

ALMA MATER STUDIORUM — UNIVERSITÀ DI BOLOGNA

Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione
"Guglielmo Marconi" — DEI

Dottorato di Ricerca in

INGEGNERIA ELETTRONICA, TELECOMUNICAZIONI
E TECNOLOGIE DELL'INFORMAZIONE

Ciclo XXXIV

Settore Concorsuale: 09/E3 — ELETTRONICA

Settore Scientifico Disciplinare: ING-INF/01 — ELETTRONICA

ALGORITHMS AND SYSTEMS FOR IOT AND EDGE COMPUTING

Presentata da:

Alex MARCHIONI

Coordinatore Dottorato:

Prof. Aldo ROMANI

Supervisore:

Prof. Riccardo ROVATTI

Esame Finale Anno 2022

*“Walking down a path without knowing where you really want to go...
Having a grace period like that isn't so bad.”*

Hatake Kakashi

ALMA MATER STUDIORUM — UNIVERSITY OF BOLOGNA

Abstract

School of Engineering
Department of Electrical, Electronic, and Information Engineering “Guglielmo
Marconi” (DEI)

Doctor of Philosophy

Algorithms and Systems for IoT and Edge Computing

by Alex MARCHIONI

The idea of distributing the signal processing along the path that starts with the acquisition and ends with the final application has given light to the Internet of Things and Edge Computing, which have demonstrated several advantages in terms of scalability, costs, and reliability. In this dissertation, we focus on designing and implementing algorithms and systems that allow performing a complex task on devices with limited resources.

Firstly, we assess the trade-off between compression and anomaly detection from both a theoretical and a practical point of view. Information theory provides the rate-distortion analysis that is extended to consider how information content is processed for detection purposes. Considering an actual Structural Health Monitoring application, two corner cases are analysed: detection in high distortion based on a feature extraction method and detection with low distortion based on Principal Component Analysis.

Secondly, we focus on streaming methods for Subspace Analysis. In this context, we revise and study state-of-the-art methods to target devices with limited computational resources. We also consider a real case of deployment of an algorithm for streaming Principal Component Analysis for signal compression in a Structural Health Monitoring application, discussing the trade-off between the possible implementation strategies.

Finally, we focus on an alternative compression framework suited for low-end devices that is Compressed Sensing. We propose a different decoding approach that splits the recovery problem into two stages and effectively adopts a deep neural network and basic linear algebra to reconstruct biomedical signals. This novel approach outperforms the state-of-the-art in terms of quality of reconstruction and requires lower computational resources.

Contents

Abstract	v
List of Figures	xi
List of Tables	xv
List of Acronyms	xvii
List of Symbols	xix
Introduction	1
I Detection and Compression	5
1 Anomaly Detection and Compression Trade-off	7
1.1 Signal model and distinguishability	9
1.1.1 Distinguishability in anomaly-aware scenario	11
1.1.2 Distinguishability in anomaly-agnostic detection	11
1.2 Gaussian framework	12
1.2.1 Optimal rate-distortion trade-off	13
1.2.2 Pointwise distinguishability	14
1.2.3 Average anomaly	16
1.2.4 Asymptotic anomaly	17
1.3 Numerical examples	17
1.3.1 Asymptotic anomaly	18
1.3.2 Encoders	18
1.3.3 Detectors	20
Conclusion	26
2 Detection and Compression in Structural Health Monitoring	27
2.1 Structural Health Monitoring	28
2.2 Viaduct Health Monitoring system	30
2.2.1 Normal behaviour	34
2.2.2 Anomalies	34
2.3 Detection with High Distortion Compression	36
2.3.1 Time domain analysis	36

2.3.2	Frequency domain analysis	40
2.4	Detection with Low Distortion Compression	42
2.4.1	Lack and Excess of Energy Detectors	43
2.4.2	Performance in a Gaussian Framework	46
2.4.3	Performance in SHM	50
	Conclusion	53
II	Streaming Principal Component Analysis	55
3	Streaming Subspace Analysis	57
3.1	Streaming Methods Classification	59
3.2	Streaming Methods	61
3.2.1	Oja’s method	61
3.2.2	Krasulina’s method	63
3.2.3	HFRANS	63
3.2.4	PAST	64
3.2.5	Incremental SVD	65
3.2.6	GROUSE	66
3.3	Subspace analysis performance	67
3.4	Implementation on edge devices	70
3.4.1	High-end edge devices	70
3.4.2	Low-end edge devices	72
3.4.2.1	Performance	74
3.5	Conclusion	76
4	History PCA for SHM systems	79
4.1	PCA-based Compression in SHM	80
4.2	History PCA	82
4.2.1	Parameter Tuning	84
4.3	HPCA Implementation	85
4.3.1	Implementation on gateway	86
4.3.2	Implementation on sensor node	87
4.4	Experimental Results	88
4.4.1	Functional Performance	88
4.4.2	Execution time and energy consumption	90
4.5	Conclusion	93
III	Compressed Sensing Decoder	95
5	Trained Support Oracle for Compressed Signals	97
5.1	Compressed Sensing	99
5.1.1	Encoder	100

5.1.2	Decoder	102
5.1.3	Performance indexes	104
5.2	Data sets	105
5.3	Support Oracle	106
5.4	Trained CS with Support Oracle	107
5.4.1	Preliminary evidence on real signals	110
5.5	Decoder self-assessment	110
5.6	Computational requirements	115
	Conclusion	118
6	Trained Support Oracle for Compressed signal	121
6.1	Compressed Sensing with Support Oracle	121
6.2	Support Oracle for Compressed Signals	123
6.2.1	Trained Support Oracle for Compressed Signals	125
6.2.2	Experimental results for ECGs	127
6.3	Quantization-aware Decoder Architecture	129
6.3.1	Support oracle quantization	131
6.3.2	LMS-based reconstruction	133
6.3.3	Architecture design and results	133
	Conclusion	136
A	Proof of Properties	139
A.1	Proof of Property 1.1	139
A.2	Proof of Property 1.2	140
A.3	Proof of Property 1.3	142
A.4	Proof of Property 1.4	142
A.5	Proof of Property 1.5	143
A.6	Proof of Lemma A.1	147
A.7	Proof of Property 2.1	147
	Bibliography	149

List of Figures

1.1	Signal flow in the model considered to assess the trade-off between anomaly detection and compression.	9
1.2	Deviation of the anomaly from white noise depending on the signal dimension n in term of Δ_2 and Δ_∞	19
1.3	Rate distortion curves for the three encoding techniques RDC, PCC, and AEC for different levels of signal localization.	20
1.4	Distinguishability measures ζ , κ and ψ against normalized distortion d in case of RDC.	24
1.5	Distinguishability measures ζ , κ and ψ against normalized distortion d in case of PCC.	25
1.6	Distinguishability measure ψ against normalized distortion d in case of AEC.	26
2.1	blocks scheme of SHM system monitoring a viaduct. Sensors send their reading to a main cloud platform via intermediate gateway. . . .	31
2.2	Monitored viaduct and adopted SHM system.	32
2.3	3-axes acceleration signal in ordinary conditions.	35
2.4	30 s of acceleration signals for all axes that contains anomalous events.	37
2.5	Distribution of the times domain features m , s^2 , p estimated over a week of data. Warmth of colors is proportional to the population density and the bounding boxes define the regions of normal behaviour.	39
2.6	Characterization of the anomalous events in the feature domain for 30 s of acceleration signals containing anomalous events. The height of the points represents the over-threshold ratio, and size counts for the number of instances triggering the alarms.	40
2.7	x axis acceleration PSD estimated in the hour before and after the strand break event.	42
2.8	Block diagram of the Lack/Excess of Energy detectors.	44
2.9	The loss L of LoED (a) and EoED (b) as a function of k for $m = 1$, with different levels of signal localization and deviation of the anomaly from the signal. Lines are median values while shaded areas show where 99% of the values fall.	48

2.10	The loss L of LoED (a) and EoED (b) as a function of k of the projection subspace, for LL signals, a small difference $\alpha = 0.1\pi$ between the normal and anomalous statistics, and various level of averaging m . Lines are median values while shaded areas show where 99% of the values fall.	50
2.11	The tendon break, strand break and earthquake anomalies observed by means of conventional spectral analysis (PSD), both EoED and LoED ($\kappa = 20$), and scores T2 and SPE (20 principal components), for different values of m . Dashed lines indicate the expected value of the observables in normal conditions.	52
3.1	General scheme of a streaming algorithm for subspace analysis.	58
3.2	Comparison of the convergenc of the Streaming methods for subspace identification.	69
3.3	Update time on a high-end device depending on n with a constant n/k	71
3.4	Matrix arithmetic optimization techniques: (a) register blocking, (b) buffered multiplication, and (c) vector outer product and matrix addition merging.	74
3.5	Maximum update time on MCU against the signal dimension n with fixed compression ratio CR.	75
3.6	Estimation error in subspace identification on MCU with a given random Φ and noise $\nu = 10^{-3}$	77
4.1	Comparison of lossy and lossless methods applied to signals from a vibration-based SHM system.	82
4.2	HPCA performance in terms of RSNR depending on the number of internal loops m for different block-size b and number of principal components k	86
4.3	Power Spectrum Density of the x -axis acceleration signal of one of the sensors in the viaduct SHM system.	89
4.4	Peak tracking of the peak at frequency $f_{\text{ref}} \sim 39.9$ Hz over an 1 week period applied to the original signal and reconstructed with different compression matrix: PCA, floating point HPCA and fixed point HPCA. Panels (a) and (b) show the tracking of frequency and amplitude, respectively.	90
4.5	Tracking of the three main natural frequencies of a tendon of in the week before and after a tendon strand break. The tracking is applied on the PSD computed from the original signal (A), and the reconstructed signal using HPCA (B).	91
4.6	Trade-off between energy consumption and memory occupation after a full pass over the training for both RPi3 and ARTIK 710 with GT2 implementation of the HPCA algorithm and $n = 500, k = 32, M = 3$	93

5.1	The effect of reducing n on the normalized sparsity κ/n in the two examples of a synthetic ECG and a synthetic EEG signal.	102
5.2	The original underdetermined problem (a) turned into a overdetermined problem (b) by exploiting the support vector s	104
5.3	Block diagram for the Trained CS with Support Oracle including self-assessment capability.	108
5.4	TCSSO performance of TCSSO compared against OMP, BP, GAMP, and WL1 (only for ECG) with rakesness adaptation (RAK) of the sensing matrices in terms of ARSNR (a)-(c) and PCR with $\text{RSNR}_{\min} = 55$ dB (b)-(d).	109
5.5	TCSSO performance for $n = 64$ and $n = 128$	110
5.6	Examples of reconstruction of real-world signals compared with the original waveforms with $n = 64$ and $m = 32$	111
5.7	Self-assessment mechanism in decoders based on a support oracle.	112
5.8	The relationship between RSNR and RMNR for ECG signals for $n = 64$ and $\kappa = 16$ with respective logarithmic histograms above and on the right.	114
5.9	Probability for GAMP decoder to reconstruct ECG and EEG signals with RSNR higher than TCSSO in case of TCSSO failure.	116
6.1	RSNR against the number of ones in z for different sensing matrices with $\text{ISNR} = 34$ dB.	125
6.2	Structure of the DNN implementing the support oracle.	126
6.3	ARSNR against CR for TOSC along with the ideal oracle (SOC) compared with the standard decoder (BPDN). The encoder stage follows Rak-CS and the standard CS encoder coupled with BPDN is provided as a reference.	128
6.4	Probability density functions of RSNR values with $\text{CR} = 4$ ($m = 32$) for the considered system configurations.	129
6.5	Block diagram of the overall system including CS encoder, measurements dispatch and the proposed TSOC decoder. The arrows represent digital quantities and for each of them the number of bits is highlighted.	130
6.6	Combination of the bathtub and cosine regularization terms.	133
6.7	Performance of the decoder with ideal support oracle for different values of the hyper-parameters q , b_s and b_y	135
6.8	Probability density functions of RSNR values with $\text{CR} = 4$ ($m = 32$) for the floating-point TSOC and the quantized TSOC (qTSOC) with either the pseudo-inversion (PINV) or the LMS filter as well as the case of post-quantization (post-qTSOC).	136
6.9	Examples of windows of ECG signals reconstructed employing qTSOC+LMS with $\text{CR} = 4$ ($m = 32$) compared with the originals.	137

List of Tables

1.1	Composition and dimension of the data sets employed for the training (when required) and the performance assessment of the detectors. . . .	21
2.1	Devices adopted for sensors and gateways in the viaduct health monitoring system.	33
2.2	Alarm thresholds obtained in training for the reference sensor.	39
2.3	Frequency of the peaks detected in the PSD of the acceleration signal before and after the strand break.	42
3.1	Classification of methods for streaming Subspace Analysis	62
3.2	Algorithm parameters, tuned value with corresponding ranges and update times (average and std) for the RPi edge-oriented implementation	68
3.3	Performance per update on STM32H743ZIT (rev. V) @1.8 V, 480 MHz, cache ON ($n = 100, m = 10$)	75
3.4	Memory requirements rule for each method (memory example with $n = 100, m = 10, 32$ -bit scalars)	76
4.1	Specs of the target gateway devices.	86
4.2	Peak detection performance over 1 week in terms of estimation error for frequency (err_f) and amplitude (err_A).	89
4.3	Comparison of GT1 and GT2 implementations in terms of execution time and energy consumption on Raspberry Pi 3 with $n = 500, b = 1, k = 32, M = 3$	91
4.4	Comparison of RPi3 with ARTIK 710 in terms of execution time [ms] of HPCA with GT2 implementation and $n = 500, b = 1, k = 32, M = 3$	92
4.5	Comparison between SNS1 and SNS2 implementation of HPCA algorithm on STM32F405RG with $n = 100, b = 1, M = 3$	93
5.1	Effect of the increase of n from 64 to 128 in term of ARSNR, computational overhead measured as AC/sample and memory footprint (# entries of A^\pm).	117
6.1	Number of parameters of TSOC for different values of m and its performance in terms of P, TP, TPR, TNR and ACC estimated as the mean over the data set.	128
6.2	Hyper-parameters for the low-resources TSOC implementation	134

List of Acronyms

AE	AutoEncoder
AUC	Area Under the ROC Curve
ARSNR	Average Reconstruction Signal to Noise Ratio
BLAS	Basic Linear Algebra Subprograms
BP	Basis Pursuit
BPDN	Basis Pursuit with DeNoising
cdf	cumulative density function
CMSIS	Common Microcontroller Software Interface Standard
CoSaMP	Compressive Sampling Matching Pursuit
CR	Compression Ratio
CS	Compressed Sensing
DNN	Deep Neural Network
DSP	Digital Signal Processing
ECG	ElectroCardioGraphy
ECG	ElectroEncephaloGraphy
ECG	ElectroMyoGraphy
EVD	Eigenvalue Decomposition
FFT	Fast Fourier Transform
FNR	False Negative Rate
FPR	False Positive Rate
FPU	Floating Point Unit
GAMP	Generalized Approximate Message Passing
GROUSE	Grassmaniann Rank-One Update Subspace Estimation
HFRANS	Housholder Rayleigh quotient-based Adaptive Noise Subspace
HPCA	History Principal Component Analysis
ISNR	Intrinsic Signal to Noise Ratio
ISVD	Incremental Singular Value Decomposition
LAPACK	Linear Algebra Package
LMS	Least Mean Squares
MAC	Multiply Accumulate
MCU	MicroController Unit
MQTT	MQ Telemetry Transport
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
MSA	Minor Subspace Analysis

NN	Neural Network
OMP	Orthogonal Matching Pursuit
PAST	Project Approximation Subspace Tracking
PCA	Principal Component Analysis
PCR	Probability of Ccorrect Reconstruction
pdf	probability density function
PSA	Principal Subspace Analysis
PSD	Power Spectral Density
ReLU	Rectified Linear Unit
RMNR	Reconstruction Measurements to Noise Ratio
ROC	Receiving Operating Characteristic
RSNR	Reconstruction Signal to Noise Ratio
SA	Subspace Analysis
SHM	Structural Health Monitoring
SNR	Signal to Noise Ratio
SPGL1	Spectral Projected Gradient for L1 Minimization
SReLU	Saturated Rectified Linear Unit
std	standard deviation
SVD	Singular Value Decomposition
TNR	True Negative Rate
TPR	True Positive Rate

List of Symbols

\mathbb{N}	set of the natural numbers
\mathbb{Z}	set of the integer numbers
\mathbb{R}	set of the real numbers
I_n	identity matrix with dimension $n \times n$
tr	trace of a matrix
\cdot^\top	transposition operator
$\Pr\{\cdot\}$	probability
$\mathbb{E}[\cdot]$	expectation
$\text{diag}(\times)$	diagonal matrix with \times as diagonal vector
sign	sign function
span	span of a set of vectors
$ \cdot $	absolute value
$\ \cdot\ $	norm
Σ	covariance matrix
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and covariance Σ
$\mathcal{U}(\times)$	uniform distribution on the set \times
\mathcal{L}_\times	Localization of the signal \times
\cdot^\dagger	Moore-Penrose pseudo-inversion operator
x	input signal
n	input signal dimension
y	compressed signal
\hat{x}	recovered signal
f_\times	probability density function of \times
F_\times	cumulative distribution function of \times
ok	normal behaviour
ko	anomaly

Introduction

The world in which we are currently living and more so the one we are designing for tomorrow is based on an interweaving of physical systems and information flows [5]. One of the most relevant and practical applications of such an interweaving is *monitoring*, which consists in the continuous collection of information from the physical world [133, 96, 115, 179, 51], let it be a smart city, a human body, an infrastructure, to name a few. The general architecture of a monitoring system comprises sensor nodes deployed close to the physical phenomena to monitor, data collectors that gather the sensor readings and processing units that extract information from the data to perform a task, make decisions, or enable a final service.

Sensor nodes are, in general, small devices that need to be low cost, as non-intrusive and self-sufficient as possible, and their readings must be accessible so that they can be deployed in large numbers with little cost for both installation and maintenance. These requirements imply limits on the geometry dimensions, weight, and energy budget, leading to a shortage of computational resources for data processing. That is the reason why, traditionally, sensor nodes are solely devoted to the acquisition and transmission of the signal. However, advances in wireless communication, energy harvesting, smart power management, and ultra-low power processors have enabled sensor nodes to process signals and transmit the extracted information through the internet to make decisions, i.e., following the Internet of Things (IoT) paradigm [195], they are Things on the Internet.

Moreover, clusters of sensors may send their readings either by radio or wired links to a middle layer that may be a single device as small as a sensor node, or it may be organized as a hierarchical structure in case of complex systems [72]. However implemented, this is a physically localized layer of processing elements that interposes between the sensing and central processing units, often interpreted by a cloud service. This layer takes the name of the edge of the cloud [167, 195]. Even though computational resources available at the edge are typically much smaller than those on the cloud service, they still provide the opportunity for local processing, i.e., Edge Computing [167].

In conventional IoT frameworks for monitoring, the whole data is uploaded to the central processing unit. This process may be critical for several reasons. The connection between sensors and central unit may require high bandwidth to transfer all data streams and guarantee stability to avoid data losses. Once in the central node, data must be stored and processed, and this may lead to high managing costs

in the case of cloud services. Moreover, the introduced delay may be critical in time-sensitive applications that require prompt actions in response to occurring events. In addition, some applications may need a certain level of privacy or security that is harder to guarantee in a system that transfers data over the internet and centralize storage and processing.

All these issues can be properly addressed by distributing the processing of the signal along the signal chain, i.e., by moving the processing closer to where the data is produced. Indeed, IoT and Edge Computing provide the technologies and the strategies to effectively design systems and algorithms that effectively process information and perform a complex task on devices with limited resources.

In this dissertation, we focus on designing and implementing algorithms and systems that exploit the advantages provided by IoT and Edge Computing technologies. We focus on monitoring systems applications where local processing for tasks like signal compression and anomaly detection is fundamental. This dissertation is organized as follows:

Part I addresses the trade-off between compression and anomaly detection from both a theoretical and a practical point of view. The aim is to show how the optimal encoder in the rate-distortion sense does not address this trade-off effectively, and we pursue a theoretical framework that can explain trends characterizing real applications.

In Chapter 1 we propose a theoretical framework that extends the rate-distortion analysis developed in the field of Information theory by including the concept of distinguishability to model the possibility for a detector to discriminate between ordinary and anomalous signal instances. Thanks to a Gaussian assumption, it is possible to derive some analytical results, which are then confirmed by numerical evidence.

On the other hand, Chapter 2 deals with two practical cases, each representing a corner case of the compression-detection trade-off, i.e., detection with high and low distortion compression. As a use case, we consider an actual Structural Health Monitoring application concerning the assessment of the condition of an Italian highway viaduct. The two approaches analyzed consist of a detector based on feature extraction that distorts the signal so that recovery is improbable, and in a detector exploiting Principal Component Analysis, which allows preserving most of the signal information.

Part II focuses on methods that perform the Subspace Analysis, of which Principal Component Analysis is a particular case, in a streaming fashion, i.e., considering the signal as a stream and processing one instance at a time. The main objective consists in providing both an overview and a use case of this class of methods aiming at low-resources devices as a target.

In Chapter 3, we revise the state-of-the-art methods providing classification and coherent framework in which discuss the several connections between the algorithm and the relations with most of the subtle variants currently available in the Literature. The streaming methods are then compared to provide a picture of the pros and cons of each method in the task of subspace identification. The final focus is on their implementation on devices representative of the ones typical of acquiring systems characterized by constraints on the computational resources.

This part also includes Chapter 4 which reports the employment of a streaming method for Principal Component Analysis in the same real Structural Health Monitoring system considered in the previous part of this dissertation. The application aims to provide the monitoring system with the ability to tune the signal compression locally by tailoring the streaming method for the involved devices. Following a discussion of compression methods for acquisition systems, different implementation strategies have been investigated, and the trade-off between them is assessed.

Part III focuses on an alternative compression framework whose encoding procedure is suited for devices with limited resources. This framework is the Compressed Sensing, whose theory is based on the sparsity and incoherence assumptions that are so broad to find application in many acquisition scenarios involving physical phenomena. In particular, this part of this dissertation focuses on the decoder and biomedical signals.

In particular, Chapter 5 proposes an innovative idea that splits the traditional recovery problem into two stages, allowing the replacement of iterative algorithms with a deep neural network and basic linear algebra operations. This novel approach outperforms the state-of-the-art decoders in terms of quality of reconstruction and provides the decoder with a self-assessment capability that enables the estimation of the recovery performance. Moreover, the implementation requires computational resources comparable with the most lightweight methods available in the Literature.

The work is then extended in Chapter 6 where we generalize the model of the signal in order to consider signals that satisfy the sparsity assumption only approximately. This extension leads to a slight change in the model and confirms the performance improvement with respect to the competitors. This chapter also discusses the implementation of the decoder in a context that considers only fixed-point arithmetic. We show that the resulting performance loss can be limited by adopting quantization-aware training techniques and iterative methods that replace the critical operation in the sense of numerical stability.

Part I

Detection and Compression

Chapter 1

Anomaly Detection and Compression Trade-off

A typical monitoring system involves a physical phenomenon sensed directly or indirectly through devices that transform physical quantities into bit-streams. The generated data needs to be collected and transferred over a network to be stored and/or processed in a facility that, in general, is remote as a central server or a cloud service. Before reaching the destination, bit-streams may pass through intermediate devices, and the communication is often a bottleneck in terms of either bandwidth requirement or energy budget. For this reason, data may be compressed.

For a significant compression level, it is necessary to adopt lossy approaches that exploit the monitored signal's statistical characteristic to focus on the most informative part at the expense of the details. Hence, there is a trade-off between the distortion imposed in the encoding stage and the number of bits employed to code the signal information content. This trade-off is assessed by the well-known rate-distortion analysis developed in the information theory field.

However, the rate-distortion trade-off considers the signal information as uniformly relevant. That is not the case of data collected for a specific task in which some information content may be completely ignored. As an extreme example, the extraction of features from a signal is a process that severely distorts the signal to focus only on the information relevant for a specific task, such as classification or anomaly detection.

Moreover, for latency or privacy reasons, some computational tasks may benefit from their deployment at the edge of the cloud [167], i.e., the same devices devoted to data acquisition and dispatch. In general, one may want to analyze the compressed stream of data locally for real-time processing but still transmit it to the cloud for offline analyses. This need is especially true when dealing, for example, with systems monitoring plants or structures in which data streams may be processed at the edge to detect possible critical events that require immediate intervention but also processed offline to analyze long-term historical trends.

Indeed, in monitoring systems, local anomaly/novelty detection is a fundamental task that identifies and recognizes something out of the ordinary in the acquired signal that may be related to an unexpected in the monitored physical phenomenon

or the monitoring system. A detector discriminates an ordinary signal from an anomaly by observing the statistical characteristics of the signal, and, since anomaly is usually completely unknown, the detector may only classify an instance as abnormal when it significantly differs from the expected behaviour.

However, lossy compression bases its effectiveness on neglecting some of the signal details that, in principle, could have been used to discriminate normal behaviours from anomalous ones. Hence, in this case, another trade-off goes in parallel to the classic rate-distortion that is the one between compression and distinguishability between normal and anomalous signals.

In this chapter, we address this multi-faceted trade-off between rate, distortion, and distinguishability with the information machinery employed in the classical rate-distortion analysis.

In this sense, the pursued analysis here resembles the information-bottleneck scheme [178, 170]. In that scheme, distortion is replaced with a general criterion that indicates the features that should be preserved when compressing the original signal's information about a second (suitably introduced) signal. However, our discussion takes a different direction as, in the considered scenario, the statistics of the anomaly may be completely unknown. Even if we have priors on that, the analysis needs to be generalized for cases in which the mutual information between normal and anomalous signals is null.

For the same reason, the analysis we propose is also different from other modifications of classical rate-distortion theory that replace energy-based distortion with perceptive criteria [33, 32].

It is also worthwhile to mention [159, 168], in which the original signal is assumed to be characterized by some parameters (e.g., mean) and the authors study how the estimation of such parameters is affected by lossy compression.

Other works also attempt to pair rate and distortion with additional figures of merit considering relevant features of the system. An example is in [70], in which the analysis of rate-distortion is paired with computational effort considerations in the case of video coding based on wavelet decomposition.

In this chapter, we extend the rate/distortion analysis to discuss the trade-off with anomaly detection. In detail, Section 1.1 presents the signal model as well as defines two distinguishability measures that model the scenarios in which the detector may either know the statistical characterization of the anomaly or not. Section 1.2 specializes the model to the case of signals distributed as Gaussian random vectors, revisits well-known results on Gaussian rate-distortion analysis, and derives some analytical results about distinguishability for high-dimensional signals in both the finite and asymptotic regime. Finally, Section 1.3 reports some numerical evidence to analyze how the distinguishability measures are affected by distortion in the case of optimal and suboptimal encoding strategies in the rate-distortion sense. Theoretical curves anticipate trends observed in the performance of practical decoder, showing that compression optimized for the rate-distortion trade-off, in general, does not

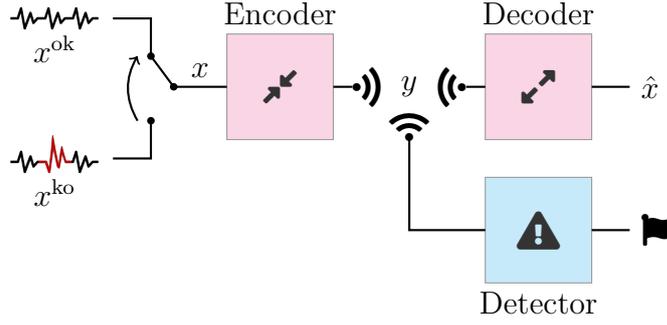


FIGURE 1.1: Signal flow in the model considered to assess the trade-off between anomaly detection and compression.

necessarily address distinguishability at best.

1.1 Signal model and distinguishability

The observable signal is modelled as a discrete-time, n -dimensional stochastic process $x[t] \in \mathbb{R}^n$ which, at any t , is either normal or anomalous. This bi-modal behaviour is modelled as two independent sources producing stationary, stochastic processes $x^{\text{ok}}[t] \in \mathbb{R}^n$ and $x^{\text{ko}}[t] \in \mathbb{R}^n$ with different probability density functions (pdfs) $f_x^{\text{ok}} : \mathbb{R}^n \mapsto \mathbb{R}^+$ and $f_x^{\text{ko}} : \mathbb{R}^n \mapsto \mathbb{R}^+$, where ok stands for normal and ko for anomalous.

$$x[t] = \begin{cases} x^{\text{ok}}[t] \sim f_x^{\text{ok}} & \text{if ok} \\ x^{\text{ko}}[t] \sim f_x^{\text{ko}} & \text{if ko} \end{cases} \quad (1.1)$$

Note that this model fits both a case in which the entries of $x[t]$ are the subsequent samples of a signal in the t -th window, and one in which they are the simultaneous readings of different sensors at time step t , as well as those intermediate cases in which the vector $x[t]$ is made of readings from different sources in a set of subsequent time steps.

The observable $x[t]$ is the input of an encoding stage which produces a compressed version $y[t]$. The output of the encoder may be either decompressed into $\hat{x}[t] \in \widehat{\mathbb{R}}^n$, where $\widehat{\mathbb{R}}^n \subset \mathbb{R}^n$ is a finite subset of \mathbb{R}^n , or fed to a detector with the objective of discriminating between normal or anomalous behaviour, i.e., to decide whether the original signal is $x[t] = x^{\text{ok}}[t]$ or $x[t] = x^{\text{ko}}[t]$. A schematic representation of signal flow we consider is reported in Figure 1.1.

Since the compression mechanism is considered a lossy procedure, the encoder is not injective. At the same time, the decoder is assumed to be injective so that, in abstract terms, the detector may be thought to work on either $y[t]$ or $\hat{x}[t]$.

The encoder is tuned on the normal behaviour, meaning compression is designed assuming $x[t] = x^{\text{ok}}[t]$ and thus $\hat{x}[t] = \hat{x}^{\text{ok}}[t]$ for every t . The average distortion D

introduced by the encoding stage is the following:

$$D = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \left\| x^{\text{ok}}[t] - \hat{x}^{\text{ok}}[t] \right\|^2 = \mathbf{E} \left[\left\| x^{\text{ok}}[t] - \hat{x}^{\text{ok}}[t] \right\|^2 \right] \quad (1.2)$$

Since $\widehat{\mathbb{R}}^n$ is finite so that one can assign a digital word with $b[t]$ bit to each possible values of $\hat{x}[t]$. Since the decoder is injective the same holds for $y[t]$. Considering a stream of compressed signals, it is possible to define the average rate as

$$R = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} b[t] = \mathbf{E} [b[t]] \quad (1.3)$$

Distortion and rate are two elements of a trade-off. The lower the number of available bits representing the information, the higher the distortion. The Pareto frontier for this trade-off is called rate-distortion function ρ and it is defined as the minimal rate given a distortion budget δ , i.e.,

$$\rho(\delta) = \inf R \text{ s.t. } D \leq \delta \quad (1.4)$$

To identify such a function, it is classical [56, Chapter 9] to consider the joint probability of $\hat{x}[t]$ and $x[t]$ denoted as $f_{\hat{x},x} : \mathbb{R}^{2n} \mapsto \mathbb{R}^+$. In such a way, it is possible to model the cascade of encoding and decoding stage as the conditional pdf of $\hat{x}[t]$ given the occurrence of $x[t]$ indicated with $f_{\hat{x}|x}$, so that $f_{\hat{x},x}(\alpha, \beta) = f_{\hat{x}|x}(\alpha, \beta) f_x(\beta)$.

As the compression is tuned on the normal behaviour $x[t] = x^{\text{ok}}[t]$, the average distortion is given by

$$D = \int_{\mathbb{R}^{2n}} \|\alpha - \beta\|^2 f_{\hat{x},x}^{\text{ok}}(\alpha, \beta) d\alpha d\beta \quad (1.5)$$

where $f_{\hat{x},x}^{\text{ok}}(\alpha, \beta) = f_{\hat{x}|x}(\alpha, \beta) f_x^{\text{ok}}(\beta)$ from which one may derive the marginal pdf of the decompressed signal $f_{\hat{x}}^{\text{ok}}(\alpha) = \int_{\mathbb{R}^n} f_{\hat{x},x}^{\text{ok}}(\alpha, \beta) d\beta$.

Considering the classical distortion theory (e.g., [56, Chapter 13]) and denoting $\mathcal{I}(x^{\text{ok}}; \hat{x}^{\text{ok}})$ as the mutual information between $\hat{x}^{\text{ok}}[t]$ and $x^{\text{ok}}[t]$ [56, Chapter 8], one can write the rate-distortion function as:

$$\rho(\delta) = \inf_{f_{\hat{x}|x}} \mathcal{I}(\hat{x}^{\text{ok}}; x^{\text{ok}}) \text{ s.t. } D \leq \delta \quad (1.6)$$

With the introduction of anomalies, the model becomes more complex. One can notice that anomalies are compressed with the encoder tuned on normal behaviour modelled by $f_{\hat{x}|x}$. In this case, the input and output are characterized by the joint pdf $f_{\hat{x},x}^{\text{ko}}(\alpha, \beta) = f_{\hat{x}|x}(\alpha, \beta) f_x^{\text{ko}}(\beta)$ and the marginal pdf $f_{\hat{x}}^{\text{ko}}(\alpha) = \int_{\mathbb{R}^n} f_{\hat{x},x}^{\text{ko}}(\alpha, \beta) d\beta$.

A detector observes the compressed version of the original signal that, assuming the decoder injective, is equivalent of observing the decompressed signal. Therefore, to discriminate between normal and anomalous behaviour, the detector works on the difference between the distributions $f_{\hat{x}}^{\text{ok}}$ and $f_{\hat{x}}^{\text{ko}}$.

This dissertation proposes two information-theoretic measures to quantify the difference between the two distributions that we will name distinguishability. One measure considers the scenario in which the detector knows f_x^{ok} and f_x^{ko} (anomaly-aware scenario) and one for the case in which only f_x^{ok} is known (anomaly-agnostic scenario).

To define both measures, it is convenient to consider the following functional:

$$L(x'; x'') = - \int_{\mathbb{R}^n} f_{x'}(\alpha) \log_2 [f_{x''}(\alpha)] d\alpha \quad (1.7)$$

which represents the average rate for a source characterized by the pdf $f_{x'}$ with a code optimized for a source with pdf $f_{x''}$. Note that $H(x) = L(x; x)$ is equal to the differential entropy of x [56, Chapter 8].

1.1.1 Distinguishability in anomaly-aware scenario

When the detector knows the statistical characterization of both normal and anomalous sources, f_x^{ok} and f_x^{ko} respectively, the distinguishability between f_x^{ok} and f_x^{ko} may be measured as

$$\kappa = L(\hat{x}^{\text{ko}}; \hat{x}^{\text{ok}}) - L(\hat{x}^{\text{ko}}; \hat{x}^{\text{ko}}) = \int_{\mathbb{R}^n} f_x^{\text{ko}}(\alpha) \log_2 \left[\frac{f_x^{\text{ko}}(\alpha)}{f_x^{\text{ok}}(\alpha)} \right] d\alpha \quad (1.8)$$

that coincides to the Kullback-Leibler divergence of the anomalous decompressed signal \hat{x}^{ko} from normal decompressed signal \hat{x}^{ok} .

The measure κ models a scenario in which the detector knows the optimal code for both normal and anomalous sources. Then, the detector observes the anomalous distorted stream $\hat{x}^{\text{ko}}[t]$ and computes the increase in the encoding rate due to the employment of the code optimized for the normal source. As a result, large values for κ correspond to systems with high detection capability.

1.1.2 Distinguishability in anomaly-agnostic detection

When there is no knowledge on the anomaly and only f_x^{ok} is known, the distinguishability may be measured as

$$\zeta = L(\hat{x}^{\text{ko}}; \hat{x}^{\text{ok}}) - L(\hat{x}^{\text{ok}}; \hat{x}^{\text{ok}}) = \int_{\mathbb{R}^n} [f_x^{\text{ko}}(\alpha) - f_x^{\text{ok}}(\alpha)] \log_2 f_x^{\text{ok}}(\alpha) d\alpha \quad (1.9)$$

The measure ζ models a scenario in which the detector only knows the code optimized for the normal source. An anomaly is then detected when the encoding rate is different from the typical encoding rate of the normal source.

Note that, in the case of an anomaly source yielding a lower encoding rate compared to normal signals, ζ may assume negative values. For this reason, when a positive quantity is needed, $|\zeta|$ is considered.

In the anomaly-aware and anomaly-agnostic case, we may want to assess average performance when the anomalies are randomly drawn from a certain set of possible behaviours.

1.2 Gaussian framework

In this section the model is specialized for the case in which the signals have Gaussian distribution.

An n -dimensional Gaussian random vector $x \sim \mathcal{N}(\mu, \Sigma)$, i.e., with mean $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$, is characterized by the following probability density function:

$$\mathcal{N}(\xi; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2}(\xi - \mu)^\top \Sigma^{-1}(\xi - \mu)\right) \quad (1.10)$$

where $|\cdot|$ indicates the determinant of its matrix argument.

For any time instant t , both the normal $x^{\text{ok}}[t]$ and anomalous $x^{\text{ko}}[t]$ signals are assumed to be n -dimensional random vectors with Gaussian distribution $x^{\text{ok}}[t] \sim \mathcal{N}(0, \Sigma^{\text{ok}})$ and $x^{\text{ko}}[t] \sim \mathcal{N}(0, \Sigma^{\text{ko}})$ where Σ^{ok} and Σ^{ko} are two $n \times n$ covariance matrices. It is also assumed that, $\forall t_1, t_2$ such that $t_1 \neq t_2$, the random vectors are independent.

With no loss of generality, one may assume Σ^{ok} diagonal as it is sufficient to change coordinates to $x[t]$ (and thus to both $x^{\text{ok}}[t]$ and $x^{\text{ko}}[t]$) by applying a proper orthonormal transformation. This means that $\Sigma^{\text{ok}} = \text{diag}(\lambda_1^{\text{ok}}, \dots, \lambda_n^{\text{ok}})$ with $\lambda_1^{\text{ok}} \geq \lambda_2^{\text{ok}} \geq \dots \geq \lambda_n^{\text{ok}} \geq 0$.

In general, $\Sigma^{\text{ok}} \neq \Sigma^{\text{ko}}$, but we will assume $\text{tr}(\Sigma^{\text{ok}}) = \text{tr}(\Sigma^{\text{ko}}) = n$. This constraint leads to signal instances that, on average, have l_2 norm equal to n , i.e., on average, each element of the random vector contributes with a unit energy.

The zero-mean assumption and the constraint on the trace of the covariance matrices are required to focus on the scenarios where the anomalies cannot be distinguished from the normal signals by simply observing the mean or the energy.

First, the well-known results for rate-distortion trade-off in the Gaussian framework are revised. Then, the Gaussian assumption is exploited to specialize the model defined in 1.1 and derive the distribution of the normal and anomalous signal in the case of an encoder tuned for the normal signal in the sense of rate-distortion optimality.

Afterwards, the assumption of Gaussian signals is exploited to specialize distinguishability in the pointwise and average case. Some important considerations are also shown for the asymptotic characterization of anomalies in the high-dimensional case.

1.2.1 Optimal rate-distortion trade-off

The derivation of the rate rate-distortion function $\rho(\delta)$ in the Gaussian case is well-known [56, Chapter 13].

Considering the scalar case $n = 1$, the distortion constraint is quadratic, and it is possible to derive that, given the constraint $D \leq \delta$, the minimum achievable rate has the lower bound

$$I(\hat{x}^{\text{ok}}; x^{\text{ok}}) \geq H(x^{\text{ok}}) - \frac{1}{2} \log_2(2\pi e\delta) \quad (1.11)$$

This lower bound is achievable only when the encoding is such that $x^{\text{ok}} - \hat{x}^{\text{ok}}$ is a Gaussian random variable [56, Chapter 13]. Assuming x^{ok} Gaussian with variance $\sigma_{\text{ok}}^2 = \lambda^{\text{ok}}$, its differential entropy is $H(x^{\text{ok}}) = \frac{1}{2} \log_2(2\pi e\sigma_{\text{ok}}^2)$. This leads to the following rate-distortion function:

$$\rho(\delta) = \begin{cases} \frac{1}{2} \log_2 \frac{\sigma_{\text{ok}}^2}{\delta} & \text{if } 0 \leq \delta \leq \sigma_{\text{ok}}^2 \\ 0 & \text{otherwise} \end{cases} \quad (1.12)$$

Generalizing to the n -dimensional case, in which x^{ok} is a vector of independent Gaussian variables, one must consider that the total distortion is the sum of the distortions imposed to each component. This aspect leads to the well-known water-filling result [56, Theorem 13.3.3]

$$\rho = \frac{1}{2} \sum_{j=1}^n \log_2 \left(\frac{\lambda_j^{\text{ok}}}{\min\{\theta, \lambda_j^{\text{ok}}\}} \right) \quad (1.13)$$

$$\delta = \sum_{j=1}^n \min\{\theta, \lambda_j^{\text{ok}}\} \quad (1.14)$$

where $\theta \in [0, \lambda_1^{\text{ok}}]$ is the parameter representing the water-level.

Note that the lower water level components are entirely distorted and do not count for the rate, while the components that survive are distorted with energy equal to the water level.

To track the effect of water filling, we consider I_n as the n -dimensional identity matrix and define the following matrices:

- $T_\theta = \min\{I_n, \theta(\Sigma^{\text{ok}})^{-1}\}$, whose diagonal elements account for the fraction of energy cancelled by distortion along each component;
- $S_\theta = I_n - T_\theta = \max\{0, I_n - \theta(\Sigma^{\text{ok}})^{-1}\}$, whose diagonal elements account for the fraction of energy that survives distortion along each component;
- $R_\theta = S_\theta T_\theta$.

Along the rate-distortion curve, the relation between compression and distortion is due to an encoding mapping $f_{\hat{x}|x}$ that is generally stochastic and produces a signal

with a pdf $f_{\hat{x}}^{\text{ok}}$. Both $f_{\hat{x}|x}$ and $f_{\hat{x}}^{\text{ok}}$ are given by the following Property, whose proof is in the appendix [A.1](#).

Property 1.1. *The optimal distortion is given by*

$$f_{\hat{x}|x}(\alpha, \beta) = \mathcal{N}\left(\alpha; \beta S_\theta, \Sigma^{\text{ok}} R_\theta\right) \quad (1.15)$$

and the optimally distorted signal has the pdf

$$\hat{x}^{\text{ok}} \sim \mathcal{N}\left(0, \Sigma^{\text{ok}} S_\theta\right) \quad (1.16)$$

The encoder characterized by $f_{\hat{x}|x}$ that optimally addresses the rate-distortion trade-off for the normal source is also employed for the anomalous signal $x^{\text{ko}}[t]$ encoded into $\hat{x}^{\text{ko}}[t]$. The pdf of $\hat{x}^{\text{ko}}[t]$ is given by the following Property, whose derivation is in the appendix [A.2](#).

Property 1.2. *If an anomalous source $x^{\text{ko}} \sim \mathcal{N}(0, \Sigma^{\text{ko}})$ is encoded with the compression scheme $f_{\hat{x}|x}$ of Property [1.1](#), then*

$$\hat{x}^{\text{ko}} \sim \mathcal{N}\left(0, S_\theta \Sigma^{\text{ko}} S_\theta + \theta S_\theta\right) \quad (1.17)$$

The distribution of the anomalous signal distorted by the encoder optimized for the normal signal in the sense of the rate-distortion trade-off has two notable corner cases.

- when the encoder introduces no distortion, i.e., $\theta \rightarrow 0^+$, Property [1.2](#) gives $\hat{x}^{\text{ko}} \sim x^{\text{ko}}$ that means that, as expected, the distribution of the anomaly is not modified.
- when the anomaly is indistinguishable from the signal $x^{\text{ko}} \sim x^{\text{ok}}$, i.e., $\Sigma^{\text{ok}} = \Sigma^{\text{ko}}$, then

$$\begin{aligned} S_\theta \Sigma^{\text{ko}} S_\theta + \theta S_\theta &= \left(S_\theta + \theta(\Sigma^{\text{ok}})^{-1}\right) \Sigma^{\text{ok}} S_\theta \\ &= \max\left\{\theta(\Sigma^{\text{ok}})^{-1}, I_n\right\} \Sigma^{\text{ok}} S_\theta = \Sigma^{\text{ok}} S_\theta \end{aligned} \quad (1.18)$$

where the last equality holds since $\max\{\theta(\Sigma^{\text{ok}})^{-1}, I_n\}$ disagrees with I_n only for the elements that are multiplied by zero by the last S_θ factor. As a result, if $x^{\text{ko}} \sim x^{\text{ok}}$, [1.17](#) corresponds to [1.16](#) and, therefore, $\hat{x}^{\text{ko}} \sim \hat{x}^{\text{ok}}$.

1.2.2 Pointwise distinguishability

Properties [1.1](#) and [1.2](#) imply that when the normal and anomalous signals are Gaussian before compression, the performance of anomaly detectors depends on how much they are able to distinguish between the two distributions in [\(1.16\)](#) and [\(1.17\)](#).

In the Gaussian framework, it is possible to obtain an expression for the distinguishability measures κ (1.8) and ζ (1.9) that only depends on the statistical characterization of the normal and anomalous signals.

Properties 1.1 and 1.2 also demonstrate that \hat{x}^{ok} and \hat{x}^{ko} are both a Gaussian n -dimensional random vectors. Depending on the water-level parameter θ , the optimal encoder cancels all the normal and anomalous components that correspond to all j for which $\lambda_j^{\text{ok}} \leq \theta$ holds, i.e., the components that in case of normal signal would be completely distorted.

Let us assume that the number of surviving components is $k_\theta = \arg \max_j \{\lambda_j^{\text{ok}} > \theta\}$, and that they are distributed according to $\mathcal{N}(0, \hat{\Sigma}_\theta^{\text{ok}})$ in case of normal signal or according to $\mathcal{N}(0, \hat{\Sigma}_\theta^{\text{ko}})$ in case of anomaly. Note that, $\hat{\Sigma}_\theta^{\text{ok}}$ is the $k_\theta \times k_\theta$ upper-left submatrix of $\Sigma^{\text{ok}} S_\theta$ in (1.16), and $\hat{\Sigma}_\theta^{\text{ko}}$ is the $k_\theta \times k_\theta$ upper-left submatrix of $S_\theta \Sigma^{\text{ko}} S_\theta + \theta S_\theta$ in (1.17).

Then, before considering κ (1.8) and ζ (1.9), one may specialize the expression of the functional $L(x'; x'')$ defined in (1.7) as in the following Property whose derivation is in the Appendix A.3.

Property 1.3. *If $f_{x'}(\alpha) = \mathcal{N}(\alpha; 0, \Sigma')$ and $f_{x''}(\alpha) = \mathcal{N}(\alpha; 0, \Sigma'')$ then*

$$L(x'; x'') = \frac{1}{2 \log 2} \left\{ \log [(2\pi)^n |\Sigma''|] + \text{tr} [(\Sigma'')^{-1} \Sigma'] \right\} \quad (1.19)$$

By properly combining the definitions in (1.8) and (1.9) with (1.19) one obtains

$$\zeta = \frac{1}{2 \log 2} \text{tr} \left[(\hat{\Sigma}_\theta^{\text{ok}})^{-1} \hat{\Sigma}_\theta^{\text{ko}} - I_{k_\theta} \right] \quad (1.20)$$

$$\kappa = \frac{1}{2 \log 2} \left(\text{tr} \left[(\hat{\Sigma}_\theta^{\text{ok}})^{-1} \hat{\Sigma}_\theta^{\text{ko}} - I_{k_\theta} \right] - \log \frac{|\hat{\Sigma}_\theta^{\text{ko}}|}{|\hat{\Sigma}_\theta^{\text{ok}}|} \right) \quad (1.21)$$

It is evident that, as expected, both measures vanish when the distorted anomaly is indistinguishable from the distorted normal signal, i.e., when $\hat{\Sigma}_\theta^{\text{ok}} = \hat{\Sigma}_\theta^{\text{ko}}$. Moreover, since both Σ^{ok} and S_θ are diagonal matrices, one may derive that $(\hat{\Sigma}_\theta^{\text{ok}})^{-1} \hat{\Sigma}_\theta^{\text{ko}}$ is the $k_\theta \times k_\theta$ upper-left submatrix of $(\Sigma^{\text{ok}})^{-1} \Sigma^{\text{ko}} S_\theta + T_\theta$.

A noteworthy particular case is when $\Sigma^{\text{ok}} = I_n$ that is when the normal signal is white. In such a case, $\theta \in [0, 1]$ and that for any $\theta < 1$, $T_\theta = \theta I_n$ and $k_\theta = n$. As a consequence, $\hat{\Sigma}_\theta^{\text{ok}} = (1 - \theta) I_n$, $\hat{\Sigma}_\theta^{\text{ko}} = (1 - \theta)^2 \Sigma^{\text{ko}} + \theta(1 - \theta) I_n$, and the distinguishability measures in (1.20) and (1.21) become

$$\zeta = 0 \quad (1.22)$$

$$\kappa = -\frac{1}{2 \log 2} \log \left| (1 - \theta) \Sigma^{\text{ko}} + \theta I_n \right| \quad (1.23)$$

Note that the measure ζ is null when the normal signal is distributed as Gaussian noise. This aspect is not surprising because the ζ depends only on the statistics of x^{ok} that has no exploitable structure.

1.2.3 Average anomaly

In the considered model, Σ^{ko} completely describes the distribution of the anomaly signal. In general, Σ^{ko} is unknown, so that one may be interested in the average anomaly over all the possibilities.

Let us decompose $\Sigma^{\text{ko}} = Q^{\text{ko}} \Lambda^{\text{ko}} Q^{\text{ko}\top}$, where $\Lambda^{\text{ko}} = \text{diag}(\lambda_0^{\text{ko}}, \dots, \lambda_{n-1}^{\text{ko}})$ is the matrix containing the eigenvalues and Q^{ko} is the matrix composed by the corresponding eigenvectors.

Since the model assume $\text{tr}(\Sigma^{\text{ko}}) = n$, the set of all possible λ^{ko} is

$$\mathcal{S}^n = \left\{ \lambda \in \mathbb{R}_+^n \mid \sum_{j=0}^{n-1} \lambda_j = n \right\} \quad (1.24)$$

The set of all possible Q^{ko} coincides with the set of orthonormal $n \times n$ matrices

$$\mathcal{O}^n = \left\{ U \in \mathbb{R}^{n \times n} \mid U^\top U = I_n \right\} \quad (1.25)$$

Let now $\mathcal{U}(\cdot)$ indicate the uniform distribution in the argument domain and assume that when λ^{ko} is not known, then $\lambda^{\text{ko}} \sim \mathcal{U}(\mathcal{S}^n)$ and that when Q^{ko} is not known, then $Q^{\text{ko}} \sim \mathcal{U}(\mathcal{O}^n)$, independently of λ^{ko} . Then, one may note that \mathcal{S}^n is invariant with respect to any permutation of λ_j and, since $\lambda^{\text{ko}} \sim \mathcal{U}(\mathcal{S}^n)$, also $\mathbf{E}[\lambda^{\text{ko}}]$ must be invariant with respect to the same permutations, i.e., for any j, l $\mathbf{E}[\lambda_j^{\text{ko}}] = \mathbf{E}[\lambda_l^{\text{ko}}]$. Since $\mathbf{E}[\lambda_j^{\text{ko}}]$ must be the same for any j and the sum of λ_j^{ko} is constrained to n , one must conclude that $\mathbf{E}[\Lambda^{\text{ko}}] = I_n$. This result has the following implication:

$$\mathbf{E}[\Sigma^{\text{ko}}] = \mathbf{E}[Q^{\text{ko}} \Lambda^{\text{ko}} Q^{\text{ko}\top}] = \mathbf{E}[Q^{\text{ko}} \mathbf{E}[\Lambda^{\text{ko}}] Q^{\text{ko}\top}] = \mathbf{E}[Q^{\text{ko}} Q^{\text{ko}\top}] = I_n \quad (1.26)$$

This implies that, in the considered Gaussian framework, the average anomaly consists in white noise. Then, one may compute the distinguishability measures in this specific case.

$$\bar{\zeta} = \zeta|_{\Sigma^{\text{ko}}=I_n} = \frac{1}{2 \log 2} \sum_{j=1}^{k_\theta} (\alpha_{\theta,j} - 1) \quad (1.27)$$

$$\bar{\kappa} = \kappa|_{\Sigma^{\text{ko}}=I_n} = \frac{1}{2 \log 2} \sum_{j=1}^{k_\theta} (\alpha_{\theta,j} - 1 - \log \alpha_{\theta,j}) \quad (1.28)$$

where $\alpha_{\theta,j} = \lambda_j^{\text{ok}^{-1}} (1 - \theta \lambda_j^{\text{ok}^{-1}}) + \theta \lambda_j^{\text{ok}^{-1}}$, that in this specific case corresponds to the j -th element of the diagonal of $(\hat{\Sigma}_\theta^{\text{ok}})^{-1} \hat{\Sigma}_\theta^{\text{ko}}$, which in turn consists in the $k_\theta \times k_\theta$ upper-left submatrix of $(\Sigma^{\text{ok}})^{-1} S_\theta + T_\theta$.

Note that, since ζ is linear and κ is convex [35, Chapter 3] with respect to $\hat{\Sigma}_\theta^{\text{ko}}$, and exploiting the Jensen's Inequality, it is evident that $\bar{\zeta} = \mathbf{E}[\zeta]$ and $\bar{\kappa} \leq \mathbf{E}[\kappa]$. This implies that $\bar{\zeta}$ consists in the average ζ over all possible anomalies while $\bar{\kappa}$ plays the role of lower bound for κ .

Furthermore, the simple expression of $\bar{\zeta}$ allows the derivation of the following Property whose proof is in the Appendix A.4.

Property 1.4. *If $\ell = \arg \max_l \{ \lambda_l^{\text{ok}} \geq \lambda_l^{\text{ko}} = 1 \}$, then $\exists \theta \in [0, \lambda_\ell^{\text{ok}}]$ such that $\bar{\zeta} = 0$*

Property 1.4 says that if x^{ko} is distributed as Gaussian white noise, there exists at least a critical level of distortion that makes the anomaly-agnostic detectors ineffective. Moreover, this level depends on how the energy is distributed along with the signal components.

1.2.4 Asymptotic anomaly

One may also be interested in the anomaly's distribution when the dimension of the signal increases. This aspect is addressed with the following Property, whose proof is in Appendix A.5.

Property 1.5. *If $\lambda^{\text{ko}} \sim \mathcal{U}(\mathbb{S}^n)$ and $Q^{\text{ko}} \sim \mathcal{U}(\mathbb{O}^n)$ then $\Sigma^{\text{ko}} = Q^{\text{ko}} \text{diag}(\lambda^{\text{ko}}) Q^{\text{ko}\top}$ tends to I_n in probability as $n \rightarrow \infty$.*

Property 1.5 means that white noise is not only the average anomaly (as shown by (1.26)) but also approaches the *typical* anomaly as the dimension n increases. This aspect implies that, for large n , $\zeta \simeq \bar{\zeta}$ and enjoys Property 1.4.

1.3 Numerical examples

In this section, the results derived in the theoretical analysis are compared to a quantitative evaluation of the performance of some anomaly detectors applied to compressed signals.

Following the Gaussian framework, the normal signal is assumed to be $x^{\text{ok}} \sim \mathcal{N}(0, \Sigma^{\text{ok}})$ and anomalies as $x^{\text{ko}} \sim \mathcal{N}(0, \Sigma^{\text{ko}})$.

