University of Bath

PHD

Acoustic Impacts of Seismic Surveys in Complex Underwater Environments

Jiménez Arranz, Guillermo

Award date:
2023

Awarding institution:
University of Bath

Link to publication

Alternative formats
If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

Copyright of this thesis rests with the author. Access is subject to the above licence, if given. If no licence is specified above, original content in this thesis is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC-ND 4.0) Licence (https://creativecommons.org/licenses/by-nc-nd/4.0/). Any third-party copyright material present remains the property of its respective owner(s) and is licensed under its existing terms.

Take down policy
If you consider content within Bath's Research Portal to be in breach of UK law, please contact: openaccess@bath.ac.uk with the details. Your claim will be investigated and, where appropriate, the item will be removed from public view as soon as possible.

Download date: 04. May. 2023

Acoustic Impacts of Seismic Surveys in Complex Underwater Environments

A thesis submitted by

Guillermo Jiménez Arranz

for the degree of

Doctor of Philosophy

of the

University of Bath

Department of Physics

June 2022

COPYRIGHT

Attention is drawn to the fact that copyright of this thesis/portfolio rests with the author and copyright of any previously published materials included may rest with third parties. A copy of this thesis/portfolio has been supplied on condition that anyone who consults it understands that they must not copy it or use material from it except as licenced, permitted by law or with the consent of the author or other copyright owners, as applicable.

Author´s signature …….

(GUILLERMO JIMÉNEZ ARRANZ)

Summary

The oil and gas industry employs powerful, low-frequency impulsive sources for exploration during seismic surveys. These sources, known as seismic air gun arrays, have the ability to propagate long distances. Seismic surveys can last for weeks and happen simultaneously at multiple locations around the world. These operations have the potential to disturb and displace marine mammals and other forms of marine life.

There is a number of mitigation strategies that can be used to reduce the impact and exposure of marine animals to noise. For seismic surveys, the most cost-effective method consists in implementing a real-time monitoring plan around a predefined exclusion zone. If a marine mammal enters that area, the operations stop as requested by the Marine Mammal Observer (MMO) or Passive Acoustic Monitoring (PAM) technician who detected and identified the animal. Nowadays, these exclusion zones are calculated from sound field measurements or models, or a combination of both.

For the purpose of sound source verification (SSV), a model of the sound field is required before the survey starts. The pre-survey simulation of the sound field is obtained as a combination of an *air gun model*, which simulates the acoustic output of the seismic source, and a *propagation model*, which simulates the pressure attenuation with distance. The pre-survey simulations are used to calculate the exclusion zone that is to be monitored before the sound mapping survey can gather the data necessary to verify the model. Once the data has been collected, a report is issued within 24-72 h. The report summarises the results of the model validation and provides an updated exclusion zone.

In this context, there are three essential tools needed for noise impact assessment: a source model and propagation model for the pre-survey sound field simulations, and a software package that can generate a map of sound pressure levels from raw audio and position data. However, the tools currently available have some fundamental limitations: (1) commercial source models of air gun arrays are costly and are not designed for noise impact assessment, and (2) sound field processing is typically carried out with non-standardised software routines developed by the companies providing the SSV services.

In this thesis we present the design and implementation of two pieces of software: (1) a **package for the computation of the sound field** from raw audio and position data, and (2) a **calibrated air gun array model** that can compete with commercial software. The purpose of these tools is to provide efficient, accurate and free-access routines that can be used for the characterisation of the acoustic source, the sound field, and the propagation environment in the context of seismic surveys.

Para Inma, José y Álvaro,

Para Rosa y Simon,

Para Naomi,

Por estar ahí.

Acknowledgements

I would like to thank, first of all, my supervisors **Dr Philippe Blondel** and **Professor Chris Budd**, for their experience, ideas, and guidance, for supporting me at all stages of this process and giving me that tap on the back when I needed it most.

Thanks to **Roy Wyatt**, **Simon Cole**, and **Mark Burnett** from Seiche for believing in me and giving me the opportunity to work on many exciting projects. I am grateful for the flexibility and constant support they offered to help me find a balance between life, studies, and work. Thanks to **Seiche** for financing the registration fees and granting the use of raw audio and navigation data for this thesis.

Thanks to **Mike Jenkerson** from ExxonMobil and **David Hedgeland** from BP, for providing me with the Svein Vaage dataset and for permitting its use in this thesis for the calibration of the air gun model.

Thanks to the team at the KTP office (**Juliet Mather**, **Izaro López García**, **Jun Zhang**, and especially **Marie Morley**) who welcomed me to the University of Bath with open arms, the place that became my second home for many years.

Thanks to Professor **Paul Lepper** from Loughborough University, for the technical guidance on underwater acoustics during my first few weeks as KTP Associate, and for showing me that acoustic modelling doesn't need to be difficult. Thanks to **Nathan Merchant** and **Adrian Farcas** from CEFAS, **Ozkan Sertlek** from TU Delft, and many other excellent researchers that I have the privilege to know in person, whose work has inspired me to learn and contribute to the field of underwater acoustics.

Thanks to **Stephen Cook**, **Nikhil Banda**, **Jon Narramore**, **Chris Pierpoint**, **Lorenzo Scala**, **Kelly Hudson**, **Marek Gmitro**, **Jan Bujalka**, **Ladipo "Dipo" Baruwa, Stephanie Barnicoat, Kelly Holder**, **Simon Denis**, **Benoît Bergès**, and all the colleagues I've worked with during my time at Seiche, for the positive energy, passion, and team spirit. Its people are what makes Seiche the great company it is. Especial mention to **James Mitchell** and **Dario Chiantello**, friends, housemates and colleagues, for the moments shared and for making life in a remote village in Devon´s countryside unique and fun. Thanks to Dario for the conversations about the thesis; sharing ideas, experiences and frustrations between PhD students was a great remedy against low motivation and "tunnel vision".

Thanks to **Carolin Villforth** for all the help she offered me in this last year, for being incredibly supportive along the process and getting me that extra time I needed.

Thanks to my parents **José Luis** and **Inmaculada**, my brother **Álvaro**, my partner **Naomi**, and my British family (**Rosa**, **Simon**, **Sue**, and **Richard**) for being the pillars that supported me every hour of this long and exciting journey.

Table of Contents

Summary iii

Acknowledgements vii

Table of Contents ix

Publications xv

Glossary of Acronyms xvii

Chapter 1 Introduction 1

Chapter 2 Background 3

2.1 Preliminary Concepts 4

2.1.1 The Wave Equation 4

2.1.2 Intensity, Power and Pressure 6

2.1.3 Source Level and Propagation Loss 8

2.2 Underwater Sound Propagation: The Water Column 10

2.2.1 Global Variation of Sea Water Properties 10

2.2.2 Multipath Propagation 13

2.2.3 Propagation Losses 17

2.3 Underwater Sound Propagation: The Seafloor 19

2.3.1 Morphology of the Seafloor 19

2.3.2 Geology of the Seafloor 21

2.3.3 Propagation in a Fluid-Fluid Interface 21

2.3.4 Propagation in a Fluid-Elastic Interface 26

2.3.5 Propagation in a Fluid Multi-Layered Medium 29

2.3.6 Propagation in an Elastic Multi-Layered Interface 32

2.4 Seismic Exploration 34

2.4.1 Seismic Surveys 35

2.4.2 The Air Gun 36

2.5 Impact of Seismic Noise on Marine Fauna 39

2.5.1 General Effects of Noise 39

2.5.2 Impact of Seismic Air Guns 41

2.5.3 Noise Mitigation 42

2.6 Sound Source Verification 43
2.6.1 Definition and Evolution of SSV 43
2.6.2 Data Acquisition Platforms 44
2.7 Underwater Acoustic Modelling 50
2.7.1 Air Gun Models 50
2.7.2 Sound Propagation Models 51
Chapter 3 Sound Field Processing 57
3.1 Introduction 58
3.2 General Features 61
3.3 Software Structure 64
3.3.1 Flow Chart 64
3.3.2 Program Files 65
3.3.3 Audio and Position 67
3.3.4 Processing Block 70
3.4 Core Modules 71
3.4.1 General Module Structure 71
3.4.2 Audio Import 72
3.4.3 Audio Detect 72
3.4.4 Audio Process 76
3.4.5 Receiver Import 77
3.4.6 Source Import 79
3.4.7 Navigation Process 81
3.5 ABACUS Example 82
3.5.1 Structure and Execution 83
3.5.2 Audio and Position Data 83
3.5.3 Configuration 84
3.5.4 Output Data 90
Chapter 4 A Neyman-Pearson Algorithm for Detecting Seismic Pulses 97
4.1 Introduction 98
4.2 Detection Theory 100
4.3 Aspects of Detection Performance 102
4.3.1 Deflection Coefficient 102
4.3.2 Energy and Variance as Test Statistics 102
4.3.3 Parameters Contributing to the Detection Performance 103

4.3.4 Visualisation of Detection Performance ... 104
4.3.5 Asymptotics ... 106
4.3.6 Gaussian vs Non-Gaussian ... 107
4.3.7 The High-Dimensional Problem ... 108
4.4 The Covariance Matrix and its Decomposition ... 109
4.4.1 Properties of the Covariance Matrix ... 109
4.4.2 Eigenvalue and Eigenvector Decomposition ... 110
4.4.3 Modified Cholesky Decomposition ... 111
4.4.4 Latent Factor Models ... 112
4.5 Methods for High-Dimensional Covariance Estimation ... 112
4.5.1 Covariance Estimation Methods ... 112
4.5.2 Estimator Performance: Loss, Risk and Bias-Variance Tradeoff ... 113
4.5.3 Penalised Regression ... 114
4.5.4 Shrinking ... 116
4.5.5 Banding ... 117
4.5.6 Tapering ... 117
4.5.7 Thresholding ... 118
4.5.8 Maximum Likelihood ... 119
4.5.9 Sparse Gaussian Graphical Models ... 120
4.5.10 The Graphical Lasso ... 120
4.6 Energy Detector ... 121
4.7 Estimator-Correlator with White Gaussian Noise ... 125
4.8 Estimator-Correlator with “Coloured” Gaussian Noise ... 131
4.9 Behaviour of Detectors with Target and Non-Target Signals ... 136
4.9.1 Energy Detector ... 138
4.9.2 Estimator-Correlator in WGN ... 138
4.9.3 Estimator-Correlator in CGN ... 141
4.10 Maximising Detection Performance and Speed ... 146
4.10.1 Window Length ... 147
4.10.2 Filtering ... 149
4.10.3 Resampling ... 152
4.11 Improving the Energy Detector’s Performance with Filtering ... 155
4.11.1 Simple Correction for the PDF Equations of the Energy Detector ... 155
4.11.2 Proposed General Form for the PDF Equations of the Energy Detector ... 155

4.11.3 Calculating the Noise Energy Ratio q_w ... 157
4.11.4 Application of the General Form of the Energy Detector ... 157
4.12 Estimating the Covariance of Filtered Signals with LOOC ... 158
4.12.1 Solutions to the High-Dimensional Problem ... 158
4.12.2 Covariance Shrinkage with LOOC ... 162
4.12.3 The Shrinkage Target ... 165
4.12.4 Application of LOOC Estimation ... 166
4.13 Implementation of the NP Detectors in ABACUS ... 173
4.13.1 Pre-Processing ... 173
4.13.2 Audio Processing ... 176
4.13.3 Detection Algorithm ... 177
4.14 Performance of the NP Detectors with Airgun Pulses ... 178
Chapter 5 Seismic Air Gun Model ... 183
5.1 Introduction ... 184
5.2 Air Gun Bubble Oscillation ... 185
5.3 Pressure at the Bubble Wall ... 187
5.3.1 Mass Transfer of Air due to Port Throttling ... 190
5.3.2 Mass Transfer of Water Vapour ... 194
5.3.3 Variation of Bubble Volume and Effect of Air Gun Body ... 197
5.3.4 Variation of Bubble Temperature ... 198
5.3.5 Heat Transfer between Bubble and Water ... 200
5.3.6 Heat Transfer from Condensation-Vaporisation ... 202
5.3.7 Turbulence and Effective Viscosity ... 203
5.3.8 Rise of the Bubble in Water ... 203
5.4 Initial Conditions ... 204
5.5 Radiated Acoustic Field ... 205
5.6 Bubble Interactions ... 207
5.7 Parameter Optimisation with Simulated Annealing ... 209
5.7.1 Introduction to Simulated Annealing ... 209
5.7.2 Implementation of the Classic Simulated Annealing Algorithm ... 210
5.8 RAGAS – Routines for Air Gun Array Simulation ... 212
5.8.1 Introduction ... 212
5.8.2 General Features ... 213
5.8.3 Software Structure ... 214

5.9 Effects of Model Parameters on the Bubble Motion ... 220
5.9.1 Mass Transfer Rate of Air $dmadt$ ... 221
5.9.2 Mass Transfer Rate of Water Vapour $dmvdt$ ... 226
5.9.3 Heat Transfer Rate through Conduction $dQw'dt$ ... 226
5.9.4 Heat Transfer Rate from Latent Heat $dQvdt$ ... 227
5.9.5 Temperature Change from Joule-Thomson Effect $dTbdt$ ... 228
5.9.6 Bubble Raise Speed $dzbdt$ ... 228
5.9.7 Effective Viscosity ... 229
5.9.8 Air Gun Presence $Vgun$ ... 230
5.9.9 Generator-Injector ... 230
5.9.10 Air Gun Clusters ... 231
5.10 Calibration of RAGAS with Simulated Annealing ... 232
5.10.1 Svein Vaage Dataset ... 232
5.10.2 Calibration Methodology ... 233
5.10.3 Calibration Results ... 234
5.10.4 Calibrated Model ... 237
Chapter 6 Conclusions ... 243
References ... 245
Appendix A ... 261
A.1 Configuration Database ... 263
A.1.1 Global ... 263
A.1.2 Audio Import ... 264
A.1.3 Audio Detect ... 264
A.1.4 Audio Process ... 267
A.1.5 Receiver Import ... 267
A.1.6 Source Import ... 268
A.1.7 Navigation Process ... 270
A.1.8 Tables ... 271
A.2 Audio Database ... 273
A.3 Detection Database ... 274
A.3.1 Raw Score Data ... 274
A.3.2 Covariance Data ... 275
A.3.3 Eigendata ... 276
A.3.4 Performance Data ... 277

A.4 Navigation Database 278
A.5 Acoustic Database 280
Appendix B 285
B.1 Read Audio 287
B.2 Resample Audio 287
B.3 Filter Audio 288
B.4 Zero-Padding before Filtering Short, Band-Limited Signals 290
B.5 Digital Filtering (Single) 296
B.6 FFT Filtering (Single) 297
B.7 FFT Filtering (Bank) 298
B.8 Digital Filtering (Bank) Package 298
B.9 Covariance Estimation 299
B.10 Read GPS 301
B.11 Read AIS 302
B.12 Read P190 302
B.13 Distance & Bearing 302
Appendix C 305
C.1 Perturbation Algorithm 307
C.2 Cost Function 307
C.3 Acceptance Criteria 307
C.4 Cooling Schedule 308
C.5 Initial Conditions 309
C.6 Stop Criteria 311

Publications

The publications below were *authored* or *co-authored* by the author of this thesis.

Reports

Jiménez-Arranz, G., Banda, N., Cook, S., and Wyatt, R. (2020). *Review on existing data on underwater sounds produced by the oil and gas industry*. Technical report P783, prepared by Seiche Ltd for E&P Sound & Marine Life (JIP), pp. 182. [online] https://gisserver.intertek.com/JIP/DMS/ProjectReports/Cat1/JIP-Proj1.4.2_Review_on_Noise_from_Industrial_Sources.pdf (Accessed 21 May 2022).

Jiménez-Arranz, G., Banda, N., Cook, S., and Wyatt, R. (2020). *Review on existing data on underwater sounds from pile driving activities*. Technical report P784, prepared by Seiche Ltd for E&P Sound & Marine Life (JIP), pp. 82. [online] https://gisserver.intertek.com/JIP/DMS/ProjectReports/Cat1/JIP-Proj1.4.2_Review_on_Pile_Driving.pdf (Accessed 21 May 2022)

Verfuss, U.K., Aniceto, A. S., Biuw, M., Fielding, S., Gillespie, D., Harris, D., **Jiménez, G.**, Johnston, P., Plunkett, R., Sivertsen, A., Solbø, A., Storvold, R., and Wyatt, R. (2016) *Literature review: Understanding the current state of autonomous technologies to improve/expand observation and detection of marine species*. Report prepared by SMRU Consulting for The Joint Industry Programme (JIP) on E&P Sound and Marine Life, Rep. SMRUC-2015-015, JIP ref. 5.4. [online] https://gisserver.intertek.com/JIP/DMS/ProjectReports/Cat5/JIP-Proj5.4_Final%20LiteratureReview.pdf [Accessed 13 Feb 2021]

Peer-Reviewed Papers

Jiménez-Arranz, G., Hedgeland, D., Cook, S., Banda, N., Johnson, P., and Oliver, E. (2019). "Acoustic Characterisation of a Mobile Offshore Drilling Unit". Proc. Mtgs. Acoust., 37(1), 11 p. Paper for *The Effects of Noise On Aquatic Life, Den Haage (Netherlands)*, 7th 12th July 2019. doi: 10.1121/2.0001193

Verfuss, U.K., Aniceto, A.S., Harris, D.V, Gillespie, D., Fielding, S., **Jiménez, G.**, Johnston, P., Sinclair, R., Sivertsen, A., Solbø, Storvold, R., Biuw, M., and Wyatt, R. (2019). "A review of unmanned vehicles for the detection and monitoring of marine fauna". Mar. Pollut. Bull. 140, p. 17-29. doi: 10.1016/j.marpolbul.2019.01.009

Conference Papers

Blondel, P., **Jiménez, G.**, and Wyatt, R. (2016) "Seismic surveys in complex environments: Effects of environmental variability on sound propagation and mitigation practice". *Proc. Inst. Acoust.*, 38 (3), pp. 16-23. Acoustics and Environmental Variability, Fluctuations and Coherence, 12-13 December 2016, Cambridge (UK), pp. 16-23.

