEEE Trans. Signal Processing*. 53(7): 2477–2488.

[73] Cover, T. and M. Chiang. 2002. "Duality Between Channel Capacity and Rate Distortion with Two-Sided State Information". *IEEE Trans. Inform. Theory*. 48(6): 1629–1638.

[74] Cover, T. and J. Thomas. 2006. *Elements of Information Theory*. second. New York, USA: John Wiley.

[75] Cui, T. and C. Tellambura. 2005. "An Efficient Generalized Sphere Decoder for Rank-Deficient MIMO Systems". *IEEE Commun. Lett.* 9(May): 423–425.

[76] D. M. Sima, S. Van Huffel, and G. H. Golub. 2004. "Regularized Total Least Squares based on Quadratic Eigenvalue Problem Solvers". *BIT Numerical Mathematics*. 44(4): 793–812.

[77] Dai, W. and O. Milenkovic. 2009. "Subspace Pursuit for Compressive Sensing Signal Reconstruction". *IEEE Trans. Inform. Theory*. 55(5): 2230–2249.

[78] Dai, W. and O. Milenkovic. 2011. "Information Theoretical and Algorithmic Approaches to Quantized Compressive Sensing". *IEEE Trans. Commun.* 59(7): 1857–1866.

[79] Dall'Anese, E., S.-J. Kim, and G. B. Giannakis. 2011. "Channel Gain Map Tracking via Distributed Kriging". *IEEE Trans. Veh. Technol.* 60(3): 1205–1211.

[80] Damen, M. O., H. E. Gamal, and G. Caire. 2003. "On Maximum-Likelihood Detection and the Search for the Closest Lattice Point". *IEEE Trans. Inform. Theory*. 49(Oct.): 2389–2402.

[81] Davenport, M. A., M. F. Duarte, Y. C. Eldar, and G. Kutyniok. 2012. "Introduction to Compressed Sensing". In: *Compressed Sensing: Theory and Applications*. Ed. by Y. C. Eldar and G. Kutyniok. Cambridge University Press. Chap. 1. 1–64.

[82] Dinkelbach, W. 1967. "On Nonlinear Fractional Programming". *Management Science*. 13(7): 492–498.

[83] Dixon, A. M. R., E. G. Allstot, D. Gangopadhyay, and D. J. Allstot. 2012. "Compressed Sensing System Considerations for ECG and EMG Wireless Biosensors". *IEEE Trans. Biomed. Circ. Syst.* 6(2): 156–166.

[84] Do, T., L. Gan, N. Nguyen, and T. Tran. 2012. "Fast and Efficient Compressive Sensing Using Structurally Random Matrices". *IEEE Trans. Signal Processing*. 60(1): 139–154.

[85] Dobrushin, R. and B. Tsybakov. 1962. "Information Transmission with Additional Noise". *IRE Trans. Inform. Theory*. 8(5): 293–304.

[86] Dong, Y., S. Chang, and L. Carin. 2005. "Rate-Distortion Bound for Joint Compression and Classification with Application to Multiaspect Scattering". *IEEE Sensors J.* 5(3): 481–492.

[87] Donoho, D. L. 2006. "Compressed Sensing". *IEEE Trans. Inform. Theory*. 52(4): 1289–1306.

[88] Donoho, D. L., M. Elad, and V. N. Temlyakov. 2006a. "Stable Recovery of Sparse Overcomplete Representations in the Presence of Noise". *IEEE Trans. Inform. Theory*. 52(1): 6–18.

[89] Donoho, D. L., A. Maleki, and A. Montanari. 2009. "Message-Passing Algorithms for Compressed Sensing". *Proceedings of the National Academy of Sciences of the USA*. 106(45): 18914–18919.

[90] Donoho, D., A. Maleki, and M. Shahram. 2006b. "Wavelab 850: Wavelet Toolbox for Matlab". http://www-stat.stanford.edu/~wavelab.

[91] Dragotti, P. L. and M. Gastpar. 2009. *Distributed Source Coding: Theory, Algorithms and Applications*. Academic Press.

[92] Duarte, M. and R. Baraniuk. 2010. "Kronecker Product Matrices for Compressive Sensing". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Dallas, TX. 3650–3653.

[93] Duarte, M. and R. Baraniuk. 2012. "Kronecker Compressive Sensing". *IEEE Trans. Image Processing*. 21(2): 494–504.

[94] Duarte, M., M. Wakin, D. Baron, and R. Baraniuk. 2006. "Universal Distributed Sensing Via Random Projections". In: *Proc. IEEE Int. Symp. on Inform. Proc. in Sensor Networks*. Nashville, Tennessee, USA. 177–185.

[95] Duarte, M., M. Wakin, D. Baron, S. Sarvotham, and R. Baraniuk. 2013. "Measurement Bounds for Sparse Signal Ensembles via Graphical Models". *IEEE Trans. Inform. Theory*. 59(7): 4280–4289.

[96] Dunford, N. and J. T. Schwartz. 1958. *Linear Operators. Part 1: General Theory*. Newyork, NY: Interscience publishers.

[97] E. Viterbo and J. Boutros. 1999. "A Universal Lattice Code Decoder for Fading Channels". *IEEE Trans. Inform. Theory*. 45(July): 1639–1642.

[98] Elad, M. 2010. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. 1st ed. Springer-Verlag New York.

[99] Elad, M. and I. Yavneh. 2009. "A Plurality of Sparse Representations Is Better Than the Sparsest One Alone". *IEEE Trans. Inform. Theory*. 55(10): 4701–4714.

[100] Eldar, Y. C. and G. Kutyniok. 2012. *Compressed Sensing: Theory and Applications*. Cambridge University Press.

[101] Eldar, Y. C. and M. Mishali. 2009. "Robust Recovery of Signals From a Structured Union of Subspaces". *IEEE Trans. Inform. Theory*. 55(11): 5302–5316.

[102] Eriksson, A. and A. van den Hengel. 2010. "Efficient Computation of Robust Low-Rank Matrix Approximations in the Presence of Missing Data Using the L_1 Norm". In: *Proc. of the 23rd IEEE Conf. on Computer Vision and Pattern Recognition*. San Francisco, CA. 771–778.

[103] Fan, J. and R. Li. 2001. "Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties". *Journal of the American Stat. Assoc.* 96(456): 1348–1360.

[104] Fazel, F., M. Fazel, and M. Stojanovic. 2011. "Random Access Compressed Sensing for Energy-Efficient Underwater Sensor Networks". *IEEE J. Select. Areas Commun.* 29(8): 1660–1670.

[105] FCC. 2002. "Spectrum Policty Task Force". *Tech. rep.* No. ET Docket 02-135.

[106] Finamore, W. and W. Pearlman. 1980. "Optimal Encoding of Discrete-Time Continuous-Amplitude Memoryless Sources with Finite Output Alphabets". *IEEE Trans. Inform. Theory*. 26(2): 144–155.