One may expect that the performance of an anomaly detector may be affected by the non-whiteness of the normal signal as shown in Section 1.2.2. To analyze the effect of different levels of non-whiteness, the covariance of the normal signal is chosen as $\Sigma_{j,l}^{\text{ok}} = \omega^{|j-l|}$ where $\omega \in]0, 1[$ is a parameter that allows controlling the whiteness of the distribution.

One may measure the distance of a generic signal x from the white distribution with the *localization* [120] that is defined as

$$\mathcal{L}_x = \frac{\text{tr}(\Sigma^2)}{\text{tr}^2(\Sigma^{\text{ok}})} - \frac{1}{n} = \frac{\sum_{j=1}^n \lambda_j^{\text{ok}2}}{\left(\sum_{j=1}^n \lambda_j^{\text{ok}}\right)^2} - \frac{1}{n} \quad (1.29)$$

where Σ is the covariance of x and features λ_j with $j = 1, \dots, n$ as eigenvalues.

Localization ranges from $\mathcal{L}_{x^{\text{ok}}} = 0$ when the signal energy is equally distributed in any direction, i.e., the signal is white, to $\mathcal{L}_{x^{\text{ok}}} = 1 - 1/n$ when the energy is completely unbalance along a single component. In this dissertation, we consider three

different values for ω that corresponds to realistic levels of localization [42], specifically $\mathcal{L}_{x^{\text{ok}}} \in \{0, 0.05, 0.2\}$.

The anomalies are generated accordingly to the uniform distribution defined in 1.2.3. Firstly, the eigenvalues profile is drawn as $\lambda^{\text{ko}} \sim \mathcal{U}(\mathbb{S}^n)$ by following [147]

$$\lambda_j^{\text{ko}} = \frac{\log \xi_j}{\sum_{l=1}^n \log \xi_l} \quad (1.30)$$

Then, the eigenvectors matrix is generated as $Q^{\text{ko}} \sim \mathcal{U}(\mathbb{O}^n)$ by following [134]. As a start, we pick a matrix belonging to the Ginibre ensemble [76], whose elements are independently drawn as $\mathcal{N}(0, 1)$. Afterwards, the matrix is orthogonalized through QR-decomposition.

1.3.1 Asymptotic anomaly

As first numerical evidence, one may be interested in empirically assessing Property 1.5, i.e., the typical anomaly tends to behave like white noise when the dimension of the signal increases.

To show such a behaviour, we compute the deviation from I_n of several Σ^{ko} generated with n ranging from 2^7 to 2^{17} . The distances employed to measure the deviation are the following

$$\Delta_2 = \frac{1}{n} \sqrt{\sum_{j,l=1}^n \left[\Sigma_{j,l}^{\text{ko}} - (I_n)_{j,l} \right]^2} \quad (1.31)$$

$$\Delta_\infty = \max_{j,l} \left| \Sigma_{j,l}^{\text{ko}} - (I_n)_{j,l} \right| \quad (1.32)$$

Figure 1.2 reports both metrics depending on the signal dimension n with solid lines to represent the median trends and shaded areas to contain 98% of the population. The figure shows that both metrics feature a vanishing trend as n increases. Despite not being theoretically supported, the empirical results reveal a traditional $1/\sqrt{n}$ convergence.

1.3.2 Encoders

For the following analysis, the dimension of the signal x is set as $n = 32$ and three types of encoding techniques are considered:

- **Rate-Distortion Compression (RDC)** which consists in the optimal encoding in the rate-distortion trade-off sense, i.e., the encoder that yields the minimum rate given a distortion level.
- **Principal Component Compression (PCC)** which projects the input signal x onto the principal subspace, i.e., the subspace spanned by the eigenvectors of Σ^{ok} corresponding to the largest eigenvalues.

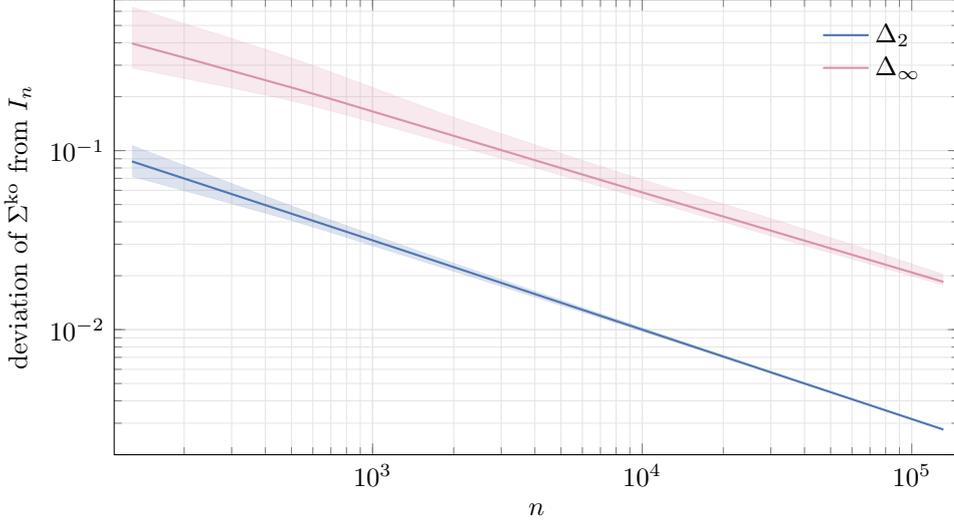


FIGURE 1.2: Deviation of the anomaly from white noise depending on the signal dimension n in term of Δ_2 and Δ_∞ .

- **Auto-Encoder Compression (AEC)** which includes a family of auto-encoder models [82, Chapter 14] to find a latent representation for the input signal. The considered models are fully connected neural networks where the encoder consists of four layers with dimension $n, 4n, 2n, \ell$ where ℓ is the dimensionality of the latent representation. The decoder is symmetric to the encoder and therefore consists of four layers with dimensions $\ell, 2n, 4n, n$. In both encoder and decoder, the adopted activation functions are ReLU [82, Chapter 6] in the hidden layers and Linear function in the output layer. The models are trained with (1.2) as loss function so that distortion is minimized. To smooth performance degradation, the models are trained sequentially with decreasing dimensionality of the latent space. Firstly, an autoencoder with $\ell = n - 1$ is trained. Then, the node of the latent representation that features the most negligible variance is dropped to produce a model with an $(\ell - 1)$ -dimensional latent space. The resulting model is re-trained using the previous weights as initialization. The procedure is repeated for each ℓ down to $\ell = 1$.

Since the encoder is tuned on the normal signal source, all three compression techniques are tuned to the normal signal.

Note that the three schemes address the trade-off between compression and distortion differently. To show that, we pair each compressor with a quantization stage that ensures the rate is finite. In detail, each element of the compressed signal \hat{x} is linearly quantized with 16 bits so that the maximum rate is limited to $16 \times n = 512$ bits for each time instant. It is assumed that quantization is sufficiently fine to consider \hat{x} approximately Gaussian. This assumption allows computing the mutual information between the original and the compressed signal $\mathcal{I}(x; \hat{x})$ as if they were jointly Gaussian with a covariance matrix estimated through Monte Carlo simulations [10].

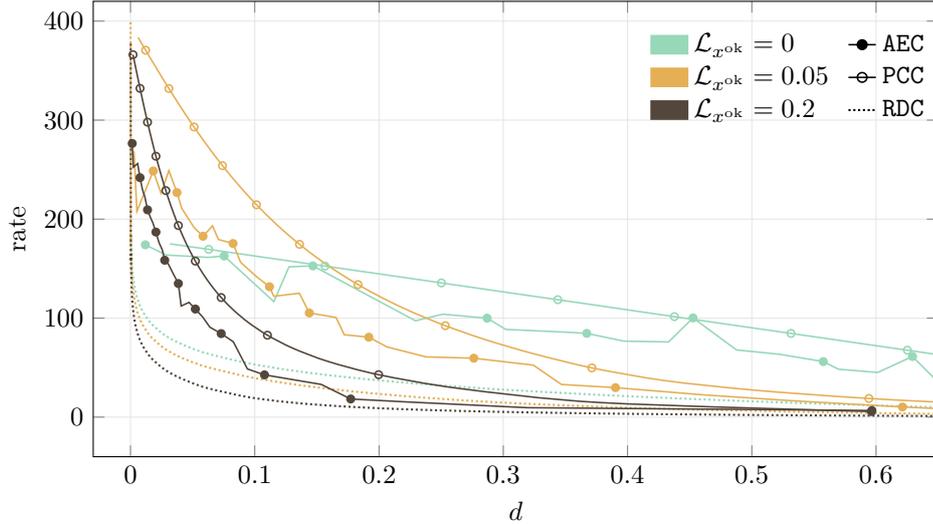


FIGURE 1.3: Rate distortion curves for the three encoding techniques RDC, PCC, and AEC for different levels of signal localization.

Figure 1.3 shows the rate-distortion curves obtained as a result of such estimation for different values of localization $\mathcal{L}_x^{\text{ok}}$. The plots are made against the normalized distortion $d = D/n$ in the range $d \in [0, 0.64]$ since larger distortion levels are usually not interesting for practical applications. As one may expect, RDC yields the lowest rates while PCC produces the largest ones, and between the two, there is AEC. It is also evident the effect of whiteness of the signal, which tends to flatten the curve as the localization decreases.

1.3.3 Detectors

The compressed representation of the original signal is then given as input to a detector whose task discriminates between normal and anomaly. To perform this task, a detector computes a score that assumes low values for normal signals and high values when the input behaves anomalously. Then, the detector matches the score against a threshold to discriminate between the two classes.

As for the distinguishability measures, two families of detectors are considered in this dissertation. The former consists of detectors that only rely on information about the normal behaviour, while the latter considers detectors that can take advantage of some knowledge about the anomaly.

From the former class, the following detectors are considered:

- **Likelihood Detector (LD)** that computes the score as the inverse of the log-likelihood of the instance \hat{x} with respect to the normal signal distribution $f_{\hat{x}}^{\text{ok}}$;
- **One-Class Support-Vector Machine (OCSVM)** [161] whose score is the distance from the envelope of normal instances estimated during the training phase in which 1% of unlabelled white instances has contaminated the training set.

detector	training			assessment		
	# Σ^{ko}	#instances $\times \Sigma^{\text{ko}}$		# Σ^{ko}	#instances $\times \Sigma^{\text{ko}}$	
		ok	ko		ok	ko
LD				1 000	1 000	1 000
OCSVM	1	99 000	1 000	1 000	1 000	1 000
NPD				1 000	1 000	1 000
DNN	50	100 000	100 000	50	100 000	100 000

TABLE 1.1: Composition and dimension of the data sets employed for the training (when required) and the performance assessment of the detectors.

Among the detectors that exploit information on the anomaly, we consider the following:

- **Neyman-Pearson Detector (NPD)** [101, Chapter 3], whose score is the difference between the log-likelihoods of the instance \hat{x} with respect to the anomalous and normal distributions, $f_{\hat{x}}^{\text{ko}}$ and $f_{\hat{x}}^{\text{ok}}$ respectively;
- **Deep Neural Network (DNN)** consisting of 3 fully connected hidden layers with $k_{\theta}, 2n, n$ neurons with a ReLU activation and a single output neuron with a sigmoid as activation function, which produces the score. The network is trained with a dataset containing an equal number of labelled normal and anomalous instances and considering the binary cross-entropy as a loss function.

Since LD and NPD rely on the knowledge of the pdf of the compressed signal, they can be only applied on signal encoded by RDC or PCC and not by AEC as pdf is not available after the nonlinear processing.

Table 1.1 reports the number of different anomalies and the number of signal instances generated for the training (when needed) and the testing of the detectors. In general, detectors performance is assessed by generating 1000 different anomaly sources and 1000 instances each Σ^{ko} . The only exception is DNN, for which the number of anomalies is limited to 50. This limitation is justified by the need of repeating the training for each anomaly with 10^5 different anomaly samples. Note that, LD and NPD do not require training since they rely on the signal pdfs.

Performance figure Performance assessment for a detector refers to the capabilities of correctly classifying an anomaly among normal signal instances. Let us consider the detection of an anomaly as a positive and of a normal signal instance as a negative event. Then, the detector makes a mistake when it classifies either a normal instance as an anomaly (False Positive) or an anomaly as a normal instance (False Negative). On the other hand, we have a True Positive or a True Negative when the detector succeeds in detecting an anomaly or a normal instance, respectively.

In general, classification is performed by applying a threshold to the score that the classifier associates to the input instance. In this dissertation, we focus on the

receiving operating characteristic (ROC) curve which plots the True Positive Rate (TPR) versus the False Positive Rate (FPR)[69]. TPR and FPR refer to the probability of having a TP or a FP, denoted as p_{TP} and p_{FP}

$$\text{TPR} = p_{\text{TP}}(\xi) = \Pr \left\{ s(x) > \xi | x \sim f_x^{\text{ko}} \right\} = 1 - F_{s|\text{ko}}(\xi) \quad (1.33)$$

$$\text{FPR} = p_{\text{FP}}(\xi) = \Pr \left\{ s(x) > \xi | x \sim f_x^{\text{ok}} \right\} = 1 - F_{s|\text{ok}}(\xi) \quad (1.34)$$

where $s(x)$ refers to the score associated by the detector to the signal instance x , ξ acts as threshold that discriminates between anomaly and normal instance, and $F_{s|\times}$ indicates the cumulative density function (cdf) of the score s given the input \times , where \times can be either ok or ko.

When $F_{s|\text{ko}}$ and $F_{s|\text{ok}}$ are not available, TPR and FPR are estimated by Monte Carlo simulations as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (1.35)$$

where TP, FN, FP, and TN refer to the number of True Positives, False Negatives, False Positives, and True Negatives.

Hence, each point of the ROC depends on the threshold employed to discriminate positives from negatives. Since thresholds often depend on the application, in this dissertation, we focus on the Area Under the Curve (AUC) which provides an independent measure as it takes into account all possible thresholds. AUC has also a probabilistic interpretation as shown in [87]. Indeed, AUC can be computed with the following integral

$$\text{AUC} = \int_0^1 p_{\text{TP}}(p_{\text{FP}}^{-1}(p)) dp \quad (1.36)$$

where p is the value on the horizontal axis of the ROC. Let us consider the change of variable $p = p_{\text{FP}}(\alpha)$ so that

$$\begin{aligned} \text{AUC} &= - \int_{-\infty}^{\infty} p_{\text{TP}}(\alpha) p'_{\text{FP}}(\alpha) d\alpha \\ &= \int_{-\infty}^{\infty} (1 - F_{s|\text{ko}}(\alpha)) f_{s|\text{ok}}(\alpha) d\alpha \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} 1_{[\beta > \alpha]} f_{s|\text{ko}}(\beta) f_{s|\text{ok}}(\alpha) d\beta d\alpha \\ &= \Pr \{ s_{\text{ko}} > s_{\text{ok}} \} \end{aligned} \quad (1.37)$$

where 1_I refers to a function that is 1 in the interval I and 0 otherwise, and the negative sign in the first step is due to the fact that for an increasing score α the probability of a FP decreases. Note that from (1.37) it is evident that AUC can be interpreted as the probability for a randomly picked anomaly to score higher than a randomly picked normal signal instance. For this reason, AUC is a positive performance index that ranges from 0 (anomalies always score lower than normal instances) to 1 (ideal setting).

Clearly, if a detector yields $\text{AUC} = 0$ is as capable of discriminating between normal and anomaly as an ideal detector, as it is sufficient to interpret the score in a reverse way to obtain $\text{AUC} = 1$. Hence, one may measure the ability to distinguish normal and anomalous signals as the distance from $\text{AUC} = 1/2$ that corresponds to a detector that is no better than coin tossing. That is the reason for the definition of the following measure

$$\psi = |\text{AUC} - 1/2| \quad (1.38)$$

which is a positive index that ranges from $\psi = 0$ when performance is poor to $\psi = 0.5$ in the case of an ideal detector.

The performance of each detector in terms of ψ is matched with the trends of $|\zeta|$ and κ to show how theoretical properties reflect on practical cases. Comparisons are qualitative as the nature of ζ , and κ is different from the nature of ψ .

Rate-Distortion Compression (RDC) Figure 1.4 reports the distinguishability measures ζ , κ , and ψ in the settings considering RDC as encoder and all aforementioned detectors. The plots are arranged in a grid of three rows and two columns. The plots on the left column concern the anomaly-agnostic scenario in which only information related to the normal behaviour is exploitable, i.e., from top to bottom, ζ , ψ for detectors LD and OCSVM. On the right columns are plots concerning the anomaly-aware scenario where some knowledge about the anomaly is workable, i.e., from top to bottom, κ , ψ for detectors NPD and DNN. Colours refer to a different level of signal localization $\mathcal{L}_{x^{\text{ok}}}$, solid lines correspond to median trends, shaded areas to the span of 50% of the Monte Carlo population, and dashed lines refer to the white anomaly (i.e., white noise). There is no curve for $|\zeta|$ and $\mathcal{L}_x = 0$ since in that case $\zeta = 0$.

With no knowledge of the anomaly (left column), $|\zeta|$ anticipates that a limited amount of distortion ($d \ll 1$) may cause distinguishability to vanish and thus detectors to fail. This phenomenon happens for LD and OCSVM detectors, and the distortion level at which they fail is also anticipated by theory and depends on $\mathcal{L}_{x^{\text{ok}}}$ as predicted by Property 1.4. Overall, $|\zeta|$ anticipates that, in the low-distortion region, more localized signals are more distinguishable from anomaly though they cause detector failures at lower distortions than less localized signals.

Detectors leveraging the knowledge of the anomaly (right column) fail completely only at 100% distortion as revealed by the abstract distinguishability measure κ . Even in this case, by comparing the trend of κ with the zoomed areas in the NPD and DNN plots, we see how theory anticipates that, in the low-distortion region, more localized signals tend to be more distinguishable from anomalies but cause more substantial performance degradation of detectors when d increases.

Principal Component Compression (PCC) From the rate-distortion point of view, PCC is largely suboptimal, as shown in Figure 1.3. Yet, due to its linear nature x and \hat{x} are still jointly Gaussian, allowing for the computation of the theoretical measures $|\zeta|$ and κ by means of (1.20) and (1.21).

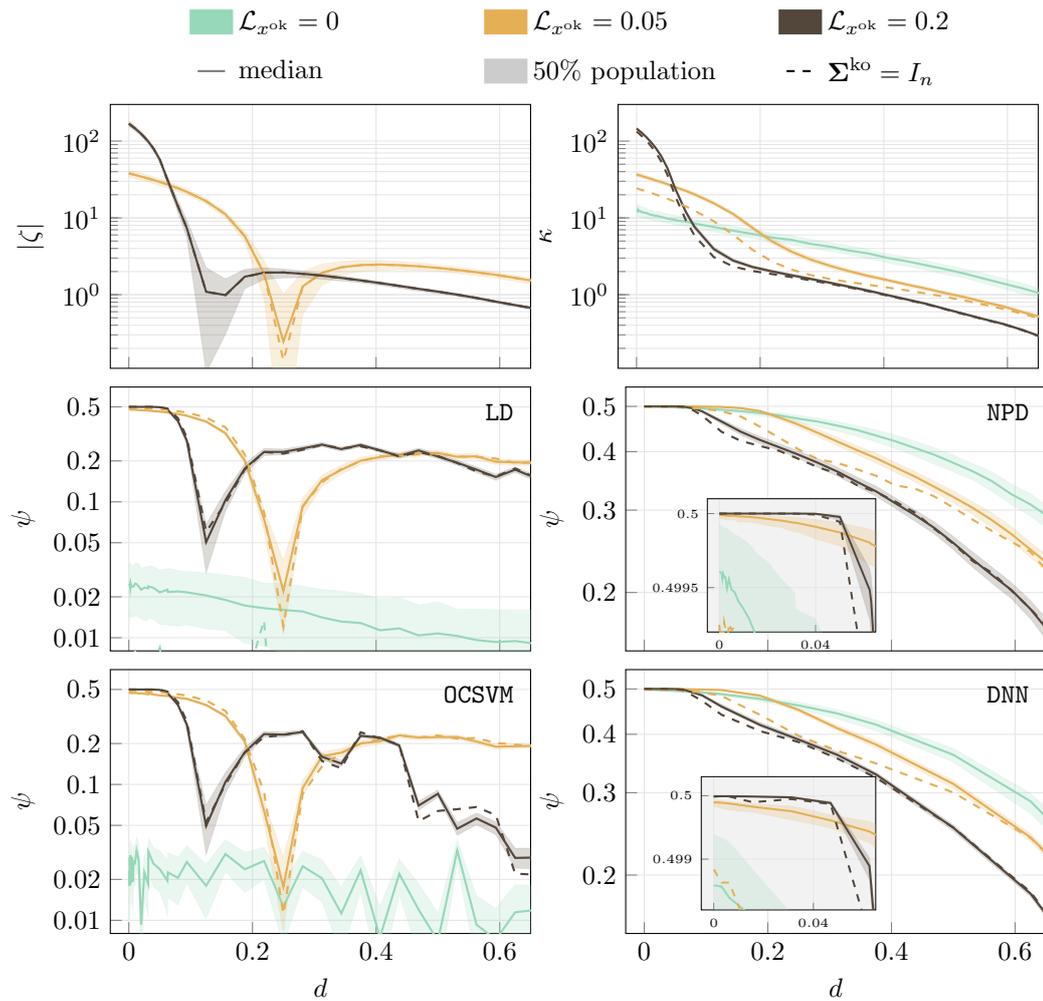


FIGURE 1.4: Distinguishability measures ζ , κ and ψ against normalized distortion d in case of RDC.

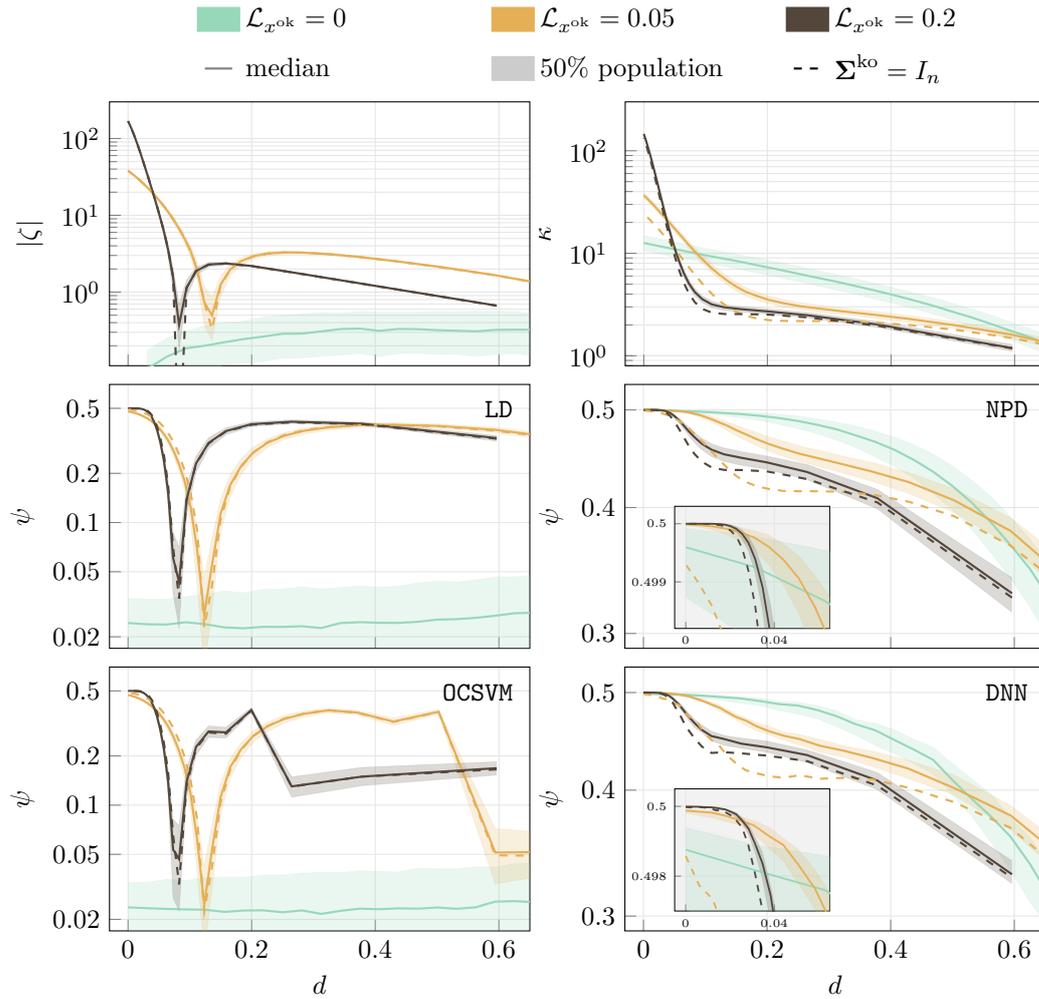


FIGURE 1.5: Distinguishability measures ζ , κ and ψ against normalized distortion d in case of PCC.

Figure 1.5 reports the distinguishability measure ζ and ψ along the lines of Figure 1.4. The qualitative behaviours seen in the previous case are confirmed, but the values of the involved quantities change compared to the RDC case. The distortion levels at which anomaly-agnostic detectors fail is lower than RDC, but the values of $|\zeta|$ beyond breakdown are slightly higher. As a consequence, one may instead adopt a sub-optimal encoding strategy in the rate-distortion sense in favour of better distinguishability as in the case of LD and OCSVM.

Autoencoder Compression (AEC) In the case of AEC as encoder, the non-linear processing of x to encode it into \hat{x} , in general, makes the relation between the two signals non-linear, and therefore they may not be jointly Gaussian. This fact prevents from computing the theoretical measures $|\zeta|$ and κ and from applying LD and NPD which rely on the knowledge of the distribution of the signals.

For this reason, Figure 1.6 only shows the performance of OCSVM and DNN detectors. Note that, even in this case, the qualitative trends still follows the theoretical

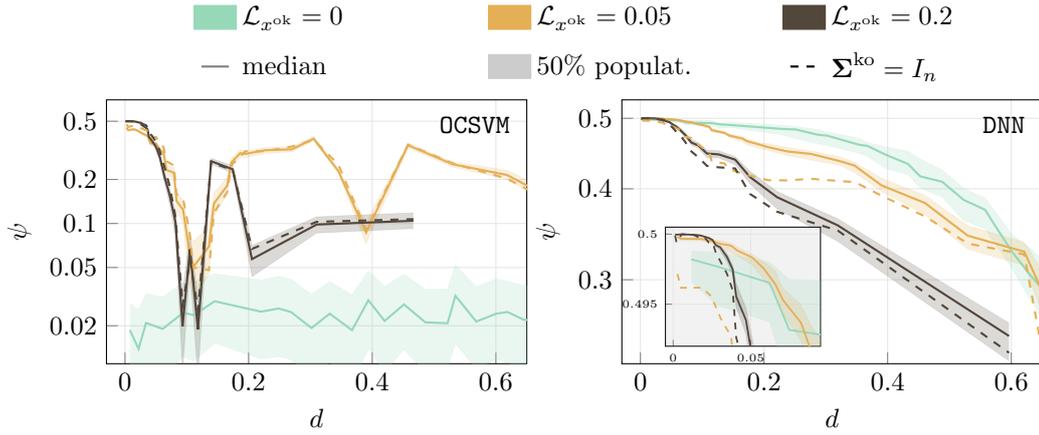


FIGURE 1.6: Distinguishability measure ψ against normalized distortion d in case of AEC.

profiles observed for the other compressors, though with a larger level of approximation.

Conclusion

Monitoring systems are often characterized by a data flow from sensors to a central server for which compression strategies are adopted to meet the communication requirements. However, the information in the acquired signals may be processed locally for early detection of anomalies leading to edge devices implementing detectors that operate on compressed bitstreams.

In this chapter, the trade-off between compression and performances of anomaly detectors is addressed. The analysis first requires the definition of two information-theoretic quantities κ and ζ that can measure the distinguishability between two sources of signal distorted by an encoding stage in two different scenarios. A source is associated with the observed normal behaviour and is exploited to tune the encoder. The other source refers to a generic anomaly whose statistical characterization may be either known (anomaly-aware scenario) or not (anomaly-agnostic scenario).

Assuming the sources as Gaussian stochastic processes, it was possible to analytically derive an expression for the two distinguishability measures κ and ζ in case of an optimal encoder in the rate-distortion sense. The Gaussian assumption also allowed us to derive that the average anomaly and the anomaly with dimension tending to infinity behave as a white distribution.

Numerical evidence then permitted to show that, in general, the trade-off between distinguishability and compression is not effectively addressed by an encoder optimal in the rate-distortion sense and that the theoretical measures κ and ζ are able to anticipate the trends observed in case of practical encoders (based on PCA and Auto-Encoder) and practical detectors (One-Class SVM and Deep Neural Network).

Chapter 2

Detection and Compression in Structural Health Monitoring

In the previous chapter, we have seen how compression influences the performance of an anomaly detector in the two scenarios in which the anomaly was either known or completely unknown. The analysis was pursued from a general point of view that includes quantities such as entropy, mutual information and average code length typical of Information theory. Here, in this chapter, we delve into a more practical point of view.

We will consider Structural Health Monitoring (SHM) as a use case. An SHM monitoring system is a good representative of a deployment of a large number of sensors that are generally heterogeneous and displaced in/on/nearby a structure to gather information related to the health of that structure. The information may be related to either the static or dynamic properties, and both the environment and operational conditions influence it. Typically monitored structures are building, bridges, roads, dams, and so on, sometimes in a remote area with no easy or stable internet connection. Since the communication between the installation and the central processing node may be critical, data compression and local processing play a vital role in the effectiveness of the monitoring system.

In particular, here, we will consider a specific SHM system installed on a remote highway viaduct counting a hundred sensor nodes equipped with a 3-axial accelerometer, a temperature, and a humidity sensor. The sensors monitor both how the structure oscillates in response to the passing traffic and the environmental conditions. The data generated by the installation is wirelessly sent to a cloud facility devoted to storage and processing. In this context, the system may benefit from either compressing or processing data locally for the early detection of anomalies.

Here two approaches are investigated: detection based on feature extraction and subspace analysis. The former summarises the signal in few quantities and generates an alarm when these quantities take abnormal values. Instead, the latter compresses the signal in lower-dimensional subspace by linear transformation and monitors how this representation behaves. Both considered approaches are designed to meet the low-resources constraint necessary to fit on the edge devices comprising the monitoring system.

In detail, this chapter starts with Section 2.1 that briefly introduces Structural Health Monitoring and the characteristics of the monitoring system employed in this field. Then, in Section 2.2, the use case is presented in detail. We describe the structure of the considered highway viaduct along with the monitoring system and its components that is the same reported in [78, 40, 38, 27, 127, 39]. We will also depict the characteristics corresponding to the ordinary behaviour of the viaduct and some of the anomalies occurring during the monitoring period used to assess the detectors efficacy.

In Section 2.3, a first anomaly detection approach based on feature extraction is described. Here, feature extraction is an extreme case of compression since an entire signal instance is reduced to some significant quantities providing a summary and not designed to recover the original signal. For this reason, we refer to this approach as a *detector with high distortion compression*. The considered approach consider features extracted in both time and frequency domains with techniques that fit the edge devices resources.

As a counterpart, in Section 2.4, we report anomaly detectors that are based on the principal subspace analysis of the monitored signal [127]. The signal is compressed in a latent representation with a linear transformation that maximizes the collected variance, corresponding to maximizing the information contained in the compressed signal. For this reason, we refer to this approach as a *detector with low distortion compression*. By observing the signal energy distribution in the latent space, the detector may discriminate between normal or anomalous signal instances. This method also has a dual version in which the linear transformation compressing the signal minimizes the variance. However, minimizing variance translates into maximizing distortion that prevents signal recovery. The two methods are first assessed in a Gaussian framework and then applied to Structural Health Monitoring signals.

2.1 Structural Health Monitoring

The role of civil structures and infrastructures is fundamental in modern societies. People live and work in buildings or large factories; to commute, people use roads and railways, which cross each other using bridges or tunnels. Besides ageing phenomena, causing fissures in concrete or steel corrosion, various other factors can damage a structure, often impacting them in a hard way to foresee. These factors are, for instance, the case of natural hazards, such as earthquakes, storms, or floods, which cannot be prevented and can generate a severe loss of structural integrity. What is worse is that structural damages may not be immediately apparent and may manifest themselves dramatically later.

Structural health monitoring (SHM) has been recently proposed as a viable solution to move toward continuous monitoring of critical infrastructures [183, 50, 68]. It concerns the continuous monitoring of infrastructures such as buildings, bridges, aircraft, dams and so on, intending to provide a continuous flow of information

about civil and mechanical structures to increase their safety and reduce their maintenance costs [67, 115].

Structural health information is extracted by produced by a heterogeneous sensor network that is installed on/in/near the structure in a non-invasive manner. The sensor network measures quantities related to both static and dynamic properties of the structures and related to the environmental conditions. Typically, the monitored physical quantities are humidity, temperature, vibration, acoustic emission, tensile/compressing stress and material degradation.

Data is then fused and analyzed to determine the state of health of the structure and to detect and then localize possible damages before they become critical. Damages can be caused by several factors such as material ageing, components wearing, the action of the environment or even accidental events. An effective SHM system is therefore able to promptly detect changes in the properties of the structure so that timely action can be taken.

Over the last 40 years, the SHM field has been significantly growing due to the considerable advances in sensing, communication and information technologies. Thanks to these advances, the SHM systems have dramatically improved in efficacy and have drastically lowered costs [49].

The SHM field is extremely varied as it concerns monitoring a wide range of structures such as aerospace aircraft, buildings, industrial machinery, and bridges, to name the most common ones. Moreover, an SHM may be applied directly to new structures (so that the monitoring system can be designed with it) and applied to already existing ones. Depending on the specific application, the challenges and correspondent solutions can be different, leading to the fact that there is no standard in the design of an SHM system [68].

A key open challenge in SHM is processing a massive amount of sensor readings to allow human experts to evaluate the structural condition. It comes immediately apparent that SHM cannot succeed in real-life deployment scenarios without an autonomous system for early anomaly detection that pre-processes the measurements in real-time and can recognize the occurrence of an anomaly as opposed to normal behaviour [190, 127]. However, even assuming the feasibility of automating anomaly detection, scaling up to hundreds of sensors per infrastructure remains a challenge. As the complexity of each installation is growing and the number of monitored structures is increasing, a common approach is to process the whole data on a *cloud* architecture [109].

Over the last ten years, the paradigm of the Internet of Things (IoT) let the SHM field have a further boost [179, 164]. Sensors have become smart, which means they are devoted to measuring physical quantities, processing data and transmitting the extracted information through the internet to make decisions, i.e., they are Things on the Internet.

Once the information reaches the internet, the Big Data paradigm permits flexibility, scalability and processing power for its management. These features provide significant advantages in controlling the monitoring system and performing the analysis necessary to elaborate the complex information regarding the health of a structure.

In general, we can divide an SHM system into three subsystems [179]:

1. **sensing and data acquisition subsystem:** data is generated by sensors that measure physical quantities directly or indirectly related to the health of the structure and the condition of the surrounding environment. The sensing elements are often included in devices that take the measure and transmit it to where the data is stored or analysed. This subsystem is therefore designed by taking into consideration the physical quantities to measure, the number of sensors, their placement, and how the data is transferred.
2. **data management subsystem:** the information content in the data generated by the sensing subsystem must be collected and analysed. This subsystem involves the procedures or protocols adopted for data collection and the processing techniques to extract valuable information from the raw data. Note that, especially in SHM systems that rely on high-frequency signals (such as vibrations) that are often redundant, data compression or feature extraction is performed at the gateway or even at the sensor level to reduce the cost of communication. This approach leads to the need of sensors capable of performing data processing.
3. **data access and retrieval subsystem:** the information extracted from the data must be accessible from the user either online or offline. Online access is often required in the case of automatic decision-making systems, where action must be promptly taken to respond to unexpected events. When the access to data needs to be in real-time, there is the need for guarantees on stability and latency, which are among the main issues of system transferring data over the internet.

A typical cloud-based system can be depicted as in Figure 2.1 where there is a large number of sensor nodes deployed on the monitored structure, a smaller number of gateways that gather the readings to send them to the cloud, which operates as a central processing unit [5, 34].

2.2 Viaduct Health Monitoring system

The infrastructure considered in this dissertation is a highway viaduct located in Italy, opened in 2006, on which many maintenance interventions have already been undertaken. The construction, displayed in Figure 2.2, is a composite box girder with external prestressed tendons used for reinforcement. The length of the viaduct

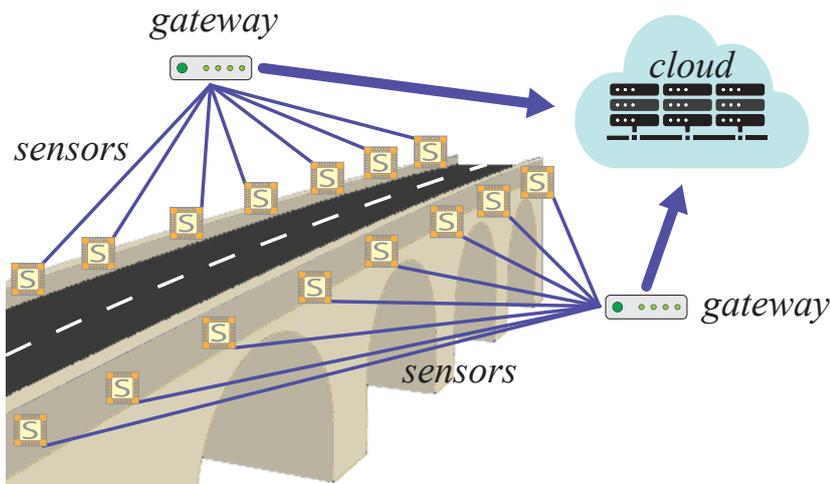
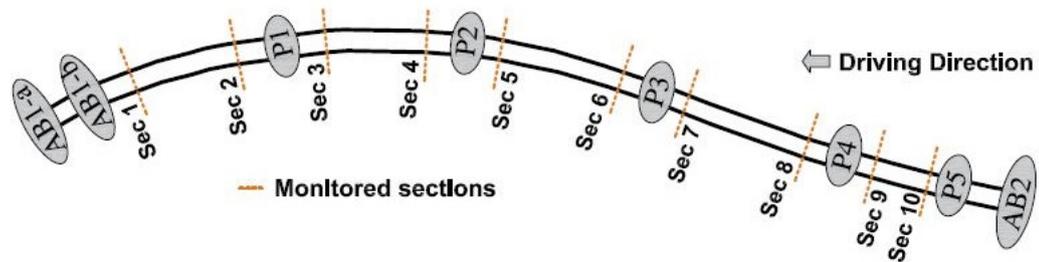


FIGURE 2.1: blocks scheme of SHM system monitoring a viaduct. Sensors send their reading to a main cloud platform via intermediate gateway.

is $580m$, and five concrete piers hold up the six spans. The height of the cross-section varies from $6m$ near each pier to $3m$ on the centre line of each span. The external prestressing is provided by unbonded tendons (each consisting of 27 strands) anchored at the abutments, with several deviators along with the structure. Figure 2.2b shows a plan view of the viaduct, while Figure 2.2b depicts the piers of the central span.

The structural safety of a prestressed viaduct depends on the durability of its prestressing cables to a large extent; thus, the monitoring is performed with sensors placed on the tendons. All the cables have roughly the same mass-over-length ratio, and they have been equally prestressed. The tendons belonging to the first four spans are $20m$ length, and their natural vibration frequencies are about $7-8Hz$; while the cables in the last span have a reduced length of about $7.30m$ that increases the natural vibration frequency up to about $20Hz$.

The SHM system analyzes the indirect dynamic response of the viaduct structure during operating conditions, using the online measurement of the natural vibration frequency of the tendons. A triaxial MEMS accelerometer is placed on each side of the external tendons selected for the measures, and the final deployment consisted of 90 sensors with a sampling rate of $100Hz$. Each triaxial MEMS measures the acceleration in three orthogonal directions (x, y, z), and is characterized by a range of $\pm 2g$, a $1.8kHz$ band, and features the lowest noise density ($50\mu g/\sqrt{Hz}$) for its class of devices [132]. The signal is initially sampled at a frequency of $25.6kHz$, then filtered using a low-pass filter with $50Hz$ cutoff, and downsampled to $100Hz$ data rate. Buildings are usually designed for low-frequency normal modes, typically $0Hz$ to $20Hz$ bandwidth, whereas external stresses (e.g., earthquakes) can increase typical frequencies up to $20-45Hz$ [77]. This aspect motivates the final $100Hz$ data rate, and the whole procedure improves the signal to quantization noise ratio and increases the acquisition resolution [78, 27].



(A) Plan view of the monitored viaduct



(B) Photo of the monitored viaduct showing span s_2 and s_3 and the piers P_2 and P_3 .



(C) Photo of the sensor nodes installed on the external steel tendons.

FIGURE 2.2: Monitored viaduct and adopted SHM system.

TABLE 2.1: Devices adopted for sensors and gateways in the viaduct health monitoring system.

	Sensor Node	Gateway
Quantity	90	2
MCU/Module	STM32F405RG	Raspberry Pi 3 Model B+
CPU	ARM Cortex M4	4× ARM Cortex A53
Memory	192 kB	1 GB
Clock freq.	168 MHz	1.2 GHz

Sensors are custom embedded systems based on the STM32F405RG microcontroller unit (MCU) and are deployed in the upper part of the prestressed cables, as in Figure 2.2c. Each node is placed with z -axis orthogonal to the road surface, y -axis parallel to the tendon and pointing to the centerline of each span. After processing the output accelerations, each sensor encodes and sends the data through a CAN bus network to the local gateway, where they are stored, aggregated and further analyzed.

The gateway is realized with a Raspberry Pi 3 Model B+ and is responsible for the sensors' acquisition and synchronization. Two gateways are employed, each collecting and managing 45 sensors. Both gateways and sensors are powered by wire. Table 2.1 reports a summary of the devices involved in the SHM system.

The samples are transmitted over the CAN bus, and a software module is dedicated to packing the data by sensor in binary files of 20 000 samples each. When a file is complete, it is sent to the cloud through the MQTT protocol [173]. The dimension of the packet is determined by the max payload supported by the protocol that is 120 kB (20 ksample of 3-axis acceleration, each one represented with 2 B are equal to 120 000 B are about 118 kB). Data is paired with another MQTT message containing the metadata in JSON format. Metadata consists of info about the data, such as the timestamp of the first sample, average temperature and humidity, and binary file format.

The data collected by the gateways is sent via Ethernet to a "Ubiquity Nano M5" station located halfway between the viaduct ends. M5 station is also connected via 5 GHz point-to-point Wi-Fi to the access point, which transfers the entire data to the cloud. The cloud system comprises a storage platform and a computing machine allocated to the IBM cloud service. IBM Cloud Object Storage is employed to store data as parquet files and, currently, a virtual machine with 2 nodes, each featuring 4 cores with 16 GB RAM. Acceleration, temperature and humidity data are stored in a cloud monitoring infrastructure, allowing real-time access and analysis. Data is processed to detect unusual patterns that do not fit the normal behaviour of the structure, such as abrupt damages or progressive degradation.

2.2.1 Normal behaviour

Figure 2.3 shows the typical profiles in the time and frequency domain of the acceleration measured on each axis concerning a sensor placed in the first span of the viaduct. Figure 2.3a displays a 200 s window (corresponding to a MQTT payload) of the acceleration a over time split in the three components a_1 (x -axis), a_2 (y -axis), a_3 (z -axis). Figure 2.3b reports the power spectral densities (PSDs) of the acceleration on each axis estimated over 1 hour interval (concatenation the content of 18 successive MQTT messages) employing Welch's method with a window length of 1024 samples and 50% overlap.

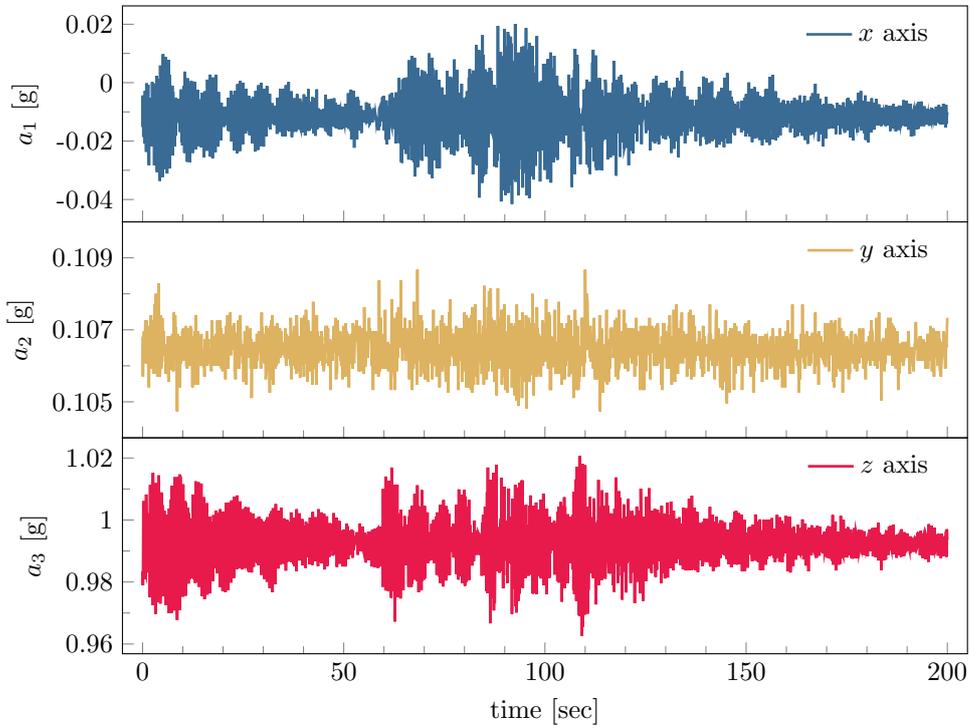
Although curves in Figure 2.3 are not representative of the entire dataset, these are an example of profiles associated with normal behaviour with some common characteristics:

- time-domain waveforms can be modelled as damped oscillations corresponding to the structure's elastic response to an occurred stimulus (the response to the vehicle traffic);
- in the sensors placement phase, the y -axis is aligned with the tendon such that in normal behaviour, the signal on the y -axis is negligible, and the x - z plane contains the main part of the damped oscillations. The z -axis is aligned with the gravity vector;
- estimated power spectral densities exhibit peaks at the natural frequencies of the monitored structure. In particular, for each sensor, it is common to observe either 4 or 5 peaks couples with frequencies, amplitudes and shapes depending on the monitored tendon and the sensor placement.

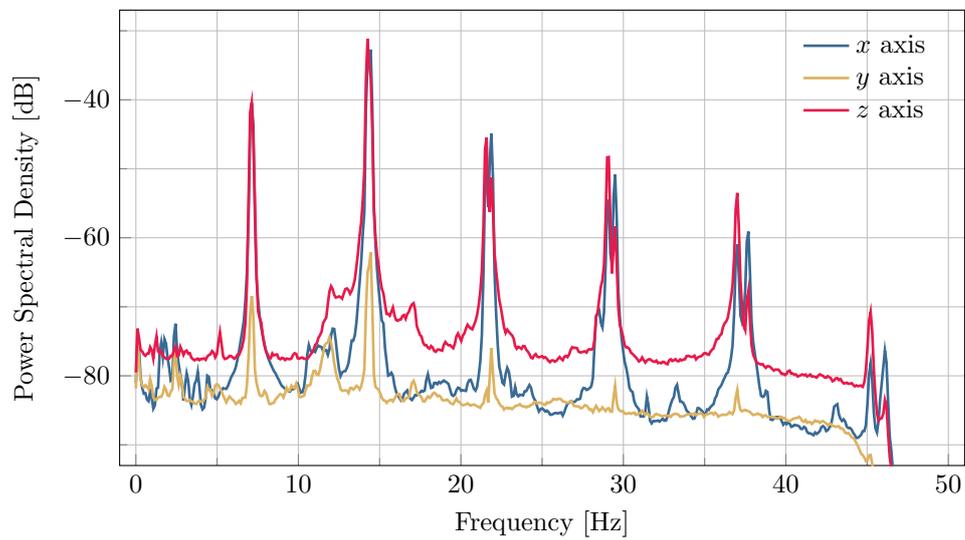
2.2.2 Anomalies

The infrastructure has been monitored since September 2017, and data considered in this dissertation refers to the first two years of the monitoring activity. In such period some anomalies have been observed, and the following analysis refers to three representative cases:

- **earthquake:** an earthquake with 4.4 magnitude and epicenter at a distance of 16 km from the viaduct with a depth of 22 km. The event excited the structure so that the tendons vibrated with a typical amplitude on the x - z plane but significantly higher on the y -axis.
- **tendon break:** a destructive event occurred in the first span of the viaduct during the structure monitoring period. As a consequence of the event, a considerable amount of energy was released that made vibrate not only the first span but also the entire structure. The event did not lead to a sensitive change in the structure properties but was destructive for the tendon involved.



(A) Signal in time domain (200 s).



(B) Signal Power Spectral Density (1 h).

FIGURE 2.3: 3-axes acceleration signal in ordinary conditions.

- **strand break:** an event with much lower intensity compared to the previous anomalies that are only detectable by the sensor placed on the involved tendon. More than one event of this type occurred during the monitoring period, but we refer to a strand break in the last span tendon in the specific case.

Figure 2.4 depicts chunks with length 30 s of the acceleration acquired during the occurrence of the three anomalies taken from a sensor placed in the last span of the viaduct (i.e., in the opposite span with respect to where the tendon break occurred). Earthquake and tendon break are events causing a high energy release throughout the entire structure which the sensor detects as damped oscillations, while, for the strand break, the energy release is limited in time.

2.3 Detection with High Distortion Compression

An extreme case of compression is the extraction of features. Features are quantities derived from the original signal that captures most of the input's valuable information for a representation in a space with a much lower dimension. In general, features are not designed to allow the recovery of the input as they are just used to avoid working with high dimensional data that may be computationally intractable or affect the performance of the algorithm that suffer from the curse of dimensionality. Therefore, features are a kind of compression that introduces a high level of distortion.

In this section, we consider a typical anomaly detection procedure based on features computed in time and frequency domain and applied to the vibration data collected from the system described in Section 2.2. Features in the time domain consist of typical statistical quantities that prompt the detection of unusual and evident events. The frequency-domain analysis allows extracting features related to the modal frequencies that are characteristic of the dynamic properties of the structure.

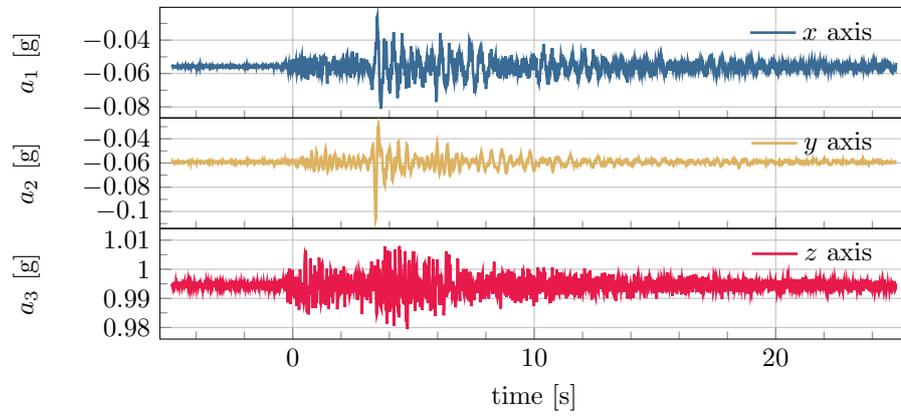
All the analyses consider a reference sensor placed in the last span of the viaduct (the opposite span with respect to where the tendon break occurred).

2.3.1 Time domain analysis

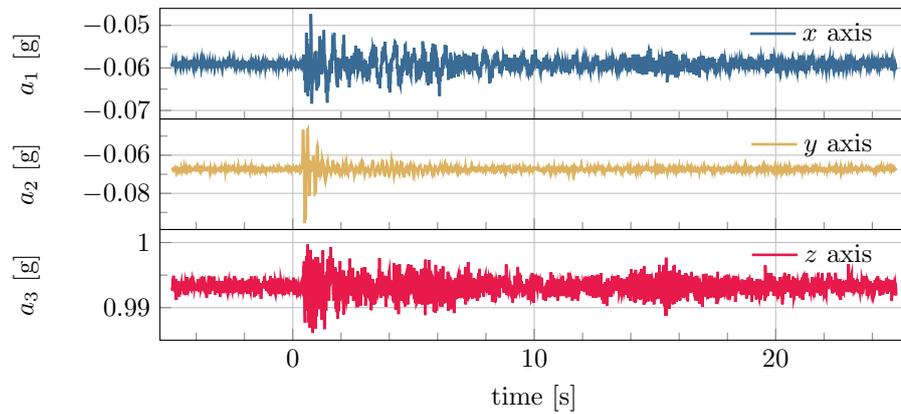
In the time domain analysis, the sensor readings are split into windows of 1 s so that for each axis i a window of $n_i = 100$ samples that are aggregated in a single signal x with dimension $n = 300$.

The choice of n results from the trade-off between system responsiveness and feature stability. Short windows allow a low delay between the unusual event occurrence and its detection, while long windows have better features estimations. This trade-off is addressed by imposing $n_i = 100$ samples, i.e., each MQTT payload A aggregates 200 successive signal instances with a maximum detection delay of 1 s.

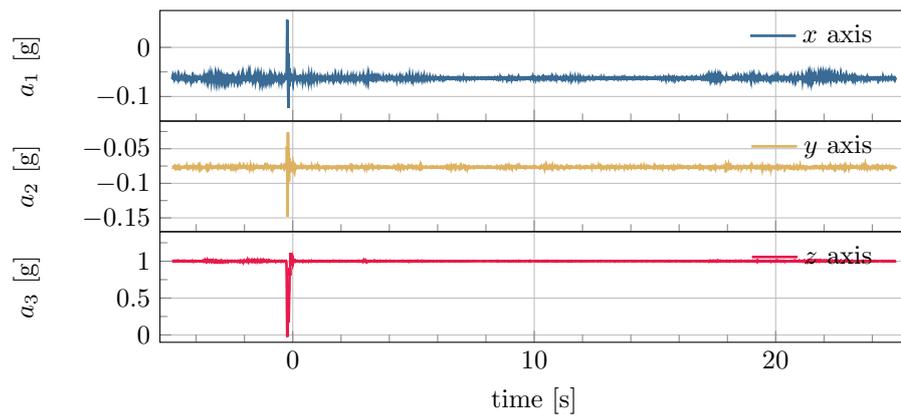
The signal x is then compressed into a 9-dimensional feature vector y . The features are computed component-wise, and consist of mean m , variance s^2 ,



(A) Earthquake



(B) Tendon break



(C) Strand break

FIGURE 2.4: 30s of acceleration signals for all axes that contains anomalous events.

and peak-to-peak value p . Considering $x_{i,j}$ as the j -th sample of the i -th axis, the features are defined as:

$$m_i = \frac{1}{n} \sum_{j=1}^n x_{i,j} \quad (2.1)$$

$$s_i^2 = \frac{1}{n-1} \sum_{j=1}^n (x_{i,j} - m_i)^2 \quad (2.2)$$

$$p_i = \max_j x_{i,j} - \min_j x_{i,j} \quad (2.3)$$

The mean $m = (m_1, m_2, m_3)$ allows to determine the actual orientation relative to the ground, so it is helpful to detect sensor movements or rotations. The variance $s^2 = (s_1^2, s_2^2, s_3^2)$ is an indicator of the oscillation energy while peak-to-peak $p = (p_1, p_2, p_3)$ vectors are useful to detect spike-like events.

Along with the physical meaning, the choice of the features is also driven by the computational cost. Indeed, all three features can be computed in an online fashion with a memory footprint independent from n , i.e., $O(1)$. The online implementation of the peak-to-peak measure is trivial, while Welford's algorithm [182, 103] allows to update mean and variance at each new incoming sample by making use of only two state variables.