Posters

Jiménez, G., Hedgeland, D., Cook, S., Banda, N., Johnston, P., and Oliver, E. *Sound level analysis for an offshore oil and gas drilling activity*. [Poster Presentation] 5th International Conference on the Effects on Noise on Aquatic Life, 7-12 July, Den Haag, The Netherlands. doi: 10.13140/RG.2.2.17494.32321

Jiménez, G., Blondel, P., Heath, B., and Wyatt, R. (2016). *Seismic surveys in complex environments – Analysing the variability of exclusion zones with field measurements and models* [Poster Presentation]. 4th International Conference on the Effects on Noise on Aquatic Life, 10-16 July, Dublin, Ireland. doi: 10.13140/RG.2.2.22946.91841

Glossary of Acronyms

ADC	Analog to Digital Converter
ADCP	Acoustic Doppler Current Profiler
ADD	Acoustic Deterrent Device
AE	Analytical Expression
AFC	Acoustic Field Characterisation
AFV	Acoustic Field Verification
AHD	Acoustic Harassment Device
AMAR	Autonomous Multichannel Acoustic Recorder
ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
BDB	Broadband
BHD	Backhoe Dredge
BLD	Bucket Ladder Dredge
BPF	Band-Pass
CDF	Cumulative Density Function
CGN	Correlated ("Coloured") Gaussian Noise
CGS	Correlated ("Coloured") Gaussian Signal
CMST	Centre of Marine Science and Technology
CPA	Closest Point of Approach
CPB	Constant Percentage Bandwidth
CSD	Cutter Suction Dredge
CSV	Comma Separated Values. Extension of "CSV" files

D	Detection
DAQ	Data Acquisition
DOF	Degrees Of Freedom
dBFS	Decibels Full Scale
DOF	Degrees of Freedom
DREP	Defence Research Establishment Pacific
EF	Energy Flux
EIA	Environmental Impact Assessment
FA	False Alarm
FD	Finite Differences
FE	Finite Elements
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GD	Grab Dredge, Group Delay
GI	Generator-Injector
GLRT	Generalised Likelihood Ratio Test
GUI	Graphical User Interface
HPF	High-Pass Filter
IID	Independent and Identically Distributed
IIR	Infinite Impulse Response
I/O	Input/Output
IOGP	International association of Oil and Gas Producers
ITTC	International Towing Tank Conference
JIP	Joint Industry Program
JSON	JavaScript Object Notation. Extension of "JSON" files.
JTE	Joule-Thomson Effect

LBL	Long Baseline
LLS	Linear Least Squares
LPF	Low-Pass Filter
LR	Lasso Regression
LRT	Likelihood Ratio Test
MAT	Extension of MATLAB M-file.
MBES	Multi-Beam Echosounder
MC	Monte Carlo
MCD	Modified Cholesky Decomposition
ME	Multipath Expansion
MMO	Marine Mammal Observer
MMSI	Maritime Mobile Service Identity
MSE	Mean Squared Error
MSFD	Marine Strategy Framework Directive
NIRL	Normal Incidence Reflection Loss
NM	Normal Modes
NMEA	National Marine Electronics Association
NMFS	National Marine Fisheries Service
NP	Neyman-Pearson
NPL	National Physics Laboratory
OBN	Ocean-Bottom Node
OOB	One-Octave Band
PAM	Passive Acoustic Monitoring
PBR	Pulse to Bubble Ratio
PCAD	Population Consequences of Acoustic Disturbance
PCM	Population Covariance Matrix, Pulse Code Modulation

PD	Positive-Definite
PDF	Probability Density Function
PE	Parabolic Equation
PF	Probability Function
PSD	Power Spectral Density
PTS	Permanent Threshold Shift
RAW	Uncompressed, PCM audio format
RMS	Root Mean Squared
RR	Ridge Regression
RT	Ray Tracing
RTP	Right-Tail Probability
SA	Simulated Annealing
SBES	Single Beam Echosounder
SBL	Short Baseline
SBP	Sub-Bottom Profiler
SCM	Sample Covariance Matrix
SEL	Sound Exposure Level
SGGM	Sparse Gaussian Graphical Model
SNR	Signal-to-Noise Ratio
SOFAR	Sound Fixing And Ranging
SSS	Sidescan Sonar
SSV	Sound Source Verification
TOB	Third Octave Band
TSHD	Trailing Suction Hopper Dredge
TTS	Temporal Threshold Shift
USBL	Ultra Short Baseline

UXO	Unexploded Ordnance
WAV	“Wave” Audio File Format. Extension of “Wave” files
WI	Wavenumber Integration
WC	Well-Conditioned
WGN	White Gaussian Noise
WGS	White Gaussian Signal
WKB	Wentzel-Kramers-Brillouin
WSS	Wide-sense stationary

Chapter 1

Introduction

Human activities at sea are widely spread around the globe and through the year. Considering that acoustic waves can travel long distances underwater, the noise produced by those activities becomes a virtually omnipresent pollutant. Hence, there is a strong need to reduce, control and mitigate anthropogenic noise in the oceans.

However, the implementation of noise control measures adds extra costs to marine operations. Some form of opposition still exists among the various industrial sectors to the adoption of noise mitigation measures. The recurrent argument within the industry is that there is still a big knowledge gap about the impacts of noise on marine animals. This leads to overconservative assumptions about the impacts of noise, which result in more restrictive mitigation measures. The stance of the industry is understandable, as well as is that of the regulators and the public that want to protect the oceans. Scientists can play an important role, as their research can help better understand the impacts of noise on marine life, raise awareness, and contribute to the development of better mitigation strategies. Bridging the gap between the interests of the industry and the oceans is a rocky path, but science, evidence and better tools can certainly help with it.

For exploration, the oil and gas industry employs powerful, low-frequency impulsive sources known as seismic air guns. These sources are combined in large groups to produce a greater acoustic output that can penetrate the seabed down to a few hundreds of metres. The reflections are used to generate a stratigraphic image that can help identify oil and gas resources. The duration and extension of these surveys has the potential to disturb and displace marine mammals and other forms of marine life, and the use of noise mitigation techniques becomes crucial during these operations.

The noise can be reduced at the source, the environment, or the receiver. New designs for low-frequency impulsive sources have been investigated by the oil and gas industry, but the research is costly and the technical benefits are small. Employing environmental barriers for seismic operations is not practical either. But it is possible to limit the exposure of an animal to noise by monitoring the area of operations for their presence,

and stopping the emissions when the animal enters a predefined exclusion zone. Real-time monitoring of exclusion zones is an effective way of minimising the noise impact even when the rest of forms of mitigation fail.

Until not long ago, a fixed distance of 500 m was commonly used to define the exclusion range. Marine animals perceive sound in different ways, with some being more sensitive to lower frequencies while others can detect high frequencies more easily. In addition, the propagation environment changes with location, and so does the way in which sound propagates. Therefore, using a fixed exclusion range is not scientifically justifiable as it can *underestimate* the exposure of the marine animal to noise, with the resulting disturbance, or it can *overestimate* it, running into unnecessary costs due to the early interruption of the operations. In summary, the calculation of exclusion zones based on the characteristics of the source, the environment, and the receiver is a more appropriate way of defining those areas, as they become consistent with the noise the animal is exposed to.

This thesis focuses on the development of tools that can help with the estimation of more reliable exclusion zones. The experience that the author has gained over the last eight years working in underwater noise impact assessments has showed that there is a need in the field for better and more consistent ways of processing, interpreting and utilising acoustic measurements. Certainly, the field of underwater acoustics has produced a wide range of solutions to predict and study the sound field. It is not the objective of this thesis to reinvent the wheel, but to contribute with improved solutions that anyone can use to characterise the sound field, the propagation environment and the source. The research goals are the following:

i. To develop a versatile software package for the computation of sound field maps from raw audio and position data recorded during an acoustic mapping survey.

ii. To investigate different types of detection algorithms that can be used in the implementation of software routines for the automatic detection of a specific type of signal.

iii. To develop an air gun array signature model that can compete in quality with existing commercial and proprietary software applications.

Chapter 2

Background

This chapter presents the *background*. The objective is to set the scene and introduce a few ideas that the reader will find useful when navigating the core chapters of this thesis. The background is divided into seven sections covering topics that range from the characteristics of the oceans and the underwater propagation of sound, to the noise impact of seismic surveys and its mitigation using measurements and models.

Section 2.1 introduces a few metrics commonly used in underwater acoustics and frequently mentioned in the thesis. Section 2.2 describes the global and local oceanographic features that ultimately affect the way in which sound travels within the oceans. The section focuses on the propagation of sound within the water column and the effects that the acoustic waves experience as they travel through this complex environment. Section 2.3 goes one step further and presents the topic of sound propagation within the seabed, after introducing the most common morphological and geological features. The first three sections prepare the context before introducing the topics that are most directly connected to the work of this thesis.

In Section 2.4 we briefly describe what a seismic survey is, to then elaborate on the characteristics of its most powerful acoustic source: the seismic air gun array. Section 2.5 presents a short review about the impact of noise on marine animals, with special focus on seismic sources, and the methods that currently exist to mitigate them. Among these mitigation methods, the real-time monitoring of marine mammals with visual and auditory techniques (MMO, PAM) has a special place, as is the only mechanism of mitigation capable of protecting marine animals after all the other methods have failed. For this type of mitigation, sound field measurements and acoustic models play an important role, as they can be used to estimate realistic noise impact areas to mitigate for. Section 2.6 introduces some of the techniques and systems currently available for the mapping of the sound field. Section 2.7 describes the techniques and software that currently exist for modelling underwater sound propagation and air gun array signatures. This thesis focuses on the development of tools that can help with the calculation of noise impact zones, based on sound field mapping measurements and acoustic models.

2.1 Preliminary Concepts

2.1.1 The Wave Equation

The ocean behaves as a semi-infinite waveguide bounded at the top by the sea surface and at the bottom by the seabed. Waves that propagate in the water column are *compressional* in nature, that is, the particle motion is parallel to the direction of propagation. Compressional waves can travel within the seabed and transform into other type of waves in a process known as *mode conversion*.

A solid medium supports compressional, surface and shear waves. In a *shear* wave the particle motion is perpendicular to the direction of propagation, and in a *surface* wave the particles experience elliptical motion. Compressional, shear and surface waves are of great importance in underwater acoustics, and all three play a key role in the acoustic propagation properties of the underwater environment.

A sound wave can be fully characterised by the instantaneous displacement of its moving particles. As the particles vibrate, the fluid experiences variations in density and pressure. The wave equation that governs the sound propagation in lossless fluids is given by

$$\Delta p = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} \tag{2.1}$$

where c is the propagation velocity or speed of sound, p the acoustic pressure, t the time and Δ the *Laplacian* operator ($\Delta p = \nabla \cdot \nabla p = \partial^2 p/\partial x + \partial^2 p/\partial y + \partial^2 p/\partial z$, with *gradient* operator ∇). This general equation is derived from the laws of conservation of momentum and mass, and the assumption of adiabatic conditions (Kuttruff, 2009).

The sound speed c is defined by the characteristics of the propagation medium, namely the *density* ρ and the *bulk modulus* K, as in (2.2). The sound speed and density in water have approximate values of 1500 m/s and 1030 kg/m^3, and vary with temperature, salinity and pressure (see Sub-Section 2.2.1).

$$c = \sqrt{\frac{K}{\rho}} \tag{2.2}$$

Studying the propagation of *harmonic* waves is of most interest in practice, as these single-frequency waves can be later combined to extend the analysis to broadband signals. The *Helmholtz* equation for frequency f is given by

$$\Delta p + k^2 p = 0 \tag{2.3}$$

where k is the wave number ($k = \omega/c = 2\pi f/c$, for angular frequency ω). The above equation can be particularised for one dimension

$$p(x,t) = p_0 e^{j(\omega t - kx)} \tag{2.4}$$

The one-dimensional solution of (2.4) represents a plane harmonic wave. A *plane wave* is a progressive (forward-propagating) wave with constant pressure and phase across the plane perpendicular to the direction of propagation. The terms p_0 and $(\omega t - kx)$ in (2.4) represent the amplitude of the wave and its phase. The wave varies in time and space with periods $T = 1/f$ and $\lambda = c/f$. The spatial period λ is known as the *wavelength* and indicates the distance travelled by the wave in one period T.

The acoustic pressure is related to the particle velocity $\mathbf{v}$ by the momentum conservation $\nabla p = -\rho_0 \partial \mathbf{v}/\partial t$. The particle velocity is a vector and the pressure a scalar. For a plane harmonic wave, the relation between particle velocity and pressure is given by

$$v(x,t) = \frac{p(x,t)}{\rho_0 c} \tag{2.5}$$

The product $\rho_0 c$ is known as the *characteristic acoustic impedance* of the medium, and is expressed in Rayleighs. In sea water, the characteristic impedance is approximately $1.5 \cdot 10^6$ Rayl or $\mathrm{kg \cdot m^{-2} \cdot s^{-1}}$.

At distances that are large compared to the wavelength ($kr \gg 1$), a plane wave is a fair assumption. Nonetheless, sound propagates in three dimensions and a more general expression for the harmonic pressure is

$$p(r,t) = \frac{p_0}{r} e^{j(\omega t - kr)} \tag{2.6}$$

where r is the distance from the centre of a *point source* of amplitude $p_0 = j\omega\rho_0 Q_0/4\pi$, with Q_0 the strength or *volume velocity* of the source. This solution corresponds to a *spherical wave*. This type of wave is characterised by a spherical wavefront and attenuation that is inversely proportional to the travelled distance r. The acoustic impedance in a spherical wave is complex, meaning that velocity and pressure are out of phase.

The above equations assume that the medium is free of losses, but the sea water is a lossy environment. The effect of attenuation within the propagation medium results in an exponential decay of the pressure amplitude as the wave propagates. This attenuation accounts for the volume absorption within the propagation medium (see Sub-Section 2.1.3). An exponential term can be added to (2.4) to account for this effect, as shown below

$$p(x,t) = p_0 e^{-\gamma x} e^{j(\omega t - kx)} \tag{2.7}$$

where γ is the attenuation constant in Np/m. Another way of accounting for the absorption is to express the wave number k as a complex quantity $k' = k - j\gamma$. In practice, a *complex sound velocity* c' is conveniently used to account for the absorption effects

($c' = c + j\mathrm{Im}\{c'\} = \omega/k'$, where $c = \mathrm{Re}\{c'\}$). Assuming $k \ll \gamma$ (i.e., $\mathrm{Im}\{c'\} \ll \mathrm{Re}\{c'\}$), one can arrive at the approximate expression for the imaginary part of c'

$$\mathrm{Im}\{c'\} \approx \frac{\gamma c^2}{\omega} \tag{2.8}$$

The attenuation is more frequently given in dB/m through the *attenuation coefficient* α (Jensen et al., 2011)

$$\alpha = -20 \log e^{-\gamma} \approx 8.69\gamma \tag{2.9}$$

In underwater acoustic modelling, it is even more common to express the attenuation α in dB/λ, since experimental evidence indicates a first-power dependence of attenuation on frequency in most sediments and at most frequencies (Hamilton, 1975; Hamilton, 1980). By combining (2.8) and (2.9) and introducing the absorption term α_λ in dB/ λ, the imaginary part of the complex sound velocity is conveniently expressed as

$$\mathrm{Im}\{c'\} \approx \frac{\alpha_\lambda c}{54.6} \tag{2.10}$$

2.1.2 Intensity, Power and Pressure

The *intensity* is a measure of the local, instantaneous energy in a sound wave. It is a vector quantity normal to the direction of propagation. The intensity is defined as the average rate of energy flow per unit area and is expressed in W/m^2. The intensity $\mathbf{I}$ is the product of the pressure p and particle velocity $\mathbf{v}$.

In underwater acoustics, the intensity is less frequently reported due to the difficulty of measuring it. This trend is expected to change as species of fish and marine invertebrates, known to be sensitive to particle motion, are included in environmental impact assessments. Currently, it is more practical to use the *equivalent plane-wave sound intensity*, a function of the mean squared pressure ${p_{rms}}^2$ and the characteristic impedance $\rho_0 c$ (ISO, 2017)

$$I_{eq} = \frac{{p_{rms}}^2}{\rho_0 c} \tag{2.11}$$

The *power* is the rate at which the acoustic energy is emitted by the source and is expressed in Watts. It is the product of the intensity magnitude I and the surface S of the wavefront ($S = 4\pi r^2$ for a spherical wave).

The *root-mean-square* (rms) *pressure* p_{rms} is a common quantity for continuous and non-impulsive signals, defined as the squared sound pressure averaged over a specified integration time T, and expressed in Pa.

$$p_{rms} = \sqrt{\frac{1}{T}\int_T p^2(t)dt} \tag{2.12}$$

For impulsive and non-impulsive signals, the *sound pressure exposure* $E_{p,T}$ is a more common quantity than p_{rms}. It is expressed in $\mathrm{Pa^2s}$. The $E_{p,T}$ is referred to as *cumulative sound pressure exposure* when used to compute the dosage of a sequence of sound events of finite duration.

$$E_{p,T} = \int_T p^2(t)dt \tag{2.13}$$

The *peak pressure* p_{pk} is the greatest magnitude of the instantaneous sound pressure during a specified time interval. For a purely harmonic signal, p_{pk} is $1.41p_{rms}$. The *peak-to-peak pressure* p_{pk-} is the sum of the greatest positive pressure and greatest magnitude of negative pressure during a specified time interval. The p_{pk} and p_{pk-pk} are both expressed in Pa.

The large range of sound pressure amplitudes that can be measured in the water makes it impractical for the strength of a sound wave to be characterised by means of physical quantities such as pressure, power or intensity. Instead, these magnitudes are converted into the 10-base logarithm of a quantity divided by a reference value, with the result expressed in decibels. Every acoustic quantity in decibels is referred to as *sound level.* For squared quantities such as intensity, power and sound pressure exposure, the logarithm is multiplied by 10. For linear quantities such as peak and root-mean-square pressure, the logarithm is multiplied by 20. In water, the reference values are $p_{ref} = 1\ \mu\mathrm{Pa}$ for p_{rms} and p_{pk}, ${p_{ref}}^2 = 1\ \mathrm{pPa}$ for $E_{p,T}$, and $I_{ref} = 0.67 \cdot 10^{-18}\ \mathrm{W/m^2}$ for I_{eq}. The latter is the equivalent plane-wave intensity for a root-mean-square pressure of $1\ \mu\mathrm{Pa}$. When converted into decibels, the quantities p_{rms}, p_{pk}, p_{pk-pk}, $E_{p,T}$ and I_{eq} are referred to as sound pressure level $L_{p,rms}$, peak sound pressure level $L_{p,pk}$, peak-to-peak sound pressure level $L_{p,pk-pk}$, and sound exposure level $L_{E,p}$.

The *peak pressure level* $L_{p,pk}$ and the *sound exposure level* $L_{E,p}$ are used as a dual-exposure criteria for assessing the impact of noise from impulsive sources on marine mammals, fish and turtles (Southall et al., 2019; Popper et al., 2014), as they relate to the risk of auditory damage from high pressure gradients and high-energy sound events. For non-impulsive sources, only the sound exposure level is used as exposure criteria. The sound pressure level $L_{p,rms}$ is well suited for quantifying continuous signals, but it tends to produce inconsistent results when used on transient signals, for which no standard windowing method exists (Madsen, 2005).

2.1.3 Source Level and Propagation Loss

The source level and propagation loss are two essential concepts in the study of the underwater sound field. The source level is a measure of the acoustic output of a source, and the propagation loss is a measure of the variation in amplitude that sound experiences when travelling between source and receiver across a specific propagation path. These parameters may be obtained from measurements, simulations or a combination of the two. A simple relationship exists between the source level, the propagation loss and the sound field level.

Source Level

The *source level* SL is by definition the rms or exposure sound pressure level that a hypothetical point source with the same *acoustic power* and *directivity* as the true source would produce in a specified direction at a distance of 1 metre in a hypothetical infinite uniform lossless medium (ISO, 2017). In other words, the source level is a measure of the acoustic strength of a source in *free field* and *far field* conditions. Therefore, the source level is not necessarily equal to the sound level measured at 1 m. The source level is expressed in decibels referred to $1\mu\text{Pa}\cdot\text{m}$ or $1\mu\text{Pa@1m}$ (for the sound exposure metric the reference is $1\mu\text{Pa}^2\text{m}^2\text{s}$ or $1\mu\text{Pa}^2\text{s@1m}$). The frequency band, averaging time, and acoustic metric (rms, peak, exposure) should always be specified.

Formally, the source level is ten times the 10-base logarithm of the *source factor* F_S, divided by the appropriate reference. The *source factor* is the product of the mean-square sound pressure $\overline{p^2}$ (or sound exposure $E_{p,T}$) and the squared distance r^2 in a specified direction in the far field of the source (ISO, 2017). The source factor is a constant which is directly related to the acoustic power of the source W, the directivity index $D(\varphi,\theta)$ in the direction of emission, and the characteristic impedance of the medium ρc by $\overline{p^2}r^2 = WD(\varphi,\theta)\rho c/4\pi$. Like the acoustic power, the source factor is independent of the propagation environment (Robinson et al., 2014, section 6.2.3). Note that the *source factor* is defined for an infinite uniform lossless medium, and its calculation in practice requires some knowledge about the attenuation experienced by the sound wave across the propagation path. This attenuation is known as *propagation loss*.

Propagation Loss

Sound waves travelling in the ocean experience distortion and attenuation due to multiple reflections (see Sub-Section 2.2.2) and various loss mechanisms (see Sub-Section 2.2.3). The standard quantity used in underwater acoustics to measure the change in strength of the acoustic signal emitted by a source as a function of frequency and space is the *propagation loss*.

The *propagation loss* PL is by definition the difference in decibels between the source level in a specified direction SL and the root-mean square sound pressure level L_p at a

specified position (ISO, 2017). The latter is typically referred to as *received level* RL. In practice, the PL is most commonly related to a single source level averaged over a range of contributing angles of emission. The PL may be calculated for a single frequency or frequency band, for a specific averaging time, and for an acoustic metric other than rms (e.g., peak, exposure), and these parameters must always be specified.

$$PL = SL - RL \tag{2.14}$$

A more generic term is the *transmission* loss TL, which refers to the reduction in sound pressure level between two specified points in the acoustic field. Both terms, PL and TL, are sometimes used interchangeably.

The *propagation loss* may be considered as the sum of losses due to *geometric spreading* and *attenuation* (Urick, 1983, section 5.1; Jensen et al, 2011, section 1.3.4). The range-dependent propagation loss at a given receiver depth may be approximated by the equation

$$PL(x) = GL \log(x) + AL\, x \tag{2.15}$$

where x is the *range* or horizontal distance from the source, GL the geometric loss factor and AL the attenuation loss factor. The logarithmic and linear terms in the equation represent the geometric spreading loss and attenuation loss.