[107] Fine, T. 1964. "Properties of An Optimum Digital System and Applications". *IEEE Trans. Inform. Theory*. 10(4): 287–296.

[108] Fleming, M., Q. Zhao, and M. Effros. 2004. "Network Vector Quantization". *IEEE Trans. Inform. Theory*. 50(8): 1584–1604.

[109] Fletcher, A. K., S. Rangan, and V. K. Goyal. 2009. “A Sparsity Detection Framework for On-Off Random Access Channels”. In: *Proc. IEEE Int. Symp. Inform. Theory*. Seoul, South Korea. 169–173.

[110] Flynn, T. and R. Gray. 1987. “Encoding of Correlated Observations”. *IEEE Trans. Inform. Theory*. 33(6): 773–787.

[111] Foster, D. P. and E. I. George. 1994. “The Risk Inflation Criterion for Multiple Regression”. *Annals of Statistics*. 22(4): 1947–1975.

[112] Foucart, S. and H. Rauhut. 2013. *A Mathematical Introduction to Compressive Sensing*. *Applied and Numerical Harmonic Analysis*. Springer New York.

[113] Fraysse, A., B. Pesquet-Popescu, and J.-C. Pesquet. 2009. “On the Uniform Quantization of a Class of Sparse Sources”. *IEEE Trans. Inform. Theory*. 55(7): 3243–3263.

[114] Freris, N. M., O. Öçal, and M. Vetterli. 2013. “Compressed Sensing of Streaming Data”. In: *Proc. Annual Allerton Conf. Commun., Contr., Computing*. Monticello, IL, USA. 1242–1249.

[115] Friedman, J., T. Hastie, H. Höfling, and R. Tibshirani. 2007. “Pathwise Coordinate Optimization”. *Annals of Applied Statistics*. 1(Dec.): 302–332.

[116] Fristedt, B. E. and L. F. Gray. 1997. *A Modern Approach to Probability Theory*. Boston, MA: Birkhäuser.

[117] Fuchs, J.-J. 1999. “Multipath Time-Delay Detection and Estimation”. *IEEE Trans. Signal Processing*. 47(1): 237–243.

[118] Fuchs, J.-J. 2001. “On the Application of the Global Matched Filter to DOA Estimation with Uniform Circular Arrays”. *IEEE Trans. Signal Processing*. 49(4): 702–709.

[119] Al-Fuqaha, A., M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. 2015. “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”. *IEEE Commun. Surveys & Tutorials*. 17(4): 2347–2376.

[120] Gallager, R. 2013. *Stochastic Processes: Theory for Applications*. New York, USA: Cambridge University Press. 536.

[121] Gamal, A. E. and Y.-H. Kim. 2012. *Network Information Theory*. New York, USA: Cambridge University Press.

[122] Gersho, A. 1982. “On The Structure of Vector Quantizers”. *IEEE Trans. Inform. Theory*. 28(2): 157–166.

[123] Giannakis, G. B., Z. Liu, X. Ma, and S. Zhou. 2007. *Space-Time Coding for Broadband Wireless Communications*. John Wiley & Sons, Inc.

[124] Gibson, J. 2013. *Information Theory and Rate Distortion Theory for Communications and Compression*. 1st ed. *Synthesis Lectures on Communications*. Morgan & Claypool.

[125] Gish, H. and J. Pierce. 1968. "Asymptotically Efficient Quantizing". *IEEE Trans. Inform. Theory*. 14(5): 676–683.

[126] Goblirsch, D. 1989. "Quantization Systems for Hidden Markov Sources". *PhD thesis*. College Park, MD, USA: The Institute for Systems Research, Univ. Maryland. URL: http://drum.lib.umd.edu/handle/1903/4942.

[127] Golub, G. and C. Loan. 1996. *Matrix Computations*. Baltimore: Johns Hopkins University Press.

[128] Gorodnitsky, I. F. and B. D. Rao. 1997. "Sparse Signal Reconstruction from Limited Data Using FOCUSS: A Re-Weighted Minimum Norm Algorithm". *IEEE Trans. Signal Processing*. 45(3): 600–616.

[129] Goyal, V., A. Fletcher, and S. Rangan. 2008. "Compressive Sampling and Lossy Compression". *IEEE Signal Processing Mag.* 25(2): 48–56.

[130] Grant, M. and S. Boyd. 2012. "CVX: Matlab Software for Disciplined Convex Programming, version 1.22". http://cvxr.com/cvx.

[131] Gray, R. 1972. "Conditional Rate-Distortion Theory". *Tech. rep.* Stanford Electron. Lab, Stanford University. 17.

[132] Gray, R. 1973. "A New Class of Lower Bounds to Information Rates of Stationary Sources via Conditional Rate-Distortion Functions". *IEEE Trans. Inform. Theory*. 19(4): 480–489.

[133] Gray, R. 1984. "Vector Quantization". *IEEE Trans. Acoust., Speech, Signal Processing*. 1(2): 4–29.

[134] Gray, R. 1990. *Source Coding Theory*. Vol. 83. *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers.

[135] Gray, R. and D. Neuhoff. 1998. "Quantization". *IEEE Trans. Inform. Theory*. 44(6): 2325–2383.

[136] Gray, R., D. Neuhoff, and J. Omura. 1975. "Process Definitions of Distortion-Rate Functions and Source Coding Theorems". *IEEE Trans. Inform. Theory*. 21(5): 524–532.

[137] Graziosi, F. and F. Santucci. 2002. “A General Correlation Model for Shadow Fading in Mobile Radio Systems”. *IEEE Commun. Lett.* 6(3): 102–104.

[138] Gu, Y., Y. Chen, and A. O. Hero. 2009. “Sparse LMS for System Identification”. In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Taipei, Taiwan.

[139] Gudmundson, M. 1991. “Correlation Model for Shadow Fading in Mobile Radio Systems”. *Elec. Lett.* 27(23): 2145–2146.

[140] Guler, B., E. MolavianJazi, and A. Yener. 2015. “Remote Source Coding with Two-Sided Information”. In: *Proc. IEEE Int. Symp. Inform. Theory*. 2176–2180.

[141] Hamilton, B. R., X. Ma, R. J. Baxley, and S. M. Matechik. 2012. “Radio Frequency Tomography in Mobile Networks”. In: *Proc. of the IEEE Stat. Sig. Proc. Workshop*. Ann Arbor, MI. 508–511.

[142] Hamilton, B. R., X. Ma, R. J. Baxley, and S. M. Matechik. 2014. “Propagation Modeling for Radio Frequency Tomography in Wireless Networks”. *IEEE J. Select. Topics Signal Processing*. 8(1): 55–65.

[143] Hansen, P. C. 1994. “Regularization Tools: A Matlab Package for Analysis and Solution of Discrete Ill-Posed Problems”. *Numerical Algorithms*. 6(1): 1–35.

[144] Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, And Prediction*. Second. New York: Springer.