As a result of the feature extraction stage, the input signal x is compressed into the 9-dimensional feature vector $y = (m, s^2, p)$. y can result from the compression of either a normal instance y^{ok} or an anomaly y^{ko} . In order to discriminate between normal or anomalous behaviour, the detector should know the distribution of both y^{ok} and y^{ko} (anomaly-aware detector) or the distribution of only y^{ok} (anomaly-agnostic detector). As in most real SHM applications, we here deal with the latter scenario in which the distribution of the anomaly, and therefore of its compressed version, is unknown.

Normal behaviour characterization To identify a normal behaviour region in the features space, we infer the data structure by applying an unsupervised clustering method. We choose DBSCAN algorithm [64, 163] because it does not require any assumption on the shape of the clusters, and it is a density-based clustering method that exhibits solid performance with 3D spatial data. The clustering algorithm finds that the data is structured in a single cluster. In Figure 2.5 the features space is split into three 3D spaces to allow a graphical representation.

The distribution of m presents the typical shape of a 3D Gaussian probability density function. Since structural vibrations have a null mean, the main contribution to this feature variability in normal conditions is the acquisition noise, which approximately has a white Gaussian distribution. Both s^2 and p distributions possess a high-density region close to the origin representing the structure behaviour in case of no stimulus. Otherwise, oscillations produce s^2 and p values that move away from the origin along the main direction.

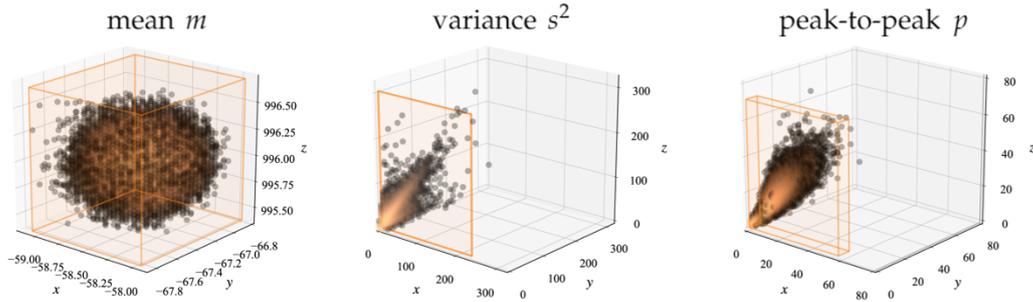


FIGURE 2.5: Distribution of the times domain features m , s^2 , p estimated over a week of data. Warmth of colors is proportional to the population density and the bounding boxes define the regions of normal behaviour.

TABLE 2.2: Alarm thresholds obtained in training for the reference sensor.

feature	x -axis		y -axis		z -axis		u.o.m.
	low	high	low	high	low	high	
m	-59.1	-57.9	-67.9	-66.8	995.4	996.7	mg
p	0.0	57.7	0.0	7.8	0.0	71.7	mg
s^2	0.0	233.6	0.0	2.7	0.0	304.3	mg ²

Once the normal behaviour is traced, a mechanism to detect anomalies is needed. To meet the low complexity requirement, we chose to draw a bounding box in the feature space that contains all the cluster elements (shown in Figure 2.5 with the light coloured parallelepiped) such that two thresholds for each feature define the interval containing the normal behaviour instances. The thresholds result from the trade-off between robustness to outliers in the training set and operative false positive rate (FPR), i.e., the probability for a normal instance to trigger a threshold once the algorithm is deployed.

We chose a normal behaviour region including 99.999% of the training set elements, which correspond to $FPR = 10^{-5}$, i.e., approximately one FP per day per sensor. Experiments confirm this rate: considering the reference sensor and the 20 weeks following the one used for training, the median and the 95-th percentile of the number of FP per day result 0.56 and 2.13 respectively. Even though this may seem a too sensitive threshold that may result in a high rate of FP, it is a conservative solution that avoids the missing of significant anomalous events (false negatives, FN).

Table 2.2 reports the thresholds obtained considering as a training set a week of historical data ($\sim 600k$ features vectors) of the reference sensor. Their visual representation is depicted in Figure 2.5 as bounding boxes surrounding the feature distributions.

Thresholds for each m_i identify intervals of length 1-2 mg, while, for both p_i and s_i^2 , thresholds for x and z axes are much higher than what we have for y , confirming

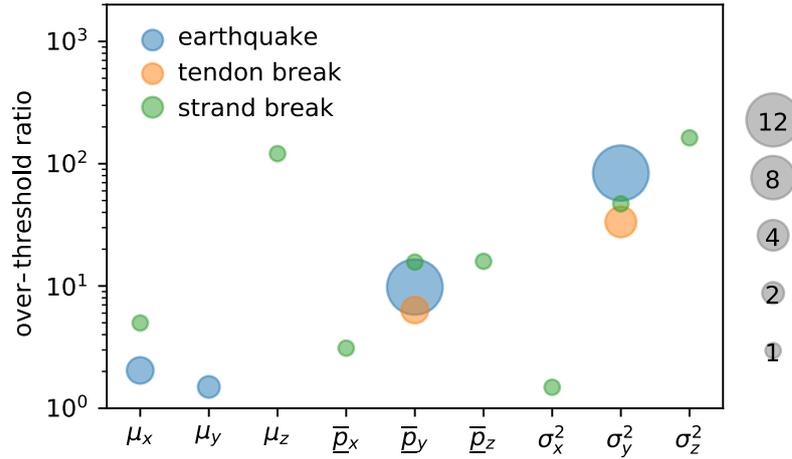


FIGURE 2.6: Characterization of the anomalous events in the feature domain for 30s of acceleration signals containing anomalous events. The height of the points represents the over-threshold ratio, and size counts for the number of instances triggering the alarms.

the fact that oscillations mainly occur on the x - z plane. Similar behaviour characterizes all sensors in the system.

Detection of the anomalies The characterization of the events in the domain of the features is reported in Figure 2.6. The scatter plot shows how much and how many times the recorded features exceeded the thresholds in the 30s reference interval. The point's height represents the ratio between maximum recorded value and the threshold it triggers¹ while the size is proportional to the number of instances triggering the alarms. For the first two events, the signal manifests anomalous behaviour mainly in the y axis where variance and peak-to-peak reach values of one or two orders of magnitude higher than the threshold and the alarms are triggered for several seconds. Conversely, the strand break is detected only in one instance, and all features exceed the thresholds except for m_y .

2.3.2 Frequency domain analysis

Frequency analysis is a powerful tool to extract information on intrinsic properties of civil structures [67, 50, 8, 47], and the algorithm considered here focuses on the identification and tracking of the tendons natural frequencies. Since each tendon has practically different mass and tension, their frequency response is slightly different, and the analysis is performed on each sensor independently.

Features consist of the position, amplitude, and width of the peaks composing the frequency response to distinguish peaks and track their movements. As mentioned, the analysis is performed on 1h basis, i.e., on data packages containing 18 successive MQTT payloads. The algorithm is split into two steps: first, the frequency

¹For over-threshold ratio values associated to m_i , we refer to the ratio between the distance of the current value from the average of m_i and the distance of the triggered threshold from the average of m_i .

response is obtained every hour by estimating the acceleration Power Spectrum Density (PSD), then a peak picking stage extracts the natural frequencies and their properties.

Note that, in this analysis, the signal coincides with the data contained in a 1 h interval, i.e., on each axis i a window consists of $n_i = 3600$ ksamples so that the whole signal x aggregating the acceleration of the three axes has dimension $n = 10.8 \times 10^6$. We refer to the elements of x regarding the i -axis as x_i . High dimensionality is needed to obtain a sufficient frequency resolution to distinguish the different peaks in the spectrum with a low estimation variance.

Given its non-parametric nature and low computational cost, Welch's method is one of the most common methods for acceleration PSD estimation in SHM application. The method works independently on each axis i and consists in dividing the entire period into N_s overlapping segments $a_i[j]$ with $j = 1, \dots, N_s$ and in applying the Fast Fourier Transform (FFT) with an appropriate window function w . Then, the squared module of each element is averaged over the different segments.

In the considered application, FFT is computed adopting the Hann window function, and the period on which the method is applied is 1 h which is split into $N_s = 18$ non-overlapping windows (corresponding to a frequency resolution of 5 mHz). As a result, at each hour, the PSD of the acceleration on the i -axis P_i is estimated as follows:

$$P_i = \frac{1}{N_s} \sum_{j=1}^{N_s} |\text{FFT}(w \circ a_i[j])|^2 \quad (2.4)$$

where \circ represents the element-wise product between two vectors.

The choice of non-overlapping segments allows computing FFTs and accumulating the results of every new MQTT message with a $L \times$ memory usage reduction with respect to the straightforward approach where the entire data package is stored.

The task of finding peaks in a frequency response obtained through FFT-based methods is not trivial due to a large number of additional peaks introduced by the spectrum estimation. Here we perform the task by imposing the following constraints: *i*) minimum peak height, fixed to $5 \times$ the noise floor (estimated with the median of the frequency response); *ii*) minimum distance between peaks, fixed to 0.02 mHz; *iii*) minimum peak height-prominence² ratio, set to 0.7. The output consists of a list of peaks features: amplitude, frequency, width, and prominence.

In the considered SHM system, batch PSD estimation and peak picking are calculated by the gateways while tracking peaks frequency, amplitude, width, and prominence is a task executed on the cloud platform in the decision-maker block. The latter block provides long-term structural health monitoring based on features in the frequency domain, and an alarm is generated, with human supervision, when a deviation from the current state is observed.

²We refer to prominence as the minimum height necessary to descend to get from the summit to any higher terrain.

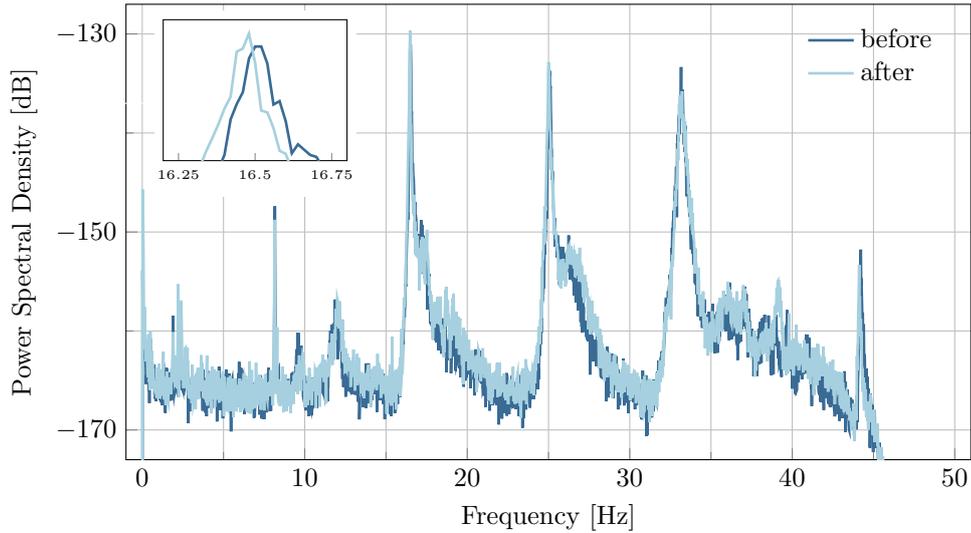


FIGURE 2.7: x axis acceleration PSD estimated in the hour before and after the strand break event.

TABLE 2.3: Frequency of the peaks detected in the PSD of the acceleration signal before and after the strand break.

	1 st peak	2 nd peak	3 rd peak	4 th peak	
before	8.160	16.505	25.070	33.160	Hz
after	8.140	16.470	25.005	33.265	Hz

Detection of the anomalies In the frequency analysis, we consider only the case of the strand break since the event's effect is significantly evident in the frequency domain. Indeed, although non-destructive, the considered strand break modified the tendon elastic properties. On the other hand, the earthquake and the tendon are not localized in the monitored tendon, and, despite the high vibration throughout the structure, they did not cause an evident change in the characteristics of the reference tendon.

Figure 2.7 shows the PSD for the hour before the event (darker colour) compared with the PSD estimated in the next hour (lighter colour). As highlighted in the zoomed area, the event has changed the peak position, slightly shifting. This phenomenon is more evident in Table 2.3. We have the main four peaks at the top, while the row below shows the correspondent peaks in the hour after the event. Despite apparently not being significant, the frequency variations are higher than the typical variation observed on adjacent windows that are at most 5-10mHz.

2.4 Detection with Low Distortion Compression

This section focuses on a specific type of detector proposed in [127] that aims at detecting changes in an observed physical phenomenon with performance similar

to methods based on spectral analysis but with a limited computational effort to fit device resources available at the sensor node or at the edge.

As a working principle, the method observes how the signal's energy is distributed over the signal space and detects an anomaly when the energy along a suitably defined subspace of the signal space differs significantly from what is expected in normal conditions.

The compression mechanism consists in considering a specific subspace of the signal that, by definition, has a lower dimension with respect to the whole signal space. In this dissertation, we refer to this method as low distortion because when the considered subspace is the principal one (i.e., the one that contains most of the signal energy), the transformation that encodes and decodes the original signal is the linear mapping that introduces the minimum distortion as defined in (1.2). Indeed, such a compression principle is exploited in several applications to reduce the dimensionality of a signal.

2.4.1 Lack and Excess of Energy Detectors

Two different detectors are considered, and they differ in the considered subspace. The first measures the energy along the principal subspace to get a quantity that can discriminate the typical behaviour from abnormal. The second considers that, under some circumstances, the projections along the anti-principal subspace may yield information about the system's state. By taking advantage of this, the second detector relies on observing the energy on the anti-principal subspace.

The input signal is modelled as a sequence of n -dimensional vectors $x[t] \in \mathbb{R}^n$ as in (1.1). Here, we suppose that f_x^{ok} and f_x^{ko} are unknown but we assume that $x^{\text{ok}}[t]$ and $x^{\text{ko}}[t]$ are realizations of a ergodic (and thus stationary) stochastic vector processes characterized by null mean vectors and constant covariance matrices $\Sigma^{\text{ok}} \neq \Sigma^{\text{ko}}$ defined as

$$\Sigma^\times = \mathbf{E}_x \left[x^\times[t] x^\times[t]^\top \right] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N x^\times[t] x^\times[t]^\top \quad (2.5)$$

where \times stands for either ok or ko.

Such matrices are symmetric and positive-semidefinite and thus admit a spectral decomposition $\Sigma^\times = Q^\times \Lambda^\times Q^{\times\top}$ with $Q^\times = \begin{pmatrix} q_1^\times & q_2^\times & \dots & q_n^\times \end{pmatrix}$ an orthonormal matrix with eigenvector columns q_j^\times and Λ^\times a diagonal eigenvalue matrix $\Lambda^\times = \text{diag}(\lambda_1^\times, \lambda_2^\times, \dots, \lambda_n^\times)$ such that $\lambda_j^\times q_j^\times = \Sigma^\times q_j^\times$ and $\lambda_1^\times \geq \lambda_2^\times \geq \dots \geq \lambda_n^\times \geq 0$.

The observable is the energy of the signal along a predefined subspace. To formalize this concept, let $U = \begin{pmatrix} u_1 & u_2 & \dots & u_k \end{pmatrix}$ be an $n \times k$ (with $k \leq n$) matrix with orthonormal columns u_j . For each time instant t , the input signal $x[t]$ is compressed as $y[t] = U^\top x[t]$. For each integer τ and for a given number m of subsequent

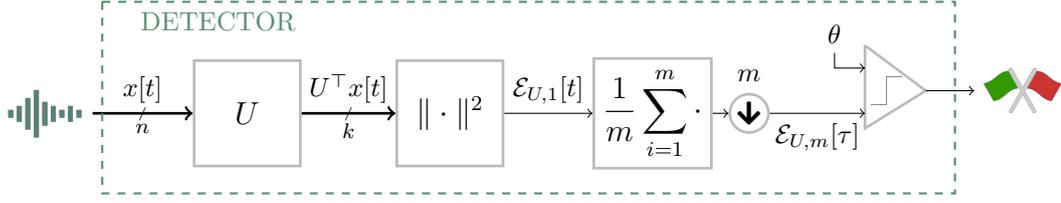


FIGURE 2.8: Block diagram of the Lack/Excess of Energy detectors.

instances, the observable is following value

$$\mathcal{E}_{U,m}[\tau] = \frac{1}{m} \sum_{t=1+m(\tau-1)}^{m\tau} \left\| U^\top x[t] \right\|^2 = \frac{1}{m} \sum_{t=1+m(\tau-1)}^{m\tau} \mathcal{E}_{U,1}[t] \quad (2.6)$$

where $\|\cdot\|$ is the standard ℓ_2 norm of a vector.

Anomalous instances are identified as those corresponding to anomalous values of $\mathcal{E}_{U,m}[\tau]$, i.e., of the average over m subsequent energies of the compressed representation $y[t]$ that is the projection of $x[t]$ on the k -dimension linear subspace spanned by the columns of U . The discrimination between a normal or anomalous set of instances is determined by a threshold θ so that an anomaly is declared when either $\mathcal{E}_{U,m}[\tau] \leq \theta$ or $\mathcal{E}_{U,m}[\tau] \geq \theta$.

Figure 2.8 summarizes the mechanism behind the proposed detector showing its main blocks. For each incoming vector $x[t]$, the energy of the projections $U^\top x[t]$ is computed, and the average of m successive windows is adopted to establish if an anomalous event is occurring. From the scheme, one may get that the number of multiply-and-accumulate (MAC) operations is determined by the projection of the signal x onto U (nk MACs), the computation of the energy (k MACs) and the computation of the average (1 MAC for each signal instance). As a result, every nm elements $m(k(n+1)+1)$ operations are required, i.e., at most $k+1$ MACs for each incoming sample.

To see that $\mathcal{E}_{U,m}$ is linked to the correlation between entries of $x[t]$, note that its average conditioned to either ok or ko is, by direct computation from (2.5),

$$\mu_{\mathcal{E}_{U,m}|\times} = \mathbf{E}_x [\mathcal{E}_{U,m}[\tau]|\times] = \text{tr} \left(U^\top \Sigma^\times U \right) \quad (2.7)$$

where the assumption of stationarity allows dropping time indications from the statistics of $\mathcal{E}_{U,m}[\tau]$, and where $\text{tr}(\cdot)$ indicates the trace of a matrix.

A possible choice for U is $u_j = q_j^{\text{ok}}$ for $j = 1, \dots, k$. Since $\lambda_1^{\text{ok}} \geq \lambda_2^{\text{ok}} \geq \dots \geq \lambda_n^{\text{ok}} \geq 0$ this amounts to take U as the principal k -dimensional subspace of the process $x^{\text{ok}}[t]$ and implies $\mu_{\mathcal{E}_{U,m}|\text{ok}} = \sum_{j=1}^k \lambda_j^{\text{ok}}$, that is the largest possible average energy collected by projecting $x^{\text{ok}}[t]$ onto any k -dimensional subspace. By the very definition of principal subspace the instantaneous energy of the corresponding projection $\mathcal{E}_{U,m}[\tau]$ is expected to be large in normal cases, whereas anomalies can be revealed by the fact that $\mathcal{E}_{U,m}[\tau]$ falls below a certain threshold $\theta_\mathcal{E}$. This method is indicated as *Lack of Energy Detection* (LoED).

From a dual point of view, we may think of choosing $u_j = q_{n-(j-1)}^{\text{ok}}$ for $j = 1, \dots, k$ so that $\mu_{\mathcal{E}_{U,m}|\text{ok}} = \sum_{j=1}^k \lambda_{n-(j-1)}^{\text{ok}}$, implying that U is the *anti*-principal k -dimensional subspace, i.e., the subspace along which projections of normal instances have the least possible average energy. By the very definition of *anti*-principal subspace the instantaneous energy of the corresponding projection $\mathcal{E}_{U,m}[\tau]$ is expected to be small in normal cases, and anomalies can be revealed by the fact the $\mathcal{E}_{U,m}[\tau]$ exceeds a certain threshold $\theta_{\mathcal{E}}$. This method is indicated as *Excess of Energy Detection* (EoED).

The effect of signal localization Both LoED and EoED rely on the fact that some subspaces exhibit distinctive features from an energetic point of view, i.e., that the signal's energy distribution is not uniform over the signal space. For this reason, we expect them to behave poorly for close-to-white signals with almost equal eigenvalues $\lambda_1^{\text{ok}} \simeq \lambda_2^{\text{ok}} \simeq \dots \simeq \lambda_n^{\text{ok}}$. Indeed, in that case, the average energy collected by the projection on any k -dimensional subspace is close to k/n times the total energy of the signal and little discrimination is possible.

On the contrary, when the eigenvalues are substantially unbalanced, principal and *anti*-principal subspaces are distinguishing features of the process that may help to identify deviations. The unbalancing of the eigenvalues implying concentration of energy along a preferential direction in the signal space is often quantified by *localization* [120] defined in (1.29). Localization is minimum ($\mathcal{L}_{x^{\text{ok}}} = 0$) for white signals with $\lambda_1^{\text{ok}} = \lambda_2^{\text{ok}} = \dots = \lambda_n^{\text{ok}}$, while it reaches its maximum $\mathcal{L}_{x^{\text{ok}}} = 1 - 1/n$ when all the energy of the signal is concentrated along the first eigenvector q_1^{ok} , i.e., $\lambda_1^{\text{ok}} > 0$ and $\lambda_2^{\text{ok}} = \lambda_3^{\text{ok}} = \dots = \lambda_n^{\text{ok}} = 0$.

The effect of uncorrelated additive white noise Let us assume that each signal instance is affected by additive white noise, uncorrelated with the signal, and with average energy per sample equal to σ^2 . With this, each possible observation can be expressed as $x^\times[t] = \bar{x}^\times[t] + v[t]$ such that $\bar{\Sigma}^\times = \mathbf{E}_{\bar{x}^\times}[\bar{x}^\times[t]\bar{x}^\times[t]^\top]$ and $\mathbf{E}_v[v[t]v[t]^\top] = \sigma^2 I$, where I is the identity matrix and the overbar indicates the noiseless quantities. Since noise and signal are assumed to be uncorrelated, we concentrate on a signal without anomalies and write

$$\Sigma^{\text{ok}} = \bar{\Sigma}^{\text{ok}} + \sigma^2 I \quad (2.8)$$

in which $\text{tr}(\bar{\Sigma}^{\text{ok}})/n\sigma^2$ represents the Signal-to-Noise Ratio (SNR).

Note that the eigenvectors of Σ^{ok} coincide with those of $\bar{\Sigma}^{\text{ok}}$ while the eigenvalues are such that $\lambda_j^{\text{ok}} = \bar{\lambda}_j^{\text{ok}} + \sigma^2$. Due to the offset, the observed normal signal is less localized compared to the noiseless normal signal, and detector performance is expected to decrease in small SNR scenarios.

2.4.2 Performance in a Gaussian Framework

To derive analytical and semi-analytical guidelines, we consider the classical theoretical setting in which the process produces vectors $x[t]$ that are independent realizations of a zero-mean Gaussian vector whose correlation matrix is either Σ^{ok} (in the normal cases) or Σ^{ko} (in the anomalous cases).

Since detectors rely on spotting significant deviations from most common behaviours, their performance is qualitatively related to the observable variance in the non-anomalous case. The lower such a variance, the closer the observable to being an *invariant* of the process, and deviations from invariant behaviours are reliable indicators of anomalies.

This guideline leads us to consider EoED along with LoED. In fact, we may recognize the following property whose proof is in the Appendix.

Property 2.1. *Let $x \in \mathbb{R}^n$ a jointly Gaussian random vector with zero mean and covariance matrix Σ with eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ and corresponding orthonormal eigenvectors q_1, q_2, \dots, q_n . Let $0 \leq j_1 < j_2 < \dots < j_k < n$ be any choice of k indices, $U = \begin{pmatrix} q_{j_1} & q_{j_2} & \dots & q_{j_k} \end{pmatrix}$ and $\mathcal{E}_{U,1} = \|U^\top x\|^2$. Setting $\mu_{\mathcal{E}_{U,1}} = \mathbf{E}_x[\mathcal{E}_{U,1}] = \sum_{j=1}^k \lambda_{j_k}$, we have*

$$\sigma_{\mathcal{E}_{U,1}}^2 = \mathbf{E}_x \left[(\mathcal{E}_{U,1} - \mu_{\mathcal{E}_{U,1}})^2 \right] = 2 \sum_{l=1}^k \lambda_{j_l}^2 \quad (2.9)$$

With this, since the $x[t]$ are independent and equally distributed Gaussian vectors, from (2.6) we have

$$\sigma_{\mathcal{E}_{U,m}}^2 = \mathbf{E}_x \left[(\mathcal{E}_{U,m} - \mu_{\mathcal{E}_{U,m}})^2 \right] = \frac{2}{m} \sum_{l=1}^k \lambda_{j_l}^2 \quad (2.10)$$

where no time indication is needed as we deal with statistics of stationary quantities.

As a consequence, the variance of the energy observed along the *anti*-principal subspace is smaller than the variance of the energy observed along the principal subspace. This difference indicates that EoED has the potential of performing better than LoED. In both cases, increasing m decreases the variance (presumably increasing performance) at the expense of a lower time resolution.

To assess the capabilities of EoED and LoED, we shall explore the detectors' design space exploiting the Gaussian assumption further to derive precise distributions for $\mathcal{E}_{U,m}$. Since x is Gaussian, also $y = U^\top x$ is Gaussian with zero average and covariance matrix $U^\top \Sigma^\times U$, where \times is either ok or ko. Such a matrix can be given a spectral decomposition as in $U^\top \Sigma^\times U = RDR^\top$ with $R \in \mathbb{R}^{k \times k}$ orthonormal and $D = \text{diag}(d_1, \dots, d_k)$. We may now consider the k -dimensional vector $z = D^{-1/2} R^\top U^\top x$ that is also a zero-mean Gaussian vector with covariance $D^{-1/2} R^\top U^\top \Sigma^\times U R D^{-1/2} = I_k$, i.e., its entries are independent normal variables with zero average and unit variance.

Since $U^\top x = RD^{1/2}z$, the observed energy can be recast in terms of z obtaining

$$\mathcal{E}_{U,1} = \left\| U^\top x \right\|^2 = z^\top D^{1/2} R^\top R D^{1/2} z = z^\top D z = \sum_{j=1}^k d_j \chi_j^2(1) \quad (2.11)$$

that is a linear combination with non-negative coefficients of 1-degree-of-freedom, independent chi-square random variables $\chi_j^2(1)$.

For $m > 1$, we may consider (2.6) in which, since the $x[t]$ are independent and equally distributed Gaussian vectors, so are the summands $\mathcal{E}_{U,1}[t]$. Hence

$$\mathcal{E}_{U,m}[\tau] = \frac{1}{m} \sum_{j=1}^k d_j \chi_j^2(m) \quad (2.12)$$

that is the average of a linear combination of m -degree-of-freedom independent random variables $\chi_j^2(m)$ with the same coefficients as in (2.11).

A plethora of analytical and numerical results are available for linear combinations in (2.12) (see [139, 66] and references therein) allowing the numerical evaluation of the cumulative distribution function (cdf) of $\mathcal{E}_{U,m}$ as

$$F_{\mathcal{E}_{U,m}}(\xi) = \Pr \{ \mathcal{E}_{U,m} \leq \xi \} = \sum_{l=0}^{\infty} c_l \frac{\Gamma \left(mk + l, \frac{\xi}{2d_1} \right)}{\Gamma (mk + l)} \quad (2.13)$$

where $\Gamma(a, b) = \int_0^b \zeta^{a-1} e^{-\zeta} d\zeta$ is the lower incomplete Gamma function, whose complete version is $\Gamma(a) = \Gamma(a, \infty)$, and the sequence of coefficients c_j for $j = 0, 1, \dots$ is computed from the coefficients d_1, \dots, d_k following [139]. The method yielding (2.13) allows to compute figures of merit such as the p_{TP} and p_{FP} (defined in (1.33) and (1.34)) and therefore metrics such as the AUC.

Performance figures Similarly to what was done in Section 1.3.3, to quantify detection performance independently from the choice of thresholds, we employ a figure of merit that is related to the area under the ROC curve (AUC). In detail, we employ the following loss

$$L = 1 - \text{AUC} \quad (2.14)$$

which corresponds to the probability for the detector to be faulty, i.e., the probability for a normal instance to score higher than an anomaly, as it can be directly induced from (1.37). The loss L also corresponds to the area under the DET curve that is sometimes preferred to the ROC as it highlights a difference in detectors performances in the critical regions [129].

Design Space Similarly to the analysis in Section 1.3, a semi-analytical assessment may assume that the normal process is made of independent zero-mean Gaussian

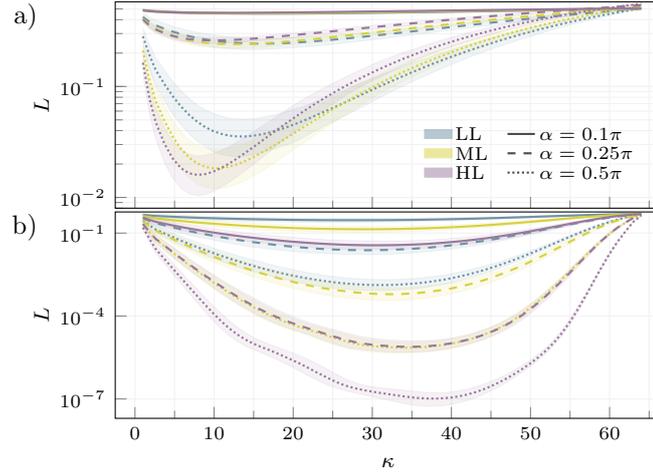


FIGURE 2.9: The loss L of LoED (a) and EoED (b) as a function of k for $m = 1$, with different levels of signal localization and deviation of the anomaly from the signal. Lines are median values while shaded areas show where 99% of the values fall.

vectors with $\Sigma_{j,k}^{\text{ok}} = \omega^{|j-k|}$ for $j, k = 1, \dots, n$ so that the parameter ω controls the signal localization defined in (1.29). A rather straightforward computation [122, Chapter 2] allows to obtain the localization $\mathcal{L}_x = \frac{2\omega^2 n(1-\omega^2) + \omega^{2n} - 1}{n(1-\omega^2)^2}$ that we use to choose values for ω implying normal signals with three different levels of localization, i.e., $\mathcal{L}_x = 0.02$ (indicated as Low Localization - LL), $\mathcal{L}_x = 0.05$ (indicated as Medium Localization - ML), and $\mathcal{L}_x = 0.1$ (indicated as High Localization - HL). As reported in Section 1.3, we expect higher localization to benefit the identifiability of anomalies.

The covariance of the anomalous process Σ^{ko} is randomly built as a perturbation of Σ^{ok} . In formulas, we extract the square root $A = (\Sigma^{\text{ok}})^{1/2}$ such as the matrix that $A^\top A = \Sigma^{\text{ok}}$ and consider its columns a_j for $j = 1, \dots, n$ that are such that $a_j^\top a_j = \Sigma_{j,j}^{\text{ok}} = 1$. We then rotate each a_j in a random direction by a certain angle α to obtain the columns of a matrix A_α from which we set $\Sigma^{\text{ko}} = A_\alpha^\top A_\alpha$, that can be seen as a perturbation of Σ^{ok} preserving the average energy of each component of $x[t]$, but gradually departing from its second-order statistics as α increases. We explore configurations with $\alpha \in \{0.1\pi, 0.25\pi, 0.5\pi\}$.

Exploration of the Design Space The detector depends on a couple of parameters k and m from which it is possible to obtain the cdf of $\mathcal{E}_{U,m}$ and then evaluate the loss L . Monte Carlo simulations are needed to average over the possible Σ^{ko} for each given value of α , and performance is assessed by averaging the loss L over 1000 trials. Moreover, due to the complexity of the design space, signal dimension is assumed as $n = 64$ that allows running simulations in an amount of time that is reasonable but sufficiently large to resemble a real-world signal window.

Figure 2.9 shows what happens to the loss L of both detectors, LoED and LoED, when the dimensionality k of the projection subspace sweeps from its minimum

$k = 1$ to its maximum $k = 64$, with $m = 1$. In both plots, lines correspond to median values, while the shaded areas contain 99% of the values.

We may observe the same trends for both EoED and LoED. First, plots show that the higher the α , the better the maximum performance that detectors may attain. Indeed, given a localization, i.e., given a colour in Figure 2.9, dotted lines ($\alpha = 0.5\pi$) can produce lower losses with respect to dashed lines ($\alpha = 0.25\pi$) that, in turn, do the same compared to solid lines ($\alpha = 0.1\pi$). This behaviour is expected since, as α increases, each anomalous $x[t]$ contains more information that can be used to reject the ok hypothesis in favour of the right ko hypothesis.

The same happens when the localization of the normal signal increases. Given a perturbation angle, i.e., given a line style in Figure 2.9, pink tracks (HL signals) reach lower losses with respect to yellow tracks (ML signals) that, in turn, reach lower losses compared to blue tracks (LL signals). As expected, more localized normal signals concentrate a more significant fraction of the energy in smaller dimensional subspaces and thus are easier to distinguish from anomalies.

Beyond these shared features, the vertical ranges of the two figures are different and highlight that EoED can perform much better than LoED. However, minimum losses localize at different values of k , especially for high localization cases.

To explain this aspect, consider an extreme case in which $\lambda_j^{\text{ok}} > 0$ only for $j = 1, \dots, \bar{j}$ with a certain $\bar{j} \ll n$. An EoED with $k = 1$ exploits the fact that the normal signal has no energy $\lambda_n^{\text{ok}} = 0$ along q_n^{ok} and declares an anomaly when the instantaneous energy is larger than a certain θ_ε . Such a detector never yields a false positive. However, an instance can be anomalous because it has energy along q_{n-1}^{ok} (that a normal signal does not have since $\lambda_{n-1}^{\text{ok}} = 0$ if $n - 1 \geq \bar{j}$). Such an anomaly would go unnoticed unless we set $k = 2$. This increase still causes no false positive but, assuming $n - 2 \geq \bar{j}$, leaves out anomalies that distribute their energy along q_{n-2}^{ok} . Following this path k can be increased until $k = n - \bar{j}$ to yield no false positive but capture all anomalies that feature energies in all the directions along which the normal signal does not. On the contrary, if we set $k = n - \bar{j} + 1$, then a normal signal that has energy along $q_{\bar{j}}^{\text{ok}}$ may produce a false positive, thus increasing the detector loss. Therefore, maximum performance is at $k = n - \bar{j}$, which is quite high since $\bar{j} \ll n$.

LoED would behave in a precisely complementary way since increasing k beyond \bar{j} causes the detector to aggregate energy along with the directions that do not contribute in the normal case and thus may cause false negatives when those directions contain energy from anomalous instances.

Figure 2.10 shows the effect of averaging on the performance of the detectors. In particular, the loss L is plotted against k for both LoED and EoED when trying to discriminate a LL signal from an anomaly whose second-order statistics is only slightly different from the normal one, i.e., for $\alpha = 0.1\pi$. The trends show how reducing the observable variance can be fundamental, improving detectors' performance.

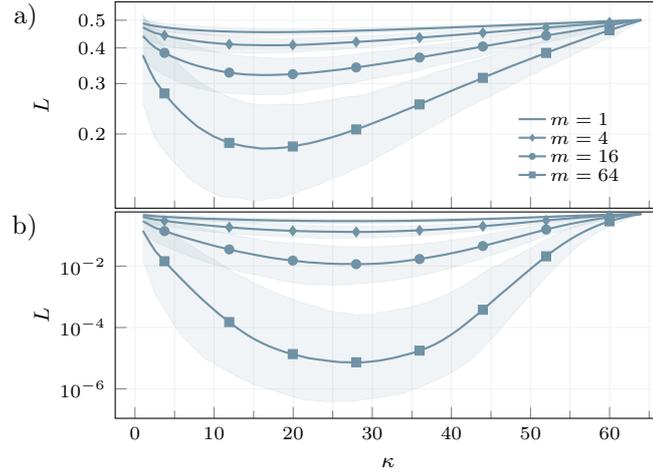


FIGURE 2.10: The loss L of LoED (a) and EoED (b) as a function of k of the projection subspace, for LL signals, a small difference $\alpha = 0.1\pi$ between the normal and anomalous statistics, and various level of averaging m . Lines are median values while shaded areas show where 99% of the values fall.

2.4.3 Performance in SHM

Considering the SHM system described in Section 2.2, the application of the LoED and EoED methods focus on the analysis of a single sensor. The streams of samples corresponding to the three axes of the acceleration are partitioned in chunks of 100 samples corresponding to a time window with length 1 s. Every second, the chunks corresponding to the three axes are arranged in a single vector, generating a sequence of $n = 300$ -dimensional vectors. These vectors account for the structure's time- and space- behaviour as perceived by that sensor. Since the traffic on the viaduct is intermittent, windows are pre-filtered to guarantee that those contributing to anomaly detection exhibit a good signal level compared to background noise. Moreover, to focus on the correlations without the bias of the signal magnitude, each window is normalized to have zero mean and unit energy. As a consequence, $x[t]$ refers to a zero-mean and unit-variance vector with dimension $n = 300$.

Using such a sieved sequence of normalized windows, Σ^{ok} is estimated with the following formula

$$\Sigma_*^{\text{ok}} = \frac{1}{N-1} \sum_{t=1}^N x[t]x[t]^\top \quad (2.15)$$

where N is the number of signal instances employed in the computation that, in this specific case, is $N = 4.5 \times 10^5$.

As a result of a tuning phase, we fix the dimension of the principal and *anti*-principal subspaces of Σ_*^{ok} to $k = 20$ and $k = 70$, respectively. When analyzing the *anti*-principal subspace, we realize that the 30 less energetic directions collect average energy that is negligible ($\leq 0.01\%$) compared to what projects on the other

directions. This observation means that, despite the overall signal being $n = 300$ -dimensional, it can be safely embedded in a 270-dimensional space. Accordingly, from now on, the *anti*-principal subspace used in EoED will span only the 40 directions obtained by discarding the 30 least energetic components from the least 70. Finally, let us refer to the matrix containing the eigenvectors spanning the k -dimensional principal and *anti*-principal subspace as U_{\uparrow} and U_{\downarrow} , respectively.

Here we consider the three kinds of anomalies occurred in the monitoring period and described in Section 2.2.2 that are tendon break, strand break, and earthquake. Differently from the analysis in Section 2.3, here we consider a sensor in the first span of the viaduct that is where the tendon break occurs.

Figure 2.11 shows how the three different anomalies are observed through an established analysis in the frequency domain [67, 50, 171] as well as with the LoED and EoED methods. Each column of the figure regards a different anomaly, while each row refers to a different detector. The same figure also reports the behaviour of two scores defined by previous PCA-based detectors [191, 94, 194]: T2 and SPE. The former accounts for the energy of the weighted projection along the first principal components, while the latter accounts for the energy on the orthogonal subspace. Scores are such that high values hint at anomalies.

The first row, i.e., Figures 2.11(a)-(c), shows the signal's spectrum along the x -axis (the one parallel to the ground and orthogonal to the tendon length) before and after the anomalies. The spectrum is estimated by averaging the periodograms over 18 non-overlapping Hanning windows of 200 s each. Hence, the computation of each spectral profile requires 18 Fourier Transforms of 2×10^4 samples. The frequency sensitivity is 5 mHz and is needed to detect variations of peak frequencies that are themselves in the order of few Hertz.

The evolution of the observable of both LoED and EoED for $k = 20$ are shown in Figure 2.11(d)-(f) and Figure 2.11(g)-(i) respectively. Tracks of different colors correspond to different averaging, namely, $m = 1$ (no average), $m = 1800$ (i.e., average over 30 minutes) and $m = 43200$ (i.e., average over 12 h). In each plot, dashed lines indicate the expected value of the observable under normal conditions, computed as the sum of the eigenvalues of Σ^{ok} corresponding to the eigenspaces spanned by the columns of U_{\uparrow} and U_{\downarrow} . Profiles for the reference scores T2 and SPE are reported in Figure 2.11(l)-(n) and Figure 2.11(o)-(q) respectively. To keep the computational complexity unaltered with respect to the LoED and EoED cases, these scores are computed considering the first 20 principal components for T2 and the corresponding residue for SPE. In all cases, the trends run for 14 days centred on the day on which the event happened, and Σ^{ok} is estimated by considering a week of data preceding the observation period.

Firstly, we consider the tendon brake, which is the more evident anomaly that any detector should notice. The tendon break drastically changes its elastic properties, as confirmed in Figure 2.11(a), in which the spectrum after the event is noticeably different from the one before. The event is also immediately detectable both

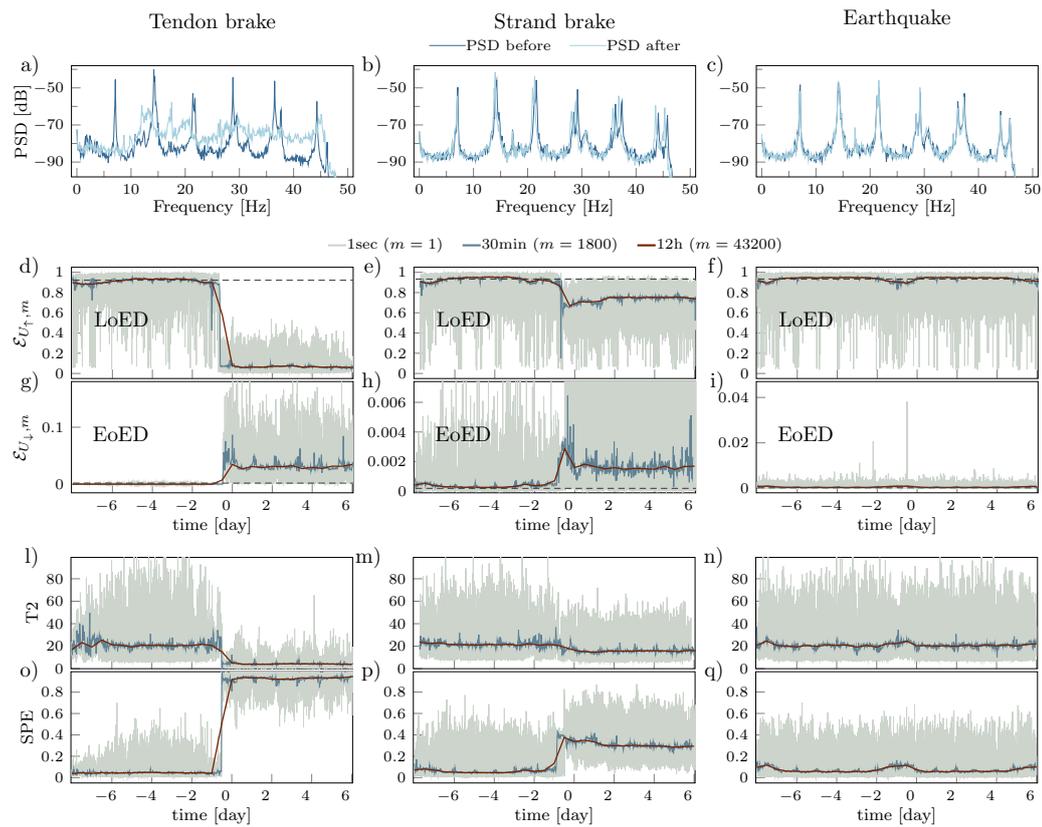


FIGURE 2.11: The tendon break, strand break and earthquake anomalies observed by means of conventional spectral analysis (PSD), both EoED and LoED ($\kappa = 20$), and scores T2 and SPE (20 principal components), for different values of m . Dashed lines indicate the expected value of the observables in normal conditions.

from EoED and LoED observables shown in Figure 2.11(d) and 2.11(g) where the new regime of the observables is utterly different from the previous one. According to their definitions, the observable of LoED decreases while the observable of EoED increases. This behaviour is also the case of the score SPE in Figure 2.11(o) while the profile of T2 in Figure 2.11(l) shows a decrease in the score that makes T2 insensitive to the tendon break.

The second column of Figure 2.11 refers to the strand break. Figure 2.11(b) shows that the occurred event results in a $\simeq 2\%$ downshift for all the harmonics. Though this is a subtle change, it is still detectable from an accurate analysis of the tendon frequency response that considers suitably long data windows. The anomaly is also visible in both the EoED and LoED observables depicted in Figure 2.11(e) and Figure 2.11(h). A permanent deviation from the observables expected values could be seen in both plots, even if their magnitude is smaller than those measured after the tendon break. Considering the cases of T2 and SPE, plots in Figure 2.11(m) and Figure 2.11(p) show that, as before, a permanent deviation is evident in SPE only while T2 exhibits a feeble drop that, again, makes it insensitive to this anomaly. In both tendon and strand cases, the comparison between trends for different values of m shows how averaging is fundamental in reducing the variance of the observables. Averaging does not impair detection capabilities if the events to reveal either have a non-negligible duration or have long-lasting effects whose period is larger than the averaging window while it filters out impulsive anomalies.

A representative of this last class of anomalies is an earthquake that hit the viaduct. The earthquake lasted few seconds and the frequency response of Figure 2.11(c), as well as the averaged observables of Figure 2.11(f), Figure 2.11(i), Figure 2.11(n) and Figure 2.11(q), do not give any hint of it. However, when $m = 1$, EoED highlights that something anomalous was happening. This behaviour is possible because, even for small m , the observable of EoED features a small variance that lets the high-energy event stand out from normal conditions. On the other hand, a much higher variance characterizes LoED, T2 and SPE scores, and thus, they exhibit values comparable to those assumed during the earthquake even in normal conditions.

Conclusion

This chapter considers a real Structural Health Monitoring System as an application for detectors operating on compressed signals. The monitoring system installed on the highway viaduct acquires the oscillation of the structure induced by the traffic, and the data collected during the monitoring period permits defining a normal behaviour with different embodiment depending on the detection approaches. Some anomalous events occurred over the monitoring period, and we consider three of

them: earthquake, strand break, and tendon break. These structural anomalies manifested differently and had different consequences, allowing us to validate and test the detection methods.

The two investigated approaches for anomaly detection on compressed signals explore two corner cases: detection with high and low distortion compression. The former approach squeezes the information in very few quantities that summarize the informational content of the signal for the detector to distinguish between normal and anomalous signal instances. We split the features into two categories depending on the domain: for each acceleration axis, three statistical measures in the time domain (mean, variance and peak-to-peak) and the frequency of the peaks in the acceleration spectrum. Those features do not allow signal recovery, and, for this reason, we refer to it as high distortion compression. However, Section 2.3 shows how these features can be effectively employed in an anomaly detector.

We refer to the other approach as detection with low distortion compression and consists of a detector based on Principal Component Analysis. PCA is employed for compression, i.e., representing the signal in a latent space with a lower dimension. The principal components represent the typical patterns and are the directions of the signal space bringing the most information. Projecting the signal onto these components makes it possible to capture the information content necessary for signal recovery. The higher the energy collected, the closer the signal is to the typical patterns encoded in the principal components, i.e., the more ordinary is the instance. The proposed detector LoED exploits this mechanism to discriminate between normal and anomalous signal instances: the amount of energy collected by the principal components indicates how normal the instance is, and classification is achieved by appropriately setting a threshold. The dual detector EoED is also investigated. Indeed, considering the *anti*-principal component, one may expect to observe very low energy, and the instance is likely to be an anomaly when the energy is over a certain threshold.

For both LoED and EoED, we first assess their performance in a synthetic setting. This framework allows us to control the input signal localization and the deviation of the anomaly from normality. Then both methods were tested on the data set provided by the considered SHM system. The methods can identify anomalies generated by destructive events causing permanent structural changes and slight alterations in the viaduct elastic properties. Moreover, EoED can even identify anomalies that do not permanently affect the structure.

Part II

Streaming Principal Component Analysis

Chapter 3

Streaming Subspace Analysis

The idea behind Subspace Analysis (SA) consists of finding a subspace of the signal space with some unique properties. SA divides into principal (PSA) and minor (MSA) subspace analysis depending on the type of subspace. In PSA, the wanted subspace is the one that explains most of the signal variance, i.e., when the signal is projected in such subspace, the energy of the new representation is maximized. MSA seeks the dual subspace, i.e., the subspace representing the signal's direction vanishing energy/variance. When the sought subspace is represented with a basis whose elements are uncorrelated, the basis elements takes the name of components, and PSA specializes in Principal Components Analysis (PCA) while MSA turns into Minor Component Analysis (MSA).

Historically, PCA/PSA has a wide variety of applications, among which, for example, pattern identification in computer network traffic analysis [200], as well as anomaly detection [127], biomedical application [165, 193], blind source separation [204], surveillance [177, 59], though the list is by far not exhaustive. In Chapter 2 and in particular in Section 2.4, we have seen an application of both PCA and MCA for anomaly detection. Indeed, LoED observes the signal energy along the principal components to distinguish between a normal or anomalous signal instance. At the same time, EoED performs the same task by focusing on the energy on the minor (or *anti-principal*) components.

Identifying and tracking principal and minor subspaces is becoming especially important as data availability and redundancy increase. It may be highly inefficient to transport and store all this data when a large part is eventually discarded when processed to extract the information content. For this reason, it is essential to reduce the data dimension close to where the data is generated.

In monitoring systems, signals are typically highly-dimensional with a noteworthy level of redundancy and are usually acquired by devices with limited computational resources. Hence, lightweight methods should be adopted to exploit the advantages of early dimensionality reduction. In this sense, subspace analysis is an attractive solution as it only requires the projections of the n -dimensional signal onto the k principal/minor components (with $k < n$) modelled as a linear transformation. Moreover, the same linear transformation is also the only operator needed to transform the compressed version back to the original signal space.

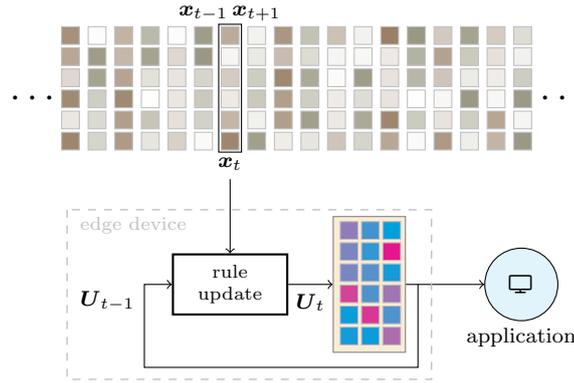


FIGURE 3.1: General scheme of a streaming algorithm for subspace analysis.

However, the principal/minor subspace must be first determined, and this operation often requires a considerable amount of memory and computing resources. Indeed, the most straightforward solution consists of *batch* methods, such as Eigenvalue Decomposition (EVD) or Singular-Value Decomposition (SVD) [98]. Both techniques require acquiring and storing many signal instances and their processing by algorithms that use standard linear algebra computations. Hence, batch methods imply extensive memory usage and have a complexity in the order of at least $\mathcal{O}(Nn^2)$, where n is the original dimension of the data, and N is the number of signal examples composing the data set.

In this light, a number of streaming algorithms (e.g. [17, 128, 108]) have been recently presented. Here, data instances are not stored but processed as they are available, resulting in complexity of $\mathcal{O}(nk^2)$ or even $\mathcal{O}(nk)$ at the price of some approximation. Figure 3.1 aims at representing a general scheme of streaming algorithms for subspace analysis, in which the current estimate of the sought subspace is represented by a matrix U_t that is updated at each element x_t of the data set viewed as a stream.

The algorithm that updates U_t is a streaming subspace analysis method, and in this chapter, we review some of the many proposals. Often the different approaches only differ for subtle details. For this reason, we here select the algorithms that incarnate main principles, that are sufficiently lightweight to fit in a low-resources device, and have performances that are representative of what can be obtained even by more specialized implementations. For example, we only consider purely streaming approaches as block-wise methods are usually a straightforward extension of sample-wise methods but require more extensive memory and computation resources without significantly improving subspace identification.

The chapter starts with Section 3.1 that provides a possible classification of the streaming algorithms from different points of view. Then, in Section 3.1 the selected methods are introduced and described with emphasis on the relation between the different methods and highlighting the differences with the other methods not considered in this dissertation. All methods are expressed within the same coherent

framework that facilitates comparisons in terms of both functionality (estimation accuracy) and resource needs. This comparison is then discussed in Section 3.3 for the task of Subspace identification. Finally, we investigate and discuss in Section 3.4 the deployment of these methods on edge devices distinguishing high-end from low-end devices. The former class of devices usually features an operating system that supports high-level programming languages, while the latter requires direct programming.

3.1 Streaming Methods Classification

We model the data stream as a discrete-time stochastic process that generates occurrences $x_t \in \mathbb{R}^n$ with $t = 1, 2, \dots$. The algorithms we describe process vectors x_t sequentially to extract a characterization of the whole data-set. We assume the statistical characterization of x_t to be constant or slowly variant, such that, x can represent any possible x_t . We also limit the analysis to the case $\mathbf{E}[x] = 0_n$, where 0_n is the n -dimensional null vector and $\mathbf{E}[\cdot]$ indicates the expectation operator. As a result, the covariance/correlation matrix is $\Sigma = \mathbf{E}[xx^\top]$, where \cdot^\top indicates transposition.

As for any correlation matrix, Σ has an EVD $\Sigma = \Psi\Lambda\Psi^\top$, Λ is a diagonal matrix with diagonal entries $\lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_{n-1} \geq 0$, and the columns of $\Psi \in \mathbb{R}^{n \times n}$ are the corresponding eigenvectors $\{\psi_0, \dots, \psi_{n-1}\}$ in orthonormal form.

Streaming algorithms aim at identifying a matrix $U \in \mathbb{R}^{n \times k}$ that is tied to either the so-called *principal subspace*, i.e., the one spanned by the $m < n$ eigenvectors associated with the largest eigenvalues, or the so-called *minor subspace* or *noise subspace*, i.e., the one spanned by the remaining $n - m$ eigenvectors associated with the smallest eigenvalues. Note that $k = m$ or $k = n - m$ for the principal or the minor case. In several applications, $m \ll n$ [191] such that the size of U drastically changes in passing from the principal to the minor subspace identification task. To identify the target matrix U , streaming algorithms follow an iterative procedure that updates an estimation U_t every time a new vector x_t is acquired such that $U_t \rightarrow U$ when t grows.

The methods we review estimate U in different ways. Here we list some key features that can make distinctions or draw connections.

principal and minor subspaces Most algorithms are able to target principal subspaces. Some of them can also target minor subspaces, and few methods are designed for the minor subspace only.

eigenvectors or subspaces When the target matrix U is $\Psi_{|m} = [\psi_1, \dots, \psi_m]$ (principal components) or $\Psi_{|m} = [\psi_{n-m+1}, \dots, \psi_n]$ (minor components), we refer to eigenvectors estimation. Alternatively, subspace estimation relates to U spanning the same subspace of the eigenvectors in either $\Psi_{|m}$ or $\Psi_{|m}$.

objective function As for many iterative methods, some streaming algorithms for subspace estimation derive from iterative procedures solving a minimization or maximization problem. We provide a classification according to the objective function characterizing the optimization problem.

Identifying the k -dimensional principal (minor) subspace consists in finding the column-orthonormal matrix $U \in \mathbb{R}^{n \times k}$ that maximizes (minimizes) the variance of the projection $y = U^\top x$ [175, 180]. Hence, some approaches consider the objective function

$$J_{\text{Var}}(U) = \mathbf{E} \left[\left\| U^\top x \right\|^2 \right] = \text{tr} \left[U^\top \Sigma U \right] \quad (3.1)$$

where $\|\cdot\|$ denotes the l_2 norm, and U is constrained to be column-orthonormal.