The *spreading loss* accounts for the weakening of the emitted signal with range as the acoustic wavefront (or locust of points with equal phase) spreads over a larger area. The sound field in an ocean waveguide may be divided into three regions characterised by different geometric spreading: *spherical*, close to the point source ($GL = 20$); *intermediate*, at a distance of several water depths; and *cylindrical*, at long ranges ($GL = 10$). The *attenuation loss* accounts for volume attenuation in the water column, bottom loss and surface scattering. Note that additional gain or losses resulting from focusing, shadowing or channelling effects are implicitly included in the GL and AL factors.

The expression (2.14) models the general aspect of the range-dependent pressure without explicitly including relations for the complex physical processes involved in underwater sound propagation. This expression is by no means a substitute to sound field measurements or verified propagation loss simulations. However, it is a useful tool for establishing a relationship between source level and receiver levels when the measurements are limited and a convenient way of reporting propagation losses is needed. Estimating the propagation loss spectrum is a primary goal in acoustic field verification and an essential step in sound source verification (see Section 2.6).

2.2 Underwater Sound Propagation: The Water Column

2.2.1 Global Variation of Sea Water Properties

Dynamic Features in the Oceans

The oceans are changing environments with a wide range of dynamic features. According their scale, these features are classified as *large* (>100 km), *meso* (100 m – 100 km) and *fine* (< 100 m). The largest scale feature is the horizontal ocean circulation. Mesoscale oceanic features include fronts, eddies and internal waves. Thermohaline staircases and Langmuir circulation are examples of fine scale oceanic features. These phenomena are briefly described below.

Horizontal ocean circulation is the result of global density gradients, known as *thermohaline* circulation, and the action of wind. The *thermohaline* circulation is caused by heat and freshwater fluxes and represents a globally interconnected system of ocean currents (Etter, 2013). The *surface currents* in this ocean conveyor belt are mainly controlled by wind patterns. The thermohaline current travels towards the North Atlantic, eventually sinking at high latitudes due to a progressive increase in density. This deep current then makes its way towards the Antarctic Ocean, where it joins a circular stream and upwells by effect of the low temperature and high salinity content of Antarctic waters. Extensive mixing of water masses takes place between the ocean basins, reducing the differences between them and making the oceans a *global system*.

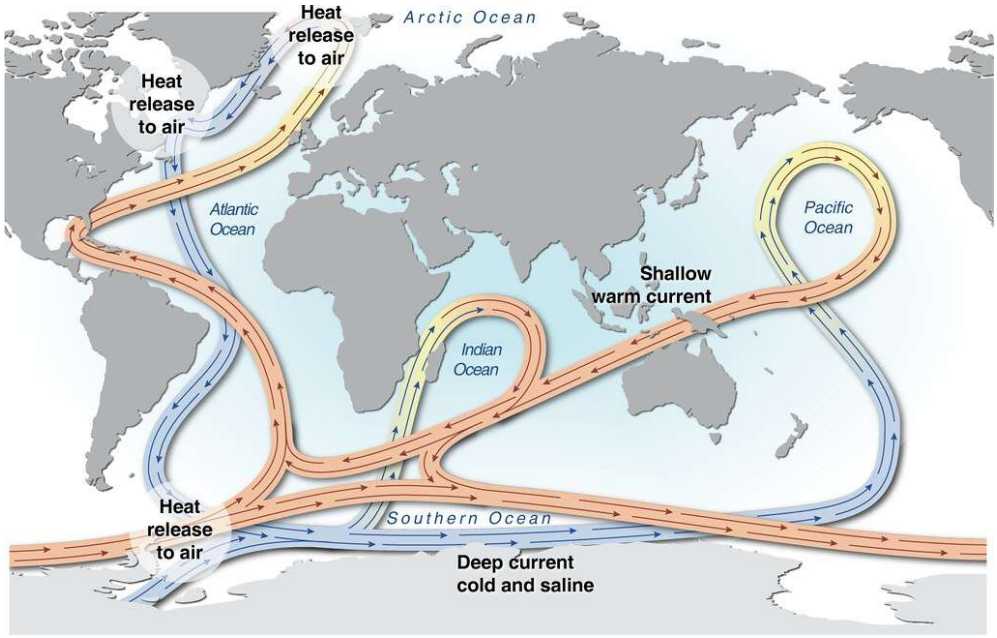

Figure 2.1 Thermohaline ocean circulation (from Hugo Ahlenius, UNEP/GRID-Arendal).

Fronts are transition zones that separate two distinct water masses of different temperature, salinity and biological content. At each side of the front, the water masses move in different directions resulting in ocean currents and vertical circulation patterns. Fronts vary in strength, duration and location, and have the ability to significantly alter the sound propagation characteristics in water. In particular, fronts can alter the surface sound speed (by up to 30 m/s), the depth of the mixed-water layer (by ~300 m), the depth of the deep-water sound channel (by up to 750 m), and the amount of scattering and refraction.

Eddies are separate temporary masses of swirling water detached from the large ocean current circulation system. These transport heat and salt around the ocean and cause nutrients from cold, deeper waters to rise to the surface. These systems have typical diameters of less than 100 km, extend down to the *pycnocline* region (~1000 m depth), last several weeks, travel long distances before dissipating, and result in large-scale water-mixing processes. Examples of eddies are found in the Atlantic Ocean near the Gulf Stream.

Internal waves propagate below the surface along the boundary of two water layers of different density, at speeds determined by the density difference. Their presence is determined by periodic oscillations of temperature at a given location. Internal waves may last hours and extend several kilometres, with much higher amplitudes (~10 m) and larger wavelengths (0.1-10 km) than surface waves. Internal waves are common in continental shelves and propagate towards the shore, where they are absorbed and reflected.

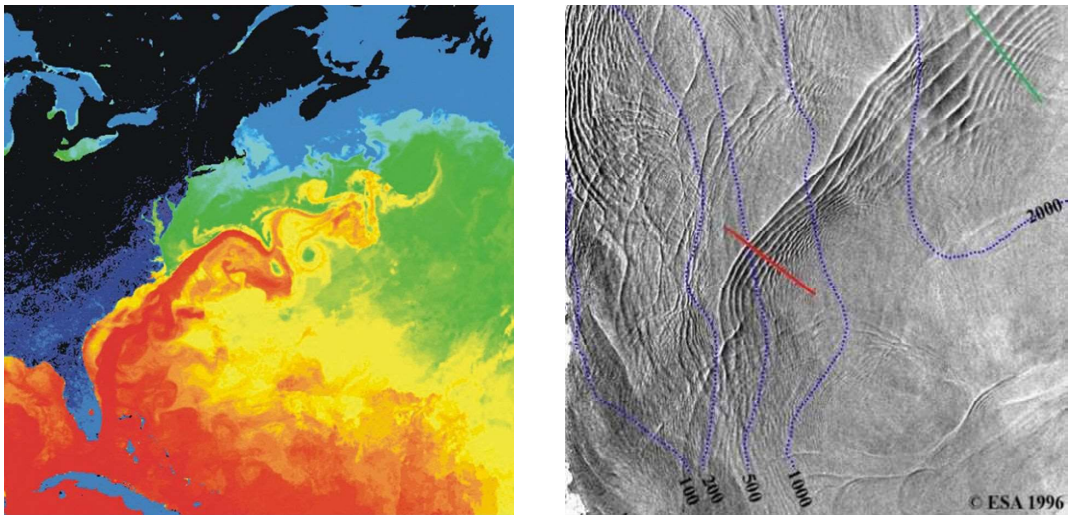

Figure 2.2 Satellite images of the Gulf Stream and its eddies (left, warm colours represent warm areas, from Coates, 2006) and in-ternal waves in the coastal Korean waters (right, from Mitnik & Dubina, 2005).

Thermohaline staircases are associated with the confluence of fresh and saline waters, which create interbedded thin water layers (~10 m) of different temperature and salinity that can cause changes of 2-3 m/s in the sound speed. The *Langmuir circulation* consists

of a series of vortices 4-6 m deep and 10-50 m wide created near the sea surface and aligned with the wind direction.

Regional Sound Speed and Density Profiles

The propagation of sound in the oceans is mainly governed by the sound speed, which is a function of temperature, salinity and depth in the water. The sound speed gives rise to channelling, focusing and shadowing effects (Coats, 2006). These effects are described in detail in Sub-Section 2.2.2.

The dependence of temperature, salinity and sound speed with depth in three different geographical zones (equatorial, tropical and high latitudes) is shown in Figure 2.3. All curves converge at the deep ocean, which is essentially the same piece of water. The main differences appear near the sea surface, where sun and winds have the highest impact. The equator is characterised by high temperatures and light winds. Medium temperatures and strong winds dominate in the tropics, and low temperatures and light winds are found at high latitudes. The wind contributes to the increase in the rate of water evaporation (higher salinity), but also helps to the downward mixing of water at higher temperatures with deeper (cold) water.

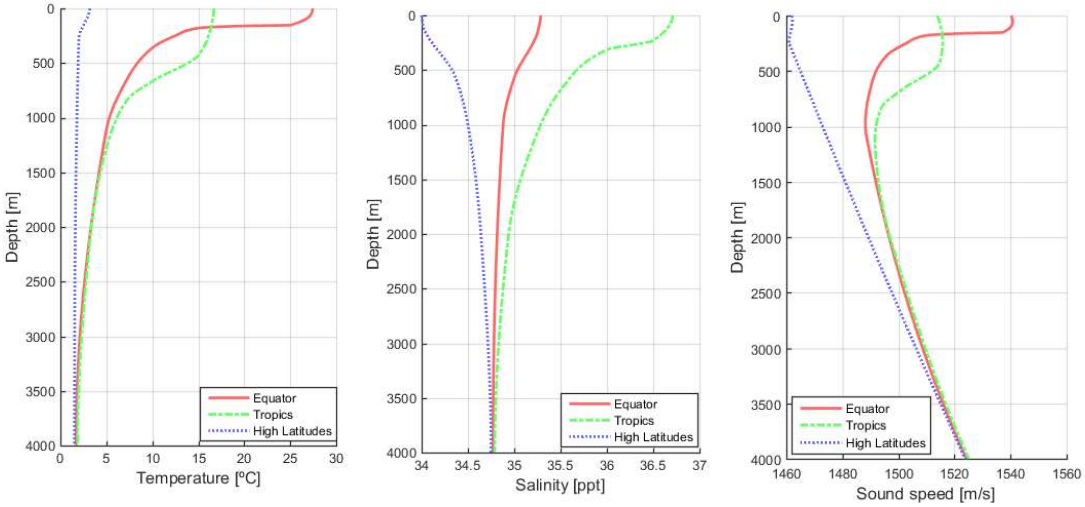

Figure 2.3 Temperature, salinity, density and sound speed profiles with depth in water, for three geographical zones: equator (0°N), tropics (23.4°N/S) and high latitudes (>60°N/S).

Sound speed profiles can be divided into three main depth regions: the sonic or *mixed* layer consisting of well-mixed isothermal water created by the effect of winds, the *thermocline* or region of rapid temperature variation, and the *isothermal* or region of constant temperature. A sound speed minimum occurs between the negative and positive gradients of the thermocline and isothermal regions. The depth of this speed minimum is referred to as the *sound channel axis*, and is deep at low latitudes (~1000 m) and appears near the surface at high latitudes (~200 m). The *critical depth* is the depth below the channel axis at which the sound speed equals the maximum value at the

surface. The channel axis and critical depth are determining factors in the long-range propagation of sound waves (see Sub-Section 2.2.2). Due to its strong depth-dependence but weak range-dependence, the sound speed is said to be *horizontally stratified*.

Several models for the calculation of sound speed in sea water have been developed over the years. The equations are functions of temperature, salinity and depth, and their coefficients are calculated based on large datasets of experimental data. Among the models available are those from Wilson (1960), Leroy (1969), Frye and Pugh (1971), Del Grosso (1974), Medwin (1975), Chen and Millero (1977), Lovett (1978), Coppens (1981), Mackenzie (1981), and Leroy et al. (2008, 2009).

A frequently used compact expression for the calculation of sound speed in sea water is the simplified formula from Medwin (1975)

$$c = 1449.2 + 4.6T - 5.5 \cdot 10^{-2}T^2 + 2.9 \cdot 10^{-4}T^3 + (1.34 - 10^{-2}T)(S - 35) + 1.6 \cdot 10^{-2}z \tag{2.16}$$

where c is the sound speed in m/s, T the temperature in °C, S the salinity in psu, and z the depth in water in m. The equation works down to depths of 1000 m. A more accurate and frequently used model is the one proposed by Chen and Millero (1977), but it presents some deficiencies in its behaviour with depth (Millero and Li, 1994). Leroy et al. (2008) proposed a simplified equation for the sound speed in sea water that is depth dependent and still shows excellent agreement with the experimental results.

The density of the sea water varies between 1020 and 1030 kg/m^3. Despite this small variation, the density is still an important parameter in underwater sound propagation. Like the sound speed, the density is a function of temperature, salinity and depth. The density can be computed with the following expression (Fofonoff & Millard, 1983)

$$\rho(S,T,P) = \frac{\rho(S,T,0)}{1 - \frac{P}{K(S,T,P)}} \tag{2.17}$$

where ρ is the density, P the hydrostatic pressure and K the secant bulk modulus. The term $\rho(S,T,0)$ is the density of sea water at 1 atm (sea surface) given by Millero and Poisson (1981). Another useful density equation is given by Mackenzie (1981).

2.2.2 Multipath Propagation

Sound waves refract as they travel through layers of different impedance, resulting in channelling, focusing and shadowing effects. Although the propagation of sound in the oceans is mainly governed by the sound speed, the bathymetry, seabed geology, and sea surface roughness also play an important role in the travelled path of the sound wave, particularly in shallow waters. The combination of the vertical sound speed profile with the upper and lower boundaries determines the *multipath propagation* scenario.

Depending on the type boundary interaction, a propagation path can be classified as "only refracted", "upward-refracted, surface-reflected", "downward-refracted bottom-reflected", and "surface and bottom-reflected". A wave of the last group experiences the highest losses, whereas one from the first group is only affected by the absorption and scattering within the water column. Some of the most common multipath propagation scenarios studied in ray theory are convergence zones, deep sound channel, surface duct, arctic half channel and shallow water propagation (Jensen et al., 2011; Etter, 2013; Urick, 1983). We briefly describe these scenarios below.

A *convergence zone* is a focusing phenomenon that produces zones of high intensity near the surface (see Figure 2.4). The phenomenon is caused by the upward refraction of sound, produced by the positive sound speed gradient in deep waters. The focusing effect is spatially periodic and occurs every 35-70 km. The distance between high-intensity regions is called the *convergence zone range*. The sound level at the convergence zones can be up to 20 dB higher compared to non-focusing conditions.

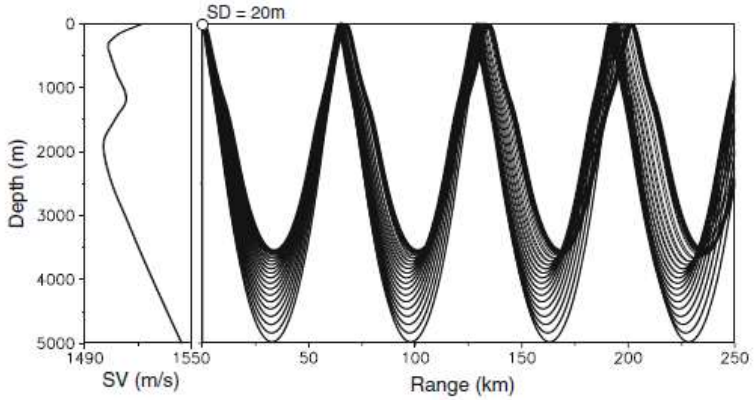

Figure 2.4 Ray tracing diagram representing the convergence-zone propagation in the North Atlantic. The source is placed at 20 m depth, and its aperture is ±10°. Ray paths refocus near the surface every 65 km (from Jensen et al, 2011)

The *deep sound channel* is a region in deep water where the sound becomes partially trapped and propagates long ranges without experiencing reflection losses at the sea surface or the seafloor (see Figure 2.5). The deep sound channel, also referred to as SOFAR, appears as a result of a minimum in the vertical sound speed profile or *channel axis*. The negative and positive sound speed gradients above and below the channel axis make a fraction of the sound energy to continually refract towards the axis. The amount of energy trapped in the channel is proportional to the angle of aperture of the acoustic rays. These waves can still experience losses by the effect of absorption in sea water.

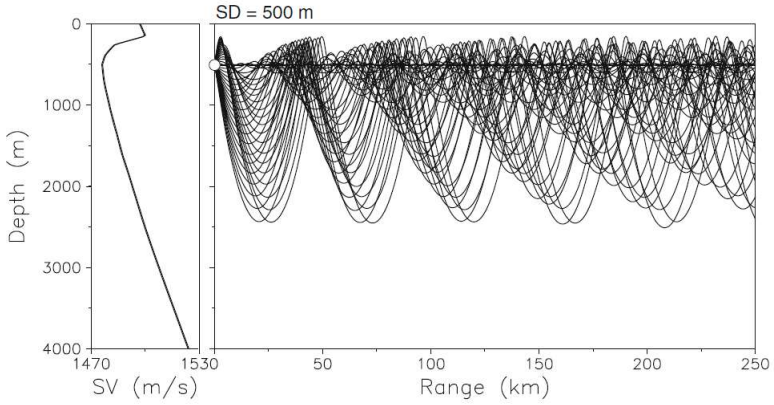

Figure 2.5 Ray tracing diagram representing the deep sound channel propagation in the Norwegian Sea. The source is placed at the *channel axis* (500 m depth) and its aperture is ±10°. No boundary interactions occur with this aperture (from Jensen et al, 2011)

The *surface duct* is formed when the temperature in the mixed layer is uniform (see Figure 2.6). The increase in hydrostatic pressure with depth creates a positive sound speed gradient in this isothermal layer. For large enough frequencies and shallow angles, the rays emitted by a source placed in the mixed layer become trapped in a surface duct, as they alternatively refract and reflect on the sea surface. A *shadow zone* with considerably lower sound levels (50-70 dB less) is formed below the mixed layer and at a certain distance from the source. A fraction of the acoustic energy penetrates the shadow zone due to diffraction and volume scattering (Urick, 1983).

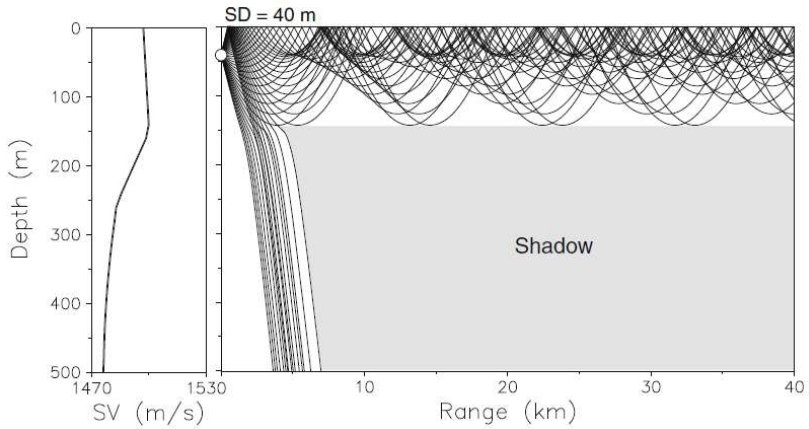

Figure 2.6 Ray tracing diagram representing the surface-duct propagation in the Norwegian Sea. The source is placed at 40 m. Only rays leaving the source at angles within ±3° are trapped in the duct; steeper rays are refracted towards the seafloor (from Jensen et al, 2011)

The *Arctic half-channel* is characterised by a steep positive sound speed gradient in the upper 200 m, followed by a less steep gradient typical of the isothermal region (see Figure 2.7). The steep gradient is caused by an increase in salinity and temperature with depth, and is responsible for generating a strong surface duct. The upward refracting conditions make the sound interact repeatedly with the ice. The scattering associated

with the rough surface of the ice canopy favours the propagation at 10 to 30 Hz (Jensen et al., 2011).