[145] Haupt, J., W. Bajwa, M. Rabbat, and R. Nowak. 2008. “Compressed Sensing for Networked Data: A Different Approach to Decentralized Compression”. *IEEE Signal Processing Mag.* 25(2): 92–101.

[146] Haupt, J. and R. Nowak. 2006. “Signal Reconstruction From Noisy Random Projections”. *IEEE Trans. Inform. Theory*. 52(9): 4036–4048.

[147] Haupt, J. and R. Nowak. 2012. “Adaptive Sensing for Sparse Recovery”. In: *Compressive Sensing: Theory and Applications*. Ed. by Y. C. Eldar and G. Kutyniok. Cambridge University Press. Chap. 6. 269–304.

[148] Herman, M. A. and T. Strohmer. 2010. “General Deviants: an Analysis of Perturbations in Compressive Sensing”. *IEEE J. Select. Topics Signal Processing*. 4(Apr.): 342–349.

[149] Huffman, D. 1952. “A Method for the Construction of Minimum-Redundancy Codes”. *Proc. IRE*. 40(9): 1098–1101.

[150] I. G. Akrotirianakis and C. A. Floudas. 2004. "Computational Experience with a New Class of Convex Underestimators: Box-Constrained NLP Problems". *Journal of Global Optimization*. 29(3): 249–264.

[151] Jacques, L., D. Hammond, and J. Fadili. 2011. "Dequantizing Compressed Sensing: When Oversampling and Non-Gaussian Constraints Combine". *IEEE Trans. Inform. Theory*. 57(1): 559–571.

[152] Jacques, L., J. N. Laska, P. T. Boufounos, and R. G. Baraniuk. 2013. "Robust 1-Bit Compressive Sensing via Binary Stable Embeddings of Sparse Vectors". *IEEE Trans. Inform. Theory*. 59(4): 2082–2102.

[153] Jansson, M., A. L. Swindlehurst, and B. Ottersten. 1998. "Weighted Subspace Fitting for General Array Error Models". *IEEE Trans. Signal Processing*. 46(9): 2484–2498.

[154] Ji, S., Y. Xue, and L. Carin. 2008. "Bayesian Compressive Sensing". *IEEE Trans. Signal Processing*. 56(6): 2346–2356.

[155] Kaltiokallio, O., M. Bocca, and N. Patwari. 2012. "Enhancing the Accuracy of Radio Tomographic Imaging Using Channel Diversity". In: *Proc. of the 9th IEEE Int. Conf. on Mobile Ad hoc and Sensor Syst.* Las Vegas, NV. 254–262.

[156] Kamilov, U. S., S. Rangan, A. K. Fletcher, and M. Unser. 2014. "Approximate Message Passing With Consistent Parameter Estimation and Applications to Sparse Learning". *IEEE Trans. Inform. Theory*. 60(5): 2969–2985.

[157] Kamilov, U., V. Goyal, and S. Rangan. 2012. "Message-Passing De-Quantization With Applications to Compressed Sensing". *IEEE Trans. Signal Processing*. 60(12): 6270–6281.

[158] Kanso, M. A. and M. G. Rabbat. 2009. "Compressed RF Tomography for Wireless Sensor Networks: Centralized And Decentralized Approaches". In: *Int. Conf. Distributed Comput. Sensor Syst.* Springer. 173–186.

[159] Khajehnejad, M., W. Xu, A. Avestimehr, and B. Hassibi. 2011. "Analyzing Weighted ℓ_1 Minimization for Sparse Recovery with Nonuniform Sparse Models". *IEEE Trans. Signal Processing*. 59(5): 1985–2001.

[160] Kim, S.-J., E. Dall'Anese, J. A. Bazerque, K. Rajawat, and G. B. Giannakis. 2014. "Advances in Spectrum Sensing and Cross-Layer Design for Cognitive Radio Networks". In: *Academic Press Library in Signal Processing: Volume 2 Communications and Radar Signal Processing R. Chellapa and S. Theodoridis, Eds.* Academic Press. Chap. 9. 471–502.

[161] Kim, S.-J., E. Dall'Anese, and G. B. Giannakis. 2011. "Cooperative Spectrum Sensing for Cognitive Radios Using Kriged Kalman Filtering". *IEEE J. Select. Topics Signal Processing*. 5(1): 24–36.

[162] Kipnis, A., Y. C. Eldar, and A. J. Goldsmith. 2018a. "Analog-to-Digital Compression: A New Paradigm for Converting Signals to Bits". *IEEE Signal Processing Mag.* 35(3): 16–39.

[163] Kipnis, A., A. J. Goldsmith, T. Weissman, and Y. C. Eldar. 2016. "Distortion Rate Function of Sub-Nyquist Sampled Gaussian Sources". *IEEE Trans. Inform. Theory*. 62(1): 401–429.

[164] Kipnis, A., G. Reeves, and Y. C. Eldar. 2018b. "Single Letter Formulas for Quantized Compressed Sensing with Gaussian Codebooks". In: *Proc. IEEE Int. Symp. Inform. Theory*. Vail, CO, USA. 71–75.

[165] Kipnis, A., G. Reeves, Y. Eldar, and A. Goldsmith. 2017. "Compressed Sensing Under Optimal Quantization". In: *Proc. IEEE Int. Symp. Inform. Theory*. Aachen, Germany. 2148–2152.

[166] Kipnis, A., S. Rini, and A. J. Goldsmith. 2015. "The Indirect Rate-Distortion Function of a Binary i.i.d Source". In: *Proc. IEEE Inform. Theory Workshop*. Jeju Island, Korea. 352–356.

[167] Knight, K. and W. J. Fu. 2000. "Asymptotics for Lasso-Type Estimators". *Annals of Statistics*. 28: 1356–1378.

[168] Ku, G., J. Ren, and J. M. Walsh. 2015. "Computing the Rate Distortion Region for the CEO Problem With Independent Sources". *IEEE Trans. Signal Processing*. 63(3): 567–575.

[169] Kushner, H. and G. G. Yin. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*. New York, NY: Springer-Verlag.

[170] Lalos, A. S., A. Antonopoulos, E. Kartsakli, M. D. Renzo, S. Tennina, L. Alonso, and C. Verikoukis. 2015. "RLNC-Aided Cooperative Compressed Sensing for Energy Efficient Vital Signal Telemonitoring". *IEEE Trans. Wireless Commun.* 14(7): 3685–3699.

[171] Lee, D., S. Kim, and G. B. Giannakis. 2017. "Channel Gain Cartography for Cognitive Radios Leveraging Low Rank and Sparsity". *IEEE Trans. Wireless Commun.* 16(9): 5953–5966.

[172] Lee, S., S. Pattem, M. Sathiamoorthy, B. Krishnamachari, and A. Ortega. 2009a. "Compressed Sensing and Routing in Sensor Networks". *Tech. rep.* USC.