Maximizing (minimizing) projection variance is equivalent to minimizing (maximizing) the average norm of residual vector $r = x - UU^\top x$. Hence, some methods [11, 186] consider the objective function

$$J_{\text{MSE}}(U) = \mathbf{E} \left[\left\| x - UU^\top x \right\|^2 \right] \quad (3.2)$$

As a further alternative, it is possible to define an objective function based on the so-called *spiked model* [97] whereby the observable x is assumed as an expansion of an m -dimensional signal s such that $s = \Psi_m^\top x$ [18, 53]. Despite their obvious link, some methods consider x and s separately and minimize (3.3) to enforce their relationship.

$$J_{\text{SM}}(U) = \mathbf{E} \left[\left\| x - Us \right\|^2 \right] \quad (3.3)$$

As a remark, the signal model behind (3.3) make this objective function effective only in case of principal subspace identification, i.e., $J_{\text{SM}}(U) = 0$ implies $\text{span}(U) = \text{span}(\Psi_m)$.

column-orthonormality Given that the target U is column-orthonormal, it is possible to distinguish between two classes of algorithms depending on the properties of the current estimate U_t : *i)* every U_t is constrained to be column-orthonormal, *ii)* U_t is only approximately column-orthonormal. In general, algorithms ensuring orthonormality have complexity $\mathcal{O}(nk^2)$, while algorithms limiting to approximate orthonormality are able to achieve $\mathcal{O}(nk)$.

Those ensuring orthonormality can be further distinguished depending on the technique used. Some do so by applying a specific orthonormalization procedure (typically a QR-decomposition) at each step. In this case, if the update produces a matrix U'_t that is not column-orthonormal, then the columns of the final U_t span the same subspace as the columns of U'_t , while being orthonormal.

Some others do so by projecting onto the Grassmannian manifold [62, 180] that contains all possible $n \times k$ orthonormal matrices. Using projection implies that the span of the result of the non-orthonormal update U'_t is not necessarily the same as that

of the final orthonormal U_t , and some of the improvements in selecting U'_t might be lost.

As a third option, orthonormality can be guaranteed by constraining the updates of U_t to be along Grassmannian geodesics. In this case, the updated estimation improves over the previous one without leaving the acceptability region.

Table 3.1 classifies the methods, whose detailed description is given in the next Section, along directions from 1) to 4) above. In the same Table, we also report the computational complexity and the field of interest in which the methods were originally conceived.

3.2 Streaming Methods

This section describes the methods we then test and implement. For each method, we give the *update step*, i.e., the sequence of operations leading from the matrix U_{t-1} that is the estimate of the subspace after $t - 1$ input observations, to the matrix U_t that is the current estimation and takes into account also the occurrence associated to x_t .

Most update steps compute intermediate quantities such as $y_t = U_{t-1}^\top x_t$ that is the vector of coefficients expressing the projection of x_t onto the subspaces as it is estimated at time $t - 1$, and $r_t = x_t - U_{t-1}y_t$ that is the residual of such a projection. Note that if U_{t-1} is orthonormal, the two vectors y_t and r_t are orthogonal.

For the sake of brevity, the computation of y_t and r_t are not explicitly mentioned in the descriptions of the methods.

3.2.1 Oja's method

Originally proposed in [145] for the principal subspace estimation with $k = m = 1$, and then extended to the rank- k cases in [146] it starts from a random column-orthonormal $U_0 \in \mathbb{R}^{n \times k}$. At the t -th sample it updates the estimation of $U = \Psi|_k$ according to the input data x_t as

$$U_t = \Omega \left(U_{t-1} + \gamma_t x_t x_t^\top U_{t-1} \right) \quad (3.4)$$

where $\Omega(\cdot)$ is an operator that orthonormalizes the columns of its argument, e.g., gives the the Q matrix in the QR decomposition of its argument [90, Chapter 2]. The parameter γ_t is the step size or learning rate that may change with time.

Notably, authors of [11, 137, 88] show that Oja is an extension of the well-know power method [81] that, in turn, is equivalent to solving a maximization problem where the objective function is (3.1) and U_t is constrained to be column-orthonormal. In particular, the gradient of (3.1) with respect to U is ΣU such that (3.4) is equivalent to the update of a stochastic gradient descent algorithm where Σ is approximated by $x_t x_t^\top$, γ_t is the learning rate, and $\Omega(\cdot)$ forces the update to yield a column-orthonormal matrix.

TABLE 3.1: Classification of methods for streaming Subspace Analysis

Method	Principal/minor subspace	Eigenvalues or subspace	Objective function	Column orthonormalization	Complexity	Original field
Oja	[145][146][11][137][88][6]	both	eigenvalues	(3.2)	QR	nk^2 neural networks
Krasulina	[106][105][175]	both	eigenvalues	(3.1)	QR	nk^2
HFRANS	[2][13][14][12]	minor	eigenvalues	(3.1)	approximate	nk
PAST	[187][186][84][135][2]	principal	subspace	(3.3)	approximate	nk adaptive filtering
ISVD	[36][102]	principal	eigenvalues	N/A	QR	nk^2
GROUSE	[18][19][196]	principal	subspace	(3.3)	Grassmannian	nk matrix completion

Actually, to save some computation, one may think of applying Ω only after a certain number of updates [11]. However, proper sizing of the number of steps without orthonormalization depends on the application.

Furthermore, when minimization is considered instead of maximization, with the same objective function and constraint, one may aim at identifying the minor subspace $U = \Psi_{|k|}$ with $k = n - m$. With respect to (3.4), here stochastic gradient descent algorithm performing minimization follows the opposite of the gradient of (3.1), i.e.,

$$U_t = \Omega \left(U_{t-1} - \gamma_t x_t x_t^\top U_{t-1} \right) \quad (3.5)$$

Lastly, since the convergence of the method depends on the choice of the initial matrix U_0 , [6] proposes a procedure of warm start that avoids the random initialization.

3.2.2 Krasulina's method

Originally proposed in [106, 105] and recently revised in [175] to include the $k > 1$ case, it also starts from a random column-orthonormal $U_0 \in \mathbb{R}^{n \times k}$. The update step is

$$U_t = \Omega \left(U_{t-1} + \gamma_t r_t v_t^\top \right) \quad (3.6)$$

According to [186, 175], the update in (3.6) converges to a matrix that approximates the minimizer of (3.2).

As for Oja, a warm start approach has been proposed in [175] and the minor subspace can be targeted by simply changing sign of the last equation in (3.6) to yield

$$U_t = \Omega \left(U_{t-1} - \gamma_t r_t v_t^\top \right) \quad (3.7)$$

In this case the method converge to $U = \Psi_{|k|}$ with $k = n - m$.

Oja and Krasulina are deeply linked. In fact, [144] proves that the

$$\Omega \left(U_{t-1} + \gamma_t x_t x_t^\top U_{t-1} \right) = U_{t-1} + \gamma_t r_t y_t^\top + o(\gamma_t^2) \quad (3.8)$$

thus ultimately establishing equivalent convergence properties for (3.4) and (3.6) as $\gamma_t \rightarrow 0$ for growing t .

3.2.3 HFRANS

The analogy of the two previous methods highlighted by (3.8) has led to a class of methods that try to provide column-orthonormality avoiding the Ω operator.

Since $U_t = U_{t-1} + \gamma_t r_t y_t^\top$ tends to be column-orthonormal for $t \rightarrow \infty$ an extreme option is to consider $o(\gamma_t^2)$ negligible and simply avoid orthonormalization.

Such a choice can be acceptable when targeting $U = \Psi_{|k|}$. However, when aiming at identifying $U = \Psi_{|k|}$, the minimal amplitude of the projection of the signal on the

minor subspace makes the whole procedure extremely error-prone and may spoil convergence.

To overcome this impasse, [2] first proposed a term to approximate the $o(\gamma_t^2)$ residue in (3.8) giving raise to the so-called OOja method that is then extended [13] with a policy that adapts γ_t to both U_t and x_t , leading to the so-called NOOja method.

OOja and NOOja further evolved [12] into HFRANS. This method is also an adjustment of FRANS, a previous Rayleigh quotient-based adaptive noise subspace method [14]. In HFRANS, Householder transformations are introduced to grant the numerical stability needed to cope with the minor subspace case.

The method starts from a random $U_0 \in \mathbb{R}^{n \times k}$ matrix and uses the following update rule that depends on a given $0 < \gamma < 2$,

$$\begin{aligned} \tau_t &= \frac{1}{\|y_t\|^2} \left[\left(1 - (2 - \gamma) \gamma \frac{\|y_t\|^2}{\|x_t\|^2} \right)^{-\frac{1}{2}} - 1 \right] \\ \hat{u}_t &= \left(1 + \tau_t \|y_t\|^2 \right) x_t - \frac{\tau_t \|x_t\|^2}{\gamma} U_{t-1} y_t \\ u_t &= \frac{\hat{u}_t}{\|\hat{u}_t\|} \\ U_t &= U_{t-1} - 2u_t u_t^\top U_{t-1} \end{aligned} \quad (3.9)$$

3.2.4 PAST

The Project Approximation Subspace Tracking (PAST) [187, 186] is an algorithm obtained by minimizing (3.3) in which the expectation is unrolled in time as an exponentially weighted sum, i.e., by setting

$$U_t = \arg \min_{U \in \mathbb{R}^{n \times k}} \sum_{l=1}^t \beta^{t-l} \|x_l - U s_l\|^2 \quad (3.10)$$

without the constraint of U_t being column-orthonormal and where $\beta \in [0, 1]$ is the forgetting factor that weights the prior samples. Equation (3.10) is based on the fact that signal observances x_l are generated accordingly to a spiked model, i.e., $s_l = \Psi_{|k}^\top x_l$.

Since, at each signal occurrence, only x_l is known, s_l is approximated by the projection vector $y_l = U_{l-1}^\top x_l$, i.e., by adopting the last estimated U . Thanks to this approximation, the problem has a closed solution, and U_t can be retrieved by mean of recursive least squares (RLS) methods, which allow for a computational cost as low as $\mathcal{O}(nk)$.

Iterations start from a random $U_0 \in \mathbb{R}^{n \times k}$ and $P_0 = \delta I_k$ for some $\delta > 0$ and the update step is

$$\begin{aligned}
g_t &= P_{t-1}y_t / \left(\beta + y_t^\top P_{t-1}y_t \right) \\
P_t &= \beta^{-1} (P_{t-1} - g_t P_{t-1}y_t) \\
U_t &= U_{t-1} - r_t g_t^\top
\end{aligned} \tag{3.11}$$

Though U_t is not guaranteed to be column-orthonormal for any finite t , [186, 135] show that column-orthonormality is achieved asymptotically as t increases. In applications where columns-orthonormality is essential at each step, PAST variants can be adopted. For instance, PASTd in [186] is a version based on the deflation technique that comes at the cost of an increase in complexity. Otherwise, in [2] nearly-orthonormality is provided by correction terms applied to each update of U that keeps complexity at $\mathcal{O}(nk)$.

PAST is the base for other approaches such as [84] that deal with the case in which data is perturbed by coloured noise, [53] that is designed to cope with the missing components in the vectors x_t , and [16, 15] where the Approximated Power Iteration (API) extends the standard power method by exploiting the same approximation used in PAST.

3.2.5 Incremental SVD

Given any $n \times t$ matrix A , Singular Value Decomposition (SVD) [81, Chapter 2] finds three factors P, D and Q such that $A = PDQ^\top$ with P and Q being square orthonormal matrices of size $n \times n$ and $t \times t$ respectively, and D is a diagonal $n \times t$ matrix whose diagonal entries are called *singular values*. SVD is symbolized as $A \xrightarrow{\text{SVD}} P, D, Q$.

Let also $X_t = \begin{bmatrix} x_1 & \dots & x_t \end{bmatrix}$ be the matrix containing samples up to the t -th. Subspace analysis has to do with the SVD of X_t with a very large t . In fact, if $X_t \xrightarrow{\text{SVD}} \bar{U}_t, \bar{S}_t, \bar{V}_t$, then the singular values $\sigma_j(t)$ in \bar{S}_t are such that $\sigma_j^2(t) \rightarrow \lambda_j$ and $\bar{U}_t \rightarrow \Psi$ as $t \rightarrow \infty$.

In [37] it is shown that the SVD of X_t can be effectively computed from the SVD of X_{t-1} . Even more, [36] shows that this is possible even if we focus on the so-called *thin* SVD (tSVD), i.e., on a decomposition that computes only the first k columns of \bar{U}_t and \bar{V}_t , the former set of columns being exactly the matrix U we look for when targeting the principal subspace.

The method relies on the decomposition $X_{t-1} \xrightarrow{\text{tSVD}} U_{t-1}, S_{t-1}, V_{t-1}$ to express the data matrix X_t as

$$X_t = \begin{bmatrix} U_{t-1} & \frac{r_t}{\|r_t\|} \end{bmatrix} \begin{bmatrix} S_{t-1} & y_t \\ 0_k^\top & \|r_t\| \end{bmatrix} \begin{bmatrix} V_{t-1}^\top & 0_k \\ 0_k^\top & 1 \end{bmatrix} \tag{3.12}$$

The equality in (3.12) holds exactly only if the rank of X_{t-1} is k and is otherwise an approximation. Moreover, (3.12) is the foundation of ISVD method, which compute the SVD of the inner matrix at each update. With more details, the method computes

the SVD of

$$\begin{bmatrix} \beta S_{t-1} & y_t \\ 0_k^\top & \|r_t\| \end{bmatrix},$$

which represents an adjusted version of the inner matrix in (3.12) where the parameter $0 < \beta < 1$ is added to control the memory of the algorithm. The computed SVD yields two orthonormal factors P and Q and a diagonal factor D whose product can be plugged into (3.12) to obtain an internal diagonal factor and thus yield the updated tSVD of X_t . Such a representation is then shrunk to the minimum by keeping only the first k columns of the left and right factor and only the first k columns and rows of the central factor.

Overall, starting from a random orthonormal $U_0 \in \mathbb{R}^{n \times k}$, the update step computes

$$\begin{aligned} \begin{bmatrix} \beta S_{t-1} & y_t \\ 0_k^\top & \|r_t\| \end{bmatrix} &\xrightarrow{\text{SVD}} P, D, Q \\ S_t &= (D)_{\Gamma_k} \\ U_t &= \left(\begin{bmatrix} U_{t-1} & \frac{r_t}{\|r_t\|} \end{bmatrix} P \right)_{|_k} \end{aligned} \quad (3.13)$$

where $(\cdot)_{|_k}$ is the same operator used before which selects the first k columns of its argument, while $(\cdot)_{\Gamma_k}$ selects the first k columns and the first k rows of its argument.

Besides, the speed of convergence is highly affected by the condition number of X_t . To partially overcome this problem authors in [102] propose the Polar Incremental Matrix Completion (PIMC) algorithm which adapts the memory factor to the norm of the observed samples $\beta = \frac{a_t}{\|S_t\|_F}$ where $a_t^2 = a_{t-1}^2 + \|x_t\|_2^2$, $a_0 = 1$ and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix.

3.2.6 GROUSE

Grassmannian Rank-One Update Subspace Estimation (GROUSE) is a streaming algorithm for subspace tracking proposed in [18]. Although it is designed to deal with the case in which some components of x_t are unknown, we here consider the version for complete data. The idea consists in applying the stochastic gradient descent to minimize (3.3) while making moves that do not exit the set of all possible column-orthonormal matrices, i.e., the Grassmannian manifold of the k -dimensional subspaces of \mathbb{R}^n .

Starting from one of such matrices, the update rule when all the components of x_t are known is

$$\begin{aligned}
 p_t &= U_{t-1}y_t \\
 \theta_t &= \arctan \left[(1 - \alpha_t) \frac{\|r_t\|}{\|p_t\|} \right] \\
 z_t &= \cos(\theta_t) \frac{p_t}{\|p_t\|} + \sin(\theta_t) \frac{r_t}{\|r_t\|} \\
 U_t &= U_{t-1} + \left(\frac{z_t}{\|z_t\|} - \frac{p_t}{\|p_t\|} \right) \frac{y_t^\top}{\|y_t\|}
 \end{aligned} \tag{3.14}$$

where the expression for θ_t is derived from eq. (3)–(4) in [196] and where α_t is meant to mitigate the effect of noise.

Convergence of GROUSE is analyzed in [19, 196] and [20] shows that GROUSE and ISVD are strictly linked. In particular, the application of the ISVD to the missing data case is equivalent to GROUSE for a particular choice of its parameters.

3.3 Subspace analysis performance

In this section, we test the functional performance of the methods described in Section 3.2 by asking each method to detect a subspace characterizing a set of signal observations.

Data are generated according to the spiked model [97] with the subspace structure of the signal that does not change in time, meaning that

$$x_t = \Phi s_t + v_t \tag{3.15}$$

where $\Phi \in \mathbb{R}^{n \times m}$, with $m < n$, is a column-orthonormal matrix that expands instances $s_t \in \mathbb{R}^m$ of a zero-mean random Gaussian source with zero mean and covariance I_m , while $v_t \in \mathbb{R}^n$ represents realizations of a zero-mean Gaussian noise term with covariance νI_n such that $\nu > 0$ controls the noise level.

With this model we have $\Sigma = \mathbf{E} [x_t x_t^\top] = \Phi \Phi^\top + \nu I_n$, and since Φ is column-orthonormal and $\nu < 1$, then Φ itself spans the m -dimensional principal subspace, while its orthogonal complement Φ^\perp spans the $(n - m)$ -dimensional minor subspace. As a result, the target $n \times k$ matrix U is Φ in the case of principal subspace estimation, and it is Φ^\perp for the minor subspace.

To quantify the effectiveness in subspace analysis, for each method we monitor the sequence of reconstruction errors

$$e_t = \left\| U - U_t U_t^\top U \right\|_F \tag{3.16}$$

where $\|\cdot\|_F$ indicates the Frobenius norm of its argument.

In case of correct estimation we have $U_t = U$ and thus $e_t = 0$ while the error is maximum when U_t is orthogonal to U , yielding $e_t = k$.

TABLE 3.2: Algorithm parameters, tuned value with corresponding ranges and update times (average and std) for the RPi edge-oriented implementation

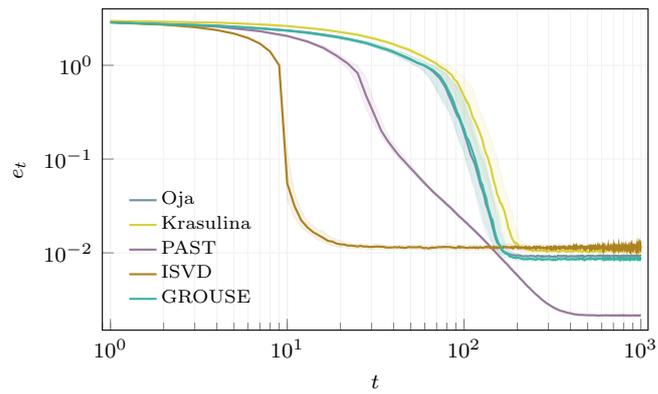
method	parameter	value	range	update time [μ s]
				ave. \pm std
Principal Subspace identification				
Oja	γ	1	[0.001, 100]	354 ± 10
Krasulina	γ	0.1	[0.001, 100]	378 ± 11
PAST	β	0.99	[0.7, 0.999]	116 ± 5
ISVD	β	0.9	[0.7, 0.999]	585 ± 13
GROUSE	α	0.2	[0, 0.5]	304 ± 10
Minor Subspace identification				
Oja	γ	0.1	[0.001, 100]	3446 ± 60
Krasulina	γ	0.1	[0.001, 100]	3472 ± 27
HFRANS	γ	1	[0.001, 100]	540 ± 29

Simulations consider $n = 100$, $m = 10$ and noise amplitude $\nu = 10^{-3}$. For each task, 100 Montecarlo trials are performed where random column-orthonormal Φ and random column-orthonormal initialization matrices U_0 are drawn independently. Each single Montecarlo trial is composed by a stretch of 1000 sample windows x_t generated independently following (3.15).

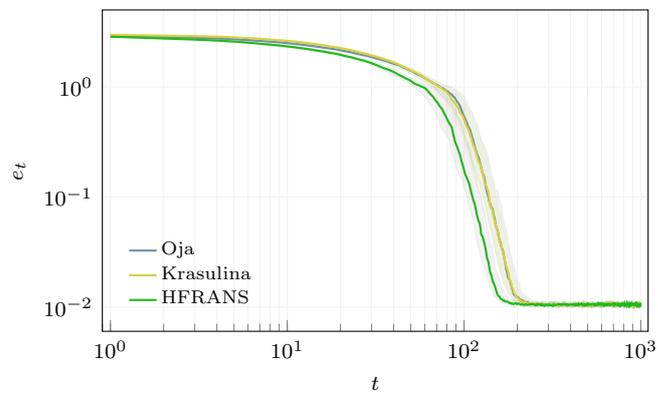
The parameters controlling each method are set as reported in Table 3.2: Oja and Krasulina need a learning rate γ_t , PAST and ISVD need to set a forgetting factor β , and finally GROUSE and HFRANS depend on α (controlling the effect of noise) and γ (that can be seen as a re-scaled learning rate). Parameters are tuned to optimize the capability to identify the target subspace.

For the methods adopting a learning rate we tested the classical trend $\gamma_t = c/t^d$ with $d \in \{0, 1/2, 1\}$ and c selected to increase convergence speed. Since, in the fixed subspace case, performances are approximately the same for the three values of d , we selected $d = 0$ and report in Table 3.2 only the value of $\gamma_t = \gamma = c$. Such a constant learning rate allows for substantial updates even when t grows. For HFRANS, the learning rate tuning does not depend on t while the value of α in GROUSE is set to cope with the noise level. Finally, the forgetting factors of ISVD and PAST are selected as the largest possible values that make the fixed subspace case converge.

Results in terms of e_t are shown in Figure 3.2. Figure 3.2a is for the principal subspace while Figure 3.2b is for the minor. Solid lines represent median values of the Montecarlo trials while shaded areas indicate the spread containing 50% of the values.



(A) Principal subspace identification



(B) Minor subspace identification

FIGURE 3.2: Comparison of the convergenc of the Streaming methods for subspace identification.

As expected, all methods can deal with the subspace identification task. As expected, ISVD and PAST are the fastest to converge, while GROUSE is in the order of the two historical approaches Oja and Krasulina. This behaviour reflects that the former are based on the spiked model employed to generate the signals, and the main GROUSE novelty is its ability to manage the missing data case.

3.4 Implementation on edge devices

Since edge devices are employed on various applications and for many different tasks, different solutions are available, ranging from a single-board system equipped with a microcontroller to more complex systems composed of different modules. Two subsections report details of two reference platforms to cope with both scenarios.

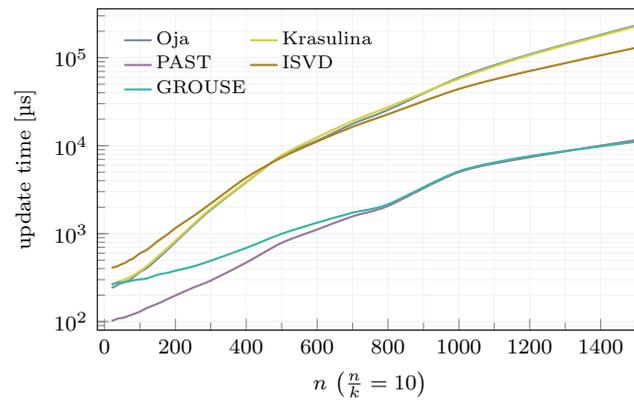
3.4.1 High-end edge devices

Here, we focus on the Raspberry pi 4 model B, which could be considered a reference for the high-end edge devices family. In detail, we refer to the board equipped with 1GB of RAM and Raspberry Pi OS (32-bit) Lite 5.4.51-v7l+ as an operative system. Clock frequency ranges from 600 MHz to 800 MHz¹. Methods are implemented in Python (version 3.7.3), employing the packages Scipy (version 1.1.0) and Numpy (version 1.16.2) for the required algebraic manipulations.

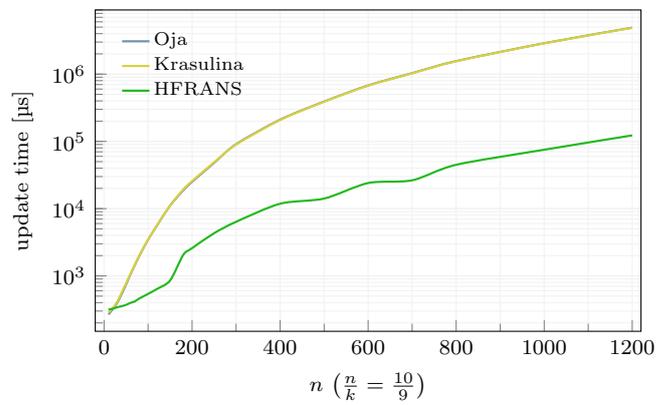
We set $n = 100$ and $m = 10$ (as in Section 3.3) reminding that the target matrix U has dimensions $n \times m$ for PSA and $n \times (n - m)$ for MSA. For each method, update time is reported in Table 3.2 as mean and standard deviation over 1 000 updates. As expected, updates for MSA methods take longer than PSA. For PSA, the fastest method is PAST, while the most time-consuming is ISVD which computes an SVD of an $(k + 1) \times (k + 1)$ matrix at each update. For MSA, the method to be preferred is HFRANS since, like PAST, no complex operation, such as orthonormalization or SVD, is required.

The analysis is enlarged by considering different values for n and k , keeping constant their ratio. Fig. 3.3 shows how the average update time increases with n in case of $n/k = 100/10$ for PSA and $n/k = 100/90$ for MSA. In both cases, the gap between the methods imposing the estimate U_t to be column-orthonormal (Oja, Krasulina, and ISVD) and the other ones (HFRANS, GROUSE, and PAST) increases with n . This trend confirms the difference in the complexity of the methods.

¹The firmware dynamically manages clock frequency depending on the CPU temperature.



(A) Principal subspace identification



(B) Minor subspace identification

FIGURE 3.3: Update time on a high-end device depending on n with a constant n/k .

3.4.2 Low-end edge devices

The implementation of the discussed methods on low resources devices is here presented. In detail, we developed a C library² targeting the ARM Cortex microcontroller (MCU) family of devices. The library back-end is mainly based on the well-known ARM CMSIS-DSP library³.

We implement the methods by adopting specific techniques, described in Section 3.4.2, to minimize memory footprint and processing time. Both GROUSE and PAST implementations adopt all these gimmicks while Oja, Krasulina and ISVD require extra care due to QR decomposition or SVD, as reported hereafter.

Then, performances in terms of update time, energy consumption and memory requirements on the MCU are analyzed, along with an assessment of the error in estimating the target matrix U in both PSA and MSA.

QR decomposition Oja and Krasulina methods employ QR decomposition to implement the $\Omega(\cdot)$ operator in (3.4) and (3.6) that makes the current estimate U_t column-orthonormal. Here we consider three common algorithms to implement the QR decomposition: Householder (Hous-QR) [162], Modified Gram-Schmidt (MGS-QR) [111], and Cholesky (Chol-QR) [176].

They all are iterative algorithms that differ from addressing the trade-off between the orthogonality of the output and computational complexity. Hous-QR provides excellent numerical stability that guarantees a column-orthonormal output even with ill-conditioned matrices, but it is the most computationally expensive. Chol-QR is the other corner case that sacrifices stability in favour of computational complexity. At the same time, MGS-QR is in the middle.

Since Oja and Krasulina methods results to be robust to estimates U_t being only approximately orthonormal, we select Chol-QR for implementing the $\Omega(\cdot)$ operator on MCU as reported in Algorithm 3.1.

Algorithm 3.1 Cholesky-based QR decomposition

- 1: $\text{cholesky}(A^\top A) \rightarrow LL^\top$, L lower triangular matrix
 - 2: $R \leftarrow L^\top$
 - 3: $R^{-1} \leftarrow \text{BS}(R)$ (BS: backward substitution)
 - 4: $Q \leftarrow AR^{-1}$
-

The implementation of Cholesky decomposition is based on the ARM CMSIS-DSP library. In addition, we implement the inversion of the upper triangular matrix R with the backward substitution technique [191]. Since R^{-1} is a lower triangular matrix, we store R and R^{-1} as the two triangular parts of a single square matrix.

SVD SVD is required in ISVD (3.13). We adopt the Golub-Reinsch technique (GR-SVD) [80], whose computational complexity is comparable to Chol-QR, and employ

²online repository https://github.com/SSIGPRO/streaming_pca

³online repository https://github.com/ARM-software/CMSIS_5

Daniel Matterson’s CControl library⁴ for implementing GR-SVD on the MCU. We slightly modify this library to reduce the memory footprint by eliding the Q matrix in (3.13), which is unnecessary for ISVD, and by overwriting the input matrix with the P matrix.

Arithmetic operations The implementation of the streaming methods on the selected MCU is based on the ARM CMSIS-DSP library, which stores any $n \times k$ matrix as a linear vector of size nk to maximize memory contiguity and minimize the number of memory accesses.

Hereafter, we overview the techniques used to improve the efficacy of vector-matrix multiplications.

- **Loop unroll** Since loops are massively employed, loop-unrolling significantly increases the performance by reducing the amount of data transferred and the number of loop index updates. This technique is automatically adopted at compile time by using `-Ofast` gcc option.
- **Register blocking** Matrix-matrix multiplication scales down to multiple subsequent vector-vector dot products performed through a sequence of multiply-and-accumulate operations whose result is an entry of the output matrix. By employing multiple accumulators simultaneously, i.e., by interleaving two or more dot products, local registers are utilised more efficiently, leading to a shorter execution time. Figure 3.4(a) illustrates a graphical representation of this technique.
- **Buffered multiplication** Multiplying a $n \times k$ matrix by a $k \times k$ matrix and its transposition ($k \times k$ times $k \times n$ matrix-matrix) is common, e.g., in Oja, Krasulina, and ISVD methods. Classical implementation requires $(2n + k)k$ operations, which, however, can be almost halved if $n \gg k$ by storing the output matrix in the same memory location of the first input matrix. This overwriting procedure is possible if, for each output row, the input row is temporarily copied in a k -sized buffer and, therefore, it comes at the cost of a slight increase in the computation time. This technique is illustrated in Figure 3.4(b).
- **Vector outer product and matrix addition merging** Some methods (Krasulina, PAST, GROUSE, HFRANS) require operations of the type $A = A + ab^\top$, where $A \in \mathbb{R}^{n \times k}$, $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^k$. Instead of firstly evaluating the outer product $B = ab^\top$ and then performing the sum $A = A + B$, one can directly sum each entry of ab^\top to each entry of A while performing the outer product, thus reducing memory needs and the number of operations. This is shown in Figure 3.4(c).
- **Transposition of square matrices** In general, the transposition of a rectangular matrix requires the copy of the entire matrix in another memory space. In

⁴online repository <https://github.com/DanielMartensson/CControl>

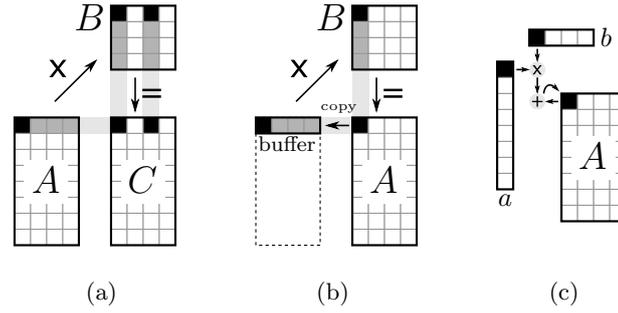


FIGURE 3.4: Matrix arithmetic optimization techniques: (a) register blocking, (b) buffered multiplication, and (c) vector outer product and matrix addition merging.

the case of a square matrix, transposition can overwrite the original matrix by swapping each value in the two triangular parts.

- **Transposed multiplications** The operations AB^T and $A^T B$ can avoid by modifying the multiplication operation and scanning the transposed matrix row-first instead of column-first (or vice-versa).
- **Column concatenation** ISVD method requires the concatenation of a column vector. Columns are not memory-contiguous while rows are. Therefore, transposing the whole method to turn column concatenation into row concatenation reduces memory space and computation time.

3.4.2.1 Performance

All methods are tested on a STM32H743ZIT (rev. V), an MCU based on ARM Cortex M7 family with a 32-bit floating-point unit, $f_{\text{CLK}} = 480 \text{ MHz}$, and both instructions cache and data cache enabled⁵.

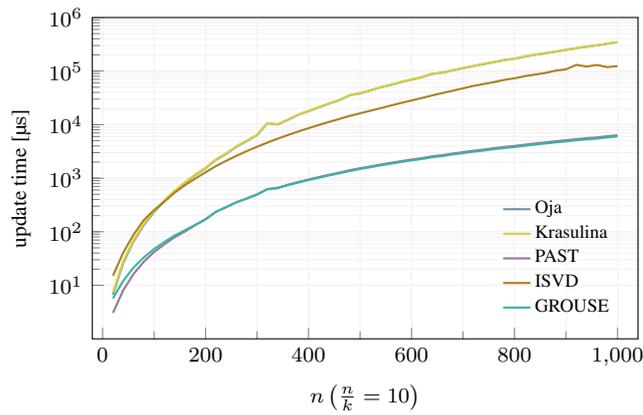
With this setup, the energy consumption of a single update has been estimated as $E_{\text{update}} = V_{\text{DD}} \times I_{\text{DD}} \times t_{\text{update}}$, where V_{DD} is the supply voltage, I_{DD} is the absorbed current and t_{update} is number of clock cycles necessary for a single update divided by the clock frequency. Values are obtained from datasheet. In particular, we refer to current values corresponding to $V_{\text{DD}} = 1.8 \text{ V}$ and with either no peripherals or all the peripherals enabled. Table 3.3 reports time and energy for a single update of each method. Figure 3.5 instead shows how the update time scales with n with a fixed n/k ratio.

For what concerns memory footprint, we can split each method's contribution into three parts: *i*) stack memory, fixed cost negligible compared to the other contributions and independent from either the adopted method or the values of n and m ; *ii*) input vector x and matrix U_t , equal for any method; *iii*) buffers of various sizes necessary for computation. This last contribution is what characterizes the memory

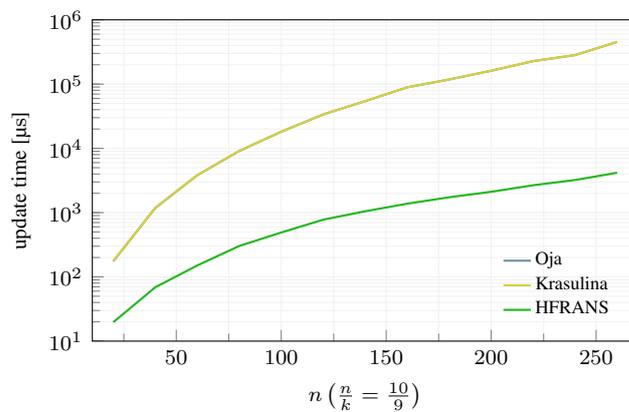
⁵code is compiled with fast target gcc option (-Ofast) in order to maximize the speed performance.

TABLE 3.3: Performance per update on STM32H743ZIT (rev. V) @1.8 V, 480 MHz, cache ON ($n = 100, m = 10$)

Method	Clock cycles [#]	Time [μ s]	Energy (no periph.) [μ J]	Energy (all periph.) [μ J]
Principal subspace				
Oja	110 k	229	45.4	90.7
Krasulina	115 k	239	47.4	94.8
ISVD	136 k	283	56.1	112.2
PAST	20 k	42	8.3	16.7
GROUSE	23 k	49	9.5	18.9
Minor subspace				
Oja	8711 k	18148	3593.4	7186.8
Krasulina	8749 k	18226	3608.8	7217.5
HFRANS	235 k	490	97.0	194.1



(A) principal subspace identification



(B) minor subspace identification

FIGURE 3.5: Maximum update time on MCU against the signal dimension n with fixed compression ratio CR.

TABLE 3.4: Memory requirements rule for each method (memory example with $n = 100$, $m = 10$, 32-bit scalars)

Method	Size of extra buffers	Overall memory rule	Memory footprint
Principal subspace			
Oja	$k \times k$	$nk + k^2 + n$	4.80 kB
Krasulina	$k \times k$	$nk + k^2 + n$	4.80 kB
ISVD	$(k + 1) \times (k + 1)$, $(k + 1), (k + 1)$	$nk + k^2 + n + 4k + 3$	4.97 kB
PAST	$k \times k, n, k$	$nk + k^2 + 2n + k$	5.24 kB
GROUSE	n, k	$nk + 2n + k$	4.84 kB
Minor subspace			
Oja	$k \times k$	$nk + k^2 + n$	17.20 kB
Krasulina	$k \times k$	$nk + k^2 + n$	17.20 kB
HFRANS	n, k	$nk + 2n + k$	9.29 kB

footprint of each method. Table 3.4⁶ reports the size of extra buffers, the size of the overall memory, and the actual memory requirement for $n = 100$ and $m = 10$.

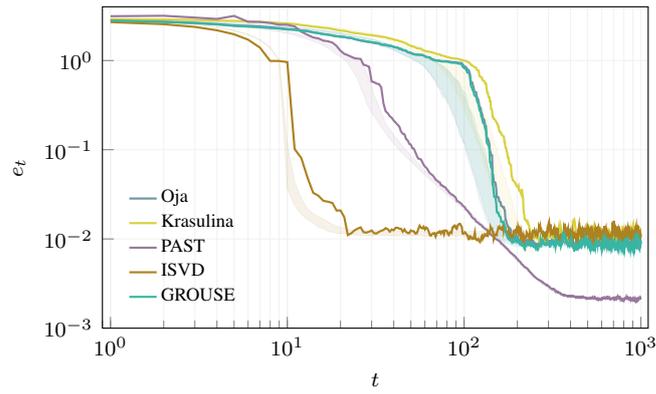
Finally, Figure 3.6 shows the performance in terms of the estimation error of each method for subspace identification implemented on MCU with the same setup reported in Section 3.3. This figure confirms the trends observed in Figure 3.2.

3.5 Conclusion

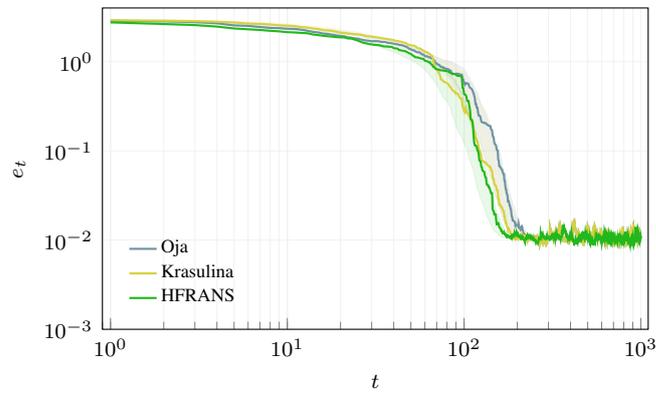
This chapter reviews six major approaches for streaming subspace identification tackling either principal or minor analysis. The aim is to give a coherent presentation of the methods giving rise to most of the subtle variants currently available in the Literature. We also perform comparisons with an eye to deployment at the edge of a system that concentrates data from distributed nodes. Beyond functional testing, we implement the methods in high-end and low-end edge devices to assess their resource requirements and compatibility with nowadays data collection systems.

At least two of the methods reviewed (PAST and HFRANS) excel in principal and minor subspace identification, delivering substantial accuracy with limited computational requirements. If heavier computations are tolerated, faster convergence in PSA is obtained by ISVD.

⁶In Table 3.4 values refer to memory footprint for methods implemented without buffered multiplication technique described in Section 3.4.2. With buffered multiplication, memory requirements for Oja and Krasulina are about 45% higher, while computation is about 19% faster compared to the results shown in this section.



(A) Principal subspace identification ($n/k = 100/10$)



(B) Minor subspace identification ($n/k = 100/90$)

FIGURE 3.6: Estimation error in subspace identification on MCU with a given random Φ and noise $\nu = 10^{-3}$.

Chapter 4

History PCA for SHM systems

Principal Subspace Analysis (PSA) in general and Principal Component Analysis (PCA) play a fundamental role in acquisition systems as they reduce the signal dimension with a lightweight encoding procedure. Indeed, the compression can be performed employing a linear transformation that projects the signal onto the directions that gather most of the signal energy, i.e., most of the signal information.

The linear transformation representing the core of the compression algorithm needs to be tuned on the data to compress. This tuning consists in estimating the principal subspace of the signal, and the straightforward approaches include the storage and processing of a great amount of historical data. Hence, the need for a large amount of storage and computational resources prevents the implementation of the tuning phase on edge devices. This aspect is common to other adapted compression techniques such as the one based on auto-encoders and implies that, in practice, the edge devices can easily compress the signal only if the linear transformation is previously provided.

However, in the case of PCA, there exist algorithms that estimate the principal subspace by processing the data set not as a whole but as a stream, i.e., by considering each signal instance separately and sequentially. They are methods that update the estimate one sample at a time and tend to approach the principal subspace as the number of observations increases. The availability of streaming approaches that allow for a tuning performed locally is another aspect that fuels the appeal of PCA-based compression.

In Chapter 2, we described an application in which PCA is employed to compress the signal and then detect anomalies. In Chapter 3 we review different methods to find the principal subspace with a focus on possible solutions for the implementation on the edge devices. Here in this chapter, we consider the application of a PCA-based compression algorithm to the signal acquired by the Structural Health Monitoring (SHM) system described in Section 2.2, focusing on the adoption of a novel streaming method, namely History PCA [188], for the identification of the principal subspace.

More specifically, here we consider PCA-based compression as a solution to reduce the traffic on the sensor network composing the monitoring system that would allow higher scalability. To perform compression, a first tuning phase is necessary,

and History PCA (HPCA) is one of the streaming solutions that allow the implementation of this phase on an edge device.

HPCA is a streaming PCA method that has been recently proposed and can be viewed as a modification of the traditional Oja's method [145, 146] in the block-wise version [137]. Among the possible streaming PCA methods, HPCA has the advantage of being very robust to the choice of the parameter. In Chapter 3, all revised methods depend on at least one parameter and especially the ones seeking the eigenvectors are particularly sensitive. As it will be shown later, HPCA address this issue by introducing the concept of *history*.

This chapter reports works published in [40, 39] and starts with Section 4.1 that introduce and review the compression algorithms targeting edge and IoT devices in the context of Structural Health Monitoring. Together with the above considerations, we demonstrate that the PCA-based compression is suited for the SHM application. Then, Section 4.2 describes in detail the HPCA algorithm and reports the process to tune the parameter showing that HPCA robustness to parameter variability. After that, Section 4.3 is devoted to the implementation on edge devices. Similarly to Chapter 3 the target devices are both high-end and low-end edge devices. Finally, in Section 4.4 the results are reported. The experimental results consider both functional requirements and performance in terms of time and energy spent running the update on the target devices for the proposed solutions.

4.1 PCA-based Compression in SHM

Large IoT sensor networks managing significant data flows are getting widespread, leading to high demand for methods and architectures able to gather and process large streams of data continuously. When these streams are collected at a central unit to be stored or processed, the communication or the cost for storage space often represents the system bottleneck [100].

This bottleneck can be solved by data reduction proposed in several real-time systems, by either compressing it or distributing part of the processing throughout the network [203, 57]. Many techniques are also proposed to optimize the workload of the node in a network, especially in the deep learning field for the production of smart data [73, 110].

This trend has filtrated to the SHM field. In [86], a system-level co-design between sensors installation and algorithm resolution allows for a reduction of the data gathered, streamed, and stored when demanded. Further moving the processing to the edge, [113] proposes a distributed execution for the eigensystem realization algorithm (ERA), a classical SHM algorithm. The proposed implementation relieves the central unit from the computation and allows to stream only "smart" data, which already contains the diagnosis information. As a drawback, this class of algorithms prevents the system from storing either the raw or an approximation of the recorded data, which can be helpful for additional analysis.

Data compression represents an alternative solution to limit the bandwidth required for communication. Lossless methods ensure no loss of information at the cost of a low compression ratio [160, 125]. For instance, LZMA [83], LZ4, and Deflate [31], widely accepted lossless methods used in file compression algorithms, reach average compression ratios (CRs) lower than 3.

On the other hand, lossy methods, which achieve a higher CR [157], leverage the fact that only part of the entire information contained in the signal is helpful for the analysis. A well-known lossy method is Compressed Sensing (CS), which allows the implementation of energy-efficient encoders [156]. The upside is that with few linear projections and low computational cost, CS captures the primary information contained in the signal, thus being very suitable for SHM applications [93]. The downside is that the energy efficiency comes at the cost of a lower compression ratio (CR) compared to other methods such as wavelet-based, as shown in [117, 41].

For instance, [112] combines wavelet transformation with distributed source coding to increase the compression performance further, reaching a compression factor of 50 with synthetic vibration data. However, this high CR is mainly due to the very high correlation between the generated synthetic data, which are not representative of other different monitoring scenarios. Indeed, in [112], the data are collected from a five-layer civil infrastructure laboratory model, with a distance as small as 15 cm between each layer and a vibration exciter.

A new promising alternative comes from the machine learning field. In [58], it is shown that an autoencoder (AE) can outperform other classical methods such as the ones based on PCA, wavelets, and Discrete Cosine Transform (DCT) with a comparable or lower computational cost. Note that in SHM, auto-encoders have already been successfully employed for temperature and humidity data [3] or embedded in a more complex damage identification system [151]. The main drawback of using autoencoders embedded on edge devices is the need for large training data sets. Moreover, structures often change over time (e.g. due to ageing), forcing the compression algorithms to be re-trained over time.

In this chapter, we focus on PCA-based compression [98], which exploits the correlation between signal components to extract the primary information. The Achievable CR is similar to the wavelet-based method, as shown in [58, 3], but, similarly to CS, PCA-based encoders require few linear projections to compress the raw signal. As a counterpart, similarly to autoencoders, PCA requires a considerable amount of data to estimate the principal components accurately. However, differently from autoencoder, PCA may be trained in streaming fashion by considering a sample (or small blocks of samples) at a time [106, 145, 137, 88, 6, 180, 36, 17, 188]. The simplicity at the encoder side and the possibility to conduct the training phase in a streaming fashion make PCA a method suited to be embedded on resources constrained devices typically involved in IoT networks and SHM systems.

To validate the effectiveness of the PCA-based compression method in an SHM application based on vibration sensing, we compared its performance with some

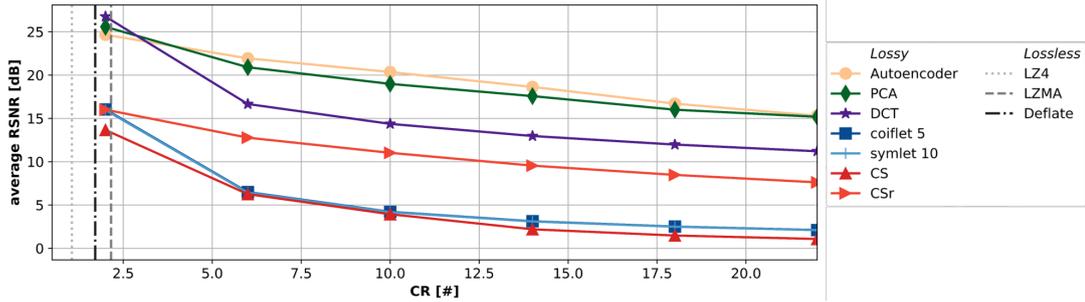


FIGURE 4.1: Comparison of lossy and lossless methods applied to signals from a vibration-based SHM system.

of the algorithms previously presented. The assessment is based on our use case dataset, and Figure 4.1 shows a figure of merits regarding the quality of reconstruction of the signal (Reconstruction Signal to Noise Ratio – RSNR) depending on the compression ratio (CR). We selected LZ4 [83], LZMA and Deflate methods [31] as lossless methods, while for lossy approaches we considered compression based on Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT) [4], Compressed Sensing (CS) [61] and autoencoder [3]. For the DWT implementation, we consider Symlet, Coiflet, Daubechies, and Haar families, but in Figure 4.1, we only show Symlet 10 and Coiflet 5 that outperform the others. For the Compressed Sensing approach, along with the standard method [61], we also consider the rakesness method that adapts the encoder stage to the class of signals [118].

The average RSNR obtained with the PCA-based method is comparable with the performance of the autoencoder while outperforming all the other methods for $CR > 4$. On the other hand, the PCA has a twofold advantage compared to the autoencoder: it requires less data for the training, and the streaming approaches allow to avoid storage of the complete training dataset, which would be prohibitive on an end-node device.

4.2 History PCA

As already extensively discussed, Principal Component Analysis PCA methods aim to estimate the principal subspace of the monitored signal in the form of a set of the eigenvectors corresponding to the k most significant eigenvalues of the signal correlation matrix. A streaming PCA method performs this task by processing input data sequentially so that at each incoming input instance, the estimate is updated with no need to store the entire data set.

Among them, History PCA (HPCA) [188] has recently emerged. HPCA is based on the block-stochastic power method [137], which in turn may be viewed as a modification of the traditional Oja’s method [145]. Compared to the other methods, HPCA has been developed to improve the principal subspace estimation accuracy by using a compact representation of the historical data, from which it takes the name. This representation of the history includes both the eigenvectors and the

corresponding eigenvalues. Since most streaming PCA methods only focus on estimating the eigenvector, HPCA is an alternative when knowledge about eigenvalues is required. Another feature of HPCA is the ability to work with blocks of data generalizing the typical streaming processing to consider one sample at a time. This generalization allows for a more efficient implementation in terms of execution time as it allows the device to run the algorithm at each block instead of each instance.

The HPCA algorithm is proposed in [188] and here is reported in Algorithm 4.1. Let us consider a data block $X_t \in \mathbb{R}^{b \times n}$ at time instant t where b is the number of instances composing a data block, and n is the dimension of the input signal. The algorithm runs for each incoming data block and updates the eigenvectors' and eigenvalues' estimates. The former are stored as columns of U_t , and the latter are arranged on the diagonal of the matrix Λ_t . With N the number of signal instances composing the dataset, after $B = N/b$ runs, the algorithm returns the final estimate U_B and, optionally, Λ_B .

Algorithm 4.1 History PCA

```

1: Input:  $X_1, \dots, X_B$ , block-size:  $b$ .
2:  $S_0 = \begin{bmatrix} S_0^{(1)} & \dots & S_0^{(k)} \end{bmatrix}$  with  $S_0^{(j)} \sim \mathcal{N}(0_n, I_n)$ 
3:  $U_1 \leftarrow \Omega(S_0)$ 
4: for  $i \leftarrow 1, \dots, M$  do
5:    $S_1 \leftarrow U_1 + \frac{1}{b} X_1^\top X_1 U_1$ 
6:    $U_1 \leftarrow \Omega(S_1)$ 
7: end for
8:  $\lambda_j \leftarrow \left\| S_1^{(j)} \right\|_2$  for  $j = 1, \dots, k$ 
9:  $\Lambda_1 \leftarrow \text{diag}(\lambda_1, \dots, \lambda_k)$ 
10: for  $t \leftarrow 2, \dots, B$  do
11:    $U_t \leftarrow U_{t-1}$ 
12:   for  $i \leftarrow 1, \dots, M$  do
13:      $S_t \leftarrow \frac{t-1}{t} U_{t-1} \Lambda_{t-1} U_{t-1}^\top U_t + \frac{1}{t} \frac{1}{b} X_t^\top X_t U_t$ 
14:      $U_t \leftarrow \Omega(S_t)$ 
15:   end for
16:    $\lambda_j \leftarrow \left\| S_t^{(j)} \right\|_2$  for  $j = 1, \dots, k$ 
17:    $\Lambda_t \leftarrow \text{diag}(\lambda_1, \dots, \lambda_k)$ 
18: end for
19: Output:  $U_B$ 

```

HPCA algorithm consists of a first initialization part that employs a first data block X_1 to build the first historical representation, followed by the algorithm's core that refines the estimation at each new incoming data block X_j with $j = 2, \dots, B$.

The core is described in lines 13–14 and somehow resembles the Power Iteration [81, Chapter 7], which multiplies a diagonalizable matrix by its current eigenvectors' estimate and orthogonalizes the result to obtain a new estimate. One may recognize two diagonalizable $n \times n$ matrices multiplied by the current eigenvector estimate U_t in line 13 and the orthogonalization in line 14. The first of the two diagonalizable $n \times n$ matrices is $U_{t-1} \Lambda_{t-1} U_{t-1}^\top$ which is a rank- k approximation of the correlation

matrix of the data already processed and embodies the history. The second diagonalizable $n \times n$ matrix is $X_t^\top X_t/b$ which is an estimate of the signal correlation matrix by means of the current data block X_t . The two contributions are weighted and combined into the matrix S_t so that history is weighted proportionally to the number of processed blocks.

The core of the algorithm (i.e., lines 13–14) is iterated for a fixed number of times M (or until convergence). Then, since the matrix S_t contains the current estimate of the eigenvectors scaled by the corresponding eigenvalues, the norm of the j -th column of S_t is a current estimate of the j -th eigenvalues λ_j . This procedure is repeated for each of the B blocks composing the data set.

The original HPCA algorithm starts from an initial matrix U_1 that comes from a random initialization (see lines 2–3). However, in other streaming PCA methods, it is possible to speed up the convergence by starting with a warm start [6, 175], and a possible warm start for HPCA consists of a matrix U_1 obtained as the left-singular vector from the rank- k thin SVD computed on an initial data block.

In HPCA, computational complexity is dominated by the matrix multiplications $\mathcal{O}(nk(k+b))$ and by the QR decomposition $\mathcal{O}(nk^2)$ while memory footprint is in the order of $(3nk + k^2 + nb)$ where nb accounts for the input block X_t , $3nk$ and k^2 for the intermediate data to compute the U_t matrix.

Computational complexity and memory footprint are consistent with the requirements of other streaming PCA algorithms that retrieve the eigenvalues other than the eigenvectors. In addition, HPCA is robust to parameter tuning. Indeed, it has no learning rate and performance variability due to the choice of the number of iterations of the internal loop M and block-size b is very limited, as discussed later.

Similarly to the analysis in Chapter 2, here we consider as input signal instances the n -dimensional time-windows of the x -axis acceleration recorded by a sensor node in the SHM system described in Section 2.2.

Note that, in a real application like the one here, data instances come at a constant rate f_s/n , the block-size parameter b plays an important role. Increasing b , the memory requirement slightly increases, but the algorithm is to be run less frequently (i.e., every nb/f_s seconds).

4.2.1 Parameter Tuning

For effective deployment of the HPCA method, the involved parameters need to be tuned on the specific application. We consider the x -axis acceleration recorded by a sensor node in two different weeks for this task. The week is chosen to account for the daily periodicity and the different traffic conditions of the weekdays.

The recording of one of the two weeks is employed to estimate the matrix U through the HPCA method, while the data from the other week is employed to assess the quality of service (QoS), measured in terms of Reconstruction Signal to

Noise Ratio (RSNR), defined as:

$$\text{RSNR} = 20 \log_{10} \left(\frac{\|x\|_2}{\|x - \hat{x}\|_2} \right) \quad (4.1)$$

where x indicates a generic signal instance and $\hat{x} = UU^T x$ is the correspondent recovered vector following PCA-based compression.

In PCA-based compression, there are two intrinsic parameters: n , which is the dimension of the signal, and k , that is the dimension of the compressed version of the signal, i.e., the number of the considered principal components.

In general, for signals coming from physical phenomena, the larger the time window, the more redundancy in each window can be exploited to increase the compression ratio $\text{CR} = n/k$ given a minimum RSNR. As a consequence, the parameter n is bounded from above by the maximum processing delay that, in this particular application, consists of $t_W = 5$ s. This constraint leads to a windows length $n = 500$ and $N \sim 120\,000$ signal instances per week.