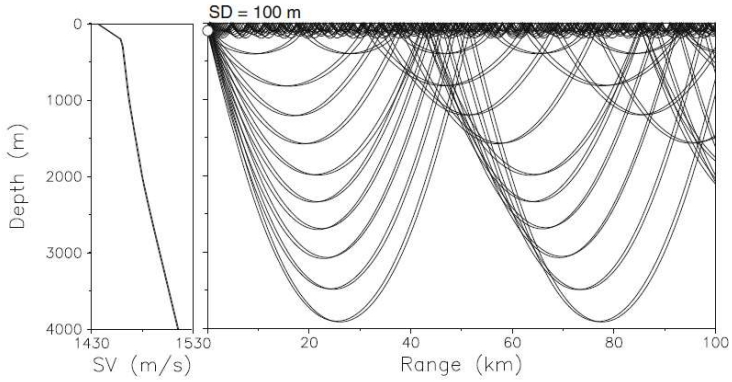

Figure 2.7 Ray tracing diagram representing Arctic propagation. Source placed at 40 m. Rays leaving the source at angles within ±17° interact only with the sea surface (Jensen et al, 2011)

In *shallow waters* the gradient of the vertical sound speed profile is generally negative. The downward refraction means that long-range propagation takes place primarily via bottom interacting paths (see Figure 2.8). In winter, the sound speed tends to become constant with depth, reducing the number of bottom interactions. Propagation losses at low and mid frequencies are dominated by bottom reflections, and at high frequencies by scattering.

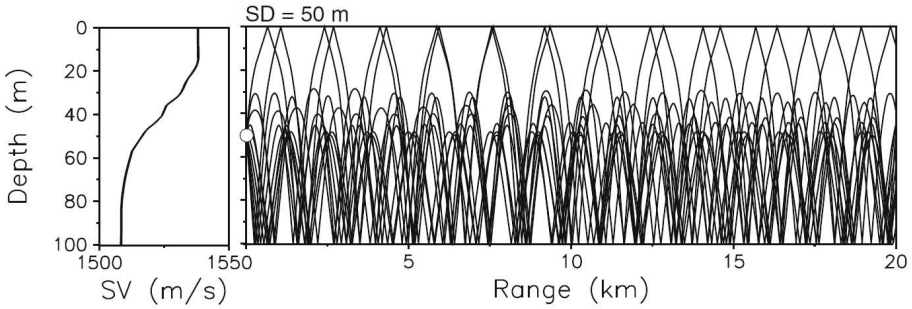

Figure 2.8 Ray tracing diagram representing shallow water propagation in the Mediterranean Sea, in summer. The source is placed at 50 m. The negative gradient of the sound speed profile make all rays interact with the seafloor (from Jensen et al, 2011)

There is a critical frequency $f_{c,sw}$, the frequency of the *first acoustic mode*, below which the shallow water environment stops acting as a waveguide. This frequency depends on water depth and bottom composition. For a homogenous water column of depth h_w and sound speed c_w and a sea bottom with sound speed c_b, the critical frequency $f_{c,sw}$ is given by (Urick, 1983)

$$f_{c,sw} = \frac{c_w}{4h_w\sqrt{1-\left(\frac{c_w}{c_b}\right)^2}} \tag{2.18}$$

The high and low frequency roll-off caused respectively by scattering losses and water depth result in an optimal propagation within a specific frequency band (200-800 Hz for 100 m depth).

2.2.3 Propagation Losses

Geometric Spreading

The *geometric spreading loss* is the attenuation that a sound wave experiences when its acoustic wavefront spreads over a larger area as it propagates away from the source. The attenuation in intensity (or pressure) is explained by the equal distribution of the acoustic power across the surface of the wavefront. In the absence of any other losses, the intensity at any point is reduced proportionally to the surface increase. Hence, in free-field and non-dissipative conditions, the sound intensity is inversely proportional to the surface of the wavefront.

The rate of sound level attenuation with range is determined by the *geometric loss factor* GL and is proportional to a power value of the range. In the free field, where reflections do not interfere with the radiated sound, the intensity decreases as the square of the range ($GL = 20$). This kind of spreading is known as *spherical* or inverse-square. In practice, spherical spreading will only occur at close ranges and deep waters, where the contribution from reflections is minimal. In an ocean waveguide, there will be a range beyond which a cylindrical wavefront is formed as a result of sound being confined between the seabed and the sea surface. This kind of spreading is said to be *cylindrical* or inverse first power ($GL = 10$). The propagation losses are complex functions of range and tend to show less predictable patterns in the transition region between spherical and cylindrical spreading.

Volume Absorption

The attenuation that occurs as the sound propagates through the mass of water is referred to as *volume attenuation*. The relaxation of *magnesium sulphate* ($MgSO_4$) and *boric acid* ($B(OH)_3$) molecules dominate the sound absorption in sea water below 100 kHz and 1 kHz, respectively (Francois and Garrison, 1982a; 1982b). Molecular relaxation is the dissociation of ionic compounds in solution due to local pressure variations caused by the acoustic wave. Recent models for the absorption of sound in sea water follow the form (Lurton, 2010)

$$\alpha = C_1 \frac{f_1 f^2}{{f_1}^2 + f^2} + C_2 \frac{f_2 f^2}{{f_2}^2 + f^2} + C_3 f^2 \tag{2.19}$$

where f is the frequency, f_i the relaxation frequencies and C_i constants. The first two terms represent the relaxation processes, and the third term the viscosity of pure water. C_i and f_i depend on temperature, salinity and hydrostatic pressure.

The most popular model is the one proposed by Francois and Garrison (1982a; 1982b), based on the model from Fisher and Simmons (1977). A simplified equation which shows good agreement with the Francois and Garrison model is that of Ainslie and McColm (1998). The absorption increases rapidly with frequency, as shown in Figure 2.9. For frequencies higher than 10 kHz, it becomes a limiting factor for sound propagation.

The scattering produced by fine scale inhomogeneities in the water plays an important role on the attenuation of the forward propagating sound. The scattering is a non-dissipative process, but is effectively an absorption mechanism (Jensen at al, 2011).

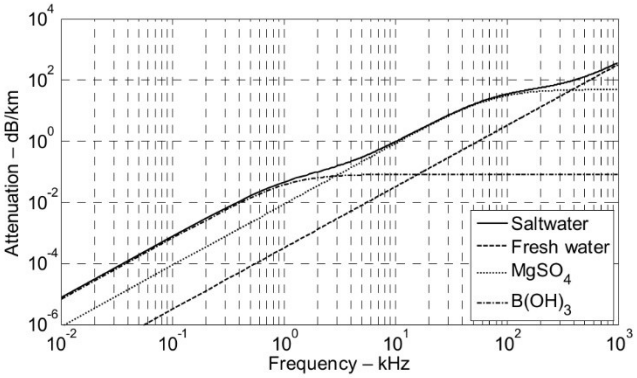

Figure 2.9 Absorption in sea water as a function of frequency for 10°C, 35 ppt and 7.8 pH. The different contributing processes are indicated as separate curves and include magnesium sulphate $MgSO_4$ and boric acid $B(OH)_3$ relaxation, and fresh water viscosity (from Hovem, 2013)

Sea Surface Scattering

There are three main factors associated with the sea surface that can have an important effect on the underwater sound field. These are scattering, turbidity and bubbles, and Lloyd mirror effect.

The sea surface under low sea state conditions behaves as an almost perfect reflector; nearly all the energy is reflected in the specular direction. In the presence of wind, the surface is altered and its roughness causes the scattering of sound in multiple directions, contributing to the reverberant field. The energy in the specular direction is reduced accordingly. The roughness of the sea surface can be quantified by the Rayleigh parameter R (Etter, 2013)

$$R = 2ka\cos\theta \tag{2.20}$$

where k is the acoustic wavenumber, a the root-mean square amplitude of the surface waves and θ the angle of incidence of the plane wave. For $R \ll 1$ the sea surface is smooth for the given frequency, and for $R \gg 1$ the surface is acoustically rough.

The wind and the action of waves generate microbubbles ranging in size from 1 to 100 μm. A layer of microbubbles causes absorption and scattering due to the different impedance of the air. These bubbles resonate at frequencies between 50 and 300 kHz.

When a sound wave hits a smooth sea surface, a reflected wave of equal amplitude and opposite phase is combined with the direct signal to produce a spatial interference pattern known as *Lloyd mirror effect*. This phenomenon diminishes with increasing surface roughness. At long ranges, the magnitude of the pressure resulting from the coherent sum of the direct and reflected signal is given by

$$|p| = \frac{2}{r}\cos\left(2k\frac{z_r z_s}{x}\right) \tag{2.21}$$

z_s and z_r are the depths of the source and receiver. At ranges beyond the last maximum in the interference pattern ($x_{max} = 4z_r z_s/\lambda$) the pressure decays as r^2.

Seabed Attenuation

The acoustic propagation characteristics of the seabed are a determining factor in the propagation loss in shallow water environments, especially in those with isothermal or negative temperature profiles (see Sub-Section 2.2.2). Sound waves reaching the seafloor will transmit, refract and reflect at its multiple interfaces, where mode conversion takes place. Mode conversion refers to the transformation of a wave from one type into another. This is crucial, as shear waves tend to experience stronger attenuations (Hamilton, 1980). The study of the acoustic propagation in an elastic, multi-layered seabed is a more involved problem compared to that of the water column. An analysis of the acoustic propagation in a real seabed is out of the scope of this thesis, but we consider it may be of interest to introduce some basic concepts about the modelling of the seabed impedance in the next section. Some early research around this topic has been carried out by the author as part of the development of a numerically stable algorithm for the computation of the complex impulse response of an elastic, multi-layered seabed. The sub-sections dedicated to this topic only intend to give an insight into this future line of research, and complicated mathematical formulation will be avoided.

2.3 Underwater Sound Propagation: The Seafloor

2.3.1 Morphology of the Seafloor

There are two main depth regions in the oceans: *shelf areas*, between the sea level and 1 km depth, and the *deep ocean*, between 1 km and 5 km depth (Seibold & Berger, 2017). The *shelf break* is the region where the shelf joins the *continental slope*, a moderately steep part of the sea floor that precedes the *continental rise*, which gently descends into the deep sea. The *continental slope* and *continental rise* connect the shelves with the deep ocean.

The most characteristic formations of the *deep ocean* are the abyssal plains, abyssal hills, trenches, abyssal fans, and seamounts. *Abyssal plains* are vast flat areas of hundreds of miles in diameter where fine debris from the continent is accumulated, while *abyssal hills* are small landforms rising from the floor of an abyssal plain. Abyssal hills are the most common landscape, covering more than 30% of the ocean floors. Large parts of the ocean floor also consist of immense mountain ranges, like the ones forming the mid-ocean ridge. *Trenches* are geological depressions with depths of up to 10 km. The new seafloor formed at the centre of the mid-ocean ridge travels towards the trenches, where it sinks. *Abyssal fans* are geological structures associated with large scale sediment deposition formed by turbidity currents. *Seamounts* are terrain elevations produced by volcanic activity.

The *continental margins* comprise the continental shelf, slope and rise, and can be classified *as passive or active*. Passive margins, like those in the Atlantic Ocean, are steadily sinking regions formed through a pulling apart of the continent which accumulate thick sequences of sediments. Active margins, like those in in the Pacific Ocean, are rising regions associated with volcanism, faulting and mountain building processes. *Shelves* are generally flat and not deeply submerged (~130 m); they are wide and built up with sediments in passive margins and narrow and rocky in active margins. Shelves can be flat and smooth, like those formed by large deltas, rugged in ice regions, or outlined by dunes, barriers, beach berms, or sand waves. About 8% of the ocean floor consists of submerged continental shelves. The terrigenous sediment from the continent mostly accumulates in the *continental slope*. The Atlantic Ocean in particular has the largest proportion of continental slope and rise areas. The continental margins are the main site of deposition for debris coming from the continents, but also the most fertile part of the ocean with a large amount of accumulated organic matter. Other characteristic formations of the continental margins are *submarine canyons*, steep-sided valleys that cut into the continental slope.

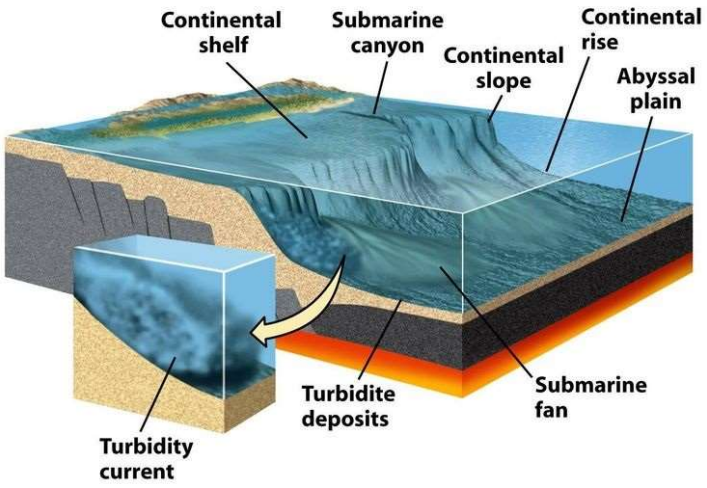

Figure 2.10 Main geological features of the oceans (from Grotzinger & Jordan, 2010)

2.3.2 Geology of the Seafloor

Marine sediments are very diverse and include debris from continents, shells and organic matter, salt precipitated from seawater and volcanic products such as ash and pumice (Seibold & Berger, 1993). Most derive from weathering rocks on land. Waves and tides take debris offshore from land, which may stay on the shelf or accumulate on the slope. The main sediment input comes from rivers, glaciers, wind and volcanic activity. High sedimentation rates occur at the edges of the continents, with abyssal regions featuring the lowest rates.

There are three main types of sediments according to their origin: *lithogenous*, consisting of detrital products of existing rocks; *hydrogenous*, formed by particles precipitated from solution; and *biogenous*, made up of remains of organisms. *Lithogenous* sediments are classified by grain size as *gravel*, *sand*, *silt* and *clay*. There is a gradual distribution of grain sizes from source to place of deposition: the bigger particles remain closer to the source, while the clay-sized particles can be carried over large distances. Sand is typical of beach and shelf deposits; silt is very common in the continental slope and rise, but may also be present in shelves under quiet conditions; clay is ubiquitous in the continental margins and deep sea, and low energy environments in general. *Biogenous* sediments comprise calcareous, siliceous and phosphatic hard parts of organic origin, both *benthic* and *planktonic*. Benthic matter is predominant in shelves and planktonic deposits in the deep sea. *Hydrogenous* sediments require an enclosed basin for the water to evaporate, and are typical in coastal lagoons and salt seas on the shelves.

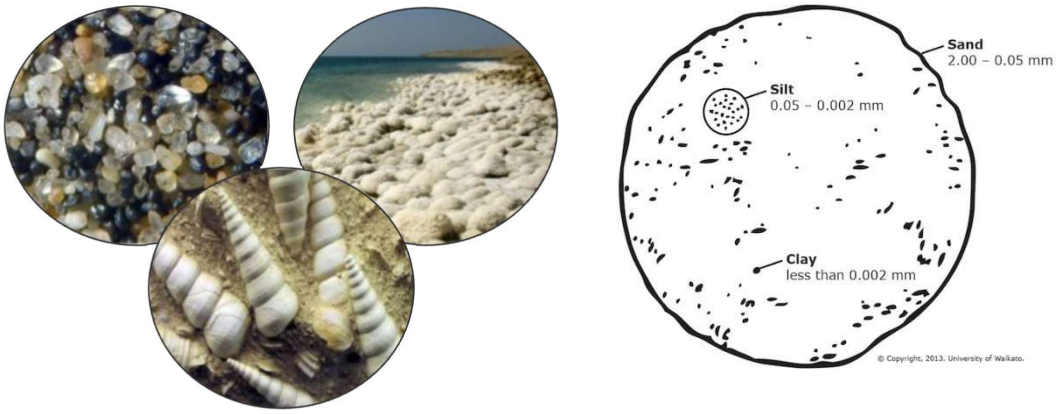

Figure 2.11 Images of the main three types of sediments according to their origin (left): *lithogenous* (top left), *biogenous* (bottom) and *hydrogenous* (top right). The image on the right shows the relative size of sediment particles

2.3.3 Propagation in a Fluid-Fluid Interface

The underwater medium can be conceived, in first approximation, as two stacked fluid and homogeneous semi-infinite layers of differing acoustic impedances (see Figure 2.12). In this configuration, the top layer represents the water column and the bottom layer

the sea floor. A plane wave will undergo reflection and transmission when it reaches the interface between the two media. The reflected wave will then travel in a direction θ_1 symmetrical to that of the incident wave, and the transmitted or *refracted* wave will travel in a direction θ_2.

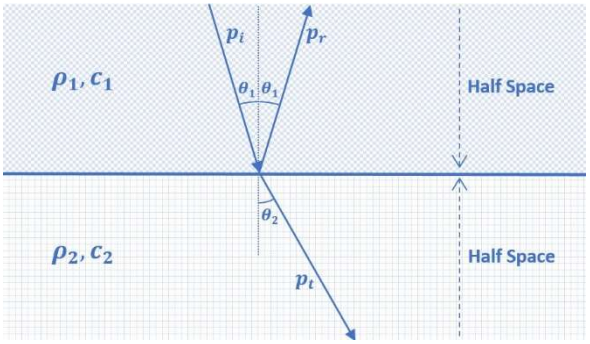

Figure 2.12 Incident, reflected and transmitted waves at the interface between two semi-infinite layers of different acoustic impedance.

The angle of refraction θ_2 is given by the Snell-Descartes law and depends on the angle of incidence θ_1 and the ratio of sound speeds between the two layers c_1/c_2

$$\frac{\sin\theta_1}{c_1} = \frac{\sin\theta_2}{c_2} \tag{2.22}$$

The acoustical properties of an interface are fully described by the complex *reflection* or *transmission factors*, R and T. These factors indicate the changes in amplitude and phase of the reflected and transmitted portions of a plane wave that reaches a medium with a different acoustic impedance.

$$R = \frac{p_r}{p_i} = |R|e^{j\varphi_r} \tag{2.23}$$

$$T = \frac{p_t}{p_i} = |T|e^{j\varphi_t} \tag{2.24}$$

$|\ |$ and φ represent the module and phase. Another quantity used to characterise the acoustic behaviour of the interface between two media is the *acoustic impedance*, which relates the particle velocity normal to the surface v_z $(-(j\omega\rho)^{-1}\partial p/\partial z)$ with the pressure p at the surface

$$Z = \left(\frac{p}{v_z}\right)_{z=0} \tag{2.25}$$

The propagation between two media may be expressed as a balance between the incident, reflected and transmitted waves at the interface ($z = 0$). This balance is described by the boundary conditions

$$p_t(0,x,t) = p_i(0,x,t) + p_r(0,x,t) \tag{2.26}$$

$$v_{t,z}(0,x,t) = v_{i,z}(0,x,t) + v_{r,z}(0,x,t) \tag{2.27}$$

The subscripts t , i and r refer to the transmitted, incident and reflected waves. These equations simply indicate the continuity of pressure and velocity at the interface, that is, $p_1|_{z=0} = p_2|_{z=0}$ and $v_1|_{z=0} = v_2|_{z=0}$. The expressions for the sound pressures $p_{i,r,t}$ and particle velocities $v_{i,r,t}$ at the interface can be found in Kuttruff (2009).

By evaluating (2.26) and (2.27), one arrives at the following two expressions for calculating the reflection and transmission factors (R, T) from the characteristic impedances of the two media $Z_{1,2}$

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \tag{2.28}$$

$$T = \frac{2Z_2}{Z_1 + Z_2} \tag{2.29}$$

The equations show that the acoustic energy is more easily reflected from sediments and rocks whose properties result in a higher impedance contrast. The characteristic impedance of medium m for a normal angle of propagation θ_m is

$$Z_m = \frac{\rho_m c_m}{\cos\theta_m} \tag{2.30}$$

R and T are also angle-dependent parameters. In general, R and T in a homogeneous sea floor layer are functions of the angle of incidence, but are *independent* of frequency, considering that the variation of sound speed with frequency is negligible in marine sediments (Hamilton, 1975). The frequency dependence of the reflection and transmission factors arises in a multi-layered seabed (see Sub-Section 2.3.5).

At the water-seabed interface, two characteristic angles of incidence can be found: the *critical angle* θ_c, above which total reflection occurs ($|R| = 1$ for $\theta_1 > \theta_c$) and the *intromission angle* θ_0, at which total transmission occurs ($|R| = 0$ for $\theta_1 = \theta_0$).