[173] Lee, S., S. Pattem, M. Sathiamoorthy, B. Krishnamachari, and A. Ortega. 2009b. "Spatially-Localized Compressed Sensing and Routing in Multi-hop Sensor Networks". In: *Proceedings of the 3rd International Conference on GeoSensor Networks*. Oxford, UK. 11–20.

[174] Leiner, B. and R. Gray. 1974. "Rate-Distortion Theory for Ergodic Sources with Side Information (Corresp.)" *IEEE Trans. Inform. Theory*. 20(5): 672–675.

[175] Leinonen, M., M. Codreanu, and M. Juntti. 2013. "Distributed Correlated Data Gathering in Wireless Sensor Networks via Compressed Sensing". In: *Proc. Annual Asilomar Conf. Signals, Syst., Comp.* Pacific Grove, CA. 418–422.

[176] Leinonen, M., M. Codreanu, and M. Juntti. 2014. "Compressed Acquisition and Progressive Reconstruction of Multi-Dimensional Correlated Data in Wireless Sensor Networks". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Florence, Italy. 6449–6453.

[177] Leinonen, M., M. Codreanu, and M. Juntti. 2015a. "Channel-Robust Compressed Sensing via Vector Pre-Quantization in Wireless Sensor Networks". In: *Proc. IEEE Global Conf. on Signal and Inform. Proc.* Orlando, Florida, USA. 383–387.

[178] Leinonen, M., M. Codreanu, and M. Juntti. 2015b. "Sequential Compressed Sensing With Progressive Signal Reconstruction in Wireless Sensor Networks". *IEEE Trans. Wireless Commun.* 14(3): 1622–1635.

[179] Leinonen, M., M. Codreanu, and M. Juntti. 2016a. "Distributed Variable-Rate Quantized Compressed Sensing in Wireless Sensor Networks". In: *Proc. IEEE Works. on Sign. Proc. Adv. in Wirel. Comms.,* invited paper. Edinburgh, UK. 1–5.

[180] Leinonen, M., M. Codreanu, and M. Juntti. 2018a. "Distributed Distortion-Rate Optimized Compressed Sensing in Wireless Sensor Networks". *IEEE Trans. Commun.* 66(4): 1609–1623.

[181] Leinonen, M., M. Codreanu, and M. Juntti. 2019a. “Practical Compression Methods for Quantized Compressed Sensing”. In: *Proc. IEEE Int. Conf. on Comp. Commun.* Paris, France. 5.

[182] Leinonen, M., M. Codreanu, and M. Juntti. 2019b. “Signal Reconstruction Performance under Quantized Noisy Compressed Sensing”. In: *Proc. IEEE Data Compr. Conf.* Snowbird, Utah. 586.

[183] Leinonen, M., M. Codreanu, M. Juntti, and G. Kramer. 2016b. “Rate-Distortion Lower Bound for Compressed Sensing via Conditional Remote Source Coding”. In: *Proc. IEEE Inform. Theory Workshop*. Cambridge, UK. 275–279.

[184] Leinonen, M., M. Codreanu, M. Juntti, and G. Kramer. 2018b. “Rate-Distortion Performance of Lossy Compressed Sensing of Sparse Sources”. *IEEE Trans. Commun.* 66(10): 4498–4512.

[185] Levin, D. A., Y. Peres, and W. L. Wilmer. 2008. *Markov Chains and Mixing Times*. 1st ed. Providence, Rhode Island: American Mathematical Society.

[186] Li, S., L. Xu, and X. Wang. 2013. “Compressed Sensing Signal and Data Acquisition in Wireless Sensor Networks and Internet of Things”. *IEEE Trans. Ind. Informat.* 9(4): 2177–2186.

[187] Lin, Z., A. Ganesh, J. Right, L. Wu, M. Chen, and Y. Ma. 2009. “Fast Convex Optimization Algorithms for Exact Recovery of a Corrupted Low-Rank Matrix”. *UIUC Technical Report UILU-ENG-09-2214*: 1–18.

[188] Linde, Y., A. Buzo, and R. Gray. 1980. “An Algorithm for Vector Quantizer Design”. *IEEE Trans. Commun.* 28(1): 84–95.

[189] Linder, T., R. Zamir, and K. Zeger. 2000. “On Source Coding with Side-Information-Dependent Distortion Measures”. *IEEE Trans. Inform. Theory*. 46(7): 2697–2704.

[190] Ling, Q. and Z. Tian. 2010. “Decentralized Sparse Signal Recovery for Compressive Sleeping Wireless Sensor Networks”. *IEEE Trans. Signal Processing*. 58(7): 3816–3827.

[191] Liu, Z., S. Cheng, A. D. Liveris, and Z. Xiong. 2006. “Slepian-Wolf Coded Nested Lattice Quantization for Wyner-Ziv Coding: High-Rate Performance Analysis and Code Design”. *IEEE Trans. Inform. Theory*. 52(10): 4358–4379.

[192] Lloyd, S. 1982. “Least Squares Quantization in PCM”. *IEEE Trans. Inform. Theory*. 28(2): 129–137.

[193] Löfberg, J. 2006. “Yalmip: Software for Solving Convex (and Nonconvex) Optimization Problems”. In: *Proc. Amer. Contr. Conf.* Minneapolis, MN.

[194] Lookabaugh, T. and R. Gray. 1989. “High-Resolution Quantization Theory and The Vector Quantizer Advantage”. *IEEE Trans. Inform. Theory*. 35(5): 1020–1033.

[195] Lu, W. and N. Vaswani. 2012. “Regularized Modified BPDN for Noisy Sparse Reconstruction With Partial Erroneous Support and Signal Value Knowledge”. *IEEE Trans. Signal Processing*. 60(1): 182–196.

[196] Luo, C., F. Wu, J. Sun, and C. Chen. 2009. “Compressive Data Gathering for Large-Scale Wireless Sensor Networks”. In: *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*. Beijing, China. 145–156.

[197] Luo, C., F. Wu, J. Sun, and C. Chen. 2010. “Efficient Measurement Generation and Pervasive Sparsity for Compressive Data Gathering”. *IEEE Trans. Wireless Commun.* 9(12): 3728–3738.

[198] M. Godavarti, A. O. H. I. 2005. “Partial Update LMS Algorithms”. *IEEE Trans. Signal Processing*. 53(7): 2382–2399.

[199] Mairal, J., F. Bach, J. Ponce, and G. Sapiro. 2010. “Online Learning for Matrix Factorization and Sparse Coding”. *J. Machine Learning Res.* 11(Jan.): 19–60.

[200] Malioutov, D. M., S. R. Sanghavi, and A. S. Willsky. 2010. “Sequential Compressed Sensing”. *IEEE J. Select. Topics Signal Processing*. 4(2): 435–444.

[201] Malioutov, D., M. Çetin, and A. S. Willsky. 2005. “A Sparse Signal Reconstruction Perspective for Source Localization with Sensor Arrays”. *IEEE Trans. Signal Processing*. 53(8): 3010–3022.

