Markus Leinonen

DISTRIBUTED COMPRESSED DATA GATHERING IN WIRELESS SENSOR NETWORKS

UNIVERSITY OF OULU GRADUATE SCHOOL,
UNIVERSITY OF OULU,
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING,
CENTRE FOR WIRELESS COMMUNICATIONS,
INFOTECH OULU
MARKUS LEINONEN

DISTRIBUTED COMPRESSED DATA GATHERING IN WIRELESS SENSOR NETWORKS

Academic dissertation to be presented with the assent of the Doctoral Training Committee of Information Technology and Electrical Engineering of the University of Oulu for public defence in Lapinsali (L3), Linnanmäki, on 12 October 2018, at 12 noon

UNIVERSITY OF OULU, OULU 2018
Leinonen, Markus, Distributed compressed data gathering in wireless sensor networks.
University of Oulu Graduate School; University of Oulu, Faculty of Information Technology and Electrical Engineering; Centre for Wireless Communications; Infotech Oulu
Acta Univ. Oul. C 676, 2018
University of Oulu, P.O. Box 8000, FI-90014 University of Oulu, Finland

Abstract

Wireless sensor networks (WSNs) consisting of battery-powered sensors are increasingly deployed for a myriad of Internet of Things applications, e.g., environmental, industrial, and healthcare monitoring. Since wireless access is typically the main contributor to battery usage, minimizing communications is crucial to prolong network lifetime and improve user experience. The objective of this thesis is to develop and analyze energy-efficient distributed compressed data acquisition techniques for WSNs. The thesis proposes four approaches to conserve sensors' energy by minimizing the amount of information each sensor has to transmit to meet given application requirements.

The first part addresses a cross-layer design to minimize the sensors’ sum transmit power via joint optimization of resource allocation and multi-path routing. A distributed consensus optimization based algorithm is proposed to solve the problem. The algorithm is shown to have superior convergence compared to several baselines.

The remaining parts deal with compressed sensing (CS) of sparse/compressible sources. The second part focuses on the distributed CS acquisition of spatially and temporally correlated sensor data streams. A CS algorithm based on sliding window and recursive decoding is developed. The method is shown to achieve higher reconstruction accuracy with fewer transmissions and less decoding delay and complexity compared to several baselines, and to progressively refine past estimates.

The last two approaches incorporate the quantization of CS measurements and focus on lossy source coding. The third part addresses the distributed quantized CS (QCS) acquisition of correlated sparse sources. A distortion-rate optimized variable-rate QCS method is proposed. The method is shown to achieve higher distortion-rate performance than the baselines and to enable a trade-off between compression performance and encoding complexity via the pre-quantization of measurements.

The fourth part investigates information-theoretic rate-distortion (RD) performance limits of single-sensor QCS. A lower bound to the best achievable compression — defined by the remote RD function (RDF) — is derived. A method to numerically approximate the remote RDF is proposed. The results compare practical QCS methods to the derived limits, and show a novel QCS method to approach the remote RDF.

Keywords: compressed sensing, cross-layer design, data compression, distributed optimization, quantization, rate-distortion theory, remote source coding, signal sparsity
Leinonen, Markus, Hajuettu pakattu datankeruu langattomissa anturiverkoissa.
Oulun yliopiston tutkijakoulu; Oulun yliopisto, Tieto- ja sähköteknikan tiedekunta; Centre for Wireless Communications; Infotech Oulu
Acta Univ. Oul. C 676, 2018
Oulun yliopisto, PL 8000, 90014 Oulun yliopisto

Tiivistelmä


Väitöskirjan ensimmäinen osa tarkastee protokollakerrosten yhteissuunnittelua, jossa minimoitaa anturien yhteislähetysteho optimoimalla resurssilokaatio ja monitieteitys. Ratkaisuksi ehdotetaan konsensukseen perustuva hajautettu algoritmi. Tulokset osoittavat algoritmin suppenemisomaisuuksien olevan olevan verrokon paremman.


Asiakasraitt: epäsuora lähdekoodaus, hajuettu optimointi, koodisuhte-säröteoria, kvantisointi, pakattu havainta, protokollakerrosten yhteissuunnittelut, signaalain harvuu, tiedonpakkaus
To my family
Preface

The research work for this thesis was conducted at the Centre for Wireless Communications (CWC), at the University of Oulu, in Finland, during 2011–2018.

I would like to express my gratitude to my principal supervisor, Professor Markku Juntti, who gave me the opportunity to pursue a doctoral degree and has had confidence in me ever since during my doctoral training. His continual support and wide-ranging technical expertise helped me to expand my research area and encouraged me towards ambitious goals. I am also grateful to my supervisor, Adjunct Professor Marian Codreanu, for his advices and in-depth technical guidance, especially on distributed optimization and compressed sensing – the two overarching themes of my thesis. Special thanks go to his immensely constructive comments provided to improve the quality of the article drafts. I also wish to thank the pre-examiners, Professor Bhaskar Rao, from the University of California, San Diego, and Associate Professor Marco Duarte, from the University of Massachusetts Amherst, who provided valuable comments and suggestions to improve the thesis. I would also like to thank Professor Marcos Katz and Dr. Heikki Karvonen for acting as my follow-up group members.

I was privileged to cooperate with leading experts in the field of information theory led by Professor Dr. sc. techn. Gerhard Kramer at the Institute for Communications Engineering, Technische Universität München, in Germany, in 2013 and 2014. I am highly grateful for his brilliant ideas, which gave a strong impetus to deepen my studies on information-theoretic approaches. I would also like to thank my master’s thesis supervisor, our former colleague Dr. Juha Karjalainen, who supported hiring me for the CWC in 2010 and initiated me into a researcher’s life. He is acknowledged for his guidance during 2010–2011 as it substantially aided in publishing my first journal article.

My doctoral training was conducted over several projects. The first phase was carried out on the Networks of 2020 (NETS2020) project funded by the Finnish Funding Agency for Technology and Innovation (TEKES), Nokia, Nokia Siemens Networks, Ericsson Finland, Renesas Mobile Europe, Elektrobit, and EXFO Nethawk. Thereafter, I worked on the Sensing, Compression, Communications and Data Fusion in Wireless Sensor Networks (SeCoFu) project funded by the Academy of Finland. The last phase was carried out on the Optimized Compressed Sensing in Wireless Sensor Networks...
Networks (ComingNets) project funded by the Academy of Finland. I would like to express my profound appreciation for all of the funding parties, principal investigators, and project managers. I was also privileged to receive a two-year funded position on the Infotech Oulu Doctoral Program in 2012–2013. I would like to thank UniOGS and Infotech for funding my research visits and several conference trips. Finally, I feel deeply honored to have received several personal research grants from the following foundations (in alphabetical order): the Emil Aaltonen Foundation, HPY Research Foundation, Kalle and Elma Keränen Foundation, Nokia Foundation, Riitta and Jorma J. Takanen Foundation, Tauno Tönning Foundation and the Walter Ahlström Foundation. Furthermore, this thesis has been financially supported by Academy of Finland 6Genesis Flagship (grant 318927).

I wish to thank all my past and present co-workers at CWC for creating a productive research environment. I would especially like to thank my ally Petri, and long term office mates Joonas and Kalle for their refreshing company. I am grateful to our administrative staff for helping me in all sorts of work duties. I also had the pleasure to supervise the bachelor’s thesis of Marko Rinta-aho, which was an enlightening and educational experience for me.

I wish to express my deepest gratitude to my parents, Helga and Heimo, for their invaluable support throughout my life and education. I would like to thank my twin sister Milla, my big brother Olli, and the Raati fellows for enriching my working days via all of the daily communications. Finally, I am grateful to my lively nephew and godson Oskari for bringing joy to my life in general.

Oulu, March 22nd, 2018
Markus Leinonen
## List of abbreviations and symbols

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMM</td>
<td>Alternating direction method of multipliers</td>
</tr>
<tr>
<td>BA</td>
<td>Blahut-Arimoto</td>
</tr>
<tr>
<td>CS</td>
<td>Compressed sensing</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete cosine transform</td>
</tr>
<tr>
<td>DD</td>
<td>Dual decomposition</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier transform</td>
</tr>
<tr>
<td>DR</td>
<td>Distortion-rate</td>
</tr>
<tr>
<td>DSC</td>
<td>Distributed source coding</td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete wavelet transform</td>
</tr>
<tr>
<td>ECVQ</td>
<td>Entropy-constrained vector quantization</td>
</tr>
<tr>
<td>FCL</td>
<td>Flow conservation law</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency division multiple access</td>
</tr>
<tr>
<td>i.i.d.</td>
<td>Independent and identically distributed</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IRW-$\ell_1$</td>
<td>Iterative reweighted $\ell_1$-minimization</td>
</tr>
<tr>
<td>JSM</td>
<td>Joint sparsity model</td>
</tr>
<tr>
<td>KCS</td>
<td>Kronecker compressed sensing</td>
</tr>
<tr>
<td>LBG</td>
<td>Linde-Buzo-Gray</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum mean square error</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean square error</td>
</tr>
<tr>
<td>NN</td>
<td>Nearest-neighbor</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability density function</td>
</tr>
<tr>
<td>QCS</td>
<td>Quantized compressed sensing</td>
</tr>
<tr>
<td>RD</td>
<td>Rate-distortion</td>
</tr>
<tr>
<td>RDF</td>
<td>Rate-distortion function</td>
</tr>
<tr>
<td>RIP</td>
<td>Restricted isometry property</td>
</tr>
<tr>
<td>SI</td>
<td>Side information</td>
</tr>
<tr>
<td>SQ</td>
<td>Scalar quantization/quantizer</td>
</tr>
<tr>
<td>SSDG</td>
<td>Single-sink data gathering</td>
</tr>
<tr>
<td>SW</td>
<td>Slepian-Wolf</td>
</tr>
<tr>
<td>VQ</td>
<td>Vector quantization/quantizer</td>
</tr>
</tbody>
</table>
WSN  Wireless sensor network

**Mathematical symbols and operators:**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊙</td>
<td>Hadamard product</td>
</tr>
<tr>
<td>⊗</td>
<td>Kronecker product</td>
</tr>
<tr>
<td>⪰</td>
<td>Component-wise inequality</td>
</tr>
<tr>
<td>[·]⁺</td>
<td>Projection onto the set of non-negative reals</td>
</tr>
<tr>
<td>⌈·⌉</td>
<td>Rounding up to the nearest integer</td>
</tr>
<tr>
<td>{·}∗</td>
<td>The optimal value of the argument inside {·}</td>
</tr>
<tr>
<td>{·}ₖ</td>
<td>The value of an argument {·} at iteration k (ADMM)</td>
</tr>
<tr>
<td>{·}ₖ⁺</td>
<td>The value of an argument {·} at iteration k (DD)</td>
</tr>
<tr>
<td>{·}ᵀ</td>
<td>The value of an argument {·} at iteration t (BA)</td>
</tr>
<tr>
<td>·</td>
<td>An estimate of the argument inside {·}</td>
</tr>
<tr>
<td>f ∘ g</td>
<td>A composite mapping of functions f(·) and g(·) as f(g(·))</td>
</tr>
<tr>
<td>↘</td>
<td>The complement of set ↘</td>
</tr>
<tr>
<td>∅</td>
<td>An empty set</td>
</tr>
<tr>
<td>0</td>
<td>A column vector with all entries 0</td>
</tr>
<tr>
<td>1</td>
<td>A column vector with all entries 1</td>
</tr>
<tr>
<td>‖·‖₀</td>
<td>The operator that counts the number of non-zero entries of a vector</td>
</tr>
<tr>
<td>‖·‖₁</td>
<td>The ℓ₁-norm</td>
</tr>
<tr>
<td>‖·‖₂</td>
<td>The ℓ₂-norm</td>
</tr>
<tr>
<td>‖·‖₉</td>
<td>The Frobenius norm</td>
</tr>
<tr>
<td></td>
<td>a</td>
</tr>
<tr>
<td></td>
<td>lesai</td>
</tr>
<tr>
<td>A</td>
<td>A matrix comprising of the column vectors as A = [a₁ · · · aₙ]</td>
</tr>
<tr>
<td>aᵢⱼ</td>
<td>The entry in the iᵗʰ row and jᵗʰ column of matrix A</td>
</tr>
<tr>
<td>aᵢ</td>
<td>The iᵗʰ row of matrix A, i.e., aᵢᵀ = [a₁ᵢ · · · aₙᵢ]</td>
</tr>
<tr>
<td>aⱼᵢ</td>
<td>The jᵗʰ column of matrix A, i.e., aⱼᵢ = [a₁ⱼ · · · aₙⱼ]ᵀ</td>
</tr>
<tr>
<td>Aᵀ</td>
<td>The transpose of matrix A</td>
</tr>
<tr>
<td>A⁻¹</td>
<td>The inverse of matrix A</td>
</tr>
<tr>
<td>A⁻ᵀ</td>
<td>The transpose of the inverse of matrix A, i.e., A⁻ᵀ = (A⁻¹)ᵀ</td>
</tr>
<tr>
<td>argmin {·}</td>
<td>The argument that minimizes the term inside {·}</td>
</tr>
<tr>
<td>diag(·)</td>
<td>A (block) diagonal matrix with the argument values (matrices) on its diagonal</td>
</tr>
</tbody>
</table>
\[ E[\cdot] \] the expectation operator
\[ \text{max}\{\cdot\} \] the maximum element inside \{\cdot\}
\[ \text{rec}(l) \] the receiving node of link \( l \)
\[ \text{supp}(x) \] the support of vector \( x \), i.e., the indices of the non-zero elements of \( x \)
\[ \text{tran}(l) \] the transmitting node of link \( l \)
\[ \text{Tr}(A) \] the trace of matrix \( A \)
\[ \text{vec}(A) \] an operator that stacks the columns of matrix \( A = [a_1 \cdots a_N] \) into the vector form \( a = [a_1^T \cdots a_N^T]^T \)
\[ \mathcal{X}^m \] the \( m \)-fold Cartesian product of set \( \mathcal{X} \)
\[ Z \rightarrow Y \rightarrow X \] a Markov chain formed by random variables \( Z, Y, \) and \( X \)

General notations for random variables and random vectors:

\( X_l \) a random variable
\( x_l \) a realization of random variable \( X_l \)
\( X_l \) a random vector
\( X_{l,\mathcal{A}} \) a random vector consisting of elements of \( X_l \) indicated by index set \( \mathcal{A} \)
\( X_{l,n} \) the \( n \)th element of random vector \( X_l \)
\( x_l \) a realization of random vector \( X_l \)
\( x_{l,n} \) the \( n \)th element of realization \( x_l \)
\( \hat{X}_l \) a reproduction random vector for \( X_l \) at the output of a decoder
\( X^m \) a block of \( m \) consecutive random vectors \( \{X_n\}_{n=1}^m \)
\( x^m \) a block of \( m \) consecutive realizations \( \{x_n\}_{n=1}^m \)

Roman alphabet notations:

\( \tilde{A} \) a node-link incidence matrix
\( A \) a reduced node-link incidence matrix
\( \mathbb{B} \) the set of binary numbers
\( B_{\text{KCS}} \) a number of data blocks associated with data matrix \( X \) (KCS)
\( B_n \) a binary support random vector at time instant \( n \)
\( b_s \) the \( s \)th element of binary support alphabet \( \mathcal{B} \)
\( \mathbb{C} \) the set of complex numbers
\( \mathcal{CN}(\cdot,\cdot) \) circularly symmetric complex Gaussian distribution
\( C \) a normalized data transportation communication cost
$c_l(p_l, w_l)$ the capacity of link $l$

$C_0$ an oversampling constant in CS

$C_p$ a power-law decay constant

$c_{i_1, i_2}$ a reconstruction codevector of message index pair $(i_1, i_2)$ for the $l$th source

$c^{E_S-D_S}_{i_1}^{E_S-D_S}$ the $i_1$th reconstruction codevector for the $l$th source (VQ-E$_S$-D$_S$)

$c^{E_S-D_T}_{i_1, i_2}$ a reconstruction codevector of index pair $(i_1, i_2)$ for the $l$th source (VQ-E$_S$-D$_T$)

$c_i$ the $i$th reconstruction codevector (ECVQ-CS)

$\mathcal{D} (\cdot)$ a dual function

$\mathcal{D}_{\text{res}} (\cdot)$ the dual function associated with a resource allocation subproblem

$\mathcal{D}_{\text{rout}} (\cdot)$ the dual function associated with a routing subproblem

$d_0$ a reference distance

$d_{i_j}, d_{ns}$ the distance between nodes $i$ and $j$, sensor $n$ and source $s$ in a WSN

$d_{\text{tp}}^k$ the objective function value for a distributed algorithm in its last iteration

$d_l$ the length of link $l$

$d_{\text{rx}}$ the maximum transmission range of a sensor

$D$ a decoder / joint decoder

$D_{\text{si}}$ a decoder with side information

$D_{\text{vq}}$ a decoder mapping of a VQ

$D_l$ the average MSE reconstruction distortion associated with $l$th source

$D$ the average sum MSE reconstruction distortion as $D = D_1 + D_2$

$D_{pq}^l$ the average MSE quantization distortion associated with $l$th pre-quantizer

$D_{\text{ave}}$ the average normalized sum MSE reconstruction distortion of sources $X_1$ and $X_2$

$d(x, \hat{x})$ the average per-letter MSE distortion between realizations $x$ and $\hat{x}$

$D$ a distortion criterion

$D_s$ a distortion criterion for the $s$th subsource

$D'_s$ a reduced distortion criterion for the $s$th subsource

$D'_{s,k}$ a distortion variable

$D_{Z|b}$ average MMSE estimation error with respect to subsource $X_b$ with support $S$
$D_{Z|B}$ total average MMSE estimation error over all subsources with support $B$.

$D_Z$ average MMSE estimation error of $X$ given $Y$.

$\hat{d}(X, \hat{x}_j|v)$ a modified distortion measure defined as the average per-letter MSE distortion between $X$ and $\hat{x}_j$ conditioned on $V = v$.

$\exp\{x\}$ the natural exponential function $e^x$.

$E_n$ the number of hops from sensor $n$ to the sink in a routing tree.

$E$ an encoder.

$E_l$ an encoder of the $l$th sensor.

$E_{si}$ an encoder with side information.

$E_{vq}$ an encoder mapping of a VQ.

$f(x)$ the joint PDF of random vectors $X$ and $Y$.

$f(x|y)$ the conditional PDF of random vector $X$ given $Y$.

$f$ the data flow on link $l$.

$f$ a data flow vector $f = [f_1 \cdots f_L]^T$ in a WSN.

$\hat{f}_l$ a local copy of flow variable $f_l$ at sensor $i$.

$\hat{f}_l$ a local flow vector associated with sensor $i$.

$f_{l,\text{prox}}$ the flow on link $l$ at iteration $k$ (DD-px).

$f_{l,\text{quad}}$ the flow on link $l$ at iteration $k$ (DD-quad).

$\hat{f}_l$ an auxiliary optimization variable of link $l$.

$\tilde{f}$ a vector of auxiliary optimization variables $\tilde{f} = [f_1 \cdots f_L]^T$.

$f_{dft,s}$ a DFT-coefficient vector associated with a Gauss-Markov sequence $\lambda_s$.

$f_{dft,s}$ a low-pass filtered version of $f_{dft,s}$.

$F_{dct}$ a DCT-matrix.

$F_{dft}$ a DFT-matrix.

$FCL_{k,\text{viol}}$ a normalized FCL violation at iteration $k$.

$F_s$ an MMSE estimation matrix.

$G(\cdot)$ a directed graph.

$g_i(\cdot, \cdot, \cdot)$ a local cost function associated with sensor $i$.

$G^{(k)}(t)$ a diagonal weight matrix at decoding instant $t$ and iteration $k$.

$g_i^{(k)}(t)$ the $i$th diagonal entry of $G^{(k)}(t)$.

$g_{l,v_j}$ the $v_j$th codepoint in pre-quantization codebook $\mathcal{G}_l$.

$G_n$ a Gaussian random vector at time instant $n$.

$\tilde{G}$ a non-Gaussian random vector.
$g_E^m$ an encoder mapping
$g_{E_{si}}^m$ an encoder mapping with support SI
$g_D^m$ a decoder mapping
$g_{D_{si}}^m$ a decoder mapping with support SI
$H(X)$ the entropy of discrete random variable $X$
$H(X,Y)$ the joint entropy of a pair of discrete random variables $X$ and $Y$
$H(X|Y)$ the conditional entropy of $X$ given $Y$
$h_{ns}$ an influence function associated with sensor $n$ and source $s$
$h_{ns}(d_{ns})$ a distance-dependent influence function associated with sensor $n$ and source $s$
$h_n$ an aggregate influence vector on sensor $n$ as $h_n = [h_{n1} \cdots h_{nS}]^T$
$h_{i,j}$ the $i$th binary source codeword in codebook $\mathcal{H}_l$
$(i,j)$ a directed link between node $i$ and node $j$
$I_N$ the identity matrix of size $N \times N$
$I_\Gamma^T$ a binary sparse matrix as $I_\Gamma = [I_{W-1} \ 0_{(W-1) \times 1}]$
$I_n(t)$ an indicator function with value 1 if sensor $n$ transmits at slot $t$
$I(X;Y)$ the mutual information between random vectors $X$ and $Y$
$I(X;Y|Z)$ the conditional mutual information between random vectors $X$ and $Y$ given random vector $Z$
$I(X;Y|Z=z)$ the conditional mutual information between random vectors $X$ and $Y$ given a realization $Z = z$
$I_l$ a random variable representing the message index of the $l$th encoder
$I$ a random variable representing a quantization index of a VQ
$i_l$ an unused message index of the $l$th encoder
$K$ the sparsity of a signal
$k_{\text{max}}$ the maximum number of iterations (IRW-$\ell_1$)
$L$ the number of wireless links in a WSN
$\log(\cdot)$ logarithm
$\mathcal{L}_{\text{admm}}(\cdot)$ an augmented partial Lagrangian (ADMM)
$\mathcal{L}_{\text{dd}}(\cdot)$ a partial Lagrangian (DD)
$\mathcal{L}(\cdot)$ a Lagrangian (BA)
$L_\mu(\mathbf{D}, \mathbf{R})$ a weighted DR cost function
$M$ a number of CS measurements
$M(t)$ the number of CS measurements for time instant $t$
$M_l$ the number of CS measurements of sensor $l$
\( \mathcal{N}(\cdot, \cdot) \) Gaussian distribution

\( N \) the number of sensors in a WSN (Chapters 2 and 3) / signal length (Chapters 4 and 5)

\( N + 1 \) the sink node of a WSN

\( N_0 \) the power spectral density of an additive white Gaussian noise

\( p(x) \) the probability associated with a random variable \( X \) as \( p(x) \triangleq \Pr(X = x) \)

\( p(x, y) \) the joint probability associated with a pair of random variables \( X \) and \( Y \) as \( p(x, y) \triangleq \Pr(X = x, Y = y) \)

\( p(x|y) \) the conditional probability as \( p(x|y) \triangleq \Pr(X = x|Y = y) \)

\( p^* \) the optimal value of an optimization problem

\( p_I \) transmit power allocated to link \( I \)

\( p \) a transmit power vector \( p = [p_1 \cdots p_L]^T \) in a WSN

\( p_i \) a local transmit power vector associated with sensor \( i \)

\( P \) the maximum allowed transmit power for a sensor

\( p_I^{tx} \) the transmit probability of a sensor at time instant \( t \)

\( P_{mc} \) a state transition matrix of Markov chain sequence \( X_s \)

\( P_0, P_1 \) parameters for adjusting the entries of \( P_{mc} \)

\( \mathcal{P}Q_l \) the \( l \)th pre-quantizer

\( q_{sk} \) the \( k \)th state of state space \( \mathcal{Q}_s \) associated with Markov chain sequence \( X_s \)

\( Q_s \) a matrix containing eigenvectors of \( \Sigma_{X_s} \) in its columns

\( \tilde{Q}_s \) a matrix containing eigenvectors of \( \Sigma_{X_s} \) in its columns

\( \mathbb{R} \) the set of real numbers

\( \mathbb{R}^{++} \) the set of positive real numbers

\( r_i \) a source rate of sensor \( i \)

\( r_{N+1} \) a sink rate

\( \tilde{r} \) a rate vector \( \tilde{r} = [r_1 \cdots r_N r_{N+1}]^T \) in a WSN

\( r \) a reduced rate vector \( r = [r_1 \cdots r_N]^T \) in a WSN

\( \tilde{R}_l \) the pre-quantization rate of the \( l \)th pre-quantizer

\( R_l \) an average encoding rate of sensor \( l \)

\( \overline{R} \) the average sum rate of the sensors as \( \overline{R} = R_1 + R_2 \)

\( R_{ave} \) the average rate per sensor measured as \( R_{ave} = \overline{R}/2 \)

\( r(\cdot) \) a rate measure as a function of the probability of an index / index pair

\( R \) a rate defined as the bits/entry of source vector \( X \)

\( R_s \) a rate associated with subsouce \( X_s \)
\( R_{\text{rem}}(D) \) the remote RDF of source \( X \) for distortion \( D \)

\( R_{\text{rem}}^{X:B}(D) \) the conditional remote RDF of source \( X \) for distortion \( D \)

\( R_{\text{rem}}^{X,b_s}(D_s) \) the conditional marginal remote RDF of source \( X \) for a fixed realization \( B = b_s \) and distortion \( D_s \)

\( R_{\text{dir}}^{Z:B}(D_s) \) the RDF of the MMSE estimator \( Z_s \) for distortion \( D_s \)

\( R_{\text{dir}}^{X,b_s}(D) \) a numerically approximated remote RDF

\( R_{\text{dir}}^{X,B}(D) \) the conditional direct RDF of \( X \)

\( S \) the number of sources in a WSN

\( \mathbb{S}^N_+ \) the set of symmetric positive semi-definite \( N \times N \) matrices

\( \mathbb{S}^N_{++} \) the set of symmetric positive definite \( N \times N \) matrices

\( T \) a number of sampling instants in a WSN

\( V \) a random variable representing the quantization index of a VQ

\( V_l \) a random variable representing the quantization index of the \( l \)th pre-quantizer

\( v_l^{(t)} \) the cell index obtained by quantizing \( y_l^{(t)} \) at the \( l \)th pre-quantizer

\( V_{l,k} \) a Gaussian noise random variable of a Gaussian forward channel

\( w_l \) the bandwidth allocated to link \( l \)

\( w \) a bandwidth vector \( w = [w_1 \cdots w_L]^T \) in a WSN

\( w_i \) a local bandwidth vector associated with sensor \( i \)

\( W_i \) total available bandwidth for sensor \( i \)

\( W_{\text{net}} \) total available bandwidth in a WSN

\( W \) the size of data window \( X(t) \)

\( W_l \) a measurement noise random vector associated with sensor \( l \)

\( w_l^{(t)} \) the \( r \)th measurement noise training vector associated with sensor \( l \)

\( W_n \) a measurement noise random vector at time instant \( n \)

\( x \) a signal vector

\( x_n(t) \) a reading of sensor \( n \) at time instant \( t \)

\( X(t) \) a data window matrix at time instant \( t \)

\( x(t) \) the vector-reshaped data window \( X(t) \)

\( x_n(t) \) a vector of the \( n \)th sensor’s readings at time instants \( \{t - W + 1, \ldots, t\} \)

\( \mathbf{X}(t) \) a vector of sensors’ readings at time instant \( t \) in a WSN

\( \hat{\mathbf{X}}(t) \) a matrix-reshaped version of \( \mathbf{x}(t) \)

\( X_{B_1}(t) \) the first \( W - 1 \) columns of \( X(t) \)

\( \hat{X}_{B_1}^{(t-1)}(t) \) a decoder buffer at time instant \( t \)
\(\hat{x}_{B}^{(t-1)}(t)\) the vector-reshaped decoder buffer \(\hat{X}_{B}^{(t-1)}(t)\)

\(\hat{x}_n^{(t)}\) an estimate of \(x_n(t)\) obtained at decoding instant \(t\)

\(\hat{\mathbf{x}}^{(t)}\) an estimate of \(\mathbf{x}(t)\) obtained at decoding instant \(t\)

\(\mathbf{X}\) a matrix of sensors’ readings in a WSN

\(\mathbf{X}_{[l]}\) the \(l\)th data block of \(\mathbf{X}\) (KCS)

\(\mathbf{X}_l\) a source random vector associated with sensor \(l\)

\(\mathbf{X}_s\) a common component of sources \(\mathbf{X}_1\) and \(\mathbf{X}_2\)

\(\mathbf{X}_l'\) an innovation component of source \(\mathbf{X}_l\)

\(\mathbf{x}_l^{(t)}\) the \(t\)th source training vector associated with sensor \(l\)

\(\mathbf{X}_n\) a source random vector at time instant \(n\)

\(\mathbf{X}_{s,n}\) a subsourse random vector for support \(\mathbf{B} = \mathbf{b}_s\) at time instant \(n\)

\(\mathbf{X}_s\) the non-zero part of subsourse \(\mathbf{X}_s\)

\(\hat{\mathbf{X}}_s\) a subsourse whose non-zero elements are non-Gaussian

\(\check{\mathbf{X}}\) a sparse source whose non-zero elements are non-Gaussian

\(\hat{\mathbf{x}}_v\) the \(v\)th reconstruction codevector in \(\hat{\mathbf{X}}\)

\(\hat{\mathbf{X}}\) q a discrete reproduction random vector at the output of a decoder

\(\mathbf{X}_{\text{est}}\) a method-dependent decoded estimate of \(\mathbf{X}\)

\(\mathbf{y}\) a measurement vector

\(\mathbf{y}(t)\) a vector of CS measurements acquired by the sink at time instant \(t\)

\(\mathbf{y}(t)\) a concatenated measurement vector containing \(\mathbf{y}(t - W + 1), \ldots, \mathbf{y}(t)\)

\(\mathbf{Y}_l\) a CS measurement random vector of sensor \(l\)

\(\mathbf{y}_l^{(t)}\) the \(t\)th measurement training vector associated with sensor \(l\)

\(\hat{y}_{l,i}\) the \(i\)th nearest-neighbor codepoint of the \(l\)th encoder (VQ-E\_S-D\_S)

\(\hat{y}_l\) the \(l\)th nearest-neighbor encoder codepoint (VQ-CE)

\(\mathbf{Y}_n\) a measurement random vector at time instant \(n\)

\(\mathbf{Y}_{s,n}\) a measurement random vector associated with subsourse \(\mathbf{X}_s\) at time instant \(n\)

\(\mathbb{Z}\) the set of integer numbers

\(\mathbf{Z}(t)\) a joint transform coefficient matrix at time instant \(t\)

\(z(t)\) the vector-reshaped version of \(\mathbf{Z}(t)\)

\(\check{z}\) a vector of optimization variables representing \(z(t)\)

\(\check{z}^{(k)}(t)\) an estimate of \(z(t)\) obtained at decoding instant \(t\) and iteration \(k\)

\(\mathbf{z}_{t,v_1,v_2}\) the centroid of the MMSE estimates of \(\mathbf{X}_t\) given such measurements

\(\mathbf{Y}_1 = \mathbf{y}_1\) and \(\mathbf{Y}_2 = \mathbf{y}_2\) that are pre-quantized to cell index pair \((v_1, v_2)\)

\(\tilde{\mathbf{X}}_s\) the MMSE estimator of \(\mathbf{X}_s\) given \(\mathbf{Y}_s\)
\(Z_s\) the MMSE estimator of \(X_s\) given \(Y_s\)
\(Z'_s\) a decorrelated MMSE estimator \(Z'_s = Q'_s Z_s\)
\(Z\) the MMSE estimator of \(X\) given \(Y\)

Greek alphabet notations:

\(\alpha_k^\nu, \alpha_k^\xi\) subgradient step sizes associated with \(\nu\) and \(\xi\) at iteration \(k\)
\(\alpha_{\nu,sc}, \alpha_{\xi,sc}\) positive parameters for adjusting diminishing step sizes \(\alpha_k^\nu\) and \(\alpha_k^\xi\)
\(\alpha_s\) a correlation parameter for Gauss-Markov sequence \(\lambda_s\)
\(\alpha_0\) a mapping from a set of message indices to binary codewords for the \(l\)th source
\(\alpha_s^{-1}\) the inverse operation of \(\alpha_s\)
\(\alpha_{pl}\) a parameter for adjusting the concentration of a PMF
\(\beta_s(t)\) the magnitude of source \(s\) at time instant \(t\)
\(\beta(t)\) a source magnitude vector \(\beta(t) = [\beta_1(t) \cdots \beta_S(t)]^T\) at time instant \(t\)
\(\beta_s\) a magnitude vector of source \(s\)
\(\beta_l\) a mapping from a set of message index pairs to reconstruction codevectors for the \(l\)th source
\(\gamma_l\) a channel condition parameter of link \(l\)
\(\gamma_B\) a regularization weight parameter (Seq-Prog-CS)
\(\gamma_{reg}\) a regularization weight parameter (Reg-Mod-CS)
\(\gamma(i)\) the length of binary codeword \(h_{l,i}\)
\(\delta_{dd}\) a positive parameter for proximal regularization (DD-px)
\(\epsilon\) a positive real number
\(\epsilon_{dd}\) a positive parameter for quadratic regularization (DD-qd)
\(\epsilon_0\) a positive stability parameter (IRW-\(\ell_1\))
\(\epsilon_G\) a positive convergence criterion parameter (IRW-\(\ell_1\))
\(\epsilon_{ba}\) a positive stopping criterion parameter (BA)
\(\eta_{pf}\) a low-pass filter parameter
\(\theta\) a transform domain coefficient vector
\(\theta(i)\) the \(i\)th largest element of \(\theta\) in magnitude
\(\tilde{\theta}\) a vector of optimization variables representing \(\theta\)
\(\theta_S(t)\) a spatial transform coefficient vector for \(x(t)\)
\(\Theta_S(t)\) a spatial transform coefficient matrix for \(X(t)\)
\(\theta_{T,n}(t)\) a temporal transform coefficient vector for \(x_n(t)\)
\( \Theta_T(t) \) a temporal transform coefficient matrix for \( X(t) \)

\( \theta_S \) a vector of optimization variables representing \( \theta_S(t) \)

\( \Theta_S(t) \) a spatial transform coefficient matrix for \( X(t) \)

\( \theta_{s,k} \) a gain parameter of a Gaussian forward channel

\( \kappa_0 \) a complex filter tap of link \( l \)

\( \kappa_t \) the last iteration of the IRW-\( l_1 \) at decoding instant \( t \)

\( \lambda_{s,l} \) a Lagrange multiplier associated with a consensus constraint of sensor \( i \)

\( \lambda_s \) a Gauss-Markov sequence \( \lambda_s = [\lambda_s(1) \cdots \lambda_s(T)]^T \) for source \( s \)

\( \hat{\lambda}_s \) a low-pass filtered version of \( \lambda_s \)

\( \Lambda_s \) a diagonal matrix containing the eigenvalues of \( \Sigma_{Z_s} \)

\( \lambda_{s,k} \) the \( k \)th eigenvalue of \( \Sigma_{Z_s} \)

\( \hat{\Lambda}_s \) a diagonal matrix containing the eigenvalues of \( \Sigma_{X_s} \)

\( \lambda_{s,k} \) the \( k \)th eigenvalue of \( \Sigma_{X_s} \)

\( \lambda \) a Lagrange multiplier associated with a sum distortion constraint (BA)

\( \mu \) a weighting parameter for adjusting the DR trade-off (DQCS-PQ)

\( \mu_c \) a non-negative parameter for adjusting \( \mu \)

\( \mu_r \) the mean component of Gauss-Markov sequence \( \lambda_s \)

\( \mu^{ec} \) a weighting parameter for adjusting the DR trade-off (ECVQ-CS)

\( \nu_l \) a dual variable associated with a capacity constraint of link \( l \) (DD)

\( \nu \) a vector of dual variables \( \nu = [\nu_1 \cdots \nu_L]^T \)

\( \nu_v \) a Lagrange multiplier associated with a conditional probability constraint (BA)

\( \xi_{cs} \) a CS reconstruction error

\( \xi_i \) a dual variable associated with an FCL constraint of sensor \( i \) (DD)

\( \xi \) a vector of dual variables \( \xi = [\xi_1 \cdots \xi_N]^T \)

\( \pi_l \) a message index mapping for the \( l \)th encoder

\( \pi_l(v_l) \) the message index associated with the \( v_l \)th cell index

\( \pi_l^{-1}(i_l) \) a set of cell indices mapped to the \( i_l \)th message index

\( \rho \) the ADMM penalty parameter

\( \rho_1, \rho_2 \) parameters for adjusting spatial correlation of a sensor data

\( \rho_{corr} \) a parameter for adjusting inter-sensor correlation in \( X_1 \) and \( X_2 \)

\( \sigma_X^2 \) the variance of random variable \( X \)

\( \Sigma_X \) the covariance matrix of random vector \( X \)

\( \Sigma_{XY} \) the cross-covariance matrix of random vectors \( X \) and \( Y \)
a noise signal at the receiver of link \((i, j)\)

the \(r\)th innovation component of Gauss-Markov sequence \(\lambda_s\)

a measurement matrix with respect to \(\mathbf{x}(t)\)

a block-diagonal measurement matrix with respect to \(X(t)\)

a measurement matrix of sensor \(l\)

a measurement matrix with respect to signal \(x\) / source \(X\)

a measurement matrix with respect to subsourc \(X_s\)

a Markov chain sequence \(\mathbf{X}_s = [\chi_s(1) \cdots \chi_s(T)]^\top\) for source \(s\)

a sparsifying basis / a Kronecker sparsifying basis

a sparsifying basis for the spatial domain

a sparsifying basis for the temporal domain

the first \(W - 1\) rows of \(\Psi_T\)

the first \(N(W - 1)\) rows of \(\Psi\)

Calligraphy letter notations:

\(\mathcal{B}\) a discrete alphabet of binary support vectors

\(\mathcal{C}_l\) a reconstruction codebook for source \(l\)

\(\mathcal{F}\) a feasible set of variables

\(\mathcal{F}_i\) a feasible set of local variables for sensor \(i\)

\(\mathcal{G}_l\) a pre-quantization codebook of the \(l\)th pre-quantizer

\(\mathcal{H}_l\) a binary source codebook associated with sensor \(l\)

\(\mathcal{I}(i)\) a set of incoming links of node \(i\)

\(\mathcal{I}_l\) a set of message indices associated with sensor \(l\)

\(\mathcal{J}\) a set of quantization indices for a VQ

\(\mathcal{L}\) a set of directed wireless links

\(\mathcal{L}'(i)\) a set of links connected to node \(i\)

\(\mathcal{N}\) a set of sensors

\(\tilde{\mathcal{N}}\) a set of nodes consisting of the sensors and the sink

\(\mathcal{N}(i)\) a subset of sensors that report their readings at time instant \(t\)

\(\mathcal{O}(i)\) a set of outgoing links of node \(i\)

\(\mathcal{Q}_s\) a state space of Markov chain sequence \(\mathbf{X}_s\)

\(\mathcal{R}\) the closure of a set of achievable \((R, D)\) pairs with respect to \(R_X^{\text{rem}}(D)\)

\(\mathcal{R}_{si}\) the closure of a set of achievable \((R, D)\) pairs with respect to \(R_{X,B}^{\text{rem}}(D)\)

\(\mathcal{S}\) a set of sources
\( S_{l,vl} \) the \( vl \)th cell of the \( l \)th pre-quantizer

\( S_v \) the \( v \)th encoder region of a VQ

\( T_s \) an index set of the \( s \)th sparsity pattern

\( \mathcal{U} \) a set of message indices

\( \mathcal{V}_l \) a set of cell indices of the \( l \)th pre-quantizer

\( \mathcal{V} \) a set of quantization indices for a VQ

\( \mathcal{X} \) a source alphabet

\( \hat{\mathcal{X}} \) a reproduction alphabet

\( \hat{\mathcal{X}}^q \) a reconstruction codebook of a VQ

\( \mathcal{Y} \) a measurement vector space
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>9</td>
</tr>
<tr>
<td>Tiivistelmä</td>
<td></td>
</tr>
<tr>
<td>Preface</td>
<td>11</td>
</tr>
<tr>
<td>List of abbreviations and symbols</td>
<td></td>
</tr>
<tr>
<td>Contents</td>
<td>25</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>29</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>29</td>
</tr>
<tr>
<td>1.2 Scope and objectives of the thesis</td>
<td>30</td>
</tr>
<tr>
<td>1.3 Literature review and thesis outline</td>
<td>31</td>
</tr>
<tr>
<td>1.3.1 Distributed cross-layer optimization in WSNs</td>
<td>31</td>
</tr>
<tr>
<td>1.3.2 Compressed sensing based data gathering in WSNs</td>
<td>36</td>
</tr>
<tr>
<td>1.3.3 Compressed sensing based source compression in WSNs</td>
<td>43</td>
</tr>
<tr>
<td>1.4 Contributions of the thesis</td>
<td>49</td>
</tr>
<tr>
<td>1.4.1 Key contributions and related works</td>
<td>49</td>
</tr>
<tr>
<td>1.4.2 Author’s contribution to the publications</td>
<td>50</td>
</tr>
<tr>
<td>2 Distributed joint resource and routing optimization</td>
<td>51</td>
</tr>
<tr>
<td>2.1 Data gathering wireless sensor network</td>
<td>51</td>
</tr>
<tr>
<td>2.1.1 Network topology</td>
<td>51</td>
</tr>
<tr>
<td>2.1.2 Multi-path routing model</td>
<td>53</td>
</tr>
<tr>
<td>2.1.3 Communication model</td>
<td>54</td>
</tr>
<tr>
<td>2.2 Joint resource and routing optimization</td>
<td>55</td>
</tr>
<tr>
<td>2.2.1 Problem formulation</td>
<td>56</td>
</tr>
<tr>
<td>2.2.2 Distributed optimization via consensus ADMM</td>
<td>57</td>
</tr>
<tr>
<td>2.2.3 Distributed optimization via DD</td>
<td>66</td>
</tr>
<tr>
<td>2.2.4 Alternative decomposition approaches</td>
<td>70</td>
</tr>
<tr>
<td>2.3 Numerical results</td>
<td>71</td>
</tr>
<tr>
<td>2.3.1 Simulation setup</td>
<td>73</td>
</tr>
<tr>
<td>2.3.2 Convergence of distributed algorithms</td>
<td>73</td>
</tr>
<tr>
<td>2.3.3 System performance under different bandwidth allocation schemes</td>
<td>79</td>
</tr>
<tr>
<td>2.4 Summary and discussion</td>
<td>81</td>
</tr>
</tbody>
</table>
3 Sequential compressed sensing with progressive signal reconstruction

3.1 Wireless sensor network monitoring
  3.1.1 Source model
  3.1.2 Energy consumption model

3.2 Sequential CS of correlated sensor data streams
  3.2.1 Sliding data window
  3.2.2 Exploiting joint spatio-temporal correlation
  3.2.3 CS encoding and decoding

3.3 Progressive CS signal reconstruction with prior information
  3.3.1 Modified CS reconstruction problem
  3.3.2 Algorithm summary

3.4 Simulation results
  3.4.1 Simulation setup
  3.4.2 Performance of the proposed method

3.5 Summary and discussion

4 Distributed variable-rate quantized compressed sensing

4.1 System model
  4.1.1 CS signal acquisition
  4.1.2 Measurement space pre-quantization
  4.1.3 Encoding and decoding
  4.1.4 Entropy coding

4.2 Distributed variable-rate QCS method
  4.2.1 Problem formulation
  4.2.2 Optimization of pre-quantizers
  4.2.3 Alternating optimization of encoders & decoder
  4.2.4 Algorithm implementation

4.3 Numerical results
  4.3.1 Simulation setup
  4.3.2 An illustrative example
  4.3.3 Distortion-rate performance
  4.3.4 Influence of the weight parameter
  4.3.5 Influence of source correlation
  4.3.6 Influence of number of measurements

4.4 Summary and discussion
5 Rate-distortion performance of lossy compressed sensing

5.1 Lossy CS via remote source coding

5.1.1 Source model

5.1.2 Noisy CS

5.1.3 Lossy CS problem

5.1.4 Remote RDF

5.2 Rate-distortion lower bound for lossy CS

5.2.1 Lossy CS problem with support SI

5.2.2 Conditional remote RDF

5.2.3 Conditional marginal remote RDF

5.2.4 Characterization of the conditional remote RDF

5.3 Numerical approximation of the remote RDF

5.3.1 Discretization of signal alphabets

5.3.2 Modified Blahut-Arimoto algorithm for lossy CS

5.3.3 Vector quantization optimization

5.4 "Estimate-and-compress” QCS method

5.5 Numerical results

5.5.1 Simulation setup

5.5.2 Rate-distortion behavior of lossy CS

5.5.3 Effect of the number of measurements

5.5.4 Effect of support probabilities

5.6 Summary and discussion

6 Conclusions and future work

6.1 Conclusions

6.2 Future work

References

Appendices
1 Introduction

1.1 Motivation

It is anticipated that there will be a burgeoning demand for the deployment of low-power smart sensors in the near future. In particular, they are expected to serve the myriad of diverse Internet of Things (IoT) applications, e.g., for environmental, industrial, healthcare, and military monitoring tasks [1–3]. The IoT opens a new era of smart user-friendly networking, where collaborative sensors gather advanced environmental data with no human intervention and provide the information to an underlying application, for example, to improve user experience, system performance and the quality of service, and reduce maintenance costs. By 2020, the number of IoT devices is envisioned to reach hundreds of billions [2, 3]. At the same time, the IoT market will become worth an amount in the order of trillions of dollars, with a major portion consisting of healthcare applications [2, 3].