A critical angle θ_c exists if the sound speed in the second medium is higher than in the first medium ($c_2 > c_1$). This condition is met for a water-seabed interface with medium to coarse grain unconsolidated sediments (silt, sand, gravel), consolidated sediments and rocks. The value of θ_c is obtained by evaluating the Snell-Descartes law from (2.22) for an angle of refraction $\theta_2 = 0$.

$$\theta_c = \operatorname{asin}\left(\frac{c_1}{c_2}\right) \tag{2.31}$$

The intromission angle θ_0 is obtained from (2.28) and the Snell-Descartes law by considering the modulus of the reflection coefficient to be zero ($|R| = 0$). In non-dissipative media (i.e., Z_1 and Z_2 are real) the intromission angle is given by

$$\theta_0 = \operatorname{acos}\left(\sqrt{\frac{\left(\frac{c_1}{c_2}\right)^2 - 1}{\left(\frac{\rho_2}{\rho_1}\right)^2 - 1}}\right) \tag{2.32}$$

In marine sediments, the intromission angle will only exist when $c_2 < c_1$ and $\rho_2 > \rho_1$. These conditions are not uncommon and may occur in a water-saturated muddy bottom.

A parameter that is key for characterising the lossy properties of a sediment is the normal-incidence reflection loss $NIRL$, a decibel quantity calculated as

$$NIRL = -20 \log|R(\theta_1 = 0)| \tag{2.33}$$

In Figure 2.13, the module and phase of the reflection factors are represented as a function of the angle of incidence for four different types of marine sediment. The same type of graph for the transmission factor is shown in Figure 2.14. The critical angle tends to be smaller on a hard, low-porosity seabed. Total reflection is observed at angles of incidence of 17°, 58° and 75.5° for basalt, gravel and silt, respectively. For the clay-type seabed, total transmission is observed at 80.5°. The $NIRL$ for the bottom materials are 14.3 dB for clay, 11.2 for silt, 7.9 for gravel and 1.9 for basalt. Note that the reference for the decibel scale in (2.33) is $R_{ref} = 1$.

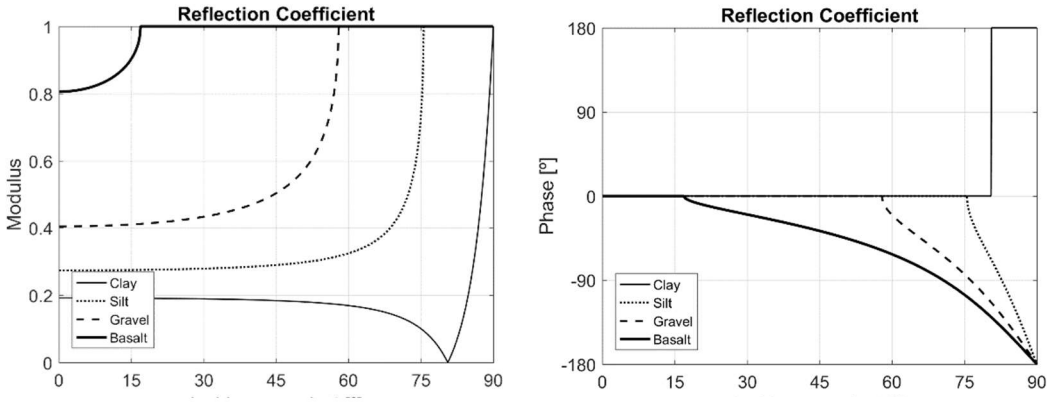

Figure 2.13 Variation of the module (left) and phase (right) of the reflection factor R with incidence angle θ_1 for four types of marine sediment or rock: clay, silt, gravel and basalt. The calculations are based on a simple model of two non-dissipative and homogeneous semi-infinite fluid layers: a top water layer ($\rho = 1000\ \mathrm{kg\,m^{-3}}$, $c = 1525\ \mathrm{m\,s^{-1}}$) and a bottom sediment layer (clay, $\rho = 1500\ \mathrm{kg\,m^{-3}}$, $c = 1500\ \mathrm{m\,s^{-1}}$; silt, $\rho = 1700\ \mathrm{kg\,m^{-3}}$, $c = 1575\ \mathrm{m\,s^{-1}}$; gravel, $\rho = 2000\ \mathrm{kg\,m^{-3}}$, $c = 1800\ \mathrm{m\,s^{-1}}$; and basalt, $\rho = 2700\ \mathrm{kg\,m^{-3}}$, $c = 5250\ \mathrm{m\,s^{-1}}$).

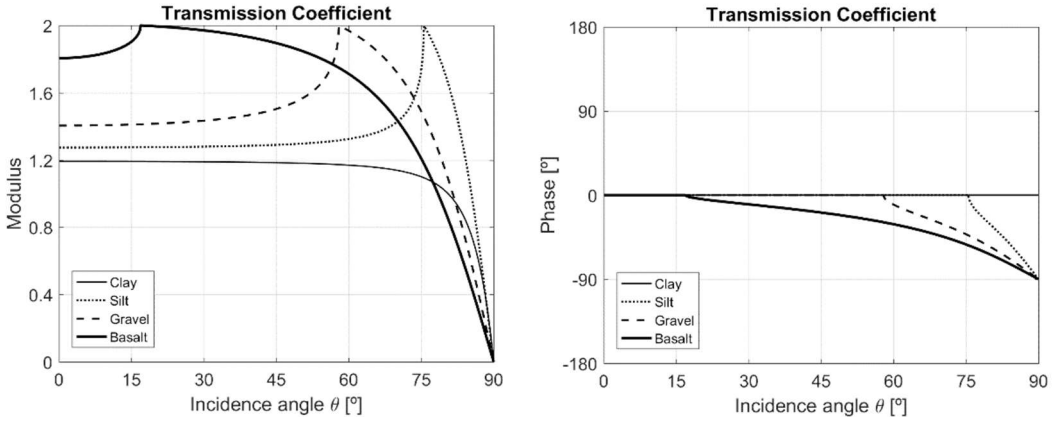

Figure 2.14 Variation of the module (left) and phase (right) of the transmission factor T with incidence angle θ_1 for four types of marine sediment or rock: clay, silt, gravel and basalt. The calculations are based on a simple model of two non-dissipative and homogeneous semi-infinite fluid layers: a top water layer ($\rho = 1000\ \mathrm{kg\,m^{-3}}$, $c = 1525\ \mathrm{m\,s^{-1}}$) and a bottom sediment layer (clay, $\rho = 1500\ \mathrm{kg\,m^{-3}}$, $c = 1500\ \mathrm{m\,s^{-1}}$; silt, $\rho = 1700\ \mathrm{kg\,m^{-3}}$, $c = 1575\ \mathrm{m\,s^{-1}}$; gravel, $\rho = 2000\ \mathrm{kg\,m^{-3}}$, $c = 1800\ \mathrm{m\,s^{-1}}$; and basalt, $\rho = 2700\ \mathrm{kg\,m^{-3}}$, $c = 5250\ \mathrm{m\,s^{-1}}$).

The same conditions in Figure 2.13 and Figure 2.14 are reproduced in Figure 2.15 and Figure 2.16 for the case of a dissipative seabed. The curves are smoothed at higher angles of incidence compared with the non-dissipative case, and there is no total reflection above the critical angle ($|R| < 1$ for $\theta_1 > \theta_c$). Typical values for the absorption coefficient in marine sediments range from 0.1 to 1 dB/λ (Lurton, 2010).

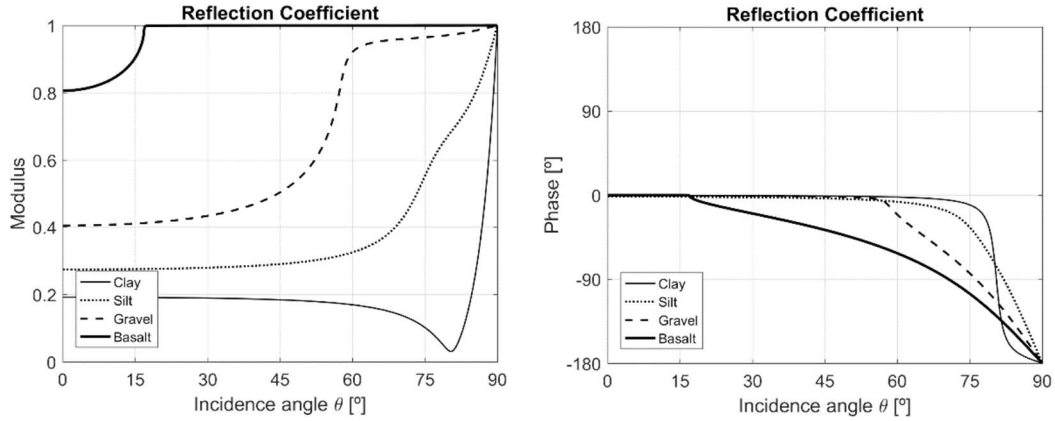

Figure 2.15 Variation of the module (left) and phase (right) of the reflection factor R with incidence angle θ_1 for four types of marine sediment or rock: clay, silt, gravel and basalt. The calculations are based on a simple model of two homogeneous semi-infinite fluid layers: a top water layer ($\rho = 1000\ \mathrm{kg\,m^{-3}}$, $c = 1525\ \mathrm{m\,s^{-1}}$) and a dissipative bottom sediment layer (clay, $\rho = 1500\ \mathrm{kg\,m^{-3}}$, $c = 1500\ \mathrm{m\,s^{-1}}$, $\alpha = 0.2\ \mathrm{dB}\ \lambda^{-1}$; silt, $\rho = 1700\ \mathrm{kg\,m^{-3}}$, $c = 1575\ \mathrm{m\,s^{-1}}$, $\alpha = 1\ \mathrm{dB}\ \lambda^{-1}$; gravel, $\rho = 2000\ \mathrm{kg\,m^{-3}}$, $c = 1800\ \mathrm{m\,s^{-1}}$, $\alpha = 0.6\ \mathrm{dB}\ \lambda^{-1}$; and basalt, $\rho = 2700\ \mathrm{kg\,m^{-3}}$, $c = 5250\ \mathrm{m\,s^{-1}}$, $\alpha = 0.1\ \mathrm{dB}\ \lambda^{-1}$).

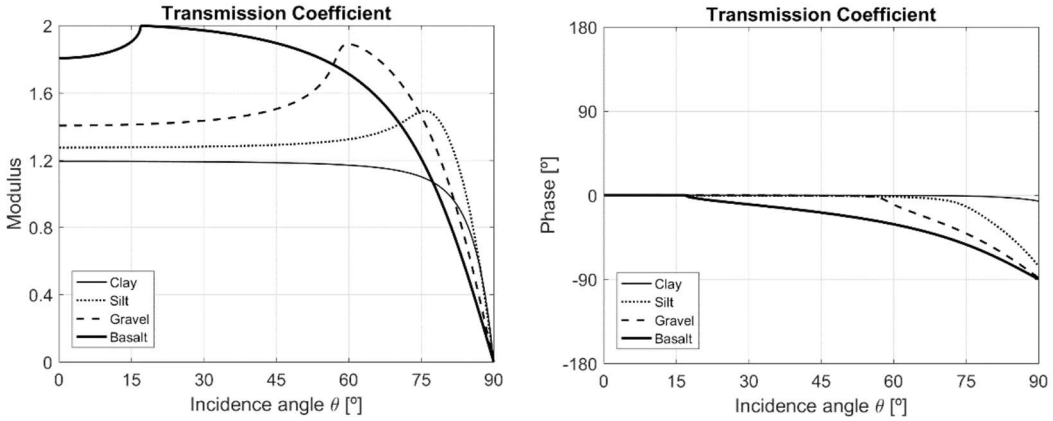

Figure 2.16 Variation of the module (left) and phase (right) of the transmission factor T with incidence angle θ_1 for four types of marine sediment or rock: clay, silt, gravel and basalt. The calculations are based on a simple model of two homogeneous semi-infinite fluid layers: a top water layer ($\rho = 1000\ \mathrm{kg\,m^{-3}}$, $c = 1525\ \mathrm{m\,s^{-1}}$) and a dissipative bottom sediment layer (clay, $\rho = 1500\ \mathrm{kg\,m^{-3}}$, $c = 1500\ \mathrm{m\,s^{-1}}$, $\alpha = 0.2\ \mathrm{dB}\,\lambda^{-1}$; silt, $\rho = 1700\ \mathrm{kg\,m^{-3}}$, $c = 1575\ \mathrm{m\,s^{-1}}$, $\alpha = 1\ \mathrm{dB}\,\lambda^{-1}$; gravel, $\rho = 2000\ \mathrm{kg\,m^{-3}}$, $c = 1800\ \mathrm{m\,s^{-1}}$, $\alpha = 0.6\ \mathrm{dB}\,\lambda^{-1}$; and basalt, $\rho = 2700\ \mathrm{kg\,m^{-3}}$, $c = 5250\ \mathrm{m\,s^{-1}}$, $\alpha = 0.1\ \mathrm{dB}\,\lambda^{-1}$).

The concept of critical angle has important consequences on the propagation of sound in shallow waters ($x \gg h_w$). The acoustic waves emitted by a source at grazing angles will experience a relative low number of bounces at the bottom and lower energy loss per bounce, compared to waves with steeper paths. Accordingly, two regions can be distinguished before and after a critical range x_c, defined as the range at which a sound wave travelling in the direction of the critical angle reaches the receiver after one bounce. If z_s and z_r are the depths of the source and the receiver, and h_w is the depth of the water column, the critical distance is $x_c = (2h_w - z_s - z_r)\tan\theta_c$. In the long-range region ($x \gg x_c$) most of the reflected energy is due to waves emitted at grazing angles ($\theta_1 > \theta_c$). This highlights the importance of the directional pattern of underwater sources and the impact it has on the distances that sound waves can travel in the sea.

2.3.4 Propagation in a Fluid-Elastic Interface

Compressional and shear waves are the two main types of *body waves* that exist in a solid elastic medium. In *shear* (S) or transverse waves, the particle motion is perpendicular to the direction of propagation, whereas in *pressure* (P) or compressional waves oscillations are longitudinal (see Figure 2.17).

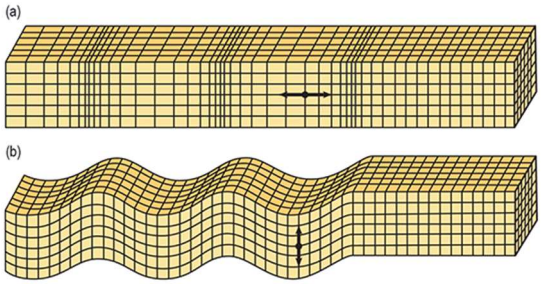

Figure 2.17 Representation of two types of body waves and the direction of particle motion: (a) compressional, (b) shear. In the grapth, the waves propagate to the right (from www.kgs.ku.edu)

For bulk modulus K, shear modulus μ, and density of the solid medium ρ the compressional velocity c_p and shear velocity c_s are given by

$$c_p = \sqrt{\frac{K + \frac{4}{3}\mu}{\rho}} \tag{2.34}$$

$$c_s = \sqrt{\frac{\mu}{\rho}} \tag{2.35}$$

Shear waves are slower than compressional waves. Independently of the characteristics of the solid medium, it will always be satisfied that $c_s < c_p/\sqrt{2}$ (Lurton, 2010).

A sound wave that reaches the sea floor from the water can potentially excite both shear and compressional waves in the solid medium. For the scenario of two infinite homogeneous half-space layers, one *fluid* (the water body) and one *elastic* (the sea bottom), the reflection factor R at the water-seabed interface is given by (Brekhovskikh and Lysanov, 1991)

$$R = \frac{Z_e - Z_1}{Z_e + Z_1} \tag{2.36}$$

$$Z_e = Z_2 \cos^2(2\theta_s) + Z_s \sin^2(2\theta_s) \tag{2.37}$$

$$Z_1 = \frac{\rho_1 c_1}{\cos\theta_1} \tag{2.38}$$

$$Z_2 = \frac{\rho_2 c_2}{\cos\theta_2} \tag{2.39}$$

$$Z_s = \frac{\rho_2 c_s}{\cos\theta_s} \tag{2.40}$$

where Z_e is the total impedance of the elastic medium, Z_1 the characteristic impedance of the fluid medium, Z_2 and Z_s the characteristic impedances of the elastic medium for the compressional and shear waves, θ_1 the plane wave incidence angle in the fluid

medium, and θ_2 and θ_s the incidence angle for the compressional and shear waves in the elastic medium.

The Snell-Descartes law is extended to shear waves as follows

$$\frac{\sin\theta_1}{c_1} = \frac{\sin\theta_2}{c_2} = \frac{\sin\theta_s}{c_s} \tag{2.41}$$

A portion of the incident compressional wave may be converted into a shear wave. Therefore, there are two transmission factors T_p and T_s associated with the compressional and shear waves in the solid medium (Brekhovskikh and Lysanov, 1991)

$$T_p = \frac{\rho_1}{\rho_2}\frac{2Z_2\cos(2\theta_s)}{Z_e + Z_1} \tag{2.42}$$

$$T_s = -\frac{\rho_1}{\rho_2}\frac{2Z_s\sin(2\theta_s)}{Z_e + Z_1} \tag{2.43}$$

The total impedance in the elastic seabed Z_e is smaller than that in a fluid seabed with the same compressional sound speed ($c_s < c_2$), so real bottoms are somewhat "softer" than what the fluid model describes. Furthermore, an elastic seabed supports mode conversion from compressional to shear waves at the fluid-solid interface, which contributes as an additional loss mechanism. Shear waves experience even higher absorption than compressional waves, which will cause an important reduction in their acoustic energy before they return to the water column as compressional waves.

According to the extended Snell-Descartes law, there are two critical angles at the fluid-solid interface, one per wave type. The critical angle for the shear wave is

$$\theta_{c,s} = \operatorname{asin}\left(\frac{c_1}{c_s}\right) \tag{2.44}$$

$\theta_{c,s}$ only exists for $c_s > c_1$, which only occurs for consolidated sediments and rocks. There are three angular regions as a result of the two critical angles: total reflection ($\theta > \theta_{c,s}$), propagation of shear waves only ($\theta_{c,s} > \theta > \theta_c$), and propagation of both compressional and shear waves ($\theta < \theta_c$).

In Figure 2.18, the module and phase of the reflection factors at the fluid-solid interface are represented as a function of the angle of incidence for four different types of marine sediment. The critical and intromission angles for the compressional wave remain the same as in the fluid-fluid model (see Sub-Section 2.3.3). A shear wave critical angle also appears in the basalt case ($\theta_{c,s,basalt} = 37.6°$). In Figure 2.19, absorption losses are included in the solid medium.

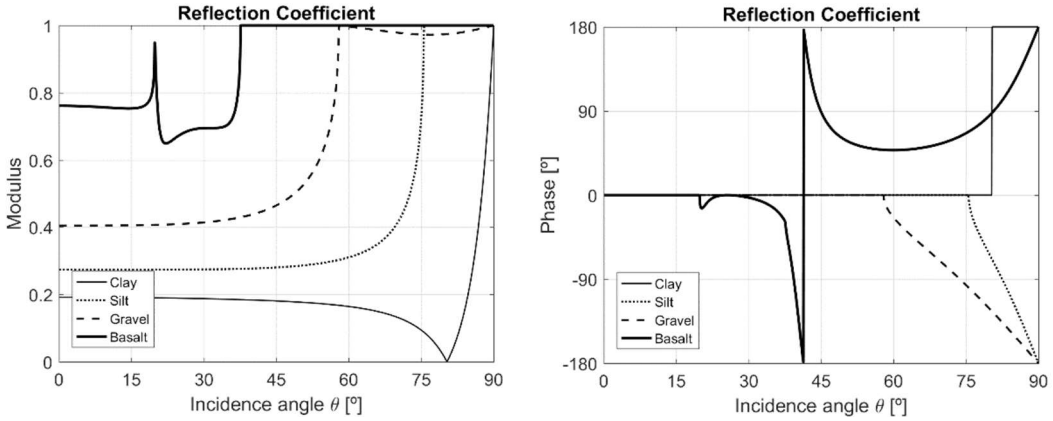

Figure 2.18 Variation of the module (left) and phase (right) of the reflection factor R with incidence angle θ_1 for four types of marine sediment or rock: clay, silt, gravel and basalt. The calculations are based on a simple model of two homogeneous and non-dissipative semi-infinite fluid layers: a fluid top water layer ($\rho = 1000\ \mathrm{kg\,m^{-3}}$, $c = 1525\ \mathrm{m\,s^{-1}}$) and an elastic bottom sediment layer (clay, $\rho = 1500\ \mathrm{kg\,m^{-3}}$, $c_p = 1500\ \mathrm{m\,s^{-1}}$, $c_s = 90\ \mathrm{m\,s^{-1}}$; silt, $\rho = 1700\ \mathrm{kg\,m^{-3}}$, $c_p = 1575\ \mathrm{m\,s^{-1}}$, $c_s = 160\ \mathrm{m\,s^{-1}}$; gravel, $\rho = 2000\ \mathrm{kg\,m^{-3}}$, $c_p = 1800\ \mathrm{m\,s^{-1}}$, $c_s = 360\ \mathrm{m\,s^{-1}}$; and basalt, $\rho = 2700\ \mathrm{kg\,m^{-3}}$, $c_p = 5250\ \mathrm{m\,s^{-1}}$, $c_s = 2500\ \mathrm{m\,s^{-1}}$).