The parameter k affects the compression level and reconstruction quality. Consequently, k is determined by the minimum value of RSNR that is retained acceptable for the application. In this specific case, the application consists in detecting the frequency, and the amplitude of the peaks in the signal spectrum and a preliminary analysis showed that $\text{RSNR}_{\min} = 16$ dB is sufficient, as it is shown in Section 4.4. PCA-based compression (with U matrix estimated with traditional batch PCA) meets the requirement ($\text{RSNR} = 16.13$ dB) with $k = 32$ principal components, i.e., with a $\text{CR} = 15.6$.

HPCA relies on two additional parameters: the block-size b , and the number of iterations in the internal loop M . Figure 4.2 shows the average RSNR when M varies for different values of b and number of principal components k . As expected, the higher k , the higher the average RSNR and all trends saturate for $M \geq 3$ independently from the adopted block size b . This behaviour demonstrates the robustness of HPCA to variation in the parameters. Note that the parameters M and b are strictly related.

With $k = 32$ (same of the PCA), $M = 3$ and $b = 1$ the average RSNR measured on a validation set is 16.11 dB, i.e., just 0.02 dB below the value reached by batch PCA and still above RSNR_{\min} .

4.3 HPCA Implementation

As other streaming PCA algorithms, HPCA fits devices with scarce computational resources. In this specific SHM application, target devices may be the gateways or the sensor nodes. Since PCA is applied independently to each acceleration axis of each sensor node of the installation, i.e., HPCA must be run a number of times equal to the number of sensors (45 for each gateway, hence 90 in total) times the number of acceleration axes to be monitored ($n_{\text{ax}} = 3$ axes).

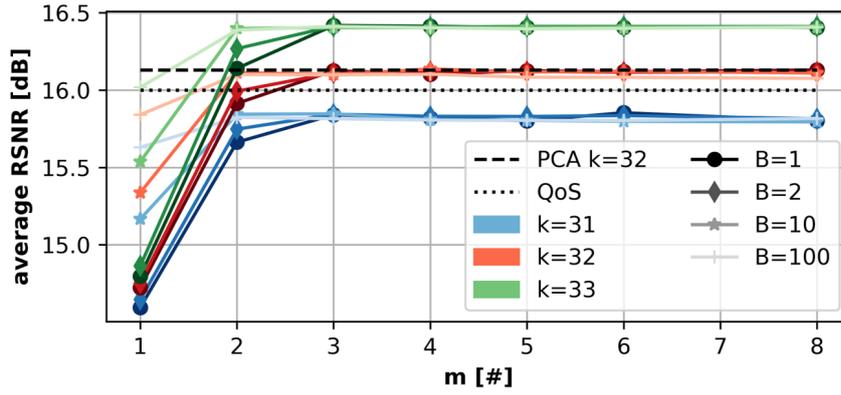


FIGURE 4.2: HPCA performance in terms of RSNR depending on the number of internal loops m for different block-size b and number of principal components k .

TABLE 4.1: Specs of the target gateway devices.

device	#core	CPU		Memory			Cache L2
		type	freq GHz	size MB	type	freq GHz	size kB
RPi3	4	Cortex A53	1.2	1000	DDR2	900	
ARTIK 710	8	Cortex A53	1.4	512	DDR3	800	256

As a result, when gateways are the target devices, they must have enough memory to run $n_s n_{ax}$ HPCA algorithms (with $n_s = 45$ sensors) and the execution time of the latter must be shorter than the period between two signal instances $t_W = n/f_s = 5$ s. Alternatively, if the HPCA implementation is distributed to the sensor nodes, requirements relax to n_{ax} HPCA execution. Both memory and time constraints are addressed in this Section.

4.3.1 Implementation on gateway

In the current installation, the IoT gateways consist of a Raspberry Pi 3 module B [71] (RPi3), but the analysis in this dissertation also includes the Samsung ARTIK 710 Module [65] as an alternative. The former is a single-board computer actively used in many fields, such as robotics, smart sensor control, and structural health monitoring, while the latter is a System-in-Module developed by Samsung that targets high-end gateways with the capabilities of local processing and analytics. Specifications of both devices are reported in Table 4.1.

Two implementation versions are proposed and analyzed. Both versions are implemented in Python 3.5 and exploit NumPy library [99] which relies on optimized BLAS [30] and LAPACK [9] libraries for linear algebra computation.

The first version (GT1) runs each HPCA algorithm sequentially and exploits parallelization to optimize the linear algebra operations. Consequently, all cores are employed to execute a single HPCA algorithm for a single trace. This choice leads to a minimal memory footprint, but computational time does not scale with the number

of core n_C since some operations (e.g. QR decomposition) can not be fully parallelized.

The latter (GT2) runs a different HPCA algorithm on each available core, allowing to process different traces simultaneously. As a result, GT2 achieves a near-ideal speed-up (which is only limited by the simultaneous accesses to the memory) but requires a n_C times higher memory space.

4.3.2 Implementation on sensor node

Sensor nodes are based on the STMicroelectronics STM32F405RG microcontroller unit (MCU). Due to the relatively high operating frequency and the floating-point unit (FPU) with a full set of DSP instructions, the STM32F405xx MCU family is prevalent for embedded computing. STM32F405RG MCU is equipped with an ARM 32 bit Cortex-M4 CPU @168 MHz with FPU, 192 kB of SRAM, and 1 MB Flash memory.

A first implementation (SNS1) consists of the straightforward porting of HPCA in C code, which, however, does not fit the constrained memory resources of the MCU, so it requires a reduction of the signal dimension n .

A second version (SNS2) is proposed to solve this memory issue. It employs 16 bit fixed point arithmetics and exploits CMSIS-DSP software library [55] that lead to a reduction in execution time and a halving of the memory footprint. As a drawback, SNS2 requires a quantization of the quantities involved, resulting in a loss of accuracy that must be carefully addressed. In particular, the critical points are the matrix multiplications (line 13) and the orthogonalization (line 14)¹.

In matrix multiplications, the loss in accuracy has been addressed by increasing the integer part of the fixed-point representation from 1 bit to 5 bit. This solution allows to capture the high dynamic range of the input signal² and the value has been determined after a grid search on a validation set composed by 4 h recording.

Orthogonalization is implemented by mean of QR decomposition with Householder reduction, which, due to better numerical stability, has been preferred to Cholesky decomposition, Gram Schmidt [74]. Following implementation in [74], the computation of the reflection vectors has been performed with an 4 bit for the integer part, while 2 bit integer part is employed for the actual Householder reflections. All value for the integer parts of the fixed-point representations is determined after a grid search on a validation set composed by 4 h recording.

The performance loss due to fixed-point arithmetic is addressed by increasing k , the dimension of the compressed signal, from 32 to 50. This leads to $\text{RSNR} = 16.02$ dB that satisfy the constraint $\text{RSNR} > \text{RSNR}_{\min}$.

¹8 bit fixed-point is not taken into account as the approximation error is such that the HPCA algorithm fails in reaching the minimum acceptable RSNR.

²input signal is normalized with mean and standard deviation estimate estimated on a training set.

4.4 Experimental Results

This section first shows that the adoption of HPCA as a streaming method to estimate the matrix U employed for PCA-based compression has negligible influence on peak detection performance in the spectrum of an acceleration signal for SHM. Then, the performance in terms of execution time and memory footprint are assessed for the different target devices and different versions of HPCA implementation.

4.4.1 Functional Performance

When subjected to external forces, structures tend to vibrate at their natural frequencies [50]. Even a subtle variation in how the structure vibrates can be a symptom of structural damage or deterioration. Here we monitor natural frequency by tracking the highest peaks in the acceleration spectrum. The target SHM system monitors the vibration of the pre-stressed tendons that feature slightly different behaviours since they have different lengths and tension. That is why traces from different sensors are processed independently from one to another.

The Power Spectral Density (PSD) is estimated by averaging the periodograms over 18 non-overlapping Hanning windows of 200 s each. Hence, every hour, an estimate is produced with a frequency resolution of 5 mHz, which is needed to detect relatively low variations of peak frequencies. The resulting profiles are smoothed by a Savitzky-Golay filter (length 11, degree 3) and processed by a peak-picking method. The peak-picking method extracts the highest 15 local maxima that have prominence³ lower than half of their height. Finally, peaks with a maximum distance of 0.2 Hz and belonging to successive time frames are grouped.

Figure 4.3 shows an example of spectrum estimation from original and reconstructed signals of the x -axis vibration signal from one of the viaduct tendons. Both the floating-point and 16 bit fixed-point implementation of HPCA are considered and compared to the classical PCA method with the set of parameters that guarantees the target RSNR. All four spectra share the same profile in the region near the peaks, but they significantly differ in the other bands. Both PCA and floating-point HPCA show a filtering effect, while the profile of fixed point HPCA remains at the same level as the original one but with a more noisy trend.

The performance of the peak-tracking algorithm is tested on an additional 1-week test set following the two 1-week periods used as training and validation sets. The figure of merit used to validate the approach is the error in the peaks detection made on reconstructed signals compared to the case where the original signal is considered. The error regards both frequency (err_f) and amplitude (err_A) of the peaks

³In topography, prominence is a measure of the independence of a peak and is computed as the height of a peak relative to the lowest contour line encircling it but containing no higher peaks.

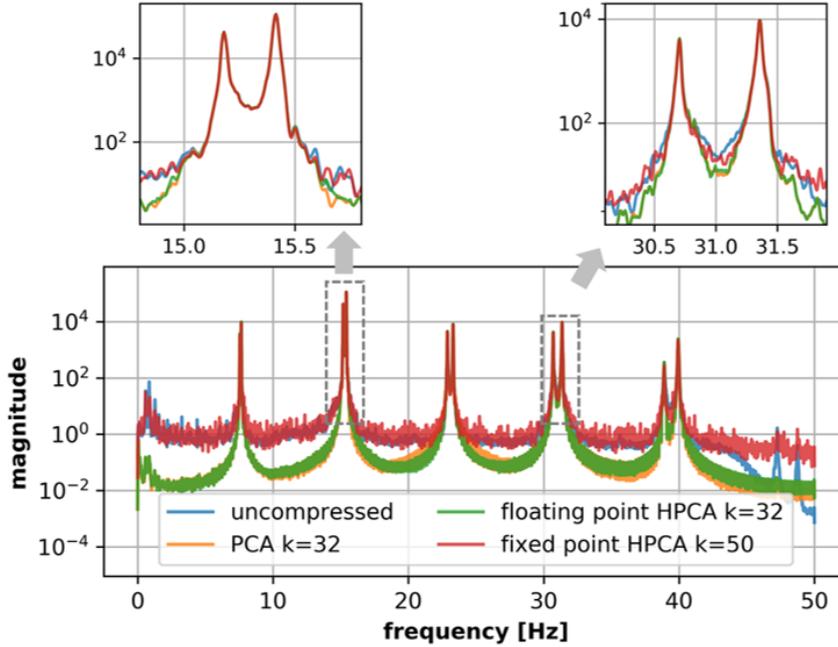


FIGURE 4.3: Power Spectrum Density of the x -axis acceleration signal of one of the sensors in the viaduct SHM system.

TABLE 4.2: Peak detection performance over 1 week in terms of estimation error for frequency (err_f) and amplitude (err_A).

	frequency [mHz]			amplitude [dB]		
	mean	std	max	mean	std	max
PCA	0.21	1.17	20	0.03	0.07	0.71
floating-point HPCA	0.21	1.18	20	0.03	0.06	0.80
fixed-point HPCA	0.41	1.78	25	0.52	0.50	2.27

and it is computed in terms of difference between the uncompressed and reconstructed cases: $\text{err}_f = |f^{\text{ref}} - f|$, $\text{err}_A = |A_{dB}^{\text{ref}} - A_{dB}|$, where the pair $(f^{\text{ref}}, A_{dB}^{\text{ref}})$ represents frequency and amplitude in the uncompressed case, and (f, A_{dB}) the peak characteristic in the case of interest.

Table 4.2 summarizes the results in terms of mean, standard deviation (std), and max value observed over the ten peaks and the whole test set. All investigated approaches obtain a low mean error for both frequency and amplitude. It is worth noting that the max err_f recorded in the test set is just 4-5 times the PSD frequency resolution which is lower than the resolution needed to detect physically meaningful frequency shifts.

To better appreciate the performance, Figure 4.4 depicts the frequency and amplitude profiles of the peak at frequency $f_{\text{ref}} \sim 39.9$ Hz over the 1-week test set period for the different approaches along with the no compression case. Despite being the peak with the highest err_f and err_A , the difference between the curves is negligible.

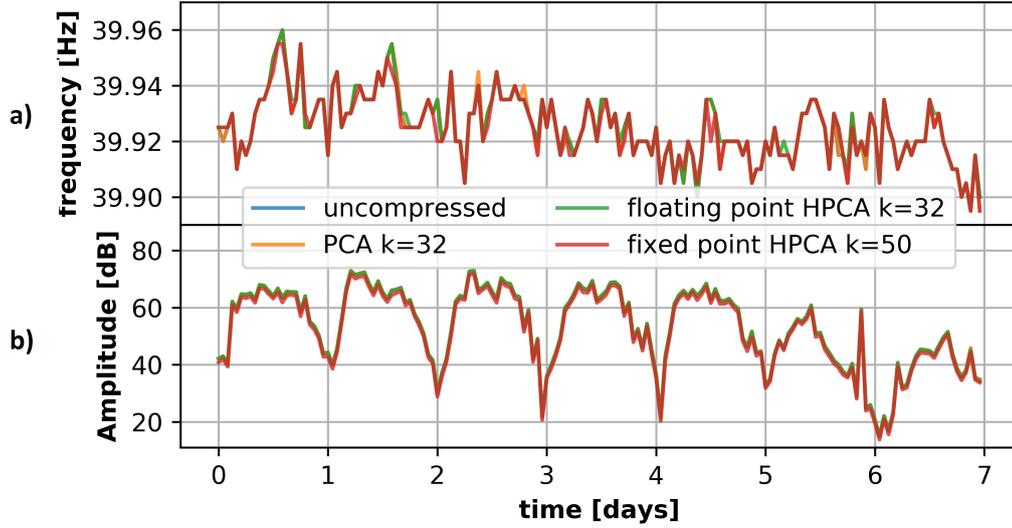


FIGURE 4.4: Peak tracking of the peak at frequency $f_{\text{ref}} \sim 39.9$ Hz over an 1 week period applied to the original signal and reconstructed with different compression matrix: PCA, floating point HPCA and fixed point HPCA. Panels (a) and (b) show the tracking of frequency and amplitude, respectively.

Damage detection The algorithm is also tested to assess the capability of identifying damages in the viaduct structure. In particular, during the monitoring period, one of the tendons in the viaduct experimented with a strand breakage. The vibrations natural frequencies are tracked before and after this event. Figure 4.5 reports the tracking of the three main modes. In detail, Figure 4.5(A) depicts the variation computed with the usage of the original signal, while Figure 4.5(B) consider the reconstructed signal with floating-point HPCA with $k = 32$. Noteworthy, both plots show a $\sim 2.0\%$ down-shift for all involved natural frequencies with a negligible loss of average shift accuracy (less than 0.1%).

4.4.2 Execution time and energy consumption

This Section considers the different target devices and the different implementations in executing an HPCA update. GT1 and GT2 are the algorithm versions targeting RPi3 and ARTIK 710 as gateway devices, while SNS1 and SNS2 are the versions for the sensor nodes equipped with STM32F405RG as MCU. Comparisons regard energy consumption and execution time. In the case of parallelization, the execution time t_{ex} refers to the inverse of the throughput that is the number of results produced per unit of time.

Firstly, we compare the gateway implementations GT1 and GT2 on RPi3, which is currently employed in the SHM system. Then, the best implementation is employed to compare RPi3 with ARTIK 710 Module. Finally, the sensor node implementations SNS1 and SNS2 are compared.

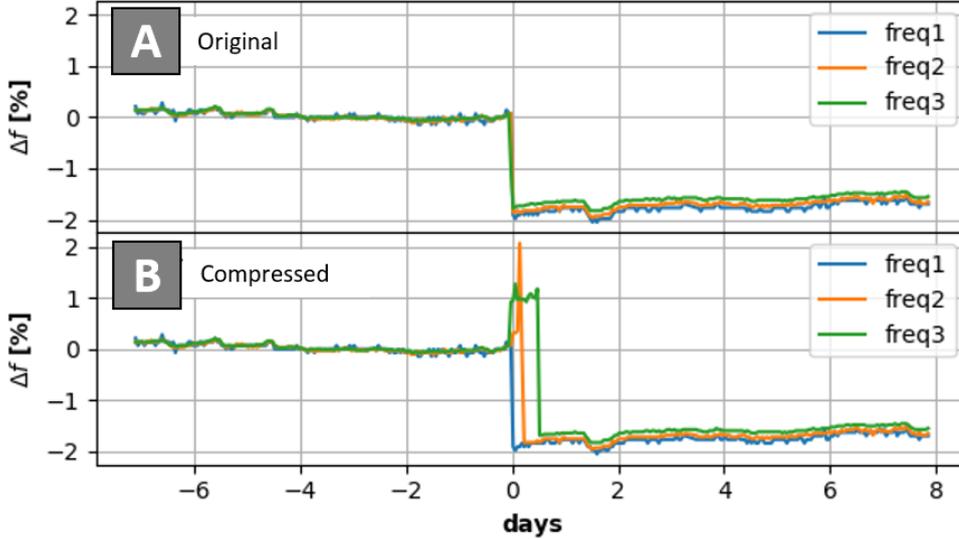


FIGURE 4.5: Tracking of the three main natural frequencies of a tendon of in the week before and after a tendon strand break. The tracking is applied on the PSD computed from the original signal (A), and the reconstructed signal using HPCA (B).

TABLE 4.3: Comparison of GT1 and GT2 implementations in terms of execution time and energy consumption on Raspberry Pi 3 with $n = 500, b = 1, k = 32, M = 3$.

# core	GT1		GT2	
	time [ms]	energy [mJ]	time [ms]	energy [mJ]
1	44.6	18.8	44.6	18.8
2	38.4 (1.2×)	26.9 (+43%)	24.1 (1.9×)	16.8 (-11%)
4	35.3 (1.3×)	44.0 (+134%)	13.3 (3.4×)	16.5 (-12%)

GT1 vs GT2 For GT1, since acceleration traces are processed sequentially, t_{ex} is equivalent to the latency of a single HPCA instance. Conversely, in GT2, t_{ex} is computed as the time needed to run n_C HPCA instances and dividing the total time by n_C . The experiments are conducted considering the set of parameters obtained as result of the tuning procedure (Section 4.2.1), i.e., $n = 500, k = 32, b = 1, M = 3$ and the results are reported in Table 4.3.

Parallelization at sensor level (GT2) achieves better performance than parallelization of the internal linear algebra operations (GT1) in terms of execution time and memory consumption. That is why GT2 is later employed to compare RPi3 with ARTIK 710 Module. Note that, parallelization on RPi3 is necessary since the single core implementation of HPCA would lead to $t_{\text{CPU}} = n_S n_{\text{ax}} t_{\text{ex}} = 6.02 \text{ s}$ which does not satisfy the constraint $t_{\text{CPU}} < t_W = 5 \text{ s}$.

TABLE 4.4: Comparison of RPi3 with ARTIK 710 in terms of execution time [ms] of HPCA with GT2 implementation and $n = 500$, $b = 1$, $k = 32$, $M = 3$.

# core	1	2	4	8
Samsung ARTIK 710				
total	34.7	17.7 (2.0×)	9.0 (3.9×)	4.9 (7.1×)
MM	11.1	5.7 (1.9×)	3.0 (3.7×)	1.6 (6.9×)
QR	22.6	11.5 (2.0×)	5.9 (3.9×)	3.1 (7.2×)
Raspberry Pi 3 Model B				
total	44.6	24.1 (1.9×)	13.3 (3.4×)	
MM	16.2	9.3 (1.8×)	5.1 (3.2×)	
QR	27.0	14.1 (1.9×)	7.8 (3.5×)	

Rpi3 vs ARTIK 710 Here RPi3 is compared with its alternative ARTIK 710. Table 4.4 portrays a detailed comparison of the HPCA execution with the GT2 implementation on the two platforms. Single-core implementations have comparable execution time, while parallelization on all available cores let ARTIK 710 run $2.7\times$ faster. This result is due to the higher number of cores and a more efficient parallelization provided by the faster DDR3. Note that, considering their most performing HPCA implementation, both ARTIK 710 and RPi3 platforms allows a single gateway to process all 90 sensors of the installation. Therefore, the system bottleneck remains the sensors-gateway communication that limits to ~ 50 the number of sensors per gateway.

The relation between energy and the size of the input block b of the HPCA iteration is also evaluated, and Figure 4.6 shows the trade-off between memory occupancy and energy consumption. Increasing b reduces the number of processed blocks, and consequently, the number of HPCA iterations for the estimation of U . For instance, by increasing b from 1 to 50, the HPCA necessitates $1.5\times$ memory (from 198 kB to 296 kB) and saves almost $50\times$ energy consumption on both the platforms.

Figure 4.6 also shows that ARTIK 710 consumes more energy than RPi3 when HPCA runs on a single core due to a more power-hungry memory. However, the more efficient parallelization allows ARTIK 710 to drastically reduce the energy, while RPi3 cannot benefit from a slower DDR2.

SNS1 vs. SNS2 As previously anticipated, the SNS1 version of the HPCA algorithm does not fit the memory of the STM32F405RG microcontroller unit. SNS2 cannot halves the memory footprint since hte accuracy loss due to the fixed-point implementation is balanced with reducing the compression ratio CR. However, it reduces the memory requirement from 200 kB to 156 kB.

For a fair comparison between the two implementations, we decrease n to 100 while keeping $CR = 15$ for SNS1 and $CR = 10$ for SNS2. This setting does not

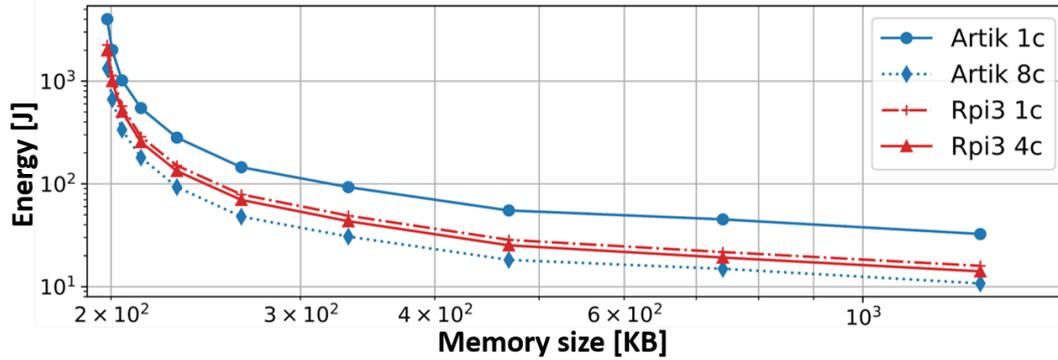


FIGURE 4.6: Trade-off between energy consumption and memory occupation after a full pass over the training for both RPi3 and ARTIK 710 with GT2 implementation of the HPCA algorithm and $n = 500, k = 32, M = 3$.

TABLE 4.5: Comparison between SNS1 and SNS2 implementation of HPCA algorithm on STM32F405RG with $n = 100, b = 1, M = 3$.

	CR	k	n_{\max}	time [ms]	energy [mJ]
SNS1	15	7	450	40.4	11.6
SNS2	10	10	500	18.0	5.2

allow to reach the RSNR_{\min} but allow for a comparison in terms of time execution and energy consumption and highlights the benefit of adopting SNS2.

As expected, the SNS2 runs $2.2\times$ faster and consumes 55% lower energy when compared to the SNS1 floating-point implementation as shown in Table 4.5. Hence, the fixed-point implementation makes the HPCA runnable on the uC and outperforms SNS1 in terms of execution time and energy consumption. As a counterpart, the SNS2 reaches a lower compression ratio that could be preferable in an MCU with a larger memory.

4.5 Conclusion

Principal Component Analysis can be an effective tool to reduce the dimension of the signal in monitoring systems which may benefit from compressing the data close to where it is produced, i.e., on edge devices. The advantages consist of both the lightweight encoding procedure and the possibility of performing the tuning locally.

In this chapter, we analyze the effectiveness of History PCA algorithm for PCA-based compression in a Structural Health Monitoring application. First, we compare PCA-based compression with other lossy and lossless compression methods in the SHM context. The comparison shows that PCA-based compression is a valid alternative that is just slightly less performing with respect to the autoencoder, which cannot be locally trained.

Then, following a discussion about the features characterizing the HPCA algorithm, we tuned its parameters on the SHM data set, demonstrating that this method

is robust to parameter variability. We show that it is sufficient to choose a sufficiently large number of internal iterations to be almost independent of the block size.

In this dissertation, we propose four different implementations for HPCA. Two are designed for the high-end edge, and two for low-end edge devices such as microcontroller-based sensor nodes. The high-end implementations differ from how parallelization is exploited. One solution distributes the linear algebra computation over the multiple cores available on the two considered edge devices. The other option takes advantage that more than one HPCA instance must be run at once, assigning one to each core. This latter solution has a higher memory footprint but permits a more effective parallelization resulting in a shorter computation time and less energy consumption.

The other two implementations targeting the low-end device differ from the arithmetic employed to run HPCA. One is the straightforward translation of the HPCA in code that requires the usage of the Floating point unit. This solution, however, does not fit the device memory and requires a reduction of the window length. To address this issue, a 16 bit fixed-point solution is proposed. The adoption of integer arithmetic and the reduction from 32 bit to 16 bit for quantities representation leads to a further loss in the quality of compression that needs to be balanced with an increase in the number of principal components employed.

All implementations are tested on the SHM use case. First, a functional assessment is performed to show that signal compression has a negligible effect on modal analysis and on the capability to detect anomalies. Then we perform an exhaustive and quantitative comparison in terms of execution time and energy consumption between the different implementations and devices. In particular, we show that deploying HPCA to the sensor nodes leads to higher scalability that would allow gateways to manage a higher number of sensors. However, only the fixed-point version of HPCA fits sensor devices, which degrades the quality of compression that can only be balanced by decreasing the compression ratio. Hence, the trade-off between scalability and compression ratio needs to be carefully addressed.

Part III

Compressed Sensing Decoder

Chapter 5

Trained Support Oracle for Compressed Signals

Compressed Sensing (CS) is a relatively new paradigm for the acquisition of signals violating the intuition behind the theorem of Shannon CS [61, 45, 22], i.e., CS affirms that the minimum number of digital words to be used to represent a signal instance is less than the number of samples acquired with a sampling rate double the signal band.

CS theory requires two surprisingly general conditions: *sparsity*, which is related to the signals of interest, and *incoherence*, which relates to the mechanism behind the acquisition/compression process where the outputs of this encoding phase are usually named measurements. Sparsity is a requirement that is associated with the idea that many natural signals have a very parsimonious representation when expressed in an appropriate *sparsity basis*. Incoherence says that it is possible to limit the number of acquisitions of a waveform that have a sparse representation if the sampling process projects the input signal on an appropriate basis, which may be incoherent with the one characterizing the sparse representation such that each new measurement can capture a non-null part of the information content. As a result, the two identified domains are that the signal is always sparse in the former and dense in the latter.

Based on these concepts, it is possible to devise an innovative protocol for sampling/compression [21, 7] able to capture the information content and produce measurements in a number connected to the number of non-zero coefficients in the sparse representation of the signal of interest. Consequently, if the sampling procedure is directly performed in the analog domain, the most significant feature consists in allowing a sensor to capture the information content of a signal without going through the acquisition of its entire profile, thus performing acquisition and compression at the same time.

In other words, CS features an encoder block that can sample sparse signals at a reduced rate with a very simple and efficient procedure. Therefore, CS employs fewer resources than standard sampling paradigms required for Analog to Digital conversion.

Sparsity characterizes many signals of interest. For example, many signals enjoy

the sparsity property in biomedical applications and can be efficiently acquired using CS, i.e., less energy, less time, or fewer samples. Possible application domains are Electrocardiographic (ECG), Electromyographic (EMG) [60] and Electroencephalographic (EEG) [202] signals. In particular, CS paves the way for the redesign of smart wireless sensor nodes [75, 150]. Another class of signals that is well suitable for the CS framework is the one containing waveforms acquired through magnetic resonance imaging (MRI) [114], for which CS permits the acceleration of the overall MRI acquisition [92].

All these advantages in the encoder design are balanced by a more complex decoder, especially when compared to the simple low-pass filtering in a standard D/A conversion. Indeed, the CS decoder inverts the acquisition/compression phase, a fundamental problem in several heterogeneous applications. In detail, the problem is recovering an n -dimensional sparse representation of the signal x from a set of m measurements y representing the CS encoder's output.

More specifically, sparsity means that a proper $S \in \mathbb{R}^{n \times m}$ sparsity basis exists such that each possible vector x when represented on S , i.e., $x = S\zeta$, is associated to a vector of coefficient ζ with only $\kappa \ll n$ non-null elements. The encoder $\text{Enc} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a linear dimensionality-reduction operator ($m < n$) and produces $y = \text{Enc}(x)$. Signal recovery is possible by finding the sparsest n -dimensional vector ζ among the infinite solutions of the hill-defined system $y = \text{Enc}(S\zeta)$, which, regrettably, is an NP-hard problem.

However, CS theory [46] says that the solution can be obtained by solving a minimization problem, called *Basis Pursuit* (BP)¹, using linear programming. In other words, the result in [46] is a key enabling point since it guarantees that the BP problem can be solved in polynomial time, thus making the use of CS practical. Moreover, the computational resources needed by the numerical algorithm solving BP may be so demanding to make its solution practically unfeasible in low-complexity nodes, like a typical gateway. To cope with this, several dedicated BP/BPDN solvers have been proposed, such as the Spectral Projected Gradient for L1 Minimization (SPGL1) [26], and the Generalized Approximate Message Passing (GAMP) [155].

Alternative solutions rely on the observation that the central issue in the recovery of x is finding a vector with a proper sparse representation. Following this path, iterative greedy approaches, which iteratively adjust their sparsity at each step, have been proposed to reduce computational costs. Possible examples are the Orthogonal Matching Pursuit (OMP) [181] and the Compressive Sampling Matching Pursuit (CoSaMP) [142]. Methods investigating another direction try to adapt the decoding process to the class of signals to acquire (see, e.g., [153, 198, 126] where the decoder stages are tuned on the reconstruction of ECGs). These schemes exploit statistical priors on the signal to favour reconstructions close to what is typical in the class of acquired signals.

¹The problem is called *Basis Pursuit with DeNoising* (BPDN) if noise corrupting y is also considered

More recently, it has been demonstrated that additional advantages in terms of a lower computational complexity or improvement in the quality of the reconstructed signal can be obtained by adopting a (Deep) Neural Network (DNN) for reconstruction [136, 107, 140, 166, 141, 91, 199, 174]. More specifically, in [141], authors have shown a probabilistic relation between CS and a stacked denoising autoencoder (SDA) implemented as a 3-layer neural network. Once adequately trained, the SDA can directly recover a sparse image from its linear (or mildly non-linear) measurements and has offered, in some cases, advantages in terms of the quality of the reconstructed images compared to the most common greedy reconstruction algorithms. A similar approach that employs fully-connected DNNs can be found in [91], where CS has been applied to videos, and the proposed approach enables fast recovery of video frames at a significantly improved reconstruction quality. In [199], authors have proposed a DNN called ISTA-Net and inspired by the Iterative Shrinkage-Thresholding Algorithm (ISTA) [23], which has been designed to optimize the solution of BP to reconstruct compressed images. Another deep learning model (BW-NQ-DNN) applied to CS acquisition/reconstruction of neural recording has been presented in [174]. Here, three networks have been jointly optimized to perform binary measurement matrix multiplication, non-uniform quantization, and reconstruction. Despite the advantage shown in terms of quality of reconstruction, this approach has a few drawbacks: *i*) it requires a pre-processing stage detecting signal peaks, which adds complexity to the encoder and specializes it to spiky signals; *ii*) it quantizes CS measurements after a programmable non-linearity, which adds further complexity.

This chapter proposes an innovative use of DNNs in a CS framework. Unlike all the cases mentioned above that use DNNs to reconstruct the input signal directly, our model only provides a *divination* of the support of the input signal, i.e., the positions of the non-null components characterizing the sparse representation ζ . Knowing the signal support drastically reduces the complexity of the signal recovery and increases achievable performances. Moreover, the approach presented here not only improves reconstruction quality compared to state of the art adapted CS frameworks but also introduces a *self-assessment capability* that allows estimating *on the fly* the quality of reconstruction. Furthermore, signals can be successfully reconstructed even when they refer to very short acquisition windows, a crucial feature that further reduces the complexity and a mixed-signal implementation of the acquisition stage. The discussion reported here also refer to the following publications [119, 124].

5.1 Compressed Sensing

This section formalizes a generic CS framework. Let us assume to work input instances that can be derived by chopping time series (or waveforms) into subsequent windows or collecting readings from several sensors at the same timestamp. In any

case, each instance is represented by a vector $x \in \mathbb{R}^n$. Say that x is k -sparse means that the sparsity basis S exist such that $x = S\zeta$, with the vector $\zeta = (\zeta_1, \dots, \zeta_n)$ does not contain more than $k < n$ non-zero entries.

The fact that x depends only on scalars that are less than its actual dimensionality makes the compression possible. CS compresses x by applying a linear operator $\text{Enc} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with $m < n$ named acquisition (or encoding) matrix. Enc is defined in such a way that $x \in \mathbb{R}^n$ can be retrieved from $y = \text{Enc}(x)$ with $y \in \mathbb{R}^m$. The ratio n/m is the *compression ratio* and will be indicated by CR.

One can intuitively accept that a less sparse signal with a larger κ requires a larger number of measurements m to guarantee the recovery of x from y and achieves a lower CR. This relationship is asymptotically identified by CS theory as $m = \mathcal{O}(\kappa \log(n/\kappa))$ [45]. In the finite and practical cases, one may often aim at using a m value proportional to κ . Nevertheless, the worst-case theoretical guarantees fail for $m < 2\kappa$. Indeed, despite the infinite number of counterimages of $y = \text{Enc}(\zeta)$, signal recovery requires that only one of them survives when we add the k -sparsity prior. Hence, given any two k -sparse vectors ζ' and ζ'' it cannot be $y = \text{Enc}(\zeta')$ and $y = \text{Enc}(\zeta'')$, i.e., $\text{Enc}(\zeta' - \zeta'')$ must be non-zero. Hence, $\zeta' - \zeta''$ cannot be in the kernel of Enc . Since, in the worst-case scenario, $\zeta' - \zeta''$ is 2κ -sparse, the only way to guarantee this condition is that AS is a maximum rank operator when restricted to any 2κ -dimensional coordinate subspace of \mathbb{R}^n . However, this is not possible because $m < 2\kappa$ and the sparsity prior can no longer guarantee signal recovery whenever the worst-case scenario is hit. In practice, though worst-case scenarios seldom appear, classical reconstruction algorithms fail before the limit $m = 2\kappa$ is reached.

Clearly, compression by Enc must be coupled with a signal reconstruction stage² $\text{Dec} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ such that ideally $x = \text{Dec}(\text{Enc}(x))$. In practice the chain of the encoding and decoding step is a lossy process and $\hat{x} = \text{Dec}(\text{Enc}(x))$ is only an approximation of x .

5.1.1 Encoder

The class of linear operators Enc that can be effectively paired with a decoder Dec is extremely large. Most notably, if A is an instance of a matrix whose entries are independent zero-average and unit-variance Gaussian random variables, then $\text{Enc}(x) = Ax$ is known to work [61, 45, 122] with very high probability. Yet, if the matrix A^\pm is defined as $A_{j,k}^\pm = \text{sign}(A_{j,k})$, then $\text{Enc}(x) = A^\pm x$ is also known to work with very high probability [85]. In the following, we will focus on $\text{Enc}(x) = A^\pm x$ as this makes the computation of $\text{Enc}(x)$ multiplierless and is thus the best option for very low resources implementations of the encoder stage.

The Literature shows that there is plenty of room for optimizing A [63, 185, 118, 120], and suitably designed matrices can increase compression considerably compared to naive random instances. This aspect paves the way to applications in all

²Terms like *decoding* or *recovery* are also used to describe this stage.

settings where the computational complexity of compression must be kept at bay, e.g., in wireless sensor networks, for which reduced computation and compression before transmission are essential to fit within a tight resource budget.

Pros/Cons of short windows It is worth stressing that, to best express its potential in reducing computational complexity at the encoder, CS should consider the shortest possible acquisition windows. To understand why, consider the processing of N given samples. They may be partitioned into N/n contiguous and non-overlapping time windows, each with n samples. Operator Enc can be applied to each window, entailing a number of operations $\mathcal{O}(n \cdot m)$. The total number of operations to process the N samples is $\mathcal{O}(n \cdot m \cdot N/n) = \mathcal{O}(n \cdot N/CR)$. However, CR is fixed to a level guaranteeing the recovery of the original n -dimensional signal x from the m measurement y with a quality deemed acceptable. Hence, at given CR and N , the computational complexity linearly increases with n , i.e., the length of individual time windows.

Another aspect that has to be considered is the signal reconstruction latency. Even assuming Dec(y) as an instantaneous operation, the signal is recoverable only when y is available, which means after n time steps. Of course, the lower n , the lower latency.

Beyond these high-level reasons, short windows may also benefit the encoder's implementation at a physical level. In purely digital realizations [197, 24, 25], the samples come from a conventional Analog-to-Digital converter, and the encoder is implemented as a sequence of sums and subtractions depending on the entries of A^\pm . In this case, the computation time and the memory needed to store A reduces when n (and m) gets smaller. In mixed-mode realizations (i.e., in the design of Analog-to-Information converter based on CS) [192, 52, 169, 75, 150], $y = A^\pm x$ is computed component-wise as $y_j = \sum_{k=0}^{n-1} A_{j,k}^\pm x_k$, i.e., accumulating the signal samples in the analog domain. This operation implies an analog storage to hold the intermediate sum value. However, independently of the actual implementation and technology, the approach is doomed to suffer from leakage, and disturbance [150, 148]. These phenomena degrade the stored value over time, and their effectiveness increases with the hold time and the number of sums. Hence, a lower n shortens the acquisition time, reduces the number of operations to compute y_j , and attenuates the degradation before digital conversion.

Regrettably, gaining all the advantages connected with reducing n is not straightforward. Indeed, real-world signals are such that, when n shrinks, the ratio κ/n is expected to increase. Since κ affects m , any reduction of n tends to impair the compression ratio. As a remark, the trend with which κ/n increases when n decreases is a feature of the class of signals considered.

To get a quantitative feeling of these trends, we show in Figure 5.1 the normalized sparsity κ/n for different values of n observed in the classes of ECG and EEG signals. Instances are obtained according to the synthetic generators described in

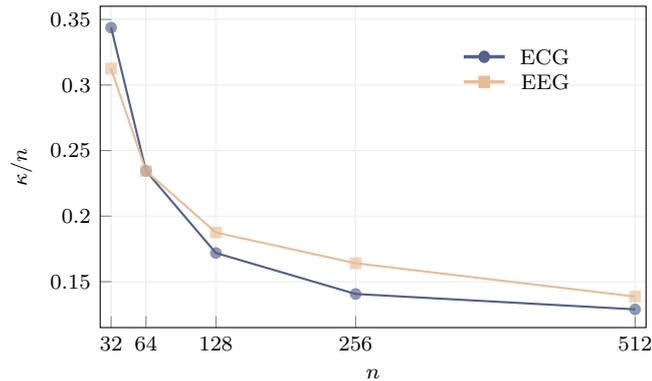


FIGURE 5.1: The effect of reducing n on the normalized sparsity κ/n in the two examples of a synthetic ECG and a synthetic EEG signal.

the Appendix. Moreover, for both classes of bio-signals, the considered sparsity basis S is a family of the orthogonal Wavelet functions [116]. In more detail, we select the Symmlet-6 family as sparsity basis for ECG signals [150], while our choice for the EEG case is the Daubechies-4 family [28].

The value of κ is a system parameter estimated at design time so that the representations on the sparsity basis of most of the signal instances feature a number of non-negligible elements not greater than κ . In Figure 5.1, κ is estimated as the least number of entries in the sparse representation that includes 99.5% energy in 99% ECG instances, and 95% energy in 99% EEG instances³. The figure shows that the smaller the n , the larger the (normalized) sparsity, and therefore the lower the attainable CR that ensures a target reconstruction quality.

The above considerations reveal the presence of a multi-faceted trade-off linking computational effort, implementation complexity, reconstruction quality, and compression.

5.1.2 Decoder

To better formalize sparsity and its consequences, recall $x = S\zeta$ and that not more than κ entries of ζ are non-null. The positions of the non-zero entries of ζ identify the so-called *support* that we will represent by means of the binary vector $s \in \{0, 1\}^n$ such that $s_j = 1$ if $\zeta_j \neq 0$ and $s_j = 0$ otherwise. Binary, n -dimensional vectors can be used to *index* a generic n -dimensional vector v so that $v|_s$ is the subvector of v collecting only the entries v_j such that $s_j = 1$. We will use binary n -dimensional vectors also to *index* matrices M with n columns so that $M|_s$ is the submatrix of M that contains only the columns whose index j is such that $s_j = 1$. With this notation, κ -sparsity is equivalent to say that two pieces of information efficiently

³For the class of EEG signals we refer to a synthetic signal that emulates event-related brain potentials, where readings in each lead contain information on the external stimulus as well as a part on other neurons activity. This latter justifies the assumption that 95% energy is enough to identify the significant components of the signal.

represent x , namely the n -dimensional binary vector s and the real vector $\zeta_{|s}$ whose dimensionality does not exceed κ .

Sparsity is fundamental in the decoding process going from y back to x . In fact, since $m < n$, the mapping $y = A^\pm S \zeta$ from ζ to y is non-injective. Hence, any given measurement vector y corresponds to an infinite number of possible ζ . However, if A is properly designed, only one of the counterimages of y is κ -sparse and can be found by relatively simple algorithmic means.

Among the many methods proposed in the literature, the most classical approach is BPDN, which recovers both pieces of information simultaneously by solving the optimization problem

$$\hat{\zeta} = \arg \min_{\zeta \in \mathbb{R}^n} \|\zeta\|_1 \quad \text{s.t.} \quad \|y - A^\pm S \zeta\|_2 \leq \epsilon \quad (5.1)$$

where $\hat{x} = S \hat{\zeta}$ is the reconstructed signal, and $\epsilon \geq 0$ accounts for the possible presence of disturbances in the computation of y by relaxing the constraint $y = A^\pm S \zeta$ that would hold in the noiseless case. The noiseless case can be tackled by setting $\epsilon = 0$, corresponding to the BP problem. Though implicitly performed, support identification is an essential ingredient in BP and BPDN and is embedded in the 1-norm used in the objective function. The use of the 1-norm minimization replaces the minimization of the cardinality of the support of ζ that would yield a combinatorial problem. Indeed, 1-norm minimization tends to select the ζ with the least number of non-zero entries among all the possible ζ satisfying the constraint [61]. This property is so critical that changing the 1-norm in the merit function would completely spoil reconstruction, while changing the 2-norm in the constraint usually still gives sensible results. Note that, despite its fundamental merit, the 1-norm minimization is only a *proxy* of support identification, which works under suitable assumptions that are not necessarily satisfied in practice, especially for large κ/n values [61].

Decoder with Support Oracle Since we enlarge the application of the CS framework to the cases where κ/n is quite large, we here consider a different approach in which support identification is performed by an *oracle* looking at the vector y and divining s . Once s is known one may note that $y = A^\pm S \zeta$ is equivalent to $y = A^\pm S_{|s} \zeta_{|s}$ to estimate $\zeta_{|s}$ as shown in Figure 5.2.

Since the goal is to compute the non-null entries of ζ , if s is unknown, the signal recovery is performed by inverting a wide matrix (an ill-defined problem), thus obtaining both null and non-null entries of ζ . Otherwise, assuming that an oracle divining s exists, the recovery problem only focuses on the computation of the non-null entries of ζ such that the recovery stage only performs the (pseudo-)inversion of a tall matrix.

Let \hat{s} be the estimate of s divined by the support oracle, $\zeta_{|s}$ may be estimated as

$$\hat{\zeta}_{|\hat{s}} = (A^\pm S_{|\hat{s}})^\dagger y \quad (5.2)$$

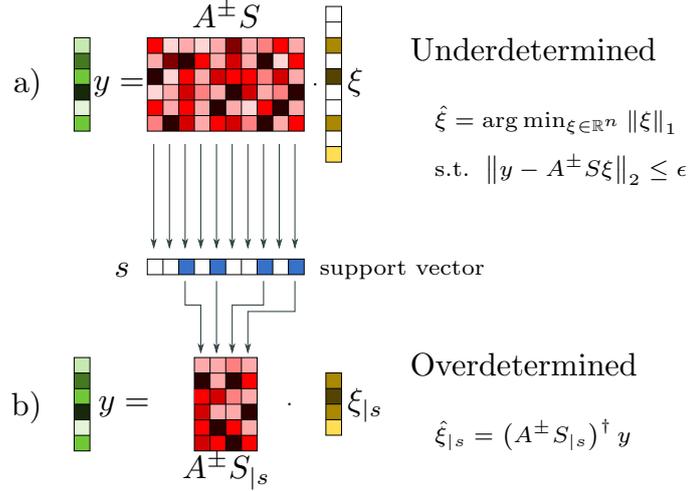


FIGURE 5.2: The original underdetermined problem (a) turned into an overdetermined problem (b) by exploiting the support vector s .

where \cdot^\dagger indicates Moore-Penrose pseudo-inversion that is needed since the number of ones in \hat{s} is in the order of $\kappa < m$ and the matrix $A^\pm S|_{\hat{s}}$ is a *tall* matrix with more rows than columns. The two estimations \hat{s} and $\hat{\xi}_{|\hat{s}}$ define the recovered signal \hat{x} .

5.1.3 Performance indexes

The encoder-decoder chain may simultaneously perform more than one useful operation on the signal (see, e.g., [43, 44, 29, 201] for its use as an encryption stage) of which compression is undoubtedly the most obvious as $m < n$. The compression performance of the encoder-decoder chain is easily assessed by the compression ratio n/m .

However, such compression is in general lossy, and some degradation appears yielding $\hat{x} \neq x$. The closer \hat{x} to x , the better the encoder-decoder chain and this can be assessed using the Reconstruction Signal-to-Noise Ratio (RSNR) defined as

$$\text{RSNR} = 20 \log_{10} \left(\frac{\|x\|_2}{\|x - \hat{x}\|_2} \right) \quad (5.3)$$

RSNR can be used to define two ensemble-level performance figures, computed starting from a set $x[t]$ (for $t = 1, \dots, T$) of signal instances recovered as $\hat{x}[t]$. The first is the Average RSNR (ARSNR)

$$\text{ARSNR} = \mathbf{E} [\text{RSNR}] \approx \frac{1}{T} \sum_{t=1}^T \text{RSNR}[t] \quad (5.4)$$

while the second is the Probability of Correct Reconstruction (PCR) that, given a RSNR_{\min} value, is defined as

$$\text{PCR} = \Pr \{ \text{RSNR} \geq \text{RSNR}_{\min} \} \approx \frac{1}{T} \# \{ t \mid \text{RSNR}[t] \geq \text{RSNR}_{\min} \} \quad (5.5)$$

where $\#$ counts the number of elements in the set. The value of RSNR_{\min} has to be set accordingly to the minimum RSNR level that is considered *sufficient* for a correct reconstruction.

5.2 Data sets

Due to the large number of signal instances needed, in general, to train a neural network, both in the ECG and in the EEG cases, we used a MATLAB code to generate synthetic instances of the two classes of signals.

As mentioned in Section 5.1.1, ECGs exhibit sparsity with respect to the orthonormal set of vectors representing the Symmlet-6 wavelet family transformation. Here, κ is set on 16 for $n = 64$ and 24 for $n = 128$. For the EEG signals the sparse vectors ξ are with respect the basis representing the Daubechies-4 wavelet transformation where $k = \{16, 26\}$ matches $n = \{64, 128\}$.

ECG The synthetic generator⁴ of ECGs is thoroughly discussed in [131]. Signals are generated as noiseless waveforms. The noisy cases are obtained by superimposing additive white Gaussian noise whose power is such that the intrinsic SNR (ISNR) is 60 dB.

The setup is the same detailed in [118]. The heart-beat rate is randomly set using an uniform distribution between 60 beat/min and 100 beat/min. We generate chunks of 2 s with a 256 sample/s sampling frequency, that are split into windows of n subsequent samples. For both $n = 64$ and $n = 128$ cases we generate 8×10^5 input vectors x such that the corresponding total number of signal chunks are 10^5 and 2×10^5 . These input vectors are randomly split between a training set and a test set where the latter contains 20% of the total vectors x .

EEG The detailed description of the code to generate the synthetic EEG signal⁵ can be found in [189]. The generator emulates event-related brain potentials, modelling an evoked potential as the series of a positive and a negative peak occurring at a fixed time relative to the event. The peaks are added to the uncorrelated background noise, whose power is set to a level such that the resulting signal is very similar to an EEG signal measured by a real scalp electrode. Though the software can generate all channels in a multi-electrode EEG according to the standard 10-20 system, we focus on the "Fz" electrode since it is in proximity (but not exactly on the top) of the simulated source of the stimulus. The sampling rate is set to 1024 sample/s with a stimulus frequency of 1 Hz.

⁴The MATLAB code is freely available for download from the Physionet website at <http://physionet.org/content/ecgsyn/>

⁵The MATLAB code is freely available for download from the Medical Research Council Brain Network Dynamics Unit at the University of Oxford website at <http://data.mrc.ox.ac.uk/data-set/simulated-ecg-data-generator>

We generate tracks corresponding to 50 different patients by starting from the parameters used in [189] and adding a random uniformly distributed offset to each of them. The ranges of the offsets for the positive peaks are ± 16 samples for the position of the peak, ± 0.05 Hz for the peak frequency and ± 1 for the peak amplitude. Ranges for the random offsets for negative peaks are ± 26 samples for the position of the peak, ± 1 Hz for the peak frequency and ± 4 for the peak amplitude.

The signal length for each patient is such that the total number of n -sample windows is 8×10^5 . After that, 20% of each patient's signal instances are randomly selected to contribute at the test set, while the remaining 80% is for the training phase.

5.3 Support Oracle

The proposed oracle is based on a DNN trained on signals with the same statistical features as the one to be acquired. The DNN SO : $\mathbb{R}^m \rightarrow [0, 1]^n$ is defined by the connection parameters, with m inputs that correspond to the m entries of the measurement vector y and n outputs.

The neural network has three intermediate fully connected layers of cardinality $2n$, $2n$, and n , all with a Rectified Linear Unit (ReLU) activation function. The output layer is also fully connected with n units and sigmoidal activation function that map any scalar a into $(1 + e^{-a})^{-1}$. Training also adapts the matrix A so that encoder and decoder are jointly optimized to improve support identification and thus to improve reconstruction performance.

Both the parameters of the support oracle SO and the matrix A are initialized as instances of independent zero-average unit-variance Gaussian random variable and adjusted by training the compound system $\text{SO} \circ \text{Enc} : \mathbb{R}^n \rightarrow [0, 1]^n$. The training set comprises a sequence of κ -sparse signals $x = S\zeta$ and the corresponding binary vectors s . The true support of ζ encoded in s and the output $o = \text{SO}(\text{Enc}(x))$ of the DNN are compared with a loss function, which is the binary cross-entropy between s (which acts as a label) and o defined by

$$L(x, s) = -\frac{1}{n} \sum_{j=1}^n [s_j \log(o_j) + (1 - s_j) \log(1 - o_j)] \quad (5.6)$$

Though $\text{Enc}(x) = A^\pm x$ in the forward pass, to prevent the sign function from interrupting error backpropagation, in the backward pass we assume $\nabla_A \text{Enc}(x) = \nabla_A (Ax)$. With this, since $A_{j,k}^\pm = \text{sign}(A_{j,k})$ for every j and k , the training acts on the continuous-valued parameters whose sign is used in feedforward computation.

Using the methods specified in Section 5.2, we generate a dataset composed of 8×10^5 signal instances for both the ECG and the EEG case. Each dataset is split into 80% for training (training set) and 20% for performance assessment (validation set).

All models proposed in this paper are implemented and trained using the TensorFlow framework [1] with the help of the high-level API provided by Keras [54]. Training is performed with stochastic gradient descent, where each gradient step is computed with a mini-batch comprising 30 signal instances and an initial learning rate value of 0.1.

With an eye on complexity, the network with $n = 64$ and m ranging in $[16, 40]$ contains from 32 128 to 36 736 parameters and in our examples is trained for 500 epochs⁶. With $n = 128$ and m ranging in $[24, 64]$, the model counts from 124 672 to 140 032 parameters and in our examples is trained for 1 000 epochs. Even assuming that each parameter is encoded in 4 B, the total memory footprint is limited below 150 KiB for $n = 64$ and below 550 KiB for $n = 128$. Such requirements may easily fit within the memory budget of commercially available devices used for small scale computation and gateway tasks.

5.4 Trained CS with Support Oracle

The trained oracle can be exploited in the definition of the decoder reported in Figure 5.3. We compute $o = \text{SO}(y)$ and, given a certain threshold $o_{\min} \in [0, 1]$, we estimate s with the binary vector $\hat{s} \in \{0, 1\}^n$ such that $\hat{s}_j = 1$ if $o_j \geq o_{\min}$ and $\hat{s}_j = 0$ otherwise. Starting from \hat{s} we finally estimate $\hat{\zeta}_{|\hat{s}}$ with (5.2).

Decoder operations depend on the value of o_{\min} that is set by a further training phase in which each vector in the training set is encoded and decoded for different values of o_{\min} . The o_{\min} yielding the highest ARSNR is selected. We name our approach Trained CS with Support Oracle (TCSSO) to summarize its main features.

We compare the performance of TCSSO with that of some well-known methods. Since TCSSO simultaneously adapts encoder and decoder, we pair some classical signal recovery algorithms with an established technique for optimizing the matrix A^\pm that can cope with the antipodality constraint on the entries.

The sensing matrix design follows the rakeness-based CS framework [118, 120] that we have verified to yield better results compared to the classical independent assignment of ± 1 to each of the entries of A^\pm . Performance improvement comes from adapting the statistics of the rows of the sensing matrix to the statistics of the acquired class of signals. As decoders, we consider BP and BPDN as presented in (5.1) along with Orthogonal Matching Pursuit (OMP) [181] and Generalized Approximate Message Passing (GAMP) [155]. OMP is a lightweight greedy approach that iteratively estimates the signal support while GAMP is often better than BP and BPDN as it exploits the Gaussian approximation of BP that usually holds for large n values. When dealing with ECGs, we also test the performance of the Weighted ℓ_1 minimization (WL1) [198] as a representative of decoders that exploit statistical priors on the signal support. In all the tested cases, BP outperforms BPDN such that, in the rest of the paper, we consider BP as a reference for standard CS decoder.

⁶In each epoch the training algorithm walks through the entire training set.