Wireless sensor networks (WSNs) consisting of battery-powered sensors will be a key technology in creating a ubiquitous networked world and smart cities under the IoT framework [3]. In a typical monitoring task, geographically distributed sensors measure a correlated information source, encode the observations separately, and disseminate the information to a sink for joint decoding of the source signals. Limited batteries, which are often non-rechargeable or irreplaceable, dictate the lifespan of a WSN. The main contributors to a sensor’s energy consumption are wireless communications [4], and in some setups, the sensing/sampling part [5]. Consequently, a well-designed sensor should acquire just the required (small) amount of data samples of a physical phenomenon of interest (e.g., temperature, humidity, or light intensity), which, after being encoded and communicated to the sink at a minimal rate, provides sufficient information to reconstruct the signal under required criteria. Such efficient communication phases enable each sensor to shut down its radio or switch to a low-power state to further conserve energy [4]. This engenders the need for energy-efficient distributed data acquisition and gathering techniques that preserve the autonomous operation of sensors and have a simple infrastructure with low battery consumption and computational complexity.
1.2 Scope and objectives of the thesis

Targets: The objective of this thesis is to develop and analyze different energy-efficient distributed data acquisition and compression techniques for future WSNs. The thesis proposes four design approaches, which are outlined in detail in the next section. An overarching aim of the approaches – which include both algorithm development and theoretical performance analysis – is to reduce the sensors’ energy consumption in a data gathering application. This is accomplished by minimizing the amount of information (i.e., the number of data units or bits) that must be communicated from each sensor to the sink to satisfy given application requirements.

Methods: Each design was carried out under a theoretical framework using well-established mathematical tools of the related field. The main theoretical foundations are i) convex optimization and optimization decomposition techniques (Chapter 2), ii) compressed sensing (CS) (Chapters 3–5), and iii) the rate-distortion (RD) theory (Chapters 4–5). Validation of the derived theoretical results, and performance evaluation of the devised methods and algorithms were performed via MATLAB simulations. The design criteria for the distributed algorithms were to deliver a low degree of computational complexity at each sensor, a minimal number of information exchanges between sensors, fast and robust convergence to near-optimal solutions, and a high compression ratio with accurate signal reconstruction. The restrictions induced by limited resources and the distributed nature of WSNs were specifically taken into account in the algorithm design phase to facilitate their practical implementation. The implementation aspects were further studied in a series of complexity analyses. A common asymmetric WSN design was adopted by assuming that a sink would have sufficient computational power and an abundant supply of energy to accomplish the decoding tasks; i.e., the energy minimization of the decoder side is beyond the scope of this thesis.

Significance: The research work is expected to influence the development and implementation of energy-efficient decentralized networking and compressed data acquisition methods by proposing various solutions to the growing demands of future ubiquitous WSNs. The proposed techniques were demonstrated to have great potential to reduce energy consumption of sensor devices. In future IoT solutions, extending the battery lives will be crucial as it will prolong the network lifetime, relieve human intervention, reduce maintenance costs, and improve user experience. Since the utilization of WSNs is anticipated to explode in the near future via the advent of
diverse IoT applications, energy-optimized solutions are in the interests of ecological technology development as well. The derived theoretical results have particular significance in establishing performance limits for practical methods and can, thus, be used for benchmarking and algorithm verification purposes. Finally, all the presented design principles, novel ideas, and comprehensive listings of possible extensions are anticipated to lay the foundations for future exploration in the related research fields.

1.3 Literature review and thesis outline

The thesis is organized as follows. Chapters 2–5 contain the main four design frameworks (see Fig. 1), and Chapter 6 concludes the thesis and discusses future research directions. This section reviews the existing literature and the main tools related to the proposed designs. In particular, Section 1.3.1 addresses the approach in Chapter 2, Section 1.3.2 addresses the approach in Chapter 3, and Section 1.3.3 addresses the approaches in Chapters 4 and 5. Each section is concluded with a detailed description of each proposed approach in this thesis.

1.3.1 Distributed cross-layer optimization in WSNs

Basic concepts

The OSI model: The basis for a traditional layered network architecture design is the Open Systems Interconnection (OSI) model [6]; for an insightful review of the standard see [7]. The OSI model consists of seven layers as depicted in Fig. 1: the application, presentation, session, transport, network, data link, and physical layer. Each layer of an OSI based system is designed independently by assuming strict inter-layer boundaries. A layer serves only its upper layer and does not pass any information to the other layers. A benefit of layering is that an algorithm that realizes the required service at each layer can flexibly be modified in isolation after implementation.

Layer coupling in WSNs: An OSI based design is typically a justified choice for wired networks. However, in general it is suboptimal for wireless networks and simplicity-driven WSNs in particular due to an inherent coupling and interplay between the protocol layers. For instance, consider industrial monitoring in which the application requests the sensors to convey more accurate temperature reading values than the current ones. This increases the bit rate target for each sensor. In order to
meet new demands, the sensors may react for example by increasing their transmit powers and allocating more bandwidth (physical layer), as well as rescheduling the transmissions and readjusting the medium access method (data link layer). Such adjustments may further necessitate reorganizing the global multi-hop routing structure (network layer) to meet the system criteria in terms of energy minimization, the quality of service, reconstruction accuracy, or a maximum network delay etc.

Centralized cross-layer design: A countermeasure to the layer coupling of WSNs is cross-layer design. By overriding the inter-layer boundaries, and sophisticatedly sharing and utilizing the information from distinct OSI layers, the overall system performance can be improved. Varying definitions, interpretations, and classes of cross-layer design were surveyed in [8]. In a straightforward centralized cross-layer implementation, a sink – abundant with computational resources – gathers necessary parameters from network nodes, performs centralized optimization, and disseminates the updated parameters back to the nodes. Drawbacks of centralized processing are
that the incurred overhead can significantly exhaust the sensors’ energy, and solution updates to account for network changes may be too slow.

**Mathematical optimization:** From the mathematical point of view, a cross-layer design is realized via the joint optimization of a set of parameters of distinct layers. A well-established tool for modelling and solving various networking problems is convex optimization [9]. If a desired system objective can be formulated as a convex optimization problem, its (globally) optimal solution can be found via many efficient methods such as the interior-point methods [9]. Some designs, however, inevitably result in non-convex problems which can be computationally intractable to solve. In such cases, convex optimization can assist in developing heuristic methods to find approximate or locally optimal solutions.

**A distributed cross-layer framework:** Due to the decentralized, autonomous nature of WSNs, in particular there is a call for a distributed cross-layer design. In brief, an optimization task is parallelized so that each sensor solves a simple local subproblem and exchanges only a few messages with its neighboring nodes to steer a sequence of node-centric solutions towards the global goal. Similarly to a centralized approach, the necessary communications in an optimization phase consume battery energy. However, optimization is executed only at the start-up, or when the operating environment considerably changes. In the latter case, the solution can cost-effectively be updated owing to the available "warm start" state. Hence, a distributed approach is conducive to saving energy because of the reduced network overhead and adaptation to parameter changes.

**Optimization decomposition techniques:** A distributed cross-layer design gives rise to decomposition techniques, which parallelize the solution process into multiple subproblems by taking advantage of the special structure of the problem [10, Sect. 3.4]. A concept known "layering as optimization decomposition" was introduced in [11, 12]. Two of the most common techniques behind distributed networking algorithms are the primal decomposition and the dual decomposition (DD) [13, 14]. Other techniques include consensus optimization [10, Sect. 3.4] and [15, 16], gossip algorithms [17, 18], and distributed Newton methods [19]. The two seminal papers by Kelly et al. [20] and Low and Lapsley [21] laid the ground for many successive primal and dual based distributed cross-layer designs. A recent survey on existing cross-layer designs in wireless networks can be found in [22], and a tutorial on network utility maximization in [23].
Fig. 1 illustrates the connections of each thesis chapter to the OSI model. The cross-layer design in Chapter 2 covers the physical and network layers; the optimization of the radio resources – the transmit power and bandwidth of each link – pertains to the physical layer, whereas the end-to-end transmissions between sensors via a multi-path routing protocol are handled by the network layer. In particular, the cross-layer approach conforms to the "back and forth information flow" type specified in [8]. Regarding the presented optimization frameworks, all problems that are solved in this thesis are convex. The different decomposition techniques and related tools that are applied in Chapter 2 are elaborated next.

**Dual decomposition**

DD is an extensively studied method in distributed wireless network optimization. It can be traced back to the 1960s with the first appearances in [24, 25]. In DD, the original *primal problem* is solved by solving its *(Lagrange) dual problem*. In a WSN, the dual problem must be solved *iteratively* in a distributed fashion. To this end, the dual problem is first decomposed into smaller per-sensor subproblems (termed a *horizontal decomposition*). Then, at each iteration, each sensor solves its subproblem and exchanges a small number of variables with adjacent nodes. The network-wise coordination of these subproblems is often realized via the *subgradient method* [26] and [27, Ch. 6], originally developed by Shor and Zhurbenko in 1970s [28]. Tutorials on Lagrange duality based cross-layer optimization in wireless networks can be found in [13, 29]. Besides horizontal decomposition, DD can also separate the dual problem with respect to the OSI layers, which is termed the *vertical decomposition* [12].

Numerous DD algorithms have been developed for distributed joint optimization over the network and physical layers in wireless multi-hop networks. To mention a few of these methods, joint routing and capacity assignment was proposed in [30], simultaneous resource allocation and routing in [31], joint optimization of source quantization, power allocation and routing in [32], joint routing, relay selection and power allocation in [33], maximum lifetime routing in [34], and joint optimization of routing, random access, and power allocation in [35]. These works solve the dual problems via the subgradient method. However, the positive aspect of being simple has its drawbacks: the subgradient methods are known to suffer from slow convergence [19]. Moreover, when the objective function is not strictly convex in some optimization variable(s), recovering the optimal primal variables from the dual
solution is difficult [10, Sect. 3.4]. As a remedy, a small quadratic regularization term can be added into an objective to make it strictly convex [31, 33, 36]. The added term, however, introduces a trade-off between the convergence rate and solution accuracy. More sophisticated means include the proximal minimization algorithm and augmented Lagrangian methods [10, Sect. 3.4] and [37].

**Consensus optimization**

A comprehensive background on consensus optimization, which rests solely on the horizontal decomposition, can be found in, e.g., [10, Sect. 3.4] and [15]. A consensus mechanism that distributes the computation across multiple network agents appears already in [38]: the initially dissimilar local estimates on a shared "global variable" are asymptotically made equal between the entities through information exchanges and appropriate combinations of the decisions. Using consensus optimization, Nedić and Ozdaglar [39] proposed a distributed subgradient method for unconstrained minimization of a sum of per-agent cost functions. As extensions, the works in [15] and [16] proposed distributed projected consensus/subgradient algorithms to solve problems involving local constraints.

**Alternating direction method of multipliers**

A powerful method to solve distributed consensus problems involving shared variables and/or coupling constraints is the alternating direction method of multipliers (ADMM) [40]. This was originally developed by Glowinski and Marrocco [41], and Gabay and Mercier [42] in the mid-1970s, albeit its principles can be traced back to as early as the 1950s. The method has recently aroused widespread interest in distributed optimization, thanks to the extensive monograph by Boyd et al. in [40]. ADMM blends the advantageous properties of DD (decomposability) and the method of multipliers (robustness), resulting in superior convergence under more general conditions compared to DD. As an augmented Lagrangian method, ADMM converges without strict convexity or the finiteness of an objective function [40, Sect. 2.3].

ADMM applied to general separable consensus problems was studied in [43]. Such a consensus ADMM method has been adopted in WSN tasks, e.g., distributed estimation [44] and event monitoring [45]. As for theoretical works, the convergence proofs of ADMM applied to separable optimization problems have been established in, e.g., [46,
The linear convergence rate of consensus ADMM applied to a problem with strictly convex local objective functions has been proved in [48]. Additionally, the convergence of ADMM has been analyzed for solving nonconvex consensus and sharing problems in [49] and nonconvex structured problems in [50].

The proposed approach of the thesis

Chapter 2 is based on the published journal article [51] © 2013 IEEE, and the initial works presented in three conference papers [52] © 2011 IEEE, [53] © 2012 IEEE, and [54] © 2012 IEEE. The chapter addresses a (decentralized) cross-layer design to minimize the sensors’ sum transmit power via the joint optimization of transmit power and bandwidth allocation, and multi-path routing for fixed source rates. A novel distributed consensus ADMM algorithm is proposed for solving the problem. Additionally, a refined DD algorithm relying on proximal regularization is proposed. Based on numerical experiments, the ADMM algorithm is suitable for energy-efficient implementation due to its fast convergence with only slight local messaging for a range of step size parameters and it has the ability to scale up to large networks. The results also show that significant system performance gains are obtained via the used bandwidth allocation scheme.

1.3.2 Compressed sensing based data gathering in WSNs

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 [55] and Shannon [56]. 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\(^1\) 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

\(^1\)Data compression is discussed in more detail in Section 1.3.3.
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. 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. In WSNs, sparse signals are encountered in many applications, e.g., environmental monitoring [57, 58], source localization [59], biomedical sensing [60], and spectrum sensing and direction of arrival estimation in cognitive radio communications [61]. Sparsity can be utilized by CS [62–70] – the recently emerged joint sampling and compression paradigm. In CS, a sparse/compressible length-$N$ signal can be accurately reconstructed from its $M < N$ (random) linear measurements. This engenders the sub-Nyquist sampling interpretation of CS [71]: instead of sampling at a rate proportional to the signal bandwidth, the sampling rate in CS is dictated by the signal’s “information content” [72].

For WSN applications, the primary asset of CS is its simple and universal encoding since most computational work load is shifted to the decoder [68]. 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 $N\log N$ [73, 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 [74]. 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 [68].

In this thesis, CS plays a focal role in Chapters 3–5. Chapter 3 focuses on efficient distributed CS based data gathering of real-valued sparse signals. Chapters 4 and 5 incorporate quantization of CS measurements and focus on quantized CS (QCS) and lossy source coding. In this thesis, 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. 2). Another line of work is analog-to-digital compression, where analog signals are encoded into bits via a combined sub-Nyquist sampling and quantization
A hardware implementation of a sub-Nyquist sampling system was presented in [76], and a unified Xampling framework was introduced in [77].

The fundamentals of CS and the literature review on CS data gathering related to Chapter 3 are presented in the next subsections. The background on QCS related to Chapters 4 and 5 are discussed in Section 1.3.3. Finally, in reference to the OSI model in Fig. 1, the sampling operation of CS can be considered to belong to the physical layer, and the compression to the presentation layer.

**CS fundamentals**

The standard CS problem is introduced herein. Let $x \in \mathbb{R}^N$ be a real-valued vector that can be represented in basis $\Psi \in \mathbb{R}^{N \times N}$ as $x = \Psi \theta$, where $\theta = [\theta_1 \ldots \theta_N]^T$ is the transform domain coefficient vector. Vector $x$ is $K$-sparse in basis $\Psi$ if $\theta$ has at most $K \leq N$ non-zero entries, i.e., $\|\theta\|_0 \leq K$, where $\|\theta\|_0 = |\text{supp}(\theta)|$, $\text{supp}(\theta) = \{i|\theta_i \neq 0, i = 1, \ldots, N\}$ is the support of $\theta$, 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 $\theta$ is concentrated, i.e., the ordered coefficients $|\theta_i| \geq \ldots \geq |\theta_N|$ 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 $p$ [64, 66, 67, 78, 79]. This signal class includes smooth and piecewise smooth signals, and images with bounded variations [64, 67].

A CS based sensor acquires $M \leq N$ linear measurements $y \in \mathbb{R}^M$ of $x$ as

$$y = \Phi x = \Phi \Psi \theta,$$

where $\Phi \in \mathbb{R}^{M \times N}$ 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. 2.

For reconstruction, one seeks the sparsest $x$ that is consistent with measurements $y$. For $M \geq 2K$, a measurement matrix $\Phi$ exists so that the optimal solution can be found via a combinatorial (non-deterministic polynomial-time) hard problem [73, Theorem 2.14]

$$\hat{\theta} := \arg \min_{\theta} \|\hat{\theta}\|_0 \quad \text{such that} \quad y = \Phi \Psi \hat{\theta},$$

where $\hat{\theta}$ is the estimated vector of the sparsest $x$. This is an NP-hard problem, and various approximation algorithms have been developed to solve it efficiently.
resulting in an estimate $\hat{x} = \Psi \hat{\theta}$. Owing to the main result of CS, the sparsity-promoting $\ell_0$-term is replaced by its best convex approximation, the $\ell_1$-norm, and one solves the basis pursuit problem [63, 64, 66, 67, 80–82]

$$\hat{\theta} := \text{argmin}_{\hat{\theta}} \|\hat{\theta}\|_1 \text{ such that } y = \Phi \Psi \hat{\theta},$$

(3)
yielding an estimate $\hat{x} = \Psi \hat{\theta}$. Since (3) can be cast as a linear program, it can be solved via polynomial-complexity solvers [79, 81]. Note that the encoding (1) is universal in the sense that an appropriate $\Psi$ is selected only in the decoding stage (3).

Two essential properties of $\Phi$ and $\Psi$ which play important roles in stable and accurate CS signal recovery are the restricted isometry property (RIP) of $\Phi \Psi$ [70, 83, 84], [85, Sect. 1.4.2] and [73, Ch. 6], and the mutual coherence between $\Phi$ and $\Psi$ [70, 81, 84, 86] and [85, Sect. 1.4.3]. For instance, $\Phi$ with i.i.d. Gaussian or binary entries is highly incoherent with any basis $\Psi$ 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 [69, 70] and [73, Ch. 1].

**CS reconstruction methods**

While the $\ell_1$-minimization (3) is admittedly the trademark of CS decoding, there exist a variety of sparse signal reconstruction algorithms of differing complexity and performance. Table 1 highlights a collection of the main classes. More algorithms are listed in, e.g., [85, Sect. 1.6], [73, Sect. 1.3] and [87, Sect. 1.3.2], and a recent review

![Fig. 2. A CS measurement setup for $N = 9$, $M = 5$, and $K = 3$.](image-url)
### Table 1. Various classes of existing CS reconstruction algorithms.

<table>
<thead>
<tr>
<th>Class</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy methods</td>
<td>- Matching pursuit [92]</td>
</tr>
<tr>
<td></td>
<td>- Orthogonal matching pursuit [93]</td>
</tr>
<tr>
<td></td>
<td>- Iterative hard thresholding [94]</td>
</tr>
<tr>
<td></td>
<td>- Subspace pursuit [95]</td>
</tr>
<tr>
<td></td>
<td>- Compressive sampling matching pursuit [96]</td>
</tr>
<tr>
<td>Convex optimization</td>
<td>- Basis pursuit [97]</td>
</tr>
<tr>
<td></td>
<td>- Basis pursuit denoising [97]</td>
</tr>
<tr>
<td></td>
<td>- Least absolute shrinkage and selection operator [98]</td>
</tr>
<tr>
<td></td>
<td>- The Dantzig selector [99]</td>
</tr>
<tr>
<td></td>
<td>- Total variation denoising [100]</td>
</tr>
<tr>
<td>Weighted-norm minimization</td>
<td>- Focal underdetermined system solver [101]</td>
</tr>
<tr>
<td></td>
<td>- Iterative reweighted (non-convex) $\ell_p$-minimization $(0 &lt; p &lt; 1)$ [102]</td>
</tr>
<tr>
<td></td>
<td>- Iterative reweighted $\ell_1$-minimization [103]</td>
</tr>
<tr>
<td></td>
<td>- Iteratively reweighted least squares [104]</td>
</tr>
<tr>
<td></td>
<td>- Homotopy based reweighted $\ell_1$-minimization [105]</td>
</tr>
<tr>
<td>Bayesian methods</td>
<td>- Sparse Bayesian learning [106, 107]</td>
</tr>
<tr>
<td></td>
<td>- Bayesian CS [108]</td>
</tr>
<tr>
<td></td>
<td>- Fast Bayesian matching pursuit [109]</td>
</tr>
<tr>
<td></td>
<td>- MMSE estimation for CS [110–112] and [113, Sect. 11.5]</td>
</tr>
<tr>
<td>Message-passing algorithms</td>
<td>- Approximate message-passing [114]</td>
</tr>
<tr>
<td></td>
<td>- Bayesian CS via belief propagation [115]</td>
</tr>
<tr>
<td></td>
<td>- Generalized approximate message-passing [116]</td>
</tr>
<tr>
<td>CS with prior information</td>
<td>- Kalman filtered CS [117]</td>
</tr>
<tr>
<td></td>
<td>- Least squares CS residual [118]</td>
</tr>
<tr>
<td></td>
<td>- Modified CS / regularized modified CS [119]</td>
</tr>
<tr>
<td></td>
<td>- Regularized modified basis pursuit denoising [120]</td>
</tr>
<tr>
<td></td>
<td>- $\ell_1$, $\ell_2$-minimization and $\ell_1$, $\ell_2$-minimization [121, 122]</td>
</tr>
</tbody>
</table>

Instead of aiming at reconstructing a sparse signal, CS algorithms have also been proposed for sparse support recovery; see, e.g., [89–91].

**Exploiting signal correlation via CS**

CS is an efficient means to reduce energy consumption of gathering sparse correlated signals in WSNs [68, 78, 123]. Particular attention is needed regarding the structure of a measurement matrix $\Phi$; besides affecting the reconstruction performance, it dictates the measurement collection structure for $y = \Phi x$, 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 [124]. Luo et al. [125] 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 CS measurements along multi-hop routing. The minimum spanning tree based routing of the CS measurements was addressed in [126]. 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 [58, 127]. Sub-sampling via sleeping modes in distributed local signal recovery was proposed in [128], and sub-sampling combined with weighing each sensor’s transmissions based on the harvested energy in [129].

Temporal correlation: If the \( N \) readings in \( x \) possess temporal (i.e., intra-signal) correlations, each sensor may communicate only \( M < N \) measurements \( y = \Phi x \) to the sink for CS decoding [123]. 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 [130].

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 [57, 131] – 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) [84] which exploits general correlation structures by combining the (possibly distinct) sparsifying bases of each signal dimension into a single basis matrix [84, 132]. 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 [133].

Dynamic CS

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 [117–120, 134–137]. 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.

**Modified CS reconstruction:** The works in [117–120] 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/signal value estimates from the previous decoding instant to improve the reconstruction performance. The works [105, 134, 135, 138] 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 [119, 136]. For other applications on recursive CS, see, e.g., [139–141].

**Weighted-norm minimization:** The intrinsic prior information in dynamic CS can also be utilized via the iterative reweighted $\ell_1$-minimization (IRW-$\ell_1$) [103, 105, 134, 142–145]. The core idea of the IRW-$\ell_1$ is to suppress the magnitude dependency of the $\ell_1$-norm so that the reweighted $\ell_1$-minimization becomes closer to the sparsity-optimal $\ell_0$-minimization [103]. The IRW-$\ell_1$ alternates between solving a weighted $\ell_1$-minimization problem and updating the weights based on the current solution [103]. While the IRW-$\ell_1$ has mainly been applied for single finite-length signals, the work in [134] 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 $\ell_1$-minimization include iteratively reweighted least squares [104, 143, 146], and non-convex $\ell_p$-minimization for $0 < p < 1$ [102, 147].

The proposed approach of the thesis

**Chapter 3** is based on the published journal article [148] © 2015 IEEE, and the initial works presented in two conference papers [149] © 2013 IEEE, and [150] © 2014 IEEE. The rest of the thesis includes CS as an integral part of each sensor. Chapter 3 addresses distributed sequential CS acquisition of spatially and temporally correlated sensor data streams. A novel CS data gathering algorithm is developed which significantly reduces sensors’ communications via 1) sub-sampling based measurement collection, 2) sliding
window processing, 3) Kronecker sparsifying bases, 4) IRW-$\ell_1$, and 5) $\ell_2$-regularization. The method recovers the sensors’ readings without an excessive decoding delay, refines the estimates of past readings via recursive decoding, and makes trade-offs between the recovery performance and decoding complexity by adjusting the window size. Numerical experiments show that the method achieves higher reconstruction accuracy with a smaller number of transmissions and lower decoding delay and complexity as compared to the state-of-the-art CS methods.

1.3.3 Compressed sensing based source compression in WSNs

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point," [151, p. 1].

A foundation for information theory, and especially for source coding, is the seminal work by Shannon in 1948 [151]. Compression removes redundancy in data to minimize the average number of bits to represent a source. Thus, source compression is a prevalently adopted means to reduce energy consumption for wireless communications [152]. Since perfect representation of a real-valued source requires an infinite number of bits, digital compression of natural continuous-valued/analog signals requires quantization [153], i.e., the compression becomes lossy [154]. Moreover, due to the decentralized structure of a WSN, there is in particular a call for distributed source coding (DSC) [152, 155, 156].

In this thesis, Chapters 4 and 5 address source compression. While CS performs sub-Nyquist sampling by dimensionality reducing projections, it does not accomplish a compression task in an information-theoretic sense; the CS measurements must be converted into bits prior to data communications/storage at a digital sensor. The main focus in Chapters 4 and 5 is to investigate CS under quantized measurements. Chapter 4 addresses lossy DSC by devising a novel QCS method for acquiring correlated sparse sources. Chapter 5 focuses on the RD performance of QCS from an information-theoretic perspective. The information-theoretic tools applied in the thesis along with a literature review on QCS are presented in the following subsections.

Lossy source coding

The foundation for lossy source coding is the RD theory [157, Ch. 2] and [158, Ch. 10], i.e., source coding with a fidelity criterion [159, Ch. 9]. 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 RD function (RDF) of a source \[160\]. However, the limit is achievable only via excessively complex encoder-decoder pairs with infinitely long block lengths [157, Sect. 3.3] and [161]. Thus, the RD theory is primarily applicable to performance analysis and benchmarking of practical lossy coding methods.

A practical counterpart of the aforementioned theoretically optimal encoder-decoder pair is a symbol-by-symbol quantizer (or simply a quantizer) that maps a single observation sample into a digital output at a time. In particular, vector quantization (VQ) \[153, 162\] is capable of achieving performance arbitrary close to the theoretical optimum \[151, 160, 163\]. The inevitable loss from a finite VQ dimension can be compensated for by using variable-length coding \[164, 165\] and \[158, Ch. 5\], with the main idea 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 \[166\], 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., \[167, 168\]. In practice, VQs/ECVQs can be optimized by the Linde-Buzo-Gray (LBG) algorithm \[169\], descending from the iterative Lloyd algorithm \[170, 171\].

Remote source coding

In WSNs, the encoder relies only on indirect observations of the source, for example, due to a dedicated measurement device (e.g., CS), or noise-corrupted measurements. Such a compression variant is referred to as remote source coding \[172\], \[157, Sect. 3.5, 4.5.4\] and \[173–176\], first introduced in the seminal work by Dobrushin and Tsybakov \[172\]. The compression limit for this class is given by the remote RDF. Other frameworks on remote/noisy/indirect compression in (non-CS) setups include \[177–179\]. VQ design for compressing a remote source was addressed in \[180\].

\[2\]The letter “Q” in acronyms VQ, SQ, and ECVQ interchangeably refers to “quantization” and “quantizer” in the thesis.

\[3\]As mentioned in the “Author's Note 1981” section of \[170\], nearly all Lloyd’s results were already presented in an unpublished Bell Labs manuscript in 1957.
**Distributed source coding**

The above considerations deal with point-to-point scenarios. However, a typical WSN setup necessitates DSC [152] in which there are multiple sensors which separately measure, encode, and communicate one or multiple (correlated) sources to a decoder. DSC originates from the landmark work by Slepian and Wolf in 1973 [181], commonly referred to as Slepian-Wolf (SW) coding. According to the fundamental result in [181], SW coding improves the compression performance by exploiting signal correlation without inter-sensor collaboration, provided that joint decoding is performed. SW coding of discrete sources in [181] was extended to two correlated Gaussian sources in [182]. A distributed quantizer design for compressing correlated sources was proposed in [183–185], which incorporates entropy coding via using rate measures similarly as in [166].

**Distributed joint estimation and compression**

By applying DSC in a multi-sensor setup that contains remote sources, the source coding problem extends to distributed joint estimation and compression [186]. Distributed compression of a single remote source was addressed in [176, 187], the chief executive officer problem in [188], and DSC of multiple remote Gaussian sources in [189]. Confined to symbol-by-symbol quantizers, distributed estimation of a quantized remote source was addressed in [176, 190, 191]. Its extension to multiple correlated remote sources was proposed in [192].

**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 only for few well-behaved source/observation distributions; see, e.g., the works in [172, 174], [157, Sect. 4.5.4] and [177–179], assuming Gaussian distributions, and [193] 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 [194] and Arimoto [195]. This algorithm has been adapted to remote sensing scenarios under joint compression and classification in [196], and the chief executive officer problem in [197].
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 [198]; similarly, an approximation of the capacity of a continuous channel was studied in [199]. Other computation methods include a mapping based method akin to deterministic annealing in [200], and Lagrange duality based convex optimization in [201].

**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 [202, 203]. The case with correlated, but not necessarily identical SI at the encoder and decoder was addressed in, e.g., [204–206]. Other (non-CS) SI-aided compression variants can be found in, e.g., [157, Sect. 6.1], [207–209], [210, Sect. 11.1] and [211, Sect. 5.6].

**Quantized compressed sensing**

While the early era of CS exclusively addressed real-valued signals, the inevitable conversion of CS measurements into bit sequences initiated QCS [212–215]. A QCS setup falls into remote source coding because the encoder inputs of a CS based sensor are (noisy) linear measurements (with dimensionality reduction) of an information source (see (1)). Existing QCS works can be divided into 1) the development of quantization-aware CS encoding/decoding methods, 2) performance analysis of QCS under symbol-by-symbol quantizers, and 3) information-theoretic QCS analysis, termed *lossy CS* henceforth. These are elaborated next.

**QCS algorithms:** 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 [213, 216, 217] 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 [214, 215, 218–220]. 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 [22, Sect. 3.2.3] and [218].

As a countermeasure, joint optimization of VQ based encoder-decoder pair(s) to minimize signal reconstruction MSE in QCS setups was first proposed in [221–224]. Shirazinia et al. [222] 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 [223, 224]. 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 [110–112] – impeding the practical implementation of the methods.

**QCS performance analysis:** Several authors have analyzed operational RD performance of QCS systems where the encoder is a symbol-by-symbol quantizer. As customary approaches, they assume perfect knowledge of the sparse signal support (i.e., genie-aided/oracle-assisted signal recovery) and restrict to asymptotic results in terms of rate (i.e., high-rate quantization) or/and signal dimensions (i.e., large system limit). Oracle-assisted compression lower bounds for various high-rate QCS setups were derived in [218, 222, 225–227] and [228, Ch. 3]. Approaches with SQ based encoders include designing high-rate SQs under the least absolute shrinkage and selection operator reconstruction in [213], optimizing SQs for oversampled signals in [220], and discovering an intrinsic trade-off between the number of measurements and quantization bits in an SQ based QCS setup in [229].

**Lossy CS:** Besides being confined to symbol-by-symbol quantizers, a finite-rate CS acquisition setup can also be analyzed from a thorough information-theoretic perspective. Such treatment is incomplete, and, thus, the remote RDF for lossy CS is unknown. A problem overview along with initial results was first given in [212]. Kipnis et al. [230] 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. [227] 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 [231–233], and the compression of (sparse) Bernoulli generalized Gaussian sources via uniform SQ was studied in [234].

The proposed approaches of the thesis

Chapter 4 is based on the published journal article [235] © 2018 IEEE, and the initial works presented in two conference papers [236] © 2015 IEEE, and [237] © 2016 IEEE. While Chapters 2 and 3 assume continuous-valued signals, the remainder of the thesis focuses on digital compression/communications by incorporating a quantizer as an integral part of each sensor. Chapter 4 addresses complexity-constrained, entropy coding based lossy DSC for acquiring correlated sparse sources via CS based sensors. A novel distributed DR optimized QCS method is proposed. As a focal operation, each sensor pre-quantizes the continuous encoder inputs via a VQ, which enables a trade-off between compression performance and encoding complexity. The proposed method is demonstrated to achieve superior compression as compared to baseline QCS methods and lend itself to versatile setups with different performance requirements.

Chapter 5 is based on the accepted journal article [238] © 2018 IEEE, and the initial work presented in the conference paper [239] © 2016 IEEE. While Chapter 4 takes an algorithmic approach under complexity constraints, Chapter 5 adheres to a rigorous information-theoretic framework and investigates the fundamental RD performance limits of lossy CS of a sparse source. In this regard, the derived results establish compression limits for the single-sensor QCS variant of Chapter 4. A lossy CS problem is formulated using remote source coding. An analytically tractable lower bound to the remote RDF is derived by providing support SI to the encoder and decoder. A modified BA algorithm is developed for numerical approximation of the remote RDF. A novel entropy coding based QCS method is proposed, which is numerically shown to approach the remote RDF. Numerical results illustrate the main RD characteristics of the lossy CS, and compare the performance of practical QCS methods against the proposed limits.
1.4 Contributions of the thesis

1.4.1 Key contributions and related works

The main novelty in Chapter 2 resides in applying consensus ADMM to a cross-layer framework in WSNs. The works [43, 44] apply ADMM in a network but do not involve cross-layer optimization. The works [31–33] which address cross-layer designs are based on DD, and, thus, suffer from slow convergence and face a difficulty in handling non-strict convexity of an objective function. Moreover, as the presented ADMM algorithm is tailored to WSNs, the implementation aspects, especially in terms of requisite inter-sensor message exchanges are specifically addressed.

The work in Chapter 3 is mostly related to [84, 105, 119, 120, 135], yet with several novelties. The work [84] applies Kronecker sparsifying bases to static signals instead of streaming signals considered herein. Differently from improving the reconstruction accuracy, the works [105, 135] focus on accelerating the recovery process. Lapped orthogonal transforms with a linear dynamic signal model are assumed in [105], and slowly changing sparsity patterns in [119, 120]. As a difference, the proposed recursive decoding in this thesis relies on the general, piecewise smooth evolution of sensor readings. Moreover, differently from [119, 120], the presented sliding window decoding 1) imposes the $\ell_1$-norm penalty within the signal support (i.e., not outside the support), 2) utilizes multiple consecutive sampling instants as prior information, 3) enables progressive estimate refinement, and 4) incorporates partial support knowledge via the IRW-$\ell_1$ similarly to [105]. Finally, instead of the general dynamic sparse signal or image processing frameworks, the proposed method is tailored to WSNs by considering 1) the sensors’ communication costs, 2) the encoding limitations set by the distributed sensing setup, and 3) multi-dimensional compressible signals.

The approach presented in Chapter 4 is novel in that DR optimization in a distributed QCS setup has not been addressed earlier. The CS model is based on the distributed CS in [57, 131]. The quantizer design is mostly related to the works [223, 224] which, differently from this thesis, 1) focus on channel-optimized VQs, 2) use fixed-rate VQs, and 3) involve complex encoders impeding practical implementation. The entropy coding part utilizes the original ideas of [166] and those extended to multiple remote sources in, e.g., [192]. Pre-quantization as a means to moderate encoding complexity, and the use of variable-rate coding in a QCS setup are
also novel. A philosophy similar to pre-quantization appears in approaches relying on multiple prototypes in, e.g., [240, 241].

The main novelty in Chapter 5 is to study lossy CS comprehensively from the information-theoretic, remote source coding perspective. Different from the QCS analyses in [218, 222, 225, 226], the coding system is not limited to a symbol-by-symbol quantizer. The work [212] gives an overview of the problem. Coluccia et al. [227] derived a DR lower bound under support SI at the decoder assuming 1) high-rate quantization, 2) a large system regime as $N \to \infty$, and 3) noiseless measurements. In this thesis, the noisy CS of a vector source encompassing a composite structure is considered, and the derived results hold for any rate. In a parallel work [230], Kipnis et al. derived a general expression for the remote RDF of lossy CS by using a replica method for a Bernoulli-Gauss (scalar) source under a large system limit $M, N \to \infty$. While a closed-form expression for the remote RDF was not found, they provided curves via the BA algorithm. Whereas most previous BA approaches, including [230], treat the discretization implicitly via "fine enough discretization" or SQ, an optimized VQ based discretization method is proposed in this thesis; the VQ design takes the remote nature of the lossy CS into account via modified distortion measures, and that yields accurate approximations of the remote RDF.

1.4.2 Author’s contribution to the publications

This thesis is written as a monograph based on the following twelve original publications: three published journal articles [51, 148, 235], one accepted journal article [238], and eight conference papers [52–54, 149, 150, 236, 237, 239]. The author of this thesis had the main responsibility in finding the original ideas of the developed methods, formulating the mathematical problems and deriving their analytical solutions, compiling the MATLAB codes to run empirical performance analysis simulations, and writing the original international peer-reviewed publications. Other co-authors provided valuable guidance for research directions, ideas on developed methods/algorithms, and comments during a preparation of each manuscript to improve their quality.
2 Distributed joint resource and routing optimization

This chapter addresses distributed sum transmit power minimization via the joint optimization of resource allocation and routing for given source rates in a multi-hop single-sink data gathering (SSDG) wireless sensor network (WSN). The presented cross-layer optimization framework takes the inherent coupling of the optimal routing, power, and bandwidth allocation into account, which considerably increases the energy efficiency of the WSN. A novel distributed optimization algorithm based on a consensus alternating direction method of multipliers (ADMM) is derived. Furthermore, a sophisticated variant of dual decomposition (DD) is developed. Numerical experiments show that the proposed ADMM algorithm has superior convergence performance compared to the DD based algorithms. The developed algorithm is appealing for practical implementation owing to its low local communication overhead, robust convergence under varying channel conditions and step size parameter values, and its scalability to large networks. The results also show that significant system performance gains are obtained via the bandwidth allocation scheme developed.

The system model for the SSDG WSN is defined in Section 2.1. The optimization problem is stated in Section 2.2, followed by the derivation of the ADMM and DD algorithms. Numerical results are presented in Section 2.3, and Section 2.4 summarizes the chapter.

2.1 Data gathering wireless sensor network

In this section, the network topology, multi-path routing model, and communication model for the SSDG WSN are defined. A generic single-commodity flow model with periodic data generation at each node is assumed.

2.1.1 Network topology

Consider a SSDG WSN consisting of a set of $N$ sensors $\mathcal{N} = \{1,\ldots,N\}$, acting as source nodes, and one sink node, denoted by $N+1$. The WSN is modeled as the directed graph $G = (\mathcal{N},\mathcal{L})$, where the vertices $\mathcal{N} = \{1,\ldots,N,N+1\}$ determine the
Fig. 3. A single-sink data gathering wireless sensor network with $N = 7$ and $L = 26$ (© 2013 IEEE).

set of $N + 1$ nodes, where $\mathcal{N} = \mathcal{N} \cup \{N + 1\}$, and the edges $\mathcal{L} = \{1, \ldots, L\}$ represent the set of $L$ directed wireless links between the nodes. A link from node $i \in \mathcal{N}$ to $j \in \mathcal{N}$, $i \neq j$, is interchangeably expressed as $l \in \mathcal{L}$ and the ordered pair $(i, j)$. A SSDG WSN with $N = 7$ and $L = 26$ (12 bidirectional and 2 unidirectional links) is illustrated in Fig. 3.