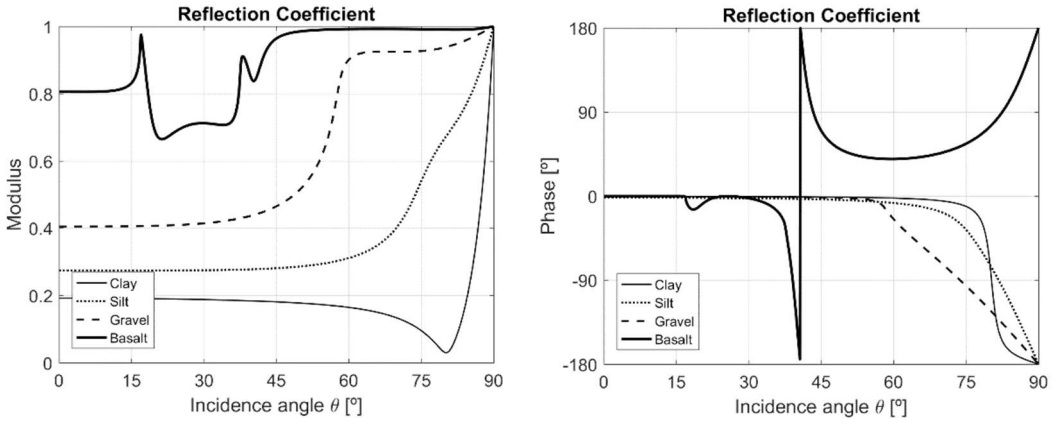

Figure 2.19 Variation of the module (left) and phase (right) of the reflection factor R with incidence angle θ_1 for four types of marine sediment or rock: clay, silt, gravel and basalt. The calculations are based on a simple model of two homogeneous semi-infinite fluid layers: a fluid top water layer ($\rho = 1000\ \mathrm{kg\,m^{-3}}$, $c = 1525\ \mathrm{m\,s^{-1}}$) and a dissipative elastic bottom sediment layer (clay, $\rho = 1500\ \mathrm{kg\,m^{-3}}$, $c_p = 1500\ \mathrm{m\,s^{-1}}$, $c_s = 90\ \mathrm{m\,s^{-1}}$, $\alpha_p = 0.2\ \mathrm{dB}\ \lambda^{-1}$, $\alpha_s = 1\ \mathrm{dB}\ \lambda^{-1}$; silt, $\rho = 1700\ \mathrm{kg\,m^{-3}}$, $c_p = 1575\ \mathrm{m\,s^{-1}}$, $c_s = 160\ \mathrm{m\,s^{-1}}$, $\alpha_p = 1\ \mathrm{dB}\ \lambda^{-1}$, $\alpha_s = 1.5\ \mathrm{dB}\ \lambda^{-1}$; gravel, $\rho = 2000\ \mathrm{kg\,m^{-3}}$, $c_p = 1800\ \mathrm{m\,s^{-1}}$, $c_s = 360\ \mathrm{m\,s^{-1}}$, $\alpha_p = 0.6\ \mathrm{dB}\ \lambda^{-1}$, $\alpha_s = 1.5\ \mathrm{dB}\ \lambda^{-1}$; and basalt, $\rho = 2700\ \mathrm{kg\,m^{-3}}$, $c_p = 5250\ \mathrm{m\,s^{-1}}$, $c_s = 2500\ \mathrm{m\,s^{-1}}$, $\alpha_p = 0.1\ \mathrm{dB}\ \lambda^{-1}$, $\alpha_s = 0.2\ \mathrm{dB}\ \lambda^{-1}$).

2.3.5 Propagation in a Fluid Multi-Layered Medium

The two-layer fluid model in Sub-Section 2.3.3 was used to introduce the concept of impedance and reflection and transmission factors. A more realistic model of the water column and sea bottom consists of a set of horizontal layers of different thickness and

characteristic impedance. Figure 2.20 shows the transmitted and reflected paths and relative amplitudes of an acoustic plane wave propagating in a three-layer fluid medium. The medium consists of a layer of finite thickness in between top and bottom half-spaces. A unit-amplitude plane wave travelling from medium 1 is partially transmitted and reflected at the interface with the second medium. The transmitted wave follows the same process at the interface between media 2 and 3. The sound wave that remains in medium 2 experiences an infinite number of reflections, transmitting a portion of its energy into the half-space layers with each reflection.

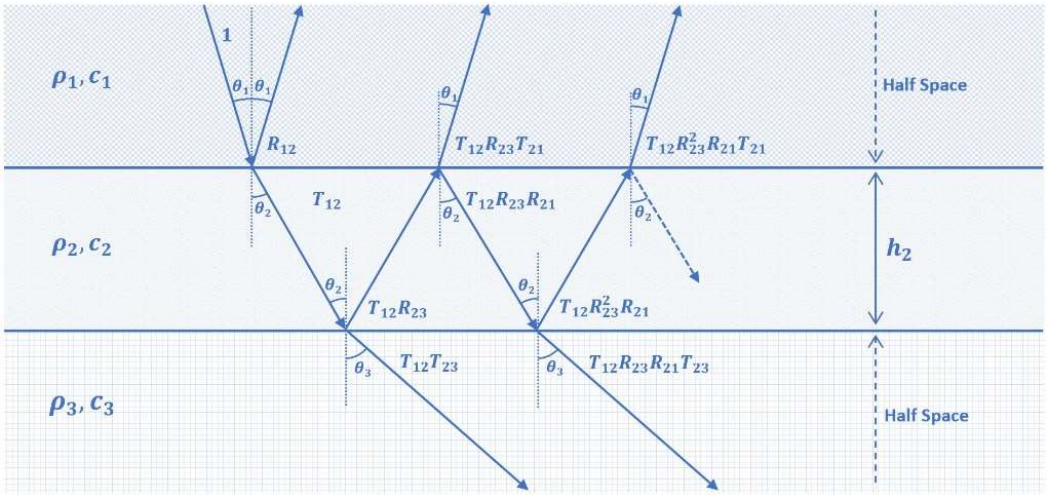

Figure 2.20 Paths and relative pressures for the transmitted and reflected compressional waves in a three-layer fluid medium (after Jensen et al., 2011)

There are three quantities of interest for the characterisation of the acoustic propagation in a multi-layered medium: total reflection factor R_{in}, total transmission factor T_{in}, and total impedance Z_{in}. These are also referred to as *equivalent input* parameters.

The expression for the input reflection factor at interface 12, can be readily obtained as the sum of the amplitudes of the up-travelling waves in medium 1 at the interface (Jensen et al., 2011)

$$R_{in,12} = R_{12} + T_{12}R_{23}T_{21}e^{j2\varphi_2} \cdot \sum_{i=0}^{\infty} \left(R_{23}R_{21}e^{j2\varphi_2}\right)^i \tag{2.45}$$

where R is the *characteristic* reflection factor and T is the *characteristic* transmission factor, defined at the interface between the two media specified by the subscripts. φ_2 is the vertical phase delay of the wave as it propagates through the layer of thickness h_2. The last term in (2.45) is a geometric series that converges to the value $1/\left(1 - R_{23}R_{21}e^{j2\varphi_2}\right)$.

If k is the wavenumber, h the thickness of the layer, and θ the normal angle of incidence, the one-way vertical phase delay in layer m is given by

$$\varphi_m = k_m h_m \cos\theta_m \tag{2.46}$$

The order of the subscript in the reflection and transmission factors matters. Using the relations $R_{12} = -R_{21}$ and $T_{12}T_{21} = 1 - R_{12}^2$ and replacing the summatory function by the value of its convergent series, the final expression for the equivalent input reflection factor at interface 12 is

$$R_{in,12} = \frac{R_{12} + R_{23}e^{j2\varphi_2}}{1 + R_{12}R_{23}e^{j2\varphi_2}} \tag{2.47}$$

Proceeding with the total transmitted amplitudes in a similar way as with (2.45), the final expression for the equivalent input transmission factor at interface 12 is

$$T_{in,12} = \frac{T_{12}T_{23}e^{j2\varphi_2}}{1 + R_{12}R_{13}e^{j2\varphi_2}} \tag{2.48}$$

By replacing R with $R_{in,12}$ and Z_2 with $Z_{in,12}$ in (2.28) and applying (2.47), one can arrive at the expression for the equivalent input impedance at interface 12

$$Z_{in,12} = \frac{Z_3 - jZ_2\tan\varphi_2}{Z_2 - jZ_3\tan\varphi_2}Z_2 \tag{2.49}$$

For an environment consisting of more than three layers, (2.47)-(2.49) have to be applied repeatedly to arrive at the equivalent input parameters. The iterative process starts at the interface of the lowest finite layer. The result from the current layer is used to compute the equivalent parameter of the layer immediately above, until the top is reached. The general equations for the iterative calculation of the equivalent input reflection factor, transmission factor, and impedance are given by

$$R_{in,mn} = \frac{R_{mn} + R_{in,mn+1}e^{j2\varphi_n}}{1 + R_{mn}R_{in,mn+1}e^{j2\varphi_n}} \tag{2.50}$$

$$T_{in,mn} = \frac{T_{mn}T_{in,mn+1}e^{j2\varphi_n}}{1 + R_{mn}R_{in,mn+1}e^{j2\varphi_n}} \tag{2.51}$$

$$Z_{in,mn} = \frac{Z_{in,mn+1} - jZ_n\tan\varphi_n}{Z_n - jZ_{in,mn+1}\tan\varphi_n}Z_n \tag{2.52}$$

where m and n are two adjacent layers, with $n - m = 1$. The subscript $mn + 1$ refers to the interface immediately below the interface mn. The terms with the "in" subscript are *equivalent input* parameters. The terms without the "in" subscript are the *characteristic* parameters, expressed in general form as

$$R_{mn} = \frac{Z_n - Z_m}{Z_n + Z_m} \tag{2.53}$$

$$T_{mn} = \frac{2Z_n}{Z_n + Z_m} \tag{2.54}$$

Z_m and Z_n are the *characteristic impedances* of medium m and n. The reflection and transmission factors R_{mn} and T_{mn} are related to the two media in direct contact with the interface, m and n, which are assumed to be semi-infinite. Note that while the reflection and transmission factors are properties of the interface between two media, the impedance is a property of the medium.

2.3.6 Propagation in an Elastic Multi-Layered Interface

Similar to the case of a multi-layered fluid environment, a real inhomogeneous elastic seabed can be modelled as N-1 homogeneous solid layers of finite thickness overlying a solid bottom half-space. This approach assumes that the properties of the seabed vary with depth but are constant with range. Let's consider a plane wave propagating in a fluid, semi-infinite medium that reaches the seabed at an angle θ_1. Knowing the boundary conditions and acoustic properties of the individual layers, it is possible to calculate the reflection coefficient at the fluid-solid interface. An effective method for the recursive calculation of wave amplitudes at adjacent layers in multi-layered media was developed by Haskell (1953), based on the formulation from Thomson (1950).

The method uses the single-layer *Haskel matrix propagator* $[A(z_n, z_{n+1})]$ to recursively calculate the displacement-stress vector $\boldsymbol{v}$ at the boundaries between any two neighbouring layers, starting from the bottom one. The displacement-stress vector is a four-element column vector containing the relevant displacement and stress values at a layer interface

$$\boldsymbol{v} = (u_x, u_z, \sigma_{zz}, \sigma_{zx})^T \tag{2.55}$$

The terms u_x and u_z are the displacements in the horizontal and vertical directions, and σ_{zz} and σ_{zx} are the stresses at the interface in the vertical and horizontal directions. The conversion between the displacement-stress vectors of connected layers $n+1$ and n, $\boldsymbol{v}(z_{n+1})$ and $\boldsymbol{v}(z_n)$, is performed with the matrix propagator

$$\boldsymbol{v}(z_n) = [A(z_n, z_{n+1})]\, \boldsymbol{v}(z_{n+1}) \tag{2.56}$$

The matrix propagator is formally represented as the product of a diagonal matrix $[D]$, a transfer matrix $[L]$ and its inverse $[L]^{-1}$ (Dunkin, 1965). The diagonal matrix contains the phase change of the compressional and shear waves as they travel within the layer.

$$[A(z_n, z_{n+1})] = [L][D][L]^{-1} \tag{2.57}$$

$$[L] = \begin{pmatrix} j\xi & j\xi & -j\beta & j\beta \\ j\alpha & -j\alpha & j\xi & j\xi \\ 2\mu\xi\gamma & 2\mu\xi\gamma & -2\mu\xi\beta & 2\mu\xi\beta \\ -2\mu\xi\alpha & 2\mu\xi\alpha & -2\mu\xi\gamma & -2\mu\xi\gamma \end{pmatrix} \tag{2.58}$$

$$[D] = \mathrm{diag}\left(e^{j\alpha(z_n - z_{n+1})}, e^{-j\alpha(z_n - z_{n+1})}, e^{j\beta(z_n - z_{n+1})}, e^{-j\beta(z_n - z_{n+1})}\right) \tag{2.59}$$

where μ is the shear modulus, ξ is the horizontal wavenumber, α and β are the vertical compressional and shear wavenumbers, and γ is a parameter dependent on the previous three. The single-layer matrix propagator $[A(z_n, z_{n+1})]$ is defined if the density ρ, compressional and shear velocities $c_{p,s}$, compressional and shear propagation angles $\theta_{p,s}$, and layer thickness $z_n - z_{n+1}$ are known. Simple but elaborate matrix and trigonometric operations can be applied to obtain a set of convenient mathematical expressions, one for each element in $[A(z_n, z_{n+1})]$ (Haskell, 1953; Brekhovskikh, 1980).

The relationship between the displacement-stress vector at the lowest boundary $\boldsymbol{v}(z_{N+1})$ and that at the boundary between the top fluid and solid layers $\boldsymbol{v}(z_2)$ is

$$\boldsymbol{v}(z_2) = [A(z_2, z_{N+1})]\, \boldsymbol{v}(z_{N+1}) \tag{2.60}$$

$$[A(z_2, z_{N+1})] = \left[A^{(2)}\right] \cdot \left[A^{(3)}\right] \cdot \ldots \cdot \left[A^{(N)}\right] \tag{2.61}$$

Once the matrix propagator $[A(z_2, z_{N+1})]$ has been calculated, the *reflection coefficient* R at the fluid-solid interface can be obtained. This is achieved by first finding the displacement-stress vectors at the boundaries of the fluid (top) and elastic (bottom) half-spaces, as described by Brekhovskikh (1980). The reader must be warned that the final expression for the reflection coefficient found in Brekhovskikh (1980) is incorrect, as pointed out by Stephen E. Ainsworth in his PhD thesis (Ainsworth, 2011). A valid expression for R can be easily derived from the formulas and concepts presented in Brekhovskikh's work.

The greatest advantage of the matrix propagator method is that a number of artificial homogenous sublayers can be combined to obtain a step-wise representation of any inhomogeneous layer in horizontally stratified media. However, the direct implementation of this method runs into numerical instabilities that occur at high frequencies and high angles of incidence in lossy media. Dunkin (1965) was the first to identify and propose a solution to the problem of numerical instabilities in the Haskell-Thomson approach. Dunkin demonstrated that working with the matrix of subdeterminants of the propagation matrix led to a more stable algorithm. Lévesque and Piché (1992) presented an improved, stable version of the Haskell-Thomson approach with generalised expressions for the reflection and transmission coefficients.

2.4 Seismic Exploration

Seismic exploration is one of many noise-emitting activities that take place at sea. Others include windfarm installation, near-shore construction, site and geophysical surveys, and decommissioning. The range of active acoustic sources and operations related to oil and gas exploration that can produce significant noise levels are numerous. An in-depth review of these sources and their noise-generating mechanisms, typical source levels and frequency spectra was carried out by the author. The review, commissioned by the Sound and Marine Life Joint Industry Program (JIP), was written as two separate reports, one for general industrial sources (Jiménez-Arranz et al., 2020a) and one for pile driving activities and noise abatement solutions (Jiménez-Arranz et al., 2020b).

The categories of acoustic source addressed in the two reports are *seismic sources* (air gun, sparker, boomer, marine vibrator, water gun, sleeve exploder), *engineering sources* (SBES, MBES, SSS, SBP), *acoustic deterrents* (ADD, AHD), *acoustic modems*, *underwater acoustic positioning*, (LBL, SBL, USBL), *acoustic releases*, *ADCPs*, *vessels* (commercial, supertankers, icebreakers, medium-sized vessels, boats and hovercrafts), *dredging* (CSD, TSHD, GD, BLD, BHD), *drilling* (fixed platforms, jack-up rigs, artificial islands, caissons, semi-submersibles, drill ships), *production* (fixed platforms, artificial islands, caissons, FPSO, tension-leg, spar-platforms, semi-submersibles, drill-ships), *pipe-laying*, *decommissioning*, *aircrafts*, and *pile driving* (impact, vibratory). The author also produced a peer-reviewed proceedings paper for the 5th International Conference on the Effect of Noise on Aquatic Life in The Hague, presenting the results from models and measurements carried out for the acoustic characterisation of a large 6th generation semi-submersible (MODU) using drift buoys and an autonomous surface vehicle (ASV) (Jiménez-Arranz et al., 2020).

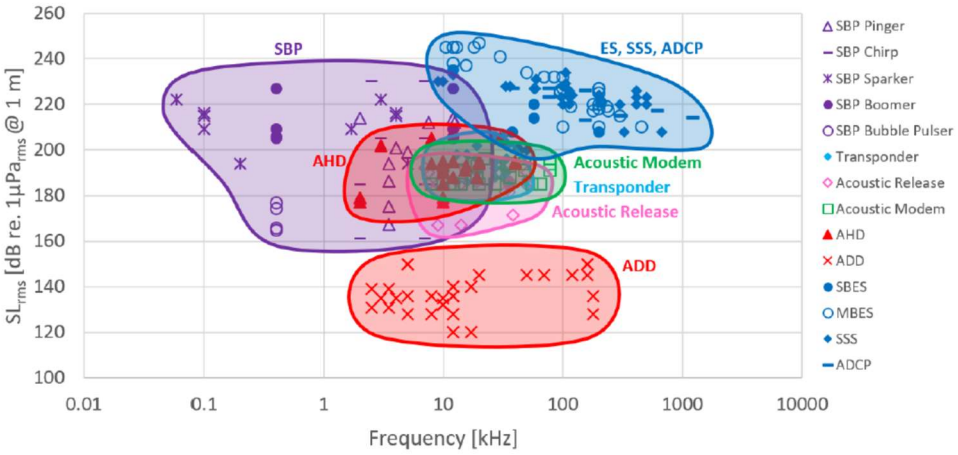

Figure 2.21 Sound level and spectral range of various categories of underwater engineering sources. The markers represent source levels at the operating frequency of the device (e.g. pinger, bubble pulser) or at the lower and upper frequencies of its operating band (e.g. chirp, air gun, acoustic modem). Extracted from Jiménez-Arranz et al. (2020a).

In this section, we describe the general aspects of seismic surveys to then introduce the *air gun*, the main acoustic source used in these operations. A discussion about the rest of noise sources related to the oil and gas industry is out of the scope of this thesis, but the reader is welcome to access the two mentioned reports available at the digital library[1] of the Sound and Marine Life's website.

2.4.1 Seismic Surveys

Seismic surveys are aimed at studying the composition of the seafloor's upper crust. These surveys are conducted for a variety of reasons: checking foundations for roads or large structures or detecting resources like groundwater or minerals are some of them, but most of commercial seismic surveys are conducted for detection (*exploration*) and monitoring (*production*) of oil and gas reserves (OGP, 2011). Every marine surveying technique uses a particular configuration of source and receivers to send sound waves into the earth and capture the reflected energy. The strength of the reflected wave and the time it takes to travel through the different layers in the seafloor and back to the receiver are processed to get a stratigraphic image of the earth's crust.