[202] Mallat, S. 2008. *A Wavelet Tour of Signal Processing*. 3rd. Academic Press.

[203] Mallat, S. G. and Z. Zhang. 1993. “Matching Pursuits with Time-Frequency Dictionaries”. *IEEE Trans. Signal Processing*. 41(12): 3397–3415.

[204] Mamaghanian, H., N. Khaled, D. Atienza, and P. Vandergheynst. 2011. “Compressed Sensing for Real-Time Energy-Efficient ECG Compression on Wireless Body Sensor Nodes”. *IEEE Trans. Biomed. Eng.* 58(9): 2456–2466.

[205] Mardani, M., G. Mateos, and G. B. Giannakis. 2013a. “Dynamic Anomalography: Tracking Network Anomalies Via Sparsity and Low Rank”. *IEEE J. Select. Topics Signal Processing*. 7(1): 50–66.

[206] Mardani, M., G. Mateos, and G. B. Giannakis. 2013b. “Dynamic Anomalography: Tracking Network Anomalies via Sparsity and Low Rank”. *IEEE J. Select. Topics Signal Processing*. 7(1): 50–66.

[207] Mardani, M., G. Mateos, and G. B. Giannakis. 2013c. “Recovery of Low-Rank Plus Compressed Sparse Matrices with Application to Unveiling Traffic Anomalies”. *IEEE Trans. Inform. Theory*. 59(8): 5186–5205.

[208] Mardani, M., G. Mateos, and G. B. Giannakis. 2015. “Subspace Learning and Imputation for Streaming Big Data Matrices and Tensors”. *IEEE Trans. Signal Processing*. 63(10): 2663–2677.

[209] Markovsky, I. and S. Van Huffel. 2007. “Overview of Total Least-Squares Methods”. *Signal Processing*. 87(10): 2283–2302.

[210] Masiero, R., G. Quer, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi. 2009. “Data Acquisition Through Joint Compressive Sensing and Principal Component Analysis”. In: *Proc. IEEE Global Telecommun. Conf.* Honolulu, HI. 1–6.

[211] Max, J. 1960. “Quantizing for Minimum Distortion”. *Proc. IRE*. 6(1): 7–12.

[212] Mehanna, O. and N. D. Sidiropoulos. 2013. “Frugal Sensing: Wideband Power Spectrum Sensing From Few Bits”. *IEEE Trans. Signal Processing*. 61(10): 2693–2703.

[213] Mehmood, Y., F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani. 2017. “Internet-of-Things-Based Smart Cities: Recent Advances and Challenges”. *IEEE Commun. Mag.* 55(9): 16–24.

[214] Miosso, C., R. von Borries, M. Argàez, L. Velazquez, C. Quintero, and C. Potes. 2009. “Compressive Sensing Reconstruction with Prior Information by Iteratively Reweighted Least-Squares”. *IEEE Trans. Signal Processing*. 57(6): 2424–2431.

[215] Mishali, M., Y. C. Eldar, O. Dounaevsky, and E. Shoshan. 2011a. “Xampling: Analog to Digital at Sub-Nyquist Rates”. *IET Circ. Dev. Syst.* 5(1): 8–20.

[216] Mishali, M., Y. C. Eldar, and A. J. Elron. 2011b. “Xampling: Signal Acquisition and Processing in Union of Subspaces”. *IEEE Trans. Signal Processing*. 59(10): 4719–4734.

[217] Mishali, M. and Y. Eldar. 2011. “Sub-Nyquist Sampling”. *IEEE Signal Processing Mag.* 28(6): 98–124.

[218] Mishali, M. and Y. Eldar. 2012. “Xampling: Compressed Sensing of Analog Signals”. In: *Compressive Sensing: Theory and Applications*. Ed. by Y. C. Eldar and G. Kutyniok. Cambridge University Press. Chap. 3. 88–147.

[219] Moshtaghpour, A., M. A. Akhaee, and M. Attarifar. 2016. “Obstacle Mapping in Wireless Sensor Networks via Minimum Number Of Measurements”. *IET Sig. Proc.* 10(3): 237–246.

[220] Mostofi, Y. 2011. “Compressive Cooperative Sensing and Mapping in Mobile Networks”. 10(12): 1769–1784.

[221] Mota, J. F. C., N. Deligiannis, and M. R. D. Rodrigues. 2017. “Compressed Sensing With Prior Information: Strategies, Geometry, and Bounds”. *IEEE Trans. Inform. Theory*. 63(7): 4472–4496.

[222] Msechu, E. J. and G. B. Giannakis. 2012. “Sensor-Centric Data Reduction for Estimation With WSNs via Censoring and Quantization”. *IEEE Trans. Signal Processing*. 60(1): 400–414.

[223] Needell, D. 2009. “Noisy Signal Recovery via Iterative Reweighted L1-Minimization”. In: *Proc. Annual Asilomar Conf. Signals, Syst., Comp.* Pacific Grove, CA. 113–117.

[224] Needell, D. and J. A. Tropp. 2009. “CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples”. *Applied and Computational Harmonic Analysis*. 26(3): 301–321.

[225] Nestares, O., D. J. Fleet, and D. J. Heeger. 2000. “Likelihood Functions and Confidence Bounds for Total-Least-Squares Problems”. In: *Proc. of Conf. on Computer Vision and Pattern Recognition*. Hilton Head Island, SC.

[226] Nesterov, Y. 2004. *Introductory Lectures on Convex optimization : A Basic Course*. Boston, MA: Kluwer Academic.

[227] Nyquist, H. 1928. “Certain Topics in Telegraph Transmission Theory”. *Trans. Amer. Inst. Elec. Eng.* 47(2): 617–644.

[228] Önhon, N. Ö. and M. Çetin. 2009. “A Nonquadratic Regularization-Based Technique for Joint SAR Imaging and Model Error Correction”. In: *Proc. of SPIE*. Vol. 7337.

[229] Oppenheim, A. and G. Verghese. 2015. *Signals, Systems and Inference*. NJ, USA: Prentice Hall. 608.

[230] Palzer, L. and R. Timo. 2016. "A Lower Bound for the Rate-Distortion Function of Spike Sources that is Asymptotically Tight". In: *Proc. IEEE Inform. Theory Workshop*. Cambridge, UK. 1–5.

[231] Pati, Y. C., R. Rezaiifar, and P. S. Krishnaprasad. 1993. "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition". In: *Proc. Annual Asilomar Conf. Signals, Syst., Comp.* Pacific Grove, CA. 40–44.

[232] Picard, R. R. and R. D. Cook. 1984. "Cross-Validation of Regression Models". *Journal of the American Stat. Assoc.* 79(387): 575–583.

[233] Pradhan, S. S., J. Kusuma, and K. Ramchandran. 2002. "Distributed Compression in a Dense Microsensor Network". *IEEE Signal Processing Mag.* 19(2): 51–60.