The network topology, i.e., the interactions between the nodes and links, can be described with a node-link incidence matrix $\tilde{A} \in \mathbb{Z}^{(N + 1) \times L}$. An entry in its $i$th row and $l$th column, associated with node $i \in \mathcal{N}$ and link $l \in \mathcal{L}$, is given by [31]

$$a_{il} = \begin{cases} 
1, & \text{if } i = \text{tran}(l), i \in \mathcal{N}, l \in \mathcal{L} \\
-1, & \text{if } i = \text{rec}(l), i \in \mathcal{N}, l \in \mathcal{L} \\
0, & \text{otherwise} 
\end{cases} \quad (4)$$
where tran($l$) and rec($l$) denote the transmitting and receiving node of link $l \in \mathcal{L}$, respectively. The generation of matrix $\hat{A}$ is elaborated in detail in Section 2.1.3.

The WSN is assumed to be connected, i.e., each sensor $i \in \mathcal{N}$ has a path to the sink. Once generated, the topology remains fixed; dynamic features such as node mobility and abrupt node failures, which would cause alterations in the network connections, are beyond the scope of the thesis.

### 2.1.2 Multi-path routing model

The generation of application data at the sensors is assumed to follow a continuous data delivery model [242] in which each sensor periodically observes a phenomenon of interest and communicates the information continuously at a constant predefined rate. The routing of data packets is assumed to follow a braided multi-path routing model, i.e., each sensor’s data is routed via alternate, partially disjoint paths [243]. Compared to single-path routing, multi-path routing balances the traffic load, mitigates network congestion, and provides significant improvements in the network utilization [244]. Transmissions of packets across the links are assumed to be lossless. Although stochastic modeling of queueing and packet losses are neglected with this tractable and consistent routing model, it adequately encompasses the average behavior of data transmissions in the WSN.

Formally, let $f = [f_1 \cdots f_L]^T$ denote the data flow vector for the WSN, where $f_l \geq 0$ is the amount of flow on link $l \in \mathcal{L}$. Let $\hat{r} = [r_1 \cdots r_N r_{N+1}]^T$ denote the rate vector, where $r_i > 0$, $i \in \mathcal{N}$, is a fixed source rate of sensor $i$, and $r_{N+1} < 0$ is the sink rate. According to the lossless transmissions, the flow conservation law (FCL) must be satisfied at each node $i \in \mathcal{N}$, i.e., [244]

$$\sum_{l \in \mathcal{O}(i)} f_l - \sum_{l \in \mathcal{I}(i)} f_l = r_i, \quad i \in \mathcal{N},$$

(5)

where $\mathcal{O}(i) \subseteq \mathcal{L}$ denotes the set of outgoing links of node $i \in \mathcal{N}$, and $\mathcal{I}(i) \subseteq \mathcal{L}$ denotes the set of incoming links of node $i \in \mathcal{N}$. Hence, the FCL imposes a quality of service requirement since all source data has to be delivered to the sink node, i.e., $r_{N+1} = -\sum_{i \in \mathcal{N}} r_i$.

The FCL equations in (5) are redundant, i.e., linearly dependent since $1^T \hat{A} = 0$ [9, p. 551]. Thus, by the procedure similar to [9, p. 551], the last row of $\hat{A}$ and the last entry of $\hat{r}$ are removed, resulting in the reduced node-link incidence matrix $A \in \mathbb{Z}^{N \times L}$, and
the reduced rate vector $r \in \mathbb{R}^{N}_{++}$. Consequently, the FCL for the WSN can be compactly expressed as [31]

$$Af = r. \quad (6)$$

By virtue of the deletion, the sink is no longer (explicitly) associated with the FCL constraints in (6). As will be seen in Sections 2.2.2 and 2.2.3, this relieves the sink from all the computation and local variable exchanges in the proposed distributed optimization algorithms.

### 2.1.3 Communication model

The sensors are assumed to communicate utilizing frequency division multiple access (FDMA), allowing simultaneously occurring, non-interfering communications at the links. Each sensor – capable of transmitting, receiving, and relaying data – is equipped with single-antenna transceivers and assumed to have perfect channel state information on the associated communication links. Modeling of coding, modulation, relaying schemes, and packet transmission protocols etc. are beyond the scope of this thesis.

In terms of a physical layer design, each sensor can optimize the transmissions via local resource allocation. Namely, each sensor $i \in \mathcal{N}$ can allocate transmit power $p_l \geq 0$ and a continuous frequency band of bandwidth $w_l \geq 0$ to each of its outgoing link $l \in \Theta(i)$. Each sensor is limited by the total power $P$ (assumed to be the same for all nodes) and total bandwidth $W_i$ as

$$\sum_{l \in \Theta(i)} p_l \leq P, \sum_{l \in \Theta(i)} w_l \leq W_i, \forall i \in \mathcal{N}. \quad (7)$$

According to the FDMA, the continuous frequency bands $W_i, i \in \mathcal{N}$, are pre-allocated to the sensors in a non-overlapping fashion, guaranteeing orthogonal transmissions in the WSN$^4$.

The existence of available wireless links in the WSN – captured by $\tilde{A}$ in (4) – is established as follows. Let $d_{ij}$ denote the distance between nodes $i \in \tilde{\mathcal{N}}$ and $j \in \tilde{\mathcal{N}}$, $i \neq j$. A wireless link $(i, j)$ exists if $d_{ij} \leq d^{tx}$, where $d^{tx}$ is the maximum transmission range of each sensor $i \in \mathcal{N}$, dominantly determined by the available total transmit

---

$^4$In general, the sensors must be capable of receiving data over the entire system bandwidth. A link can be established by informing the associated receiver of the allocated frequency.

54
Due to the equal communication range, the wireless links between the sensors are bidirectional, i.e., \((i,j) \iff (j,i), i,j \in \mathcal{N}, i \neq j\).

Communication channels are assumed to be Rayleigh flat fading, i.e., each channel can be represented by a single complex filter tap whose magnitude is Rayleigh distributed [245, Ch. 2]. Assuming the inverse-square path loss model [246, Sect. 2.6] and [31], the capacity of link \(l \in \mathcal{L}\) as a function of the transmit power \(p_l\) and bandwidth \(w_l\) is given as

\[
c_l(p_l, w_l) = w_l \log \left( 1 + \frac{d_0}{d_l} \left( \frac{|\kappa_l|^2 p_l}{N_0 w_l} \right) \right), \quad l \in \mathcal{L},
\]

where \(d_0\) is a reference distance so that \(d_0 \leq d_l, \forall l \in \mathcal{L}\), \(d_l\) is the link length (cf. \(d_{ij}\)), \(N_0\) is the power spectral density of additive white Gaussian noise present at the receiver\(^6\) (assumed to be the same for all nodes), and channel coefficient \(\kappa_l \sim \mathcal{CN}(0, \sigma^2_{\kappa_l})\) is a circularly symmetric complex Gaussian random variable with variance \(\sigma^2_{\kappa_l}\) (i.e., \(|\kappa_l|\) is Rayleigh distributed). For the sake of brevity, the channel condition parameters are summarized as

\[
\gamma_l \triangleq \left( \frac{d_0}{d_l} \right)^2 \frac{|\kappa_l|^2}{N_0}, \quad l \in \mathcal{L},
\]

and, thus, the capacity in (8) can be written as

\[
c_l(p_l, w_l) = w_l \log \left( 1 + \frac{\gamma_l p_l}{w_l} \right), \quad l \in \mathcal{L}.
\]

Successful data transmissions across the wireless links require that the total flow on each link \(l \in \mathcal{L}\) does not exceed the link capacity, i.e.,

\[
f_l \leq c_l(p_l, w_l), \quad l \in \mathcal{L}.
\]

2.2 Joint resource and routing optimization

In this section, a total transmit power minimization problem relying on joint resource allocation and routing optimization is stated. Due to the decentralized, autonomous operation of a WSN, the main objective is to find the solution in a distributed manner.

---

\(^5\)More precisely, the transmission range is determined by the received signal-to-noise ratio and receiver sensitivity. Nevertheless, due to the equal total power \(P\) and time-averaging the effect of small-scale fading in (8), the transmission range of each sensor is roughly the same.

\(^6\)In Fig. 3, the noise signal at the receiver of link \((i,j)\) is denoted by \(\varsigma_{ij}\).
To this end, the original centralized problem is split into smaller subproblems, which are locally solved by the sensors via a small amount of variable exchange. Hence, the solution is found without gathering all network data to a single location (e.g., the sink) which would cause large energy expenditure of the sensors, congestion of the network traffic etc. As another benefit of the distributed design, the routing can be adapted on the fly to alterations in network topology and channel conditions etc.

2.2.1 Problem formulation

The objective is to find a feasible transmit power and bandwidth allocation, and data routing for the wireless links so that all sensor data is delivered to the sink with a minimum sum transmit power usage. Accordingly, the total transmit power minimization can be expressed as a joint optimization problem as

\[
\begin{align*}
\text{minimize} & \quad \sum_{l \in \mathcal{L}} p_l \\
\text{subject to} & \quad Af = r \\
& \quad f_l \leq w_l \log \left( 1 + \frac{p_l}{w_l} \right), \quad l \in \mathcal{L} \\
& \quad \sum_{i \in \mathcal{N}(i)} p_l \leq P, \quad i \in \mathcal{N} \\
& \quad \sum_{i \in \mathcal{N}(i)} w_l \leq W, \quad i \in \mathcal{N} \\
& \quad p_l \geq 0, \quad w_l \geq 0, \quad f_l \geq 0, \quad l \in \mathcal{L},
\end{align*}
\]

where the optimization variables are the link powers \( p_l \), bandwidths \( w_l \), and flows \( f_l \), \( l \in \mathcal{L} \). The first constraint represents the FCL at each node \( i \in \mathcal{N} \) (see (6)), the second constraint set ensures the flows do not exceed the link capacities (see (11)), the third and fourth inequalities limit the maximum power and bandwidth at each sensor \( i \in \mathcal{N} \), respectively (see (7)), and the last inequalities constrain the variables so they are non-negative. It is worth mentioning that since the source rates are fixed, the total transmit power minimization also maximizes the energy efficiency (bits/Joule) in the WSN [247]. A problem similar to (12), albeit with a different link capacity model, has been addressed in [33].

The problem (12) is convex: the objective function is a non-negative weighted sum of convex functions, the inequality constraints are convex, and the equality constraints are affine [9, Sect. 4.2.1]. In particular, the link capacities in (8) are concave and monotone increasing in variables \( p_l \) and \( w_l \) [31], and, thus, each constraint

56
\( f_l - c_l(p_l, w_l) \leq 0, \quad l \in \mathcal{L}, \) is convex. Owing to the convexity, a primal optimal (centralized) solution can be efficiently found via, e.g., interior-point methods [9, Ch. 11].

Due to the FCL constraint \( A f = r, \) the problem is not separable with respect to the nodes. Namely, a slight variation of the flow on one link affects the entire global routing solution. More precisely, the flow on each link \((i, j)\) has two perspectives: it stands for the outgoing flow of node \( i, \) and the incoming flow of node \( j. \) In order to arrive at a decentralized algorithm, this coupling has to be appropriately handled. Utilizing two different decoupling techniques, distributed optimization algorithms based on the consensus ADMM and DD are derived in Section 2.2.2 and Section 2.2.3, respectively.

### 2.2.2 Distributed optimization via consensus ADMM

The centralized problem (12) is approached by consensus optimization in conjunction with the ADMM. Roughly speaking, each global flow variable is duplicated into local flow copies that are presented to the end nodes of the link. Each such local variable can be interpreted as a sensor’s opinion about the corresponding global flow variable. The purpose of the ADMM is to drive these local copies into agreement, i.e., to achieve consensus, in a distributed fashion [40]. The proposed algorithm is novel in that the distributed consensus ADMM technique has not been previously applied for cross-layer optimization in a multi-hop WSN setup.

**Consensus problem**

Let \( \mathcal{L}(i) \subset \mathcal{L} \) denote the set of links connected to sensor \( i \in \mathcal{N}, \) i.e., \( \mathcal{L}(i) = \mathcal{I}(i) \cup \mathcal{O}(i). \) Each sensor \( i \in \mathcal{N} \) is associated with a set of local copies of the global flow variables \( f_l \) of links \( l \in \mathcal{L}(i), \) denoted as \( \hat{f}_{il}. \) In consequence, each sensor \( i \in \mathcal{N} \) is associated with a set of local optimization variables\(^7\): \( \hat{f}_{il}, l \in \mathcal{L}(i), \) and \( p_l, w_l, l \in \mathcal{O}(i). \) The construction of this variable set is illustrated in Fig. 4. Note that due to the reduced node-link incidence matrix \( A \) formed in Section 2.1.2, local flows are not needed for the sink.

\(^7\)The resource variables \( p_l \) and \( w_l \) of links \( l \in \mathcal{O}(i) \) are inherently local variables for the transmitting node \( \text{tran}(l). \)
Regarding the local flows as auxiliary variables, the problem in (12) can be reformulated as a consensus problem as

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in \mathcal{N}} \sum_{l \in \mathcal{O}(i)} p_l \\
\text{subject to} & \quad \hat{f}_{il} = f_l, \quad i \in \mathcal{N}, \ l \in \mathcal{L}(i) \\
& \quad \sum_{l \in \mathcal{O}(i)} a_{il} \hat{f}_{il} = r_i, \ i \in \mathcal{N} \\
& \quad \hat{f}_{il} \leq w_l \log \left(1 + \frac{\gamma l p_l}{w_l}\right), \ i \in \mathcal{N}, \ l \in \mathcal{O}(i) \\
& \quad \sum_{l \in \mathcal{O}(i)} p_l \leq P, \ i \in \mathcal{N} \\
& \quad \sum_{l \in \mathcal{O}(i)} w_l \leq W, \ i \in \mathcal{N} \\
& \quad p_l \geq 0, \ w_l \geq 0, \ l \in \mathcal{L} \\
& \quad \hat{f}_{il} \geq 0, \ i \in \mathcal{N}, \ l \in \mathcal{L}(i),
\end{align*}
\]

with the optimization variables \(p_l, w_l, f_l, l \in \mathcal{L}, \) and \(\hat{f}_{il}, i \in \mathcal{N}, l \in \mathcal{L}(i)\). The first set of equality constraints represents the consensus constraints, i.e., they enforce the local flow copies so they are in agreement with the corresponding global ones. All other constraints constitute local constraint sets for each node \(i \in \mathcal{N}\). Thus, due to the consensus constraints, the problem (13) still necessitates a centralized solver.

For the sake of brevity, let \(p_i \in \mathbb{R}^{\mathcal{O}(i)}, w_i \in \mathbb{R}^{\mathcal{O}(i)}, \) and \(\hat{f}_i \in \mathbb{R}^{\mathcal{L}(i)}\) be the vectors comprising of the local power, bandwidth, and flow variables \(p_l, w_l, l \in \mathcal{O}(i), \) and \(\hat{f}_{il}, l \in \mathcal{L}(i),\) respectively, at sensor \(i \in \mathcal{N}.\) Accordingly, a feasible set of local variables
for each sensor $i \in \mathcal{N}$ is defined as

$$
\mathcal{F}_i = \left\{ \begin{array}{l}
\sum_{l \in \mathcal{O}(i)} p_l = r_i \\
\hat{f}_l \leq w_l \log \left( 1 + \frac{g_l \hat{f}_l}{w_l} \right), l \in \mathcal{O}(i) \\
\sum_{l \in \mathcal{O}(i)} p_l \leq P \\
\sum_{l \in \mathcal{O}(i)} w_l \leq W_i \\
p_i \geq 0, w_i \geq 0, \hat{f}_i \geq 0 
\end{array} \right\}, i \in \mathcal{N}. \quad (14)
$$

Consequently, a local cost function is defined as

$$
g_i(p_i, w_i, \hat{f}_i) = \left\{ \begin{array}{l}
\sum_{l \in \mathcal{O}(i)} p_l \in \mathcal{F}_i \\
+\infty \text{ otherwise.}
\end{array} \right\} \quad (15)
$$

Using the above notations, the consensus problem in (13) can be compactly written as

$$
\begin{array}{l}
\text{minimize} \\
\sum_{i \in \mathcal{N}} g_i(p_i, w_i, \hat{f}_i)
\end{array} \quad (16)
$$

subject to $\hat{f}_l = f_i, i \in \mathcal{N}, l \in \mathcal{L}(i)$,

with the local optimization variables $p_i, w_i, \hat{f}_i, i \in \mathcal{N}$, and global optimization variables $f = [f_1 \cdots f_L]^T$. Note that although the objective function in (16) is separable, the consensus constraints – similarly to those in (13) – couple the problem across the sensors.

The consensus ADMM

Next, the ADMM is applied to solve the problem in (16) in a distributed way. The ADMM routine is initiated by forming an augmented Lagrangian in respect to the consensus constraints [40] and [10, Sect. 3.4.4]. On top of a conventional Lagrangian, the augmented Lagrangian contains an additional (thus augmented) quadratic regularization term, i.e., a squared $\ell_2$-norm term with respect to the consensus constraints [40] and [10, Sect. 3.4.4].

Formally, let $\lambda_{il}, i \in \mathcal{N}, l \in \mathcal{L}(i)$, be the Lagrange multipliers associated with the consensus constraints of (16). The augmented (partial) Lagrangian for (16) is defined
as

\[
\Sigma_{\text{admm}} \left( \{ p_i, w_i, f_i \}_{i \in \mathcal{N}}, \{ \lambda_{il} \}_{i \in \mathcal{N}, j \in \mathcal{L}(i)} \right) = \sum_{i \in \mathcal{N}} g_i(p_i, w_i, f_i) + \\
\sum_{i \in \mathcal{N}} \sum_{l \in \mathcal{L}(i)} \lambda_{il} (\hat{f}_{il} - f_i) + \frac{\rho}{2} \sum_{i \in \mathcal{N}} \sum_{l \in \mathcal{L}(i)} (\hat{f}_{il} - f_i)^2,
\]

where \( \rho > 0 \) is a positive parameter for adjusting the convergence properties of the ADMM. Fundamentally, the augmentation can be seen as a penalty term added to the primal objective function [40]. The regularization term facilitates the algorithm to reach consensus in respect to the local and global variables. It is worth emphasizing that the solution of the original optimization problem (16) is not affected by the augmentation since the term vanishes for any set of primal feasible variables.

The ADMM is based on sequential optimization in which the augmented Lagrangian is first optimized with respect to the local variables, then with respect to the global variables, and finally, the dual variables are updated via the method of multipliers [40] and [10, Sect. 3.4.4]. Thus, at each iteration \( k \), the ADMM loop for solving (16) consists of the following three steps:

\[
\begin{align*}
\{ p_{i}^{k+1}, w_{i}^{k+1}, f_{i}^{k+1} \}_{i \in \mathcal{N}} & := \arg \min_{\{ p_i, w_i, f_i \}_{i \in \mathcal{N}}} \left\{ \sum_{i \in \mathcal{N}} g_i(p_i, w_i, f_i) + \sum_{l \in \mathcal{L}(i)} \lambda_{il}^{k} (\hat{f}_{il} - f_i^k) + \frac{\rho}{2} \sum_{l \in \mathcal{L}(i)} (\hat{f}_{il} - f_i^k)^2 \right\} \\
\hat{f}_{i}^{k+1} & := \arg \min_{f_i \in \mathbb{R}} \left\{ \sum_{i \in \mathcal{N}} \sum_{l \in \mathcal{L}(i)} \lambda_{il}^{k} (\hat{f}_{il}^{k+1} - f_i) + \frac{\rho}{2} \sum_{l \in \mathcal{L}(i)} (\hat{f}_{il}^{k+1} - f_i)^2 \right\} \\
\lambda_{il}^{k+1} & := \lambda_{il}^{k} + \rho \left( \hat{f}_{il}^{k+1} - f_i^{k+1} \right), \ i \in \mathcal{N}, \ l \in \mathcal{L}(i),
\end{align*}
\]

where the superscript \( k \) denotes the iteration index. The above sequence of updates is referred to as a distributed consensus ADMM [48]. For other variants of the method, see, e.g., the work in [248] which considers asynchronous updates, and the work in [249] which considers a time-varying objective function.

In other words, the iterative optimization process is done by alternating the direction of the variables by sequentially solving parallel subproblems [40]. As the standard method of multipliers applies joint minimization of the primal variables, the ADMM can be seen as its special case by having only a single Gauss-Seidel pass over a set of variables at a time [40]. Contrary to the method of multipliers involving such a joint
optimization phase, the sequential optimization in the ADMM is exactly what enables
the decomposition of the problem across network nodes [10, Sect. 3.4.4]. Next, each
ADMM step in (18) is elaborated in detail.

Step 1) – Local variable update: The first step (18a) is completely decentralized, i.e.,
finding the optimal power, bandwidth, and local flow variables $p_{i}^{k+1}$, $w_{i}^{k+1}$, and $\hat{f}_{i}^{k+1}$
respectively, decouples into $N$ node specific subproblems. Namely, after eliminating
the constant terms which do not affect the solution, each sensor $i \in \mathcal{N}$ solves a convex
optimization problem at iteration $k$ as

$$
\text{minimize} \quad \sum_{l \in \mathcal{O}(i)} p_{l} + \sum_{l \in \mathcal{L}(i)} \left[ \frac{\rho}{2} (\hat{f}_{l} - f_{l})^2 + \lambda_{\text{tran}}^{k} \hat{f}_{l} \right]
\text{subject to} \quad (p_{i}, w_{i}, \hat{f}_{i}) \in \mathcal{F}_{i},
$$

(19)

where the variables are $p_{i}$, $w_{i}$ and $\hat{f}_{i}$, and $\mathcal{F}_{i}$ is the feasible set given in (14). Since
the problem is convex, it can be efficiently solved by, e.g., interior-point methods [9,
Ch. 11].

Step 2) – Global variable update: The second step (18b) separates into $L$
unconstrained link specific minimization subproblems. After removing the unnecessary
terms, the subproblems can be written as

$$
\hat{f}_{l}^{k+1} := \arg\min_{f_{l}} \left\{ \frac{\rho}{2} (\hat{f}_{\text{tran}(l)}^{k} - f_{l})^2 - \lambda_{\text{tran}}^{k} f_{l} + \frac{1}{\rho} \lambda_{\text{rec}}^{k} \hat{f}_{l} \right\}, \quad l \in \mathcal{L} \setminus \mathcal{I} (N + 1)
$$

(20)

$$
\hat{f}_{l}^{k+1} := \arg\min_{f_{l}} \left\{ \frac{\rho}{2} (\hat{f}_{\text{rec}(l)}^{k} - f_{l})^2 - \lambda_{\text{rec}}^{k} f_{l} \right\}, \quad l \in \mathcal{I} (N + 1),
$$

where the lower equation is associated with the incoming links of the sink node, and
the upper one with every other link in the WSN. Thanks to the quadratic regularization
term in the augmented Lagrangian (17), each subproblem (20) is strictly convex in $f_{l}$,
$l \in \mathcal{L}$. Thus, the unique optimal solution can be found by setting the derivative to zero,
resulting in

$$
\hat{f}_{l}^{k+1} := \begin{cases}
\frac{1}{2} (\hat{f}_{\text{tran}(l)}^{k+1} + f_{\text{rec}(l)}^{k+1}) + \frac{1}{2\rho} \left( \lambda_{\text{tran}}^{k} + \lambda_{\text{rec}}^{k} \right), & l \in \mathcal{L} \setminus \mathcal{I} (N + 1),
\\
\frac{1}{\rho} \lambda_{\text{rec}}^{k} f_{l}^{k}, & l \in \mathcal{I} (N + 1).
\end{cases}
$$

(21)
The following property is used to simplify the equations in (21). Initializing the dual variables with zeros results in 
\[ \lambda^{k}_{\text{tran}(l)} + \lambda^{k}_{\text{rec}(l)} = 0, \forall l \in \mathcal{L} \setminus \mathcal{I} (N + 1), \]
and 
\[ \lambda^{k}_{\text{tran}(l)} = 0, \forall l \in \mathcal{I} (N + 1), \]
at each iteration \( k = 1, 2, \ldots \) [40]. Consequently, the optimal global flow variables become

\[
 f_{l}^{k+1} = \begin{cases} 
 \frac{1}{2} \left( f_{\text{tran}(l)}^{k+1} + f_{\text{rec}(l)}^{k+1} \right), & l \in \mathcal{L} \setminus \mathcal{I} (N + 1) \\
 f_{\text{tran}(l)}^{k+1}, & l \in \mathcal{I} (N + 1). 
\end{cases}
\] (22)

Thus, the global flow variables are obtained at each iteration \( k \) by averaging out the corresponding (updated) local flows. This gives rise to the philosophy of interpreting the current local flow copies as the nodes’ opinions about the optimal global variables. Since the dual variables are no longer involved in (22), the local communication overhead is also reduced to perform the step, discussed more in Section 2.2.2.

**Step 3) – Dual variable update:** The final step of the ADMM in (18c) is the dual variable update. The use of regularization parameter \( \rho \) as the step size is motivated by the method of multipliers in which after each sequence of joint optimization of primal variables followed by the dual update the solution is dual feasible [40]. Even though this does not hold for the ADMM due to the sequential optimization of primal variables, the dual residual as well as the primal residual converge to zero when the algorithm proceeds [40].

**Penalty parameter \( \rho \)**

Naturally, the convergence properties of the ADMM are affected by parameter \( \rho \). A proper choice for \( \rho \) is not straightforward due to its equivocal meaning: besides penalizing the consensus constraint violation in the augmented Lagrangian (17), it acts as the step size for the dual variable update in (18c). As a consequence, large values of \( \rho \) tend to strive for small primal residuals, yet this occurs at the cost of an increased dual residual, and vice versa [40]. While the convergence results of the ADMM rely on a constant \( \rho \), potential extensions include varying \( \rho \) at each iteration based on the primal and dual residuals, or increasing it gradually to make it more insensitive to the initial choice [40] and [27, Sect. 4.2]. The former, however, requires centralized coordination, making it infeasible for a distributed algorithm. As will be exemplified by the numerical results in Section 2.3, the standard ADMM implementation possesses a decent level of robustness against the choice of a constant \( \rho \).
Algorithm 1 Distributed joint resource and routing optimization via consensus ADMM

a) Initializations:
   i) Set $k = 1$; ii) initialize the dual variables as $\lambda_{il}^k = 0$, $\forall i \in \mathcal{N}$, $l \in \mathcal{L}(i)$, and the global flow variables as $f_{il}^k = 0$, $\forall l \in \mathcal{L}$.

b) Iterations:
   I. Each sensor node $i = 1, 2, \ldots, N$:
      1) Finds the optimal power and bandwidth variables $p_{il}^{k+1}$, and $w_{il}^{k+1}$, $l \in \mathcal{O}(i)$, respectively, and the local flow variables $\hat{f}_{il}^{k+1}$, $l \in \mathcal{L}(i)$, by solving the problem in (19).
      2) Broadcasts the updated local flow variables $\hat{f}_{il}^{k+1}$, $l \in \mathcal{L}(i)$, to its adjacent nodes, i.e., the end nodes of links $l \in \mathcal{L}(i)$.
      3) Sets the optimal global flow variables $f_{il}^{k+1}$, $l \in \mathcal{L}(i)$, according to (22).
      4) Performs the dual variable update in (18c) to obtain $\lambda_{il}^{k+1}$, $l \in \mathcal{L}(i)$.
   II. If a predefined stopping criterion is met then stop; otherwise, set $k = k + 1$, and go to Step I.

Algorithm implementation and scalability

The proposed distributed consensus ADMM algorithm for the joint resource allocation and routing optimization in (12) is summarized in Algorithm 1. The optimization steps 1) – 4), and the corresponding required local information exchanges with respect to node 1 are depicted in Fig. 5. The horizontal decomposition of the solution process is clearly visible in Step I: the sensors perform optimization in parallel, whereas the coordination towards the global goal is regulated by interchanging only local information.

The communication overhead in one iteration loop consists of conveying $\hat{f}_{il}^{k+1}$ and $\hat{f}_{il}^{k+1}$ across each link $l \in \mathcal{L} \setminus \mathcal{S}(N + 1)$. Clearly, the amount of local information exchanged per sensor is dependent on the number of neighboring nodes, i.e., the total communication overhead at each iteration scales as $\sum_{i \in \mathcal{N}} |\mathcal{L}(i)|$. As a consequence, the scalability of the proposed algorithm is mainly dictated by the network density rather than the number of nodes. Thus, by increasing the number of nodes while keeping their density fixed in the WSN, the convergence of the ADMM is not significantly
Step 1: Obtain local flow variables, and power and bandwidth variables

Step 2: Exchange of the local flow variables

Step 3: Obtain global flow variables

Step 4: Obtain Lagrange multipliers

Fig. 5. Optimization steps 1) – 4), and the local information exchanges in respect to node $i$ within an iteration cycle of the proposed ADMM algorithm (Algorithm 1). The black dash-dotted areas encompass the sets of variables involved in the optimization steps, the red-colored variables are the ones that are updated, and the dashed arrows represent the exchanges of real-valued scalars across the links ([51] © 2013 IEEE).

slowed down. On this account, the ADMM is suitable for large-scale problems [40]. This is supported by the numerical results in Section 2.3.

Infeasible problem

The derivations so far have assumed that the original problem (12) is feasible, i.e., it has a solution. While this is true for a typical well-designed WSN, infeasibility may arise under several circumstances. For instance, this case occurs when the sensors are short of resources to route all of the source data (which is controlled by a current application) to the sink, i.e., when $P$ and $W_i$ are small, and $r_i$ is large, $i \in \mathcal{N}$. In consequence of an unreliable and harsh operation environment, infeasibility may also be caused by degraded link qualities, node failures etc. A sensor’s ability to detect infeasibility on the fly would be beneficial as the algorithm could be terminated to save the battery and the sink could be informed to make requisite adjustments to re-establish the data gathering. This is discussed in the following.
Recall that the only constrained subproblem in the ADMM algorithm is (19). Similarly to the original problem (12), (19) can be infeasible, albeit under different conditions. These conditions are compared next. For (12), the minimum amount of resources $P$ and $W_i$ for each sensor $i \in \mathcal{N}$ to support feasible data traffic is determined through the global FCL constraints (6), which intrinsically couple the outgoing flows to the incoming flows. However, owing to the local FCL constraint $\sum_{l \in \mathcal{O}(i)} f_{il} - \sum_{l \in \mathcal{I}(i)} f_{il} = r_i$ in (19), each sensor $i \in \mathcal{N}$ can independently assign $\hat{f}_{il} = 0, \forall l \in \mathcal{I}(i)$, to induce a minimal total outgoing flow $\sum_{l \in \mathcal{O}(i)} \hat{f}_{il} = r_i$, and, thus, the minimal possible resource usage via allocated $p_l$ and $w_l, l \in \mathcal{O}(i)$. This is ruled out by the global FCL in (12) unless each sensor transmits its data to the sink with a single hop. Therefore, the smallest $P$ and $W_i, i \in \mathcal{N}$, required so that (12) is feasible are always equal to or larger than those required so that each (19) is feasible. In other words, if the original problem (12) is feasible, all subproblems (19) are feasible.

According to the above reasoning, the infeasibility of (12) may arise in the ADMM algorithm in two ways. The first is straightforward: if any sensor finds that (19) is infeasible at the first iteration $k = 1$, the problem (12) is infeasible, and the sensor may initiate the algorithm termination. The second case is more intricate because all subproblems (19) may be feasible at each iteration even if the problem (12) is not. Accordingly, each sensor $i \in \mathcal{N}$ finds a feasible solution, but, due to the lack of resources, the obtained local flow variables cannot coordinate with the global ones. In this case, transmitting nodes $\text{tran}(l)$ of some links $l \in \mathcal{L} \setminus \mathcal{S}(N + 1)$ have a tendency to increase $\hat{f}_{\text{trans}(l)}$, while the corresponding receiving nodes $\text{rec}(l)$ are inclined to decrease $\hat{f}_{\text{rec}(l)}$. Consequently, the nodes will never reach consensus on the flow $f_l$, and the algorithm diverges.

Fortunately, the latter type of infeasibility can be identified from the evolution of the dual variables, updated as $\lambda_{il}^{k+1} = \lambda_{il}^k + \rho(f_{il}^{k+1} - f_{il}^k)$ in (18c). The recurrent disagreements of local flows for a link $l \in \mathcal{L} \setminus \mathcal{S}(N + 1)$ cause $\hat{f}_{\text{trans}(l)}^{k+1} - f_{il}^{k+1} > 0$ and $\hat{f}_{\text{rec}(l)}^{k+1} - f_{il}^{k+1} < 0$ at each iteration $k$. Since $f_{il}^{k+1}$ is the average of $\hat{f}_{\text{trans}(l)}^{k+1}$ and $\hat{f}_{\text{rec}(l)}^{k+1}$ (see (22)), these differences are of equal size. In consequence, when $k \to \infty$, $\hat{\lambda}_{\text{trans}(l)}^{k+1} \to +\infty$ and $\hat{\lambda}_{\text{rec}(l)}^{k+1} \to -\infty$ with equal rates. Therefore, if the algorithm has not converged after a reasonable number of iterations while $\hat{\lambda}_{\text{trans}(l)}^{k+1}$ and $\hat{\lambda}_{\text{rec}(l)}^{k+1}, l \in \mathcal{L} \setminus \mathcal{S}(N + 1)$, have been constantly increasing and decreasing, respectively, for some pair(s) of sensors, the problem (12) can be inferred to be infeasible, and the sensors can terminate the algorithm.
2.2.3 Distributed optimization via DD

As an alternative to the proposed ADMM algorithm, a distributed method based on DD for solving (12) is derived. DD is a common technique in the field of decentralized optimization, and, thus, establishes a sound benchmark for the ADMM algorithm. DD decomposes the problem (12) horizontally across the nodes while respecting the layering ideology, referred to as vertical decomposition. To this end, each sensor solves a resource allocation subproblem involving the physical layer and a routing subproblem involving the network layer. The subproblems are coordinated by the dual problem and local exchanges of some optimization variables.

Lagrange dual problem

Proximal regularization: Since the objective function of (12) is not strictly convex in \( f \), the recovery of these primal variables from the associated dual problem is difficult [31, 33]. A customary approach to overcome this inherent problem is to add a small quadratic regularization term \( \sum_{l \in \mathcal{L}} \varepsilon_{dd} f_l^2 \) to the objective, where \( \varepsilon_{dd} > 0 \) is a positive parameter [31, 33]. A drawback of this is that the solution of such a regularized problem deviates from the original one. In this thesis, the lack of strict convexity is handled in a novel way by adding a proximal regularization term

\[
\frac{1}{2\delta_{dd}} \| f - \bar{f} \|_2^2 = \frac{1}{2\delta_{dd}} \sum_{l \in \mathcal{L}} (f_l - \bar{f}_l)^2
\]

\( \bar{f} = [\bar{f}_1 \cdots \bar{f}_L]^T \) are auxiliary optimization variables for the links, and \( \delta_{dd} > 0 \) is a positive scalar parameter [10, Sect. 3.4.3].

Partial Lagrangian: A partial Lagrangian for the proximally regularized problem in respect to the capacity constraint (10) (i.e., a complicating constraint) and the FCL constraint (6) (i.e., a coupling constraint) can be written as

\[
\begin{align*}
\Sigma_{dd}(p, w, f, \bar{f}, \nu, \xi) &= \sum_{l \in \mathcal{L}} \left( p_l + \frac{1}{2\delta_{dd}} (f_l - \bar{f}_l)^2 \right) + \\
&\sum_{l \in \mathcal{L}} \nu_l \left( f_l - w_l \log \left( 1 + \frac{p_l}{w_l} \right) \right) + \sum_{i \in \mathcal{N}} \xi_i \left( a_i^T f - r_i \right),
\end{align*}
\]

(23)

where \( \nu = [\nu_1 \cdots \nu_L]^T \) are the dual variables associated with the capacity constraints,

\( \xi = [\xi_1 \cdots \xi_N]^T \) are the dual variables associated with the FCL constraints,

\( p = [p_1 \cdots p_L]^T \) and \( w = [w_1 \cdots w_L]^T \) are the transmit power and bandwidth vectors, respectively, and \( a_i^T = [a_{i1} \cdots a_{id}] \) is the \( i \)th row of matrix \( A \).
Dual function: Let $\mathcal{F}$ be a feasible set of variables as

$$
\mathcal{F} = \left\{ (p, w, f, \tilde{f}) \mid \begin{array}{l}
\sum_{l \in \Theta(i)} p_l \leq P, \ i \in \mathcal{N} \\
\sum_{l \in \Theta(i)} w_l \leq W, \ i \in \mathcal{N} \\
p \succeq 0, \ w \succeq 0, \ f \succeq 0, \ \tilde{f} \in \mathbb{R}^L
\end{array} \right\}. \quad (24)
$$

The dual function associated with the partial Lagrangian in (23) is given as

$$
\mathcal{D}(\nu, \xi) = \inf_{(p, w, f, \tilde{f}) \in \mathcal{F}} \mathcal{L}(p, w, f, \tilde{f}, \nu, \xi). \quad (25)
$$

The dual function can be evaluated separately in the flow and auxiliary variables $(f, \tilde{f})$, and the resource variables $(p, w)$, respectively, i.e.,

$$
\mathcal{D}(\nu, \xi) = \mathcal{D}_{\text{rout}}(\nu, \xi) + \mathcal{D}_{\text{res}}(\nu), \quad (26)
$$

where $\mathcal{D}_{\text{rout}}(\nu, \xi)$ is the dual function associated with a routing subproblem

$$
\mathcal{D}_{\text{rout}}(\nu, \xi) = \inf_{f \succeq 0} \left\{ \sum_{l \in \mathcal{L}} \left[ v_l f_l + \frac{1}{2 \delta_{\text{ld}}} (f_l - \tilde{f}_l)^2 \right] + \sum_{i \in \mathcal{N}} \xi_i (a_i^T f - r_i) \right\}, \quad (27)
$$

and $\mathcal{D}_{\text{res}}(\nu)$ is the dual function associated with a resource allocation subproblem

$$
\mathcal{D}_{\text{res}}(\nu) = \inf_{p, w} \left\{ \sum_{l \in \mathcal{L}} \left[ p_l - v_l w_l \log \left( 1 + \frac{\hat{b}_l p_l}{w_l} \right) \right] \right\}. \quad (28)
$$

Dual problem: Using (26) – (28), the dual problem of the proximally regularized version of the primal problem (12) is given as

$$
\begin{align*}
\text{maximize} \quad & \mathcal{D}_{\text{rout}}(\nu, \xi) + \mathcal{D}_{\text{res}}(\nu) \\
\text{subject to} \quad & \nu \succeq 0, \ \xi \in \mathbb{R}^N
\end{align*} \quad (29)
$$

with optimization variables $\nu$ and $\xi$. Since the problem (12) is convex, and Slater’s condition for constraint qualification is assumed to hold\(^8\), the duality gap is zero, and the primal problem can be solved via its dual [9, Sect. 5.2.3].

\(^8\)The capacity constraint (10) is assumed to hold with the strict inequality for a primal feasible solution. Note that as this is typically true in practice, the assumption is not very strict [31].
Distributed dual based algorithm

In order to arrive at a distributed algorithm, the dual problem in (29) is solved iteratively via the projected subgradient method [13, 26, 31, 33]. Thus, at each iteration $k$, the dual functions in (27) and (28) are evaluated for given dual variables $\nu^k$ and $\xi^k$, followed by subgradient updates for $\nu^k$ and $\xi^k$. By virtue of the separable dual function, the method leads to independently solving a routing subproblem in the network layer ($D_{\text{rout}}(\nu, \xi)$) and a resource allocation subproblem in the physical layer ($D_{\text{res}}(\nu)$).

**Dual variable update:** Let $f^k$, $p^k$, and $w^k$ denote the optimal primal variables obtained from (27) and (28) at iteration $k$. The dual variables are updated at each iteration $k$ with the projected subgradient method as

$$
\nu^{k+1}_l = \nu^k_l + \alpha^\nu_k \left( f^k_l - w^k_l \log \left( \frac{\| p^k_L \|_{\infty}}{w^k_L} \right) \right), \; l \in \mathcal{L}
$$

$$
\xi^{k+1}_i = \xi^k_i + \alpha^\xi_k \left( \sum_{l \in \mathcal{L}} a^T_l f^k_l - r_i \right), \; i \in \mathcal{N},
$$

where $\alpha^\nu_k$ and $\alpha^\xi_k$ are the step sizes at iteration $k$, and $\left[ \cdot \right]^+$ denotes the projection onto the set of non-negative reals. Similar subgradient updates, albeit with a different regularization, can be found in, e.g., [33]. It is worth mentioning that neither the regularization used in [33] nor the proximal regularization (23) affect the derived subgradients.

The remaining problem is how to obtain the primal variables $f^k$, $p^k$, and $w^k$ for given $\nu^k$ and $\xi^k$ at each iteration $k$. These are elaborated next.

**Routing subproblem:** The routing subproblem in the network layer involves evaluating (27) for given dual variables $\nu^k$ and $\xi^k$ at iteration $k$, i.e., solving

$$
\begin{align*}
\text{minimize} & \quad \sum_{l \in \mathcal{L}} \left[ \nu^k_l f_l + \xi^k_l f_l + \frac{1}{2\delta_{dl}} (f_l - \bar{f}_l)^2 \right] - \sum_{i \in \mathcal{N}} \xi^k_i r_i \\
\text{subject to} & \quad f_l \geq 0, \; \bar{f}_l \in \mathbb{R}, \; l \in \mathcal{L},
\end{align*}
$$

with variables $f = [f_1 \cdots f_L]^T$ and $\bar{f} = [\bar{f}_1 \cdots \bar{f}_L]^T$, where $\bar{a}_l = [a_{l1} \cdots a_{LN}]^T$ is the $l$th column of matrix $A$. It can be seen that for a fixed $\bar{f}$, the objective function of (32) is strictly convex in $f$, and vice versa. Hence, (32) can be solved via the nonlinear Gauss-Seidel method which alternately minimizes over $f$ and $\bar{f}$ [10, Sect. 3.4.3]. Owing to the separability of (32), these alternate minimization steps with an initial $\bar{f}_l^0$, $l \in \mathcal{L}$, can
be written as per-link subproblems as

\[
\begin{align*}
 f^{k*}_{l} &= \arg\min_{f_l \geq 0} \left\{ v^k_l f_l + \frac{1}{2\delta_{dd}} \left( f_l - \tilde{f}^{k*}_l \right)^2 + \mathbf{u}^T \mathbf{x}^k f_l \right\}, \quad l \in \mathcal{L} \quad (33a) \\
 \tilde{f}^{k*}_l &= \arg\min_{f_l \in \mathbb{R}} \left( \bar{f}^{k*}_l - \tilde{f}_l \right)^2, \quad l \in \mathcal{L}. \quad (33b)
\end{align*}
\]

The optimal solution to (33b) is \( \tilde{f}^{k*}_l = \bar{f}^{k*}_l \). On this account, the auxiliary variables \( \tilde{f} \) can be eliminated from the loop by substituting \( \tilde{f}^{k*}_l = \bar{f}^{k*}_l, \forall l \in \mathcal{L} \), in (33a). Thus, the Gauss-Seidel iterations for solving (32) become

\[
 f^{k*}_l = \arg\min_{f_l \geq 0} \left\{ v^k_l f_l + \frac{1}{2\delta_{dd}} \left( f_l - \bar{f}^{k*}_l \right)^2 + \mathbf{u}^T \mathbf{x}^k f_l \right\}, \quad l \in \mathcal{L}. \quad (34)
\]

Instead of repeating the minimization (34) until convergence, the method is applied by taking only one step with respect to each flow \( f_l, l \in \mathcal{L} \), at each iteration \( k \). Hence, the flows are asymptotically driven towards the optimum while quadratically penalizing for the deviation from the previously attained values \( \bar{f}^{k*}_l, l \in \mathcal{L} \). This can be interpreted as a one state memory in terms of the flow variables.