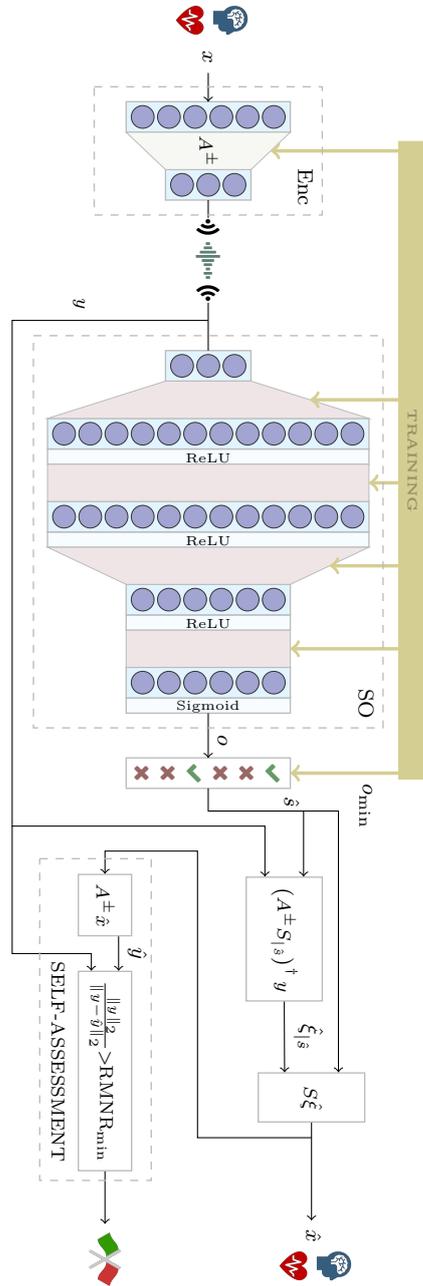


FIGURE 5.3: Block diagram for the Trained CS with Support Oracle including self-assessment capability.

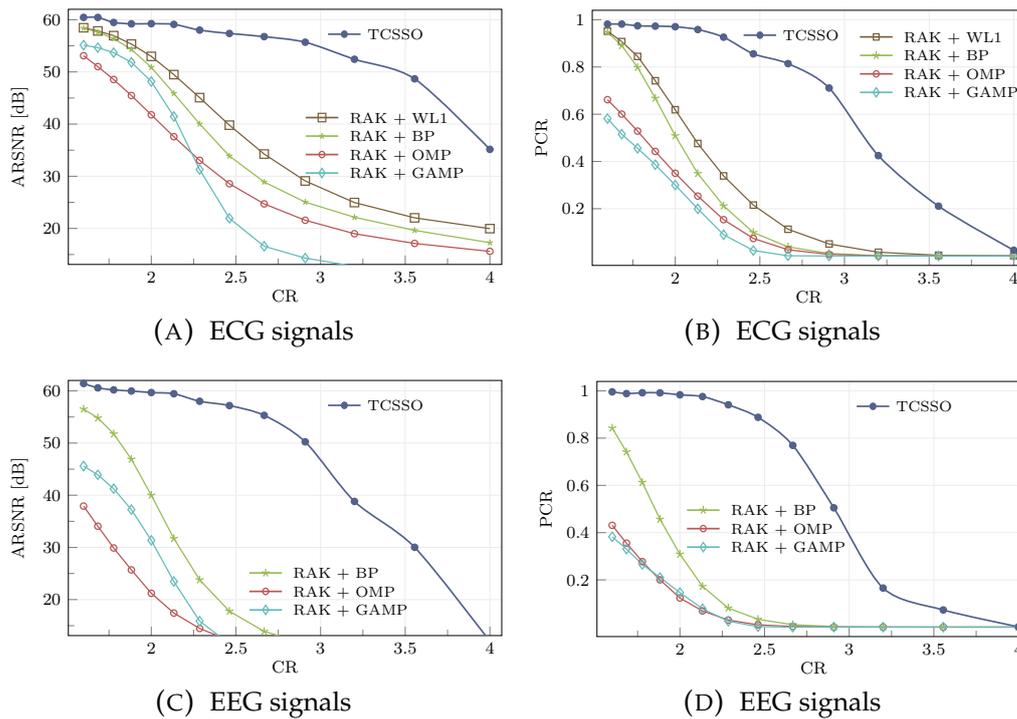


FIGURE 5.4: TCSO performance of TCSO compared against OMP, BP, GAMP, and WL1 (only for ECG) with rakesness adaptation (RAK) of the sensing matrices in terms of ARSNR (a)-(c) and PCR with $\text{RSNR}_{\min} = 55$ dB (b)-(d).

We evaluate ARSNR and PCR by Montecarlo simulations using the samples of the validation set for both ECG and EEG cases with a superimposed noise resulting in an Intrinsic Signal-to-Noise Ratio (ISNR) equal to 60 dB. The achieved performances are reported in Figure 5.4 for the $n = 64$ and $\kappa = 16$ case. In all plots, the number of measurements sweeps from $m = 40$ down to $m = 16$ thus focusing on compression ratios from $\text{CR} = 1.6$ up to $\text{CR} = 4$. TCSO outperforms all other techniques and allows us to work at compression ratios much larger than those commonly achievable while still requiring a limited computational effort since $n = 64$. For example, to guarantee $\text{ARSNR} = 50$ dB, results in Figure 5.4(a),(c) show that by using TCSO one may get $\text{CR} \approx 3.5$ for ECGs and ≈ 2.9 for EEGs. In the same setting, RAK+WL1 is the best performing competitor for ECGs with $\text{CR} \approx 2.2$ while RAK+BP is considered a benchmark for the EEGs with $\text{CR} \approx 1.8$.

Figure 5.5 shows how the situation changes when n increases from 64 to 128. Performance is reported only in terms of ARSNR and for TCSO along with its best competitor. The increase of n positively impacts performances in general since κ/n decreases. Nevertheless, TCSO still outperforms the best of the traditional CS frameworks. Considering $\text{ARSNR} = 50$ dB as the desired quality of service, TCSO works with $\text{CR} \approx 4.4$ and $\text{CR} \approx 2.9$ while the competitors give at most $\text{CR} \approx 2.7$ and $\text{CR} \approx 2.2$ in the case of ECGs and EEGs, respectively.

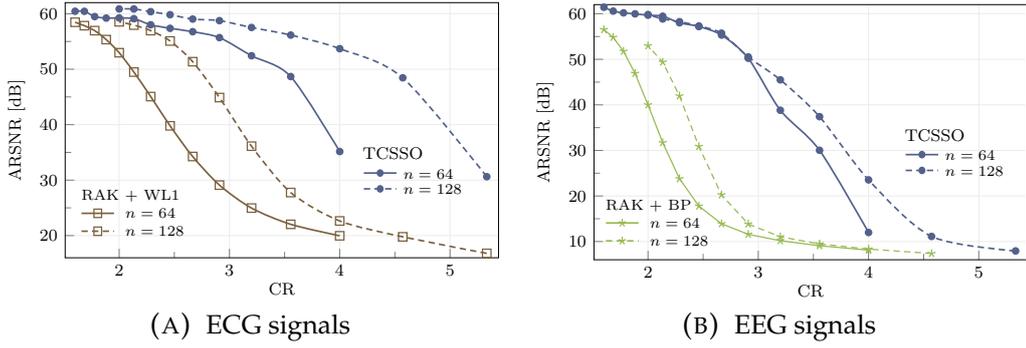


FIGURE 5.5: TCSSO performance for $n = 64$ and $n = 128$.

5.4.1 Preliminary evidence on real signals

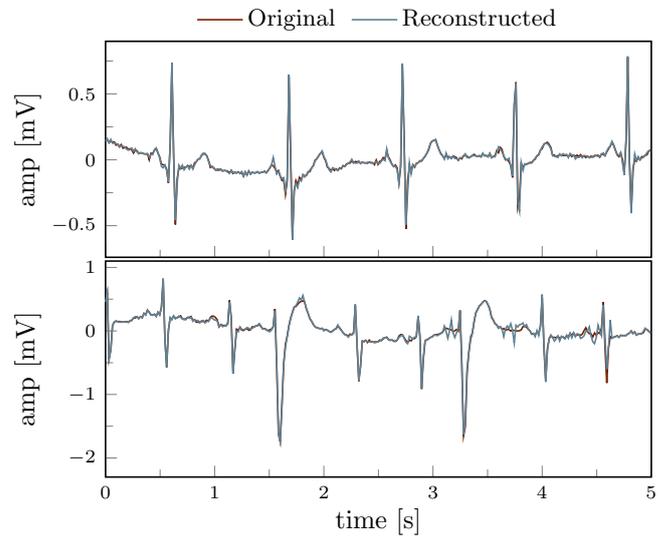
The ideal path for applying our method in real-world cases is to collect enough acquisitions to substantiate both a training and a validation set. The acquisitions we have access to do not currently allow such a thorough assessment. However, some evidence can be given using synthetic data for the learning phase and real-world signals for a preliminary assessment. This approach is suboptimal since there is no guaranteed coherence between the training set used for learning and the validation set used for assessment. The results are still encouraging.

In particular, we may consider the waveforms contained in the MIT database for testing compression of ECG signals [79, 138] (records 11950_03 and 12531_03 in the MIT-BIH ECG compression test database) and the pool of acquisitions used in [28] for EEG (recording from the “Fz” electrode). A sample comparison between original and reconstructed waveforms for $n = 64$ and $m = 32$ (which is equal to 2κ) is reported in Figure 5.6. Despite the appearance of some artefacts introduced by the decoder, the plots show that, even with the suboptimal setting, our method can yield acceptable reconstructions with extremely small n and with m below the classical threshold 2κ .

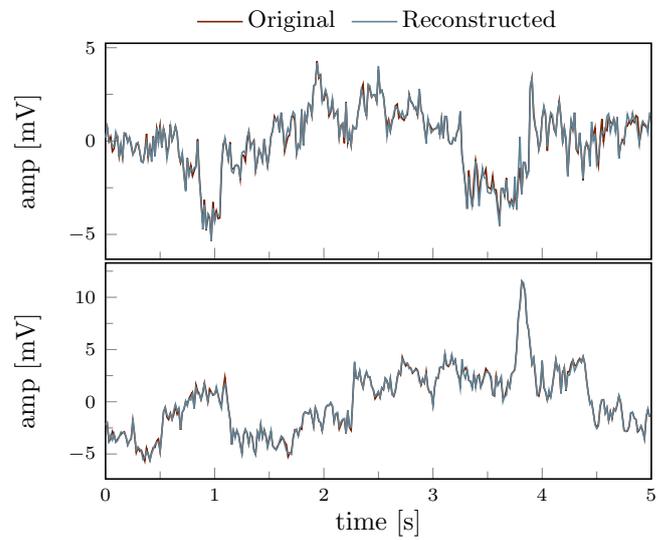
5.5 Decoder self-assessment

The TCSSO architecture described in the previous section can be extended by exploiting a property that stems from the fact that s is estimated separately from $\tilde{\zeta}_{|s}$.

In fact, assume that no noise is present and that the size and content of A^\pm are such that $y = A^\pm S \zeta$ is satisfied by one and only one κ -sparse ζ , i.e., that recovery of the true signal is theoretically possible. If the oracle is successful in divining the support, then $\hat{s} = s$ and $y = A^\pm S_{|s} \tilde{\zeta}_{|s}$ implies that $y \in \text{span}(A^\pm S_{|s})$. This has a twofold consequence: *i*) (5.2) computes $\hat{\zeta}_{|s} = \tilde{\zeta}_{|s}$, *ii*) if $\hat{\zeta}$ is mapped back we have $A^\pm S \hat{\zeta} = y$.



(A) ECG signals.



(B) EEG signals

FIGURE 5.6: Examples of reconstruction of real-world signals compared with the original waveforms with $n = 64$ and $m = 32$.

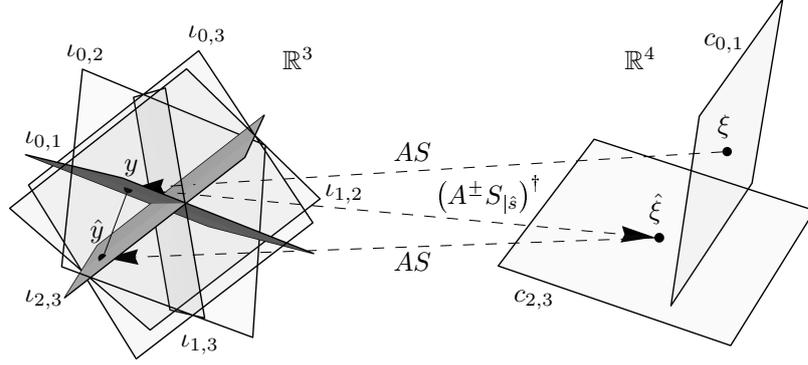


FIGURE 5.7: Self-assessment mechanism in decoders based on a support oracle.

However, if the oracle fails, then $\hat{s} \neq s$ and since ζ is the unique κ -sparse solution of $y = A^\pm S \zeta$ then $y \notin \text{span}(A^\pm S_{|s})$. This has a twofold consequence: i) (5.2) computes $\hat{\zeta}_{|s} \neq \zeta_{|s}$, ii) if $\hat{\zeta}$ is mapped back we have $A^\pm S \hat{\zeta} \neq y$.

Clearly, the decoder cannot check the correctness of $\hat{\zeta}$ as the true ζ is unknown. Nevertheless, it may map $\hat{\zeta}$ back to measurement obtaining $\hat{y} = A^\pm S_{|s} (A^\pm S_{|s})^\dagger y = A^\pm \hat{x}$ that could be different from y . As a result, $\|y - \hat{y}\|_2$ is most naturally linked to the decoder failure and grants a useful self-assessment capability. In particular, one may monitor the quantity

$$\text{RMNR} = \frac{\|y\|_2}{\|y - \hat{y}\|_2} \quad (5.7)$$

that is the Reconstruction Measurements-to-Noise Ratio, and declare that the oracle, and thus the TCSO decoder, has succeeded when $\text{RMNR} \geq \text{RMNR}_{\min}$ for a certain threshold.

This situation can be exemplified in the small-dimensional case $n = 4$, $\kappa = 2$ and $m = 3$ with

$$A^\pm = \begin{pmatrix} +1 & +1 & +1 & +1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & +1 & -1 \end{pmatrix}$$

Since $\kappa = 2$, the instances of the original signal $\zeta \in \mathbb{R}^4$ may have at most two non-null components and thus lay on the union of all the possible coordinate planes in \mathbb{R}^4 . We may indicate one of those planes as $c_{j,k}$ where j and k are the indexes of the non-null coordinates of its points. The matrix A^\pm maps each of those 6 coordinate planes into a plane in \mathbb{R}^3 that can be distinguished from the others. This is exemplified in Figure 5.7 on the left of which we draw the 6 planes $l_{j,k} \subset \mathbb{R}^3$ that are the images through $A^\pm S$ of the coordinate planes $c_{j,k} \subset \mathbb{R}^4$. Note that, due to dimensionality reduction, images are not pairwise orthogonal. However, recovery is theoretically possible as no two images $l_{j,k}$ and $l_{j',k'}$ are the same. Therefore a sufficiently clever algorithm can establish the support by looking at the measurement vector y .

Assume now that $s = (1, 1, 0, 0)$, i.e., that the true signal $\zeta \in c_{0,1}$ is mapped

by $A^\pm S$ into a measurement vector $y \in \iota_{0,1}$. Assume also that the oracle mistakes the support and estimates $\hat{s} = (0, 0, 1, 1)$, implying $\hat{\zeta} \in c_{2,3}$. By computing (5.2), the vector y is mapped back to $\hat{\zeta}$ on that plane, which is therefore different from ζ . Though only approximately, the same holds in the noisy case and explains why the difference between y and \hat{y} assesses the correctness of the divined \hat{s} , i.e., the quality of the reconstruction \hat{x} .

As an example of the underlying mechanism, Figure 5.8 reports some Montecarlo evidence on the relationship between RMNR and RSNR for the ECG signals and in three different configurations. In Figure 5.8a no noise is present and $m = 32 = 2\kappa$; in Figure 5.8b ISNR = 60 dB and $m = 32 = 2\kappa$, whereas in Figure 5.8c no noise is present, but $m = 24 < 2\kappa$.

The two-dimensional plots show an estimation of the joint-probability, conditioned to the *positive* events, i.e., the support has been correctly identified ($\hat{s}_j \geq s_j$ for all $j = 1, \dots, n$, orange points) or to the *negative* events, i.e., at least one entry in the support is neglected ($\hat{s}_j < s_j$ for at least one j , blue points). Darker colours stand for higher densities.

The one-dimensional plots at the bottom of the figure report the error probabilities of a self-assessment procedure that calls for a positive event whenever $\text{RMNR} \geq \text{RMNR}_{\min}$ and for a negative event otherwise. As the threshold RMNR_{\min} increases, the probability of a false positive decreases since only very high RMNR reconstructions are declared correct. On the contrary, the probability of a false negative increases since for larger RMNR_{\min} even good reconstructions can be declared incorrect.

The ideal conditions in Figure 5.8a result in perfect self-assessment capabilities. When noise is added as in Figure 5.8b, positive and negative cases get mixed but remain identifiable by looking at RMNR.

Though no noise is present in Figure 5.8c, the fact that $m < 2\kappa$ makes the number of measurements insufficient for signal reconstruction, as there is no guarantee that only one κ -sparse signal ζ corresponds to the given y through $A^\pm S$. Hence, more than one support corresponding to the measurement exists. In these conditions, it may happen that the oracle divines a support that includes the true one (more than κ outputs of the network are larger than σ_{\min}) as well as components of other possible supports. In this case, the oracle is not missing the support (orange point in the lower-right cluster in the scatter plot of Figure 5.8c). However, pseudo-inversion spreads the reconstruction over all the available components, thus failing to reconstruct the signal. It may also happen that the oracle divines a support different from the true one. In this case, the oracle is wrong (blue points in the lower-right cluster in the scatter plot of Figure 5.8c), and pseudo inversion identifies a sparse signal that is not the true one. Both cases give rise to points for which RMNR is very high, but the RSNR is very low, and no matter how high the RMNR_{\min} , the probability of a false positive is not vanishing.

Luckily enough, the above cases are the ones breaking *worst-case* guarantees and

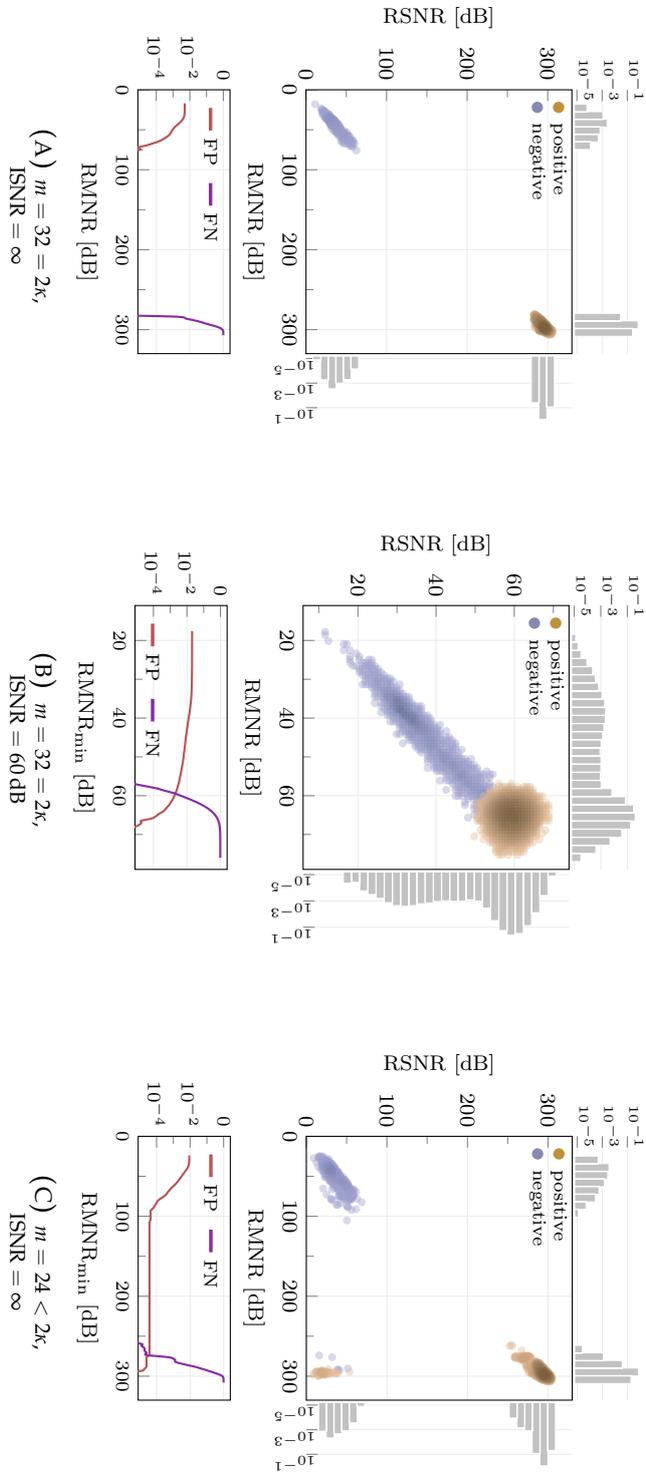


FIGURE 5.8: The relationship between RSNR and RMNR for ECG signals for $n = 64$ and $\kappa = 16$ with respective logarithmic histograms above and on the right.

happen quite rarely: in our 1.6×10^5 validation set, for $n = 32$, $\kappa = 16$ and $m = 24$, the oracle divines a support in excess of the true one only 109 times, and a support different from the true one only six times. The statistics commonly used to assess performance remain substantially unaltered by these failures that are undetectable by looking at the RMNR.

In general, the value of RMNR_{\min} can be decided once that σ_{\min} is set, by a further pass over the training set. This threshold allows us to estimate false positive and false negative curves as in Figure 5.8, and use them as criteria. In the following, we will set RMNR_{\min} as the largest value for which false negative probability is negligible. Whenever a failure is detected, the decoder may take different actions whose effectiveness depends on the final applications.

The exploration of all the possibilities of the resulting two-level decoder is out of the scope of this paper. However, it can be easily recognized that quite a few options are available, such that:

- i) raising a warning and mark the current window as potentially incorrect;
- ii) feeding the warning back to the encoder and require further information to correct the reconstruction (thus lowering the CR for this instance);
- iii) triggering another decoder on the same measurement vector hoping that this will improve reconstruction;
- iv) any combination of the above.

We provide a partial and non-optimized example whose only aim is to show that some information can still be extracted from the measurements when first-attempt TCSSO decoder fails. In detail, we trigger GAMP⁷ as a second-wind decoder.

Figure 5.9 plots the probability that GAMP yields a RSNR larger than what is given by TCSSO when applied to the instances that the latter marks as incorrectly recovered as $\text{RMNR} < \text{RMNR}_{\min}$, as a function of CR for the $n = 64$, $\kappa = 16$ case. A second-wind decoding is functional when such a probability is larger than 50%, i.e., approximately for $\text{CR} \leq 2$.

5.6 Computational requirements

As noted previously, CS-based compression methods result in a multi-faceted trade-off between compression ratio, reconstruction quality, and computational complexity. This section gives further detail on the last aspect, distinguishing what is required at the encoder (that we want to minimize) and at the decoder (that we want to be not worse than the needs of classical recovery methods). In all cases, we refer to the computational burden *per processed sample*, i.e., we divide the number of operations by the number of samples n contained in the processed window.

⁷GAMP has achieved better results compared to the other classical reconstruction algorithms in this setting, i.e., when the sensing matrix is not designed according to the rakesness-based CS.

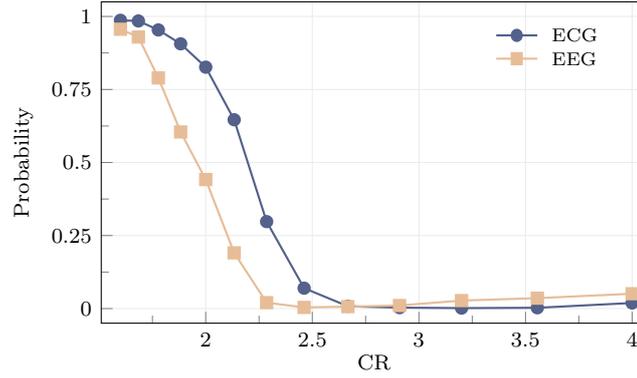


FIGURE 5.9: Probability for GAMP decoder to reconstruct ECG and EEG signals with RSNR higher than TCSSO in case of TCSSO failure.

Encoder The complexity of the encoder is briefly introduced in Section 5.1.1 as one of the leading design criteria. The number of signed accumulations (AC) is $nm = n^2CR^{-1}$ thus yielding nCR^{-1} AC/sample. Further to time-complexity, the memory footprint is dominated by the storage of the matrix A^\pm , and requires a number of entries equal to $nm = n^2CR^{-1}$. In principle, matrix entries are bits. However, microcontroller-based implementations may favour 1-byte-per-entry or even 4-bytes-per-entry solutions. In fact, in some architectures, the alignment of entries at word boundaries ensures better performance both in terms of speed and energy (see, e.g., [126]), this is why we express the memory footprint as the number of entries in A^\pm .

From the blue curves in Figure 5.5, one gets that a higher n results in better reconstruction performance for the same CR, and thus there is a trade-off between encoder complexity and window length.

Table 5.1 reports the comparison between the increase of reconstruction quality, complexity, and memory footprint for ECG signals when n goes from 64 to 128, with CR ranging from 2 to 4. At high CR levels, an increase in terms of ARSNR (e.g., with CR = 4, +18.0 dB for ECG and +11.5 dB for EEG) may be worth the $\times 2$ in terms of computational effort and the $\times 4$ in terms of memory footprint. However, for lower compression ratios, the increase in resource needs is not justified by the limited increase in performance: for CR = 2, memory footprint and complexity increase as before, but one only gains +1.7 dB in the ECG case and +0.1 dB in the EEG case.

Decoder In CS-based schemes, decoding is computationally more intensive than encoding. We may evaluate the complexity of the TCSSO decoder by counting the number of Multiply-and-Accumulate (MAC) operations needed to compute \hat{x} , disregarding the training phase, starting from the fact that the number of MAC operations required in a fully connected layer with n nodes, each with i inputs, is ni .

Neglecting the input layer that has m nodes and that requires no operations, the oracle SOC is composed by 3 fully connected hidden layers with $2n$, $2n$ and n nodes,

TABLE 5.1: Effect of the increase of n from 64 to 128 in terms of AR-SNR, computational overhead measured as AC/sample and memory footprint (# entries of A^\pm).

CR	ARSNR [dB]				AC/sample		# $A_{j,k}^\pm$	
	ECG		EEG		64	128	64	128
	n	64	128	64				
4.00	35.2	+18.0	12.0	+11.5	16	×2	1.0 Ki	×4
3.20	52.4	+5.1	38.8	+6.7	20	×2	1.3 Ki	×4
2.67	56.8	+2.2	55.3	+0.4	24	×2	1.5 Ki	×4
2.29	58.0	+2.4	58.0	+0.2	28	×2	1.8 Ki	×4
2.00	59.2	+1.7	59.7	+0.1	32	×2	2.0 Ki	×4

and a final fully connected output layer with n nodes. Therefore, the number of inputs of these layers is $m, 2n, 2n$ and n , respectively. The layer-by-layer number of MACs required for the forward pass is $2nm, 4n^2, 2n^2$ and n^2 , giving rise to a total of $(2m + 7n)n = (2/\text{CR} + 7)n^2$ MACs for each window thus yielding $(2/\text{CR} + 7)n$ MAC/sample.

After support estimation, additional MACs are needed to compute \hat{x} . In particular, we focus on the computational cost of $\hat{\xi} = B^\dagger y$, with B^\dagger the Moore-Penrose pseudoinverse of $B = A^\pm S|_s$, i.e., of a matrix with m rows and a number c of columns $\kappa \leq c \leq n$, with $c \simeq \kappa$ being the most frequent case. The computational complexity of pseudo-inversion reflects its analytical formula such that $\hat{\xi} = B^\top (BB^\top)^{-1} y$ must be computed. Since B is a $m \times c$ matrix, BB^\top requires $m^2 c$ MACs, and the inversion entails $2m^3$ MACs. Now, the right-multiplication $(BB^\top)^{-1}$ by y costs m^2 MACs and the final left-multiplication results in mc MACs. Considering all contributions, we arrive at estimating a total of $m(2m^2 + m\kappa + m + \kappa)$ for the typical $c = \kappa$ case. The complexity is then equal to $(2n\text{CR}^{-2} + n\kappa/n\text{CR}^{-1} + \kappa/n + \text{CR}^{-1})n\text{CR}^{-1}$ MAC/sample.

We may compare the complexity of TCSSO decoding with that of OMP, which is known to be one of the most light-weighted approaches. We consider the standard implementation of OMP as described in [149]. A modified version (bWOMP) of this algorithm has been proposed in [126] to exploit the same statistical prior described in [198] and improve reconstruction performance with no significant increase of computational complexity. The detailed description of OMP is out of the scope of this paper, and we refer to [149] for details. Knowing that OMP is an iterative algorithm that estimates the signal support in at least κ iteration, we limit ourselves to provide the complexity in terms of the number of MAC for the j -th iteration that is $nm + 2m(j - 1) + 2m + 2jm$. This yields a total of at least $2\kappa m + 2\kappa^2 m + \kappa nm$ MACs. After that, OMP computes the pseudo-inverse of a matrix of the same size as the $B = A^\pm S|_s$ in TCSSO. The total complexity of the iterative part is therefore given by $(2 + 2n\kappa/n + n)n\kappa/n\text{CR}^{-1}$ MAC/sample and must be compared with the computational effort required by the oracle that is $(2/\text{CR} + 7)n$ MAC/sample.

Though the contributions to the computational complexities computed above have different asymptotic behaviours, their magnitude in the small- n cases can be appreciated only by numerical evaluation. As an example, for $n = 64$, $\kappa = 16$ and $CR = 2$ (one of our ECG cases) the first part of OMP entails some 784 MAC/sample, the oracle in TCSSO requires some 512 MAC/sample, while the common pseudo-inversion amounts to 1304 MAC/sample. As a further, somehow opposite, example, for $n = 128$, $\kappa = 26$ and $CR = 4$ (one of our EEG cases) the first part of OMP entails some 1183 MAC/sample, the oracle in TCSSO requires some 960 MAC/sample, while the common pseudo-inversion amounts to 735 MAC/sample.

In both cases, the complexity of TCSSO and OMP are analogous, showing that, though TCSSO allows implementing extremely lightweight encoders, the decoder does not have to compensate by increasing its computational requirements compared to conventional decoders. Consequently, since bWOMP has complexity similar to OMP, at least in the settings we analyzed, the complexity of TCSSO is comparable to the one of decoders that use a statistical prior on the signal support.

Conclusion

Compressed Sensing is a compression scheme designed for devices with limited computational resources. Similarly to PCA-based compression, the encoding stage consists of a simple linear transformation. However, CS can guarantee perfect reconstruction in the noiseless case that satisfies the sparsity and incoherence assumptions. Perfect reconstruction and lightweight encoding come at the price of a complex decoding procedure that traditionally is resolved with iterative algorithms.

This chapter described an alternative decoding procedure that outperforms the state-of-the-art in the case of biosignals such as ECG and EEG and has very limited requirements on computational resources, even lower than properly designed lightweight greedy algorithms such as OMP. The advantage is twofold: CS decoder may be implemented on an edge device, and shorter signal windows may be considered.

The idea behind the proposed decoder resides in splitting CS reconstruction into two stages. First is support divination that finds the positions of the non-null elements of the sparse representation of the input signal corresponding to the measurement vector. Then, pseudo-inversion allows retrieving the amplitudes of the non-null coefficients, which can be linearly transformed into the original signal. This chapter shows that the support oracle can be implemented with a fully connected DNN that can also be trained with the encoding matrix.

The resulting decoder largely outperforms classical approaches (even when the latter are paired with one of the most effective adaptation policies for the encoding matrix) and allows CS to be applied to signal windows containing a limited number of samples. The adoption of short windows is highly beneficial in many directions;

one of the most remarkable is the computational complexity of the encoder. Nevertheless, short windows are usually out of the reach of classical CS mechanisms as the sparsity assumption on which they hinge tends to fail when the dimensionality of the waveform to compress decreases. Hence, our proposal allows the implementation of extremely low complexity encoders that still feature remarkable compression capabilities.

Furthermore, the separation between support guessing and magnitude calculation allows our decoder to detect cases in which the reconstruction may be affected by significant errors, thus paving the way, for example, to additional processing that further increases the reconstruction performance.

Chapter 6

Trained Support Oracle for Compressed signal

This chapter extends the work in Chapter 5 that describe a novel CS decoder based on a DNN-based two-step reconstruction designed for biomedical signals such as ECG and EEG. The extension aims at addressing some of the open problems left.

As a first aspect, the work in Chapter 5 refers to exactly sparse signals that correspond to an unrealistic scenario. For this reason, here, the signal model is generalized to *compressible* signals, i.e., signals that are non-perfectly sparse in the sense that only a few elements have a significant magnitude and the others are negligible but have magnitude higher than zero. We address this generalization by introducing a new definition of support that considers the non-perfect sparsity of the signal and possible sources of noise.

The second aspect concerns the implementation. The two-stage decoder in Chapter 5 has already a computational burden that is comparable to the most lightweight methods present in the Literature, but here a further step is taken. Indeed, this chapter focuses on limiting the performance degradation in the case of fixed-point arithmetic and optimizing the memory footprint required by both DNN and pseudo-inversion. We reach these goals by training the DNN-based support oracle with quantization-aware techniques and replacing the Moore-Penrose pseudo-inversion with a Least Mean Square (LMS) filter.

6.1 Compressed Sensing with Support Oracle

Similarly to Chapter 5 we consider an n -dimensional input signal $x \in \mathbb{R}^n$, that when expressed on a proper sparsity basis S , it can be represented as $x = S\zeta$, where the coefficient vector $\zeta \in \mathbb{R}^n$ contains at most $\kappa \ll n$ non-zero entries, i.e., x is κ -sparse. Let us also consider the support $s \in \{0, 1\}^n$ of ζ such that $s_j = 1$ if $\zeta_j \neq 0$ and $s_j = 0$ otherwise.

In the case of sparse signals, each instance x depends only on a number of scalars that is much smaller than n . This prior is used to define an encoder procedure that compresses x by applying a linear operator $\text{Enc} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that is modelled with a sensing matrix $A \in \mathbb{R}^{m \times n}$ with $m < n$. The encoder output is a m -dimensional

measurement vector y obtained by projecting x over the rows of A . Considering non-idealities in the system implementation as term v that is added to the signal input encoding is modelled as

$$y = \text{Enc}(x + v) = A(x + v) \quad (6.1)$$

The decoder aims at recovering the sparse representation of the input signal ζ , or at least its best approximation $\hat{\zeta}$, by leveraging on the sparsity prior. The standard BPDN approach [46, 48] consists of finding the sparsest n -dimensional vector ζ among the infinite solutions of the ill-defined system $y = AS\zeta$ by considering the optimization problem in (5.1). The estimate $\hat{\zeta}$ is then used to reconstruct the input as $\hat{x} = S\hat{\zeta}$.

Reconstruction is possible when the number of measurements m is sufficient and, intuitively, this number is related to the value of κ . CS theory identifies this relationship as $m = \mathcal{O}(\kappa \log(n/\kappa))$ [46], and in practical cases m is often chosen proportional to κ . However, such worst-case theoretical guarantees fail for $m < 2\kappa$ since, when this happens, two κ -sparse signals with non-overlapping supports can be potentially mapped in the same measurement vector.

In a CS framework, requirements are not limited to a minimum number of measurements but also include the proper design of the rows of A . Most notably, if the elements of a generic sensing matrix row a are drawn as instances of independent and identical distributed (i.i.d.) random variables with zero-mean and unit-variance Gaussian distribution, then ζ can be recovered from y [61, 45, 122] with very high probability. Reconstruction reaches the same level of quality even if the elements of a are instances of i.i.d. antipodal (i.e. +1 or -1) random variables [85, 122]. As in Chapter 5, due to the obvious great implementation advantages of the latter choice, we focus on this class of sensing sequences.

This agnostic and general approach can then be specialized to a suitable class of signals in many ways by adopting an adapted CS approach [121]. Among the several approaches that, thanks to adaptation, guarantee better performance with respect to agnostic CS, state of the art for the design of antipodal sensing matrices is the rakeness-based CS (Rak-CS) [118, 120, 121].

The Rak-CS approach models the sensing sequences not as instances of i.i.d. variables but as a stochastic process whose correlation matrix $\mathcal{A} = \mathbf{E}[aa^\top]$ is obtained as:

$$\mathcal{A} = \frac{1}{2} \frac{n\mathcal{X}}{\text{tr}(\mathcal{X})} + \frac{1}{2} I_n \quad (6.2)$$

where $\mathcal{X} = \mathbf{E}[xx^\top]$ is the correlation matrix of the stochastic process generating input instances. Roughly speaking, the statistical adaptation of the sensing matrix proposed by Rak-CS is a middle ground between standard CS theory (that suggests i.i.d. based sensing, and for which it is $\mathcal{A} = I_n$) and an over-adapted setting where $\mathcal{A} = \mathcal{X}$. Such an approach has been proved to guarantee good performance also in case of uncommon instances [150].

Let us now consider the operator $\cdot|_s$ that, when applied to a n -dimensional vector, selects only the elements corresponding to non-null entries of s , while, when applied to n -column matrices, returns the submatrix composed of the columns whose index corresponds to $s_j = 1$. Then, any κ -sparse signal x can be represented by the n -dimensional binary vector s and by $\zeta|_s$, a non-sparse vector that contains no more than κ real values. As deeply discussed in Chapter 5 and firstly proposed in [119], this notation paves the way to a completely different decoder approach, composed of two consecutive blocks. The first one, described in 5.3 as *support oracle* (SO), is devoted to identifying the support and is capable of divining s by looking at the vector y . Then, assuming s is known, by defining $B = AS$, we can observe that

$$y = A(S\zeta + v) = B\zeta + Av = B|_s\zeta|_s + Av. \quad (6.3)$$

The fact that $\kappa < m$ makes $B|_s$ a *tall* matrix (the number of rows exceeds the number of columns) such that each measurement vector y , ignoring the noise v , possesses a unique counterimage given by $\zeta|_s = B|_s^\dagger y$.

In other words, given (6.3), to recover the input signal, it is enough a second stage computing

$$\hat{\zeta}|_s = B|_s^\dagger (y - Av) = B|_s^\dagger y - B|_s^\dagger Av \quad (6.4)$$

that is a much simpler operation with respect to any CS recovery algorithm. The term $B|_s^\dagger Av$ defines the signal recovery error in the sparse representation.

6.2 Support Oracle for Compressed Signals

As a further critical remark, one can note that almost all classes of real signals are only *approximately sparse* since the vector $\zeta = S^\top x$ is composed of a few entries with magnitude significantly greater than zero while the remaining entries are close to zero. In these cases, signals are not sparse but *compressible*.

As a result, it is impossible to define a support for x by looking at the vector ζ since any possible definition would cause rejection of a part of the signal information content. Let us therefore indicate with z a n -size binary vector such that $x = S|_z\zeta|_z + x_d$, where x_d contains signal details of minor interest. With this notation, (6.4) can be reformulated as

$$\hat{\zeta}|_z = B|_z^\dagger [y - A(x_d + v)] = B|_z^\dagger y - B|_z^\dagger A(x_d + v) \quad (6.5)$$

where the reconstructed signal is now $\hat{x} = S|_z\hat{\zeta}|_z$ and the reconstruction error in the sparse representation is $B|_z^\dagger A(x_d + v)$.

The choice of the support z is fundamental in limiting the error since, along with A and v , it defines the maximum achievable performance in terms of RSNR. Increasing the number of ones in z reduces the error contribution due to x_d . On the other hand, reducing the number of ones in z reduces the reconstruction error due to v .

The optimal support z^* that balances this trade-off can be obtained, in principle, by computing the RSNR for all possible z . Unfortunately, finding z^* implies the solution of a combinatorial optimization problem (6.6) that requires an exhaustive search over all possible 2^n binary z vectors.

$$z^* = \arg \max_{z \in \{0,1\}^n} \frac{\|x\|_2}{\|x - S_{|z}\hat{\zeta}|_z\|_2} \quad (6.6)$$

We overcome this impasse by proposing a greedy approach providing a good approximation of z^* with complexity as low as n steps. First, the entries of ζ are sorted in decreasing order according to their magnitude. Following this order, the z vector, initialized with all zeros, is built iteratively by adding, step by step, a new non-zero element. At each step, the RSNR is computed, and the support z is identified as the vector that achieves the highest RSNR value. Note that the support z that we estimate is the support extension applied to a compressible signal once compressed by the CS encoder and depends on the sensing matrix A and the noise ν . We refer to z as the *Support of Compressed signal*, and we name the CS decoder based on the pre-computation of z as *Support Oracle for Compressed signals* (SOC).

To validate this approach, a dataset of synthetic ECGs generated according to [131] as described in Section 5.2 and in [119, 124] has been considered. However, we here consider the generated ECG as a compressible signal, i.e., we do not impose x_d to be null. Synthetic ECG instances are generated with sample rate 256 sample/s and time windows composed of $n = 128$ successive samples. The signal is then corrupted by additive Gaussian noise so that the intrinsic signal-to-noise ratio (ISNR) is set to 34 dB [205].

Figure 6.1 plots the RSNR as a function of the number of ones in z for an instance of x and ν . The figure shows how both the maximum RSNR and the corresponding z cardinality depend on the number m of rows of A . The figure also highlights the already observed trade-off. When the cardinality of z is low, each new element inserted in the support is associated with a signal component with a large magnitude. This insertion increases the RSNR because the projection of x along the corresponding column of S certainly exceeds in magnitude the projection of ν . Conversely, when the cardinality of z is high, the additional signal information content brought by the new column of S may be smaller than the corresponding noise contribution. Moreover, the RSNR values depend on the adopted sensing matrix since the higher the compression ratio, the more complex the reconstruction.

For each profile in Figure 6.1, the highlighted point represents the number of ones in z that maximizes the RSNR. Hence, according to the definition of z and the greedy method we propose, this point corresponds to the support z for a specific compressible signal instance x , a noise vector ν , and a sensing matrix A .

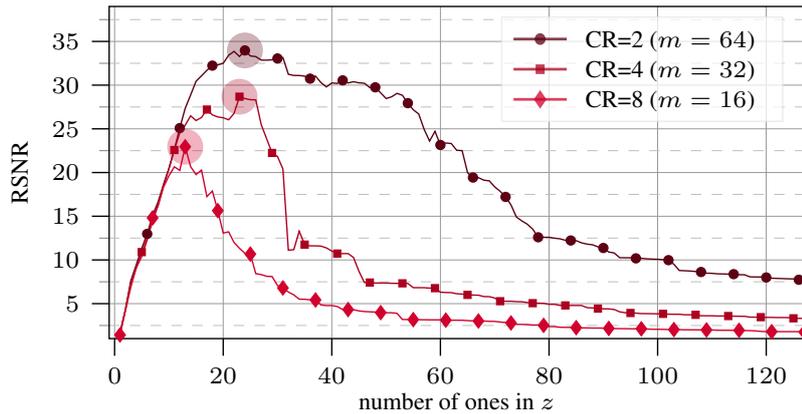


FIGURE 6.1: RSNR against the number of ones in z for different sensing matrices with $\text{ISNR} = 34$ dB.

6.2.1 Trained Support Oracle for Compressed Signals

A support oracle decoder for compressed signals assumes that retrieving the signal support z from observing the measurements vector y is possible. In Chapter 5 and [119], it was shown that it is possible to retrieve the support s of a sparse signal by mean of a Deep Neural Network (DNN) that solves a multi-class classification task in which each one of the n output elements o_j corresponds to the probability for s_j to be 1. Taking inspiration from it, we propose a DNN that deals with compressible signals to divine z .

The structure of the network is reported in Figure 6.2, where it is shown that the m -dimensional input layer receives the measurement vector. Then, there are three fully connected hidden layers with $2n$, $2n$ and n neurons, respectively and Rectified Linear Unit (ReLU) as activation functions. Finally, a fully connected output layer with n neurons and sigmoid activation functions produces a vector o with entries in $[0, 1]$. The final estimated support \hat{z} is obtained by applying a threshold $\theta \in [0, 1]$ to o such that $\hat{z}_j = 1$ if $o_j \geq \theta$, and $\hat{z}_j = 0$ otherwise. The CS decoder that adopts this DNN to divine z is named *Trained Support Oracle for Compressed signal* (TSOC).

Compared to the DNN discussed Chapter 5, the matrix A characterizing the encoder stage is not trained with the support oracle since the labels used during the training, i.e., the supports z , also depend on the sensing matrix. In light of that, the training exploits measurement vectors y computed with a fixed sensing matrix A , which is still an antipodal matrix generated according to the Rak-CS approach for the considered class of signals. Note that A plays the role of a set of hyperparameters that therefore cannot be trained with the parameters characterizing the neural network. In addition, A influences the network architecture since a different number of rows m of the sensing matrix corresponds to a different number of neurons in the input layer.

The DNN parameters set W (including weights and biases for each layer) are trained with a dataset of 2×10^6 signal instances x split in 95% for the actual training and 5% as a test set for performance assessment. For each different matrix A , DNN

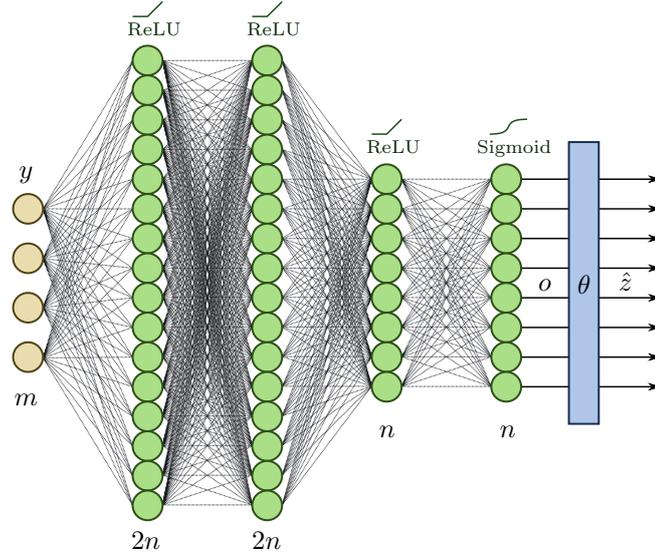


FIGURE 6.2: Structure of the DNN implementing the support oracle.

input-output pairs (y, z) are obtained from both vectors x and randomly drawn noise contributions v such that $y = A(x + v)$. For each value of m , we generate 100 different random candidates A according to the Rak-CS framework. Rak-CS (6.2) requires the signal correlation matrix \mathcal{X} which is estimated from 5000 signal instances generated especially for this purpose. Among these candidates A , the matrix chosen for the encoder stage of the TSOC is the one obtaining the best ARSNR over 1000 signal reconstructions when a BPDN decoder is used¹.

Since m corresponds to the number of rows in A and the number of neurons in the input layer, the parameters characterizing the first hidden layer are $2nm$ weights and $2n$ biases. For the case $n = 128$, Table 6.1 reports the total number of parameters for the adopted DNN settings where m ranges from 16 to 48.

Each of the proposed models is implemented in the TensorFlow framework [1], and the loss function is minimized using stochastic gradient descent with a batch size of 50 instances over 500 epochs and an initial learning rate value equal to 0.1. Considering a generic example comprising a pair (y, z) , the loss function is the binary cross-entropy between z and the output of the DNN o

$$L(y, z) = -\frac{1}{n} \sum_{j=1}^n [z_j \log(o_j) + (1 - z_j) \log(1 - o_j)] \quad (6.7)$$

Finally, 5000 new instances are used to tune the threshold θ applied to the DNN output o . The resulting values of θ in all the considered settings are close to the middle-range value 0.5, i.e., values in o vectors concentrate close to the two boundary values, zero and one.

¹The Spectral Projected Gradient for L1 minimization (SPGL1) toolbox [26] is employed to implement BPDN.

6.2.2 Experimental results for ECGs

To assess the performance of the neural network architecture, we consider different CS settings, each of which considers a matrix A with a different value for m that ranges from $m = 16$ (CR = 4) to $m = 48$ (CR = 2.7). Therefore, for each setting, we train a different set of parameters. As anticipated in the previous section, the task of the DNN is equivalent to a multi-label classification [158] that considers each label as an independent binary classification problem. More precisely, for each input y , the DNN produces n outputs that are independently either positive or negative. If the j -th output is positive $\hat{z}_j = 1$, otherwise $\hat{z}_j = 0$.

To assess the capability of correctly estimating z , metrics related to the difference between z and \hat{z} need to be adopted. Let us first introduce the metrics for a single binary classification problem and then generalize for the case of n binary classifications performed by the DNN.

If both entries z_j and \hat{z}_j are equal to 1, we mark this classification as a single true positive, while $z_j = \hat{z}_j = 0$ is a single true negative. In case of single misclassifications we have either a single false positive ($\hat{z}_j = 1$ and $z_j = 0$) or a single false negative ($\hat{z}_j = 0$ and $z_j = 1$). Finally, 5000 new instances are used to tune the threshold θ applied to the DNN output o . The resulting values of θ in all the considered settings are close to the middle-range value 0.5, i.e., values in o vectors concentrate close to the two boundary values, zero and one. Hence, the performance of a prediction can be expressed in terms of the following quantities:

- Positive (P) and Negative (N):

$$P = \sum_{j=1}^n \hat{z}_j, \quad N = \sum_{j=1}^n (1 - \hat{z}_j) = n - P \quad (6.8)$$

- True Positive (TP) and True Negative (TN)

$$TP = \sum_{j=1}^n z_j \hat{z}_j, \quad TN = \sum_{j=1}^n (1 - z_j)(1 - \hat{z}_j) \quad (6.9)$$

- TP Rate (TPR), TN Rate (TNR) and Accuracy (ACC)

$$TPR = \frac{TP}{P}, \quad TNR = \frac{TN}{N}, \quad ACC = \frac{1}{n}(TP + TN) \quad (6.10)$$

A summary of the average values for P, TP, TPR, TNR and ACC over the whole test set can be found in Table 6.1 to show the ability of the network to correctly detects the ones in z .

The results in Table 6.1 show that the number of ones in \hat{z} increases with m , confirming the behaviours of a single instance reported in Figure 6.1. The difference between P and TP, i.e., the number of ones wrongly estimated, is roughly constant and less than 2. Consequently, the TPR tends to increase with m while the TNR

TABLE 6.1: Number of parameters of TSOC for different values of m and its performance in terms of P, TP, TPR, TNR and ACC estimated as the mean over the data set.

m	# param	P	TP	TPR	TNR	ACC
16	119 552	14.4	12.5	0.885	0.997	0.983
20	120 576	17.1	15.4	0.908	0.996	0.983
24	121 600	20.3	18.6	0.926	0.994	0.982
28	122 624	23.2	21.6	0.933	0.992	0.981
32	123 648	25.3	24.0	0.949	0.990	0.981
36	124 672	27.0	25.4	0.945	0.989	0.979
40	125 696	28.5	26.9	0.947	0.988	0.978
44	126 720	29.6	28.0	0.951	0.985	0.976
48	127 744	30.3	28.5	0.945	0.985	0.974

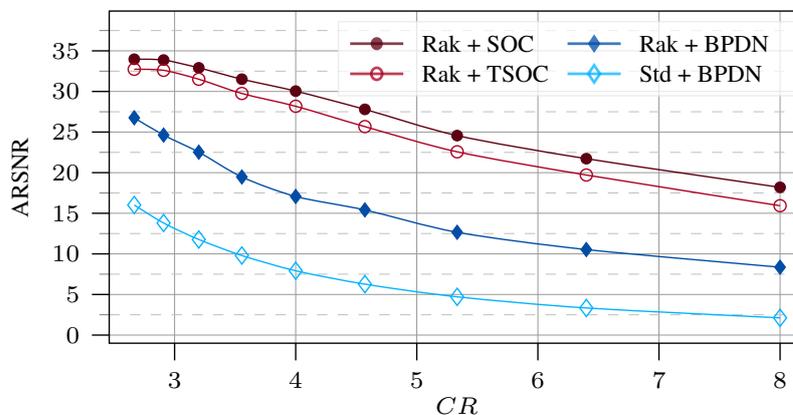


FIGURE 6.3: ARSNR against CR for TOSC along with the ideal oracle (SOC) compared with the standard decoder (BPDN). The encoder stage follows Rak-CS and the standard CS encoder coupled with BPDN is provided as a reference.

slightly decreases. Since in general \hat{z} contains more zeros than ones, the accuracy is dominated by TN so it slightly decreases with m , starting from 0.983 for $m = 16$ to 0.974 for $m = 48$.

Accuracy values very close to one ensure a high overlap between z and \hat{z} . Nevertheless, even when $z = \hat{z}$, the decoder still commits an error in the reconstruction as modelled in (6.5). Thus, the overall performance of TSOC must be evaluated in terms of either ARSNR or RSNR distribution.

Figure 6.3 compares the performance in terms of the achieved ARSNR of the proposed TSOC approach with that of SOC (ideal oracle) and BPDN (SPGL1 decoder). All these approaches share the same tuned Rak-CS antipodal sensing matrices A . Further comparison shows BPDN approach coupled with matrices A following the standard CS theory (Std) where -1 and +1 occur with the same probability. Rak + TSOC outperforms Rak + BPDN with a gap of at least 5 dB, while the loss with respect to the ideal oracle (Rak + SOC) never exceeds 2.5 dB. Std + BPDN performance is not even comparable with the ones of the other frameworks.

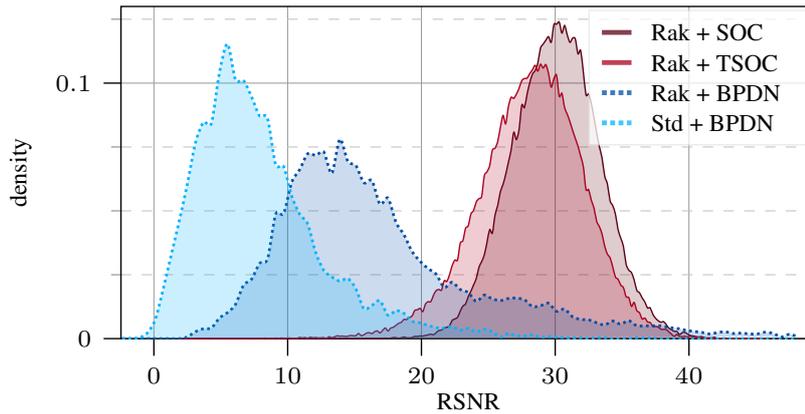


FIGURE 6.4: Probability density functions of RSNR values with CR=4 ($m = 32$) for the considered system configurations.

To provide a further comparison between these approaches, Figure 6.4 shows the RSNR distributions in case of CR = 4 ($m = 32$). The proposed TSOC, along with the ideal SOC, shows an RSNR variance that does not increase compared to the already presented Std + BPDN and Rak + BPDN.

6.3 Quantization-aware Decoder Architecture

This section investigates the implementation of the TSOC-based system in the presence of possible hardware limitations, e.g., a limited precision arithmetic unit. The block scheme of the overall system is depicted in Figure 6.5, in which the bits of each digital signal is highlighted. The decoder is composed of two main blocks: *i*) the support oracle divining z , *ii*) the reconstructor that uses the oracle output to recover the original waveform.

The representation of each system quantity with a finite number of bits addresses a trade-off between computational burden/memory footprint and reconstruction performance. A preliminary investigation on this direction is reported in [154] where authors study the performance loss due to the parameter post-quantization with the two-stage decoder proposed in [119]. They first design the two blocks composing the decoder with full precision, and then simply quantize the entries of both B and S and the DNN parameters.

Here, different strategies are considered to limit the performance loss by including quantization-aware techniques and a different approach in the pseudoinverse operation and the quantization of the measurement vector y .