[234] Pradhan, S. S. and K. Ramchandran. 2003. "Distributed Source Coding Using Syndromes (DISCUS): Design and Construction". *IEEE Trans. Inform. Theory*. 49(3): 626–643.

[235] Protter, M., I. Yavneh, and M. Elad. 2010. "Closed-Form MMSE Estimation for Signal Denoising Under Sparse Representation Modeling Over a Unitary Dictionary". *IEEE Trans. Signal Processing*. 58(7): 3471–3484.

[236] Qin, Z., D. Goldfarb, and S. Ma. 2015. "An Alternating Direction Method For Total Variation Denoising". *Optim. Method Softw.* 30(3): 594–615.

[237] Quer, G., R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi. 2009. "On the Interplay Between Routing and Signal Representation for Compressive Sensing in Wireless Sensor Networks". In: *Information Theory and Applications Workshop*. San Diego, CA. 206–215.

[238] Quer, G., R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi. 2012. "Sensing, Compression, and Recovery for WSNs: Sparse Signal Modeling and Monitoring Framework". *IEEE Trans. Wireless Commun.* 11(10): 3447–3461.

[239] Raghunathan, V., C. Schurgers, S. Park, and M. B. Srivastava. 2002. "Energy-Aware Wireless Microsensor Networks". *IEEE Signal Processing Mag.* 19(2): 40–50.

[240] Rani, M., S. B. Dhok, and R. B. Deshmukh. 2018. "A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications". *IEEE Acc.* 6: 4875–4894.

[241] Rasmussen, C. and C. Williams. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

[242] Razaviyayn, M., M. Sanjabi, and Z. Luo. 2016. “A Stochastic Successive Minimization Method for Nonsmooth Nonconvex Optimization with Applications to Transceiver Design in Wireless Communication Networks”. *Math. Program.* 157(2): 515–545.

[243] Rebollo-Monedero, D. 2007. “Quantization and Transforms for Distributed Source Coding”. *PhD thesis*. Dept. Elect. Eng., Stanford University.

[244] Rebollo-Monedero, D., R. Zhang, and B. Girod. 2003. “Design of Optimal Quantizers for Distributed Source Coding”. In: *Proc. IEEE Data Compr. Conf.* Snowbird, UT, USA. 13–22.

[245] Recht, B., M. Fazel, and P. A. Parrilo. 2010. “Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization”. *SIAM Rev.* 52(3): 471–501.

[246] Reeves, G. and M. Gastpar. 2012. “The Sampling Rate-Distortion Tradeoff for Sparsity Pattern Recovery in Compressed Sensing”. *IEEE Trans. Inform. Theory*. 58(5): 3065–3092.

[247] Rissanen, J. and G. Langdon. 1979. “Arithmetic Coding”. *IBM Journal of Research and Development*. 23(2): 149–162.

[248] Rivenson, Y. and A. Stern. 2009. “Compressed Imaging With a Separable Sensing Operator”. *IEEE Signal Processing Lett.* 16(6): 449–452.

[249] Rose, K. 1994. “A Mapping Approach to Rate-Distortion Computation and Analysis”. *IEEE Trans. Inform. Theory*. 40(6): 1939–1952.

[250] Rose, K., E. Gurewitz, and G. C. Fox. 1992. “Vector Quantization by Deterministic Annealing”. *IEEE Trans. Inform. Theory*. 38(4): 1249–1257.

[251] Rose, K. and D. Miller. 1993. “A Deterministic Annealing Algorithm for Entropy-Constrained Vector Quantizer Design”. In: *Proc. Annual Asilomar Conf. Signals, Syst., Comp.* Pacific Grove, CA. 1651–1655.

[252] Rudin, L. I., S. Osher, and E. Fatemi. 1992. “Nonlinear Total Variation Based Noise Removal Algorithms”. *Phys. D: Nonlin. Phenom.* 60(1): 259–268.

[253] Sakrison, D. 1968. “Source Encoding in the Presence of Random Disturbance (corresp.)” *IEEE Trans. Inform. Theory*. 14(1): 165–167.

[254] Sayed, A. H. 2003. *Fundamentals of Adaptive Filtering*. John Wiley & Sons.

[255] Scharf, L. L. and B. Friedlander. 1994. “Matched Subspace Detectors”. *IEEE Trans. Signal Processing*. 42(8): 2146–2157.

[256] Schizas, I. D., G. B. Giannakis, and N. Jindal. 2008. “Distortion-Rate Bounds for Distributed Estimation Using Wireless Sensor Networks”. *EURASIP J. Advances Signal Processing*. 1: 12. article ID: 748605.

[257] Schniter, P., L. C. Potter, and J. Ziniel. 2008. “Fast Bayesian Matching Pursuit”. In: *Proc. Inform. Theory and Appl. Workshop*. San Diego, CA, USA. 326–333.

[258] Schwarz, G. 1978. “Estimating the Dimension of a Model”. *Annals of Statistics*. 6(2): 461–464.

[259] Shannon, C. E. 1949. “Communication in the Presence of Noise”. *Proc. IRE*. 37(1): 10–21.

[260] Shannon, C. 1959. “Coding Theorems for a Discrete Source With a Fidelity Criterion”. *IRE Nat. Conv. Rec.* 7: 142–163.

[261] Sharma, S. K., E. Lagunas, S. Chatzinotas, and B. Ottersten. 2016. “Application of Compressive Sensing in Cognitive Radio Communications: A Survey”. *IEEE Commun. Surveys & Tutorials*. 18(3): 1838–1860.

[262] Shirazinia, A. 2014. “Source and Channel Coding for Compressed Sensing and Control”. *PhD thesis*. KTH Royal Institute of Technology. URL: http://kth.diva-portal.org/smash/get/diva2:709596/FULLTEXT01.pdf.

[263] Shirazinia, A., S. Chatterjee, and M. Skoglund. 2013. “Analysis-by-Synthesis Quantization for Compressed Sensing Measurements”. *IEEE Trans. Signal Processing*. 61(22): 5789–5800.

[264] Shirazinia, A., S. Chatterjee, and M. Skoglund. 2014a. “Distributed Quantization for Compressed Sensing”. In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Florence, Italy. 6439–6443.

[265] Shirazinia, A., S. Chatterjee, and M. Skoglund. 2014b. “Joint Source-Channel Vector Quantization for Compressed Sensing”. *IEEE Trans. Signal Processing*. 62(14): 3667–3681.

[266] Shirazinia, A., S. Chatterjee, and M. Skoglund. 2015. “Distributed Quantization for Measurement of Correlated Sparse Sources over Noisy Channels”. available at http://arxiv.org/abs/1404.7640v2.

[267] Slavakis, K., S.-J. Kim, G. Mateos, and G. B. Giannakis. 2014. "Stochastic Approximation vis-a-vis Online Learning for Big Data Analytics". *IEEE Signal Processing Mag.* 31(Nov.).