Due to the strict convexity of (34), the unique optimal solution can be found via the derivative, resulting in

\[
 f^{k*}_{l,\text{prox}} = \left[ f^{k*}_l - \delta_{dd} \left( v^k_l + \mathbf{u}^T \mathbf{x}^k \right) \right]^+, \quad l \in \mathcal{L}. \quad (35)
\]

The step (35) can be seen as the gradient projection iteration [10, Sect. 3.4.3]. The parameter \( \delta_{dd} \) affects the convergence in two ways as it weights the quadratic difference in the objective of (34) and acts as the step size in (35).

Numerical results are also provided for the quadratically regularized DD method in Section 2.3. Accordingly, the term \( \frac{1}{2\delta_{dd}} (f_l - \bar{f}_l)^2 \) in (32) is replaced with \( \varepsilon_{dd} f_l^2 \), and the optimal flow variables at each iteration \( k \) become [33]

\[
 f^{k*}_{l,\text{quad}} = \frac{1}{2\varepsilon_{dd}} \left[ -v^k_l - \mathbf{u}^T \mathbf{x}^k \right]^+, \quad l \in \mathcal{L}. \quad (36)
\]

**Resource allocation subproblem:** The resource allocation subproblem in the physical layer involves evaluating (28) for given dual variables \( \mathbf{v}^k \) at iteration \( k \), i.e.,
solving

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in \mathcal{N}} \sum_{l \in \mathcal{O}(i)} \left[p_l - \nu_k l w_l \log \left(1 + \gamma_l p_l w_l \right)\right] \\
\text{subject to} & \quad \sum_{l \in \mathcal{O}(i)} p_l \leq P, \quad \sum_{l \in \mathcal{O}(i)} w_l \leq W_i, \quad i \in \mathcal{N}, \\
& \quad p_i \succeq 0, \quad w_i \succeq 0, \quad i \in \mathcal{N},
\end{align*}
\]  

(37)

with variables \( p_i \in \mathbb{R}^{\mathcal{O}(i)} \) and \( w_i \in \mathbb{R}^{\mathcal{O}(i)}, \quad i \in \mathcal{N} \). The problem (37) decomposes into \( N \) per-node subproblems, i.e., each sensor \( i \in \mathcal{N} \) obtains optimal local resource variables by solving the convex optimization problem

\[
\left\{ p_i^k, w_i^k \right\}_{i \in \mathcal{O}(i)} = \arg\min_{\sum_{l \in \mathcal{O}(i)} p_l \leq P, \sum_{l \in \mathcal{O}(i)} w_l \leq W_i, \quad p_i \succeq 0, \quad w_i \succeq 0} \sum_{i \in \mathcal{O}(i)} \left[p_l - \nu_k l w_l \log \left(1 + \gamma_l p_l w_l \right)\right] 
\]  

(38)

with variables \( p_i \) and \( w_i \). Due to the convexity, (38) can be solved by using, e.g., interior-point methods [9, Ch. 11]. It is worth pointing out that for a fixed bandwidth allocation at a sensor \( i \in \mathcal{N} \), the optimal power allocation \( p_i^k, \quad l \in \mathcal{O}(i) \), can easily be found via the standard iterative water-filling method [31], [9, p. 245] and [158, Sect. 9.4].

Algorithm implementation

The distributed DD algorithm for the joint resource allocation and routing optimization in (12) is summarized in Algorithm 2. The communication overhead at each iteration \( k \) consists of conveying \( f_i^k \) and \( \xi_i^{k+1} \) across each link \( l \in \mathcal{L} \setminus \mathcal{I}(N + 1) \), which is precisely the same amount as for the ADMM method in Algorithm 1.

2.2.4 Alternative decomposition approaches

To clarify and compare the distinct decomposition structures, Fig. 6 presents a hierarchical illustration of decomposing the global primal problem (12) via the distributed consensus ADMM (i.e., Algorithm 1), and the DD method (i.e., Algorithm 2). The DD first decouples the problem vertically (i.e., with respect to the protocol layers) into routing and resource allocation subproblems, followed by the horizontal decomposition across the sensors. The coordination between the layers is mastered by the dual variables \( \mathbf{v} \), each of which represents the price for allocating a certain amount of resources for each link. The consensus ADMM applies only one level
Algorithm 2 Distributed joint resource and routing optimization via DD

a) Initializations:
   i) Set $k = 1$; ii) initialize the flow variables $f_k^l$, $l \in \mathcal{L}$, and the dual variables $\nu_k^l$, $l \in \mathcal{L}$, and $\xi^k_i$, $\forall i \in \mathcal{N}$, e.g., with zeros.

b) Iterations:
   I. Each sensor node $i = 1, 2, \ldots, N$:
      
      **Routing subproblem in the network layer**
      1) Sets the optimal flow variables $f_k^l$, $l \in \mathcal{O}(i)$, according to (35).
      2) Communicates the flow variables $f_k^l$, $l \in \mathcal{O}(i)$, to the receiving nodes rec($l$) of its outgoing links $l \in \mathcal{O}(i)$.

      **Resource allocation subproblem in the physical layer**
      3) Finds the optimal power and bandwidth variables $p_k^l$ and $w_k^l$, $l \in \mathcal{O}(i)$, respectively, by solving the problem in (38).

      **Dual problem**
      4) Updates the dual variables according to (30) and (31) to obtain $\nu_{i}^{k+1}$, $l \in \mathcal{O}(i)$, and $\xi_{i}^{k+1}$, respectively.
      5) Communicates the dual variable $\xi_{i}^{k+1}$ to the transmitting nodes tran($l$) of its incoming links $l \in \mathcal{S}(i)$.
   II. If a predefined stopping criterion is met then stop; otherwise, set $k = k + 1$, and go to Step I.

decomposition by solely decoupling the problem between the sensors. In summary, the ADMM routine relies on fewer decomposition levels and also overrides the philosophy on the protocol layer separation.

### 2.3 Numerical results

This section presents numerical results to show the convergence behavior of the proposed ADMM algorithm in comparison to the DD based approaches. Moreover, energy efficiency improvements in terms of sum transmit power usage under different bandwidth allocation schemes are demonstrated.
Fig. 6. Hierarchical representation of different decomposition steps for (a) the distributed consensus ADMM algorithm, and (b) the DD algorithm ([51] © 2013 IEEE).
2.3.1 Simulation setup

Consider WSNs with \( N = 4 \), \( N = 8 \), and \( N = 16 \) as depicted in Fig. 7. A network is created by deploying one sensor in each 100 \( \times \) 100 square area according to a uniform distribution, and placing the sink in the center area. The maximum communication range \( d_{tx} \) is empirically chosen to ensure typical WSNs with local connectivity. The rates \( r_i \), \( i \in \mathcal{N} \), are determined similarly to [250, Sect. VI]: Slepian-Wolf (SW) coding is applied by assuming a shortest path tree, jointly Gaussian sources with unit variance, and the distance based correlation model \( \exp\{-4.5 \times 10^{-5}d_{ij}^2\} \). The resulted rates are between 1.62 to 2.0 (bits/transmission slot) for \( N = 4 \), 1.2 to 2.0 for \( N = 8 \), and 0.6 to 2.0 for \( N = 16 \). Channel coefficients \( |\kappa_l| \), \( l \in \mathcal{L} \), drawn from the Rayleigh distribution, are spatially uncorrelated and uncorrelated in time between different channel initializations. Consequently, the network realizations are unique and have different optimal solutions. The convex problems are solved with CVX, a package for specifying and solving convex programs [251].

2.3.2 Convergence of distributed algorithms

The convergence behavior of the proposed ADMM algorithm (i.e., Algorithm 1) is compared against the DD based algorithms (i.e., Algorithm 2 with both quadratic (DD-qd) and proximal (DD-px) regularization terms). Diminishing step sizes of the form \( \alpha_k^\nu = \alpha_{\nu,sc}/\sqrt{k} \) and \( \alpha_k^\xi = \alpha_{\xi,sc}/\sqrt{k} \) [26] are used for the subgradient updates in the DD algorithms, where \( \alpha_{\nu,sc} \) and \( \alpha_{\xi,sc} \) are positive constant parameters. All step size parameters \( \alpha_{\nu,sc} \), \( \alpha_{\xi,sc} \), \( \epsilon_{dd} \), and \( \delta_{dd} \) in the DD, and penalty parameter \( \rho \) in the ADMM are chosen by empirical test runs. If the results are presented for "fine-tuned" parameters, it refers to the case where each algorithm was run multiple times with a set of step size parameters, and selecting the parameter(s) that yielded the best convergence.

Constraint feasibility and solution accuracy

Since the ADMM and DD algorithms require the same amount of local information exchanges at each iteration, they are compared by measuring the total number of required iterations to generate near-feasible solutions, and then, evaluating the resulting objective function accuracy. Since the FCL (6) is the only constraint that remains infeasible until a distributed algorithm converges, a stopping criterion is defined by
means of the normalized FCL violation as

\[ FCL^k_{\text{viol}} \triangleq \sum_{i\in\mathcal{S}} \frac{|a_i^T c^k - r_i|}{r_i}, \quad (39) \]

where \( c^k = [c_1(p^k_1, w^k_1) \cdots c_L(p^k_L, w^k_L)]^T \) is the link capacity vector at iteration \( k \). Hence, at each iteration \( k \), \( |a_i^T c^k - r_i|/r_i \) represents the unbalance of the incoming and outgoing flows of the \( i \)th sensor, normalized in respect to its source rate \( r_i, i \in \mathcal{N} \). Instead of the flow variables, the capacities \( c_l(p^k_l, w^k_l), l \in \mathcal{L} \), are used to determine the supported

Fig. 7. The used network topologies with (a) \( N = 4 \) and \( L = 16 \), (b) \( N = 8 \) and \( L = 40 \), and (c) \( N = 16 \) and \( L = 150 \). The axes stand for the unit lengths, the red square is the sink node, and all available links are shown with blue lines ([51] © 2013 IEEE).
rates on the links with a current resource allocation. If an algorithm converges, \( FCL_k \rightarrow 0 \) as \( k \rightarrow \infty \). As the stopping criterion, an algorithm is terminated when \( FCL_k < 0.01 \).

Let \( p^* \) denote the optimal value of the problem (12). Let \( d_{\text{stop}}^k \) be the objective function value of a distributed algorithm in its last iteration. After an algorithm has converged, i.e., reached \( FCL_k \rightarrow 0 \), the solution suboptimality is measured via the normalized difference as \( |d_{\text{stop}}^k - p^*|/p^* \).

**Convergence behavior example**

Fig. 8 illustrates the convergence of the ADMM, DD-qd, and DD-px algorithms with fine-tuned step size parameters for \( N = 8 \) and fixed bandwidths \( w_l = 1, \forall l \in L \). Fig. 8(a) shows the evolution of the FCL violation, and Fig. 8(b) indicates the normalized error in the final objective value [%]. DD-qd is plotted for \( \varepsilon_{\text{dd}} = \{0.02, 0.01, 0.005\} \) to show the impact of the regularization weight parameter on the convergence.

Fig. 8 shows the superior convergence speed of the ADMM algorithm: it converged in 21 iterations to a solution with a 0.5 % normalized error in the objective value. The DD approaches are significantly slower: DD-px converged in 133 iterations and DD-qd within 163 – 750 depending on the value of \( \varepsilon_{\text{dd}} \). The trade-off of \( \varepsilon_{\text{dd}} \) in DD-qd is clearly visible: by reducing \( \varepsilon_{\text{dd}} \), the convergence rate slows down, but the accuracy of the objective value increases, and vice versa. Thus, the quadratic regularization sacrifices the solution accuracy at the cost of increased convergence rate. Since DD-px converged much faster and yielded higher solution accuracy than any DD-qd variant, the developed DD-px is used to generate all subsequent results for the DD.

**Convergence of flow variables**

Fig. 9 depicts the convergence of the ADMM against the DD for fine-tuned step size parameters by showing the evolution of flow variables (global flow variables for the ADMM) for single network instances with \( N = 4, N = 8 \), and \( N = 16 \). For \( N = 4 \) and \( N = 8 \), bandwidth optimization (BW-opt) was incorporated with \( W_i = 3, \forall i \in N \).

---

Note that the capacity constraint in (12) becomes tight at the optimal solution, i.e., \( \lim_{k \to \infty} f^k_{l} = c_l(p^k_l, w^k_l) \), \( \forall l \in L \).
Fig. 8. Convergence examples of the ADMM algorithm, and the DD algorithms with different regularization terms for $N = 8$. (a) FCL violation, and (b) normalized error in the final objective value ([51] © 2013 IEEE).
whereas fixed bandwidths (BW-fix) $w_l = 1, \forall l \in \mathcal{L}$, were used for $N = 16$. The vertical dashed lines indicate the points when an algorithm reached $F_{\text{CL}}^{l_{\text{inal}}} < 0.01$.

The figure shows that the ADMM converges for all network sizes and for both bandwidth allocation schemes in just dozens of iterations (13 – 33), whereas the DD required several hundreds (156 – 573). The difference is roughly one order of magnitude which implies a significantly increased communication overhead for the DD algorithm compared to the ADMM. Moreover, Figs. 9(b) and (d) show that, regardless of the network size, the DD is incompatible with BW-opt as it severely suffers from slow and fluctuating convergence behavior. All methods obtained decent objective accuracy with error values $< 0.1 \%$.

### Average convergence behavior over multiple channel realizations

To demonstrate the average convergence behavior and robustness against fluctuating channel conditions, the algorithms were run over 500 random channel realizations for $N = 8$ and $N = 16$ with fixed bandwidths $w_l = 1, \forall l \in \mathcal{L}$. The average number of required iterations and the average solution accuracy are depicted in Figs. 10(a) and (b), respectively. For each network size, the algorithms were run with empirically tuned sets of step sizes which remained constant over all channel realizations, i.e., there was no fine-tuning between the network instances. While the choice for an optimal $\rho$ remains an open problem, it can be inferred that $\rho$ should be of the same order as the typical values of the objective function. As the source rates are around units, and the sensors operate at a low signal-to-noise ratio region, the objective function $\sum_{l \in \mathcal{L}} p_l$ gets values of around the units as well. Thus, the penalty parameter was varied as $\rho = \{0.25, 0.5, 0.75, 1.0\}$ for $N = 8$, and $\rho = \{0.6, 0.8, 1.0\}$ for $N = 16$ to observe its impact on the convergence.

Fig. 10(a) points out an advantageous features of the ADMM: it converged on average after around 30 – 39 iterations for $N = 8$, and 45 – 51 for $N = 16$ with a range of values of $\rho$ without tuning it separately for each network instance. The solution accuracies shown in Fig. 10(b) vary between 0.09 – 1.76 % depending on the size of $\rho$; the larger $\rho$ is, the less accurate is the final solution. Note, however, that if an application requires more accurate solutions, the stopping criterion can readily be adjusted on demand to allow for few more steps. The DD required on average over 200 iterations for $N = 8$, and over 640 for $N = 16$ to converge to near-optimal solutions with the solution accuracy of around 0.08 %. Comparing these average convergence
Fig. 9. Convergence of the (global) flow variables for (a) ADMM with BW-opt, \(N = 4, \rho = 0.008\), (b) DD with BW-opt, \(N = 4, \delta_{AD} = 4.0, \alpha^2 = 0.01/\sqrt{k}\), (c) ADMM with BW-opt, \(N = 8, \rho = 0.01\), (d) DD with BW-opt, \(N = 8, \delta_{AD} = 5.0, \alpha^2 = 0.05/\sqrt{k}\), (e) ADMM with BW-fix, \(N = 16, \rho = 0.6\), and (f) DD with BW-fix, \(N = 16, \delta_{AD} = 5.0, \alpha^2 = 0.08/\sqrt{k}, \alpha^2 = 0.25/\sqrt{k}\) [51] © 2013 IEEE.
results to the cases where the step sizes were fine-tuned for a single network setup (i.e., Fig. 8 for $N = 8$, and Figs. 9(e) and (f) for $N = 16$), the DD is substantially more sensitive to the choice of the step size parameters than the ADMM. On this account, the proposed ADMM algorithm is more suitable for practical implementation where frequent parameter adjustments are not typically possible.

2.3.3 System performance under different bandwidth allocation schemes

To illustrate the influence of bandwidth allocation optimization on the total transmit power usage in WSNs, problem (12) was incorporated following three different orthogonal bandwidth allocation schemes:

1. BW-net: Global bandwidth optimization, where the per-node bandwidth constraints of (12) are changed to $\sum_{l \in L} w_l \leq W_{\text{net}}$, where $W_{\text{net}}$ is the total available bandwidth in the WSN. This scheme requires centralized coordination, and, thus, is used as a benchmark.

2. BW-node: Per-node local bandwidth optimization equal to (12), where each node $i \in \mathcal{N}$ is pre-allocated the same portion of total bandwidth, i.e., $W_i = \frac{W_{\text{net}}}{N}$.

3. BW-link: Fixed bandwidth allocation, where each sensor $i \in \mathcal{N}$ allocates its available total bandwidth $W_i = \frac{W_{\text{net}}}{N}$ evenly to its outgoing links, i.e., $w_l = \frac{W_{\text{net}}}{N|\mathcal{O}(i)|}$, $l \in \mathcal{O}(i)$.

Table 2 lists the achieved minimum, average, and maximum ratios [%] in terms of the total power usage under different bandwidth allocation schemes averaged over 500 channel realizations for $N = 8$ and $N = 16$. It can be inferred that compared to the BW-link method, the BW-node method substantially reduces the use of transmit power: the obtained savings are on average from 70 % up to 90 %, increasing with the network size, and even 98 % in a single network realization for $N = 16$. However, there is only an 8 % – 12 % average reduction in the total power when the bandwidth is optimized globally in the WSN (centralized BW-net) instead of performing local optimization (decentralized BW-node). Hence, the largest proportional enhancement in the energy efficiency is achievable via the proposed BW-node scheme which, except for the preliminary bandwidth allocation, is local and, thus, preserves the desired distributed functionality in the network.
Fig. 10. Convergence of the ADMM and DD for $N = 8$ and $N = 16$ averaged over 500 random channel realizations: (a) the average number of required iterations, and (b) average normalized error in the final objective value ([51] © 2013 IEEE).
Table 2. Percentage ratios for the total transmit power usage under different bandwidth allocation schemes with $N = 8$ and $N = 16$ ([51] © 2013 IEEE).

<table>
<thead>
<tr>
<th>$N$</th>
<th>$p^*(\text{BW-node}) \times 100$ [%]</th>
<th>$p^*(\text{BW-link}) \times 100$ [%]</th>
<th>$p^*(\text{BW-net}) \times 100$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min / mean / max</td>
<td>min / mean / max</td>
<td>min / mean / max</td>
</tr>
<tr>
<td>8</td>
<td>11.3 / 31.0 / 46.2</td>
<td>9.30 / 28.8 / 45.2</td>
<td>76.1 / 92.1 / 99.6</td>
</tr>
<tr>
<td>16</td>
<td>2.04 / 7.89 / 16.0</td>
<td>1.86 / 7.00 / 14.9</td>
<td>65.9 / 87.6 / 96.2</td>
</tr>
</tbody>
</table>

### 2.4 Summary and discussion

This chapter addressed distributed total transmit power minimization via a cross-layer design in a multi-hop SSDG WSN. A novel distributed consensus ADMM algorithm, which jointly optimizes the resource allocation and the routing for given source rates, was derived. The algorithm decomposes the original problem into sensor-specific subproblems involving only local variables, which are then driven into consensus with the ADMM using few local information exchanges. Moreover, a new DD variant based on the proximal regularization was developed for the benchmark.

Based on the numerical experiments, the proposed ADMM algorithm converges to near-feasible and near-optimal solutions within an order of magnitude smaller number of iterations compared to state-of-the-art DD based methods. Additionally, the ADMM was demonstrated to have good scalability with respect to the network size and to enjoy robust convergence under various channel conditions without fine-tuning the step size. Since data communications substantially consume the batteries, the requirement of only a few dozen variable exchanges in the optimization phase of the ADMM is conducive to practical implementation. Besides the energy savings, the fast optimization phase also reduces the delay for initiating the transmissions of actual application data. In terms of system performance, the results showed that notable gains are achievable by incorporating bandwidth optimization in an FDMA based data gathering WSN. In particular, a majority of the gain that is achievable via the global (centralized) bandwidth optimization can be achieved by local per-node optimization, i.e., via (12).

The optimization phase of a distributed algorithm consumes sensors’ battery energy during information exchanges. The contribution of these communications to the sensors’ energy consumption was not explicitly taken into account in the optimization design and neither in the simulations. Nevertheless, in such a WSN application where the source rates and channel conditions are not constantly changing, a distributed routing solution can be used over several consecutive sensing periods. On this account, the significantly reduced transmit power usage obtained via the joint optimization
offsets the slight additional cost from the training phase in the long run. Moreover, compared to centralized optimization, the distributed ADMM algorithm can rapidly adapt to slight changes in the networking environment with a low communication cost by updating the solution via local messaging by means of a "warm start". In conclusion, applying the consensus ADMM for a network optimization framework that involves coupling constraints is a promising approach when striving for an energy-efficient distributed data gathering solution in WSNs.
This chapter focuses on distributed sequential compressed acquisition and progressive reconstruction of spatially and temporally correlated sensor data streams in WSNs through compressed sensing (CS). A novel sliding window based recursive CS data collection method is devised. The method uses $\ell_2$-regularization and iterative reweighted $\ell_1$-minimization (IRW-$\ell_1$) 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 state-of-the-art CS methods.

The chapter is organized as follows. The WSN monitoring model is defined in Section 3.1. Section 3.2 introduces the framework of sequential CS for correlated sensor data streams. The progressive CS reconstruction method is derived in Section 3.3. The simulation results are provided in Section 3.4. The chapter is summarized in Section 3.5.

3.1 Wireless sensor network monitoring

Consider a multi-hop SSDG 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 (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.
Fig. 11. A data gathering WSN with spatially and temporally correlated data for \( N = 8 \) and \( s = 5 \) ([148] © 2015 IEEE).

### 3.1.1 Source model

The sensor observations are assumed to encompass both spatial and temporal correlations, accounting for various environmental sensing applications in a densely deployed WSN [125, 252]. The prevailing physical phenomenon in the monitoring field is generated by a set of independent, randomly located, time-varying sources \( \mathcal{S} = \{1, \ldots, S\} \). For example, the sources could be 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. 11.

The impact of each source \( s \in \mathcal{S} \) on the sensors’ readings is modeled through time-invariant, real-valued influence functions \( h_{nn} \in \mathbb{R}, n \in \mathcal{N} \). 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 [128]

\[
x_n(t) = \sum_{s \in \mathcal{S}} h_{nn} \beta_s(t) = h_{nn}^T \mathbf{\beta}(t), \quad n \in \mathcal{N}, \quad t = 1, 2, \ldots,
\]  

### (40)
where vector $\mathbf{\beta}(t) = [\beta_1(t) \cdots \beta_S(t)]^T$ consists of the source magnitudes at time instant $t$, and $\mathbf{h}_n = [h_{n1} \cdots h_{nS}]^T$ is the aggregate influence vector on sensor $n \in \mathcal{N}$.

By (40), 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\(^\text{10}\) [252–254] 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 piecwise 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. For instance, 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 [58, 125, 254, 255].

\subsection*{3.1.2 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 of (40) to the sink should be minimized. For assessing the energy consumption of wireless access, a customary approach used in CS data gathering works is adopted [58]: the transmit power usage of a sensor is quantified in proportion to the number of transmitted CS measurements (this will become clearer in Section 3.2.3). 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. The protocol layers are assumed to be appropriately designed to support the successful delivery of data packets across the WSN.

\(^{10}\)This is not a requirement for the developed methods; other spatial correlation models could be incorporated as well.
3.2 Sequential CS of correlated sensor data streams

In this section, a sequential CS framework for delivering jointly correlated sensor data streams, i.e., the observed sensor readings $x_n(t)$ of (40), to the sink with the aim to minimize the necessary number of transmitted data units is developed. The sensors’ readings are periodically delivered to the sink in the form of CS 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 derived 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.

3.2.1 Sliding data window

Let $X(t) \in \mathbb{R}^{N \times W}$ denote a data window at time instant $t$ with a window size$^{11} W \geq 1$. The matrix $X(t)$ consists of $W$ consecutive readings of all $N$ sensors at time instants $\{t - W + 1, \ldots, t\}$ as

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

(41)

where $x_n(t)$ is the reading of sensor $n \in \mathcal{N}$ at time instant $t$ given in (40). Let vector $x_n(t) = [x_n(t - W + 1) \cdots x_n(t)]^T$ denote the $n$th row of $X(t)$, containing the readings of sensor $n$ at time instants $\{t - W + 1, \ldots, t\}$. Similarly, let vector $x(t) = [x_1(t) \cdots x_N(t)]^T$ denote the columns of $X(t)$, each of which contains the sensors’ readings at a time instant $t$. Accordingly, $X(t)$ can be expressed as

$$X(t) = [x(t - W + 1) \cdots x(t)] = [x_1(t) \cdots x_N(t)]^T.$$  

(42)

Unless otherwise stated, $t \geq W$ is assumed in the following derivations. The sliding window structure with respect to the sensor data streams is illustrated in Fig. 12(a).

$^{11}$To clarify the presentation, a fixed window length $W$ is assumed. As a possible extension, the method could benefit from using a window which size varies over the time instants.
3.2.2 Exploiting joint spatio-temporal correlation

Recall that according to the sensor data model defined in Section 3.1, each data window \( X(t) \) in (41) encompasses both a spatial and temporal correlation. It is assumed that
there exists a basis $\Psi_S \in \mathbb{R}^{N \times N}$ for the spatial domain in which each column of $X(t)$ has a compressible representation, i.e., $x(t) = \Psi_S \theta(t)$, where vector $\theta(t) \in \mathbb{R}^N$ contains the spatial transform coefficients at slot $t$. Accordingly, $X(t)$ can be written as

$$X(t) = [x(t-W+1) \cdots x(t)]$$

$$= \Psi_S [\theta(t-W+1) \cdots \theta(t)]$$

$$= \Psi_S \Theta_S(t),$$

(43)

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

$$X_T(t) = [x_1(t) \cdots x_N(t)]$$

$$= \Psi_T [\theta_{T,1}(t) \cdots \theta_{T,N}(t)]$$

$$= \Psi_T \Theta_T(t),$$

(44)

where $\Theta_T(t) \triangleq [\theta_{T,1}(t) \cdots \theta_{T,N}(t)]$.

Kronecker sparsifying bases can succinctly combine the individual sparsifying bases of each signal dimension into a single transformation matrix [84, 256]. Thus, by merging the transformations in (43) and (44), $X(t)$ can be represented as

$$x(t) = \text{vec}(X(t)) = \text{vec}(\Psi_S Z(t) \Psi_T^T)$$

$$= (\Psi_T \otimes \Psi_S) \text{vec}(Z(t))$$

$$= \Psi z(t),$$

(45)

where $x(t) = [x^T(t-W+1) \cdots x^T(t)]^T \in \mathbb{R}^{NW}$ is the vector-reshaped data window, the operator vec($A$) stacks the columns of matrix $A = [a_1 \cdots a_N]$ into the vector form $a = [a_1^T \cdots a_N^T]^T$, $Z(t) \in \mathbb{R}^{N \times W}$ is a matrix of the joint transform coefficients for $X(t)$, $\otimes$ denotes the Kronecker product, $\Psi = (\Psi_T \otimes \Psi_S) \in \mathbb{R}^{NW \times NW}$ is the Kronecker sparsifying basis, and $z(t) = \text{vec}(Z(t)) \in \mathbb{R}^{NW}$ is the joint transform coefficient vector. In summary, $X(t)$ has a 2D-separable transform $Z(t) = \Psi_S^{-1} X(t) \Psi_T^T$, where $\Psi_S^{-1}$ operates on the columns of $X(t)$ and $\Psi_T^{-1}$ on its rows [132]. As a similar type of separable structure, the channel covariance matrix of a multiple-input multiple-output
channel can be well approximated by the Kronecker product of the covariance matrices seen from the transmitter and receiver [257].

In this thesis, data-independent bases $\Psi_S$ and $\Psi_T$ are used. 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 [133, 258–260]. 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., [125, 133]. Alternative transformations include data-dependent basis learning via principal component analysis [58, 261], and topology-dependent diffusion and graph wavelets [78].

### 3.2.3 CS encoding and decoding

#### CS encoding

Consider a CS encoding procedure, where at each time instant $t \geq 1$, the measurements are taken in respect to the current sensors’ readings $x(t)$. Thus, the sink acquires $M(t)$ linear CS measurements\(^{12}\) $y(t) = [y_1(t) \cdots y_{M(t)}(t)]^T \in \mathbb{R}^{M(t)}$ as

$$y(t) = \Phi(t)x(t), \quad t \geq 1,$$

(46)

where $\Phi(t) \in \mathbb{R}^{M(t) \times N}$, $M(t) \leq N$, is the measurement matrix for time instant $t$. According to (46), the measurement ensemble with respect to each data window $X(t)$ has a block-diagonal structure\(^{13}\) as

$$\begin{bmatrix}
  y(t - W + 1) \\
  \vdots \\
  y(t)
\end{bmatrix} =
\begin{bmatrix}
  \Phi(t - W + 1) & \cdots & 0 \\
  \vdots & \ddots & \vdots \\
  0 & \cdots & \Phi(t)
\end{bmatrix}
\begin{bmatrix}
  x(t - W + 1) \\
  \vdots \\
  x(t)
\end{bmatrix}. \quad (47)$$

By forming the vector $y(t) = [y^T(t - W + 1) \cdots y^T(t)]^T \in \mathbb{R}^{\sum_{t-W+1}^{t} M(t)}$ and the matrix $\Phi(t) = \text{diag}(\Phi(t - W + 1), \ldots, \Phi(t)) \in \mathbb{R}^{\sum_{t-W+1}^{t} M(t) \times NW}$, the measurement

\(^{12}\)Although noiseless CS measurements are assumed, the proposed method can straightforwardly be extended to handle noisy measurements with appropriate modifications.

\(^{13}\)For a line of work with overlapping measurement systems, see, e.g., [134, 135].
ensemble in (47) can be compactly written as

\[ y(t) = \Phi(t)x(t). \]  

(48)

Note that the measurement matrices \( \Phi(t) \in \mathbb{R}^{M(t) \times N}, \ t = 1, 2, \ldots \), 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. 12(b).

**Delivery of CS measurements**

As the structure of each \( \Phi(t) \) dictates how the measurements in (46) are delivered to the sink, each \( \Phi(t) \) greatly affects the sensors’ energy expenditure via the required wireless transmissions across the WSN [262]. Hence, to achieve efficient and applicable measurement acquisition for (46), a particular sparse binary \( \Phi(t) \in \mathbb{B}^{M(t) \times N} \) is used [58, 149, 261]: 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 the sensors, denoted by \( N(t) \subseteq N \), \( |N(t)| = M(t) \), i.e., the set of reading values becomes \( \{x_n(t) \mid n \in N(t) \} \). This scheme has been empirically shown to notably lower the communication costs in data gathering WSNs [149]. Additionally, it benefits from fast and efficient implementation of matrix multiplications in the CS decoding [133, 263].

The chosen sub-sampling type \( \Phi(t) \), and consequently \( \Phi(t) \), fall into the class of binary sparse measurement matrices [74, 133, 263–269], which are capable of achieving a CS recovery performance comparable to that of dense ones. The performance is affected by the mutual coherence between \( \Phi(t) \) and \( \Psi \), and by the RIP-\( p \) property, which is a weaker form of the restricted isometry property (RIP) introduced in [265]. An analysis and empirical studies on the mutual coherence between binary sparse measurement matrices and varying sparsifying bases can be found in, e.g., [263, 268, 269]. Intuitively, as the projections via the sub-sampling \( \Phi(t) \) are highly localized, successful CS decoding requires that the energy of the readings (40) is sufficiently spread out between the sensors, i.e., \( \Psi \) has a dense structure [266, 267, 269]. This is exactly the case in typical WSN applications where the correlated data has a compressible representation under a transformation such as DCT or DWT.
As the underlying phenomena may be geographically localized in WSNs, the measurements (46) should have a sufficient spatial distribution (i.e., resolution) to capture enough signal information for successful CS recovery. This can be achieved by a distributed random sampling scheme [58], 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 $^{14} p_{tx}^t \in [0,1]$. Consequently, for each $t$, on average $p_{tx}^t \mathcal{N}$ sensors communicate the readings to the sink. The scheme is favorable in WSNs since the routing becomes simple and topology-independent with a low communication overhead. Moreover, it can be realized with existing multi-hop routing protocols, including asynchronous individual transmissions and packet aggregation techniques [58, 68]. The random sampling scheme also enables a sensor to switch into a sleep mode, if during data gathering round $t$, it does not belong to $\mathcal{N}(t)$ and does not participate in the multi-hop packet forwarding as an intermediate node.

**CS decoding**

By exploiting the joint spatio-temporal compressibility (45), each data window $X(t)$ can be recovered from measurements (48) by solving the $\ell_1$-minimization problem (cf. (3))

$$\hat{x}(t) := \arg\min_{\hat{x}} \|\hat{x}\|_1 \quad \text{such that} \quad y(t) = \Phi(t) \Psi \hat{x},$$

(49)

where $\hat{x}(t)$ is an estimate of $x(t)$ that is used to form an estimate of $x(t)$ as $\hat{x}(t) = \Psi \hat{x}(t)$. The decoded data window $\hat{x}(t)$ is then reshaped as $\hat{X}(t) = [\hat{x}(t-W+1) \cdots \hat{x}(t)]$, where $\hat{X}(t) = [\hat{x}_1(t) \cdots \hat{x}_N(t)]^T$ contains the estimates of the sensors’ readings $x_n(t)$ of (40).

Thus, each decoding instant (49) produces estimates for the current sensors’ readings$^{15}$ and the $W-1$ previous ones. Note that with $W = 1$, (49) reduces to reconstructing each $\hat{x}(t)$ separately from measurements (46) by utilizing only the spatial domain compressibility (43), i.e., solving

$$\hat{\theta}_S(t) := \arg\min_{\hat{\theta}_S} \|\hat{\theta}_S\|_1 \quad \text{such that} \quad y(t) = \Phi(t) \Psi_S \hat{\theta}_S,$$

(50)

and reconstructing $\hat{x}(t) = \Psi_S \hat{\theta}_S(t)$.

$^{14}$The classic CS sampling process relies on non-adaptive incoherent measurements, i.e., each measurement has an equal priority [68]. For a line of work on CS with adaptive measurements, see, e.g., [270].

$^{15}$A slotted sliding window processing is assumed, i.e., at each time slot $t$, $x(t)$ is referred to as the current sensors’ readings which are reconstructed by (49) within the same slot.
Remark 1. The encoding/decoding in (49) without streaming processing constitutes a Kronecker CS (KCS) scheme [84]: a single data window \( X(t) \) is reconstructed from the associated measurement ensemble by treating \( X(t) \) as a finite-length signal. In general, a large \( W \) is preferable as it allows the KCS to utilize the temporal domain compressibility of \( X(t) \) over a longer interval, resulting in improved CS recovery performance. However, this causes two prominent drawbacks: 1) involving \( W \) consecutive sensors’ readings in the encoding induces a delay which is proportional to \( W \) prior to obtaining estimates for all the involved readings, and 2) the decoding complexity of (49) may grow exceedingly high.

3.3 Progressive CS signal reconstruction with prior information

In this section, a novel sequential CS method is derived. Differently from the KCS that processes the data in non-overlapping blocks (See Remark 1), the proposed method treats the sensor data as continuous-time streams. This is motivated by the fact that at each time instant \( t \geq W \), (49) utilizes the CS measurements and joint compressibility associated with the current sensor reading vector \( \mathbf{x}(t) \), and the \( W-1 \) past ones \( \mathbf{x}(t-W+1), \ldots, \mathbf{x}(t-1) \). The streaming processing mitigates the disadvantages of the KCS by 1) eliminating the 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 extensively illustrated by the empirical results in Section 3.4.

3.3.1 Modified CS reconstruction problem

For window sizes \( W > 1 \), the sensors’ readings \( \mathbf{x}(t) \) reappear in the \( W \) consecutive data windows \( X(t), \ldots, X(t+W-1) \) (see (42) and Fig. 12(a)). Hence, as \( \mathbf{x}(t) \) is involved in the \( W \) successive measurement vectors \( y(t), \ldots, y(t+W-1) \) by (47), it will be decoded \( W \) times via (49). Due to this, a novel recursive CS recovery method is derived by modifying (49) so that, at each time instant \( t \geq W \), it utilizes the previously decoded estimates of \( \mathbf{x}(t-W+1), \ldots, \mathbf{x}(t-1) \) to facilitate the reconstruction of the current sensors’ readings \( \mathbf{x}(t) \). Moreover, this improves the reconstruction accuracy of the past readings \( \mathbf{x}(t-W+1), \ldots, \mathbf{x}(t-1) \). As demonstrated in Section 3.4, the developed method substantially reduces the amount of necessary sensor communications in the considered WSN setup. As another benefit, albeit beyond the scope of this thesis, the
overlapping signal portions can further be utilized to develop iterative warm start based recovery algorithms with fast convergence [105, 134, 135, 138].

Decoder buffer

Let \( X_B(t) \in \mathbb{R}^{N \times (W-1)} \) denote the first \( W-1 \) columns of data window \( X(t) \) (See Fig. 12(a)). Thus, at time instant \( t \geq W \), it consists of the \( W-1 \) previous sensors’ readings, i.e., \( X_B(t) = [\hat{x}(t-W+1) \cdots \hat{x}(t-1)] \). Accordingly, let \( \hat{X}_B^{(t-1)}(t) \in \mathbb{R}^{N \times (W-1)} \) denote a decoder buffer at time instant \( t \), consisting of the estimates of \( X_B(t) \) obtained at the previous instant \( t-1 \), i.e.,

\[
\hat{X}_B^{(t-1)}(t) = \begin{bmatrix} \hat{x}_1^{(t-1)}(t-W+1) & \cdots & \hat{x}_N^{(t-1)}(t-1) \end{bmatrix},
\]

(51)

where \( \hat{x}_i^{(t-1)}(t-d) = [\hat{x}_i^{(t-1)}(t-d) \cdots \hat{x}_i^{(t-1)}(t-1)]^T, \) \( d = 1, \ldots, W-1 \), contains the estimates of \( x_i(t-d) \) obtained at the decoding instant \( t-1 \), i.e., \( \hat{x}_i^{(t-1)}(t-d) \) denotes the estimate of sensor reading \( x_i(t-d) \) of (40) obtained at decoding instant \( t-1 \). Hence, the last column of \( \hat{X}_B^{(t-1)}(t) \) 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 (49) at consecutive time instants \( t-1 \) and \( t \) reconstruct the data windows \( X(t-1) = [\hat{x}(t-W) \ X_B(t)] \) and \( X(t) = [X_B(t) \ \hat{x}(t)] \), respectively. Owing to the fact that they share the common signal part \( X_B(t) \) (see Fig. 12(a)), the recovery problem (49) is modified such that, at each time instant \( t \geq W \), it utilizes the previous estimate of \( X_B(t) \), i.e., the decoder buffer \( \hat{X}_B^{(t-1)}(t) \) in (51), to reconstruct \( X(t) \). To this end, the objective function of (49) will be augmented with a regularization term which induces an additional penalty in relation to \( \| \hat{X}_B^{(t)}(t) - \hat{X}_B^{(t-1)}(t) \|_F \), i.e., the deviation between the estimates of \( X_B(t) \) obtained at the consecutive decoding instants \( t \) and \( t-1 \).

Let \( \Psi_T' \in \mathbb{R}^{(W-1) \times W} \) denote the matrix consisting of the first \( W-1 \) rows of temporal domain basis \( \Psi_T \) in (44), i.e., \( \Psi_T' = [\psi_{T,1} \cdots \psi_{T,(W-1)}]^T \), where \( \psi_{T,j} = [\psi_{T,1j} \cdots \psi_{T,Wj}]^T \) is the \( j \)th row of \( \Psi_T \), \( i = 1, \ldots, W-1 \). Then, a matrix
The differences of the proposed method compared to the "modified CS reconstruction" in [119, 120] are elaborated in Section 1.4.

\( \Psi' \in \mathbb{R}^{N(W-1) \times NW} \) is formed as

\[
\Psi' = \Psi_S \otimes \Psi_S
\]

\[
= [\Psi_1 \cdots \Psi_{N(W-1)}]^T,
\]

which extracts the first \( N(W-1) \) rows of Kronecker sparsifying basis \( \Psi \) in (45), where \( \Psi_i = [\psi_1 \cdots \psi_{(NW)}]^T \) is the \( i \)th row of \( \Psi, i = 1, \ldots, N(W-1) \). Details are provided in Appendix 1.