The first issue we address is the quantization of the input of the decoder y . y is assumed to be quantized by a mid-tread uniform quantizer, with 2^{b_y} levels. Quantization may come either from quantizing a measurement vector computed by an analog CS encoder block or the digital processing of a digital input signal x . In this setting, we need to remember that if n is large enough, each y_i , $i = 1, \dots, m$ can be considered as a zero-mean Gaussian distributed random variable, and setting a

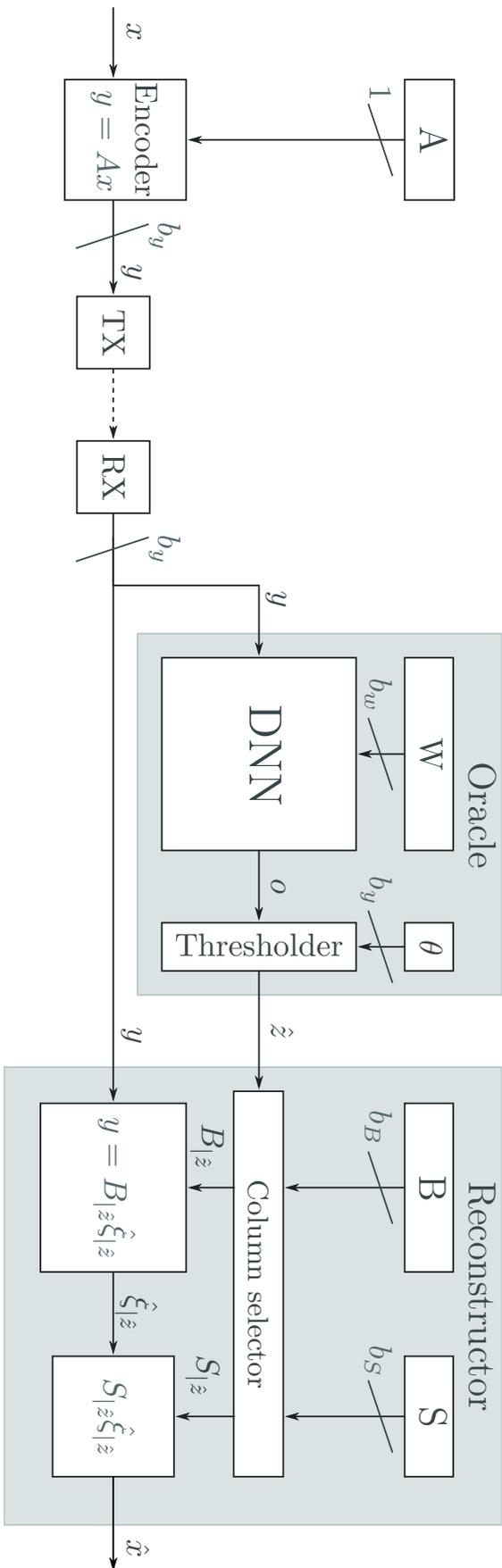


FIGURE 6.5: Block diagram of the overall system including CS encoder, measurements dispatch and the proposed TSOC decoder. The arrows represent digital quantities and for each of them the number of bits is highlighted.

conversion range that includes all possible values is not possible, or simply not convenient [123, 150].

Being Δ the quantization step, σ_y^2 the variance of the distribution of each value in y , and γ a positive coefficient, we set $2^{b_y}\Delta = 2\gamma\sigma_y$, so that each entry of y is represented with b_y bits and ranges from $y_{\min} = -2^{b_y-1}\Delta = -\gamma\sigma_y$ to $y_{\max} = (2^{b_y-1} - 1)\Delta \approx \gamma\sigma_y$. After quantization, each element of y can be considered a fixed point number belonging to the set $\Omega_y = \{-1, -1 + \Delta_y, \dots, 1 - \Delta_y\}$, with $\Delta_y = \Delta/(2\gamma\sigma_y^2)$.

The quantized values of y feed both the support oracle and the reconstructor. Each of the two structures internally uses parameters that can be quantized to reduce the complexity and memory footprint.

6.3.1 Support oracle quantization

As a first modification to reduce the computational complexity of our DNN structure, we replace the sigmoid function in the output layer with a linear function. Since the sigmoid is monotone and the output is thresholded, it is sufficient to adapt the threshold value θ .

Then, the oracle parameters are quantized to minimize the memory footprint and reduce the resources needed for the support divination. All parameters are encoded with b_w bits and constrained to be in the discrete set $\Omega_w = \{-1, -1 + \Delta_w, \dots, 1 - \Delta_w\}$, where Δ_w is the quantization step chosen be compliant with the adopted fixed-point representation.

Moreover, in the hidden layers, all the neurons outputs (activations) are represented with only the b_y most significant bits such that their representation is coherent with the one of the DNN input. As a result, since both activations and parameters are represented in fixed-point, integer arithmetic is sufficient to produce the oracle output. The inputs are multiplied by the weights in each neuron and then summed. Therefore, the number of bits required by the arithmetic unit is $b_y + b_w + \log_2(n_L + 1)$ where n_L is the number of neuron inputs that in our case never exceeds $2n$.

Regarding the parameters quantization, a possible choice is reported in [154] where quantization is applied at the end of the training. However, it is possible to adopt strategies during the training that help to reduce the performance loss due to quantization. Here, we investigate some approaches that we group by task:

- limiting the parameters in the set Ω_w to avoid clipping in quantization;
- limiting the activations in the set Ω_y to avoid overflow during inference;
- updating parameters with approaches that accounts for the effect of the quantization (quantization-aware training).

Limiting the parameters range We force the parameters to assume values in the set Ω_w with the combination of two methods: “bathtub” regularization and parameter “recycling”.

The “bathtub” regularization consists of a regularization term that is added to the cost function (6.7) and penalizes the parameters that are outside the desired range. We can define the “bathtub” regularization function with its derivative:

$$\frac{\partial R_{\text{bathtub}}(w_{l,i})}{\partial w_{l,i}} = \begin{cases} -1 & \text{for } w_{l,i} < -1 \\ 0 & \text{for } -1 \leq w_{l,i} \leq 1 - \Delta_w \\ 1 & \text{for } w_{l,i} > 1 - \Delta_w \end{cases} \quad (6.11)$$

where $w_{l,i}$ is the i -th parameter in the l -th DNN layer. This regularization term affects the cost gradient used to update the parameters in the back-propagation algorithm, and its effect consists of pushing the parameter’s value inside the desired set.

The parameter “recycling” is still applied in training but only at the beginning of each epoch. The parameters with values outside the desired set are replaced with a new value randomly chosen in Ω_w with uniform probability.

Bounding activations Since input and activations are in fixed-point representation, we want to force each neuron to produce an output that assumes values in the set Ω_y . For this reason, we replace the ReLU with the Saturated ReLU (SReLU) as the activation function in all hidden layers. Given the weighted sum of the neuron inputs v , SReLU is defined as follows:

$$\text{SReLU}(v) = \begin{cases} 0 & \text{for } v < 0 \\ v & \text{for } 0 \leq v < 1 \\ 1 & \text{for } v \geq 1 \end{cases} \quad (6.12)$$

Quantization-aware training Among the many solutions proposed in the literature, we investigate two techniques, namely fake-quantization [152] and cosine regularization [172]. Fake-quantization [152] suggests training the network with full precision parameters and adopting quantized values only during the feed-forward phase. This procedure has the effect of emulating the loss of precision due to quantization. Cosine regularization [172] is a further regularization term $R_{\text{cosine}}(W)$ added to (6.7) and defined as follows:

$$R_{\text{cosine}}(W) = -\frac{1}{2^{b_w}} \sum_{l=1}^{\ell} \sum_{i=1}^{N_l} \cos(2^{b_w} w'_{l,i} \pi) \quad (6.13)$$

where ℓ is the number of layers composing the DNN, N_l is the number of parameter in the l -th layer, and $\hat{w}_{l,i}$ is $w_{l,i}$ constrained to the set Ω_w . Consequently, the parameters are pushed near the 2^{b_w} values allowed by the quantizer and, therefore, the quantization error is reduced. A visual representation of the combination of the cosine and bathtub regularization is shown in Figure 6.6.

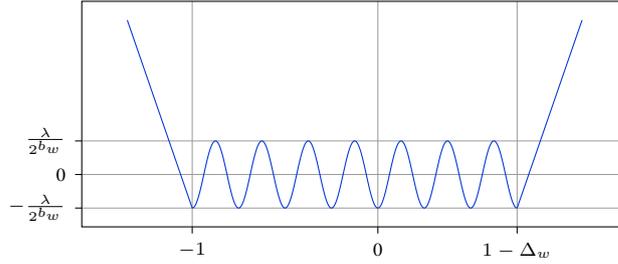


FIGURE 6.6: Combination of the bathtub and cosine regularization terms.

6.3.2 LMS-based reconstruction

The final stage retrieves the vector $\zeta_{|\hat{z}}$ by solving the equation $y = B_{|\hat{z}}\zeta_{|\hat{z}}$. Note that this is a resource-hungry operation and numerically critical as it requires a matrix inversion and several square roots operations. As a consequence, devices embedding a floating-point unit are preferred.

The first step adopted to reduce the reconstruction complexity is quantizing matrices S and B with b_S and b_B bits. As a result, the memory footprint required for storing them is significantly reduced.

Moreover, since pseudo-inversion fundamentally solves a Least Mean Squares (LMS) problem, it is possible to compute $\zeta_{|\hat{z}}$ as the coefficients of a 1-st order LMS filter [184], [89, Chapter 6]. In detail, the rows of $B_{|\hat{z}}$ act as input vectors, y as the desired response, and the elements of $\hat{\zeta}_{|\hat{z}}$ as the coefficients of the LMS filter to estimate. Then, the t -th step of the LMS algorithm can be summarize as

$$\hat{y}^{(t)} = B_{|\hat{z}}\hat{\zeta}_{|\hat{z}}^{(t-1)} \quad (6.14)$$

$$e^{(t)} = y - \hat{y}^{(t)} \quad (6.15)$$

$$\hat{\zeta}_{|\hat{z}}^{(t)} = \hat{\zeta}_{|\hat{z}}^{(t-1)} + \eta(e^{(t)})^\top B_{|\hat{z}} \quad (6.16)$$

where the initial value $\hat{\zeta}_{|\hat{z}}^{(0)}$ is a null vector, $\hat{y}^{(t)}$ is the filter output and $e^{(t)}$ is the estimation error at step t , and $\eta \ll 1$ is the learning rate. The update step is repeated for q times before assuming convergence is reached.

Since LMS filter employs only additions and multiplications, fixed-point arithmetic is may be adopted for a more efficient implementation.

6.3.3 Architecture design and results

In this section, we first tune the hyper-parameters characterizing the decoder, and then the performance is assessed with a comparison with the state-of-the-art CS framework and including standard CS as a reference.

Tuning of the hyper-parameters Table 6.2 summarizes the hyper-parameters that need to be tuned for implementation of the decoder. We focus on the case $CR =$

TABLE 6.2: Hyper-parameters for the low-resources TSOC implementation

Decoder input	
b_y	Number of bits used to represent measurements y
$\gamma\sigma_y$	Half range of measurements y
Oracle	
b_w	Number of bits used to represent DNN weights
λ	Strength of cosine regularization
Reconstructor	
b_S	Number of bits used to store matrices S
b_B	Number of bits used to store matrices $B = AS$
q	Iterations of the LMS filter
η	LMS filter learning rate

4 and split the tuning into two phases: determining the parameter of the LMS-based reconstructor considering an ideal support oracle, then choosing the hyper-parameters of the DNN that implements the support oracle.

For the LMS setting, we first perform a series of tests with 5000 signal instances, imposing $\hat{z} = z$, i.e., replacing the DNN with the ideal oracle. For γ , b_B and η , it is possible to find a values that maximize the ARSNR that are $\gamma = 4$, $b_B = 9$, $\eta = 2^{-6}$. Conversely, q , b_S and b_y exhibit profiles that saturate so we select the lowest values below which the performance starts to degrade significantly. We select Figure 6.7 show their trends and the chosen values that are $q = 512$, $b_S = 10$ and $b_y = 10$.

Once the LMS-based reconstructor is set, the hyper-parameters of the DNN implementing the support oracle are tuned. The tuning considers λ and b_w and is performed on a validation set. Even considering different values for λ , the performance is almost constant when b_w is equal or greater than 4, and it significantly degrades only for b_w lower than 4. For this reason, we set $b_w = 4$, and for that value, the optimal λ results to be 10^{-8} .

Note that, considering $b_w = 4$ the DNN can be stored in 495 kbit, while 37 kbit are dedicated for B ($b_B = 9$ bit and 4096 entries), and 164 kbit for S ($b_S = 10$ bit and 16 384 entries). The overall memory footprint of the decoder is therefore 695 kbit, to which we must add 4 kbit needed for the 32×128 antipodal sensing matrix A in the encoder.

Performance assessment We finally test the whole quantized architecture, with quantized y , W , B and S , along with either the pseudo-inversion on a floating-point unit (FPU) or the LMS filter approach with fixed-point arithmetic. The test set counts 9×10^4 ECG windows with the same setup described in Section 6.2. The distribution of the RSNR can be found in Figure 6.8, along with the results presented in Section 6.2 for the TSOC case (no quantization). The main three cases to consider are

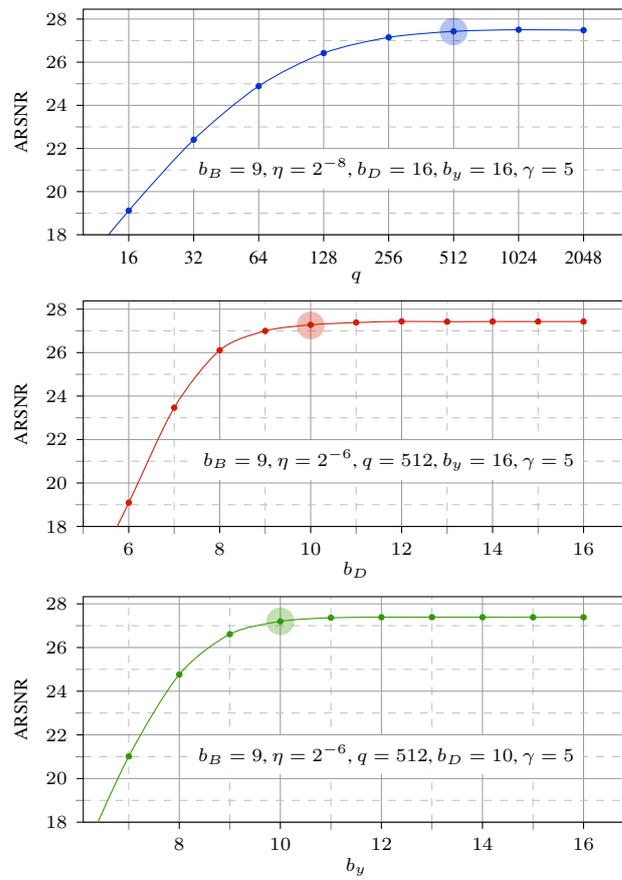


FIGURE 6.7: Performance of the decoder with ideal support oracle for different values of the hyper-parameters q , b_S and b_y .

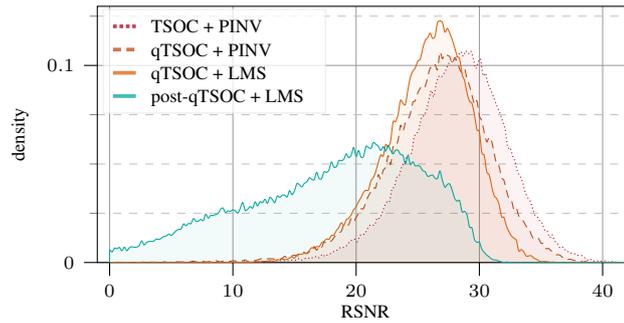


FIGURE 6.8: Probability density functions of RSNR values with $CR = 4$ ($m = 32$) for the floating-point TSOC and the quantized TSOC (qTSOC) with either the pseudo-inversion (PINV) or the LMS filter as well as the case of post-quantization (post-qTSOC).

therefore the ideal (i.e. high-precision) TSOC with pseudo-inversion on FPU (PINV) and the two “constrained” cases in which TSOC is quantized (qTSOC) with either PINV or the LMS filter.

As one can observe in Figure 6.8, the quantization of the support oracle (qTSOC+PINV) reduces the ARSNR from 28.2 dB to 26.6 dB, and the adoption of the LMS filter for the reconstruction (qTSOC+LMS) further decreases ARSNR to 25.9 dB. The variances of the three distributions do not change significantly. As a result, the fully fixed-point implementation suffers from a loss in performance quantifiable in 2.3 dB. As an additional reference, Figure 6.8 reports the distribution of ARSNR in case of post-quantization (post-qTSOC), i.e., the case in which the DNN parameters are trained with no dedicated techniques. Here, ARSNR drops to 17.7 dB.

Finally, Figure 6.9 shows some examples of reconstruction of ECG windows with qTSOC+LMS compared to the original instances.

Conclusion

This chapter extends the work reported in Chapter 5 that introduced a two-stage decoder for sparse biomedical signals. The extension generalizes the signal model from perfectly sparse to compressible and targets an actual edge device deployment.

In detail, we define a support for the compressible signal that affects the functionalities of the support oracle. Since with this new definition, support of a signal depends not only on the signal itself but also on the encoding matrix, the DNN representing the oracle’s core must be trained separately from the sensing matrix, allowing the adoption of independent strategies for sensing matrix adaptation, such as the state-of-the-art Rakeness approach. The resulting two-stage decoder TSOC is assessed on the task of reconstruction of synthetic ECG signals confirming the preliminary results shown in Chapter 5 that depicted the two-stage decoder outperforming other state-of-the-art approaches.

We have also investigated the implementation of TSOC on real hardware. For this reason, we focus on fixed-point arithmetic that requires the quantization of the

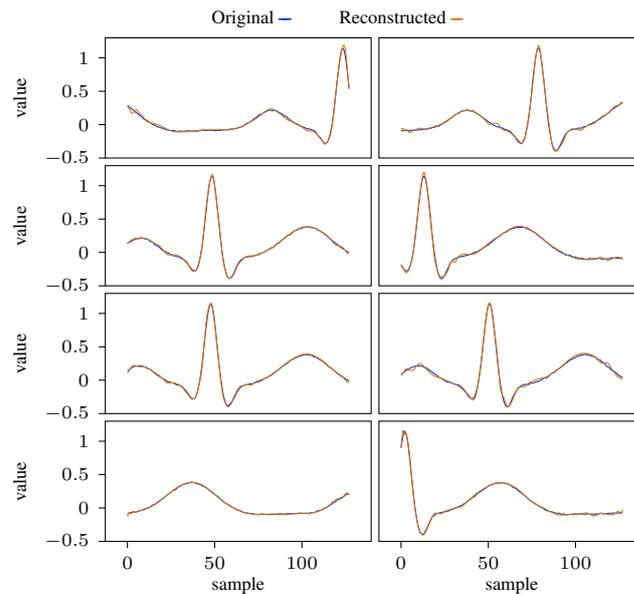


FIGURE 6.9: Examples of windows of ECG signals reconstructed employing qTSOC+LMS with CR = 4 ($m = 32$) compared with the originals.

quantities involved. Thanks to quantization-aware techniques, the DNN parameters can be quantized down to 4 bit with just a slight decrease of the performances. Moreover, approximating pseudo-inversion with an iterative LMS filter numerical stability and robustness to numerical error is also achieved with fixed-point arithmetic.

Nevertheless, the adoption of a DNN in a CS decoder introduces two possible limitations with respect to traditional approaches: *i*) the need for a sufficiently large data set for the neural network training; *ii*) the need for an entire training in case of a change in the sensing matrix.

Appendix A

Proof of Properties

A.1 Proof of Property 1.1

Proof. The encoder is tuned to the normal behaviour which consider a memory-less source, i.e., each generated vector is independent from the previous ones. For this reason, time indications is dropped so that one may focus on a vector x^{ok} with independent components $x_j^{\text{ok}} \sim \mathcal{N}(0, \lambda_j^{\text{ok}})$ for $j = 1, \dots, n$.

Given a value of the parameter θ , the encoder transforms each component x_j^{ok} into \hat{x}_j^{ok} separately [104], such that:

$$\hat{x}_j^{\text{ok}} = \begin{cases} 0 & \text{if } \lambda_j^{\text{ok}} \leq \theta \\ x_j^{\text{ok}} + \Delta_j & \text{if } \lambda_j^{\text{ok}} > \theta \end{cases} \quad (\text{A.1})$$

where, to achieve the Shannon lower bound, Δ_j must be an instance of a Gaussian random variable independent of \hat{x}_j^{ok} . As a consequence, the three quantities \hat{x}_j^{ok} , x_j^{ok} and Δ_j must be jointly Gaussian such that $(\hat{x}_j^{\text{ok}}, x_j^{\text{ok}}, \Delta_j)^\top \sim \mathcal{N}(0, \Sigma_{\hat{x}_j^{\text{ok}}, x_j^{\text{ok}}, \Delta_j})$ with

$$\Sigma_{\hat{x}_j^{\text{ok}}, x_j^{\text{ok}}, \Delta_j} = \begin{pmatrix} \lambda_j^{\text{ok}} - \theta & \lambda_j^{\text{ok}} - \theta & 0 \\ \lambda_j^{\text{ok}} - \theta & \lambda_j^{\text{ok}} & -\theta \\ 0 & -\theta & \theta \end{pmatrix} \quad (\text{A.2})$$

Since the covariance between \hat{x}_j^{ok} and x_j^{ok} is represented by the non-diagonal elements $\lambda_j^{\text{ok}} - \theta$, which are positive, this explain that \hat{x}_j^{ok} encodes x_j^{ok} in the sense that are positively correlated with a level that depends on the distortion.

Generalizing the Gaussian with null variance with a Dirac's delta, one may write that $\hat{x}_j^{\text{ok}} \sim \mathcal{N}(0, \max\{0, \lambda_j^{\text{ok}} - \theta\})$ and, by joining the pdf of each component into a vecotr pdf, one may obtain the pdf of the optimally distorted signal as follows

$$\hat{x}^{\text{ok}} \sim \mathcal{N}(0, \Sigma^{\text{ok}} S_\theta) \quad (\text{A.3})$$

which corresponds to (1.16).

Then, to compute the optimal distortion $f_{\hat{x}|x}$ may be obtained by considering the joint distribution of the j -th components x_j^{ok} and \hat{x}_j^{ok} that consists in $(\hat{x}_j^{\text{ok}}, x_j^{\text{ok}})^\top \sim$

$\mathcal{N}\left(0, \Sigma_{\hat{x}_j^{\text{ok}}, x_j^{\text{ok}}}\right)$, where $\Sigma_{\hat{x}_j^{\text{ok}}, x_j^{\text{ok}}}$ is the upper-left 2×2 submatrix of $\Sigma_{\hat{x}_j^{\text{ok}}, x_j^{\text{ok}}, \Delta_j}$ in (A.2). Assuming $\theta < \lambda_j^{\text{ok}}$, the $f_{\hat{x}|x}$ limited to the j -th component is the pdf of \hat{x}_j^{ok} given x_j^{ok} , i.e.,

$$\begin{aligned} f_{\hat{x}_j|x_j}^{\text{ok}}(\alpha, \beta) &= \frac{f_{\hat{x}_j, x_j}^{\text{ok}}(\alpha, \beta)}{f_{x_j}^{\text{ok}}(\beta)} = \frac{\mathcal{N}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}; 0, \Sigma_{\hat{x}_j^{\text{ok}}, x_j^{\text{ok}}}\right)}{\mathcal{N}(\beta; 0, \lambda_j^{\text{ok}})} \\ &= \frac{1}{\sqrt{2\pi \frac{\theta}{\lambda_j^{\text{ok}}} (\lambda_j^{\text{ok}} - \theta)}} \exp\left(-\frac{1}{2} \frac{[\lambda_j^{\text{ok}} \alpha - (\lambda_j^{\text{ok}} - \theta) \beta]^2}{\lambda_j^{\text{ok}} \theta (\lambda_j^{\text{ok}} - \theta)}\right) \\ &= \frac{1}{\sqrt{2\pi \lambda_j^{\text{ok}} \tau_j (1 - \tau_j)}} \exp\left(-\frac{1}{2} \frac{[\alpha - (1 - \tau_j) \beta]^2}{\lambda_j^{\text{ok}} \tau_j (1 - \tau_j)}\right) \end{aligned} \quad (\text{A.4})$$

where $\tau_j = \min\{1, \theta/\lambda_j^{\text{ok}}\}$ which ranges from 0 (when no distortion occurs, i.e., $\theta = 0$) to 1 (when the j -th component is completely distorted, i.e., $\theta \geq \lambda_j^{\text{ok}}$). Note that, if $\tau_j \rightarrow 1$, (A.4) degenerates into the Dirac's delta $\delta(\alpha)$, while when τ_j approaches to 0, $f_{\hat{x}_j|x_j}^{\text{ok}}$ tends to $\delta(\alpha - \beta)$.

Considering the matrices $T_\theta = \text{diag}(\tau_0, \dots, \tau_{n-1}) = \min\{I_n, \theta(\Sigma^{\text{ok}})^{-1}\}$ and $R_\theta = (I_n - T_\theta)T_\theta$, we can collect the component pdfs into a vector pdf to get:

$$f_{\hat{x}|x}(\alpha, \beta) = \mathcal{N}\left(\alpha; \beta S_\theta, \Sigma^{\text{ok}} R_\theta\right) \quad (\text{A.5})$$

which corresponds to (1.15). \square

A.2 Proof of Property 1.2

Proof. The pdf of \hat{x}^{ko} distorted by means of $f_{\hat{x}|x}$ can be computed as

$$f_{\hat{x}}^{\text{ko}}(\alpha) = \int_{\mathbb{R}^n} f_{\hat{x}, x}^{\text{ko}}(\alpha, \beta) d\beta = \int_{\mathbb{R}^n} f_{\hat{x}|x}(\alpha, \beta) f_x^{\text{ko}}(\beta) d\beta \quad (\text{A.6})$$

Let us assume first the low-distortion condition $\theta < \lambda_n^{\text{ok}}$ and then generalize for the case where $\theta \geq \lambda_n^{\text{ok}}$.

Low-distortion condition implies $T_\theta = \theta(\Sigma^{\text{ok}})^{-1}$ and one may write

$$\begin{aligned}
f_{\hat{x}}^{\text{ko}}(\alpha) &= \int_{\mathbb{R}^n} \mathcal{N}(\alpha; S_\theta \beta, \Sigma^{\text{ok}} R_\theta) \mathcal{N}(\beta; 0, \Sigma^{\text{ko}}) d\beta \\
&= \mathcal{N}(\alpha; 0, \Sigma^{\text{ok}} R_\theta) \\
&\quad \times \int_{\mathbb{R}^n} \exp\left(-\frac{1}{2} \left[\beta^\top S_\theta (\Sigma^{\text{ok}} R_\theta)^{-1} S_\theta \beta - 2\alpha^\top (\Sigma^{\text{ok}} R_\theta)^{-1} S_\theta \beta \right]\right) \\
&\quad \times \mathcal{N}(\beta; 0, \Sigma^{\text{ko}}) d\beta \\
&= \mathcal{N}(\alpha; 0, \Sigma^{\text{ok}} R_\theta) \frac{1}{\sqrt{(2\pi)^n |\Sigma^{\text{ko}}|}} \underbrace{\int_{\mathbb{R}^n} \exp\left(-\frac{1}{2} (\beta^\top V \beta - 2v^\top \beta)\right) d\beta}_{g(\alpha)}
\end{aligned} \tag{A.7}$$

where

$$V = S_\theta (\Sigma^{\text{ok}} R_\theta)^{-1} S_\theta + (\Sigma^{\text{ko}})^{-1} = (\theta I_n)^{-1} - (\Sigma^{\text{ok}})^{-1} + (\Sigma^{\text{ko}})^{-1} \tag{A.8}$$

$$v = (\Sigma^{\text{ok}} R_\theta)^{-1} S_\theta \alpha = \alpha / \theta \tag{A.9}$$

Since V is obtained as sum and product of real and symmetric matrices, it is real and symmetric and it can be decomposed as $V = QDQ^\top$ with D diagonal and Q orthonormal. Then, setting $\beta' = D^{1/2} Q^\top \beta$ so that $\beta = QD^{-1/2} \beta'$ and $d\beta = d\beta' / \sqrt{|V|}$ one can write

$$\begin{aligned}
g(\alpha) &= \frac{1}{\sqrt{|V|}} \int_{\mathbb{R}^n} e^{-\frac{1}{2} (\beta'^\top \beta' - 2v^\top QD^{-1/2} \beta')} d\beta' \\
&= \frac{1}{\sqrt{|V|}} \int_{\mathbb{R}^n} e^{-\frac{1}{2} (\|\beta' - D^{-1/2} Q^\top v\|^2 - v^\top V^{-1} v)} d\beta' \\
&= \sqrt{\frac{(2\pi)^n}{|V|}} e^{\frac{1}{2} v^\top V^{-1} v}
\end{aligned} \tag{A.10}$$

Putting this back into $f_{\hat{x}}^{\text{ok}}$ we get

$$f_{\hat{x}}^{\text{ko}}(\alpha) = \mathcal{N}(\alpha; 0, V \Sigma^{\text{ko}} \Sigma^{\text{ok}} R_\theta) \tag{A.11}$$

Considering the definition of V one may expand the covariance matrix into

$$\begin{aligned}
V \Sigma^{\text{ko}} \Sigma^{\text{ok}} R_\theta &= \left[(\theta I_n)^{-1} - (\Sigma^{\text{ok}})^{-1} + (\Sigma^{\text{ko}})^{-1} \right] \Sigma^{\text{ko}} \Sigma^{\text{ok}} \theta (\Sigma^{\text{ok}})^{-1} S_\theta \\
&= \left[\Sigma^{\text{ko}} - \theta (\Sigma^{\text{ok}})^{-1} \Sigma^{\text{ko}} + \theta I_n \right] S_\theta \\
&= \left[I_n - \theta (\Sigma^{\text{ok}})^{-1} \right] \Sigma^{\text{ko}} S_\theta + \theta S_\theta \\
&= S_\theta \Sigma^{\text{ko}} S_\theta + \theta S_\theta
\end{aligned} \tag{A.12}$$

which corresponds to the covariance in the statement of the Property.

To address the case in which $\theta \geq \lambda_n^{\text{ok}}$, one may notice that the lowest and right-most element of S_θ tend to 0 as $\theta \rightarrow (\lambda_n^{\text{ok}})^-$. As a result, the whole covariance tends to have zeros in its last row and column. Since we can generalize a 0 variance Gaussian as a Dirac's delta, the expression (A.12) also holds for this case. Moreover, iterating this consideration for $\theta \rightarrow (\lambda_j^{\text{ok}})^-$ with $j = n-1, n-2, \dots, 1$ one can conclude that (A.12) holds for any value of θ . \square

A.3 Proof of Property 1.3

Proof. Let x' and x'' be two n -dimensional Gaussian random vectors characterized by the pdfs $f_{x'}(\alpha) = \mathcal{N}(\alpha; 0, \Sigma')$ and $f_{x''}(\alpha) = \mathcal{N}(\alpha; 0, \Sigma'')$ then

$$\begin{aligned} L(x'; x'') &= - \int_{\mathbb{R}^n} f_{x'}(\alpha) \log_2 [f_{x''}(\alpha)] \, d\alpha \\ &= - \int_{\mathbb{R}^n} \mathcal{N}(\alpha; 0, \Sigma') \log_2 [\mathcal{N}(\alpha; 0, \Sigma'')] \, d\alpha \\ &= \frac{1}{2} \log_2 [(2\pi)^n |\Sigma''|] \int_{\mathbb{R}^n} \mathcal{N}(\alpha; 0, \Sigma') \, d\alpha \\ &\quad + \frac{1}{2 \log 2} \int_{\mathbb{R}^n} \alpha^\top (\Sigma'')^{-1} \alpha \mathcal{N}(\alpha; 0, \Sigma') \, d\alpha \\ &= \frac{1}{2} \log_2 [(2\pi)^n |\Sigma''|] + \frac{1}{2 \log 2} \text{tr} [(\Sigma'')^{-1} \Sigma'] \end{aligned} \quad (\text{A.13})$$

where the last summand has been computed as the expectation of a quadratic form in a Gaussian multivariate for which Corollary 3.2b.1 in [130, chapter 3] gives a formula. \square

A.4 Proof of Property 1.4

Proof. To prove Property 1.4, one may first prove that $\bar{\zeta}$ is continuous in its domain and then prove that its sign is opposite at the boundaries.

One may write (1.27) as $\bar{\zeta} = Z(\theta)/2 \log 2$ with $Z(\theta) = \sum_{j=1}^{k_\theta} \alpha_j(\theta)$ and

$$\alpha_j(\theta) = 1 - \frac{1}{\lambda_j^{\text{ok}}} \left(1 - \frac{\theta}{\lambda_j^{\text{ok}}} \right) - \frac{\theta}{\lambda_j^{\text{ok}}} \quad (\text{A.14})$$

With no loss of generality, one can set $\lambda_{n+1}^{\text{ok}} = 0$, assume $\lambda_1^{\text{ok}} > \lambda_2^{\text{ok}} > \dots > \lambda_n^{\text{ok}} > \lambda_{n+1}^{\text{ok}} = 0$, and define $\Theta_j =]\lambda_{j+1}^{\text{ok}}, \lambda_j^{\text{ok}}[$ for $j = 1, \dots, n$ so that if $\theta \in \Theta_j$ then $k_\theta = j$.

$\bar{\zeta}$ is continuous in θ if it is continuous if $Z(\theta)$ is continuous in θ . It is continuous in each interval Θ_j as $\alpha_j(\theta)$ is continuous. It is also continuous in the boundaries of each Θ_j since

$$\begin{aligned}
\lim_{\theta \rightarrow \lambda_j^{\text{ok}-}} Z &= \lim_{\theta \rightarrow \lambda_j^{\text{ok}-}} \sum_{l=1}^j \alpha_l(\theta) = \lim_{\theta \rightarrow \lambda_j^{\text{ok}-}} \alpha_j(\theta) + \sum_{l=1}^{j-1} \alpha_l(\theta) \\
&= \lim_{\theta \rightarrow \lambda_j^{\text{ok}+}} \sum_{j=1}^{\bar{j}-1} \alpha_j(\theta) = \lim_{\theta \rightarrow \lambda_j^{\text{ok}+}} Z
\end{aligned} \tag{A.15}$$

where it was exploited the fact that $\alpha_j(\lambda_j^{\text{ok}}) = 0$ and that the $\alpha_j(\theta)$ are continuous and thus their left and right limits coincide.

Let us first consider the value of $\bar{\zeta}$ at the left-hand boundary ($\theta = 0$) that is when no distortion is imposed by the encoder. In this case, $k_\theta = n$, $\alpha_j(0) = (1 - 1/\lambda_j^{\text{ok}})$ and one can write

$$Z = \sum_{j=1}^n \left(1 - \frac{1}{\lambda_j^{\text{ok}}}\right) = n - \sum_{j=1}^n \frac{1}{\lambda_j^{\text{ok}}} \leq n - n = 0 \tag{A.16}$$

where the inequality follows from the fact that $\sum_{j=1}^n 1/\lambda_j^{\text{ok}} \geq n$ when $\sum_{j=1}^n \lambda_j^{\text{ok}} = n$, which is an assumption in the signal model.

On the right-hand boundary of the domain of $\bar{\zeta}$ ($\theta = \lambda_1^{\text{ok}}$), that is when distortion is maximum, $Z = \alpha_1(\lambda_1^{\text{ok}}) = 0$ and in turn $\bar{\zeta} = 0$. Since this gives no information about the sign, one may observe the partial derivative of Z with respect to θ as θ approaches λ_1^{ok} .

$$\frac{\partial}{\partial \theta} Z = \sum_{j=1}^{k_\theta} \left(\frac{1}{\lambda_j^{\text{ok}}} - 1 \right) \frac{1}{\lambda_j^{\text{ok}}} \tag{A.17}$$

Note that, for any $\theta \geq \lambda_\ell^{\text{ok}}$ with $\ell = \arg \max_k \{\lambda_k^{\text{ok}} \geq 1\}$, $\frac{\partial}{\partial \theta} Z < 0$. This implies that Z , i.e., $\bar{\zeta}$, is monotonically decreasing as θ approaches λ_1^{ok} where it vanishes. Therefore, in the right-most part of the domain Z must be positive.

Since $Z < 0$ at $\theta = 0$ and $Z > 0$ for $\theta \in]\lambda_\ell^{\text{ok}}, \lambda_1^{\text{ok}}[$, Z must pass through 0 at least once in the interval $\theta \in]0, \lambda_\ell^{\text{ok}}[$. \square

A.5 Proof of Property 1.5

Proof. We will use the following Lemma whose proof is reported in Appendix A.6.

Lemma A.1. *If $\lambda^{\text{ko}} \sim \mathcal{U}(\mathbb{S}^n)$, then for any integrable function $f : \mathbb{R} \rightarrow \mathbb{R}$ and any $j = 1, \dots, n$*

$$\mathbf{E} \left[f(\lambda_j^{\text{ko}}) \right] = \frac{n-1}{n^{n-1}} \int_0^n f(\xi) (n-\xi)^{n-2} d\xi \tag{A.18}$$

From [95] we know that if Q is uniformly distributed in \mathbf{O}^n (i.e., if it is distributed according to the Haar measure on the orthogonal group) then, for any sequence of integers $M_n < n$ increasing with n but such that $M_n = o\left(\frac{n}{\log n}\right)$, the entries of the first M_n columns of Q converge in probability to independent random variables such that $\sqrt{n}Q_{j,k} \sim \mathcal{N}(0, 1)$ for $j = 1, \dots, n$, $k = 1, \dots, M_n$.

Such a property can be extended to any subset of M_n columns. In fact, given any subset of m_n columns of Q , there is a permutation matrix P such that QP has such columns as the first ones.

Yet, since $P \in \mathcal{O}^n$ and Q is distributed according to the Haar measure in that group, also QP is distributed according to that measure and the entries in those columns tend to independent Gaussians.

Divide now n by M_n as in $n = N_n M_n + m_n$, where N_n is the quotient and $0 \leq m_n < M_n$ is the remainder. We can look at Q as the concatenation of N_n matrices Q_i , for $i = 1, \dots, N_n$, each $n \times M_n$, and of a last matrix Q_{N_n} that is $n \times m_n$. From $M_n = o\left(\frac{n}{\log n}\right)$ we have $N_n = O(\log n)$ but we can choose M_n such that $N_n = o(n^\alpha)$ for any $\alpha > 0$.

If we set $Q^{\text{ko}} = Q$ then, $\Sigma^{\text{ko}} = Q^{\text{ko}} \Lambda^{\text{ko}} Q^{\text{ko}\top}$ can be written componentwise

$$\Sigma_{j,k}^{\text{ok}} = \sum_{l=1}^n Q_{j,l}^{\text{ko}} \lambda_l^{\text{ko}} Q_{k,l}^{\text{ko}} = \underbrace{\sum_{i=1}^{N_n} \sum_{l=1+(i-1)M_n}^{iM_n} \frac{1}{n} v_{j,l} \lambda_l^{\text{ko}} v_{k,l}}_{W_i} + \underbrace{\sum_{l=1+n-m_n}^n \frac{1}{n} v_{j,l} \lambda_l^{\text{ko}} v_{k,l}}_{W_{N_n}} \quad (\text{A.19})$$

where $i = 1, \dots, N_n$ scans the first submatrices Q_i , the last summand accounts for the *remainder* matrix Q_{N_n} , and, thanks to the above considerations, $v_{j,l} \sim v_{k,l} \sim \mathcal{N}(0, 1)$ for all j, k, l .

We now have to address that case $j \neq k$ and the case $j = k$ separately.

Asymptotics of $\Sigma_{j,k}^{\text{ko}}$ for $j \neq k$

Let us now consider W_1 as the representative of all other W_i for $i = 1, \dots, N_n$, written as $W_1 = \sum_{l=1}^{M_n} X_{n,l}$ with $X_{n,l} = \frac{1}{n} v_{j,l} \lambda_l^{\text{ok}} v_{k,l}$. All the normal random variables involved in such a sum are asymptotically independent but this is not true for the λ_l^{ok} since the eigenvalues are constrained to sum to n .

Therefore, W_1 is a triangular array of row-dependent random variables whose asymptotic behaviour can be analyzed by means of [143, Theorem 2.1] that is essentially a Lindeberg-Feller Central Limit Theorem with the row-wise independence relaxed to asymptotic row-wise incorrelation. To analyze the asymptotics of W_1 we note that

$$\mathbf{E}[X_{n,l}] = \mathbf{E}[\lambda_l^{\text{ko}}] \mathbf{E}[v_{j,l}] \mathbf{E}[v_{k,l}] = 0 \quad (\text{A.20})$$

Note also that, though not independent, the covariance and the correlation between $X_{n,l'}$ and $X_{n,l''}$ is

$$\begin{aligned} \mathbf{E}[X_{n,l'} X_{n,l''}] &= \frac{1}{n^2} \mathbf{E} \left[\lambda_{l'}^{\text{ko}} \lambda_{l''}^{\text{ko}} \right] \mathbf{E} [v_{j,l'} v_{j,l''}] \mathbf{E} [v_{k,l'} v_{k,l''}] \\ &= \begin{cases} 0 & \text{if } l' \neq l'' \\ \frac{1}{n^2} \mathbf{E}[(\lambda_l^{\text{ko}})^2] & \text{if } l' = l'' = l \end{cases} \quad (\text{A.21}) \end{aligned}$$

With this we also know that

$$\sigma_{W_1}^2 = \mathbf{E} [W_1^2] = \sum_{l'=1}^{M_n} \sum_{l''=1}^{M_n} \mathbf{E}[X_{n,l'} X_{n,l''}] = \sum_{l=1}^{M_n} \mathbf{E}[X_{n,l}^2] = \frac{1}{n^2} \sum_{l=1}^{M_n} \mathbf{E}[(\lambda_l^{\text{ko}})^2] \quad (\text{A.22})$$

To compute the last expectation we may resort to Lemma A.1 that gives

$$\mathbf{E}[(\lambda_l^{\text{ko}})^2] = \frac{n-1}{n^{n-1}} \int_0^n p^2 (n-p)^{n-2} dp = \frac{2n}{n+1} \quad (\text{A.23})$$

Hence we have $\sigma_{W_1}^2 = \frac{2M_n}{n(n+1)} \rightarrow 0$ for $n \rightarrow \infty$.

This helps satisfying the Lindeberg condition since, if for a given $\epsilon > 0$ we indicate with $\mathbf{E} [X_{n,l}^2 | |X_{n,l}| \geq \epsilon]$ the expectation of $X_{n,l}^2$ restricted to its values that are not less than ϵ in modulus, then

$$\sum_{l=1}^n \mathbf{E} [X_{n,l}^2 | |X_{n,l}| \geq \epsilon] \leq \sum_{l=1}^n \mathbf{E} [X_{n,l}^2] = \sigma_{W_1}^2 = \frac{2M_n}{n(n+1)} \quad (\text{A.24})$$

that vanishes asymptotically.

Finally, let $L_{M_n}, R_{M_n} \subset \{1, \dots, M_n\}$ be two index subsets such that $L_{M_n} \cap R_{M_n} = \emptyset$. If $g_{L_{M_n}}$ is any function of the random variables $X_{n,l}$ with $l \in L_{M_n}$ and $h_{R_{M_n}} = \prod_{l \in R_n} X_{n,l}$, the covariance between $g_{L_{M_n}}$ and $h_{R_{M_n}}$ is

$$\begin{aligned} \mathbf{E} [g_{L_{M_n}} h_{R_{M_n}}] &= \mathbf{E} \left[g_{L_m} \prod_{l \in R_n} \frac{1}{n} \lambda_l^{\text{ko}} v_{j,l} v_{k,l} \right] \\ &= \mathbf{E} \left[g_{L_m} \prod_{l \in R_n} \lambda_l^{\text{ko}} \right] \prod_{l \in R_n} \frac{1}{n} \mathbf{E} [v_{j,l}] \mathbf{E} [v_{k,l}] = 0 \end{aligned} \quad (\text{A.25})$$

that is enough to satisfy the assumptions in equation (2.3) and in equation (2.4) of [143, Theorem 2.1]. From that Theorem, we finally now that $W_1 \sim \mathcal{N} \left(0, \frac{2M_n}{n(n+1)} \right)$ when $n \rightarrow \infty$. Clearly, the same happens to any W_i for $i = 1, \dots, N_n$, while $W_{N_n} \sim \mathcal{N} \left(0, \frac{2m_n}{n(n+1)} \right)$.

Let us now consider

$$\begin{aligned} \Pr \left\{ \left| \boldsymbol{\Sigma}_{j,k}^{\text{ko}} \right| \leq \epsilon \right\} &= \Pr \left\{ \left| \sum_{i=1}^{N_n} W_i + W_{N_n} \right| \leq \epsilon \right\} \\ &\geq \Pr \left\{ \sum_{i=1}^{N_n} |W_i| + |W_{N_n}| \leq \epsilon \right\} \\ &\geq \Pr \left\{ \max \{ |W_1|, \dots, |W_{N_n}| \} \leq \frac{\epsilon}{N_n} \right\} \\ &= 1 - \Pr \left\{ \max \{ |W_1|, \dots, |W_{N_n}| \} > \frac{\epsilon}{N_n} \right\} \end{aligned} \quad (\text{A.26})$$

Yet

$$\begin{aligned} \Pr \left\{ \max \{ |W_1|, \dots, |W_{N_n}| \} > \frac{\epsilon}{N_n} \right\} &= \Pr \left\{ |W_1| > \frac{\epsilon}{N_n} \vee \dots \vee |W_{N_n}| > \frac{\epsilon}{N_n} \right\} \\ &\leq N_n \Pr \left\{ |W_1| > \frac{\epsilon}{N_n} \right\} \end{aligned} \quad (\text{A.27})$$

so that

$$\Pr \left\{ \left| \Sigma_{j,k}^{\text{ko}} \right| \leq \epsilon \right\} \geq 1 - N_n \Pr \left\{ |W_1| > \frac{\epsilon}{N_n} \right\} \quad (\text{A.28})$$

From this and from the asymptotic normality of W_1 we may say

$$\Pr \left\{ \left| \Sigma_{j,k}^{\text{ko}} \right| \leq \epsilon \right\} \geq 1 - \frac{N_n}{2} \operatorname{erfc} \left(\frac{\epsilon}{2\sqrt{\frac{N_n^2 M_n}{n(n+1)}}} \right) \quad (\text{A.29})$$

in which the probability tends to 1 for $n \rightarrow \infty$, and $M_n = o\left(\frac{n}{\log n}\right)$ chosen to have $N_n = o(n^{1/2})$.

Asymptotics of $\Sigma_{j,k}^{\text{ko}}$ for $j = k$

In this case, exploiting the fact that $\sum_{j=1}^n \lambda_j^{\text{ko}} = n$, we may write

$$\Sigma_{j,j}^{\text{ok}} = 1 + \underbrace{\sum_{i=1}^{N_n} \sum_{l=1+(i-1)M_n}^{iM_n} \frac{1}{n} (v_{j,l}^2 - 1) \lambda_l^{\text{ko}}}_{W_i} + \underbrace{\sum_{l=1+n-m_n}^n \frac{1}{n} (v_{j,l}^2 - 1) \lambda_l^{\text{ko}}}_{W_{N_n}} \quad (\text{A.30})$$

in which the summands W_i are a triangular arrays of elements with features similar to the previous ones.

In fact, we may focus on $W_1 = \sum_{l=1}^{M_n} X_{n,l}$ with $X_{n,l} = \frac{1}{n} \lambda_l^{\text{ko}} (v_{j,l}^2 - 1)$ and note that $\mathbf{E}[X_{n,l}] = 0$ and

$$\begin{aligned} \mathbf{E}[X_{n,l'} X_{n,l''}] &= \frac{1}{n^2} \mathbf{E}[\lambda_{l'}^{\text{ko}} \lambda_{l''}^{\text{ko}}] \mathbf{E}[(v_{j,l'}^2 - 1)(v_{j,l''}^2 - 1)] \\ &= \begin{cases} 0 & \text{if } l' \neq l'' \\ \frac{2}{n^2} \mathbf{E}[(\lambda_l^{\text{ko}})^2] & \text{if } l' = l'' = l \end{cases} \end{aligned} \quad (\text{A.31})$$

where we have exploited the fact that $\mathbf{E}[(v_{j,l}^2 - 1)^2] = \mathbf{E}[v_{j,l}^4] - 2\mathbf{E}[v_{j,l}^2] + 1 = 2$. With this, $\sigma_{W_1}^2 = \frac{4M_n}{n(n+1)} \rightarrow 0$ for $n \rightarrow \infty$.

As before, this makes the Lindeberg condition automatically satisfied and is also enough to satisfy the covariance constraints in equations (2.3) and (2.4) of [143, Theorem 2.1] from which we get that $W_1 \sim \mathcal{N}\left(0, \frac{4M_n}{n(n+1)}\right)$ for $n \rightarrow \infty$. An analogous path leads to the asymptotic behaviour $W_{N_n} \sim \mathcal{N}\left(0, \frac{4m_n}{n(n+1)}\right)$.

The same inequalities as before lead to

$$\Pr \left\{ \left| \Sigma_{jj}^{\text{ko}} - 1 \right| \leq \epsilon \right\} \geq 1 - \frac{N_n}{2} \operatorname{erfc} \left(\frac{\epsilon}{4 \sqrt{\frac{N_n^2 M_n}{n(n+1)}}} \right) \quad (\text{A.32})$$

in which the probability tends to 1 for $n \rightarrow \infty$, and $M_n = o\left(\frac{n}{\log n}\right)$ chosen to have $N_n = o(n^{1/2})$.

From (A.29) and (A.32) we finally get that Q^{ko} tends to I_n in probability as $n \rightarrow \infty$. \square

A.6 Proof of Lemma A.1

Proof. For any function $f : \mathbb{R} \mapsto \mathbb{R}$ we have

$$\begin{aligned} \mathcal{I}[f(\xi)] &= \int_{\mathbb{S}^n} f(\xi_1) d\xi_1 \dots d\xi_n \\ &= \int_0^n f(\xi_1) \int_0^{n-\xi_1} \int_0^{n-\xi_1-\xi_2} \dots \int_0^{n-\xi_1-\xi_2-\dots-\xi_{n-2}} d\xi_1 \dots d\xi_{n-1} \\ &= \int_0^n f(\xi_1) \frac{(n-\xi_1)^{n-2}}{(n-2)!} d\xi_1 \end{aligned} \quad (\text{A.33})$$

Since λ^{ko} is uniformly distributed over \mathbb{S}^n the probability density is the constant $1/\mathcal{I}[1] = n^{-(n-1)}(n-1)!$ and the expectation of f is

$$\begin{aligned} \mathbf{E}[f(\lambda_j^{\text{ko}})] &= \frac{(n-1)!}{n^{n-1}} \mathcal{I}[f(\xi)] \\ &= \frac{(n-1)!}{n^{n-1}} \int_0^n f(\xi) \frac{(n-\xi)^{n-2}}{(n-2)!} d\xi \\ &= \frac{n-1}{n^{n-1}} \int_0^n f(\xi) (n-\xi)^{n-2} d\xi \end{aligned} \quad (\text{A.34})$$

\square

A.7 Proof of Property 2.1

Proof. From (2.7), and independently of the Gaussian assumption, one immediately gets the well-known average energy $\mu_{\mathcal{E}_U} = \sum_{j=1}^k \lambda_j$.

As far as the second-order statistics are concerned, note that

$$\mathcal{E}_U^2 = \left[\left(U^\top x \right)^\top U^\top x \right]^2 = x^\top U U^\top x x^\top U U^\top x = \operatorname{tr} \left(U U^\top x x^\top U U^\top x x^\top \right) \quad (\text{A.35})$$

that, brought down to sums over indexed quantities, gives

$$\begin{aligned}
\mathbf{E}_x [\mathcal{E}_U^2] &= \mathbf{E}_x \left[\sum_{a,b,c,d,e,f} U_{a,b} U_{c,b} x_c x_d U_{d,e} U_{f,e} x_f x_a \right] \\
&= \sum_{a,b,c,d,e,f} U_{a,b} U_{c,b} U_{d,e} U_{f,e} \mathbf{E}_x [x_c x_d x_f x_a]
\end{aligned} \tag{A.36}$$

where a sum in a set of indices shortens a sequence of sum each summing over a different index going from 1 to n .

Since x is Gaussian and zero-mean $\mathbf{E}_x [x_c x_d x_f x_a] = \Sigma_{c,d} \Sigma_{f,a} + \Sigma_{c,f} \Sigma_{d,a} + \Sigma_{c,a} \Sigma_{d,f}$. Hence,

$$\begin{aligned}
\mathbf{E}_x [\mathcal{E}_U^2] &= \sum_{a,b,c,d,e,f} U_{a,b} U_{c,b} \Sigma_{c,d} U_{d,e} U_{f,e} \Sigma_{f,a} + \\
&\quad + \sum_{a,b,c,d,e,f} U_{a,b} U_{c,b} \Sigma_{c,f} U_{f,e} U_{d,e} \Sigma_{d,a} + \\
&\quad + \sum_{a,b,c} U_{a,b} U_{c,b} \Sigma_{c,a} \sum_{d,e,f} U_{f,e} U_{d,e} \Sigma_{d,f}
\end{aligned} \tag{A.37}$$

where factors and sums have been rearranged and distributed to reconstruct high-level matrix and vector operations. By recognizing such operations we have

$$\begin{aligned}
\mathbf{E}_x [\mathcal{E}_U^2] &= 2\text{tr} (UU^\top \Sigma UU^\top \Sigma) + \text{tr} (UU^\top \Sigma) \text{tr} (UU^\top \Sigma) \\
&= 2\text{tr} (U^\top \Sigma UU^\top \Sigma U) + \text{tr}^2 (U^\top \Sigma U) \\
&= 2 \sum_{l=1}^k \lambda_{jl}^2 + \left(\sum_{l=1}^k \lambda_{jl} \right)^2
\end{aligned} \tag{A.38}$$

where we have exploited the fact that since U has eigenvectors of Σ as columns, $U^\top \Sigma U = \text{diag} (\lambda_{j_1}, \lambda_{j_2}, \dots, \lambda_{j_k})$.

Finally,

$$\sigma_{\tilde{\varepsilon}_U}^2 = \mathbf{E}_x [(\mathcal{E}_U - \mu_{\mathcal{E}_U})^2] = \mathbf{E}_x [\mathcal{E}_U^2] - \mu_{\mathcal{E}_U}^2 = 2 \sum_{l=1}^k \lambda_{jl}^2 \tag{A.39}$$

□

Bibliography

- [1] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from [tensorflow.org](https://www.tensorflow.org/). 2015. URL: <https://www.tensorflow.org/>.
- [2] K. Abed-Meraim, A. Chkeif, and Y. Hua. “Orthogonal Oja Algorithm”. In: *IEEE Signal Processing Letters* 7.5 (May 2000), pp. 116–119. ISSN: 15582361. DOI: [10.1109/97.841157](https://doi.org/10.1109/97.841157).
- [3] Mohammad Abu Alsheikh et al. “Rate-Distortion Balanced Data Compression for Wireless Sensor Networks”. In: *IEEE Sensors Journal* 16.12 (2016), pp. 5072–5083. DOI: [10.1109/JSEN.2016.2550599](https://doi.org/10.1109/JSEN.2016.2550599).
- [4] N. Ahmed, T. Natarajan, and K. R. Rao. “Discrete Cosine Transform”. In: *IEEE Transactions on Computers* C-23.1 (Jan. 1974), pp. 90–93. ISSN: 2326-3814. DOI: [10.1109/T-C.1974.223784](https://doi.org/10.1109/T-C.1974.223784).
- [5] Atif Alamri et al. “A Survey on Sensor-Cloud: Architecture, Applications, and Approaches”. In: *International Journal of Distributed Sensor Networks* 9.2 (2013), p. 917923. DOI: [10.1155/2013/917923](https://doi.org/10.1155/2013/917923).
- [6] Zeyuan Allen-Zhu and Yuanzhi Li. “First efficient convergence for streaming k-pca: A global, gap-free, and near-optimal rate”. In: *Annual Symposium on Foundations of Computer Science - Proceedings*. Vol. 2017-Octob. IEEE Computer Society, Nov. 2017, pp. 487–492. ISBN: 9781538634646. DOI: [10.1109/FOCS.2017.51](https://doi.org/10.1109/FOCS.2017.51).
- [7] D. Allstot, R. Rovatti, and G. Setti (Eds.) “Special Issue on Circuits, Systems and Algorithms for Compressed Sensing”. In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2.3 (July 2012). ISSN: 2156-3357.
- [8] J. P. Amezquita-Sanchez and H. Adeli. “Signal Processing Techniques for Vibration-Based Health Monitoring of Smart Structures”. In: *Archives of Computational Methods in Engineering* 23.1 (Mar. 2016), pp. 1–15. ISSN: 1886-1784. DOI: [10.1007/s11831-014-9135-7](https://doi.org/10.1007/s11831-014-9135-7).
- [9] E. Anderson et al. *LAPACK Users’ Guide*. Third. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999. ISBN: 0-89871-447-8 (paperback).
- [10] Reinaldo B. Arellano-Valle, Javier E. Contreras-Reyes, and Marc G. Genton. “Shannon Entropy and Mutual Information for Multivariate Skew-Elliptical Distributions”. In: *Scandinavian Journal of Statistics* 40.1 (2013), pp. 42–62. DOI: [10.1111/j.1467-9469.2011.00774.x](https://doi.org/10.1111/j.1467-9469.2011.00774.x).