There are four big groups of marine seismic surveys, classified according to its receiver characteristics and operation. These are towed streamer (TS), shallow water or transition zone (TZ), ocean bottom node (OBN), and vertical seismic profiling (VSP). Towed streamer and ocean bottom are the most common.

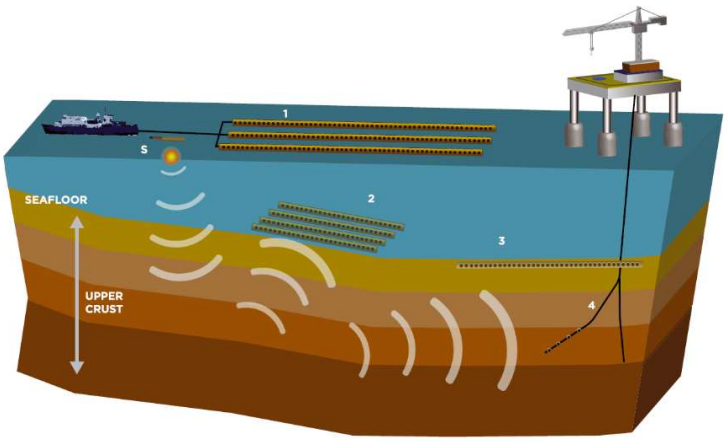

Figure 2.22 Illustration of the different marine seismic acquisition techniques. All marine seismic surveys involve a source (S) and a particular receiver configuration (numbers): 1. Towed streamer, 2. Ocean bottom node, 3. Buried array, 4. VSP (from OGP, 2011).

The operations of a seismic survey with towed streamers consist of five main stages: mobilisation, deployment and testing, run-in, shooting, run-out, and line-change. During *mobilisation* supplies are brought to the ship, the crew arrives and the survey layout and plan are provided. In the *deployment and testing* phase the streamers and air gun

[1] https://www.soundandmarinelife.org/library/

array are deployed to ensure the neutral buoyancy of the streamer and the correct operation of the air guns and compressor. The *run-in* sets the start of the survey, where the seismic vessel approaches the start of line after the streamers have been aligned. A 30-minute pre-survey visual (MMO) and acoustic (PAM) monitoring takes place to ensure that no marine mammals are in the area before the air guns start emitting. During the *shooting* phase the air gun array is fired at regular distance intervals, every 10-12 seconds for usual vessel speeds. An initial *ramp-up*, where the air guns are sequentially activated to slowly increase the acoustic power, takes place to alert any nearby marine mammals. Once the ship reaches the end of a sail line, the array is turned off (or *powered down* as a mitigation measure) and the vessel sails to the new line.

2.4.2 The Air Gun

The air gun is the most powerful acoustic source used in seismic exploration. It consists of one or more pneumatic chambers capable of creating a fast acoustic pulse by releasing the enclosed compressed air rapidly into the water. The pulse generated by an air-gun is *predictable*, *repeatable* and *controllable* (OGP, 2011).

The defining properties of an airgun are the *gun pressure* and the *volume* of the chamber. Typical airgun volumes vary from 20 to 800 in^3. A broadly used gun pressure value in seismic industry is 2000 psi (Landrø and Amundsen, 2010).

The *air-gun array* is the source used in marine surveys. It consists of a few groups of 6-8 aligned air guns. Each group is called a *string* and an air-gun array can be made up of 1-6 of these strings (see Figure 2.23 and Figure 2.24). The volume of an air-gun array is the sum of the volumes of the individual guns, and is typically in the range of 3000-8000 in^3. The air guns hang beneath floats, between 3 and 10 m below the sea surface and generally at 6 m. During a survey the array fires every 10-15 s. The shot interval depends on the firing distance and the speed of the source vessel, both constant.

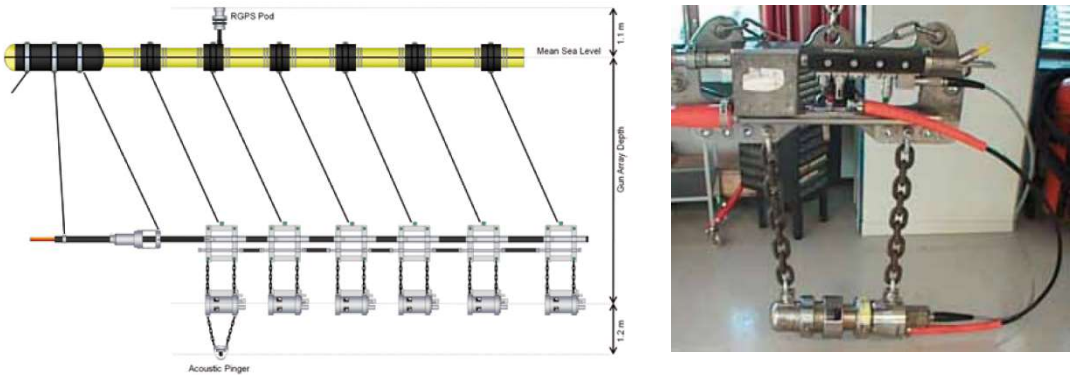

Figure 2.23 Side view of a string of airguns. There are 12-guns arranged in 2-gun clusters. The yellow tube provides flotation and the guns hung from it at the desired depth. Precise positioning is provided by combining information from the RGPS in the float and the acoustic pingers under the guns (left). Single air gun suspended (right) (from OGP, 2011)

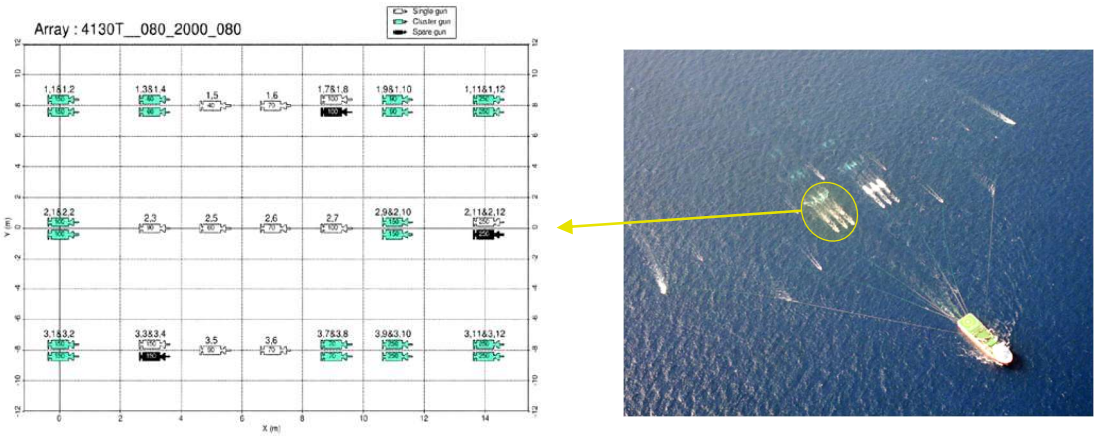

Figure 2.24 Airgun array layout (left) and deployment (right). Two arrays are towed.

The *strength* or peak-to-peak amplitude of the pulse generated by the array in the vertical direction is in the range of 10-100 bar-m (240-260 dB re µPa@1m). The *strength* of an airgun array is (1) linearly proportional to the number of guns in the array, whenever all the sources integrating the array have the same characteristics, (2) close to linearly proportional to the firing pressure of the array, and (3) roughly proportional to the cube root of its volume V, namely directly proportional to $\sqrt[3]{V}$ for a single airgun and less than that for an array (Caldwell and Dragoset, 2000).

The *pressure signature of a single air gun* is the time response of the pressure of the air gun emitted at 1 m in the vertical direction. Two parameters are used to characterise this signature: the *strength* or amplitude peak-to-peak of the primary acoustic pulse, and the *bubble period*. The energy of the primary pulse of an airgun is concentrated between 10 and 200 Hz, which is the typical range of interest in seismic surveying.

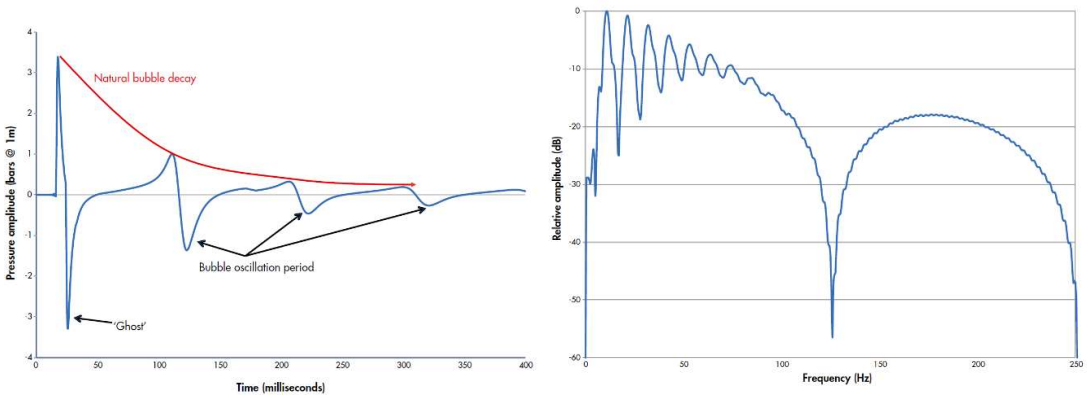

Figure 2.25 Source signature of a 150 in3 airgun in bars (left) and its normalised amplitude spectrum in decibels (right) (from OGP, 2011).

The *pressure signature of an air-gun array* is measured in the far-field (200-250 m beneath the array), where the array behaves as a point-source. The signature is then normalized to a distance of 1 m. The main parameters used to characterise the array's

signature are the strength of the primary pulse and the *primary-to-bubble ratio* (PBR). The higher the PBR the closer the signature is to an ideal pulse (see Figure 2.26 left).

When an air gun fires it creates a high-level pulse that is followed by lower amplitude oscillations (see Figure 2.25, left). These oscillations are caused by the bubbles formed from the air released by the air gun. The bubbles experience a harmonic damped oscillation as they reach the equilibrium state. Air guns with higher volume, higher gun pressure, or smaller depths produce bubbles with lower dominant frequency.

The *source ghost* refers to the phase-inverted acoustic wave reflected at the sea surface and combined with the original downward travelling wave. The ghost introduces a comb filter effect in the "free-field" spectrum that reduces the effective bandwidth (see Figure 2.25 and Figure 2.26, right). The depth of the source is a key design parameter in a seismic survey, as it determines the width of the useful frequency band, delimited by the first (0 Hz) and second notch in the comb filter. High resolution, shallow subsurface surveys require *shallower depths*, which result in lower resolution due to the shift of the second notch to higher frequencies. Deep penetration surveys require *deeper depths*.

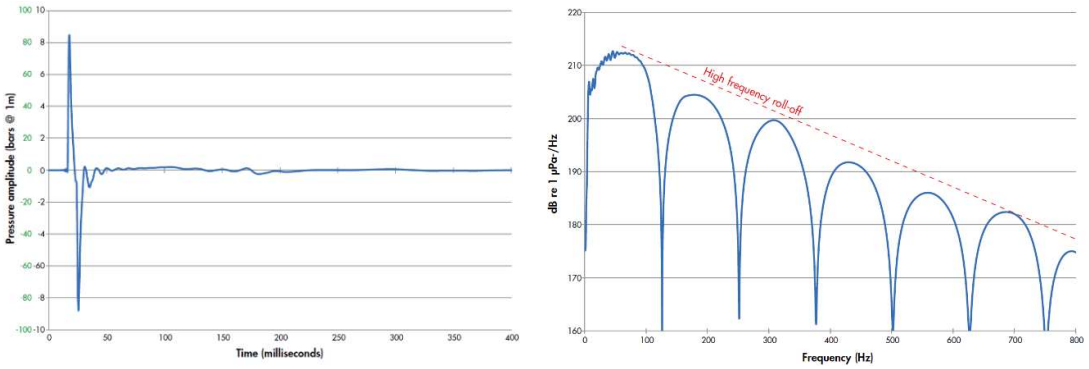

Figure 2.26 Source signature from a 4450 in3 array with 33 guns (left) and its normalised amplitude spectrum in decibels (right) (from OGP, 2011)

There are several reasons for using array configurations rather than individual air guns. The first is to increase the power of the source. The basic idea is that a source array of n single sources produce n times the acoustic pressure of the single source. Neither the chamber pressure nor the total volume of the array have a significant influence on its strength. The second is to minimize the bubble oscillations by *tuning* the array. The tuning consists in combining different air gun volumes, each associated with a different bubble period, to achieve constructive summation of the primary peak and destructive summation of the bubble oscillations.

The acoustic energy emitted by an airgun array is not the same in all directions. The pressure levels emitted horizontally are 15-24 dB lower, which indicates that the energy of the array is mainly projected downwards. Factors that contribute to this directional radiation are the arrangement of the sources in a horizontal plane, the dimensions of the array, the distribution of the airgun volumes within the array, the sequential (non-

simultaneous) firing of the individual airguns, and the proximity of the array to the water surface. Figure 2.27 shows that most of the energy between 20 and 180 Hz is projected downwards, high frequency radiation presents secondary lobes, and the array is more directive in the cross-line direction.

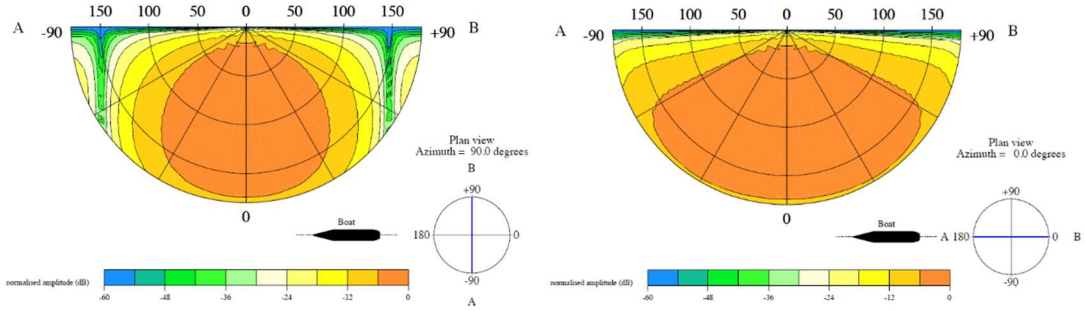

Figure 2.27 Modelled sound pressure level emitted by a 1760 in3 airgun array in a vertical plane in cross-line (left) and in-line (right) directions. The vertical angle of emission is represented by the radial lines and the frequencies (0-180 Hz) are shown in concentric circles. The coloured pattern represents the sound pressure levels. The array has two identical strings with three paired airguns, which leads to an asymmetry in the horizontal directivity pattern.

2.5 Impact of Seismic Noise on Marine Fauna

2.5.1 General Effects of Noise

Industrial and commercial activity in the oceans has contributed to an overall increase in underwater noise levels over the years. Shipping alone has caused an average increase in low-frequency ambient noise levels of over 3 dB in a matter of a few decades (Frisk, 2012). Important noise-emitting activities include pile driving, seismic exploration, UXO clearance, site and geophysical surveying, drilling, dredging and rig decommissioning. Many are confined to small areas, but the number, frequency and extension of these activities, and the ability of some of them to produce detectable noise levels at tens of kilometres, represent a high risk to marine fauna with potential short and long-term consequences at individual and population levels.

Underwater anthropogenic noise has become a topic of growing concern. Interest and awareness about the issue developed in the late 1970's, at a time when concerns and increased opposition existed around commercial hunting of marine mammals and the effects human-led environmental degradation (Richardson et al., 1995). Since then, the number of studies about the impact of underwater noise on marine organisms has dramatically increased. In this section, we will give a brief overview of the effects of noise on marine fauna, with especial focus on the impact of seismic sources, to then discuss standard mitigation solutions.

Marine animals rely on sound for communication, navigation, foraging, mating and predator avoidance. Noise has the potential to disrupt these activities that are vital for their survival. Masking, physiological and physical impact, and induced changes in behaviour are some of the consequences of noise exposure. If prolonged over time, separate instances of exposure on individuals can have effects on population, affecting general presence and migration routes. All these effects are briefly discussed below.

Masking occurs when noise interferes with a sound of similar frequency and amplitude, reducing the ability of the animal to detect and interpret it. The *Lombard effect*, a common behavioural effect of masking, describes the modification of the amplitude, duration and spectral content of an animal's vocalisation in an attempt to make the signal audible to another animal.

Behavioural responses to noise are difficult to assess in practice. The reason is that a noisy environment will not necessarily trigger a response as this will depend on several factors, including habituation, sensitisation, age and sex, auditory sensitivity, and activity at the time of exposure (Gordon et al., 2004). *Habituation* occurs when the response to the stimulus becomes weaker after repeated exposures. If the animal had an aversive reaction to the noise, it can become *sensitised.* The animal will establish a link between the unpleasant event and the noise, causing a behavioural reaction even at lower levels, when the signal is just audible. Different individuals will also react differently depending on their role, age and gender. The activity or behaviour of the animal at the time of exposure can also determine the degree of the response. If the animal is foraging or mating, it will be prone to endure higher noise exposures. Short-term behavioural responses caused by noise include vocal modifications and avoidance manoeuvres such as increase in distance from the source, change of course and speed, and changes in surfacing and diving patterns. Long-term effects include a reduced presence in the area or disturbance of biological patterns such as nursing or mating, with potential implications in *ecosystem dynamics*.

Among the *physiological* effects of noise are elevated respiration rates and heart rate and increased levels of stress hormones. Stress and general changes in behaviour are related to the release of hydrocortisone.

Physical effects are caused by high energy events and long exposures. The formation of gas bubbles and tissue damage associated with decompression sickness (the *bends*) can occur on dolphins during long dives (Ridgway & Howards, 1982) or as a result of exposure to high-pressure sounds (Crum & Mao, 1996). Long dives can be triggered by an avoidance response to noise. A less dramatic physical effect of noise is the temporary reduction of hearing sensitivity, known as temporary threshold shift or TTS (Southall et al., 2019). Higher intensity sounds or longer exposures can lead to permanent hearing damage, characterised by a permanent threshold shift or PTS.

It is recognised that noise exposure of individuals will ultimately have an effect on the population growth rates and ecosystem dynamics, as a result of the affected fitness and

fecundity of those individuals (Borsani et al., 2015). A modelling framework developed by the National Research Council, known as PCAD, integrates these individual behavioural and physiological effects to predict the consequences at a population level (NRC, 2005).

2.5.2 Impact of Seismic Air Guns

As with other types of sources, the behavioural and physiological responses of marine animals to the noise produced by seismic surveys are quite variable (Gordon et al., 2004). However, there is mounting evidence of the impact of seismic pulses on marine organisms from multiple studies. Stone (2003) presents an exhaustive analysis of changes in behaviour of marine mammals during seismic surveys using data from sightings of over 28,000 individuals occurring during 201 seismic surveys. Statistically meaningful results showed at least some form of response during the periods of air gun activity for most of the taxonomic groups studied. Gordon et al. (2004) also present a comprehensive review of the impacts of seismic surveys on marine mammals from available literature. The extent by which different mammal groups are affected by air gun pulses vary considerably, but cases of negative responses are reported for all of them.

Given the great variation of responses, a systematic analysis of the effect of seismic sources on the different marine mammal groups would not necessarily lead to solid conclusions. A review of the reactions reported by different studies on each group also appears to be unnecessary since such reviews have already been made by other authors. Instead, we summarise the most common responses and their magnitude for a few marine mammal groups based on the work from Stone (2003) and Gordon et al. (2004).

Odontocetes are high-frequency specialised marine mammals, and as such are likely to be less affected by the primarily low-frequency content of air gun arrays. This group has been observed to tolerate seismic pulses down to ranges of 1 km from the source. Nonetheless, due to the impulsive character of air gun emissions these sources can still emit energy at frequencies of up to 20 kHz. Full-sized air gun arrays can therefore be heard by odontocetes at ranges of 8 kHz or more (Goold and Fish, 1998). During periods of air gun emission, dolphins were shown to remain further from the source compared to quiet periods. Avoidance, increase in swimming speed, and a reduction in positive interactions such as *bow-riding* are some of the observed behavioural responses.