**Regularization:** By means of \( \Psi' \) in (52) and \( \hat{x}^{(\ell-1)}_B(t) \) in (51), the following regularization term is introduced in the objective function of (49):

\[
\gamma_h \left\| \Psi' \hat{z} - \text{vec} \left( \hat{x}^{(\ell-1)}_B(t) \right) \right\|_2^2
\]

where \( \gamma_h \geq 0 \) is a non-negative regularization weight parameter, and \( \hat{z} \in \mathbb{R}^{NW} \) are the optimization variables. The first term in (53), which, according to (52) is equal to \( \Psi' \hat{z} = [\psi_1 \cdots \psi_{N(W-1)}]^T \hat{z} \), leads to the following logic: by solving (49) with added regularization (53), the term \( \Psi' \hat{z} \) constitutes an estimate of \( \text{vec}(X_B(t)) \) at time instant \( t \), i.e., \( \Psi' \hat{z} \triangleq \text{vec} \left( \hat{x}^{(\ell)}_B(t) \right) \). On this account, the regularization term can be interpreted as \( \gamma_h \left\| \text{vec} \left( \hat{x}^{(\ell)}_B(t) \right) - \text{vec} \left( \hat{x}^{(\ell-1)}_B(t) \right) \right\|_2^2 \), i.e., it assigns an extra cost for the deviation between the consecutive estimates of \( X_B(t) \). Clearly, the weight parameter \( \gamma_h \) controls the emphasis between the regularization term and the sparsity-promoting \( \ell_1 \)-norm term.

**IRW-\( \ell_1 \):** In addition to the \( \ell_2 \)-regularization above, the IRW-\( \ell_1 \) algorithm [103] is adapted to problem (49). Accordingly, at each decoding instant \( t \), the IRW-\( \ell_1 \) alternates between solving a weighted \( \ell_1 \)-minimization problem, and updating the weights based on the obtained solution [103]. Specifically, \( \| \hat{z} \|_1 \) in (49) is replaced with \( \left\| G^{(k)}(t) \hat{z} \right\|_1 \), where \( G^{(k)}(t) = \text{diag} \left( g^{(k)}_1(t), \ldots, g^{(k)}_{NW}(t) \right) \) is a diagonal weight matrix at decoding instant \( t \) and iteration \( k \) with positive weights \( g^{(k)}_i(t) > 0, i = 1, \ldots, NW \).

By combining the IRW-\( \ell_1 \) with the \( \ell_2 \)-regularization term (53), at each decoding instant \( t \geq W \) and iteration \( k \), the decoder, which assumes temporal (piecewise) smoothness in \( X(t) \), first solves a modified CS recovery problem\(^{16}\)

\[
\hat{x}^{(k)}(t) := \arg \min_{\hat{z}} \left\{ \left\| G^{(k)}(t) \hat{z} \right\|_1 + \gamma_h \left\| \Psi' \hat{z} - \hat{x}^{(\ell-1)}_B(t) \right\|_2^2 \right\} \quad \text{such that} \quad y(t) = \Phi(t) \Psi' \hat{z}
\]

\(^{16}\)The differences of the proposed method compared to the "modified CS reconstruction" in [119, 120] are elaborated in Section 1.4.
to obtain an estimate of \( z(t) \) (See (45)), where \( \hat{x}_{B}^{(t-1)}(t) = \text{vec} \left( \hat{X}_{B}^{(t-1)}(t) \right) \) is the vector-reshaped decoder buffer of (51). Then, the decoder uses the obtained estimate vector \( \hat{z}^{(k)}(t) = \left[ \hat{z}_{1}^{(k)}(t) \cdots \hat{z}_{NW}^{(k)}(t) \right]^T \) to update the weights as [103]

\[
G_{i}^{(k+1)}(t) := \left( \left| \hat{z}_{i}^{(k)}(t) \right| + \epsilon_{0} \right)^{-1}, \ i = 1, \ldots, NW,
\]

where \( \epsilon_{0} > 0 \) is a small positive stability parameter. In practice, the two iteration steps (54) and (55) of the IRW-\( \ell_{1} \) are alternated until either the weights have converged (e.g., when \( \| G^{(k+1)}(t) - G^{(k)}(t) \|_{F} \leq \epsilon_{G} \) with a pre-defined tolerance \( \epsilon_{G} > 0 \)), or until a maximum number of allowed iterations \( k_{\text{max}} \) is reached. Typically, most of the gain is achieved in the first few iterations of the IRW-\( \ell_{1} \) [103].

Prior to describing the complete algorithm in Section 3.3.2, the two modifications – the \( \ell_{2} \)-regularization term (53) and the IRW-\( \ell_{1} \) – which differentiate problem (54) from (49) are separately elaborated. An effective weight initialization strategy for the IRW-\( \ell_{1} \) is also proposed. Note that for \( G^{(k)}(t) = I_{NW} \) and \( \gamma_{B} = 0 \), the problem (54) is equivalent to (49).

**The \( \ell_{2} \)-regularization term**

Recall that \( \gamma_{B} \left\| \Psi \hat{z} - \text{vec} \left( \hat{X}_{B}^{(t-1)}(t) \right) \right\|_{2} \) adds regularization to the variations in the estimates \( \hat{X}_{B}^{(i)}(t) \) and \( \hat{X}_{B}^{(t-1)}(t) \) 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 \( \hat{X}_{B}^{(i)}(t) \) compared to the previous one \( \hat{X}_{B}^{(t-1)}(t) \). Owing to the characteristics of the \( \ell_{2} \)-norm, the regularization assigns very small weights to small residuals in \( \left\| \text{vec} \left( \hat{X}_{B}^{(i)}(t) \right) - \text{vec} \left( \hat{X}_{B}^{(t-1)}(t) \right) \right\|_{2} \), i.e., in \( \left( \sum_{i=t-W+1}^{t} \hat{z}_{i}^{(t)}(\tau) - \hat{z}_{i}^{(t-1)}(\tau) \right)^2 \) [9, Ch. 6]. Correspondingly, it has a low incentive to make small deviations even smaller, and, thus, the \( \ell_{2} \)-regularization allows \( \Psi \hat{z} = \hat{x}_{B}^{(i)}(t) \) to slightly differ from the previous estimate \( \hat{x}_{B}^{(t-1)}(t) \). Thus, besides facilitating the recovery of the current sensors’ readings \( \hat{x}(t) \), the \( \ell_{2} \)-regularization allows the algorithm to refine the previous estimates \( \hat{x}(t-W+1), \ldots, \hat{x}(t-1) \) through the sliding window processing. This is numerically demonstrated in Section 3.4.
The IRW-ℓ₁

The use of a weighted ℓ₁-norm is motivated by the imbalance of the ℓ₀- and ℓ₁-penalty: while the ℓ₀-term sets an equal penalty for each non-zero coefficient, the ℓ₁-norm penalizes them linearly in proportion to their magnitudes [103]. Therefore, although the ℓ₁-minimization most likely identifies the locations of large entries in \( z(t) \), the magnitudes – especially of small elements – may remain inaccurate. As a remedy, the reweighting (55) compensates for the magnitude dependency of the ℓ₁-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 (54) become roughly equalized as

\[
|g_i^{(k)}(t)\tilde{z}_i| \approx 1, \quad i = 1, \ldots, NW.
\]

Hence, in the course of iterations, the weighted ℓ₁-minimization starts to resemble the ℓ₀-minimization, thereby improving the CS decoding accuracy [103].

Weight initialization: Naturally, the CS recovery performance of the recursive IRW-ℓ₁ depends on the starting point, since all successive iterations for \( k \geq 2 \) implicitly rely on the initial solution [103]. An intuitive strategy also found to work well in practice is to use \( G^{(1)}(t) = I_{NW} \) (i.e., the ℓ₁-minimization) [103]. With no prior signal information, this is a reasonable choice, as the ℓ₁-minimization has the best theoretically established recovery threshold amongst polynomial-complexity decoding algorithms for sparse signals [145]. However, the decoding problems (54) at consecutive instants \( t - 1 \) and \( t \) reconstruct the estimates for \( X(t-1) = [\bar{x}(t-W) \ X_B(t)] \) and \( X(t) = [X_B(t) \ \bar{x}(t)] \), respectively. Thus, because the data windows share the overlapping block of temporally (and spatially) correlated sensors’ readings \( X_B(t) \), the corresponding joint transform coefficients \( z(t-1) = \Psi^{-1} \text{vec}(X(t-1)) \) and \( z(t) = \Psi^{-1} \text{vec}(X(t)) \) can respectively be expected to be close to each other. This supplementary information can be used in the weight initialization to improve the reconstruction of the sensor data streams, discussed next.

The following weight initialization strategy for the IRW-ℓ₁ is proposed. The weights at decoding instant \( t > W \) are initialized with the final weights used in the previous instant \( t - 1 \). Let \( k_\ell \) denote the last iteration \( k \) taken for the IRW-ℓ₁ at decoding instant \( t \). Accordingly, \( z^{(k)}(t) = [\tilde{z}_1^{(k)}(t) \cdots \tilde{z}_{NW}^{(k)}(t)]^T \) are the final variables obtained for (54) at decoding instant \( t \); \( G^{(k)}(t) \) is the corresponding weight matrix. Hence, instead of

\[
G^{(k)}(t) \rightarrow G^{(k)}(t-1) \quad \text{if } t - 1 \leq W.
\]

As an alternative, the authors of [134] 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.
confining to the $\ell_1$-minimization ($G^{(1)}(t) := I_{NW}$) in the first iteration, the weight matrix is set as $G^{(1)}(t) := G^{(k_t-1)}(t-1)$ at each $t > W$, i.e., the initial weights become

$$g^{(1)}_i(t) := \left(\left|z^{(k_t-1)}_i(t-1) + \epsilon_0\right|^{-1}, \; i = 1, \ldots, NW. \right. \tag{56}$$

The compression gains brought by this weight initialization are numerically exemplified in Section 3.4.

### 3.3.2 Algorithm summary

The proposed sequential compressed data acquisition method with a progressive CS signal recovery technique based on the $\ell_2$-norm regularization and the IRW-$\ell_1$ (termed Seq-Prog-CS) is described in Algorithm 3. The main operations are summarized as follows. At each time slot $t \geq W$, the sink gathers the CS measurements (46) by acquiring the readings from a subset of the sensors (Step I.). Then, the decoder solves the modified CS recovery problem (54) via the IRW-$\ell_1$, resulting in an estimate of $X(t)$ as $\hat{X}^{(t)}(t) = \begin{bmatrix} \hat{x}^{(t)}(t-W+1) & \cdots & \hat{x}^{(t)}(t) \end{bmatrix}$ (Step II.). Once reconstructed, its $W-1$ last columns $\begin{bmatrix} \hat{x}^{(t)}(t-W+2) & \cdots & \hat{x}^{(t)}(t) \end{bmatrix}$ are used to form the decoder buffer $\hat{X}^{(t)}_B(t+1)$ in (51) for the next instant $t+1$ (Step III.). Then, $\hat{X}^{(t)}(t)$ is used to update the estimates of $x(t-W+1), \ldots, x(t)$ as $\{\hat{x}(t-W+1), \ldots, \hat{x}(t)\} \leftarrow \{\hat{x}^{(t)}(t-W+1) \cdots \hat{x}^{(t)}(t)\}$, 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 $\hat{x}(1), \ldots, \hat{x}(W-1)$ are required at the decoder to form the decoder buffer $\hat{X}^{(t-1)}_B(t)$ and initialize the IRW-$\ell_1$. These can be attained for example via (49). As there is no prior signal information available, it may be necessary to use more CS measurements compared to the subsequent steps $t \geq W$ [119]. In general, due to the recursive decoding, all preceding estimates from time slots $\{\ldots, t-2, t-1\}$ implicitly affect the recovery performance at a time slot $t$. Thus, inaccurate consecutive estimates may induce error propagation. The stability over time can be improved via intermittently requesting more measurements from the sensors (up to $M(t) = N$) to improve the estimation accuracy, and consequently, to provide more reliable prior information for the sliding window processing.
Algorithm 3 Sequential compressed data acquisition with progressive CS signal recovery (Seq-Prog-CS)

Parameters: $M_t$, $W$, $k_{\text{max}}$, $\gamma_B$, $\varepsilon_0$ and $\varepsilon_G$.

Initializations: (i) Set $t = W$; (ii) obtain $[\hat{x}(1) \cdots \hat{x}(t-1)]$ to form $\hat{X}_{\text{B}}^{(t-1)}(t)$; (iii) set $\kappa_{t-1} = 1$ and $G^{(1)}(t-1) = I_{NW}$.

I. CS measurements

Deliver the CS measurements $\mathbf{y}(t) = \Phi \mathbf{x}(t)$ in (46) to the sink.

II. Progressive signal reconstruction

a) Construct $\mathbf{y}(t) = [\mathbf{y}^T(t-W+1) \cdots \mathbf{y}^T(t)]^T$.

b) Set $k = 1$, and initialize $G^{(k)}(t)$ according to (56).

c) Run the IRW-$\ell_1$ to obtain $\hat{z}^{(k)}(\kappa_t)(t)$ and $G^{(k)}(\kappa_t)(t)$:

repeat
1) Solve (54) to obtain $\hat{z}^{(k)}(t) = [\hat{z}_1^{(k)}(t) \cdots \hat{z}_{NW}^{(k)}(t)]^T$.

2) Set weight matrix $G^{(k+1)}(t)$ according to (55).

3) Set $\kappa_t = k$ and $k = k + 1$.

until $\left\|G^{(k)}(t) - G^{(k-1)}(t)\right\|_F / \left\|G^{(k-1)}(t)\right\|_F \leq \varepsilon_G$ or $k > k_{\text{max}}$.

d) Reconstruct the estimate for $X(t)$ as $\hat{X}^{(l)}(t) = \Psi \hat{z}^{(k_{\text{max}})}(t)$, and reshape it as

$\hat{X}^{(l)}(t) = [\hat{x}^{(l)}(t-W+1) \cdots \hat{x}^{(l)}(t)]$.

III. Decoder buffer update

Set the decoder buffer in (51) as $\hat{X}_B^{(l)}(t+1) = [\hat{x}^{(l)}(t-W+2) \cdots \hat{x}^{(l)}(t)]$.

IV. Estimate update

Set $\{\hat{x}(t-W+1), \ldots, \hat{x}(t)\} \leftarrow \{\hat{x}^{(l)}(t-W+1) \cdots \hat{x}^{(l)}(t)\}$.

V. Sliding window advance step

Set $t = t + 1$, and go to Step I.

3.4 Simulation results

Numerical experiments are presented to illustrate the performance of the proposed Seq-Prog-CS method in terms of achievable energy savings and signal reconstruction accuracy as compared to several state-of-the-art CS methods.
3.4.1 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 \( X = [\mathbf{x}(1) \cdots \mathbf{x}(T)] \in \mathbb{R}^{N \times T} \). 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 \( 100 \times 100 \) square is randomly deployed with one sensor according to a uniform distribution. The sensors route their data to the sink, located at the center of a WSN, through a shortest path tree. Two sensors \( n \) and \( j \) are able to communicate if \( d_{n,j} \leq 100\sqrt{5} \), \( n, j \in \mathcal{N} \), \( n \neq j \), which guarantees a connected WSN\(^{18}\).

Generation of WSN data

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

**Spatial correlation:** A power exponential correlation model [252–254] is used to model the influence functions in (40): \( h_{ns}(d_{ns}) = \exp\left\{ -(d_{ns}/\rho_1)^{\rho_2} \right\} \), where parameters \( \rho_1 > 0 \) and \( \rho_2 \in [0, 2] \) adjust the correlation decay rate and geometrical properties of the field, respectively [253].

**Temporal correlation:** Temporally correlated, piecewise smooth source sequences \( \beta_s \in \mathbb{R}^T, s \in \mathcal{S} \), in (40), are generated as follows.

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

\[
\lambda_s(t) = \alpha_s [\lambda_s(t-1) - \mu_s] + (1 - \alpha_s) \varsigma_s(t) + \mu_s, \quad t = 2, \ldots, T, \tag{57}
\]

where \( \alpha_s \in [0, 1] \) is a correlation parameter, \( \varsigma_s(t) \sim \mathcal{N}(0, \sigma_{2}^2) \) is the innovation component, and \( \mu_s \) is the mean component [271–273]. Thus, for \( \alpha_s = 1 \), \( \lambda_s \) remains constant, whereas for \( \alpha_s = 0 \), it behaves as an uncorrelated Gaussian random process.

In order to create smoothly evolving sequences, each \( \lambda_s \) is low-pass filtered by applying the \( T \)-point DFT as \( \tilde{f}_{dft,s} = F_{dft} \lambda_s \), where \( F_{dft} \in \mathbb{C}^{T \times T} \) is the DFT-matrix, and \( f_{dft,s} \in \mathbb{C}^T \) is the vector of the DFT-coefficients. Each \( f_{dft,s} \) is used to form a vector \( \tilde{f}_{dft,s} \in \mathbb{C}^T \) by preserving \( \eta_{lpf} \in [1, T/2] \) lowest frequency components of \( f_{dft,s} \), while setting the rest to zero. Finally, taking the \( T \)-point inverse DFT, a low-pass version of

\(^{18}\)The maximum distance between any pair of sensors residing in the vertically or horizontally adjacent grids is \( 100\sqrt{5} \).
whose number of unpredictable changes by \( P \) via the used influence functions \( h \). The degree of temporal correlation are adjusted by \( \eta \). Similarly to (45), the generated source magnitude sequences \( \beta_s, s \in \mathcal{S} \), the sensor data streams of (40) are only compressible (i.e., not exactly sparse) in both the spatial and temporal domain.

### Spatial domain transform

2D-DCT is applied to \( X \) to obtain a compressible spatial domain representation of each \( X(t), t = 1, \ldots, T \). Firstly, each \( X(t) \) is reorganized into a matrix \( \mathbf{X}(t) \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}} \), 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 (45), \( \mathbf{X}(t) \) can be expressed as

\[
\mathbf{X}(t) = F_{\text{det}} \mathbf{\Theta}_S(t) F_{\text{det}}^T, \quad \text{i.e.,} \quad \mathbf{X}(t) = \text{vec} (\mathbf{X}(t)) = \Psi_S \mathbf{\theta}_S(t),
\]

where \( F_{\text{det}} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}} \) is the inverse of a DCT-matrix, \( \mathbf{\Theta}_S(t) \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}} \) contains the DCT-coefficients in a matrix form, \( \Psi_S = (F_{\text{det}}^{-1} \otimes F_{\text{det}}^{-1}) \) is the (Kronecker) sparsifying basis for the spatial domain, and \( \mathbf{\theta}_S(t) = \text{vec} (\mathbf{\Theta}_S(t)) \in \mathbb{R}^N \).

### Performance metrics

The performance of the proposed Seq-Prog-CS method is evaluated in respect to the data transportation costs and signal reconstruction accuracy. The total cost of delivering
the necessary amount of data to the sink to reconstruct $X$ is measured via a normalized communication cost \[ C = \frac{\sum_{t=1}^{T} \sum_{n \in N} E_n I_n(t)}{\sum_{t=1}^{T} \sum_{n \in N} E_n}. \] (59)

where $E_n$ denotes the number of hops from sensor $n \in N$ to the sink, and $I_n(t)$ is an indicator function with $I_n(t) = 1$, if $n \in N(t)$, 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 carrying out 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\[19\].

For a given $C$, the CS reconstruction error is measured as

\[ \xi_{cs} = \frac{\| \text{vec}(\hat{X}) - \text{vec}(X) \|_2}{\| \text{vec}(X) \|_2}, \] (60)

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

**CS methods**

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

1. KCS (Kron-CS) that splits $X$ into $B_{kcs}$ consecutive non-overlapping blocks\[20\] as $X = [X[1] \cdots X[B_{kcs}]]$, where $X[l] \in \mathbb{R}^{N \times (T/B_{kcs})}$, $l = 1, \ldots, B_{kcs}$, and reconstructs them separately via (49) with $W := T/B_{kcs}$.
2. Spatial CS (Spat-CS) that reconstructs each $x(t)$, $t = 1, \ldots, T$, separately via (50).
3. A dynamic version of the regularized modified-CS (Reg-Mod-CS) proposed in [119], which utilizes the support and estimate knowledge recursively from the preceding decoding instant.

For the ease of comparison, the same randomly generated $\Phi(t)$ with $M(t) = M$ is used for each CS method at each time instant $t$, guaranteeing equal $C$ in (59) for all methods. For given $N$ and $W$, empirically tuned, fixed regularization weight parameters

---

\[19\] 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.

\[20\] In order to restrain the decoding complexity in the KCS, the reconstruction of $X$ is partitioned into smaller blocks.
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 (49). To illustrate the gradual estimate refinement, the Seq-Prog-CS method is reported in respect to 1) the first estimates obtained for each x(t), and 2) the estimates after decoded W times, which are marked with the superscripts (·)(t) and (·)(t+W−1), respectively. The Seq-Prog-CS(t) is also included without the ℓ2-regularization and the IRW-ℓ1 (Seq-CS), which reconstructs each X(t) non-recursively (yet with streaming processing) via (49).

For the IRW-ℓ1, the parameters are set as k_{max} = 5, \epsilon_0 = 1 \times 10^{-1}, and \epsilon_G = 1 \times 10^{-3}. For Reg-Mod-CS, b = 99.5 is used in the support detection [119, Sect. V].

The optimization problems are solved using ℓ1-MAGIC [275] and CVX [251]. The DWT-matrices are generated with the Wavelab toolbox [276].

3.4.2 Performance of the proposed method

Influence of window size

The influence of window size W on the CS recovery performance of the proposed method is investigated in the setup with N = 16, T = 512, and B_{kcs} = 8. The spatial correlation is set by p_1 = 1 \times 10^2 and p_2 = 2, and the temporal by P_0 = 1.0 (smooth signals). The DCT-matrix is used for \Psi T. In order to strictly confine the results to the impact of W, Seq-CS was run instead of Seq-Prog-CS.

Fig. 13 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 the temporal domain compressibility, its performance is inferior to all other methods. It can be observed that also utilizing the compressibility in the temporal domain significantly reduces the necessary sensor communications: for instance, the Seq-CS method with W = 16 resulted in a reconstruction accuracy of around ξ_{cs} = 0.04 while requiring only a half of the sensors’ readings to be communicated to the sink. The figure also illustrates the trade-off between the decoding complexity and the CS recovery performance for Seq-CS: as 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-Próg-CS method for various network sizes, consider setups with $N = 9$, $N = 16$, and $N = 25$. Other simulation parameters are set as $T = 256$, $B_{\text{kcs}} = 4$, $\rho_1 = 5 \times 10^2$, $\rho_2 = 2$, and $P_0 = 0.97$. $\Psi_T^{-1}$ is set as the Daubechies-4 DWT-matrix. The resulting average CS recovery error $\xi_{\text{cs}}$ versus communication cost $C$ for each CS method is depicted in Fig. 14.

Similarly as shown in Fig. 13, by exploiting the spatio-temporal correlation in Seq-CS, Seq-Próg-CS, Reg-Mod-CS, and Kron-CS, the methods significantly reduce the sensors’ communication costs compared to the Spat-CS method. The benefits of utilizing prior information in the CS decoding are clearly visible: the Seq-Próg-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 $C$). Clear improvements are also achieved by the other recursive CS method, Reg-Mod-CS, although it has lower accuracy than Seq-Próg-CS for all $N$. Interestingly, the
Fig. 14. The CS recovery 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 \(B_{\text{cs}} = 4\) for (a) \(N = 9\), (b) \(N = 16\), and (c) \(N = 25\) ([148]©2015 IEEE).
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. Moreover, compared to Kron-CS, the decoding complexity of the Seq-Prog-CS method is lower by a factor of \(W/(T/B_{kcs}) = \{1/16, 1/8, 1/4\}\) for \(W = \{4, 8, 16\}\), respectively. Furthermore, as demonstrated by the Seq-Prog-CS\(^{(t)+W-1}\) method, Seq-Prog-CS also considerably improves the reconstruction accuracy of the past sensors’ readings via the progressive decoding.

**Influence of prior information**

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 \(N = 16\), \(T = 256\), \(B_{kcs} = 4\), and \(W = 8\). Particularly, the influences of the IRW-\(\ell_1\) (i.e., \(k_{\text{max}} > 1\)), the \(\ell_2\)-regularization (i.e., \(\gamma_B > 0\)), and the weight initialization (\(G_{\text{init}}\)) of (56) (i.e., \(G^{(1)}(t) := G^{(\kappa+1)}(t-1)\) as opposed to \(G^{(1)}(t) := I_{NW}\)) are studied by running the following six variants of Seq-Prog-CS:

- Seq-CS, which excludes both the IRW-\(\ell_1\) and \(\ell_2\)-reg
- The proposed method in Algorithm 3, which incorporates all three forms of prior knowledge, termed Seq-Prog-CS [IRW-\(\ell_1+\ell_2\)-reg+\(G_{\text{init}}\)] (\(\gamma_B = 10\))
- Seq-Prog-CS [IRW-\(\ell_1\)]
- Seq-Prog-CS [IRW-\(\ell_1+G_{\text{init}}\)]
- Seq-Prog-CS [\(\ell_2\)-reg] (\(\gamma_B = 3\))
- Seq-Prog-CS [IRW-\(\ell_1+\ell_2\)-reg] (\(\gamma_B = 10\))

The correlation is set by \(\rho_1 = 5 \times 10^2\), \(\rho_2 = 2\), and \(\rho_0 = 0.97\), and the Daubechies-8 DWT-matrix is used for \(\Psi^{-1}\).

Figs. 15(a) and (b) depict the resulting average CS recovery error \(\xi_{\text{cs}}\) versus the communication cost \(C\) for the above methods reported against the Seq-Prog-CS\(^{(t)}\) and Seq-Prog-CS\(^{(t)+W-1}\) methods, respectively. The figure highlights the significance of an adequate starting point for the IRW-\(\ell_1\); using the \(\ell_1\)-minimization as the initial step, [IRW-\(\ell_1\)] can only slightly improve the CS performance from that of the Seq-CS method. However, with the proposed weight initialization (56), Seq-Prog-CS\(^{(t)}\)[IRW-\(\ell_1+G_{\text{init}}\)] outperforms both [\(\ell_2\)-reg] and [IRW-\(\ell_1+\ell_2\)-reg]. Because the proposed method with
Fig. 15. The CS recovery performance with different types of prior signal information used in the signal reconstruction with $N = 16$, $T = 256$, $B_{cs} = 4$, and $W = 8$ in terms of (a) Seq-Prog-CS$^0$, and (b) Seq-Prog-CS$^{(W-1)}$ ([148] © 2015 IEEE).

[IRW-$\ell_1+\ell_2$-reg+$G_{min}$] incorporates all three forms of prior signal knowledge, it indisputably yields the best CS reconstruction accuracy of all the considered methods.
Table 3. CS recovery performance ($\xi_{cs} \times 100$ [%]) with different parameters for spatial ($\rho_1$) and temporal ($P_0, \alpha_s$) correlation with $N = 16$, $T = 256$, $B_{kcs} = 4$, $W = 8$, and $M = 6$ ([148] © 2015 IEEE).

<table>
<thead>
<tr>
<th>$\rho_1$</th>
<th>$P_0$</th>
<th>$\alpha_s$</th>
<th>$\xi_{cs}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$5 \times 10^2$</td>
<td>$1 \times 10^2$</td>
<td>$1 \times 10^4$</td>
</tr>
<tr>
<td>Spat-CS</td>
<td>1.7</td>
<td>1.46</td>
<td>0.392</td>
</tr>
<tr>
<td>Seq-CS</td>
<td>7.2</td>
<td>6.72</td>
<td>4.12</td>
</tr>
<tr>
<td>Seq-Prog-CS$_{\psi}$</td>
<td>11.1</td>
<td>11.0</td>
<td>6.06</td>
</tr>
<tr>
<td>Seq-Prog-CS$_{\psi}$</td>
<td>11.1</td>
<td>11.1</td>
<td>6.05</td>
</tr>
<tr>
<td>Kron-CS</td>
<td>11.2</td>
<td>11.1</td>
<td>6.51</td>
</tr>
</tbody>
</table>

Influence of spatial and temporal correlation

The CS methods were also tested under different signal characteristics by varying the parameters adjusting the spatial ($\rho_1$) and temporal ($P_0$ and $\alpha_s$) correlation properties of the sensor data streams. The case with $N = 16$, $T = 256$, $B_{kcs} = 4$, and $W = 8$ with a fixed number of measurements $M = 6$ was considered. Simulations were run for $\rho_2 = 2$ using the Daubechies-4 DWT-matrix as $\Psi^{-1}$ for $P_0 < 1.0$, and the DCT-matrix for $P_0 = 1.0$.

Table 3 reports the average error $\xi_{cs}$ for the CS methods under various signal statistics. Obviously, as the temporal correlation worsens, i.e., the parameters $P_0$ or/and $\alpha_s$ 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, as in most of the cases, it clearly outperforms all the other methods while producing estimates with an accuracy of $\xi_{cs} < 0.07$. The only exception is Kron-CS with highly correlated smooth signals (the first column), where Kron-CS logically makes the most of processing large batches of size $T/B_{kcs} = 64$. As a final remark, no method was capable of producing accurate estimates for signals with very poor spatial correlation (the last column).

3.5 Summary and discussion

This chapter addressed the framework of distributed compressed acquisition and progressive reconstruction of spatially and temporally correlated sensor data streams in multi-hop WSNs. A novel sequential CS method relying on sliding window processing
summarized in Algorithm 3 was proposed. By means of the Kronecker sparsifying basis, $\ell_2$-regularization, and adaption of the IRW-$\ell_1$, 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 the state-of-the-art CS methods.

Owing to the high compression performance demonstrated by the numerical experiments, the proposed method has great potential to prolong the lifetime of battery-powered sensors in various monitoring applications. Furthermore, the ability to make a trade-off between the performance and complexity via the window size makes the algorithm versatile for applications with different requirements for 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 knowledge of the correlation structure and the implementation of a computationally demanding decoding algorithm is shifted to the sink/fusion center.

Besides environmental monitoring, the principles of the derived CS method can also be applied to another data acquisition applications that encompass a certain level of dynamic signal characteristics. For instance, multimedia applications such as compressed video encoding of sequences with inter-frame correlation could benefit from the proposed sequential processing. In summary, the proposed method is a competent candidate for streaming type compression applications when striving for high signal reconstruction accuracy in a cost-effective manner.
This chapter addresses lossy distributed source coding (DSC) for acquiring correlated sparse sources using CS in WSNs. Accordingly, noisy CS 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 Chapter 3 is that the framework now involves the quantization of CS measurements, which is a practical necessity for finite-rate data communications/storage. The outcome is a distributed complexity-constrained variable-rate quantized CS (QCS) method that minimizes a weighted sum between the mean square error (MSE) signal reconstruction distortion and the average encoding rate.

A focal target is to restrain the encoding complexity of each sensor while achieving a decent distortion-rate (DR) performance. This is realized by pre-quantizing the encoder inputs, i.e., the CS measurements, via vector quantization (VQ) at each sensor. Differently from the prior works [222, 223], 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. In answer to this, techniques to further lower the encoding complexity will be discussed.

Each encoder is modeled as a quantizer followed by a lossless entropy encoder, and variable-rate coding is incorporated via rate measures of an entropy bound. The model is confined to a two-sensor system in which the necessary optimality conditions are derived. Furthermore, practical training algorithms are proposed, and the complexities of the training and communication phases are analyzed. Numerical results illustrate that the proposed method achieves superior compression performance compared to baseline methods and lends itself to versatile setups with different performance requirements.

The chapter is organized as follows. The system model is defined in Section 4.1. In Section 4.2, a novel distributed variable-rate QCS method is developed, practical training algorithms are proposed, and a complexity analysis is provided. Numerical results are presented in Section 4.3. Section 4.4 summarizes the chapter.
4.1 System model

Consider a distributed QCS system consisting of two CS based sensors and one sink, as depicted in Fig. 16. In the DSC approach, each sensor acquires noisy CS 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. Because the main focus is on source coding, the transmissions from each sensor to the sink are assumed to be error-free.

It is worth pointing out 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. Nevertheless, the main features of the considered scheme are more lucidly highlighted in a two-sensor setup while avoiding cumbersome derivations. Naturally, it is straightforward to apply the design for a single-sensor case.

4.1.1 CS signal acquisition

The correlated sources $X_1$ and $X_2$ are given by JSM-2 model [57, 131] as

$$X_l = \tilde{X} + X'_l, \ l = 1, 2,$$

(61)
where the common component $\mathbf{X}$ 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 $\mathbf{X}$, $\mathbf{X}'_1$, and $\mathbf{X}'_2$ are assumed to be independent of each other. For example, JSM-2 signals are encountered in a group of sensors monitoring an audio source or spectrum occupancy [57]. 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 [277–279].

Let $\mathcal{T}_s \subseteq \{1, \ldots, N\}$ be an index set representing the $s$th 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 $\Phi_l \in \mathbb{R}^{M_l \times N}$ as

$$\mathbf{Y}_l = \Phi_l \mathbf{X}_l + \mathbf{W}_l, \ l = 1, 2,$$

where $\mathbf{Y}_l$ is the length-$M_l$ measurement random vector, and $\mathbf{W}_l \sim \mathcal{N}(0, \sigma^2 W I_{M_l})$ is the measurement noise random vector. Whereas $K \leq M_l \leq N$ is assumed for a conventional CS setup, the 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, over-sampling is a practical – and often cost-effective – means to improve the reconstruction accuracy [215].

It is worth emphasizing that because (62) 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 [172].

The structure of each $\Phi_l$, $l = 1, 2$, has a significant impact on the CS signal recovery performance; see Section 1.3.2 for the discussions on the RIP and the coherence of $\Phi_l$. Nonetheless, no restrictive assumptions of $\Phi_1$ and $\Phi_2$ are needed in the derivations.
4.1.2 Measurement space pre-quantization

Prior to the actual source encoding, the input of the encoding system at each sensor $l = 1, 2$, i.e., the measurement random vector $Y_l$ of (62), is discretized with a VQ. Each pre-quantizer$^{21}$ $PQ_l$ is a focal block in restraining the encoding complexity of a sensor while providing high compression performance. As a by-product, this simplifies the optimization design by converting the optimization over continuous random variables into optimization over discrete ones. It thus facilitates offline training by allowing pre-computation of required input quantities. A general description of each pre-quantizer is given next, whereas their specific optimization is deferred until Section 4.2.2.

Let $V_l \triangleq \{1, \ldots, |V_l|\}$ be a set of cell indices $v_l \in V_l$ with pre-quantization rate $\bar{R}_l = \log_2|V_l|$ bits/vector $Y_l$ for sensor $l = 1, 2$. Let $G_l \triangleq \{g_{l,1}, \ldots, g_{l,|V_l|}\}$ be a pre-quantization codebook consisting of codepoints $g_{l,v} \in \mathbb{R}^{M_l}$. Each pre-quantizer $PQ_l$ is a $|V_l|$-level VQ that partitions the $M_l$-dimensional measurement vector space determined by (62) into cells $\mathcal{S}_{l,v_l}$, i.e., $\mathcal{S}_{l,v_l} \cap \mathcal{S}_{l,v'_l} = \emptyset$, $v_l \neq v'_l \in V_l$, and $\bigcup_{v_l=1}^{|V_l|} \mathcal{S}_{l,v_l} = \mathbb{R}^{M_l}$. Thus, $PQ_l$ is a lossy mapping

$$PQ_l : \mathbb{R}^{M_l} \rightarrow V_l, \ l = 1, 2,$$

i.e., for a given measurement realization, it assigns a cell index as $PQ_l(y_l) = v_l \in V_l$, if $y_l \in \mathcal{S}_{l,v_l}$.

Remark 2. The codepoints $g_{l,1}, \ldots, g_{l,|V_l|}$ are intermediate quantities for the actual encoding at $E_l$, i.e., they are solely used to determine to which cell $\mathcal{S}_{l,v_l}$ each realization $y_l$, $l = 1, 2$, belongs. This classification is made explicit by the encoding rule defined in Section 4.2.2. To summarize this: the quantized version of random vector $Y_l$ is never reconstructed in the system.

4.1.3 Encoding and decoding

The outputs of each $PQ_l$, i.e., the cell indices $v_l \in V_l$, are fed to encoder $E_l$ at each sensor $l = 1, 2$. Following a customary approach, $E_l$ is modelled as the concatenation of a (lossy) quantizer and a lossless entropy encoder [183, 184]. Accordingly, each sensor encodes the cell indices into message indices, and further, into binary source codewords.

$^{21}$Pre-quantization has also a pragmatic aspect: prior to source encoding, the sensor inputs are necessarily discretized with an analog-to-digital converter in any digital sensor device.
The sink uses the received pairs of codewords to jointly reconstruct estimates of $X_1$ and $X_2$. The next part of this chapter describes distributed quantizer blocks; the treatment of entropy coding is elaborated in Section 4.1.4.

**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 $\mathcal{H}_l \triangleq \{h_{l,1}, \ldots, h_{l,|\mathcal{H}_l|}\}$, $|\mathcal{H}_l| = |\mathcal{I}_l|$, be a source codebook consisting of binary codewords. Each encoder is a composite mapping $E_l : \alpha_l \circ \pi_l$ as

$$E_l : \mathcal{I}_l \rightarrow \mathcal{H}_l, \quad l = 1, 2. \quad (64)$$

While the number of pre-quantization cells $|\mathcal{I}_l|$ 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 $\pi_l$, 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 $|\mathcal{I}_l| < |\mathcal{V}_l|$, $\pi_l$ 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, as each sensor has a concatenation of two quantizers: the VQ of PQ and the message index mapping $\pi_l$ in $E_l$. Interconnections between the indices of PQ and $E_l$ are illustrated in Fig. 17. Note that the pre-quantization allows assigning non-contiguous cells to the same message index, conducive to compression performance.

Given an 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 $h_{l,i_l} \in \mathcal{H}_l$, i.e., $\alpha_l(i_l) = h_{l,i_l}, \quad i_l \in \mathcal{I}_l$. 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 $h_{l,i_l} \in \mathcal{H}_l$ have, in general, different lengths.

**Joint decoder**

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

$$D : (\mathcal{H}_1 \times \mathcal{H}_2) \rightarrow (\mathcal{I}_1 \times \mathcal{I}_2) \rightarrow \mathcal{C}_l, \quad l = 1, 2. \quad (65)$$
The first mappings $\alpha_l^{-1}: \mathcal{I}_l \to \mathcal{F}_l$, $l = 1, 2$, decode the entropy codes, i.e., each received codeword $h_{li} \in \mathcal{H}_l$ is used to recover the (transmitted) message index $i_l \in \mathcal{I}_l$. Through this procedure, the approach is confined to the separate/independent coding 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 $X_1$ and $X_2$ as $\hat{s}_1 := \beta_1(i_1, i_2) = c_{1_i_1, i_2}$ and $\hat{s}_2 := \beta_2(i_1, i_2) = c_{2_i_1, i_2}$, where $c_{1_i_1, i_2} \in \mathbb{R}^N$ is the codevector of a reconstruction codebook $\mathcal{C}_l \triangleq \{c_{1_i_1, i_2} \mid i_1, i_2 \in \mathcal{I}_1, \mathcal{I}_2\}$, $|\mathcal{C}_l| = |\mathcal{I}_1||\mathcal{I}_2|$, $l = 1, 2$. Thus, the decoder $D$ performs a single operation to 1) take account of the quantization/encoding steps applied to $Y_1$ and $Y_2$, and 2) reconstruct the signal estimates of $X_1$ and $X_2$, given the underlying distributed CS setup.

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.
4.1.4 Entropy coding

The pairs \((\alpha_1, \alpha_1^{-1})\) and \((\alpha_2, \alpha_2^{-1})\) can realize different entropy coding classes in the system. In practice, the choice for the used source code may depend 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 introduced next.

Average rate

Let \(R_l\) be the average encoding rate in bits/vector \(Y_l\) for sensor \(l = 1, 2\). Thus, the average sum rate of the sensors is

\[
R \triangleq R_1 + R_2. \tag{66}
\]

In practice, \(R_l\) is determined by the average codeword length of the codebook \(H_l\), i.e.,

\[
R_l \triangleq \mathbb{E}[\gamma(\cdot)] = \sum_{i_l=1}^{\vert I_l \vert} p(i_l) \gamma(i_l), \quad l = 1, 2, \tag{67}
\]

where \(\gamma(i_l)\) is the length of codeword \(h_{1,i_l} \in H_l\), and \(p(i_l) \triangleq \text{Pr}(I_l = i_l)\) is the probability of index \(i_l \in I_l\). An alternative rate definition is presented in the following.

Rate measure

Following the approaches in [166, 183, 184] and [185, Sect. 4.2], instead of using (67), the average rate is approximated via the entropy bound of a source code\(^{23}\). Let \(r(p(i_l))\) be a rate measure which is a function of the message index probabilities \(p(i_l), i_l \in I_l, l = 1, 2\). Accordingly, \(R\) in (66) is given as the expectation of the rate measures, i.e.,

\[
R \triangleq \sum_{l=1}^{2} \sum_{i_l=1}^{\vert I_l \vert} p(i_l) r(p(i_l)). \tag{68}
\]

As a major benefit, the rate definition in (68) permits flexible treatment of the entropy coding without tying the design to any particular source code.

\(^{23}\)The design of practical codes is beyond the scope of this thesis.
Entropy bounds and their associated rate measures for various coding settings have been listed in, e.g., [183] and [185, Sect. 4.2]. For the considered separate coding, \( r(p(i)) = -\log_2 p(i), \) \( i \in \mathcal{I}_l \), and, thus, the average sum rate is approximated as \( \bar{R} = H(I_1) + H(I_2) \), where \( H(I_l) = -\sum_{i=1}^{\left|\mathcal{I}_l\right|} p(i) \log_2 p(i) \) is the entropy of message index \( I_l, l = 1, 2 \).

### 4.2 Distributed variable-rate QCS method

This section describes a DR optimization framework for the variable-rate communication system of Fig. 4. A novel distributed QCS method for the efficient acquisition of the correlated sparse sources \( X_1 \) and \( X_2 \) under complexity-constrained encoding is developed. Practical training algorithms are proposed, and implementation and complexity aspects are discussed.

#### 4.2.1 Problem formulation

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

\[
D \triangleq D_1 + D_2 = \sum_{l=1}^2 \mathbb{E} \left[ \| X_l - \hat{X}_l \|_2^2 \right],
\]

where the expectation \( \mathbb{E}[\cdot] \) is taken over the distributions of \( X_l \) and \( W_l, l = 1, 2 \). Furthermore, let \( \mathcal{L}_\mu(D, R) \) be a weighted DR cost function as

\[
\mathcal{L}_\mu(D, R) \triangleq (1 - \mu)D + \mu \bar{R},
\]

where \( \mu \in [0, 1] \) is a weighting parameter for adjusting the DR trade-off, and \( \bar{R} \) is the average sum rate given in (68).