- [11] Raman Arora et al. "Stochastic optimization for PCA and PLS". In: *2012 50th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2012*. 2012, pp. 861–868. ISBN: 9781467345385. DOI: [10.1109/Allerton.2012.6483308](https://doi.org/10.1109/Allerton.2012.6483308).
- [12] S. Attallah. "The generalized Rayleigh's quotient adaptive noise subspace algorithm: a householder transformation-based implementation". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 53.1 (Jan. 2006), pp. 3–7. ISSN: 1057-7130. DOI: [10.1109/TCSII.2005.854587](https://doi.org/10.1109/TCSII.2005.854587).
- [13] S. Attallah and K. Abed-Meraim. "Fast algorithms for subspace tracking". In: *IEEE Signal Processing Letters* 8.7 (July 2001), pp. 203–206. ISSN: 10709908. DOI: [10.1109/97.928678](https://doi.org/10.1109/97.928678).
- [14] S. Attallah and K. Abed-Meraim. "Low-cost adaptive algorithm for noise subspace estimation". In: *Electronics Letters* 38.12 (June 2002), pp. 609–611. ISSN: 00135194. DOI: [10.1049/e1:20020388](https://doi.org/10.1049/e1:20020388).
- [15] R. Badeau, B. David, and G. Richard. "Fast approximated power iteration subspace tracking". In: *IEEE Transactions on Signal Processing* 53.8 (Aug. 2005), pp. 2931–2941. ISSN: 1053-587X. DOI: [10.1109/TSP.2005.850378](https://doi.org/10.1109/TSP.2005.850378).
- [16] R. Badeau et al. "Approximated power iterations for fast subspace tracking". In: *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings. IEEE, 2003*, pp. 583–586. ISBN: 0-7803-7946-2. DOI: [10.1109/ISSPA.2003.1224944](https://doi.org/10.1109/ISSPA.2003.1224944).
- [17] Laura Balzano, Yuejie Chi, and Yue M Lu. "Streaming PCA and Subspace Tracking: The Missing Data Case". In: *Proceedings of the IEEE* 106.8 (2018), pp. 1293–1310. DOI: [10.1109/JPROC.2018.2847041](https://doi.org/10.1109/JPROC.2018.2847041).
- [18] Laura Balzano, Robert Nowak, and Benjamin Recht. "Online identification and tracking of subspaces from highly incomplete information". In: 2010, pp. 704–711. ISBN: 978-1-4244-8214-6. DOI: [10.1109/ALLERTON.2010.5706976](https://doi.org/10.1109/ALLERTON.2010.5706976).
- [19] Laura Balzano and Stephen J. Wright. "Local Convergence of an Algorithm for Subspace Identification from Partial Data". In: *Foundations of Computational Mathematics* 15.5 (Oct. 2015), pp. 1279–1314. ISSN: 16153383. DOI: [10.1007/s10208-014-9227-7](https://doi.org/10.1007/s10208-014-9227-7). URL: <http://link.springer.com/10.1007/s10208-014-9227-7>.
- [20] Laura Balzano and Stephen J. Wright. "On GROUSE and incremental SVD". In: *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP 2013*. 2013, pp. 1–4. ISBN: 9781467331463. DOI: [10.1109/CAMSAP.2013.6713992](https://doi.org/10.1109/CAMSAP.2013.6713992).
- [21] R. G. Baraniuk et al. "Special Issue on Applications of Sparse Representation and Compressive Sensing". In: *Proceedings of the IEEE* 98.6 (June 2010). ISSN: 0018-9219.

- [22] R. G. Baraniuk et al. "Special issue on Compressive Sampling". In: *IEEE Signal Processing Magazine* 25.2 (Mar. 2008). ISSN: 1053-5888.
- [23] Amir Beck and Marc Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202. DOI: [10.1137/080716542](https://doi.org/10.1137/080716542).
- [24] D. Bellasi et al. "A Broadband Multi-Mode Compressive Sensing Current Sensor SoC in 0.16 μ m CMOS". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 66.1 (Jan. 2019), pp. 105–118. ISSN: 1549-8328. DOI: [10.1109/TCSI.2018.2846573](https://doi.org/10.1109/TCSI.2018.2846573).
- [25] D. E. Bellasi et al. "A Low-Power Architecture for Punctured Compressed Sensing and Estimation in Wireless Sensor-Nodes". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 62.5 (May 2015), pp. 1296–1305. ISSN: 1549-8328. DOI: [10.1109/TCSI.2015.2418833](https://doi.org/10.1109/TCSI.2015.2418833).
- [26] E. van den Berg and M. P. Friedlander. *SPGL1: A solver for large-scale sparse reconstruction*. <http://www.cs.ubc.ca/labs/scl/spgl1>. June 2007.
- [27] Gabriele Bertagnoli et al. "A Large Scale SHM System: A Case Study on Pre-stressed Bridge and Cloud Architecture". In: *Dynamics of Civil Structures, Volume 2*. Ed. by Shamim Pakzad. Cham: Springer International Publishing, 2020, pp. 75–83. ISBN: 978-3-030-12115-0.
- [28] N. Bertoni et al. "Low-power EEG monitor based on compressed sensing with compressed domain noise rejection". In: *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2016, pp. 522–525. DOI: [10.1109/ISCAS.2016.7527292](https://doi.org/10.1109/ISCAS.2016.7527292).
- [29] T. Bianchi, V. Bioglio, and E. Magli. "Analysis of One-Time Random Projections for Privacy Preserving Compressed Sensing". In: *IEEE Transactions on Information Forensics and Security* 11.2 (Feb. 2016), pp. 313–327. ISSN: 1556-6013. DOI: [10.1109/TIFS.2015.2493982](https://doi.org/10.1109/TIFS.2015.2493982).
- [30] L Susan Blackford et al. "An updated set of basic linear algebra subprograms (BLAS)". In: *ACM Transactions on Mathematical Software* 28.2 (2002), pp. 135–151.
- [31] Davis Blalock, Samuel Madden, and John Guttag. "Sprintz: Time Series Compression for the Internet of Things". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2.3 (Sept. 2018). DOI: [10.1145/3264903](https://doi.org/10.1145/3264903).
- [32] Yochai Blau and Tomer Michaeli. "Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, 2019, pp. 675–685.

- [33] Yochai Blau and Tomer Michaeli. "The Perception-Distortion Tradeoff". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6228–6237. DOI: [10.1109/CVPR.2018.00652](https://doi.org/10.1109/CVPR.2018.00652).
- [34] Alessio Botta et al. "Integration of Cloud computing and Internet of Things: A survey". In: *Future Generation Computer Systems* 56 (2016), pp. 684–700. ISSN: 0167-739X. DOI: [10.1016/j.future.2015.09.021](https://doi.org/10.1016/j.future.2015.09.021).
- [35] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN: 0521833787.
- [36] Matthew Brand. "Fast low-rank modifications of the thin singular value decomposition". In: *Linear Algebra and Its Applications* 415.1 (May 2006), pp. 20–30. ISSN: 00243795. DOI: [10.1016/j.laa.2005.07.021](https://doi.org/10.1016/j.laa.2005.07.021).
- [37] James R Bunch and Christopher P Nielsen. "Updating the singular value decomposition". In: *Numerische Mathematik* 31.2 (1978), pp. 111–129. DOI: [10.1007/BF01397471](https://doi.org/10.1007/BF01397471).
- [38] A. Burrello et al. "Enhancing Structural Health Monitoring with Vehicle Identification and Tracking". In: *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. 2020, pp. 1–6.
- [39] Alessio Burrello et al. "Embedded Streaming Principal Components Analysis for Network Load Reduction in Structural Health Monitoring". In: *IEEE Internet of Things Journal* 8.6 (2021), pp. 4433–4447. DOI: [10.1109/JIOT.2020.3027102](https://doi.org/10.1109/JIOT.2020.3027102).
- [40] Alessio Burrello et al. "Embedding Principal Component Analysis for Data Reduction in Structural Health Monitoring on Low-Cost IoT Gateways". In: *Proceedings of the 16th ACM International Conference on Computing Frontiers*. CF '19. Alghero, Italy: Association for Computing Machinery, 2019, 235–239. ISBN: 9781450366854. DOI: [10.1145/3310273.3322822](https://doi.org/10.1145/3310273.3322822).
- [41] V. Cambareri et al. "A Case Study in Low-Complexity ECG Signal Encoding: How Compressing is Compressed Sensing?" In: *IEEE Signal Processing Letters* 22.10 (Oct. 2015), pp. 1743–1747. DOI: [10.1109/LSP.2015.2428431](https://doi.org/10.1109/LSP.2015.2428431).
- [42] V. Cambareri et al. "A rakesness-based design flow for Analog-to-Information conversion by Compressive Sensing". In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2013, pp. 1360–1363. DOI: [10.1109/ISCAS.2013.6572107](https://doi.org/10.1109/ISCAS.2013.6572107).
- [43] V. Cambareri et al. "Low-Complexity Multiclass Encryption by Compressed Sensing". In: *IEEE Transactions on Signal Processing* 63.9 (2015), pp. 2183–2195. DOI: [10.1109/TSP.2015.2407315](https://doi.org/10.1109/TSP.2015.2407315).
- [44] V. Cambareri et al. "On Known-Plaintext Attacks to a Compressed Sensing-Based Encryption: A Quantitative Analysis". In: *IEEE Transactions on Information Forensics and Security* 10.10 (Oct. 2015), pp. 2182–2195. ISSN: 1556-6013. DOI: [10.1109/TIFS.2015.2450676](https://doi.org/10.1109/TIFS.2015.2450676).

- [45] E. J. Candes, J. Romberg, and T. Tao. "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information". In: *IEEE Trans. on Inf. Theory* 52.2 (Feb. 2006), pp. 489–509. ISSN: 0018-9448. DOI: [10.1109/TIT.2005.862083](https://doi.org/10.1109/TIT.2005.862083).
- [46] E. J. Candes and T. Tao. "Decoding by linear programming". In: *IEEE Transactions on Information Theory* 51.12 (Dec. 2005), pp. 4203–4215. ISSN: 0018-9448. DOI: [10.1109/TIT.2005.858979](https://doi.org/10.1109/TIT.2005.858979).
- [47] J. V. Candy et al. "Anomaly detection for a vibrating structure: A subspace identification/tracking approach". In: *The Journal of the Acoustical Society of America* 142.2 (2017), pp. 680–696. DOI: [10.1121/1.4996142](https://doi.org/10.1121/1.4996142).
- [48] E. J. Candès. "The restricted isometry property and its implications for compressed sensing". In: *Comptes Rendus Mathématique* 346.9 (2008), pp. 589–592. ISSN: 1631-073X. DOI: [10.1016/j.crma.2008.03.014](https://doi.org/10.1016/j.crma.2008.03.014).
- [49] Peter Cawley. "Structural health monitoring: closing the gap between research and industrial deployment". In: *Structural Health Monitoring* 17.5 (Jan. 2018), pp. 1225–1244. DOI: [10.1177/1475921717750047](https://doi.org/10.1177/1475921717750047).
- [50] Peter C. Chang, Alison Flatau, and S. C. Liu. "Review Paper: Health Monitoring of Civil Infrastructure". In: *Structural Health Monitoring* 2.3 (2003), pp. 257–267. DOI: [10.1177/1475921703036169](https://doi.org/10.1177/1475921703036169).
- [51] Ling-Jyh Chen et al. "ADF: An Anomaly Detection Framework for Large-Scale PM2.5 Sensing Systems". In: *IEEE Internet of Things Journal* 5.2 (Apr. 2018), pp. 559–570. ISSN: 2327-4662. DOI: [10.1109/JIOT.2017.2766085](https://doi.org/10.1109/JIOT.2017.2766085).
- [52] Xi Chen et al. "A Sub-Nyquist Rate Compressive Sensing Data Acquisition Front-End". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2.3 (Sept. 2012), pp. 542–551. ISSN: 2156-3357. DOI: [10.1109/JETCAS.2012.2221531](https://doi.org/10.1109/JETCAS.2012.2221531).
- [53] Yuejie Chi, Yonina C. Eldar, and Robert Calderbank. "PETRELS: Subspace estimation and tracking from partial observations". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. 2012, pp. 3301–3304. ISBN: 9781467300469. DOI: [10.1109/ICASSP.2012.6288621](https://doi.org/10.1109/ICASSP.2012.6288621).
- [54] François Chollet et al. *Keras*. <https://keras.io>. 2015.
- [55] *CMSIS-DSP, Digital Signal Processing Software Library*. URL: <https://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>.
- [56] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991. ISBN: 0471062596.
- [57] Amir V. Dastjerdi and Rajkumar Buyya. "Fog Computing: Helping the Internet of Things Realize Its Potential". In: *Computer* 49.8 (Aug. 2016), pp. 112–116. ISSN: 0018-9162. DOI: [10.1109/MC.2016.245](https://doi.org/10.1109/MC.2016.245).

- [58] D. Del Testa and M. Rossi. "Lightweight Lossy Compression of Biometric Patterns via Denoising Autoencoders". In: *IEEE Signal Processing Letters* 22.12 (Dec. 2015), pp. 2304–2308. DOI: [10.1109/LSP.2015.2476667](https://doi.org/10.1109/LSP.2015.2476667).
- [59] Chuanwei Ding et al. "Non-Contact Human Motion Recognition Based on UWB Radar". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 8.2 (June 2018), pp. 306–315. ISSN: 21563357. DOI: [10.1109/JETCAS.2018.2797313](https://doi.org/10.1109/JETCAS.2018.2797313).
- [60] A. M. R. Dixon et al. "Compressed Sensing System Considerations for ECG and EMG Wireless Biosensors". In: *IEEE Transactions on Biomedical Circuits and Systems* 6.2 (Apr. 2012), pp. 156–166. ISSN: 1932-4545. DOI: [10.1109/TBCAS.2012.2193668](https://doi.org/10.1109/TBCAS.2012.2193668).
- [61] David L. Donoho. "For most large underdetermined systems of linear equations the minimal ℓ_1 -norm solution is also the sparsest solution". In: *Communications on Pure and Applied Mathematics* 59.6 (2006), pp. 797–829. DOI: [10.1002/cpa.20132](https://doi.org/10.1002/cpa.20132).
- [62] Alan Edelman, TOMÁS A. Arias, and Steven T. Smith. "The Geometry of Algorithms with Orthogonality Constraints". In: *SIAM Journal on Matrix Analysis and Applications* 20.2 (Jan. 1998), pp. 303–353. ISSN: 0895-4798. DOI: [10.1137/S0895479895290954](https://doi.org/10.1137/S0895479895290954).
- [63] Michael Elad. "Optimized Projections for Compressed Sensing". In: *IEEE Transactions on Signal Processing* 55.12 (2007), pp. 5695–5702. DOI: [10.1109/TSP.2007.900760](https://doi.org/10.1109/TSP.2007.900760).
- [64] Martin Ester et al. "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.
- [65] Samsung ARTIK 7 Family. *Samsung ARTIK 710 IoT Module*. 2016. URL: <https://www.artik.io/modules/artik-710/>.
- [66] R.W. Farebrother. "Algorithm AS204: The Distribution of a Positive Linear Combination of χ^2 Random Variables". In: *Journal of the Royal Statistical Society (Series C)* 33.3 (1984), pp. 332–339. DOI: [10.2307/2347721](https://doi.org/10.2307/2347721).
- [67] Charles R Farrar and Keith Worden. "An introduction to structural health monitoring". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365.1851 (2007), pp. 303–315. DOI: [10.1098/rsta.2006.1928](https://doi.org/10.1098/rsta.2006.1928).
- [68] Charles R. Farrar and Keith Worden. *Structural Health Monitoring: A Machine Learning Perspective*. West Sussex, U.K: Wiley, 2013. ISBN: 978-1-119-99433-6.

- [69] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. DOI: [10.1016/j.patrec.2005.10.010](https://doi.org/10.1016/j.patrec.2005.10.010).
- [70] Brian Foo, Yiannis Andreopoulos, and Mihaela van der Schaar. "Analytical Complexity Modeling of Wavelet-based Video Coders". In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*. Vol. 3. 2007, pp. III-789–III-792. DOI: [10.1109/ICASSP.2007.366798](https://doi.org/10.1109/ICASSP.2007.366798).
- [71] Raspberry Pi Foundation. *Raspberry Pi 3 Module B*. 2016. URL: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/>.
- [72] Jun-Song Fu et al. "Secure Data Storage and Searching for Industrial IoT by Integrating Fog Computing and Cloud Computing". In: *IEEE Transactions on Industrial Informatics* 14.10 (2018), pp. 4519–4528. DOI: [10.1109/TII.2018.2793350](https://doi.org/10.1109/TII.2018.2793350).
- [73] Chinthaka Gamanayake et al. "Cluster Pruning: An Efficient Filter Pruning Method for Edge AI Vision Applications". In: *IEEE Journal of Selected Topics in Signal Processing* 14.4 (2020), pp. 802–816. DOI: [10.1109/JSTSP.2020.2971418](https://doi.org/10.1109/JSTSP.2020.2971418).
- [74] W. Gander. *Algorithms for the QR decomposition*. Seminar for Applied Mathematics. Apr. 1980. URL: <https://people.inf.ethz.ch/gander/papers/qrneu.pdf>.
- [75] D. Gangopadhyay et al. "Compressed Sensing Analog Front-End for Bio-Sensor Applications". In: *IEEE Journal of Solid-State Circuits* 49.2 (Feb. 2014), pp. 426–438. ISSN: 0018-9200. DOI: [10.1109/JSSC.2013.2284673](https://doi.org/10.1109/JSSC.2013.2284673).
- [76] Jean Ginibre. "Statistical Ensembles of Complex, Quaternion, and Real Matrices". In: *Journal of Mathematical Physics* 6.3 (1965), pp. 440–449. DOI: [10.1063/1.1704292](https://doi.org/10.1063/1.1704292).
- [77] A. Girolami, D. Brunelli, and L. Benini. "Low-cost and distributed health monitoring system for critical buildings". In: *2017 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*. July 2017, pp. 1–6. DOI: [10.1109/EESMS.2017.8052686](https://doi.org/10.1109/EESMS.2017.8052686).
- [78] A. Girolami et al. "Modal Analysis of Structures with Low-cost Embedded Systems". In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2018, pp. 1–4. DOI: [10.1109/ISCAS.2018.8351705](https://doi.org/10.1109/ISCAS.2018.8351705).
- [79] Ary L. Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals". In: *Circulation* 101.23 (June 2000), pp. 215–220. DOI: [10.5555/3009657.3009745](https://doi.org/10.5555/3009657.3009745).
- [80] Gene H. Golub and Christian H. Reinsch. "Singular value decomposition and least squares solutions". In: *Numerische Mathematik* 14.5 (Apr. 1970), pp. 403–420. ISSN: 0945-3245. DOI: [10.1007/BF02163027](https://doi.org/10.1007/BF02163027).

- [81] Gene H Golub and Charles F Van Loan. *Matrix computations*. Vol. 3. JHU Press, 2012.
- [82] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [83] M. Grossberg et al. "A Comparative Study of Lossless Compression Algorithms on Multi-spectral Imager Data". In: *2009 Data Compression Conference*. 2009, pp. 447–447. DOI: [10.1109/DCC.2009.68](https://doi.org/10.1109/DCC.2009.68).
- [84] Tony Gustafsson. "Instrumental variable subspace tracking using projection approximation". In: *IEEE Transactions on Signal Processing* 46.3 (1998), pp. 669–681. ISSN: 1053587X. DOI: [10.1109/78.661334](https://doi.org/10.1109/78.661334).
- [85] Javier Haboba et al. "A Pragmatic Look at Some Compressive Sensing Architectures With Saturation and Quantization". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2.3 (Sept. 2012), pp. 443–459. ISSN: 2156-3357. DOI: [10.1109/JETCAS.2012.2220392](https://doi.org/10.1109/JETCAS.2012.2220392).
- [86] Gregory Hackmann et al. "Cyber-Physical Codesign of Distributed Structural Health Monitoring with Wireless Sensor Networks". In: *IEEE Transactions on Parallel and Distributed Systems* 25.1 (Jan. 2014), pp. 63–72. ISSN: 1045-9219. DOI: [10.1109/TPDS.2013.30](https://doi.org/10.1109/TPDS.2013.30).
- [87] David J. Hand. "Measuring classifier performance: a coherent alternative to the area under the ROC curve". In: *Machine Learning* 77.1 (Oct. 2009), pp. 103–123. ISSN: 1573-0565. DOI: [10.1007/s10994-009-5119-5](https://doi.org/10.1007/s10994-009-5119-5).
- [88] Moritz Hardt and Eric Price. "The Noisy Power Method: A Meta Algorithm with Applications". In: *Advances in Neural Information Processing Systems* 27. Ed. by Ghahramani Z. et al. Curran Associates, Inc., 2014, pp. 2861–2869.
- [89] Simon O. Haykin. *Adaptive filter theory*. 5th Edition. Ontario, Canada: Pearson, 2014. ISBN: 978-0-13-267149-1.
- [90] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.
- [91] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos. "Deep fully-connected networks for video compressive sensing". In: *Digital Signal Processing* 72 (2018), pp. 9–18. ISSN: 1051-2004. DOI: [10.1016/j.dsp.2017.09.010](https://doi.org/10.1016/j.dsp.2017.09.010).
- [92] ON Jaspan, R Fleysher, and M L Lipton. "Compressed sensing MRI: a review of the clinical literature". In: *British Journal of Radiology* 88.1056 (2015). ISSN: 1051-2004. DOI: [10.1259/bjr.20150487](https://doi.org/10.1259/bjr.20150487).
- [93] M. Jayawardhana et al. "Compressive sensing for efficient health monitoring and effective damage detection of structures". In: *Mechanical Systems and Signal Processing* 84 (2017), pp. 414–430. ISSN: 0888-3270. DOI: [10.1016/j.ymsp.2016.07.027](https://doi.org/10.1016/j.ymsp.2016.07.027).

- [94] Qingchao Jiang, Xuefeng Yan, and Biao Huang. "Performance-Driven Distributed PCA Process Monitoring Based on Fault-Relevant Variable Selection and Bayesian Inference". In: *IEEE Transactions on Industrial Electronics* 63.1 (Jan. 2016), pp. 377–386. ISSN: 1557-9948. DOI: [10.1109/TIE.2015.2466557](https://doi.org/10.1109/TIE.2015.2466557).
- [95] T. Jiang. "How many entries of a typical orthogonal matrix can be approximated by independent normals?" In: *The Annals of Probability* 34.4 (2006), pp. 1497–1529. DOI: [10.1214/009117906000000205](https://doi.org/10.1214/009117906000000205).
- [96] J. Jin et al. "An Information Framework for Creating a Smart City Through Internet of Things". In: *IEEE Internet of Things Journal* 1.2 (Apr. 2014), pp. 112–121. ISSN: 2327-4662. DOI: [10.1109/JIOT.2013.2296516](https://doi.org/10.1109/JIOT.2013.2296516).
- [97] Iain M. Johnstone. "On the distribution of the largest eigenvalue in principal components analysis". In: *Ann. Statist.* 29.2 (Apr. 2001), pp. 295–327. DOI: [10.1214/aos/1009210544](https://doi.org/10.1214/aos/1009210544).
- [98] I. T. Jolliffe. *Principal components analysis*. Series in Statistics. Springer, 1986. ISBN: 978-0-387-96269-6. DOI: [10.1007/978-1-4757-1904-8](https://doi.org/10.1007/978-1-4757-1904-8).
- [99] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–. URL: <http://www.scipy.org/>.
- [100] S. Kaisler et al. "Big Data: Issues and Challenges Moving Forward". In: *2013 46th Hawaii International Conference on System Sciences*. Jan. 2013, pp. 995–1004. DOI: [10.1109/HICSS.2013.645](https://doi.org/10.1109/HICSS.2013.645).
- [101] Steven Kay. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*. Prentice Hall PTR, 1988. ISBN: 013504135x.
- [102] Ryan Kennedy, Camillo J. Taylor, and Laura Balzano. "Online completion of Ill-conditioned low-rank matrices". In: *2014 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2014*. Institute of Electrical and Electronics Engineers Inc., Feb. 2014, pp. 507–511. ISBN: 9781479970889. DOI: [10.1109/GlobalSIP.2014.7032169](https://doi.org/10.1109/GlobalSIP.2014.7032169).
- [103] D. E. Knuth. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997, p. 232. ISBN: 0-201-89684-2.
- [104] Andrey Kolmogorov. "On the Shannon theory of information transmission in the case of continuous signals". In: *IRE Transactions on Information Theory* 2.4 (1956), pp. 102–108. DOI: [10.1109/TIT.1956.1056823](https://doi.org/10.1109/TIT.1956.1056823).
- [105] T P Krasulina. "Method of stochastic approximation in the determination of the largest eigenvalue of the mathematical expectation of random matrices". In: *Automation and Remote Control* 2 (1970), pp. 215–221.

- [106] T. P. Krasulina. "The method of stochastic approximation for the determination of the least eigenvalue of a symmetrical matrix". In: *USSR Computational Mathematics and Mathematical Physics* 9.6 (Jan. 1969), pp. 189–195. ISSN: 00415553. DOI: [10.1016/0041-5553\(69\)90135-9](https://doi.org/10.1016/0041-5553(69)90135-9).
- [107] K. Kulkarni et al. "ReconNet: Non-Iterative Reconstruction of Images from Compressively Sensed Measurements". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 449–458. DOI: [10.1109/CVPR.2016.55](https://doi.org/10.1109/CVPR.2016.55).
- [108] Nacerredine Lassami, Abdeldjalil Aïssa-El-Bey, and Karim Abed-Meraim. "Low cost sparse subspace tracking algorithms". In: *Signal Processing* 173 (Aug. 2020), p. 107522. ISSN: 01651684. DOI: [10.1016/j.sigpro.2020.107522](https://doi.org/10.1016/j.sigpro.2020.107522).
- [109] Y. Liao et al. "SnowFort: An Open Source Wireless Sensor Network for Data Analytics in Infrastructure and Environmental Monitoring". In: *IEEE Sensors Journal* 14.12 (Dec. 2014), pp. 4253–4263. ISSN: 2379-9153. DOI: [10.1109/JSEN.2014.2358253](https://doi.org/10.1109/JSEN.2014.2358253).
- [110] Ji Lin et al. *MCUNet: Tiny Deep Learning on IoT Devices*. 2020. arXiv: [2007.10319 \[cs.CV\]](https://arxiv.org/abs/2007.10319).
- [111] F. Ling, D. Manolakis, and J. Proakis. "A recursive modified Gram-Schmidt algorithm for least-squares estimation". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34.4 (Aug. 1986), pp. 829–836. ISSN: 0096-3518. DOI: [10.1109/TASSP.1986.1164877](https://doi.org/10.1109/TASSP.1986.1164877).
- [112] Shengpu Liu and Liang Cheng. "Efficient Data Compression in Wireless Sensor Networks for Civil Infrastructure Health Monitoring". In: *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*. Vol. 3. 2006, pp. 823–829. DOI: [10.1109/SAHCN.2006.288567](https://doi.org/10.1109/SAHCN.2006.288567).
- [113] X. Liu et al. "Distributed Sensing for High-Quality Structural Health Monitoring Using WSNs". In: *IEEE Transactions on Parallel and Distributed Systems* 26.3 (Mar. 2015), pp. 738–747. ISSN: 1045-9219. DOI: [10.1109/TPDS.2014.2312911](https://doi.org/10.1109/TPDS.2014.2312911).
- [114] Michael Lustig et al. "Compressed Sensing MRI". In: *IEEE Signal Processing Magazine* 25.2 (2008), pp. 72–82. DOI: [10.1109/MSP.2007.914728](https://doi.org/10.1109/MSP.2007.914728).
- [115] J. P. Lynch, C. R. Farrar, and J. E. Michaels. "Structural health monitoring: technological advances to practical implementations [scanning the issue]". In: *Proceedings of the IEEE* 104.8 (Aug. 2016), pp. 1508–1512. ISSN: 0018-9219. DOI: [10.1109/JPROC.2016.2588818](https://doi.org/10.1109/JPROC.2016.2588818).
- [116] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Access Online via Elsevier, 2008.

- [117] H. Mamaghanian et al. "Compressed Sensing for Real-Time Energy-Efficient ECG Compression on Wireless Body Sensor Nodes". In: *IEEE Transactions on Biomedical Engineering* 58.9 (Sept. 2011), pp. 2456–2466. DOI: [10.1109/TBME.2011.2156795](https://doi.org/10.1109/TBME.2011.2156795).
- [118] M. Mangia, R. Rovatti, and G. Setti. "Rakeness in the Design of Analog-to-Information Conversion of Sparse and Localized Signals". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 59.5 (May 2012), pp. 1001–1014. ISSN: 1558-0806. DOI: [10.1109/TCSI.2012.2191312](https://doi.org/10.1109/TCSI.2012.2191312).
- [119] M. Mangia et al. "Deep Neural Oracles for Short-Window Optimized Compressed Sensing of Biosignals". In: *IEEE Transactions on Biomedical Circuits and Systems* 14.3 (2020), pp. 545–557. DOI: [10.1109/TBCAS.2020.2982824](https://doi.org/10.1109/TBCAS.2020.2982824).
- [120] M. Mangia et al. "Rakeness-Based Design of Low-Complexity Compressed Sensing". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 64.5 (2017), pp. 1201–1213. DOI: [10.1109/TCSI.2017.2649572](https://doi.org/10.1109/TCSI.2017.2649572).
- [121] Mauro Mangia et al. "Adapted compressed sensing: A game worth playing". In: *IEEE Circuits Syst. Mag.* 20.1 (2020), pp. 40–60. ISSN: 1531-636X. DOI: [10.1109/MCAS.2019.2961727](https://doi.org/10.1109/MCAS.2019.2961727).
- [122] Mauro Mangia et al. *Adapted Compressed Sensing for Effective Hardware Implementations: A Design Flow for Signal-Level Optimization of Compressed Sensing Stages*. Springer International Publishing, 2018.
- [123] Mauro Mangia et al. "Coping with saturating projection stages in RMPI-based Compressive Sensing". In: *2012 IEEE int. Symp. Circuits syst.* IEEE, May 2012, pp. 2805–2808. DOI: [10.1109/ISCAS.2012.6271893](https://doi.org/10.1109/ISCAS.2012.6271893).
- [124] Mauro Mangia et al. "Low-power ECG acquisition by compressed sensing with deep neural oracles". In: *2020 2nd IEEE international conference on artificial intelligence circuits and systems (AICAS)*. Aug. 2020, pp. 158–162. DOI: [10.1109/AICAS48895.2020.9073945](https://doi.org/10.1109/AICAS48895.2020.9073945).
- [125] Francesco Marcelloni and Massimo Vecchio. "A Simple Algorithm for Data Compression in Wireless Sensor Networks". In: *IEEE Communications Letters* 12.6 (2008), pp. 411–413. DOI: [10.1109/LCOMM.2008.080300](https://doi.org/10.1109/LCOMM.2008.080300).
- [126] A. Marchioni et al. "Sparse sensing matrix based compressed sensing in low-power ECG sensor nodes". In: *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. Oct. 2017, pp. 1–4. DOI: [10.1109/BIOCAS.2017.8325155](https://doi.org/10.1109/BIOCAS.2017.8325155).
- [127] Alex Marchioni et al. "Subspace Energy Monitoring for Anomaly Detection @Sensor or @Edge". In: *IEEE Internet of Things Journal* 7.8 (2020), pp. 7575–7589. DOI: [10.1109/JIOT.2020.2985912](https://doi.org/10.1109/JIOT.2020.2985912).

- [128] Teodor Vanislavov Marinov, Poorya Mianjy, and Raman Arora. "Streaming Principal Component Analysis in Noisy Settings". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Stockholmsmässan, Stockholm Sweden: PMLR, July 2018, pp. 3413–3422.
- [129] Alvin Martin et al. "The DET curve in assessment of detection task performance". In: *Proc. 5th European Conference on Speech Communication and Technology (Eurospeech 1997)*. 1997, pp. 1895–1898.
- [130] A.M. Mathai and Serge B. Provost. *Quadratic Forms in Random Variables: Theory and Applications*. Statistics: textbooks and monographs. Marcel Dekker, 1992.
- [131] P.E. McSharry et al. "A dynamical model for generating synthetic electrocardiogram signals". In: *IEEE Transactions on Biomedical Engineering* 50.3 (2003), pp. 289–294. DOI: [10.1109/TBME.2003.808805](https://doi.org/10.1109/TBME.2003.808805).
- [132] *MEMS inertial sensor*. LIS344ALH. Rev. 3. STMicroelectronics. Apr. 2008.
- [133] Hagit Messer, Artem Zinevich, and Pinhas Alpert. "Environmental Monitoring by Wireless Communication Networks". In: *Science* 312.5774 (2006), pp. 713–713. ISSN: 0036-8075. DOI: [10.1126/science.1120034](https://doi.org/10.1126/science.1120034).
- [134] Francesco Mezzadri. "How to generate random matrices from the classical compact groups". In: *Notices of the American Mathematical Society* 54.5 (2006), pp. 592–604.
- [135] Yongfeng Miao and Yingbo Hua. "Fast subspace tracking and neural network learning by a novel information criterion". In: *IEEE Transactions on Signal Processing* 46.7 (1998), pp. 1967–1979. ISSN: 1053587X. DOI: [10.1109/78.700968](https://doi.org/10.1109/78.700968).
- [136] A. Mirrashid and A. A. Beheshti. "Compressed Remote Sensing by Using Deep Learning". In: *2018 9th International Symposium on Telecommunications (IST)*. Dec. 2018, pp. 549–552. DOI: [10.1109/ISTEL.2018.8661112](https://doi.org/10.1109/ISTEL.2018.8661112).
- [137] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. "Memory Limited, Streaming PCA". In: *Advances in Neural Information Processing Systems* 26. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 2886–2894.
- [138] G.B. Moody, R.G. Mark, and A.L. Goldberger. "Evaluation of the 'TRIM' ECG data compressor". In: *Proceedings. Computers in Cardiology 1988*. 1988, pp. 167–170. DOI: [10.1109/CIC.1988.72591](https://doi.org/10.1109/CIC.1988.72591).
- [139] P.G. Moschopoulos and W.B. Canada. "The distribution function of a linear combination of chi-squares". In: *Computers & Mathematics with Applications* 10.4 (1984), pp. 383–386. ISSN: 0898-1221. DOI: [10.1016/0898-1221\(84\)90066-X](https://doi.org/10.1016/0898-1221(84)90066-X).

- [140] A. Mousavi and R. G. Baraniuk. "Learning to invert: Signal recovery via Deep Convolutional Networks". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2017, pp. 2272–2276. DOI: [10.1109/ICASSP.2017.7952561](https://doi.org/10.1109/ICASSP.2017.7952561).
- [141] A. Mousavi, A. B. Patel, and R. G. Baraniuk. "A deep learning approach to structured signal recovery". In: *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Sept. 2015, pp. 1336–1343. DOI: [10.1109/ALLERTON.2015.7447163](https://doi.org/10.1109/ALLERTON.2015.7447163).
- [142] D. Needell and J.A. Tropp. "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples". In: *Applied and Computational Harmonic Analysis* 26.3 (2009), pp. 301–321. ISSN: 1063-5203. DOI: [10.1016/j.acha.2008.07.002](https://doi.org/10.1016/j.acha.2008.07.002).
- [143] M. H. Neumann. "A central limit theorem for triangular arrays of weakly dependent random variables, with applications in statistics". In: *ESAIM: Probability and Statistics* 17 (2013), pp. 120–134. DOI: [10.1051/ps/2011144](https://doi.org/10.1051/ps/2011144).
- [144] Erkki Oja. "Principal components, minor components, and linear neural networks". In: *Neural Networks* 5.6 (Nov. 1992), pp. 927–935. ISSN: 08936080. DOI: [10.1016/S0893-6080\(05\)80089-9](https://doi.org/10.1016/S0893-6080(05)80089-9).
- [145] Erkki Oja. "Simplified neuron model as a principal component analyzer". In: *Journal of Mathematical Biology* 15.3 (1982), pp. 267–273. DOI: [10.1007/BF00275687](https://doi.org/10.1007/BF00275687).
- [146] Erkki Oja and Juha Karhunen. "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix". In: *Journal of Mathematical Analysis and Applications* 106.1 (Feb. 1985), pp. 69–84. ISSN: 10960813. DOI: [10.1016/0022-247X\(85\)90131-3](https://doi.org/10.1016/0022-247X(85)90131-3).
- [147] Shmuel Onn and Ishay Weissman. "Generating uniform random vectors over a simplex with implications to the volume of a certain polytope and to multivariate extremes". In: *Annals of Operations Research* 189.1 (2011), pp. 331–342. DOI: [10.1007/s10479-009-0567-7](https://doi.org/10.1007/s10479-009-0567-7).
- [148] C. Paolino et al. "A Practical Architecture for SAR-based ADCs with Embedded Compressed Sensing Capabilities". In: *2019 15th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*. July 2019.
- [149] F. Pareschi et al. "Energy Analysis of Decoders for Rakeness-Based Compressed Sensing of ECG Signals". In: *IEEE Transactions on Biomedical Circuits and Systems* PP.99 (2017), pp. 1–12. ISSN: 1932-4545. DOI: [10.1109/TBCAS.2017.2740059](https://doi.org/10.1109/TBCAS.2017.2740059).
- [150] Fabio Pareschi et al. "Hardware-Algorithms Co-Design and Implementation of an Analog-to-Information Converter for Biosignals Based on Compressed Sensing". In: *IEEE Transactions on Biomedical Circuits and Systems* 10.1 (Feb. 2016), pp. 149–162. ISSN: 1932-4545. DOI: [10.1109/TBCAS.2015.2444276](https://doi.org/10.1109/TBCAS.2015.2444276).

- [151] Chathurdara Sri Nadith Pathirage et al. "Structural damage identification based on autoencoder neural networks and deep learning". In: *Engineering Structures* 172 (2018), pp. 13–28. ISSN: 0141-0296. DOI: [10.1016/j.engstruct.2018.05.109](https://doi.org/10.1016/j.engstruct.2018.05.109).
- [152] Valentino Peluso and Andrea Calimera. "Energy-Driven Precision Scaling for Fixed-Point ConvNets". In: *2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*. 2018, pp. 113–118. DOI: [10.1109/VLSI-SoC.2018.8644902](https://doi.org/10.1109/VLSI-SoC.2018.8644902).
- [153] L. F. Polanía et al. "Exploiting Prior Knowledge in Compressed Sensing Wireless ECG Systems". In: *IEEE Journal of Biomedical and Health Informatics* 19.2 (Mar. 2015), pp. 508–519. DOI: [10.1109/JBHI.2014.2325017](https://doi.org/10.1109/JBHI.2014.2325017).
- [154] L. Prono et al. "Low-power fixed-point compressed sensing decoder with support oracle". In: *2020 IEEE international symposium on circuits and systems (ISCAS)*. Oct. 2020, pp. 1–5. DOI: [10.1109/ISCAS45731.2020.9180502](https://doi.org/10.1109/ISCAS45731.2020.9180502).
- [155] S. Rangan. "Generalized approximate message passing for estimation with random linear mixing". In: *2011 IEEE International Symposium on Information Theory Proceedings*. July 2011, pp. 2168–2172. DOI: [10.1109/ISIT.2011.6033942](https://doi.org/10.1109/ISIT.2011.6033942).
- [156] M. Rani, S. B. Dhok, and R. B. Deshmukh. "A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications". In: *IEEE Access* 6 (2018), pp. 4875–4894. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2018.2793851](https://doi.org/10.1109/ACCESS.2018.2793851).
- [157] Md Abdur Razzaque, Chris Bleakley, and Simon Dobson. "Compression in Wireless Sensor Networks: A Survey and Comparative Evaluation". In: *ACM Transaction on Sensor Networks* 10.1 (Dec. 2013), 5:1–5:44. ISSN: 1550-4859. DOI: [10.1145/2528948](https://doi.org/10.1145/2528948).
- [158] Jesse Read et al. "Classifier chains for multi-label classification". In: *Machine Learning* 85.3 (June 2011), p. 333. ISSN: 1573-0565. DOI: [10.1007/s10994-011-5256-5](https://doi.org/10.1007/s10994-011-5256-5).
- [159] Miguel R. D. Rodrigues et al. "Rate-distortion trade-offs in acquisition of signal parameters". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 6105–6109. DOI: [10.1109/ICASSP.2017.7953329](https://doi.org/10.1109/ICASSP.2017.7953329).
- [160] Christopher M. Sadler and Margaret Martonosi. "Data Compression Algorithms for Energy-Constrained Devices in Delay Tolerant Networks". In: *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*. SenSys '06. Boulder, Colorado, USA: Association for Computing Machinery, 2006, 265–278. ISBN: 1595933433. DOI: [10.1145/1182807.1182834](https://doi.org/10.1145/1182807.1182834).
- [161] Bernhard Schölkopf et al. "Support Vector Method for Novelty Detection". In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. NIPS'99. Denver, CO: MIT Press, 1999, 582–588.

- [162] Robert Schreiber and Charles Van Loan. "A Storage-Efficient WY Representation for Products of Householder Transformations". en. In: *SIAM Journal on Scientific and Statistical Computing* 10.1 (Jan. 1989), pp. 53–57. ISSN: 0196-5204, 2168-3417. DOI: [10.1137/0910005](https://doi.org/10.1137/0910005). (Visited on 03/30/2021).
- [163] Erich Schubert et al. "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN". In: *ACM Trans. Database Syst.* 42.3 (July 2017), 19:1–19:21. ISSN: 0362-5915. DOI: [10.1145/3068335](https://doi.org/10.1145/3068335).
- [164] Carmelo Scuro et al. "IoT for structural health monitoring". In: *IEEE Instrumentation Measurement Magazine* 21.6 (2018), pp. 4–14. DOI: [10.1109/MIM.2018.8573586](https://doi.org/10.1109/MIM.2018.8573586).
- [165] Vanessa Senger and Ronald Tetzlaff. "New Signal Processing Methods for the Development of Seizure Warning Devices in Epilepsy". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 63.5 (May 2016), pp. 609–616. ISSN: 15498328. DOI: [10.1109/TCSI.2016.2553278](https://doi.org/10.1109/TCSI.2016.2553278).
- [166] W. Shi et al. "Deep networks for compressed image sensing". In: *2017 IEEE International Conference on Multimedia and Expo (ICME)*. July 2017, pp. 877–882. DOI: [10.1109/ICME.2017.8019428](https://doi.org/10.1109/ICME.2017.8019428).
- [167] Weisong Shi et al. "Edge Computing: Vision and Challenges". In: *IEEE Internet of Things Journal* 3.5 (2016), pp. 637–646. DOI: [10.1109/JIOT.2016.2579198](https://doi.org/10.1109/JIOT.2016.2579198).
- [168] Nir Shlezinger, Yonina C. Eldar, and Miguel R. D. Rodrigues. "Hardware-Limited Task-Based Quantization". In: *IEEE Transactions on Signal Processing* 67.20 (2019), pp. 5223–5238. DOI: [10.1109/TSP.2019.2935864](https://doi.org/10.1109/TSP.2019.2935864).
- [169] Mahsa Shoaran et al. "Compact Low-Power Cortical Recording Architecture for Compressive Multichannel Data Acquisition". In: *IEEE Transactions on Biomedical Circuits and Systems* 8.6 (Dec. 2014), pp. 857–870. ISSN: 1932-4545. DOI: [10.1109/TBCAS.2014.2304582](https://doi.org/10.1109/TBCAS.2014.2304582).
- [170] Noam Slonim and Naftali Tishby. "Agglomerative Information Bottleneck". In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. NIPS'99. MIT Press, 1999, 617–623. DOI: [10.5555/3009657.3009745](https://doi.org/10.5555/3009657.3009745).
- [171] Hoon Sohn. "Effects of environmental and operational variability on structural health monitoring". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365.1851 (2007), pp. 539–560. DOI: [10.1098/rsta.2006.1935](https://doi.org/10.1098/rsta.2006.1935).
- [172] Chang Song et al. "A quantization-aware regularized learning method in multilevel memristor-based neuromorphic computing system". In: *2017 IEEE 6th Non-Volatile Memory Systems and Applications Symposium (NVMSA)*. 2017, pp. 1–6. DOI: [10.1109/NVMSA.2017.8064465](https://doi.org/10.1109/NVMSA.2017.8064465).

- [173] A. Stanford-Clark and A. Nipper. *MQ Telemetry Transport (MQTT)*. URL: <http://mqtt.org/>.
- [174] B. Sun et al. "A Deep Learning Framework of Quantized Compressed Sensing for Wireless Neural Recording". In: *IEEE Access* 4 (2016), pp. 5169–5178. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2016.2604397](https://doi.org/10.1109/ACCESS.2016.2604397).
- [175] Cheng Tang. "Exponentially convergent stochastic k-PCA without variance reduction". In: *Advances in Neural Information Processing Systems* 32. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 12393–12404.
- [176] Takeshi Terao, Katsuhisa Ozaki, and Takeshi Ogita. "LU-Cholesky QR algorithms for thin QR decomposition". en. In: *Parallel Computing* 92 (Apr. 2020), p. 102571. ISSN: 0167-8191. DOI: [10.1016/j.parco.2019.102571](https://doi.org/10.1016/j.parco.2019.102571). (Visited on 03/30/2021).
- [177] Lucas A. Thomaz et al. "Anomaly Detection in Moving-Camera Video Sequences Using Principal Subspace Analysis". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.3 (Mar. 2018), pp. 1003–1015. ISSN: 15498328. DOI: [10.1109/TCSI.2017.2758379](https://doi.org/10.1109/TCSI.2017.2758379).
- [178] Naftali Tishby, Fernando C. Pereira, and William Bialek. "The information bottleneck method". In: *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*. 1999, 368–377.
- [179] C. Arcadius Tokognon et al. "Structural Health Monitoring Framework Based on Internet of Things: A Survey". In: *IEEE Internet of Things Journal* 4.3 (June 2017), pp. 619–635. ISSN: 2327-4662. DOI: [10.1109/JIOT.2017.2664072](https://doi.org/10.1109/JIOT.2017.2664072).
- [180] Nilesh Tripuraneni et al. "Averaging Stochastic Gradient Descent on Riemannian Manifolds". In: ed. by Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet. Vol. 75. *Proceedings of Machine Learning Research*, 2018, pp. 1–38.
- [181] Joel A. Tropp and Anna C. Gilbert. "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit". In: *IEEE Transactions on Information Theory* 53.12 (2007), pp. 4655–4666. DOI: [10.1109/TIT.2007.909108](https://doi.org/10.1109/TIT.2007.909108).
- [182] B. P. Welford. "Note on a Method for Calculating Corrected Sums of Squares and Products". In: *Technometrics* 4.3 (Aug. 1962), pp. 419–420. ISSN: 00401706. DOI: [10.2307/1266577](https://doi.org/10.2307/1266577).
- [183] M. J. Whelan, M. V. Gangone, and K. D. Janoyan. "Highway Bridge Assessment Using an Adaptive Real-Time Wireless Sensor Network". In: *IEEE Sensors Journal* 9.11 (Nov. 2009), pp. 1405–1413. ISSN: 2379-9153. DOI: [10.1109/JSEN.2009.2026546](https://doi.org/10.1109/JSEN.2009.2026546).
- [184] Bernard Widrow and Marcian E. Hoff. "Adaptive Switching Circuits". In: *1960 IRE WESCON Convention Record* (1960). Reprinted in *Neurocomputing* MIT Press, 1988 ., pp. 96–104.

- [185] Jianping Xu, Yiming Pi, and Zongjie Cao. "Optimized Projection Matrix for Compressive Sensing". In: *EURASIP Journal on Advances in Signal Processing* 2010.1 (2010), p. 560349. ISSN: 1687-6180. DOI: [10.1155/2010/560349](https://doi.org/10.1155/2010/560349).
- [186] Bin Yang. "Projection Approximation Subspace Tracking". In: *IEEE Transactions on Signal Processing* 43.1 (1995), pp. 95–107. DOI: [10.1109/78.365290](https://doi.org/10.1109/78.365290).
- [187] Bin Yang. "Subspace tracking based on the projection approach and the recursive least squares method". In: *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Minneapolis, MN, USA: Institute of Electrical and Electronics Engineers (IEEE), Apr. 1993, pp. 145–148. DOI: [10.1109/icassp.1993.319615](https://doi.org/10.1109/icassp.1993.319615).
- [188] Puyudi Yang, Cho-Jui Hsieh, and Jane-Ling Wang. *History PCA: A New Algorithm for Streaming PCA*. 2018. arXiv: [1802.05447](https://arxiv.org/abs/1802.05447) [stat.ML].
- [189] Nick Yeung et al. "Detection of synchronized oscillations in the electroencephalogram: An evaluation of methods". In: *Psychophysiology* 41.6 (2004), pp. 822–832. DOI: [10.1111/j.1469-8986.2004.00239.x](https://doi.org/10.1111/j.1469-8986.2004.00239.x).
- [190] Ting-Hua Yi, Hai-Bin Huang, and Hong-Nan Li. "Development of sensor validation methodologies for structural health monitoring: A comprehensive review". In: *Measurement* 109 (2017), pp. 200–214. ISSN: 0263-2241. DOI: [10.1016/j.measurement.2017.05.064](https://doi.org/10.1016/j.measurement.2017.05.064).
- [191] S. Yin et al. "A Review on Basic Data-Driven Approaches for Industrial Process Monitoring". In: *IEEE Transactions on Industrial Electronics* 61.11 (Nov. 2014), pp. 6418–6428. ISSN: 1557-9948. DOI: [10.1109/TIE.2014.2301773](https://doi.org/10.1109/TIE.2014.2301773).
- [192] Juhwan Yoo et al. "A 100MHz-2GHz 12.5x sub-Nyquist rate receiver in 90nm CMOS". In: *2012 IEEE Radio Frequency Integrated Circuits Symposium*. June 2012, pp. 31–34. DOI: [10.1109/RFIC.2012.6242225](https://doi.org/10.1109/RFIC.2012.6242225).
- [193] B. Yu et al. "Real-time FPGA-based multichannel spike sorting using hebbian eigenfilters". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 1.4 (Dec. 2011), pp. 502–515. ISSN: 21563357. DOI: [10.1109/JETCAS.2012.2183430](https://doi.org/10.1109/JETCAS.2012.2183430).
- [194] W. Yu et al. "A Global Manufacturing Big Data Ecosystem for Fault Detection in Predictive Maintenance". In: *IEEE Transactions on Industrial Informatics* 16.1 (Jan. 2020), pp. 183–192. ISSN: 1941-0050. DOI: [10.1109/TII.2019.2915846](https://doi.org/10.1109/TII.2019.2915846).
- [195] Wei Yu et al. "A Survey on the Edge Computing for the Internet of Things". In: *IEEE Access* 6 (2018), pp. 6900–6919. DOI: [10.1109/ACCESS.2017.2778504](https://doi.org/10.1109/ACCESS.2017.2778504).
- [196] D. Zhang and L. Balzano. "Global convergence of a Grassmannian gradient descent algorithm for subspace estimation". In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Ed. by Arthur Gretton and Christian C. Robert. Cadiz, Spain: PMLR, 2016, pp. 1460–1468. URL: <http://proceedings.mlr.press/v51/zhang16b.html>.

- [197] J. Zhang et al. "An Efficient and Compact Compressed Sensing Microsystem for Implantable Neural Recordings". In: *IEEE Transactions on Biomedical Circuits and Systems* 8.4 (Aug. 2014), pp. 485–496. ISSN: 1932-4545. DOI: [10.1109/TBCAS.2013.2284254](https://doi.org/10.1109/TBCAS.2013.2284254).
- [198] J. Zhang et al. "Energy-Efficient ECG Compression on Wireless Biosensors via Minimal Coherence Sensing and Weighted ℓ_1 Minimization Reconstruction". In: *IEEE Journal of Biomedical and Health Informatics* 19.2 (Mar. 2015), pp. 520–528. ISSN: 2168-2194. DOI: [10.1109/JBHI.2014.2312374](https://doi.org/10.1109/JBHI.2014.2312374).
- [199] Jian Zhang and Bernard Ghanem. "ISTA-Net: Interpretable Optimization-Inspired Deep Network for Image Compressive Sensing". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. June 2018, pp. 1828–1837. DOI: [10.1109/CVPR.2018.00196](https://doi.org/10.1109/CVPR.2018.00196).
- [200] Pei Zhang et al. "Privacy-preserving anomaly detection across multi-domain networks". In: *Proceedings - 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2012*. 2012, pp. 1066–1070. DOI: [10.1109/FSKD.2012.6234272](https://doi.org/10.1109/FSKD.2012.6234272).
- [201] Y. Zhang et al. "A Review of Compressive Sensing in Information Security Field". In: *IEEE Access* 4 (2016), pp. 2507–2519. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2016.2569421](https://doi.org/10.1109/ACCESS.2016.2569421).
- [202] Z. Zhang et al. "Compressed Sensing for Energy-Efficient Wireless Telemonitoring of Noninvasive Fetal ECG Via Block Sparse Bayesian Learning". In: *IEEE Transactions on Biomedical Engineering* 60.2 (Feb. 2013), pp. 300–309. ISSN: 0018-9294. DOI: [10.1109/TBME.2012.2226175](https://doi.org/10.1109/TBME.2012.2226175).
- [203] Zichao Zhao et al. "A Novel Framework of Three-Hierarchical Offloading Optimization for MEC in Industrial IoT Networks". In: *IEEE Transactions on Industrial Informatics* 16.8 (2020), pp. 5424–5434. DOI: [10.1109/TII.2019.2949348](https://doi.org/10.1109/TII.2019.2949348).
- [204] Xiao Long Zhu et al. "Adaptive nonlinear PCA algorithms for blind source separation without prewhitening". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 53.3 (Mar. 2006), pp. 745–753. ISSN: 10577122. DOI: [10.1109/TCSI.2005.858489](https://doi.org/10.1109/TCSI.2005.858489).
- [205] Y. Zigei, A. Cohen, and A. Katz. "The weighted diagnostic distortion (WDD) measure for ECG signal compression". In: *IEEE Trans. on Biom. Eng.* 47.11 (Nov. 2000), pp. 1422–1430. ISSN: 0018-9294. DOI: [10.1109/TBME.2000.880093](https://doi.org/10.1109/TBME.2000.880093).