[268] Slepian, D. and J. Wolf. 1973. "Noiseless Coding of Correlated Information Sources". *IEEE Trans. Inform. Theory*. 19(4): 471–480.

[269] Solo, V. and X. Kong. 1995. *Adaptive Signal Processing Algorithms: Stability and Performance*. Upper Saddle River, NJ: Prentice-Hall.

[270] Stoica, P. and P. Babu. 2012. "SPICE and LIKES: Two Hyperparameter-Free Methods for Sparse-Parameter Estimation". *Signal Processing, Elsevier*. 92(7): 1580–1590.

[271] Sturm, J. F. 1999. "Using SeDuMi 1.02, a MATLAB Toolbox for Optimization over Symmetric Cones". *Optimization Methods Software*. 11–12(Aug.): 625–653.

[272] Subramaniam, A. D. and B. D. Rao. 2003. "PDF Optimized Parametric Vector Quantization of Speech Line Spectral Frequencies". *IEEE Trans. Speech Audio Processing*. 11(2): 130–142.

[273] Sun, J. and V. Goyal. 2009. "Optimal Quantization of Random Measurements in Compressed Sensing". In: *Proc. IEEE Int. Symp. Inform. Theory*. Seoul, Korea. 6–10.

[274] Tang, G. and A. Nehorai. 2010. "Performance Analysis for Sparse Support Recovery". *IEEE Trans. Inform. Theory*. 56(3): 1383–1399.

[275] Taubock, G. and F. Hlawatsch. 2008. "A Compressed Sensing Technique for OFDM Channel Estimation in Mobile Environments: Exploiting Channel Sparsity for Reducing Pilots". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Las Vegas, NV.

[276] Tian, C. and J. Chen. 2009. "Remote Vector Gaussian Source Coding With Decoder Side Information Under Mutual Information and Distortion Constraints". *IEEE Trans. Inform. Theory*. 55(10): 4676–4680.

[277] Tian, Z., G. Leus, and V. Lottici. 2009. "Detection of Sparse Signals under Finite-Alphabet Constraints". In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Taipei, Taiwan. 2349–2352.

[278] Tibshirani, R. 1996. "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society*. 58(1): 267–288.

[279] Tipping, M. E. 2001. "Sparse Bayesian Learning and the Relevance Vector Machine". *Journal of Machine Learning Research*. 1(Sept.): 211–244.

[280] Tseng, P. 2001. “Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization”. *J. Opt. Theory Appl.* 109(3): 475–494.

[281] Turek, J. S., I. Yavneh, and M. Elad. 2011. “On MMSE and MAP Denoising Under Sparse Representation Modeling Over a Unitary Dictionary”. *IEEE Trans. Signal Processing*. 59(8): 3526–3535.

[282] Valenzuela, R. A. 1994. “Ray Tracing Prediction of Indoor Radio Propagation”. In: *Proc. IEEE 5th Int. Symp. on Personal Indoor and Mobile Radio Comm.* The Hague, Netherlands. 140–144.

[283] Van Huffel, S. and J. Vandewalle. 1991. *The Total Least Squares Problem: Computational Aspects and Analysis*. Vol. 9. *Frontier in Applied Mathematics*. Philadelphia: SIAM.

[284] Vaswani, N. 2008. “Kalman Filtered Compressed Sensing”. In: *Proc. IEEE Int. Conf. Image Proc.* San Diego, CA. 893–896.

[285] Vaswani, N. 2010. “LS-CS-Residual (LS-CS): Compressive Sensing on Least Squares Residual”. *IEEE Trans. Signal Processing*. 58(8): 4108–4120.

[286] Vaswani, N. and W. Lu. 2010. “Modified-CS: Modifying Compressive Sensing for Problems With Partially Known Support”. *IEEE Trans. Signal Processing*. 58(9): 4595–4607.

[287] Vaswani, N. and W. Lu. 2014. “Recursive Reconstruction of Sparse Signal Sequences”. In: *Compressed Sensing & Sparse Filtering*. Ed. by A. Y. Carmi, L. Mihaylova, and S. J. Godsill. *Signals and Communication Technology*. Springer Berlin Heidelberg. Chap. 11. 357–380.

[288] Verdu, S. 1998. *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press.

[289] Vetterli, M., P. Marziliano, and T. Blu. 2002. “Sampling Signals with Finite Rate of Innovation”. *IEEE Trans. Signal Processing*. 50(6): 1417–1428.

[290] Wainwright, M. J. 2009. “Sharp Thresholds for High-Dimensional and Noisy Sparsity Recovery Using ℓ_1-Constrained Quadratic Programming (Lasso)”. *IEEE Trans. Inform. Theory*. 55(5): 2183–2202.

[291] Wang, J., S. Tang, B. Yin, and X.-Y. Li. 2012. “Data Gathering in Wireless Sensor Networks through Intelligent Compressive Sensing”. In: *Proc. IEEE Int. Conf. on Comp. Commun.* 603–611.

[292] Wang, W., M. Garofalakis, and K. Ramchandran. 2007. "Distributed Sparse Random Projections for Refinable Approximation". In: *Proc. IEEE Int. Symp. on Inform. Proc. in Sensor Networks*. 331–339.

[293] Weidmann, C. and M. Vetterli. 2012. "Rate Distortion Behavior of Sparse Sources". *IEEE Trans. Inform. Theory*. 58(8): 4969–4992.

[294] Wernersson, N., J. Karlsson, and M. Skoglund. 2009. "Distributed Quantization over Noisy Channels". *IEEE Trans. Commun.* 57(6): 1693–1700.

[295] White, G. P., Y. V. Zakharov, and J. Liu. 2008. "Low-Complexity RLS Algorithms using Dichotomous Coordinate Descent Iterations". *IEEE Trans. Signal Processing*. 56(July): 3150–3161.

[296] Wijewardhana, U. L. and M. Codreanu. 2017. "A Bayesian Approach for Online Recovery of Streaming Signals From Compressive Measurements". *IEEE Trans. Signal Processing*. 65(1): 184–199.

[297] Wilson, J. and N. Patwari. 2010. "Radio Tomographic Imaging with Wireless Networks". 9(5): 621–632.

[298] Wilson, J. and N. Patwari. 2011. "See-Through Walls: Motion Tracking Using Variance-Based Radio Tomography Networks". 10(5): 612–621.

[299] Wimalajeewa, T., Y. C. Eldar, and P. K. Varshney. 2015. "Subspace Recovery From Structured Union of Subspaces". *IEEE Trans. Inform. Theory*. 61(4): 2101–2114.

[300] Wimalajeewa, T. and P. K. Varshney. 2014. "OMP Based Joint Sparsity Pattern Recovery Under Communication Constraints". *IEEE Trans. Signal Processing*. 62(19): 5059–5072.