The objective is to minimize \( \mathcal{L}_\mu(D, R) \) in (70) for a given \( \mu \) 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 \( \pi_l \)), and the decoder \( D \) (i.e., the reconstruction codebooks \( \mathcal{C}_l \)) so that the pre-quantization cells \( \mathcal{S}_{l,Y_l}, v_l \in \mathcal{Y}_l \), at each \( PQ_l, l = 1, 2 \), satisfy a nearest-neighbor (NN) condition [280]. As will be elaborated later, this structural constraint is important in restraining 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 4.2.2), followed by the optimization of
E₁, E₂, and D for fixed pre-quantizers (Section 4.2.3). Despite the sub-optimality, the approach is empirically shown to yield satisfactory performance (Section 4.3).

The rate term \( \mu R \) in (70) eliminates the need of constructing specific codebooks \( \mathcal{H}_1 \) and \( \mathcal{H}_2 \) in the optimization phase. As demonstrated in [166] (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 [166]. These include Huffman codes [164] and [158, Sect. 5.6] and arithmetic codes [281]. Note that \( \mu = 0 \) realizes a minimum-distortion fixed-rate method with \( R_l = \log_2 |\mathcal{I}_l|, l = 1, 2. \)

### 4.2.2 Optimization of pre-quantizers

At each \( \text{PQ}_l \), the cells \( \mathcal{S}_{l,v_l} \) and codepoints \( g_{l,v_l}, v_l \in \mathcal{V}_l \), of the \( |\mathcal{V}_l| \)-level VQ are optimized to minimize the MSE distortion induced by discretizing the measurement vector space (see (62)), i.e., the distortion

\[
D_{pq}^l \triangleq \sum_{v_l=1}^{\mathcal{V}_l} p(v_l) \mathbb{E} \left[ \| Y_l - g_{l,v_l} \|_2^2 | V_l = v_l \right], \ l = 1, 2. \tag{71}
\]

This approach inherently results in the required NN encoding (cf. (72)). Since finding the globally optimal partition and codebook of a quantizer is intractable, an alternating optimization technique [169–171, 183, 222, 223, 282–284] is adopted to derive the necessary optimality conditions. Such conditions serve as a practical means to train each \( \text{PQ}_l, l = 1, 2 \), via principles of the iterative Linde-Buzo-Gray (LBG) algorithm [169], elaborated in Section 4.2.4.

For given codepoints \( g_{l,v_l}, v_l \in \mathcal{V}_l \), the optimal cells which minimize \( D_{pq}^l \) satisfy the NN condition [169],

\[
\mathcal{S}_{l,v_l}^* = \left\{ y_l : \| y_l - g_{l,v_l} \|_2^2 \leq \| y_l - g_{l,v'_l} \|_2^2, \ \forall v'_l \neq v_l \right\}, \ v_l \in \mathcal{V}_l, \tag{72}
\]

i.e., the cells form a Voronoi partition [161, Sect. 5.1]. For given cells \( \mathcal{S}_{l,v_l}, v_l \in \mathcal{V}_l \), the optimal codepoints satisfy the centroid condition [169]

\[
g_{l,v_l}^* = \mathbb{E}[Y_l | V_l = v_l] = \frac{1}{p(v_l)} \int_{\mathcal{S}_{l,v_l}} y_l f(y_l) dy_l, \ v_l \in \mathcal{V}_l, \tag{73}
\]

where \( f(y_l) \) is the probability density function (PDF) of \( Y_l. \)
4.2.3 Alternating optimization of encoders & decoder

In this section, the encoders $E_1$ and $E_2$, and decoder $D$ are optimized to minimize $\mathcal{L}_\mu(D,R)$ in (70) for fixed $PQ_1$ and $PQ_2$ (designed as described in Section 4.2.2 and 4.2.4). The optimization involves six sets of optimization variables: the message index mappings $\pi_1$ and $\pi_2$, the rate measures $r(p(i_1))$ and $r(p(i_2))$, and the codevectors $c_{1,i_1,i_2}$ and $c_{2,i_1,i_2}$, $i_1 \in S_1$, $i_2 \in S_2$. Due to the intractability of joint optimization, the alternating optimization principles (see Section 4.2.2) are applied, and the necessary optimality conditions for each variable set while keeping the others fixed are derived.

The implementation perspective is detailed in Section 4.2.4.

Message index mappings

Firstly, the minimization of $\mathcal{L}_\mu(D,R)$ over the message index mapping $\pi_l$ is independent of $r(p(i_l))$, $i_l \in S_l$, $l \neq l'$. Hence, for fixed $r(p(i_l))$, $i_l \in S_l$, $l \neq l'$, and $\pi_l$, the optimal message index mapping $\pi_l^\dagger$ for sensor $l \neq l'$ is the one that minimizes $\mathcal{L}_\mu(D,R)$. The distortion term $D$ of $\mathcal{L}_\mu(D,R)$ can be reformulated as

\[
D = \sum_{l=1}^{2} \sum_{i_1 \in S_1} \sum_{i_2 \in S_2} \sum_{v_1} \int_{Y_1} \int_{Y_2} p(v_1,v_2) \left[ \mathbb{E}\left[ \left\| X_l - c_{l,i_1(i_1),i_2(v_1)} \right\|^2 \right] \right] f(y_1,y_2) dy_1 dy_2
\]

\[
= \sum_{l=1}^{2} \sum_{i_1 \in S_1} \sum_{i_2 \in S_2} \sum_{v_1} \sum_{v_2} \left\{ \mathbb{E}\left[ \left\| X_l \right\|^2 \right] \right\} f(y_1,y_2) dy_1 dy_2
\]

where (a) follows from i) the Markov properties\(^{24}\) $V_l \rightarrow Y_l \rightarrow Y_{l'} \rightarrow V_{l'}$, $X_l \rightarrow (V_l,V_2) \rightarrow \mathcal{C}_l$, $X_l \rightarrow Y_l \rightarrow V_l$, and $X_l \rightarrow Y_{l'} \rightarrow V_{l'}$, $l \neq l'$, and ii) the fact that $p(y_l | y_{l'}) = 1$, if $y_l \in \mathcal{C}_{l,w}$, and 0 otherwise, $l = 1,2$. (b) follows from $p(v_1,v_2) = \int_{y_1 \in \mathcal{S}_{i_1}} \int_{y_2 \in \mathcal{S}_{i_2}} f(y_1,y_2) dy_1 dy_2$, and the definition of a vector

---

\(^{24}\)Random variables $X$, $Y$, and $Z$ form a Markov chain $Z \rightarrow Y \rightarrow X$ (in this order) if $X$ and $Z$ are conditionally independent given $Y$ [158, Sect. 2.8]. Consequently, $p(x,y,z) = p(x)p(y|x)p(z|y)$. 

118
\[ z_{l,v_1,v_2} \in \mathbb{R}^N \text{ as} \]
\[ z_{l,v_1,v_2} \triangleq \mathbb{E}[X_l | V_1 = v_1, V_2 = v_2], \quad l = 1,2 \]
\[ = \frac{1}{p(v_1,v_2)} \int_{y_1 \in \mathcal{Y}_1} \int_{y_2 \in \mathcal{Y}_2} \mathbb{E}[X_l | Y_1 = y_1, Y_2 = y_2]/(y_1,y_2) dy_1 dy_2, \]

(75)

which represents the centroid of the minimum mean square error (MMSE) estimates of source \( X_l, l = 1,2 \), for those measurement realizations \( y_1 \) and \( y_2 \) 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 \( \text{PQ}_1 \) and \( \text{PQ}_2 \).

**Remark 3.** A closed-form expression of the MMSE estimate \( \mathbb{E}[X_l | Y_1 = y_1, Y_2 = y_2], l = 1,2 \), in (75) has been derived for a similar signal setup in, e.g., [223, Proposition 1]. Nevertheless, as it will turn out later, computation of these complex estimates is obviated by means of the pre-quantizers \( \text{PQ}_1 \) and \( \text{PQ}_2 \) both in the offline training and online communication phase.

Finally, the rate term \( R \) in \( \Sigma_{\mu} [D, R] \) can be extended as
\[ R = \sum_{l=1}^{2} \sum_{i_l \in \mathcal{I}_l} p(i_l) r(p(i_l)) \]
\[ = \sum_{l=1}^{2} \sum_{i_l \in \mathcal{I}_l} p(i_l) r(p(i_l)). \]

(76)

By combining (74) and (76), and dropping the terms \( \mathbb{E} [\|X_l\|_2^2 | V_1 = v_1, V_2 = v_2] \) which do not affect the minimization, finding the optimal message index mapping \( \pi_l^* = \{ \pi_1^*(1), \ldots, \pi_l^*(|\mathcal{Y}_l|) \} \) for each sensor \( l = 1,2 \) separates into \( |\mathcal{Y}_l| \) subproblems. Namely, the optimal message index for a \( v_1 \)th cell of sensor 1 is given by the minimization problem
\[ \pi_1^*(v_1) = \arg \min_{\pi_1(v_1) \in \mathcal{I}_1} \left\{ (1 - \mu) \sum_{d=1}^{2} \sum_{v_d \in \mathcal{Y}_d} p(v_1,v_2) \left( \|e_l,\pi_1(v_1),\pi_2(v_2)\|_2^2 - 2c_{l,\pi_1(v_1),\pi_2(v_2)}^T e_{l,v_1,v_2} \right) + \mu p(v_1) r(p(\pi_1(v_1))) \right\}, \forall v_1 \in \mathcal{Y}_1. \]

(77)

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

The rate measure update follows the procedures in [166, 183]. Accordingly, for fixed \( \pi_l \), the optimal rate measures for each sensor \( l = 1, 2 \) are given as

\[
\begin{align*}
    r^*(p(i_l)) &= -\log_2 p(i_l), \quad \forall i_l \in \mathcal{I}_l, \\
    &= -\log_2 \sum_{v_l \in \pi_l^{-1}(i_l)} p(v_l).
\end{align*}
\]

By (78), the average sum rate (68) becomes \( \mathbf{R} = H(I_1) + H(I_2) \), where \( H(I_l) = -\sum_{i_l=1}^{\pi_l} p(i_l) \log_2 p(i_l) \) is the entropy of \( I_l, l = 1, 2 \).

Reconstruction codebooks

Firstly, the minimization of \( \mathbf{X}_l(D) \) in (70) with respect to the codebook \( \mathcal{C}_l \) is independent of \( \mathcal{R}_l \) and \( \mathcal{R}_l', l' \neq l \). Accordingly, for a fixed \( \pi_1 \) and \( \pi_2 \), the optimal reconstruction codebook \( \mathcal{C}_l^* = \{c_{i_1}^*, \ldots, c_{\pi_l}^*, \ldots, c_{\pi_1}^*, \ldots, c_{\pi_2}^* \} \) 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

\[
    c_{i_1, i_2}^* = \arg\min_{c_{i_1, i_2} \in \mathbb{R}^N} \mathbb{E} \left[ \|X_l - c_{i_1, i_2}\|^2 \mid I_1 = i_1, I_2 = i_2 \right].
\]

From (79), each \( c_{i_1, i_2}^* \) is given by the MMSE estimate of source \( X_l, l = 1, 2 \), given the message index pair \( (i_1, i_2) \) [223], i.e.,

\[
    c_{i_1, i_2}^* = \mathbb{E}[X_l \mid I_1 = i_1, I_2 = i_2], \quad \forall i_1 \in \mathcal{I}_1, i_2 \in \mathcal{I}_2
\]

\[
= \sum_{v_1 \in \pi_1} \sum_{v_2 \in \pi_2} p(v_1, v_2) \mathbb{E}[X_l \mid I_1 = i_1, I_2 = i_2, v_1 = v_1, v_2 = v_2]

\]

\[
= \sum_{v_1 \in \pi_1} \sum_{v_2 \in \pi_2} \frac{p(v_1, v_2) p(i_1, i_2)}{p(i_1, i_2)} \mathbb{E}[X_l \mid v_1 = v_1, v_2 = v_2]

\]

\[
= \sum_{v_1 \in \pi_1} \sum_{v_2 \in \pi_2} \frac{p(v_1, v_2)}{p(i_1, i_2)} \mathbb{E}[X_l \mid v_1 = v_1, v_2 = v_2]

\]

\[
= \frac{1}{\sum_{v_1 \in \pi_1^{-1}(i_1)} \sum_{v_2 \in \pi_2^{-1}(i_2)} p(v_1, v_2)}
\]

where (a) follows from the Markov properties \( X_l \rightarrow V_l \rightarrow I_l \) and \( X_l \rightarrow V_{l'} \rightarrow I_{l'}, l \neq l' \); (b) follows from the Markov property \( I_l \rightarrow V_l \rightarrow V_{l'} \rightarrow I_{l'}, l \neq l' \), and the fact that \( p(i_l | v_l) = 1, \pi_l(v_l) = i_l, \) and 0 otherwise; (c) follows from (75).
Algorithm 4 Training algorithm for PQ, l = 1, 2 (offline).

**Inputs:** a) CS matrix \( \Phi_l \); b) measurement training vectors \( \{y^{(1)}_l, y^{(2)}_l, \ldots \} \); c) pre-quantization rate \( R_l \).

**Initialization:** Initial codebook \( \mathcal{G}_l = \{g_{l,1}, \ldots, g_{l,|V_l|}\} \).

**Repeat**

1) For given \( \mathcal{G}_l \), find the optimal cells \( \mathcal{S}_{l,v}^* \), \( v_l \in \mathcal{Y}_l \), by classifying the vectors \( \{y^{(1)}_l, y^{(2)}_l, \ldots \} \) according to (72).

2) For a given \( \mathcal{S}_{l,v}, v_l \in \mathcal{Y}_l \), compute the optimal codepoints \( g_{l,v}^* \), \( v_l \in \mathcal{Y}_l \), as the conditional expectations in (73).

**until convergence**

**Output:** Pre-quantization codebook \( \mathcal{G}_l = \{g_{l,1}, \ldots, g_{l,|V_l|}\} \).

### 4.2.4 Algorithm implementation

The 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 proposed. Each algorithm’s complexity in terms of its computational and memory requirements are also analyzed.

**Offline training phase**

The proposed offline training algorithm for optimizing each PQ, \( l = 1, 2 \), is described in Algorithm 4, and the algorithm for optimizing \( E_1, E_2, \) and \( D \) is presented in Algorithm 5. The foundation of both algorithms is the iterative Lloyd algorithm [169, 170]: in Algorithm 4, each PQ, \( l = 1, 2 \), is optimized by successively applying the necessary optimality conditions (72) and (73) (Steps 1 and 2)). Similarly, the optimization of \( E_1, E_2, \) and \( D \) in Algorithm 5 relies on the six-step iteration loop\(^{25}\), in which the optimality conditions (77), (78), and (80) (Steps 1 – 3)) are first applied in respect to sensor 1, and then, in respect to sensor 2. Once Algorithm 5 has converged, the resulting message index probabilities \( p\{i_l, i_l \in \mathcal{F}_l, l = 1, 2 \) are used to generate the desired binary source codebooks \( \mathcal{H}_l \) and \( \mathcal{H}_2 \).

\(^{25}\)Other optimization orders within the sequential algorithm could be considered as well.
The requisite training data sets for Algorithms 4 and 5 are generated as follows. Source and noise samples \( \{x_1^{(1)}, x_2^{(2)}, \ldots\} \) and \( \{w_1^{(1)}, w_2^{(2)}, \ldots\} \) are created by sampling from their respective distributions. Then, the measurement training vectors \( \{y_1^{(1)}, y_2^{(2)}, \ldots\} \) generated according to (62) are used to optimize each \( \text{PQ}_l \), \( l = 1, 2 \). The MMSE estimate centroids \( z_1, v_1, v_2 \in V_1, v_2 \in V_2 \), for Algorithm 5 are pre-computed as conditional expectations (75) using the source samples \( \{x_1^{(1)}, x_2^{(2)}, \ldots\} \) and the indices \( \{v_1^{(1)}, v_2^{(2)}, \ldots\} \) obtained from quantizing \( \{y_1^{(1)}, y_2^{(2)}, \ldots\} \) via \( \text{PQ}_l \), \( l = 1, 2 \). The multi-dimensional integrals in (73) and (80) are similarly evaluated via Monte Carlo integration techniques.

**Convergence:** The convergence of the iterative descent algorithms in Algorithms 4 and 5 rests on the rationale behind the Lloyd and LBG algorithms [169–171]; 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 [166, 169, 184, 284]. 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., \( \pi_2, \mathcal{C}_1, \) and \( \mathcal{C}_2 \)). One technique to mitigate this problem is the splitting method [169], which is applied in the numerical experiments in Section 4.3. Other robust designs include a deterministic annealing method for fixed-rate VQ design in [285], and its extension as an entropy-constrained design in [286].

**Remark 4.** The entropy-constrained design has a built-in tendency to reduce the codebook sizes, especially for large values of \( \mu \) [166]. Namely, as Algorithm 5 proceeds, some message index, say \( \mathbf{\hat{i}}_1 \in \mathcal{S}_1 \), may become unpopulated (i.e., \( \pi_1^{-1}(\mathbf{\hat{i}}_1) = \emptyset \)) in Step 1). Consequently, the rate measure in Step 2) becomes \( r(p(\mathbf{\hat{i}}_1)) = \infty \), and the codevectors \( c_{1,\mathbf{\hat{i}}_1}, \ldots, c_{1,|\mathcal{S}_2|} \) and \( c_{2,\mathbf{\hat{i}}_1}, \ldots, c_{2,|\mathcal{S}_2|} \) in Step 3) become undefined. It was conjectured in [166] that, unlike for a fixed-rate quantizer, there is no rationale to re-include such unassigned indices in the subsequent iterations.

**Complexity:** The computational and memory requirements are assessed for the most influential factors by estimating the number of involved training vectors, summands, and variables (i.e., indices) in one optimization step of an algorithm. The complexity of optimizing the cells of each \( \text{PQ}_l \), \( l = 1, 2 \), via (72) scales as the number of levels \( |\mathcal{Y}_l| \) and the size of the training data set; the complexity of optimizing the codepoints via (73) scales as \( |\mathcal{Y}_l| \). The complexity of computing the MMSE estimate centroids
Algorithm 5 Training algorithm for $E_1$, $E_2$, and $D$ (offline).

**Inputs:** a) CS matrices $\Phi_l$; b) pre-quantization rates $\bar{R}_l$; c) codebook sizes $|\mathcal{F}_l|$; d) rate measures $r(p(i_l)), i_l \in \mathcal{F}_l$; e) weight parameter $\mu \in [0,1]$; f) MMSE estimate centroids $z_{l,v_1}, v_1 \in \mathcal{Y}_1, v_2 \in \mathcal{Y}_2, l = 1,2$.

**Initializations:** a) Initial codebooks $C_l = \{c_{l,1,1}, \ldots, c_{l,|I_1|}, c_{l,|I_2|}\}, l = 1,2$; b) initial message index mapping $\pi_2 = \{\pi_2(1), \ldots, \pi_2(|V_2|)\}$.

**Repeat**

for $l = 1,2$ do

1) For given $C_1, C_2, \pi_l, l' \neq l$, and $r(p(i_l)), i_l \in \mathcal{F}_l$, find the optimal message index mapping $\pi_l^* = \{\pi_l^*(1), \ldots, \pi_l^*(|V_l|)\}$ via (77).

2) For given $\pi_l$, update the rate measures $r(p(i_l)), i_l \in \mathcal{F}_l$, according to (78).

3) For given $\pi_1$ and $\pi_2$, compute the optimal codevectors $c_{i_1,i_2}^*, i_1 \in \mathcal{F}_1, i_2 \in \mathcal{F}_2$, as the conditional expectations in (80).

**end for**

until convergence

Generate the binary source codebooks $H_l$ using the index probabilities $p(i_l), i_l \in \mathcal{F}_l, l = 1,2$.

**Outputs:** Message index mappings $\pi_l = \{\pi_l(1), \ldots, \pi_l(|V_l|)\}$; source codebooks $H_l = \{h_{l,1}, \ldots, h_{l,|H_l|}\}$; reconstruction codebooks $G_l = \{c_{l,1,1}, \ldots, c_{l,|V_1||V_2|}\}$; encoder mappings $\alpha_l$; decoder mappings $\alpha_l^{-1}$ and $\beta_l, l = 1,2$.

of (75) scales as $2^{|V_1||V_2|}$ and the size of the training data set. Representing the most demanding training steps, the complexity of optimizing $E_1$ and $E_2$ via (77) scales as $|V_1||V_2||\mathcal{F}_1|$ and $|V_1||V_2||\mathcal{F}_2|$, respectively. The complexity of optimizing $D$ via (80) scales as $2^{|V_1||V_2|}$.

**Online communication phase**

The main operations executed for reconstructing a pair of source realizations $x_1$ and $x_2$ in the online phase of the DQCS-PQ method are summarized in Algorithm 6. At each sensor $l = 1,2$, the measurement vector $y_l$ is mapped to the (optimal) cell index $v_l^* \in \mathcal{F}_l$.
Algorithm 6 Distributed variable-rate QCS method with complexity-constrained encoding (DQCS-PQ) (online).

**Inputs:** a) CS matrices $\Phi_l$; b) PQl: codebook $\mathcal{G}_l$; c) $E_l$: mappings $\pi_l$ and $\alpha_l$, and codebook $\mathcal{H}_l$; d) $D$: mappings $\alpha_l^{-1}$ and $\beta_l$, and codebooks $\mathcal{H}_l$ and $\mathcal{G}_l$, $l = 1, 2$.

**Separate encoding at sensor $l = 1, 2$:**

1. Acquisition of CS measurements $y_l$ according to (62).
2. PQl: Assignment of pre-quantization index $v_l^*$ via (81) 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 $h_{l, i_l^*} = \alpha_l(i_l^*)$.

**Joint decoding at the sink:**

4. D: a) Decoding of binary codewords as $(i_l^1, i_l^2) = \left(\alpha_l^{-1}(h_{1, i_l^1}), \alpha_l^{-1}(h_{2, i_l^2})\right)$; b) Joint reconstruction of source estimates as $\hat{x}_l = \beta_l(i_l^1, i_l^2) = \epsilon_{l, i_l^1, i_l^2}$ using $\mathcal{G}_l$, $l = 1, 2$.

(i.e., the output of PQl) using $\mathcal{G}_l = \{g_{l, 1}, \ldots, g_{l, |\mathcal{Y}_l|}\}$ as

$$v_l^* = \underset{v_l \in \mathcal{Y}_l}{\text{argmin}} \|y_l - g_{l, v_l}\|_2^2, \quad l = 1, 2.$$ (81)

For a given $\pi_l = \{\pi_l(1), \ldots, \pi_l(|\mathcal{Y}_l|)\}$, $\alpha_l$, and $\mathcal{H}_l = \{h_{l, 1}, \ldots, h_{l, |\mathcal{Y}_l|}\}$, the index $v_l^*$ is mapped to the message index $i_l^* = \pi_l(v_l^*)$ and further to the binary codeword $h_{l, i_l^*} = \alpha_l(i_l^*)$ at $E_l$. The joint decoder $D$ maps the received codewords to the index pair $(i_l^1, i_l^2) = \left(\alpha_l^{-1}(h_{1, i_l^1}), \alpha_l^{-1}(h_{2, i_l^2})\right)$ and reconstructs source estimates as $\hat{x}_l = \beta_l(i_l^1, i_l^2) = \epsilon_{l, i_l^1, i_l^2}$ using $\mathcal{G}_l = \{\epsilon_{l, 1, 1}, \ldots, \epsilon_{l, |\mathcal{Y}_l|, |\mathcal{Y}_l|}\}$, $l = 1, 2$.

**Complexity:** Thanks to the imposed NN constraint for each PQl, $l = 1, 2$, the encoding complexity remains tolerable: a sensor only has to calculate $|\mathcal{Y}_l|$ distances in (81), followed by the table look-ups associated with $\pi_l$, $\alpha_l$, and $\mathcal{H}_l$. Adjustment of the pre-quantization rate $R_l = \log_2|\mathcal{Y}_l|$ allows DQCS-PQ to make a trade-off between the encoding complexity and compression performance. Clearly, a typical achievable performance range depends on the scaling and operation point of the sensing setup, i.e., through the source statistics and parameters $N$, $M_l$, $K$, $\Phi_l$, and $\sigma^2_W$, $l = 1, 2$. Thus, the need to upscale $R_l$ in a high-dimensional setup may expand the VQ look-up table size $2^{R_l}$ intolerably large. In this regard, the proposed DQCS-PQ is mainly applicable to setups with moderate signal dimensions.
Remark 5. Even though an MMSE estimate $\mathbb{E}[X_l|Y_1 = y_1, Y_2 = y_2]$, $l = 1, 2$, is a constituent quantity in defining the optimal encoders and decoder, the novel pre-quantization technique relieves the sensor of reconstructing this in the online communication phase (see Algorithm 6). This is different from the related works [222, 223], where the necessity of forming these exponentially complex estimates [110–112] hinders the practical implementation. Recall that the pre-quantization also obviates the need of computing the estimates in the offline training phase (see Algorithm 5).

Optimal codeword lengths

As the rate approximation (68) merely steers and primes the quantizer design in subsuming a desired source code, 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 [164] and [158, Sect. 5.6] and arithmetic coding [281] and [158, Sect. 13.3] perform close to the entropy bound $R_l \geq H(I_l)$, $l = 1, 2$ [166, 184].

Remark 6. 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 $R_1 + R_2 \geq H(I_1, I_2)$ [181] and [158, 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. (68)) becomes $R = H(I_1, I_2)$, 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., [287]. The optimal codeword lengths, however, are unknown [184]. Typically, low-density parity-check codes, turbo codes, and syndrome based codes are employed to approach the SW boundary [152, 288, 289]. 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 [290, 291]. Nevertheless, these are beyond the scope of this thesis.
4.3 Numerical results

Numerical results are presented here to illustrate the DR behavior of the proposed DQCS-PQ method (Algorithm 6) and provide comparison against the baseline QCS methods.

4.3.1 Simulation setup

Consider a setup with equal numbers of measurements $M \triangleq M_1 = M_2$, and equal support probabilities $p(s) = 1/(N^2/K)$, $\forall s = 1, \ldots, (N^2/K)$. For a given $\mathcal{S}_s$, the non-zero parts of $\bar{X}$ and $X'$, denoted as $\bar{X}_{\mathcal{S}_s}$ and $X'_{\mathcal{S}_s}$, are defined as i.i.d. Gaussian random variables $\bar{X}_{\mathcal{S}_s} \sim \mathcal{N}(0, \sigma_{\bar{X}}^2 I_K)$, and $X'_{\mathcal{S}_s} \sim \mathcal{N}(0, \sigma_{X'}^2 I_K)$, $s = 1, \ldots, (N^2/K)$, $l = 1, 2$. Thus, $X_l, T_s \sim \mathcal{N}(0, (\sigma_{\bar{X}}^2 + \sigma_{X'}^2) I_K)$. The (spatial) correlation between the sensors is adjusted by parameter $\rho_{\text{corr}} \triangleq \sigma_{\bar{X}}^2 / \sigma_{X'}^2$ with $\sigma_{\bar{X}}^2 = 1$. Two types of measurement matrices $\Phi_1$ and $\Phi_2$ are considered by 1) drawing the entries from Gaussian distribution $\mathcal{N}(0, 1/M)$, and normalizing the columns as $\| \cdot \|_2 = 1$, and 2) taking the first (last) $M$ rows of an $N \times N$ DCT-matrix, and normalizing the columns as $\| \cdot \|_2 = 1$.

The splitting method [162, 169] is employed to optimize both the pre-quantizers (Algorithm 4), and the encoders and decoder (Algorithm 5). In the splitting procedure, a quantization system is first optimized for codebook sizes $|I_l| = 2$, $l = 1, 2$ ($|V_l|$ for PQ). Then, each resultant codevector is split through small perturbation, and the algorithm is run for $|I_l| = 4$ using the new codevectors as initial ones (i.e., a "warm start"). This gradual bifurcation is repeated until the desired codebook sizes are reached.

Unless otherwise stated, $R_1 = R_2 = 10$ bits, and $\log_2 |I_l| = \log_2 |V_l| = \{1, \ldots, 8\}$ bits. 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 (67). The weight parameter is set by a rule $\mu \triangleq \mu_c / \log_2 |I_l|$, where $\mu_c \geq 0$ is a non-negative parameter. When $E_1$, $E_2$, and $D$ are optimized for multiple (ascending) values of $\mu_c$, the quantities $\pi_2$, $\phi_1$, and $\phi_2$ obtained for the preceding (smaller) $\mu_c$ are used to initialize Algorithm 5 for the next $\mu_c$.

Two baseline QCS methods which conform to the customary approaches of early QCS methods by using fixed rate $R_l = \log_2 |I_l|$, and relying on a VQ encoder that minimizes MSE quantization distortion are considered:
1. VQ-E\textsubscript{SD}: a method with separate NN encoding, and separate decoding, where the VQ encoder of the \textit{l}th sensor minimizes \( \sum_{i_l \in \mathcal{I}_l} p(i_l) E \left[ \| Y_{1,i_l} - \hat{Y}_{1,i_l} \|_2^2 | I_l = i_l \right] \) with encoder codepoints \( \hat{y}_{1,i_l} \in \mathbb{R}^M \), and the decoder consists of the MSE-optimal decoder codevectors \( c_{E_{SD}1} = E[X_l | I_l = i_l], i_l \in \mathcal{I}_l, \) for the associated Voronoi regions, \( l = 1, 2, \) such as 

The distortion is measured as \( D_{ave} \triangleq 10 \log_{10} \left(D_1/E[\| X_1 \|_2^2] + D_2/E[\| X_2 \|_2^2] \right) \) (dB), where \( E[\| X_l \|_2^2] = \sum_{i=1}^{N} p(\mathcal{S}_i) E \left[ \| X_l, \mathcal{S}_i \|_2^2 \right] = K(\sigma_W^2 + \sigma_{X_l}^2), \) \( l = 1, 2. \) The rate is measured as \( R_{ave} \triangleq R/2 \) (bits). Due to the exponential complexity of quantizers in the offline training phase, the experiments are restricted to moderate quantization rates and signal dimensions. Data sets of size \( 5 \times 10^5 \) are used for training and performance evaluation.

4.3.2 An illustrative example

In order to visualize the general operation of DQCS-PQ, consider a low-dimensional setup with \( N = 3, \ M = 2, \ K = 1, \ \sigma_W^2 = 0.003, \ \rho_{corr} = 10^2 \), Gaussian measurement matrices

\[
\Phi_1 = \begin{bmatrix}
0.8472 & 0.8044 & -0.6982 \\
0.5312 & 0.5941 & -0.7159 \\
0.9966 & -0.5589 & -0.0016 \\
0.0827 & -0.8293 & 1.0000
\end{bmatrix}, \quad \Phi_2 = \begin{bmatrix}
0.8472 & 0.8044 & -0.6982 \\
0.5312 & 0.5941 & -0.7159 \\
0.9966 & -0.5589 & -0.0016 \\
0.0827 & -0.8293 & 1.0000
\end{bmatrix}, \quad (82)
\]

and quantizer parameters \( \mu_c = 0.5, \ |\mathcal{Y}| = 32, \) and \( |\mathcal{S}_l| = 8. \) The encoder/decoder sides optimized via Algorithms 4 and 5 are depicted in Fig. 18: (a) and (b) illustrate the outcomes of pre-quantization of \( Y_1 \) and \( Y_2, \) and the subsequent message index mappings \( \pi_1 \) and \( \pi_2, \) respectively; (c) and (d) depict the reconstruction codebooks \( \mathcal{C}_1 \) and \( \mathcal{C}_2, \) respectively; (e) tabulates \( \pi_1 \) and \( \pi_2. \) The source codebooks \( \mathcal{H}_1 \) and \( \mathcal{H}_2 \) are presented in Table 4. To get a better understanding of the main assets of DQCS-PQ, VQ-E\textsubscript{SD} is similarly illustrated in Fig. 19.
Firstly, the resulting performance figures 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-Es-DJ. 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.

The key function of PQ1 and PQ2, "encoder shaping", can be perceived from Figs. 18(a) and (b). Take as examples the message index $I_1 = 2$ and $I_2 = 7$: the 5 cells assigned 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-contiguous cells by $\pi^{-1}_2(7) = \{25, 32\}$. This shaping feature allows the cells $S_{l,1}, \ldots, S_{l,32}$ to merge (i.e., classify measurement (training) vectors) into message indices $I_l = \{1, \ldots, 8\}$, $l = 1, 2$, by taking the collective effect of the CS and DSC into account. As a consequence, to some extent the partitioning in Fig. 18(a) follows the sparsity-dependent linear projections $\phi_{l,n} X_{l,n} + W_l$, where $\phi_{l,n} \in \mathbb{R}^M$ is the $n$th column of $\Phi_l$, and $X_{l,n}$ is the $n$th element of $X_l$, $n = 1, 2, 3$. Similarly, Figs. 18(c) and (d) show that nearly all reconstruction codevectors are (approximately) $K$-sparse vectors. Because they are also 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 $X_1$ and $X_2$. In contrast, the encoder regions of VQ-Es-DJ in Fig. 19(a) are solely based on the probability mass of $Y_1$, and the codevectors in Figs. 19(c) and (d) are dispersed with no $K$-sparse structure. Such incognizance of the CS resulted in poor performance.

Table 4 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, $R_{\text{ave}}$ for

\[ 26 \text{This can be interpreted as an index reuse, characteristic to a distributed quantization setup [176, 284].} \]
DQCS-PQ is 39% lower (and $D_{\text{ave}}$ 5.7 dB lower) than for VQ-E5-D5 that uses fixed-length 3-bit codewords.

Fig. 18. Illustration of DQCS-PQ in terms of (a) the pre-quantizer $\text{PQ}_1$ and encoder $E_1$, (b) the pre-quantizer $\text{PQ}_2$ and encoder $E_2$, (c) the reconstruction codebook $\mathcal{C}_1$, (d) reconstruction codebook $\mathcal{C}_2$, and (e) message index mappings $\pi_1$ and $\pi_2$. In (a) and (b), the black dots represent the codepoints $g_{l,1}, \ldots, g_{l,32}$ of $\text{PQ}_1$ and $\text{PQ}_2$; each dotted line represents the sparsity-dependent span $\phi_{l,n} X_{l,n} = 1, 2, 3$; each color indicates the set $\pi_{l-1}(i)$ for message index $i \in \mathcal{X}_l$, $l = 1, 2$ (the colors are equivalent in (e)). In (c) and (d), the dots represent the codevectors $c_{l,1,1}, \ldots, c_{l,8,8}$; the coordinate axes are shown as dashed lines [235] © 2018 IEEE.)
Fig. 19. Illustration of VQ-E\(_S\)-D\(_J\) in terms of (a) the encoder at sensor 1, (b) the encoder at sensor 2, (c) the decoder codevectors for \(X_1\), and (d) the decoder codevectors for \(X_2\). The lines, dots, and colors have the same meanings as those in Fig. 18 ([235] © 2018 IEEE).

### 4.3.3 Distortion-rate performance

Fig. 20(a) depicts the average distortion \(D_{\text{ave}}\) versus the average rate \(R_{\text{ave}}\) for \(N = 20\), \(M = 8\), \(K = 2\), \(\sigma_W^2 = 0.01\), \(\rho_{\text{corr}} = 10^3\), and the Gaussian type \(\Phi_1\) and \(\Phi_2\). By completely disregarding the signal correlation, the VQ-E\(_S\)-D\(_J\) method performs poorly. Significant gains are achieved by using the VQ-E\(_S\)-D\(_J\) for which the joint decoding effectively expands the codebook size from \(|\mathcal{S}|\) to \(|\mathcal{S}|^2\) (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 \(\mu_e\), including the
fixed-rate version \( (\mu_c = 0) \). A similar performance trend is true for all the subsequent experiments as well.

It can be seen that VQ-E_S-D_S becomes saturated with the distortion at around \( D_{\text{ave}} = -5 \) dB. Such an unavoidable constant error level is caused by the remote source coding nature of CS, and, thus, also occurs 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 \( X_1 \) and \( X_2 \) at the decoder, even for the rates approaching infinity. For the VQ-E_S-D_S method, the error level is defined by the rate-independent MMSE estimation error of the form
\[
\sum_{l=1}^{2} \int_{y_l} \left[ \| X_l - E[X_l|Y_l = y_l] \|_2^2 \right] f(y_l) dy_l.
\]
For VQ-E_S-D_J and DQCS-PQ, the level is given by
\[
\sum_{l=1}^{2} \int_{y_1} \int_{y_2} \left[ \| X_l - E[X_l|Y_1 = y_1, Y_2 = y_2] \|_2^2 \right] f(y_1, y_2) dy_1 dy_2.
\]

### 4.3.4 Influence of the weight parameter

The impact of weight parameter \( \mu \) in DQCS-PQ can be perceived from Fig. 20(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, \( R_{\text{ave}} = 5 \) bits due to bad channel conditions, or congested data traffic. Using the entropy coding of the message indices, the reconstruction performance is improved from \( D_{\text{ave}} = -5.8 \) dB \( (\mu_c = 0) \) to \( D_{\text{ave}} = -7.6 \) dB \( (\mu_c = 0.75) \). Alternatively, consider an application with a minimum required reconstruction fidelity of, say, \( D_{\text{ave}} = -7 \) dB. With the aid of variable-length coding, a significant rate reduction of 1.5 bits from \( R_{\text{ave}} = 6.0 \) bits \( (\mu_c = 0) \) to \( R_{\text{ave}} = 4.5 \) bits \( (\mu_c = 0.75) \) is achieved.

### 4.3.5 Influence of source correlation

Figs. 20(a) – (c) show the compression performance for different signal correlation levels \( \rho_{\text{corr}} = \{10^3, 10^2, 10^1\} \). As a first remark, the DQCS-PQ approach 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 \( \rho_{\text{corr}} = 10^3 \) (high correlation), a distortion of \( D_{\text{ave}} = -7 \) dB is achieved at rates \( R_{\text{ave}} = 6.7 \) bits for VQ-E_S-D_J, and 4.5 bits for DQCS-PQ \( (\mu_c = 0.75) \). For \( \rho_{\text{corr}} = 10^1 \) (low correlation),
Fig. 20. Average distortion vs. average rate for $N = 20$, $M = 8$, $K = 2$, the Gaussian type $\Phi_1$ and $\Phi_2$, 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) ([235] © 2018 IEEE).
the respective rates are $R_{\text{ave}} = 7.4$ and 6.0 bits. Note that while VQ-E5-DJ and DQCS-PQ perform better with the increasing correlations, the indiscernible disparities of the curves of fully separate VQ-E5-D5 are solely attributed to different signal powers $E[\|X_l\|_2^2]$.

### 4.3.6 Influence of number of measurements

Fig. 21 demonstrates the influence of different number of CS measurements $M = \{2, 3, 4\}$ in the setup with $N = 10$, $K = 1$, $\sigma_w^2 = 0.01$, $\rho_{\text{corr}} = 10^2$, and the DCT type $\Phi_1$ and $\Phi_2$. 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 which allows each QCS method to compress the sparse sources more reliably. However, whereas VQ-E5-D5 and VQ-E5-DJ 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, a further increase of $M$ 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.

### 4.4 Summary and discussion

This chapter addressed lossy DSC for efficiently acquiring correlated sparse sources from quantized noisy CS measurements in WSNs. A novel 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 pre-discretizing 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.
Fig. 21. Average distortion vs. average rate for $N = 10$, $K = 1$, $\rho_{corr}^L = 10^2$, the DCT type $\Phi_1$ and $\Phi_2$, and number of CS 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) ([235] © 2018 IEEE).
The numerical results illustrated that the proposed method had 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.

A few generalities underlying the presented design are worth mentioning. First, as the signal characteristics were not explicitly used in the derivations, the signal and measurement model is not restricted to sparse signals and CS based sensors. Accordingly, the proposed design can straightforwardly be applied to a general distributed joint estimation and compression scheme. Second, although separate encoding of message indices was used, the design can readily be modified to incorporate various entropy coding classes in the system. In particular, practical codes that approach a point on the SW boundary are of great interest.
5 Rate-distortion performance of lossy compressed sensing

This chapter 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 derived results shed light on the compression performance limits of the single-sensor version of the QCS setup in Chapter 4. The minimum achievable rate for a given distortion fidelity in a QCS setup is represented by the remote rate-distortion function (RDF). In this chapter, an analytically tractable lower bound to the remote RDF is derived by providing support side information (SI) to the encoder and decoder. A variant of the Blahut-Arimoto (BA) algorithm is developed to numerically approximate the remote RDF. Furthermore, a novel entropy-constrained VQ (ECVQ) based QCS method is proposed, which is numerically shown to approach the remote RDF. Numerical results illustrate the main RD characteristics of the lossy CS and compare the performance of practical QCS methods against the proposed limits.

The chapter is organized as follows. Section 5.1 defines the system model and formulates the lossy CS problem. A lower bound to the remote RDF is derived in Section 5.2. A method for numerical approximation of the remote RDF is developed in Section 5.3. A novel QCS method is developed in Section 5.4. Simulation results are provided in Section 5.5. Section 5.6 summarizes the chapter.

5.1 Lossy CS via remote source coding

The objective in this section is to investigate the RD performance of the model depicted in Fig. 22, 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 remote source coding because the encoder accesses the source only through noisy measurements. The source and the CS measurement model are defined next, followed by the formal statement of the problem.
5.1.1 Source model

Let \( \{X_n\}_{n=1}^{\infty} \) be a discrete-time memoryless vector source sequence\(^{27}\). Each vector \( X_n = [X_{n,1} \cdots X_{n,N}]^T \) is \( K \)-sparse\(^{28}\), where \( K \leq N \), i.e., it takes on values in the continuous source alphabet \( \mathcal{X} = \{ x \in \mathbb{R}^N : ||x||_0 = K \} \). The set \( \mathcal{X} \) thus consists of the union of \( \binom{N}{K} \) subspaces, i.e., the signal model is nonlinear [79, 292]. It is further assumed that the source sequence is generated from the memoryless sequence of tuples \( \{ (G_n, B_n) \}_{n=1}^{\infty} \) so that \( X_n = G_n \odot B_n \), where \( \odot \) denotes the Hadamard product; \( G_n \) is a length-\( N \) zero mean Gaussian random vector \( G_n \sim \mathcal{N}(0, \Sigma_G) \) with a covariance matrix \( \Sigma_G \in \mathbb{S}^N^{++} \); \( B_n \) is a length-\( N \) binary support random vector, independent of \( G_n \), with the discrete alphabet \( B = \{ b_1, \ldots, b_{|\mathcal{B}|} \} \), where \( |\mathcal{B}| = \binom{N}{K} \) is the number of all possible sparsity patterns. Each \( b_s = [b_{s,1} \cdots b_{s,N}]^T \in \mathcal{B} \) is unique, contains \( K \) ones and \( N - K \) zeros, and is associated with the \emph{a priori} probability \( p(b_s) \triangleq \Pr(B = b_s) \) with \( p(b_s) \in [0, 1] \) and \( \sum_{s=1}^{|\mathcal{B}|} p(b_s) = 1 \).

5.1.2 Noisy CS

Let \( \Phi \in \mathbb{R}^{M \times N} \) be a fixed and known measurement matrix, \( K \leq M \leq N \). The sensor (i.e., the encoder) observes \( \{X_n\}_{n=1}^{\infty} \) indirectly [62, 63, 65, 70] as

\[
Y_n = \Phi X_n + W_n, \quad n = 1, 2, \ldots \tag{83}
\]

where \( W_n, n = 1, 2, \ldots \), are length-\( M \) random measurement noise vectors independent of \( \{X_n\}_{n=1}^{\infty} \), and each \( Y_n \) is a length-\( M \) measurement random vector that takes values in the measurement vector space \( \mathcal{Y} \). It is assumed that \( W_n \sim \mathcal{N}(0, \Sigma_W) \) with a covariance

---

\(^{27}\)Due to the independence over time, the time index \( n \) will often be suppressed for brevity whenever not explicitly needed.

\(^{28}\)With a slight abuse of terminology, a \( K \)-sparse signal in this chapter contains \textit{exactly} (instead of \textit{at most}) \( K \) non-zero elements.
matrix $\Sigma_W \in \mathbb{G}_+^M$. No restricting assumptions of $\Phi$ are made in the derivations; the impact of $\Phi$ on the CS recovery performance is discussed in Section 1.3.2.

### 5.1.3 Lossy CS problem

Let $X^n \triangledown \{X_n\}_{n=1}^m$ and $x^n \triangledown \{x_n\}_{n=1}^m$ 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 $\hat{X}$ 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 $x = [x_1 \cdots x_N]^T \in \mathcal{X}$ and $\hat{x} = [\hat{x}_1 \cdots \hat{x}_N]^T \in \hat{\mathcal{X}}$ as

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

(84)

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

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

(85)

The lossy source coding system in Fig. 22 operates as follows [161, Sect. 2.1], [158, Sect. 10.2] and [210, Sect. 3.5, 3.6]. The encoder $E$ observes a block of CS measurements $y^m \in \mathcal{Y}^m$ and compresses it into a message represented by an index $u \in \mathcal{U}$ of rate $mNR$ bits using an encoder mapping

$$g^m_E : \mathcal{Y}^m \rightarrow \mathcal{U} \triangleq \{1, \ldots, 2^{mNR}\}$$

(86)

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

$$g^m_D : \mathcal{U} \rightarrow \hat{\mathcal{X}}^m.$$ 

(87)

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

**Definition 1.** *(Lossy CS source coding problem)* Amongst all the E-D pairs of mappings (86) and (87), determine the infimum of (achievable) rates $R$ such that $X$ can be reproduced with the average distortion satisfying $E\left[d(X^m, \hat{X}^m)\right] \leq D + \varepsilon$ for
any positive real number \( \varepsilon \), i.e., define [158, Sect. 10.2]

\[
R_X^\text{rem}(D) = \inf_{(R,D) \in \mathcal{R}} R.
\]  

(88)

\( R_X^\text{rem}(D) \) is called the remote RDF of source \( X \).

It is worth pointing out that the used MSE distortion implies, in general, non-sparse reconstruction, which might be undesirable in certain CS applications. The mathematical expression for \( R_X^\text{rem}(D) \) is derived next.

### 5.1.4 Remote RDF

The general expression of the remote RDF for a discrete memoryless source with discrete memoryless observations has been derived in [157, Eqs. (3.5.1) – (3.5.5)]. Adapting the result to continuous-valued signals \( X \) and \( Y \), \( R_X^\text{rem}(D) \) in (88) can be expressed as

\[
R_X^\text{rem}(D) = \min_{f(\hat{x}|y): \|d(x,\hat{x})\| \leq D} \frac{1}{N} I(Y;\hat{X})
\]

(89a)

where the optimization is over the conditional PDF \( f(\hat{x}|y) \), commonly referred to as the test channel, and \( d(x,\hat{x}) \) is the distortion in (84). The mutual information\(^{29}\) between \( Y \) and \( \hat{X} \) is

\[
I(Y;\hat{X}) = \int_y \int_{\hat{x}} f(y)f(\hat{x}|y)\log \frac{f(\hat{x}|y)}{f(\hat{x})} dy d\hat{x}
\]

(89b)

and the average MSE distortion between \( X \) and \( \hat{X} \) is

\[
E[d(X,\hat{X})] \overset{(a)}{=} \int_x \int_{\hat{x}} f(x)f(y|x)f(\hat{x}|y)d(x,\hat{x})dx dy d\hat{x}
\]

(89c)

where (a) follows from \( f(\hat{x}|y) = f(\hat{x}|y,x) \) because \( X \to Y \to \hat{X} \) forms a Markov chain. Note that the remote sensing mechanism is captured by the conditional PDF \( f(y|x) \), governed by the CS measurements in (83).

Due to the time-varying sparsity of \( \{X_n\}_{n=1}^\infty \) through \( \{B_n\}_{n=1}^\infty \), the PDFs of \( X \) and consequently, of \( Y \) are mixture distributions, which seems to make the optimization over \( f(\hat{x}|y) \) in (89a) difficult. Hence, the treatment of the lossy CS problem of Definition 1 is confined to the following two approaches. In Section 5.2, an analytically tractable lower bound to \( R_X^\text{rem}(D) \) is derived, whereas in Section 5.3, a method to

\(^{29}\)The logarithms are in base 2 in this chapter.
numerically approximate $R_{X}^{\text{rem}}(D)$ is developed. Note that a difficulty of the same kind resides also in the direct compression of $X$ for which only RD bounds have been derived [231–233].

### 5.2 Rate-distortion lower bound for lossy CS

A lower bound to $R_{X}^{\text{rem}}(D)$ in (89a) is derived by considering the compression setup of Fig. 23, where, compared to Fig. 22, the encoder $E_{\text{si}}$ and decoder $D_{\text{si}}$ possess SI on sequence $\{B_{n}\}$. Having the support SI at the decoder is often optimistic in practice, but sometimes the encoder may acquire SI on $B$ (i.e., an estimate $\hat{B}$) from the measurements $Y$ at a moderate cost via a sparse signal reconstruction algorithm (see Section 1.3.2). Nevertheless, the shared support SI allows to derive an analytically tractable lower bound to $R_{X}^{\text{rem}}(D)$ which sheds light on the RD behavior of the original setup in Fig. 22, and establishes a benchmark for practical coding methods. The associated RD problem is formulated below.

#### 5.2.1 Lossy CS problem with support SI

Owing to the support SI, an informed lossy source code is defined as follows [202, 204, 207] and [208, Sect. 2.3.1]. The encoder $E_{\text{si}}$ observes a block of CS measurements $y^{m} \in Y^{m}$ along with the SI $b^{m} \in B^{m}$ and compresses it to a message index $u \in \mathcal{U}$ using an encoder mapping

$$g_{E_{\text{si}}}^{m} : Y^{m} \times B^{m} \rightarrow \mathcal{U}. \quad (90)$$

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

$$g_{D_{\text{si}}}^{m} : \mathcal{U} \times B^{m} \rightarrow \hat{X}^{m}. \quad (91)$$
A pair \((R,D)\) for distortion \(D \geq 0\) is achievable if there exists a sequence of informed \(2^{mNR}\)-RD codes with mappings \(g_{E_s}^m, g_{D_s}^m\) so that
\[
\lim_{m \to \infty} \mathbb{E} \left[ d(X^m, \hat{X}^m) \left( g_{E_s}^m(Y^m, B^m), B^m \right) \right] \leq D.
\]
Let \(\mathcal{R}_s\) be the closure of the set of such achievable \((R,D)\) pairs.

**Definition 2.** *(Lossy CS source coding problem with support SI)* Amongst all the \(E_s-D_s\) pairs of mappings (90) and (91), determine the infimum of (achievable) rates \(R\) so that \(X\) can be reproduced with the average distortion satisfying \(\mathbb{E} \left[ d(X^m, \hat{X}^m) \right] \leq D + \varepsilon\) for any positive real number \(\varepsilon\), i.e., define
\[
R_{X:B}^{\text{rem}}(D) = \inf_{(R,D) \in \mathcal{R}_s} R. \tag{92}
\]

\(R_{X:B}^{\text{rem}}(D)\) is called the *conditional remote RDF* of source \(X\). Clearly, \(\mathcal{R}_s \subseteq \mathcal{R}_i\), and \(R_{X:B}^{\text{rem}}(D)\) establishes a lower bound to the best possible compression performance of the lossy CS as
\[
R_X^\text{rem}(D) \geq R_{X:B}^{\text{rem}}(D). \tag{93}
\]

The remainder of the section is devoted to deriving \(R_{X:B}^{\text{rem}}(D)\).

### 5.2.2 Conditional remote RDF

The conditional RDF for a discrete source along with the respective coding theorems is given in [202]. Extending the results to a remote compression setup, the conditional remote RDF \(R_{X:B}^{\text{rem}}(D)\) can be expressed as
\[
R_{X:B}^{\text{rem}}(D) = \min_{\{f(\hat{X}|y)=|B|\}} \frac{1}{N} \mathbb{I}(Y; \hat{X}|B) \tag{94a}
\]
where the optimization is over the \(|B|\) different test channels \(f(\hat{X}|y,b_s), s = 1, \ldots, |B|\), the conditional mutual information between \(Y\) and \(\hat{X}\) given \(B\) is
\[
\mathbb{I}(Y; \hat{X}|B) = \sum_{s=1}^{|B|} p(b_s) \mathbb{I}(Y; \hat{X}|B=b_s) \tag{94b}
\]
and the average MSE distortion between \(X\) and \(\hat{X}\) is
\[
\mathbb{E} \left[ d(X, \hat{X}) \right] = \sum_{s=1}^{|B|} p(b_s) \mathbb{E} \left[ d(X, \hat{X}) | B = b_s \right] \tag{94c}
\]
where, compared to (89c), the expectation is also taken over B. Since B is provided at no cost in the system shown in Fig. 23, \( R_{X:B}^{\text{rem}}(D) \) determines the complementary information rate that must be conveyed to the decoder \( D_n \) to reconstruct \( X \) with fidelity \( D \).

Observe that (94b) and (94c) decompose with respect to realizations \( B = b_s \), \( s = 1, \ldots, |\mathcal{B}| \). The conditional remote RDF \( R_{X:B}^{\text{rem}}(D) \) in (94a) can thus be expressed as the weighted sum minimization [202, Theorem 5]

\[
R_{X:B}^{\text{rem}}(D) = \min_{\sum_{s=1}^{|\mathcal{B}|} p(b_s) R_{X:b_s}^{\text{rem}}(D_s)} \sum_{s=1}^{|\mathcal{B}|} p(b_s) R_{X:b_s}^{\text{rem}}(D_s) \tag{95}
\]

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

\[
R_{X:b_s}^{\text{rem}}(D_s) = \min_{f(\hat{x}|y,b_s) : \mathbb{E}[d(X|\hat{X},B=b_s)] \leq D_s} \frac{1}{N} \sum_{s=1}^{|\mathcal{B}|} p(b_s) I(Y;\hat{X}|B=b_s) \tag{96a}
\]

where the mutual information between \( Y \) and \( \hat{X} \), conditioned on \( B = b_s \), is

\[
I(Y;\hat{X}|B=b_s) = \int_Y \int_{\hat{X}} f(y|b_s) f(\hat{x}|y,b_s) \log \frac{f(\hat{x}|y,b_s)}{f(\hat{x}|b_s)} dy d\hat{x} \tag{96b}
\]

and the average MSE distortion between \( X \) and \( \hat{X} \), conditioned on \( B = b_s \), is

\[
\mathbb{E}[d(X,\hat{X})|B=b_s] = \int_X \int_{\hat{X}} \int_Y f(x|b_s) f(\hat{x}|x,b_s) f(y|\hat{x},b_s) d(x,\hat{x})dy d\hat{x} \tag{96c}
\]

where (a) follows from \( f(\hat{x}|y,x,b_s) = f(\hat{x}|y,b_s) \) because \( X \to Y \to \hat{X} \) forms a Markov chain when conditioned on \( B \). Owing to the support SI, all the PDFs above are equivalent to those in (89), except they are conditioned on the realization \( B = b_s \).

Regarding the above formulations, the characterization of \( R_{X:B}^{\text{rem}}(D) \) in (95) boils down to deriving each \( R_{X:b_s}^{\text{rem}}(D_s), s = 1, \ldots, |\mathcal{B}| \), in (96a). This is carried out in the next section.

5.2.3 Conditional marginal remote RDF

Fundamentally, the conditional marginal remote RDF \( R_{X:b_s}^{\text{rem}}(D_s), s = 1, \ldots, |\mathcal{B}| \), in (96a) determines the minimum (achievable) rate \( R_s \) so that \( X \) can be reproduced with the
average distortion satisfying $\mathbb{E}\left[d(X, \hat{X}) | B = b_s\right] \leq D_s$ in the setup depicted in Fig. 24, where $\Sigma_{i=\mathcal{B}} p(b_s) R_s = R$. As preliminaries for deriving $R_{\text{rem}} (D_s)$, three definitions are introduced.

**Definition 3. (Subsource)** Let $\{X_{s,n}\}_{n=1}^{\infty} = \{G_n \oplus b_s\}_{n=1}^{\infty}$ be the memoryless sequence of the $s$th subsource, consisting of $K$-sparse source vectors $\{X_s\}_{n=1}^{\infty}$ restricted to a fixed realization $B = b_s$, $s = 1, \ldots, |\mathcal{B}|$. Each subsource $X_s$ comprises of two parts: 1) the length-$K$ random vector

$$X_s \triangleq G_{\text{supp}(b_s)} \sim \mathcal{N}(0, \Sigma_{X_s})$$

(97)

that extracts the entries of $X_s$ (i.e., the entries of $G$) restricted to the support of $b_s$, where $\text{supp}(b_s) \triangleq \{k \in \{1, \ldots, N\} | b_{s,k} \neq 0\}$ denotes the support of vector $b_s$. $G_{\text{supp}(b_s)}$ extracts the entries $G_k$ from $G$ for indices $k \in \text{supp}(b_s)$, and the covariance matrix

$\Sigma_{X_s} \in \mathbb{S}_+^K$ extracts the entries $\Sigma_{G}(k,k')$ from $\Sigma_G$ for indices $k,k' \in \text{supp}(b_s)$; 2) the all-zero vector $0_{N-K}$ corresponding to the entries of $X_s$ for indices $k \in \text{supp}(b_s)^c$, where $\text{supp}(b_s)^c \triangleq \{1, \ldots, N \setminus \text{supp}(b_s)\}$ is the complement of $\text{supp}(b_s)$.

It is worth emphasizing that although the subsources $X_s$, $s = 1, \ldots, |\mathcal{B}|$, are virtual, i.e., not actually present in the system, they play an instructive role in the derivations. In light of decomposability, the subsources bear a resemblance to composite sources\(^{30}\) [157, Sect. 6.1.1] and [208].

**Definition 4. (CS measurements of a subsource)** Let $\{Y_{s,n}\}_{n=1}^{\infty}$ be the memoryless sequence of the CS measurements of form (83) restricted to a fixed realization $B = b_s$, i.e., the CS measurements of subsource $X_s$, $s = 1, \ldots, |\mathcal{B}|$, in (97), defined as

$$Y_s \triangleq \Phi X_s + W = \Phi_s X_s + W$$

(98)

\(^{30}\) The sequence pair $\{(X_s, B_s)\}_{s=1}^{\infty}$ forms a jointly stationary and ergodic regenerative composite source with stationary memoryless subsource processes; $\{B_s\}_{s=1}^{\infty}$ is the hidden switch sequence that controls the output process $\{X_s\}_{s=1}^{\infty}$ by randomly activating the subsources $\{X_s\}_{s=1}^{\infty}$ according to probabilities $p(b_s)$, $s = 1, \ldots, |\mathcal{B}|$ [157, Sect. 6.1.1] and [208].
where matrix $\Phi_s \in \mathbb{R}^{M \times K}$ extracts the $K$ columns of $\Phi$ with indices $k \in \text{supp}(b_s)$, and thus, $Y_s \sim \mathcal{N}(0, \Sigma_{Y_s})$ with a covariance matrix $\Sigma_{Y_s} = \Phi_s \Sigma_X \Phi^T_s + \Sigma_W \in \mathbb{S}_+^M$.

**Definition 5.** (MMSE estimator of a subsource) Let $Z_s$ be a length-$N$ random vector representing the MMSE estimator of source $X$ given $Y$ for a fixed realization $B = b_s$, i.e., the MMSE estimator of subsource $X_s$ in (97) given $Y_s$ in (98). Each $Z_s$ is given by the conditional expectation as [293, Sect. 8.2]

$$Z_s \triangleq \mathbb{E}[X_s | Y, B = b_s], \ s = 1, \ldots, |\mathcal{F}|,$$  

which, owing to the sparsity of $X_s$ (cf. (97)), decomposes into two parts: 1) the length-$K$ random vector $Z_s \triangleq \mathbb{E}[X_s | Y, B = b_s] = \Sigma_{X,Y_s} \Sigma_{Y_s}^{-1} Y_s = F_s Y_s \sim \mathcal{N}(0, \Sigma_{Z_s})$ (100)

that represents the MMSE estimator of $X_s$ given $Y$ and $B = b_s$; 2) $0_{N-K}$ that corresponds to the MMSE estimator of the zero part of $X_s$. For jointly Gaussian random vectors, $Z_s$ is linear [294, Sect. 10.2], where the cross-covariance matrix is $\Sigma_{X,Y_s} = \Sigma_X \Phi^T_s \in \mathbb{R}^{K \times M}$, the MMSE estimation matrix is $F_s \triangleq \Sigma_{X,Y_s} \Sigma_{Y_s}^{-1} \in \mathbb{R}^{K \times M}$, and $Z_s \sim \mathcal{N}(0, \Sigma_{Z_s})$ with covariance matrix $\Sigma_{Z_s} = F_s \Sigma_{X,Y_s} \Phi^T_s \in \mathbb{S}_+^K$.

$R_{x|b}(D_s)$ in (96a) can be characterized by a two-stage encoding structure, where the encoder first optimally estimates the subsource $X_s$ (see (97)) from measurements $Y_s$ (see (98)), and then optimally encodes the constructed estimator $Z_s$ in (100). This is elaborated next.

**MMSE distortion separation**

Let $\hat{X}_s$ be a length-$N$ random vector representing the reproduction of subsource $X_s$ at the decoder output (see Fig. 24). Accordingly, the average conditional MSE distortion $\mathbb{E}\left[d(\hat{X}_s, \hat{X}_s)\right] \triangleq \mathbb{E}\left[d(\hat{X}_s, \hat{X}_s) | B = b_s\right]$ in (96c) separates as

$$\mathbb{E}\left[d(\hat{X}_s, \hat{X}_s)\right] = N^{-1} \mathbb{E}\left[\|\hat{X}_s - Z_s + Z_s - \hat{X}_s\|^2\right]$$

$$\overset{(a)}{=} N^{-1} \mathbb{E}\left[\|\hat{X}_s - Z_s\|^2\right] + N^{-1} \mathbb{E}\left[\|Z_s - \hat{X}_s\|^2\right]$$

$$\overset{(b)}{=} D_{Z|b_s} + \mathbb{E}\left[d(Z_s, \hat{X}_s)\right]$$

145
where (a) follows from the MMSE orthogonality principle [293, Sect. 8.2.1] (see Appendix 2); (b) follows after denoting the (rate-dependent) average MSE distortion between $Z_s$ and $\hat{X}_s$ as $E\left[d(Z_s, \hat{X}_s)\right] = N^{-1}E\left[\|Z_s - \hat{X}_s\|_2^2\right]$, and defining the (rate-independent) average MMSE estimation error with respect to subsource $X_s$ as [294, Sect. 10.2] (see Appendix 3)

$$D_{Z;b_s} = N^{-1}\text{Tr}\left(\Sigma_{X_s} - \Sigma_{Z_s}\right).$$

(102)

Separation similar to (101) appears also in, e.g., [172–174].

**Reduced distortion**

Due to the decomposability of (100), the last term in (101) splits as

$$E\left[d(Z_s, \hat{X}_s)\right] = E\left[d(Z_s, \hat{X}_s)\right] + E\left[d(0_{N-K}, \hat{X}_{\text{supp}(b_s)^c})\right]$$

(103)

where $\hat{X}_s$ is the length-$K$ reproduction random vector associated with $X_s$, and $\hat{X}_{\text{supp}(b_s)^c}$ is the reproduction random vector associated with the zero part of $X_s$. Since an RDF is a monotonic nonincreasing function of the distortion [157, Sect. 2], it is optimal for $R_{\text{rem}}(D_s)$ to set $\hat{X}_{\text{supp}(b_s)^c} = 0_{N-K}$, and thus the distortion in (101) reduces to

$$E\left[d(X_s, \hat{X}_s)\right] = D_{Z;b_s} + E\left[d(Z_s, \hat{X}_s)\right].$$

(104)

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

$$D'_s \triangleq D_s - D_{Z;b_s} \geq 0, \ s = 1, \ldots, |B|,$$

(105)

where $D_s \geq 0$ is the distortion criterion in (96a), and $D_{Z;b_s}$ is given in (102). Note that according to (104), $E\left[d(Z_s, \hat{X}_s)\right] \leq D'_s$ implies $E\left[d(X_s, \hat{X}_s)\right] \leq D_s$. 

146
**Estimate-and-compress separation**

Let $R_{\text{dir}}^{\text{Z}_{bs}}(D'_s)$ denote the direct RDF of the MMSE estimator $Z_s$ defined in (100) for the reduced distortion $D'_s$ in (105), i.e., define

$$R_{\text{dir}}^{\text{Z}_{bs}}(D'_s) = \min_{f(\hat{z}_s|z_s) \in [d(Z_s, \hat{Z}_s)] \leq D'_s} \frac{1}{N} I(Z_s; \hat{Z}_s)$$  \hspace{1cm} (106a)

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

$$I(Z_s; \hat{Z}_s) = \int_{z_s} \int_{\hat{z}_s} f(z_s) f(\hat{z}_s|z_s) \log \frac{f(\hat{z}_s|z_s)}{f(\hat{z}_s)} dz_s d\hat{z}_s$$  \hspace{1cm} (106b)

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

$$\mathbb{E}[d(Z_s, \hat{Z}_s)] = \int_{z_s} \int_{\hat{z}_s} f(z_s) f(\hat{z}_s|z_s) d(z_s, \hat{z}_s) dz_s d\hat{z}_s.$$  \hspace{1cm} (106c)

The RDF $R_{\text{dir}}^{\text{Z}_{bs}}(D'_s)$ can be derived by decorrelating the Gaussian (effective) source $Z_s \sim \mathcal{N}(0, \Sigma_{Z_s})$ via the Karhunen-Loève transform, and applying reverse water-filling \[158, \text{Sect. 10.3.3}\]. Accordingly, let $\Sigma_{Z_s} = QQ^\top_s = \Lambda_s$ be the eigendecomposition, where the diagonal matrix $\Lambda_s \triangleq \text{diag}(\lambda_{s,1}, \ldots, \lambda_{s,K})$ contains the eigenvalues $\lambda_{s,1} \geq \cdots \geq \lambda_{s,K} \geq 0$ of $\Sigma_{Z_s} \in \mathbb{S}^K_{++}$, and the columns of $Q_s \in \mathbb{R}^{K \times K}$ are the corresponding eigenvectors. Consequently, $R_{\text{dir}}^{\text{Z}_{bs}}(D'_s)$ is given as

$$R_{\text{dir}}^{\text{Z}_{bs}}(D'_s) = \min_{\sum_{k=1}^K D'_s \lambda_{s,k}} \frac{1}{N} \sum_{k=1}^K \left\{ \max \left\{ 0, \frac{1}{2} \log \frac{\lambda_{s,k}}{D'_s \lambda_{s,k}} \right\} \right\}$$  \hspace{1cm} (107)

where $D'_{s,k}, k = 1, \ldots, K$, are the optimization variables.

The following proposition gives an expression for the conditional marginal remote RDF $R_{\text{rem}}^{\text{X}_{bs}}(D_s)$.

**Proposition 1.** The conditional marginal remote RDF of $X_s$ in (96a) is given as the (direct) RDF of the MMSE estimator $Z_s$ in (107), i.e.,

$$R_{\text{rem}}^{\text{X}_{bs}}(D_s) = R_{\text{dir}}^{\text{Z}_{bs}}(D'_s), \ s = 1, \ldots, |\mathcal{B}|,$$  \hspace{1cm} (108)

147
where $D' = D - D_{Z|b}$ $\geq 0$ is the reduced distortion in (105), and $D_{Z|b}$ is given in (102).

**Proof.** The proposition follows from the proofs in [172, 174].

According to Proposition 1, the remote source coding problem of Definition 2 separates into 1) the MMSE estimation of $X_s$ given $Y_s$, 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 $X_s$ from the noisy linear measurements $Y_s$ in (98), $s = 1, \ldots, |B|$, followed by the optimal coding of the extracted messages. The estimate-and-compress separation is illustrated in Fig. 25.

**Remark 7.** The two expressions (106) and (107) for $R_{Z|b}^{\text{dir}}(D')$ are interrelated by Gaussian "forward channels" depicted in Fig. 25. Namely, the optimal conditional PDF $f(\hat{z}_s|z_s)$ can be described via $Z'_s = \theta_s \hat{Z}'_s + V_s$, $k = 1, \ldots, K$, with parameters $\theta_s = \frac{\lambda_s - D'_s}{\lambda_s}$ and $\sigma^2_{V_s} = \theta_s D'_s$, where $D'_s$ are optimal variables for (107), $Z'_s$ is the $k$th element of the decorrelated MMSE estimator $Z'_s = \mathbf{Q}_s Z_s$, and $V_s \sim \mathcal{N}(0, \sigma^2_{V_s})$ is a zero mean Gaussian random variable independent of $Z'_s$ [157, 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 8.** If $\text{rank}(\Sigma_{Z_s}) < K$, then the covariance matrix $\Sigma_{Z_s}$ has a null-space. Consequently, the random vector $Z_s$ is a degenerate Gaussian vector [295, Lecture 7], and $Z'_s$ contains a deterministic zero part. This case is inherently handled in (107).
by allocating $D_{\mathcal{S}} = 0$ for $k = \text{rank}(\Sigma_{\mathcal{Z}_s}) + 1, \ldots, K$. The null-space may be caused for example by the rank deficiency of matrix $\Phi_s$.

**Remark 9.** A proof of the optimality of the two-step coding structure is implicitly present in the seminal work by Dobrushin and Tsybakov [172, 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) [172, Sect. 7]. Later, Wolf and Ziv [174] 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 1 is also valid for non-Gaussian sources/observations; however, finding analytical expressions for $R_{\mathcal{X}_b, \mathcal{S}}^{\text{dir}}(D_{\mathcal{S}})$ and $D_{\mathcal{Z}, \mathcal{S}}$ may be difficult. Similar separation results appear in, e.g., [175, 178, 179] and [157, Sect. 4.5.4].

**Remark 10.** $R_{\mathcal{X}_b, \mathcal{S}}^{\text{rem}}(D_s)$ is an upper bound to the conditional marginal remote RDF of a subsource $\tilde{\mathcal{X}}_s = \tilde{G} \circ b_s$, where $\tilde{G}$ is a non-Gaussian random vector with covariance matrix $\Sigma_{\tilde{G}} = \Sigma_G$ [157, p. 130].

### 5.2.4 Characterization of the conditional remote RDF

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

$$D_{\mathcal{Z}, \mathcal{B}} \triangleq \sum_{s=1}^{|\mathcal{S}|} p(b_s) D_{\mathcal{Z}, b_s},$$

$$= (a) N^{-1} \sum_{s=1}^{|\mathcal{S}|} p(b_s) \text{Tr} \left( \Sigma_{\mathcal{X}_s} - \Sigma_{\mathcal{Z}_s} \right) \quad (109)$$

where $(a)$ follows from (102). The conditional remote RDF $R_{\mathcal{X}_b, \mathcal{S}}^{\text{rem}}(D)$ is given by the following theorem.

**Theorem 1.** For distortion range $D_{\mathcal{Z}, \mathcal{B}} \leq D < \frac{1}{N} \sum_{s=1}^{|\mathcal{S}|} p(b_s) \text{Tr} \left( \Sigma_{\mathcal{X}_s} \right)$, $R_{\mathcal{X}_b, \mathcal{S}}^{\text{rem}}(D)$ is positive and can be evaluated via the convex minimization problem as

$$R_{\mathcal{X}_b, \mathcal{S}}^{\text{rem}}(D) = \min_{\sum_{s=1}^{|\mathcal{S}|} p(b_s) \sum_{k=1}^K D_{\mathcal{Z}, b_s}^{\prime} = D - D_{\mathcal{Z}, \mathcal{B}}} \frac{1}{N} \sum_{s=1}^{|\mathcal{S}|} p(b_s) \sum_{k=1}^K \max \left\{ 0, 1 - \frac{\lambda_{s,k}}{D_{\mathcal{Z}, b_s}^{\prime}} \right\} \quad (110)$$

149
where $\Sigma_X$ is the covariance matrix of $X$ in (97); $\lambda_s, 1, \ldots, \lambda_s, K$ are the eigenvalues of the covariance matrix $\Sigma_Z$ of $Z_s$ in (100); $D'_{s,k}$ are the optimization variables, $k = 1, \ldots, K, s = 1, \ldots, |B|$. If the distortion values satisfy $D \geq \frac{1}{N} \sum_{s=1}^{\left| B \right|} p(b_s) \text{Tr}\left( \Sigma_X \right)$, then $R_{X|B}^{\text{rem}}(D)$ is zero.

Proof. The proof is given in Appendix 4.

Remark 11. $R_{X|B}^{\text{rem}}(D)$ is an upper bound to the conditional remote RDF of a source $\tilde{X} = \tilde{G} \odot B$, where $\tilde{G}$ is a non-Gaussian random vector with covariance matrix $\Sigma_{\tilde{G}} = \Sigma_G$ [157, p. 130].

In Theorem 1, $R_{X|B}^{\text{rem}}(D)$ is determined by a weighted sum of the RDFs of the MMSE estimators $Z_s$ under a reduced distortion criterion, where the weights, i.e., the prior probabilities of the sparsity patterns $p(b_s), s = 1, \ldots, |B|$, represent the "appearance frequencies" of such estimators. In particular, (110) involves finding the optimal allocation of the distortion components not only for the $|B|$ different sparsity patterns, but also for the $K$ entries of each decorrelated random vector $Z'_s$. This type of weighted minimization is discernibly a consequence of the composite source structure.

$R_{X|B}^{\text{rem}}(D)$ reflects the remote sensing nature of the lossy CS: regardless of the rate, the lowest achievable distortion is ultimately dictated by $D_{Z|B}$ which is a constant term solely governed by the noisy measurement model in (98). This unavoidable degradation in compression performance, which is caused by the indirect observations of the source, distinguishes the lossy CS from directly compressing $X$; see, e.g., the works in [231–233] which derive RD bounds for compressing sparse sources. Note that a constant distortion floor occurs whether or not the support $S_I$ is available and only the respective levels for $R_{X|B}^{\text{rem}}(D)$ and $R_{X}^{\text{rem}}(D)$ are different. This is demonstrated by the numerical results in Section 5.5.

5.3 Numerical approximation of the remote RDF

Since finding an analytical solution for the lossy CS problem of Definition 1 seems to be elusive, a method based on the BA algorithm [194, 195] for the numerical approximation of the remote RDF $R_{X}^{\text{rem}}(D)$ in (89a) will now be developed. The standard BA algorithm is designed for direct (i.e., for intact measurements $Y = X$) source coding with discrete input/output alphabets. Hence, the algorithm needs to be adapted to handle 1) continuous-valued signals $X$ and $Y$, 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.

### 5.3.1 Discretization of signal alphabets

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 $\mathcal{Y} \triangleq \{1, \ldots, |\mathcal{Y}|\}$ be an index set. The $|\mathcal{Y}|$-level VQ is determined by 1) the encoder regions $\mathcal{S}_v$, $v \in \mathcal{Y}$, 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 $\bigcup_{v=1}^{|\mathcal{Y}|} \mathcal{S}_v = \mathcal{Y}$; 2) the reconstruction codebook $\hat{\mathcal{X}}_q \triangleq \{\hat{x}_1, \ldots, \hat{x}_{|\mathcal{Y}|}\}$ with codevectors $\hat{x}_v \in \mathbb{R}^N$, $v \in \mathcal{Y}$. The VQ encoder is a mapping $E_{vq}: \mathcal{Y} \rightarrow \mathcal{V}$ such that for an input $y \in \mathcal{S}_v$, it produces an index $E_{vq}(y) = v \in \mathcal{V}$. The VQ decoder performs an inverse mapping $D_{vq}: \mathcal{V} \rightarrow \hat{\mathcal{X}}_q$ as $D_{vq}(v) = \hat{x}_v \in \hat{\mathcal{X}}_q$. The random variable $V$ represents the VQ output.

Since the next section illuminates the role of the VQ in RD approximation, the specific optimization of the VQ is deferred until Section 5.3.3.

### 5.3.2 Modified Blahut-Arimoto algorithm for lossy CS

Consider a VQ as described above with $p(v) \triangleq \Pr(V = v) = \int_{y \in \mathcal{S}_v} f(y)dy$, $v \in \mathcal{Y}$. Consequently, index $v \in \mathcal{Y}$ represents all the measurement vectors that belong to VQ region $\mathcal{S}_v$. Similarly, let $\hat{\mathcal{X}}$ be a discrete reproduction random vector at the output of decoder $D$ with alphabet $\hat{\mathcal{X}}_q \triangleq \{\hat{x}_1, \ldots, \hat{x}_{|\mathcal{Y}|}\}$ (i.e., the VQ codebook). Replacing $Y$ with $V$ and $\hat{X}$ with $\hat{\mathcal{X}}$ in (89a), $R_{X_{\text{rem}}}^\text{ba}(D)$ can be approximated as

$$R_{X_{\text{rem}}}^\text{ba}(D) = \min_{P(\hat{x}_j|v): \mathbb{E}[d(X,\hat{X})] \leq D} \frac{1}{N} I(V; \hat{\mathcal{X}})$$

(111a)

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

$$I(V; \hat{\mathcal{X}}) = \sum_{v=1}^{|\mathcal{Y}|} \sum_{j=1}^{|\mathcal{Y}|} p(v) p(\hat{x}_j|v) \log \frac{p(\hat{x}_j|v)}{p(\hat{x}_j)}$$

(111b)

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

$$\mathbb{E}[d(X, \hat{\mathcal{X}})] = \sum_{v=1}^{|\mathcal{Y}|} \sum_{j=1}^{|\mathcal{Y}|} p(v) p(\hat{x}_j|v) \bar{d}(X, \hat{x}_j|v)$$

(111c)
where $\hat{d}(\mathbf{X}, \hat{x}_j|v) \geq 0$ is the modified distortion measure, defined as the average per-letter MSE distortion between $\mathbf{X}$ and $\hat{x}_j$ conditioned on $V = v$, i.e.,

$$
\hat{d}(\mathbf{X}, \hat{x}_j|v) = \mathbb{E}[d(\mathbf{X}, \hat{x}_j)|V = v], \: v, \: j \in \mathcal{Y}
$$

\begin{align}
\hat{d}(\mathbf{X}, \hat{x}_j|v) &= \frac{1}{N} \int \mathbb{E} \left[ \| \mathbf{X} - \hat{x}_j \|_2^2 | V = v, Y = y \right] f(y|v) dy \\
&= \frac{1}{N} \int \frac{p(v|y)}{p(v)} \mathbb{E} \left[ \| \mathbf{X} - \hat{x}_j \|_2^2 | V = v, Y = y \right] f(y|v) dy \\
&= \frac{1}{N} \int p(v) \frac{1}{p(v)} \mathbb{E} \left[ \| \mathbf{X} - \hat{x}_j \|_2^2 | V = v \right] f(y|v) dy
\end{align}

(111d)

where $(a)$ follows from the Markov chain $\mathbf{X} \rightarrow Y \rightarrow V$, and from $p(v|y) = 1$, if $y \in \mathcal{Y}$, $v \in \mathcal{Y}$, and 0 otherwise. Note that pre-calculated $|\mathcal{Y}|^2$ quantities $\hat{d}(\mathbf{X}, \hat{x}_j|v)$ remain fixed in the BA algorithm. In the context of discrete remote sources, a distortion measure similar to (111d) appears in, e.g., [157, Sect. 3.5] and [175, 196].

Consider a Lagrangian for (111a) as

$$
\mathcal{L} \left( \{ p(\hat{x}_j|v) \}_{v,j=1}^{|\mathcal{Y}|}, \lambda, \{ v_j \}_{v=1}^{|\mathcal{Y}|} \right) = \frac{1}{N} \sum_{v=1}^{|\mathcal{Y}|} \sum_{j=1}^{|\mathcal{Y}|} p(v)p(\hat{x}_j|v) \log \frac{p(\hat{x}_j|v)}{p(\hat{x}_j)} + \lambda \sum_{v=1}^{|\mathcal{Y}|} \sum_{j=1}^{|\mathcal{Y}|} p(v)p(\hat{x}_j|v)d(\mathbf{X}, \hat{x}_j|v) + \sum_{v=1}^{|\mathcal{Y}|} v_j \sum_{j=1}^{|\mathcal{Y}|} p(\hat{x}_j|v)
$$

(112)

where $\lambda > 0$ is the Lagrange multiplier associated with the sum distortion constraint, and $v_i, \: v \in \mathcal{Y}$, are the Lagrange multipliers associated with the valid conditional probability constraints $\sum_{j=1}^{|\mathcal{Y}|} p(\hat{x}_j|v) = 1, \: \forall v \in \mathcal{Y}$. Following a standard BA procedure, an $(R, D)$ point of $R_{X_{ba}}^X(D)$ in (111a) is obtained by sequentially updating the conditional probabilities $p(\hat{x}_j|v)$ and the reproduction probabilities $p(\hat{x}_j)$ for a fixed $\lambda$ at each iteration $t = 1, 2, \ldots$ as [158, Sect. 10.8]

$$
p(\hat{x}_j|v)^{t+1} := \frac{p(\hat{x}_j)^t \exp \left[ -\lambda \hat{d}(\mathbf{X}, \hat{x}_j|v) \right]}{\sum_{j=1}^{|\mathcal{Y}|} p(\hat{x}_j)^t \exp \left[ -\lambda \hat{d}(\mathbf{X}, \hat{x}_j|v) \right]}, \: \forall v, \: j \in \mathcal{Y} \quad (113a)
$$

$$
p(\hat{x}_j)^{t+1} := \sum_{v=1}^{|\mathcal{Y}|} p(\hat{x}_j|v)^{t+1} p(v), \: \forall j \in \mathcal{Y}, \quad (113b)
$$

until convergence, and by evaluating the rate according to (111b) and the distortion according to (111c). Hence, different values for $\lambda$ sweep the curve for $R_{X_{ba}}^X(D)$, which approximates the remote RDF $R_{X_{ba}}^X(D)$ in (89a) with an accuracy that increases with the number $|\mathcal{Y}|$. 

152
Remark 13 The proposed method is summarized in Algorithm 7. The algorithm can be terminated when the quantities do not significantly change (e.g., when $\sum_{i=1}^{y} \left( p(\hat{X}_i) - p(\hat{X}_i') \right)^2 < \varepsilon_{ba}$ for a pre-defined positive constant $\varepsilon_{ba} > 0$). The algorithm inputs $p(v)$, $\hat{X}_v$, and $d(X, \hat{X}_v)$, $v, j \in \mathcal{Y}$, are the outcomes of the VQ optimization. The optimization is carried out next.

5.3.3 Vector quantization optimization

In Algorithm 7, the accuracy of distortion evaluation through (111c) is ultimately limited by the $|\mathcal{Y}|^2$ fixed quantities $d(X, \hat{X}_j)$, $v, j \in \mathcal{Y}$, of (111d). Taking this into account, the $E^{v_8}$-$D^{v_9}$ pair is optimized to minimize the average MSE distortion between the source $X$ and its $|\mathcal{Y}|$-level reproduction $\hat{X}_q$. Namely, the VQ design involves finding the encoder regions $\mathcal{S}_v$ and codevectors $\hat{X}_v$, $v \in \mathcal{Y}$, as

$$
\{ \mathcal{S}_v, \hat{X}_v \}_{v \in \mathcal{Y}} := \arg \min_{\{ \mathcal{S}_v, \hat{X}_v \}_{v \in \mathcal{Y}}} \frac{1}{N} \sum_{y \in \mathcal{Y}} \sum_{v \in \mathcal{S}_v} p(v|y) \mathbb{E} \left[ \|X - \hat{X}_v\|^2_2 | V = v, Y = y \right] f(y) dy
$$

where $(a)$ follows from the Markov chain $X \rightarrow Y \rightarrow V$, and $p(v|y) = 1$, if $y \in \mathcal{S}_v$, $v \in \mathcal{Y}$, and 0 otherwise; $(b)$ follows from (111d).

Remark 12 Besides $d(X, \hat{X}_j)$, $v, j \in \mathcal{Y}$, the VQ affects the final distortion in (111c) through the conditional probabilities $p(\hat{X}_j|v)$, $v, j \in \mathcal{Y}$ – the variables to be optimized in iterative steps (113a) and (113b). In addition, the VQ affects the rate approximation in (111b) through the index probabilities $p(v)$, $v \in \mathcal{Y}$. Therefore, a better approximation of $R^{v_8}(D)$ is achievable by incorporating the VQ optimization in the iterative loop of Algorithm 7, and, thus, generating a unique VQ for each $\lambda$. For example, the mapping approach in [200] adapts the reproduction points within the optimization loop. Nevertheless, the proposed non-adaptive discretization seems to yield decent accuracy for all distortion values $D$, as demonstrated in Section 5.5.