[301] Wipf, D. P. and B. D. Rao. 2004. "Sparse Bayesian Learning for Basis Selection". *IEEE Trans. Signal Processing*. 52(8): 2153–2164.

[302] Wipf, D. and S. Nagarajan. 2010. "Iterative Reweighted l_1 and l_2 Methods for Finding Sparse Solutions". *IEEE J. Select. Topics Signal Processing*. 4(2): 317–329.

[303] Witsenhausen, H. 1980. "Indirect Rate Distortion Problems". *IEEE Trans. Inform. Theory*. 26(5): 518–521.

[304] Wolf, J. and J. Ziv. 1970. "Transmission of Noisy Information to a Noisy Receiver with Minimum Distortion". *IEEE Trans. Inform. Theory*. 16(4): 406–411.

[305] Wölfle, G., R. Wahl, P. Wertz, P. Wildbolz, and F. Landstorfer. 2005. "Dominant Path Prediction Model for Indoor Scenarios". In: *Proc. German Microwave Conf.* 176–179.

[306] Wood, R. 1969. "On Optimum Quantization". *IEEE Trans. Inform. Theory*. 15(2): 248–252.

[307] Wright, J., A. Ganesh, K. Min, and Y. Ma. 2013. "Compressive Principal Component Pursuit". *Information and Inference*. 2(1): 32–68.

[308] Wu, T. T. and K. Lange. 2008. "Coordinate Descent Algorithms for Lasso Penalized Regression". *Annals of Applied Statistics*. 2(Mar.): 224–244.

[309] Xiang, L., J. Luo, and C. Rosenberg. 2013. "Compressed Data Aggregation: Energy-Efficient and High-Fidelity Data Collection". *IEEE/ACM Trans. Networking*. 21(6): 1722–1735.

[310] Xiao, J.-J., A. Ribeiro, Z.-Q. Luo, and G. Giannakis. 2006. "Distributed Compression-Estimation Using Wireless Sensor Networks". *IEEE Signal Processing Mag.* 23(4): 27–41.

[311] Xiong, Z., A. Liveris, and S. Cheng. 2004. "Distributed Source Coding for Sensor Networks". *IEEE Signal Processing Mag.* 21(5): 80–94.

[312] Xu, W. and B. Hassibi. 2012. "Fundamental Thresholds in Compressed Sensing: A High-Dimensional Geometry Approach". In: *Compressive Sensing: Theory and Applications*. Ed. by Y. C. Eldar and G. Kutyniok. Cambridge University Press. Chap. 7. 305–347.

[313] Yang, G., V. Y. F. Tan, C. K. Ho, S. H. Ting, and Y. L. Guan. 2013. "Wireless Compressive Sensing for Energy Harvesting Sensor Nodes". *IEEE Trans. Signal Processing*. 61(18): 4491–4505.

[314] Yuan, M. and Y. Lin. 2006a. "Model Selection and Estimation in Regression with Grouped Variables". *J. Royal Stat. Soc – Series B*. 68(1): 49–67.

[315] Yuan, M. and Y. Lin. 2006b. "Model Selection and Estimation in Regression with Grouped Variables". *Journal Royal Stat. Society: Series B*. 68: 49–67.

[316] Zanella, A., N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. 2014. "Internet of Things for Smart Cities". *IEEE Internet of Things J.* 1(1): 22–32.

[317] Zeger, K. and A. Gersho. 1988. "Vector Quantizer Design for Memoryless Noisy Channels". In: *Proc. IEEE Int. Conf. Commun.* Philadelphia, PA, USA. 1593–1597.

[318] Zhan, J. and N. Vaswani. 2015. “Time Invariant Error Bounds for Modified-CS based Sparse Signal Sequence Recovery”. *IEEE Trans. Inform. Theory*. 61(3): 1389–1409.

[319] Zhao, Q. and M. Effros. 2003a. “Lossless and Near-Lossless Source Coding for Multiple Access Networks”. *IEEE Trans. Inform. Theory*. 49(1): 112–128.

[320] Zhao, Q. and M. Effros. 2003b. “Low Complexity Code Design for Lossless and Near-Lossless Side Information Source Codes”. In: *Proc. IEEE Data Compr. Conf.* 3–12.

[321] Zheng, L., A. Maleki, H. Weng, X. Wang, and T. Long. 2017. “Does ℓ_p-Minimization Outperform ℓ_1-Minimization?” *IEEE Trans. Inform. Theory*. 63(11): 6896–6935.

[322] Zhu, H. and G. B. Giannakis. 2011. “Exploiting Sparse User Activity in Multiuser Detection”. *IEEE Trans. Commun.* 59(2): 454–465.

[323] Zhu, H., G. B. Giannakis, and G. Leus. 2011a. “Weighted and Structured Sparse Total Least-Squares for Perturbed Compressive Sampling”. In: *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*. Prague, Czech Republic,

[324] Zhu, H., G. Leus, and G. B. Giannakis. 2011b. “Sparsity-Cognizant Total Least-Squares for Perturbed Compressive Sampling”. *IEEE Trans. Signal Processing*. 59(5): 2002–2016.

[325] Zhu, J., D. Baron, and F. Krzakala. 2017. “Performance Limits for Noisy Multimeasurement Vector Problems”. *IEEE Trans. Signal Processing*. 65(9): 2444–2454.

[326] Ziniel, J. and P. Schniter. 2010. “Tracking and Smoothing of Time-Varying Sparse Signals via Approximate Belief Propagation”. In: *Proc. Annual Asilomar Conf. Signals, Syst., Comp.* Pacific Grove, CA. 808–812.

[327] Ziniel, J. and P. Schniter. 2013. “Dynamic Compressive Sensing of Time-Varying Signals Via Approximate Message Passing”. *IEEE Trans. Signal Processing*. 61(21): 5270–5284.

[328] Ziv, J. 1985. “On Universal Quantization”. *IEEE Trans. Inform. Theory*. 31(3): 344–347.

[329] Ziv, J. and A. Lempel. 1978. “Compression of Individual Sequences via Variable-Rate Coding”. *IEEE Trans. Inform. Theory*. 24(5): 530–536.

[330] Zordan, D., G. Quer, M. Zorzi, and M. Rossi. 2011. "Modeling and Generation of Space-Time Correlated Signals for Sensor Network Fields". In: *Proc. IEEE Global Telecommun. Conf.* Houston, Texas, USA. 1–6.

[331] Zou, H. 2006. "The Adaptive Lasso and its Oracle Properties". *J. of the American Stat. Assoc.* Dec.: 1418–1429.

[332] Zou, H. and R. Li. 2008. "One-Step Sparse Estimates in Nonconcave Penalized Likelihood Models". *Annals of Statistics*. 36(4): 1509–1533.

[333] Zymnis, A., S. Boyd, and E. J. Candés. 2010. "Compressed Sensing With Quantized Measurements". *IEEE Signal Processing Lett.* 17(2): 149–152.