Remark 13 Setting the conditional probabilities (113a) as $p(\hat{X}_j|v) = 1$ for $v = j$, and 0 otherwise, results in reproduction probabilities $p(\hat{X}_j) = p(v)$ for $v = j$, and 0 otherwise.
Algorithm 7 Modified BA algorithm for approximating the remote RDF $R_{X,ba}^{rem}(D)$

Inputs: a) Lagrange multipliers $\lambda > 0$; b) codevectors $\hat{x}_v$, and index probabilities $p(v)$, $v \in \mathcal{V}$, obtained as described in Section 5.3.3; c) modified distortion measures $\bar{d}(X,\hat{x}_j|v)$, $v, j \in \mathcal{V}'$, of form (111d).

Initializations: a) Set $t := 1$; b) set $p(\hat{x}_j)^t := 1/|\mathcal{V}'|$, $j \in \mathcal{V}'$.

for a given $\lambda$

repeat

1) Update the conditional probabilities $p(\hat{x}_j|v)^{t+1}$, $v, j \in \mathcal{V}'$, according to (113a).

2) Update the reproduction probabilities $p(\hat{x}_j)^{t+1}$, $j \in \mathcal{V}'$, according to (113b).

3) Set $t := t + 1$.

until a pre-defined stopping criterion is met.

4) Compute the rate $R_{\lambda}$ according to (111b), and the distortion $D_{\lambda}$ according to (111c).

end for

Output: $R_{X,ba}^{rem}(D)$ curve determined by the $(R_{\lambda}, D_{\lambda})$ pairs.

in (113b). The distortion (111c) for $R_{X,ba}^{rem}(D)$ then becomes equal to the VQ distortion in (114), and the rate (111b) becomes $R = -\sum_{v=1}^{\mathcal{V}'} p(v)\log(p(v)) = H(V)$, i.e., the entropy of quantization index $V$. $R_{X,ba}^{rem}(D)$ is thus a “noisy VQ” that randomizes the mapping $V \rightarrow \mathcal{Y}^q$ via conditional probabilities $p(\hat{x}_j|v)$ that determine a noisy channel between the encoder output and decoder input.

The joint optimization over $\mathcal{S}_v$ and $\hat{x}_v$, $v \in \mathcal{V}'$, in (114) is intractable, and thus, a common alternating minimization is used to derive necessary optimality conditions. Accordingly, the proposed VQ is equivalent to the VQ in [222] designed for noiseless channels. The optimal encoder regions for fixed codevectors satisfy a generalized nearest-neighbor condition

$$\mathcal{S}_v^* = \left\{ y : \|z - \hat{x}_v\|^2 \leq \|z - \hat{x}_{v'}\|^2, \forall v' \neq v \right\}, \forall v \in \mathcal{V}',$$

(115)
where \( z \in \mathbb{R}^N \) is the MMSE estimate of \( X \) given \( Y = y \) (i.e., \( Z \) is the MMSE estimator of \( X \) given \( Y \]), defined as [110–112] and [113, Sect. 11.5]

\[
\begin{align*}
  z & \triangleq \mathbf{E}[X | Y = y] \\
  &= \sum_{|B|} p(b_i | y) \mathbf{E}[X | Y = y, B = b_i] \\
  &= \sum_{|B|} \frac{p(b_i) f(y | b_i)}{\sum_{|B|} p(b_i) f(y | b_i)} Z_i,
\end{align*}
\]

(116)

where the conditional PDF \( f(y | b_i) \) is Gaussian as \( \mathcal{N}(0, \Sigma Y_i) \) (see Definition 4), and \( Z_i \triangleq \mathbf{E}[X | Y = y, B = b_i] \) is the MMSE estimate of \( X \) given \( Y = y \) and \( B = b_i \), which, according to Definition 5, comprises of vectors \( z_i = F_i y_i \in \mathbb{R}^K \) and \( 0_{N-K} \).

Similarly, the optimal codevectors for fixed encoder region \( s \) satisfy a generalized centroid condition

\[
\hat{x}_s^* = \frac{1}{p(v)} \int_{y \in S_v} \mathbf{E}[X | Y = y] f(y) dy, \quad \forall v \in \mathcal{Y}.
\]

(117)

The VQ can be trained offline via the iterative Lloyd algorithm [169, 170, 282] by successively applying the necessary optimality conditions (115) and (117) for training data sets.

5.4 "Estimate-and-compress" QCS method

Approaching \( R_{\text{cm}}(D) \) in (89a) requires encoding (large) blocks of vectors, which is infeasible in practice. Therefore, a symbol-by-symbol QCS method that follows the optimal "estimate-and-compress" principle underlying \( R_{\text{cm}}(D) \) is proposed. The method is empirically shown to approach \( R_{\text{cm}}(D) \) in Section 5.5.

The proposed method, termed ECVQ-CS, relies on ECVQ [166] and minimizes a weighted distortion-rate cost function (cf. (70))

\[
(1 - \mu_{\text{ec}}) \sum_{i \in \mathcal{I}} p(i) \left[ \|X - c_i\|_2^2 / I = i \right] - \mu_{\text{ec}} \sum_{i \in \mathcal{I}} p(i) \log(p(i))
\]

(118)

where \( I \) is the quantization index with index set \( \mathcal{I} \triangleq \{1, \ldots, |\mathcal{I}|\} \), \( c_i \in \mathbb{R}^N \) are the reconstruction codevectors, and the parameter \( \mu_{\text{ec}} \in [0, 1] \) adjusts the DR trade-off. Using the alternating optimization in line with DQCS-PQ in Algorithm 6 of Chapter 4, ECVQ-CS can be trained via a three-step iterative algorithm, where 1) the encoder
regions are formed as

\[
\mathcal{S}^*_i = \left\{ y : (1 - \mu)\|z - c_i\|^2 - \mu \log(p(i)) \leq (1 - \mu)\|z - c_i\|^2 - \mu \log(p(\ell)), \forall \ell \neq i \right\}, \forall i \in \mathcal{I},
\]

(119)

where \( z = \mathbb{E}[X|Y = y] \) is the MMSE estimate of (116); 2) the rate terms \(-\log(p(i))\), \( i \in \mathcal{I} \), are updated given the new regions; 3) the codevectors are set equivalently to (117). Finally, the index probabilities \( p(i) = \int_{y \in \mathcal{S}_i} f(y)dy \), \( i \in \mathcal{I} \), are used to generate a binary source codebook with an average codeword length close to the index entropy \( H(I) \) (e.g., Huffman codes [164]).

The name "estimate-and-compress" describes the two main steps: the sensor compresses each measurement realization \( y \) by 1) forming the MMSE estimate \( z = \mathbb{E}[X|Y = y] \), and 2) obtaining the optimal encoding index as \( i^* = \arg\min_{i \in \mathcal{I}} (1 - \mu)\|z - c_i\|^2 - \mu \log(p(i)). \) Because ECVQ-CS adheres to this optimal principle, its complexity is relatively high. This is discussed in Section 5.5.1.

5.5 Numerical results

Numerical results are presented here to illustrate the RD behavior of lossy CS, assess the tightness of the proposed lower bound, and compare the performance of several QCS methods against the proposed limits.

5.5.1 Simulation setup

Consider setups with \( \Sigma_G = \sigma^2_G I_N \) with \( \sigma^2_G = 1 \), and \( \Sigma_W = \sigma^2_W I_M \) with \( \sigma^2_W = 0.01 \). The following curves and QCS methods are evaluated:

1) \( R_{rem|B}^{X}(D) \): the conditional remote RDF of Theorem 1.
2) \( R_{rem|B}^{X,b}(D) \): a numerically approximated remote RDF of Algorithm 7.
3) \( R_{dir|B}^{X}(D) \): the conditional direct RDF of \( X \), corresponding to the lossy compression of \( X \) with \( B \) available as the SI at the encoder and decoder, derived in Appendix 5 (see [232, Sect. VII-A]). Note that \( R_{dir|B}^{X}(D) \leq R_{rem|B}^{X}(D) \).
4) \( R_{dir|b}^{X}(D) \): a numerically approximated direct RDF of \( X \) which represents lossy compression of \( X \) (with no support SI), and is obtained by applying
the discretization of Section 5.3.3 and Algorithm 7 using \( Y = X \). Note that \( R_{X:B}^{\text{dir}}(D) \leq R_{X:b}^{\text{rem}}(D) \leq R_{X:b}^{\text{rem}}(D) \).

5) ECVQ-CS: the proposed "estimate-and-compress" method of Section 5.4 with \( \mu_{\mathrm{ec}} = 0.1 / \log(|\mathcal{F}|) \) and Huffman codewords.

6) VQ-CS: the fixed-rate QCS method in [222], where the \( |\mathcal{F}| \)-level VQ is optimized for noiseless channels.

7) VQ-CE: a baseline fixed-rate QCS method that performs the two stages of ECVQ-CS in the reverse (suboptimal) order: the encoder optimally quantizes \( Y \) in the MSE-sense, unaware of \( X \), and the decoder estimates \( X \) from these quantized measurements. The encoder of VQ-CE is an \( |\mathcal{F}| \)-level VQ that minimizes the distortion
\[
\sum_{i \in \mathcal{F}} p(i) \mathbb{E} \left[ \| Y - \hat{y}_i \|_2^2 | I = i \right]
\]
with nearest-neighbor vectors \( \hat{y}_i \in \mathbb{R}^M \), and its decoder consists of the MSE-optimal codevectors of form (117). This "compress-and-estimate" approach underlies many early QCS methods, cf. [215, 218].

8) \( D_{Z|B} \): the average MMSE estimation error in (109) (known \( B \)).

9) \( D_Z \): numerically evaluated average MMSE estimation error of \( X \) given \( Y \), i.e.,
\[
D_Z \triangleq N^{-1} \mathbb{E} \left[ \| X - Z \|_2^2 \right];
\]
estimator \( Z \triangleq \mathbb{E}[X|Y] \) takes values according to (116) (unknown \( B \)).

The quantization rate is set as \( \log|\mathcal{F}| = 12 \) bits for \( R_{X:b}^{\text{rem}}(D) \), and varied as \( \log|\mathcal{F}| = 1, \ldots, 12 \) bits for the QCS methods. The measurement matrix \( \Phi \) is generated by taking the first \( M \) rows of an \( N \times N \) discrete cosine transform matrix, and normalizing the columns as \( \| \cdot \|_2 = 1 \). The distortion is measured as
\[
10 \log_{10} \left( \mathbb{E}[d(X, X^{\text{est}})] / N^{-1} \mathbb{E}[\| X \|_2^2] \right) \text{ dB},
\]
where \( X^{\text{est}} \) is the method-dependent decoded estimate of \( X \), and \( \mathbb{E}[\| X \|_2^2] = \sum_{s=1}^{|\mathcal{F}|} p(b_s) \text{Tr} \left( \Sigma_{X_s} \right) \). The rate is measured as \( R \) bits/entry of \( X \). The convex minimization problems are solved via CVX [251].

Complexity: As the complexities of Algorithm 7 and the QCS methods increase exponentially with the number of VQ levels and \( |\mathcal{F}| \), the experiments are confined to moderate signal dimensions and quantization rates; the most complex setup used in the thesis involves \( N = 20, M = 8, K = 2, \) and \( \log|\mathcal{F}| = \log|\mathcal{F}| = 12 \) bits. As for any discretized BA algorithm, it is worth remarking that the same complexity issue due to a large number of variables remains regardless of the quantization method. In fact, due to the VQ advantages [296], the proposed algorithm enjoys a superior trade-off between the approximation accuracy and the complexity compared to scalar quantization (SQ). The complexities of the QCS methods can be reduced by using SQ or low-complexity VQ variants such as tree-structured, multi-step, and lattice
VQs [162]. Since $Y$ has a Gaussian mixture density, computationally efficient VQs designed in [297] are also potential candidates. An alternative is to reconstruct the signal from quantized measurements through, e.g., standard $\ell_1$-minimization or a greedy reconstruction method, albeit with a low compression performance [212]. The complexity of the ECVQ-CS method could be reduced by approximating the MMSE estimates (116) by, e.g., the randomized orthogonal matching pursuit [110]. These considerations, which could allow more realistic setups with $N$ and $M$ being hundreds, are left for future study.

5.5.2 Rate-distortion behavior of lossy CS

Consider a setup with $N = 7$, $M = 5$, $K = 1$, and equal support probabilities $p(b_s) = 1/|B|$, $\forall s = 1, \ldots, |B|$. Fig. 26(d) depicts the average distortion versus the average rate for different compression schemes.

Consider first the SI aided lower bounds $R_{X:B}^{\text{dir}}(D)$, $R_{X,ba}^{\text{dir}}(D)$, and $R_{X:B}^{\text{rem}}(D)$ to the remote RDF $R_{X:B}^{\text{rem}}(D)$ in (89a). Owing to the direct observations with support SI, $R_{X:b}^{\text{dir}}(D)$ appears as the line (in log scale) with slope $-6\frac{N}{K} = -42$ dB/bit [232] and yields the lowest $R$ for all values of $D$, as expected. The substantially increased rate for $R_{X,b}^{\text{dir}}(D)$ compared to $R_{X:B}^{\text{rem}}(D)$ is caused by the necessity of conveying the support of $X$ to the decoder. While $R_{X:b}^{\text{dir}}(D)$ and $R_{X:B}^{\text{rem}}(D)$ nearly coincide at high distortion, the curves diverge for moderate to low distortion values. The gradually increasing gap between $R_{X:B}^{\text{rem}}(D)$ and $R_{X:b}^{\text{dir}}(D)$ 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 $R_{X:B}^{\text{rem}}(D)$ and $R_{X:b}^{\text{dir}}(D)$ (i.e., $\lim_{D \to \infty} R_{X:B}^{\text{dir}}(D) = \infty$ and $\lim_{D \to 0} R_{X,b}^{\text{dir}}(D) = \infty$), the lowest achievable distortion for $R_{X:B}^{\text{rem}}(D)$ is the MMSE estimation error floor $D_{Z:B}$ (i.e., $\lim_{D \to 0} R_{X:B}^{\text{rem}}(D) = \infty$).

Focus now on the approximate remote RDF $R_{X,b}^{\text{rem}}(D)$, i.e., the best achievable performance of any QCS method. The gap between $R_{X:B}^{\text{rem}}(D)$ and $R_{X,b}^{\text{rem}}(D)$ represents the compression loss induced by the random measurements taken without knowing the sparse support in a QCS setup [212]. 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 proposed lower bound $R_{X:B}^{\text{rem}}(D)$ 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 $D_{Z}$. 

158
As a special remark, the slope of $R_{X,ba}^{(D)}$ 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, $R_{X,ba}^{dir}(D)$ nearly coincides with $R_{X,ba}^{rem}(D)$; for higher rates, the impact of noisy CS measurements increases, thereby degrading the performance of $R_{X,ba}^{rem}(D)$. Regarding the approximation accuracy of $R_{X,ba}^{dir}(D)$ and $R_{X,ba}^{rem}(D)$, observe that the highest obtained rate is $R \approx 1.3$ bits, so the "over-sampling ratio" of the VQ discretization is at least $|\gamma|/2^{NR} \approx 7.5$.

Because the encoder of VQ-CE is CS-blind, its performance is the worst amongst the QCS methods. The advantages of entropy coding are shown by the proposed ECVQ-CS curves which, for moderate rates, approach the compression limit $R_{X,ba}^{rem}(D)$.
As a proof of validity, the VQ-CS method eventually saturates to $D_Z$, which is expected to also happen for the other QCS methods at sufficiently high rates.

### 5.5.3 Effect of the number of measurements

For the setup of Section 5.5.2, Figs. 26(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 $D_Z$ decreases, and the performance of each method that has no support SI moves closer to the lower bound $R_{X:B}^{\text{rem}}(D)$. 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 $R_{X:B}^{\text{dir}}(D), R_{X:ba}^{\text{dir}}(D), R_{X:B}^{\text{rem}}(D)$, and $D_Z$ remain unaltered.

### 5.5.4 Effect of support probabilities

Consider a setup with $N = 20$, $M = 8$, $K = 2$, and unequal support probabilities as $p(b_s) = \frac{\alpha_{pl}^s}{\sum_{s=1}^{|\mathcal{S}|} \alpha_{pl}^s}, s = 1, \ldots, |\mathcal{S}|$, where $0 < \alpha_{pl} \leq 1$ is a parameter that adjusts the concentration of the probability mass function of $\mathcal{B}$, and $1 > p(b_1) \geq \cdots \geq p(b_{|\mathcal{S}|}) > 0$. For small values of $\alpha_{pl}$, the probability mass concentrates around a fraction of elements in $\mathcal{B} = \{b_1, \ldots, b_{|\mathcal{S}|}\}$, and vice versa. $\alpha_{pl} = 1$ corresponds to the uniform distribution, whereas $\alpha_{pl} \to 0$ approaches remote compression of only a single sparse vector. The vectors $b_s$ in alphabet $\mathcal{B}$ are ordered so that the decimal number of a binary string represented by $b_s + 1$ is greater than that of $b_s, s = 1, \ldots, |\mathcal{S}| - 1$ ($|\mathcal{S}| = 190$).

Fig. 27 shows the average distortion versus the average rate for $\alpha_{pl} = \{0.98, 0.90, 0.72\}$. Decreasing $\alpha_{pl}$ 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 $R_{X:ba}^{\text{rem}}(D)$ towards $R_{X:B}^{\text{rem}}(D)$ and $R_{X:ba}^{\text{dir}}(D)$ towards $R_{X:B}^{\text{dir}}(D)$, and the diminution of the gap between $D_Z$ and $D_Z:B$, which is related to the best possible support recovery for a given setup. This exemplifies that, for a sufficiently concentrated probability mass of $\mathcal{B}$, the proposed ECVQ-CS
Fig. 27. RD performance of lossy CS schemes for $N = 20$, $M = 8$, $K = 2$, and power law type support probabilities with parameters (a) $\alpha_{pl} = 0.98$, (b) $\alpha_{pl} = 0.90$, and (c) $\alpha_{pl} = 0.72$. The colors and markers of the curves in (b) and (c) are equivalent to those in (a) ([238] © 2018 IEEE).
efficiently encodes sparse vectors from noisy CS measurements: its performance approaches the best achievable performance of a support unaware QCS method (i.e., $R_{X,bs}^\text{rem}(D)$). The result illustrates the MSE separation principle (cf. (101)): an efficient QCS method implicitly (successfully) recovers $X$ from $Y$ and encodes the resulting estimates optimally.

5.6 Summary and discussion

This chapter 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 devised 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 proposed limits. In particular, the proposed ECVQ based QCS method was numerically shown to approach the compression limit.

As illustrated by the numerical experiments, the sparsity of a signal is a feature that provides substantial compression gains, and consequently, energy savings in a CS based acquisition setup. Accordingly, finding an appropriate trade-off point in terms of the compression rate and allowed distortion in a low-power sensor application involving sparse signals is worth striving for. While the lossy CS source coding problem specified in Definition 1 was not analytically solved, the proposed lower bound, and the numerically evaluated remote RDF are anticipated to pave the way for further studies in the framework.
6 Conclusions and future work

6.1 Conclusions

The objective of this thesis was to develop and analyze energy-efficient distributed data gathering techniques that could reduce sensors’ energy consumption in future WSNs. Four different approaches were investigated: 1) distributed cross-layer based network optimization, 2) distributed sequential CS acquisition of correlated sensor data streams, 3) distributed DR optimized QCS data compression, and 4) an information-theoretical RD performance analysis of lossy CS.

The first approach to prolong the sensors’ battery lifetimes focused on cross-layer optimization of the physical and network layer. The proposed design minimizes the sum transmit power of the sensors by jointly optimizing the transmit power and bandwidth allocation and multi-hop routing for fixed source rates in an SSDG WSN. A novel consensus ADMM algorithm was developed to solve the problem in a decentralized fashion. Moreover, an advanced variant of the conventional DD algorithm was developed for benchmarking purposes. Compared to DD approaches, the proposed ADMM algorithm was empirically demonstrated to possess superior convergence properties with a low local communication overhead, robustness against fluctuating channel conditions, insensitivity to the step size parameter value, and scalability to large networks. The considered bandwidth allocation was also shown to provide significant system performance gains.

The remaining approaches of the thesis focused on the utilization of CS to efficiently acquire sparse/compressible sources in WSNs. Under this framework, sequential distributed CS data acquisition of spatially and temporally correlated sensor data streams was addressed. A sliding window based CS algorithm to gather data with reduced communications was developed. The novel factors of the algorithm include 1) acquiring spatial sub-sampling based measurements, 2) sliding window processing, 3) the use of Kronecker sparsifying bases, and 4) recursive CS decoding that uses past estimates as prior information via the IRW-$\ell_1$ and $\ell_2$-regularization. The proposed method recovers the sensors’ readings with a low decoding delay, progressively refines the past estimates of the reading values, and makes trade-offs between the recovery performance and decoding complexity via the window size. The algorithm was numerically shown to obtain significant energy savings and result in higher
reconstruction accuracy with less decoding delays and less complexity compared to state-of-the-art CS methods.

The sequential CS framework (as well as the cross-layer design) assumed real-valued signals. The two last approaches of the thesis modified a CS signal acquisition setup by incorporating a quantizer for each sensor. In such a QCS context, a complexity-constrained lossy DSC design was proposed to efficiently compress correlated sparse sources via two CS based sensors. A novel distributed variable-rate DR optimized QCS method that minimizes a weighted sum between the MSE signal reconstruction distortion and an average encoding rate was developed. As the key idea, the continuous encoder inputs are discretized via a pre-quantizer, enabling a trade-off between the compression performance and encoding complexity. The proposed method was shown to outperform baseline QCS methods in terms of compression efficiency and to be adaptable to settings with different performance requirements.

The final approach adhered to an information-theoretic remote source coding framework to study the RD performance of lossy CS. The idea was to investigate the fundamental RD performance limits of finite-rate acquisition of a sparse source via noisy CS. An analytically tractable lower bound to the remote RDF was derived by providing support SI to the encoder and decoder. A discretization based BA variant was developed for numerical approximation of the remote RDF. Numerical results illustrated the main RD characteristics of the lossy CS and demonstrated the compression performance of practical QCS methods against the derived limits. Interestingly, the devised ECVQ based QCS method was numerically shown to approach the remote RDF.

6.2 Future work

In the cross-layer design, the objective function was chosen as the sensors’ sum transmit power. This does not guarantee fairness in the sense that sensors close to the sink are burdened with more data traffic due to relaying, and are thus more prone to battery depletion. Nevertheless, sporadic node failures may not completely terminate the ongoing application because of the spatial correlation in WSN data. A more balanced solution is achievable via network lifetime maximization; see a recent survey in [298]. Whereas multi-path routing optimization provides a general flow solution, it would be interesting to implement and compare different distributed routing algorithms (see, e.g., [299]) against the analytical solution. While the assumed orthogonal FDMA advocates
a simple WSN infrastructure, it would be interesting to extend the algorithm to a setup with interfering links. Design principles can be assimilated from, e.g., [300] which addresses a multicell downlink environment. In the ADMM algorithm, the sensors update the variables synchronously. Since this demand may be too restrictive in some applications, delayed updates via asynchronous ADMM [248, 301] could improve the time-efficiency and enable better scaling for large networks.

In the sequential CS framework, the sliding window was designed to advance one slot at a time. This is beneficial for a delay-stringent WSN application, since each decoding instant \( t \) produces estimates for the current sensor readings \( \mathbf{x}(t) \). In the course of time, these estimates are progressively refined in the next \( W - 1 \) decoding instants. A natural modification is to advance the sliding window multiple slots at a time. While this precludes an instantaneous recovery of the current readings, each sensor could take linear measurements of its consecutive readings (i.e., intra-node CS measurements) to possibly further reduce the transmissions. The proposed method could be extended to use a different 1) window size \( W \), 2) regularization weight parameter \( \gamma_B \), and/or 3) temporal sparsifying basis \( \Psi_T \) – either a universal basis or principal component analysis basis – for the time slots \( t = 1, 2, \ldots \) to dynamically adapt to time-varying statistics of sensor data streams. The presented experiments used synthetic signals. An interesting practical consideration is to tailor the proposed method to some specific application and assess its performance with optimized parameters using a real-world sensor data. Another line of work left for future study is the development of computationally efficient iterative "warm start" decoding algorithms. Competent candidates include the homotopy based algorithms developed in [105, 134]. Finally, since sensor data with a high degree of spatio-temporal correlation encompasses a low-rank (or approximately low-rank) structure [302], rank minimization [303] is a promising alternative for signal reconstruction.

Regarding the devised distributed QCS method, the pre-quantizers were chosen as VQs. While VQ is justifiable owing to the coding optimality [161, Sect. 3.4], its applicability is limited to moderate signal dimensions/quantization rates. At the cost of performance, the complexity could be restrained by approximate methods, including SQ based approaches, and low-complexity VQ variants such as tree-structured, multi-step, and lattice VQs [162]. The design could be refined via 1) the DSC of message indices (see Remark 6 in Chapter 4), and 2) designing the pre-quantizers jointly with the encoders and decoder. While JSM-2 was used as the signal model mainly due to its relevance to timely WSN applications such as collaborative spectrum sensing, other
signal types including JSM-1 and JSM-3 [57, 131], and common sparse signals in [277–279] are straightforwardly applicable. In any case, by limiting studies to one particular signal model, the conducted experiments provide distinct insights and benchmarks for future studies – both experimental and theoretical ones. The proposed design assumes source and measurement distributions that do not change as time passes; a system could benefit from a “universal coding” strategy which, by taking the fluctuating signal characteristics into account in optimization, obtains decent compression performance for various natural (sparse) signals.

Furthermore, since the distributed QCS design adheres solely to source coding, reliable communications of encoder outputs over noisy channels require channel coding. If one wishes to merely concatenate the devised source encoders with off-the-shelf channel encoders, a few erroneously received bits of a variable-length code induce error propagation, known as the catastrophic effect of channel errors [304]. Namely, while separate source and channel coding is asymptotically optimal in a point-to-point setup owing to the celebrated Shannon’s separation theorem [151, Theorem 21], the result does not hold for general multi-user networks [210, Sect. 1.4.4]. Thus, an optimal solution may be obtained via distributed joint source-channel coding [304–307].

In the information-theoretic lossy CS context, finding a closed-form expression for the remote RDF \( R_{X}^{\text{rem}}(D) \) remains the ultimate goal. Alternatively, one could try to accurately approximate \( R_{X}^{\text{rem}}(D) \), or find a substitute for the shared support SI to derive a tighter lower bound as \( R_{X:B}^{\text{rem}}(D) \). It would also be interesting to modify the proposed BA algorithm to make it applicable to high-dimensional signal setups. Alternatively, one could develop a method based on deterministic annealing [200] and assess its performance against the devised BA algorithm. Other open problems include finding theoretical results for 1) the perceived (almost) linear slope of \( R_{X:ba}^{\text{rem}}(D) \) at low rates, and 2) the gap between \( D_{Z} \) and \( D_{Z:B} \), which is ultimately determined by the ability to recover the sparse support of a signal from compressed measurements. It has been shown for direct source compression that if a quantizer is optimized for a memoryless Gaussian source, and the quantizer is then used to compress a non-Gaussian source, the resulting distortion is as bad as (but not worse) if the source were actually Gaussian [308]. If this can be shown for remote compression, the achieved curves for the QCS methods are upper bounds to sparse sources \( \tilde{X} = \tilde{G} \odot B \), where \( \tilde{G} \) is a non-Gaussian random vector with covariance matrix \( \Sigma_{\tilde{G}} = \Sigma_{G} \). In the end, extending the analysis to a distributed multi-sensor setup would provide insightful results for realistic WSNs.
Finally, a combination of the distinct approaches and principles presented in the thesis in a wide-ranging cross-layer design framework merits future investigation. Namely, such joint optimization has great potential to provide further energy savings in data acquisition and offer enhanced compression solutions. While there is an indisputable trade-off between the tractability of a system design and the achievable gains, certain extensions may be viable. To mention a few, incorporating multi-hop routing and transmission energy optimization into a sequential CS framework would further optimize the data collection phase, thereby reducing the communication costs. Such an approach is likely to result in mixed integer programming [126]. Another extension for the considered multi-hop WSN is to also optimize the source rates. A starting point can be assimilated from [250] which combines SW coding with transmission structure optimization in a network. Distributed QCS for communicating correlated sparse sources over noisy channels under sensors’ power constraints would also be a research direction which would be relevant to practical setups. The works in [124, 223] lay a foundation for this approach.
References


176


199. Chang CI & Davison LD (1988) On calculating the capacity of an infinite-input finite
Stanford University.
203. Gray R (1973) A new class of lower bounds to information rates of stationary sources via
204. Cover T & Chiang M (2002) Duality between channel capacity and rate distortion with
205. Cheng S, Stankovic V & Xiong Z (2005) Computing the channel capacity and rate-
4418–4425.
207. Leiner B & Gray R (1974) Rate-distortion theory for ergodic sources with side information
Institute for Systems Research, Univ. Maryland, College Park, MD, USA. URI: http://
drum.lib.umd.edu/handle/1903/4942.
New York, USA.
211. Gibson J (2013) Information Theory and Rate Distortion Theory for Communications and
559–571.
218. Dai W & Milenkovic O (2011) Information theoretical and algorithmic approaches to
Barcelona, Spain.

179


Appendix 1 Construction of matrix $\Psi'$ in (52)

The matrix $\Psi'$ in (52) can be expressed as

$$
\Psi' = \Psi'_T \otimes \Psi_S
\tag{120}
$$

where (a) follows from defining a binary sparse matrix $I'_T \triangleq [I_{W-1} \ 0_{(W-1)\times 1}] \in \mathbb{R}^{(W-1)\times W}$ that performs a row selection of $\Psi_T$ as $\Psi'_T = I'_T \Psi_T$; (b) follows because $\Psi_S = I_N \Psi_S$; (c) follows from $AB \otimes CD = (A \otimes C)(B \otimes D)$; (d) follows from (45); (e) follows from performing a row selection of $\Psi$ via a binary sparse matrix $I'_T \otimes I_N = [I_{N(W-1)} \ 0_{N(W-1)\times N}] \in \mathbb{R}^{N(W-1)\times NW}$ as $(I'_T \otimes I_N)\Psi = [\psi_1 \cdots \psi_{N(W-1)}]^T$, which is the desired result.
Appendix 2 MMSE orthogonality principle for (101)

In order to show that
\[ E \left[ \|X_s - Z_s + \hat{X}_s - \hat{X}_s\|^2 \right] = E \left[ \|X_s - Z_s\|^2 \right] + E \left[ \|Z_s - \hat{X}_s\|^2 \right], \]
the cross-term \( E \left[ (X_s - Z_s)^T (Z_s - \hat{X}_s) \right] \) in (101) is shown to be zero. The term can be written as
\[ E \left[ (X_s - Z_s)^T (Z_s - \hat{X}_s) \right] = E \left[ Z_s^T (X_s - Z_s) \right] - E \left[ \hat{X}_s^T (X_s - Z_s) \right]. \] (121)

By the law of total expectation, the first term of (121) can be written as
\[ E \left[ Z_s^T (X_s - Z_s) \right] = E \left\{ E \left[ Z_s^T (X_s - Z_s) \right] \bigg| Y_s \right\} \]
\[ = E \left( E \left[ Z_s^T (X_s - Z_s) \right] \bigg| Y_s \right) \]
\[ = E \left[ Z_s^T (X_s - Z_s) \right] \] (a)
\[ = 0 \] (122)

where (a) follows from (99). Similarly, the second term of (121) can be written as
\[ E \left[ \hat{X}_s^T (X_s - Z_s) \right] = E \left\{ E \left[ \hat{X}_s^T (X_s - Z_s) \right] \bigg| Y_s \right\} \]
\[ = E \left( E \left[ \hat{X}_s^T (X_s - Z_s) \right] \bigg| Y_s \right) \]
\[ = E \left[ \hat{X}_s^T (X_s - Z_s) \right] \] (a)
\[ = E \left[ \hat{X}_s^T (Y_s - Z_s) \right] \]
\[ = E \left[ \hat{X}_s^T (Y_s - Z_s) \right] \] (b)
\[ = 0 \] (123)

where (a) follows because \( X_s \rightarrow Y_s \rightarrow \hat{X}_s \) forms a Markov chain, and (b) follows from (99). From (122) and (123), \( E \left[ (X_s - Z_s)^T (Z_s - \hat{X}_s) \right] = 0. \)
Appendix 3 MMSE estimation error in (102)

The average MMSE estimation error in (101) can be expressed as

$$D_{Z|b_s} = N^{-1}E\left[\|X_s - Z_s\|^2\right]$$

$$\overset{(a)}{=} N^{-1}E\left[\|X_s - Z_s\|^2\right]$$

$$= N^{-1}\text{Tr}\left\{E\left[(X_s - Z_s)(X_s - Z_s)^T\right]\right\}$$

$$= N^{-1}\text{Tr}\left\{E\left[X_sX_s^T - X_sZ_s^T - Z_sX_s^T + Z_sZ_s^T\right]\right\}$$

$$= N^{-1}\text{Tr}\left\{\Sigma_{X_s} - \Sigma_{X_sZ_s} - \Sigma_{X_sZ_s}^T + \Sigma_{Z_s}\right\}$$

where \((a)\) follows from removing the zero parts of \(X_s\) and \(Z_s\) (see (97) and (100)), and the cross-covariance matrix \(\Sigma_{X_sZ_s} \in \mathbb{R}^{K \times K}\) is

$$\Sigma_{X_sZ_s} = E[X_s(F_sY_s)^T] = \Sigma_{X_sY_s}F_s^T = \Sigma_{X_sY_s}\Sigma_{Y_s}^{-1}\Sigma_{X_sY_s}^T = \Sigma_{Z_s}.$$  (125)

Substituting (125) into (124) results in

$$D_{Z|b_s} = N^{-1}\text{Tr}\left(\Sigma_{X_s} - \Sigma_{Z_s}\right).$$
Appendix 4 The proof of Theorem 1

Using the equivalence \( R_{\text{rem}}^{\text{rem}}(D_x) = R_{\text{Z|B}}^{\text{dir}}(D'_x) \) of Proposition 1, and \( D'_x = D_x - D_{\text{Z|B}} \geq 0 \) in (105), the conditional remote RDF in (95) can be recast as

\[
R_{X|B}^{\text{rem}}(D) = \min_{D'_x \geq 0, s = 1, \ldots, \binom{N}{K}} \frac{1}{N} \sum_{s=1}^{\binom{N}{K}} p(b_s) \max \left\{ 0, \frac{1}{2} \log \frac{\lambda_{x,k}}{D_{s,k}} \right\}
\]

with optimization variables \( D'_x, \ b_s \). Let \( D'_{x,k} \geq 0, \ k = 1, \ldots, K, \ s = 1, \ldots, \binom{N}{K} \), be auxiliary variables for (126). Inserting the new variables with constraint \( \sum_{s=1}^{\binom{N}{K}} D'_{x,k} = D'_{x} \), and substituting \( D_{\text{Z|B}} = \sum_{s=1}^{\binom{N}{K}} p(b_s) D_{\text{Z|B}} \) in (109) and the expression of \( R_{\text{Z|B}}^{\text{dir}}(D'_x) \) in (107), \( R_{X|B}^{\text{rem}}(D) \) in (126) can be equivalently expressed as

\[
R_{X|B}^{\text{rem}}(D) = \min_{D'_x \geq 0, k = 1, \ldots, K, s = 1, \ldots, \binom{N}{K}} \frac{1}{N} \sum_{s=1}^{\binom{N}{K}} p(b_s) \sum_{k=1}^{K} \max \left\{ 0, \frac{1}{2} \log \frac{\lambda_{x,k}}{D_{s,k}} \right\}
\]

with optimization variables \( D'_x, \ b_s \), \( k = 1, \ldots, K, \ s = 1, \ldots, \binom{N}{K} \). Finally, eliminating the variables \( D'_x, s = 1, \ldots, \binom{N}{K} \), by substituting the second set of equality constraints into the first one yields the expression for \( R_{X|B}^{\text{rem}}(D) \) in (110).

Remark 14. A valid distortion is \( D \geq D_{\text{Z|B}} \geq 0 \) because the distortion criterion must be non-negative.

Remark 15. For all distortion criteria \( D \geq \frac{1}{K} \sum_{s=1}^{\binom{N}{K}} p(b_s) \text{Tr}(\Sigma_{X_s}) \), \( R_{X|B}^{\text{rem}}(D) = 0 \); if the encoder sends no information (i.e., \( R = 0 \)), then the decoder can set \( \hat{X} = 0_N \), resulting in an admissible distortion because

\[
\mathbb{E} [d(X, \hat{X})] = \frac{1}{N} \mathbb{E} [\|X - \hat{X}\|_2^2] = \frac{1}{N} \mathbb{E} [\|X\|_2^2] = \frac{1}{N} \sum_{s=1}^{\binom{N}{K}} p(b_s) \text{Tr}(\Sigma_{X_s}) \leq D.
\]

Combining the above derivations with Remarks 14 and 15, \( R_{X|B}^{\text{rem}}(D) \) is characterized as in Theorem 1, which concludes the proof. \( \square \)
Appendix 5 The conditional direct RDF

The conditional direct RDF $R_{X|B|}(D)$ determines the minimum achievable rate $R$ for distortion $D \geq 0$ in the compression scheme, where the encoder observes $X$ directly (i.e., without CS), and $B$ is available as SI at the encoder and decoder. It is defined as (cf. (94))

$$R_{X|B|}(D) = \min_{\left\{ f(\hat{X}|x,b) \right\}_{D_{x,k}=D}} \frac{1}{N} I(X;\hat{X}|B)$$

(128a)

where the optimization is over the conditional PDFs $f(\hat{x}|x,b), s = 1, \ldots, |\mathcal{D}|$. The conditional mutual information between $X$ and $\hat{X}$ given $B$ is

$$I(X;\hat{X}|B) = \sum_{s=1}^{\left|\mathcal{B}\right|} p(b_s) I(X;\hat{X}|B = b_s)$$

(128b)

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

$$\mathbb{E}\left[d(X,\hat{X})\right] = \sum_{s=1}^{\left|\mathcal{B}\right|} p(b_s) \mathbb{E}\left[d(X,\hat{X})\big|B = b_s\right]$$

$$= \sum_{s=1}^{\left|\mathcal{B}\right|} p(b_s) \int_{\mathbb{R}^K} f(x|b_s) f(\hat{x}|x,b_s) |d(x,\hat{x})|dx\,d\hat{x}.$$  

(128c)

Following the steps analogous to those for $R_{X|B|}^{\text{rem}}(D)$, the conditional direct RDF is given as (cf. (110))

$$R_{X|B|}(D) = \min_{\left\{ d_{x,k} \right\}_{D_{x,k}=0,k=1,\ldots,K,s=1,\ldots,|\mathcal{D}|}} \frac{1}{N} \sum_{s=1}^{\left|\mathcal{B}\right|} p(b_s) \sum_{k=1}^{K} \max \left\{ 0, \frac{1}{2} \log \frac{\lambda_{s,k}}{d_{x,k}} \right\}$$

(129)

where $d_{x,k}, k = 1, \ldots, K, s = 1, \ldots, |\mathcal{D}|$, are the optimization variables; $\lambda_{s,1} \geq \ldots \geq \lambda_{s,K} > 0$ are the eigenvalues of the covariance matrix $\Sigma_X = \bar{Q}_s \bar{\Lambda}_s \bar{Q}_s^T$, where the columns of $\bar{Q}_s \in \mathbb{R}^{K \times K}$ are the eigenvectors of $\Sigma_X \in \mathbb{S}^{K}_{++}$, and $\bar{\Lambda}_s \triangleq \text{diag}(\lambda_{s,1}, \ldots, \lambda_{s,K})$. 

193
<table>
<thead>
<tr>
<th>No.</th>
<th>Author(s)</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>660</td>
<td>Tervo, Oskari</td>
<td>Transceiver optimization for energy-efficient multiantenna cellular networks</td>
</tr>
<tr>
<td>661</td>
<td>Menberu, Meseret Walle</td>
<td>Hydrology of peat-dominated headwater catchments: theories and empirical analysis of the impacts of anthropogenic disturbance</td>
</tr>
<tr>
<td>662</td>
<td>Hiitava, Anne</td>
<td>Electrical behaviour of submerged arc furnace’s charge materials</td>
</tr>
<tr>
<td>663</td>
<td>Lappalainen, K. Matti</td>
<td>Itämeren rehevöitymisen uudistettu diagnoosi ja paradigma</td>
</tr>
<tr>
<td>664</td>
<td>Ahmad, Ijaz</td>
<td>Improving software defined cognitive and secure networking</td>
</tr>
<tr>
<td>665</td>
<td>Layemo, Ayotunde Oluwaseun</td>
<td>High speed moving networks in future wireless systems</td>
</tr>
<tr>
<td>666</td>
<td>Kaleva, Jarlko</td>
<td>Decentralized multiantenna transceiver optimization for heterogeneous networks</td>
</tr>
<tr>
<td>667</td>
<td>Hänninen, Tuomo</td>
<td>Detection algorithms and FPGA implementations for SC-FDMA uplink receivers</td>
</tr>
<tr>
<td>668</td>
<td>Huotari, Jori</td>
<td>Vanadium oxide nanostructures and thin films for gas sensor applications</td>
</tr>
<tr>
<td>669</td>
<td>Anttila, Severi</td>
<td>Influence of minor elements on some weldability issues of intermediate purity stabilized ferritic stainless steels</td>
</tr>
<tr>
<td>670</td>
<td>Hartmann, Robert</td>
<td>Flotation using cellulose-based chemicals</td>
</tr>
<tr>
<td>671</td>
<td>Porambage, Pawani</td>
<td>Lightweight Authentication and Key management of Wireless Sensor Networks for Internet of Things</td>
</tr>
<tr>
<td>672</td>
<td>Bibikova, Olga</td>
<td>Plasmon-resonant gold nanoparticles for bioimaging and sensing applications</td>
</tr>
<tr>
<td>673</td>
<td>Venkatraman, Ganesh</td>
<td>Traffic-aware resource allocation for multi-antenna OFDM systems</td>
</tr>
<tr>
<td>674</td>
<td>Isokangas, Elina</td>
<td>Quantifying the groundwater dependence of boreal ecosystems using environmental tracers</td>
</tr>
<tr>
<td>675</td>
<td>Tervazäkäi, Petri</td>
<td>Reaction and mass transfer kinetics in multiphase bioreactors: Experimental and modelling studies</td>
</tr>
</tbody>
</table>

Book orders:
Granum: Virtual book store
http://granum.uta.fi/granum/
DISTRIBUTED COMPRESSED
DATA GATHERING
IN WIRELESS SENSOR
NETWORKS

Markus Leinonen