Baleen whales are low-frequency specialised marine mammals, making them more vulnerable to masking from seismic pulses. Some degree of localised avoidance, with further recorded distances during shooting periods, has been observed. Clear avoidance of *bowhead whales* for levels between 150-180 dB re 1µPa (6-8 km) and subtle avoidance for levels of 120-130 dB re 1µPa (20 km) have been reported. Avoidance reactions of *gray whales* such changes in speed and course, or shielding using the surrounding topography to reduce the noise exposure have also been observed. Sound levels of ~170 dB re 1µPa have shown to produce an avoidance response on 50% of the individuals. One study

showed that *humpback* whales remained at distances higher than 3 km during air gun operation. *Blue whales* have shown to stop vocalising at long-distances from the source during shooting periods, suggesting a higher sensitivity to air gun pulses than other types of whales. *Sperm whales* are low-frequency specialised marine mammals, however results about their response tend to be contradictory.

Studies on *pinnipeds* also show mixed results, some reporting avoidance (rapid swimming away from the source) and others strong habituation with no difference in behaviour between "quiet" or active periods of the air gun array.

Of particular importance is the impact of seismic operations on fish, as it can have indirect effects on marine mammals through a reduction in prey abundance. Fish with swim bladders are sensitive to pressure and, as such, more susceptible to be affected by the low-frequency emissions from air guns. The effect of seismic operations on fish ranges from lethal injuries at short ranges to strong avoidance leading to displacement.

2.5.3 Noise Mitigation

There exist various mitigation methods to reduce the exposure of marine animals to noise. These include sound level reduction, deterrence, spatiotemporal restrictions, and supervision of exclusion areas (Weir & Dolman, 2007).

Regarding *sound level reductions*, emissions can be minimized by acting *on the source* or *on the propagation path*. Working on the source and on the characteristics that make its acoustic signature harmful to marine life is the most effective mitigation practice. Simply the decision of what source to use (if there is an alternative) would make a difference. Mechanical and operational improvements will also help to shape the acoustic emission into a signal that is more suitable for the particular surveying application and less harmful to marine animals (Merchant et al., 2020). This can be achieved, for example, by shifting the main spectral energy content outside the range of audibility of the animal, controlling the directivity or including attenuation mechanisms. Acting on the sound propagation path is generally a more expensive and less effective way to control the acoustic emission from the source. Acoustic barriers (e.g., bubble curtains), a common solution for pile driving, are aimed at minimising the transmission of acoustic energy into the water column (Verfuss et al., 2019a; Jiménez-Arranz et al., 2020).

In relation to *deterrence*, Acoustic Deterrent Devices (ADDs) emit an acoustic signal averse to certain marine mammals and have been widely used to keep the animals away from aquaculture facilities and fishing nets. Other than foraging, marine mammals can approach marine industrial activities or other hazardous environments led by their curiosity. In marine seismic surveys and offshore pile driving, a progressive increase of the acoustic output of the source (*ramp-up*) takes place previous to full power operations to warn marine mammals. It is also typical in a seismic survey for a small air gun to remain active during the line turn of the source vessel. The effectiveness of the ramp-

up or ADDs to dissuade marine mammals has not been demonstrated (Coram et al., 2014) and there is no clear answer about whether these solutions are of any use or are just worsening the situation by adding noise into the environment (Findlay et al., 2021).

Choosing the time of the year when is less probable to find certain types of endangered marine species is also a good way to reduce the risk of noise impact. Selecting the location of the activities is not an option in many cases. If the possibility exists, the operations can be moved to locations with a lower presence of sensitive marine species, or to areas where the topography will constrain the long-range propagation of noise.

The *supervision* of exclusion areas is a key mitigation strategy, as it can reduce the noise exposure even when all the other options for noise reduction have been exhausted. A visual inspection of the water surface is carried out by Marine Mammal Observers (MMOs) to ensure that no marine mammals are found within the limits of the exclusion area or *mitigation zone*. For practical reasons, a mitigation zone is normally defined as a circular area, however its shape can be complex depending on the bathymetry, geology of the seafloor, water properties, and source directivity (Blondel, Jiménez & Wyatt, 2016).

The *mitigation zone* is the region around the source where the sound pressure levels are expected to have an adverse effect on the marine animal. The coverage of the mitigation zone is determined by specific threshold levels (NMFS, 2018), which vary with marine mammal group (e.g., Otariidae or Phocidae pinnipeds), type of signal (continuous or impulsive), acoustic parameter ($cSEL_{24h}$ or SPL_{pp}), or degree of disturbance (behavioural effect or temporal threshold shift). Formally, the exclusion range is the distance from the source at which the target threshold level has been exceeded. Defining the exclusion areas for the different threshold levels is part of the Sound Source Verification process.

2.6 Sound Source Verification

2.6.1 Definition and Evolution of SSV

A good understanding of the acoustic output of a source and the propagation environment are essential for estimating the acoustic footprint of a marine activity. Sound Source Verification (SSV), sometimes also referred to as Acoustic Field Verification (AFV) or Acoustic Field Characterisation (AFC), refers to the process of validating the results from an available acoustic model using field measurements. This is the formal definition, but the concept is wider in practice. The terms SSV, ASV and AFC are used to describe any approach whose end goal is to produce a sound field map from which the acoustic properties of the source, the propagation losses of the environment, or the exclusion zones for sensitive local species can be obtained.

For example, a sound field mapping survey may be planned to measure sound levels at a few locations with the objective of verifying the results of a sound field model

generated before the survey. It can also occur that a model is not available, and the measurements are used instead to calibrate a sound field model produced during or after the SSV survey. But not always the company in charge of the SSV survey has the capabilities to simulate the sound field, in which case the sound field mapping strategy is designed to collect a larger, more representative number of measurements that can be confidently used to extrapolate the data to longer or shorter ranges.

In the absence of a propagation model, the transmission loss curve from (2.15) can be used to fit the range-dependent sound level data. In this way, sound levels can be estimated at 1 m to obtain the *source level* (see Sub-Section 2.1.3) or at long ranges to compute the *exclusion zone* for a specific target threshold (see Sub-Section 2.5.3). This is the most common approach due to its simplicity and cost. A more elaborate alternative consists in optimising the environment parameters of a sound propagation model based on the measured sound levels (see Section 2.7 for details about sound propagation models). This method has the advantage of producing a calibrated sound field map with greater accuracy outside the measurement range.

SSV procedures will change depending on the requirements, and on the resources and experience of the SSV surveyor. However, they all share one essential step: the sound field mapping. The use sound field measurements for the assessment of noise impacts on marine life is a recent requirement in some countries, but is has been part of guidelines and industry recommendations for around two decades. Until the breakthrough in the understanding of the hearing response of marine mammals (Southall et al., 2007), and more recently of fish and turtles (Popper et al., 2014), recommendations were lax and unstandardised. Before then, the use of an exclusion zone of 500 m, as stipulated by the JNCC guidelines from 1994 (JNCC, 2015), was the standard for seismic operations. Currently, there is a wider consensus on the way underwater measurements should be made (Robinson et al., 2014) and procedures to be followed to calculate exclusion zones from sound field data (NMFS, 2018).

The author of this thesis has worked on SSV projects since 2014, during a time when the use of field measurements for sound field characterisation was still a proof of concept. In the next sub-section, we introduce some of the most common data acquisition techniques and platforms employed on SSV. Having experienced the evolution of these techniques ourselves, with direct involvement in their configuration, maintenance and deployment, we would like to present a summary of the systems encountered over these years and their deployment strategies.

2.6.2 Data Acquisition Platforms

The sampling strategy for sound source verification is conditioned by the use that will be given to the data, the deployment method and the constraints imposed by the operations in the area. The most common data acquisition methods are vessel-towed hydrophones, drift buoys, moored acoustic recorders, and unmanned vehicles.

Three types of data are needed to generate a sound field map: audio, position and log information. The acquisition platforms record audio and position data at the receiver. The source position is typically provided by the bridge personnel in the source vessel. A detailed log report is produced during the survey, with deployment times and positions, source activity and configuration, sea state information, and other relevant events that can help interpret and process the collected data.

The processing required to produce these maps varies between companies. In many cases the approach is poorly described and the software routines are rarely available. To address these limitations, we have developed a software in MATLAB to produce sound field maps for individual sources based on raw audio and position data. The software structure and capabilities are described in Chapter 3, with a detailed analysis of its statistical detection algorithms presented in Chapter 4.

Drifting Buoy

Drifting buoys are a type of moving platform that takes advantage of tides and currents to collect audio and position data at multiple ranges and angles from a source. This platform is typically used in sets of four or more units and are most effective in areas with strong currents. The buoys are deployed upstream at the beginning of a tidal cycle and allowed to drift downstream for data collection (see Figure 2.28). The initial positions are selected according to a planned sampling strategy designed based on the required area coverage, tide charts and weather forecast. This methodology is especially effective in tidal regions with particularly fast currents, where buoys can cover distances of several kilometres in a single deployment.

Figure 2.28 Set of 6 modular buoy units from Seiche used in The Cook Inlet deployment (left) and diagram of the deployment method (right)

In the design of the sampling strategy the topology of the area and the tides must be considered. Tidal patterns vary with region. Most areas have two high tides and two low tides each day (semidiurnal), but others like the Gulf of Mexico have only one high and one low tide each day (diurnal). The speed of the tidal current depends on the tidal pattern and *tidal ranges* (the amplitude difference between high and low tide), and this affects the covered distances, the spacing between sampling points and the duration of

the deployment. The sampling strategy should be designed to obtain a representative sample of the sound field, but also to prevent the buoys from straying or moving across shallow tracks, which could affect the quality of the data.

Concerns about the presence of moving objects during sensitive marine operations such as seismic surveys, pile driving, or drilling lead restrictions in their use. Particularly, drift buoys are allowed when (1) the buoy is deployed far away from the source (CPA of not 500 m), (2) there is low to moderate activity of nearby vessels, and (3) there is a dedicated fast vessel for quick recovery in the event of a stray buoy.

Figure 2.29 Stages in a drift buoy deployment: (1) *Strap*, the electronics box and the mast are attached to the buoy at the time the deployment location is reached; (2) *Deploy*, the buoy is deployed and the vessel moves to the next deployment location, then the process is repeated with the rest of the buoys; (3) *Search*, about six hours later the satellite tracker is used to locate the buoys; (4) *Collect*, one after the other, the buoys are retrieved, the boxes are collected and the data is backed up.

To the knowledge of the author, Seiche was the first company to design and employ rugged data acquisition buoys[2] for large dataset recording in tidal regions. Another

[2] https://www.seiche.com/modular-buoy-systems/

model that came after is the RUBHY RB-SDA14[3] system from RTSYS. The Acoustic Monitoring Buoy System (AMBS) from Konsberg Maritime are individual *peli case* data acquisition units, similar to the electronics box in Seiche's system, which can be installed on an existing buoy.

Each modular buoy system from Seiche consists of a *mast* with a LED beacon and a radar reflector for visual and remote RF detection, the *data acquisition box* containing a satellite tracker for real time positioning, the ADC unit (DAQ) and battery, the *float* that provides the buoyancy and stability to the system, and the *wet leg*, a metal housing that protects the hydrophone cable (see Figure 2.30). The hydrophone array consists of two coincident preamplified hydrophones of different gains, selected to extend the system's dynamic range to reduce the chance of digital clipping or low SNR. The buoys can record locally to a hard drive or include an RF link for remote transmission to a receiver station, where the audio data is visualised and recorded in real time.

Figure 2.30 Modular buoy system from Seiche (left) and RUBHY SDA14 from RTSYS (right).

Towed Hydrophone

The *towed hydrophone* is a standalone system consisting of a data acquisition unit, a deck hydrophone cable, and a hydrophone array. The system can be installed on board of a vessel to gather audio and position data while the vessel controls the recording path. This platform benefits from its flexibility and easy installation, deployment and recovery. If installed on a vessel that the SSV operator has full control over, the system gives the freedom to collect data anywhere in the area that the vessel is allowed access to. There are two main limitations to this system: (1) when towed, the hydrophone is exposed to flow noise and vibrations transmitted through the cable, and (2) the exact

[3] https://rtsys.eu/buoys

position of the hydrophone is unknown and an estimate must be obtained with some model that preferably incorporates cable dynamics.

If used as a static receiver, the system can be employed to record any type of sources. When towed, the flow noise and vibrations make this system practical only for the recording of high-level sources, such as seismic pulses or piling blows. The system shown in Figure 2.31 is a Passive Acoustic Monitoring (PAM) system manufactured by Seiche for real-time monitoring and recording of acoustic data. The system comprises a PC with data acquisition software, a computer interface, an ADC unit, GPS and AIS devices for vessel positioning, and battery. The hydrophone cable is divided into two independent sections: the *deck cable*, which remains onboard and is used to cover the distance between operator's room and the deck; and the *hydrophone array*, which includes a set of one to three hydrophone pairs for acquisition at different frequency ranges. The system is intended for PAM operations focused on the detection and classification of marine mammal calls, but has also been successfully used as a platform for the recording of air gun pulses during seismic operations.

Figure 2.31 Towed hydrophone system from Seiche Ltd. The *data acquisition unit* with the screen and buffer box in show on the top right. The black *deck* cable is used to cover the distance between the operations room and the deck. The yellow cable is the *hydrophone array*.

Moored ARU

An *autonomous recording unit* (ARU) is a standalone static data acquisition unit installed at a short distance from the seabed with a mooring system. This type of platform provides the highest quality measurements when installed in mid-depth waters with weak currents, due to the absence of flow noise and the protection from the action of waves. Energy efficiency is at the core of these platforms, designed to record over long periods. This focus on efficiency requires a compromise between the maximum sampling frequency, number of channels and duty cycle that can be used. The deployment and recovery are more complex than in the case of towed hydrophones or drifting buoys.

Commercial models for these systems are widely available. Some examples include the Autonomous Multichannel Acoustic Recorder (AMAR) from JASCO, the Sound Meter SM4M from Wildlife Acoustics, or the Orca unit from RS AQUA.

Figure 2.32 Autonomous Multichannel Acoustic Recorders AMAR G4 (from www.jasco.com)

Unmanned Systems

The recent developments in autonomous underwater and surface vehicles (AUV, ASV) have opened up a new range of possibilities for the study and monitoring of the oceans. A comprehensive review of their use for the recording of the underwater soundscape is given by Verfuss et al. (2019b; 2016), where I participated as co-author.

This type of platform has all the benefits of a vessel-towed hydrophone array system, with the additional advantages of its quiet operation, improved mission safety, ability to enter areas that are less accessible to vessels, and reduced operational costs. Its use is not only limited to acoustic recordings and PAM, as it can also play an important role in the monitoring of exclusion zones and marine mammal population density studies. These platforms can be equipped with a variety of sensors that make them suitable for an even larger range of applications.

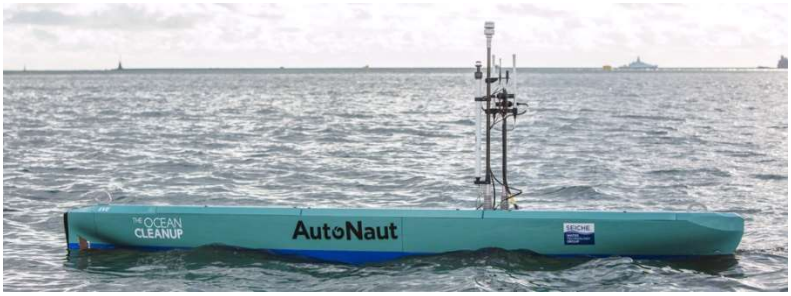

Figure 2.33 Low noise, wave-propelled autonomous surface vessel from AutoNaut.

2.7 Underwater Acoustic Modelling

2.7.1 Air Gun Models

Among all the sources that contribute to the underwater soundscape, impact pile driving and seismic air gun arrays are the ones that have received the most attention. This interest may be justified by the fact that these are the two most frequent and powerful acoustic sources used at sea, and that their outputs are controlled by simple underlying physics. In this subsection we describe the air gun models that are currently available. Pile driving models, despite their importance in marine mammal noise impact assessment, are out of the scope of this thesis.

There are, to the knowledge of the author, only five air gun array models available: Gundalf™ from Oakwood Computing Associates[4], Nucleus+ from PGS[5], CMST Air Gun Array Model (CAGAM) from the Centre of Marine Science and Technology, Air Gun Array Source Model (AASM) from JASCO, and AGORA from University of Leiden.

Gundalf™ is the standard for air gun array modelling in seismic exploration and has been around for many years. The software was developed by some of the scientists directly involved in the advancement of the theory of air gun arrays (Robert Laws, Les Hatton, and Matthijs Haartsen). The software offers the highest standard and reliability supported by the years devoted to the improvement of its physics and recurrent calibrations. A description of its underlying physics, assumptions and limitations can be found in its website. This is a server-based commercial model, and as it could be expected for an industry standard, its license is costly.

Nucleus+ is another commercial software developed by PGS (Goertz et al., 2013). Like Gundalf™, Nucleus+ have been developed over two decades and is another industry standard for seismic modelling for the oil and gas industry. No detailed information is available about its physics, assumptions and limitations.

CAGAM and AASM are models developed for exclusive use of the CMST and JASCO. The modelling approach and assumptions are described in Duncan & Gavrilov (2019) and MacGillivrai (2019; 2006). AGORA was developed by Sertlek & Ainslie (2015) and is the only freely-available software.

Gundalf™, CAGAM, AASM and AGORA were compared during an air gun modelling workshop held in the 4th International Conference on the Effects of Noise on Aquatic Life, in Dublin (Ainslie et al., 2016). The workshop was organised to assess the performance of these four models for three air gun array configurations in a specific propagation medium. Large differences between the models were observed at frequencies above

[4] https://www.gundalf.com/

[5] https://www.pgs.com/marine-acquisition/design-a-survey/nucleus/

1 kHz, explained by differences in the modelling of the *initial rise time*. This outcome highlights the importance of the initial conditions and physical processes involved in the generation of the primary pulse. The validation of all models was identified as a priority.

Except for AGORA, the rest of the air gun models are calibrated against reference source signatures (Ainslie et al., 2016). Gundalf™ uses a number of datasets of near-field recorded signatures to calibrate its parameters for different configurations of single air guns and clusters. CAGAM is calibrated against Gundalf™ (v. 8.1k) for the total energy of the acoustic signature. AASM uses multiple single air gun measurements from the Svein Vaage dataset for calibration (Mattson et al., 2012).

AGORA is said to be calibrated for single air gun measurements from the Svein Vaage dataset (Ainslie et al., 2019) but this is not exactly true. AGORA directly applies the calibration parameters from MacGillivrai (2006), an early version of the AASM model that at the time employed for its calibration a collection of single air gun signatures from a DREP technical report (Racca & Scrimger, 1986). Suggesting that AGORA is calibrated with the same data as AASM is misleading, and instead the authors should have stated that the software used the "same calibration parameters" as MacGillivrai's air gun array model. In conclusion, AGORA is not an explicitly calibrated source model and this reflects on its performance, as shown in the signature comparisons included in the workshop paper.

2.7.2 Sound Propagation Models

The large-scale submarine warfare that started during the Second World War motivated the development of underwater sound propagation theories which would result in an advantage over the enemy. Since then, ocean acoustic propagation models have been used in many applications including sonar, acoustic communications, acoustic tomography, marine mammal detection and oceanography (Wang et al, 2014).

Propagation models are classified into seven main groups according to the method they use to solve the wave equation: ray tracing, normal mode, parabolic equation, wave-number integration or fast field, multipath expansion, energy flux and finite elements method. The models are either based on ray-tracing, a form of numerical integration of the wave equation, or a mixture of both. The latter are aimed at minimising the weaknesses of the other two approaches and are referred as *hybrid models* (Wang et al, 2014; Etter, 2013).

The domain of applicability varies between propagation models, and is defined in terms of frequency, water depth, and range dependence. Outside their range of applicability these models may either not work accurately or require intense computation. Figure 2.34 presents the cases for which the five most common propagation models are suitable or have limited or none applicability.

Model type	Applications							
	Shallow water				Deep water			
	Low frequency		High frequency		Low frequency		High frequency	
	RI	RD	RI	RD	RI	RD	RI	RD
Ray theory	○	○	◑	●	◑	◑	●	●
Normal mode	●	◑	●	◑	●	◑	◑	○
Multipath expansion	○	○	◑	◑	◑	◑	●	◑
Fast field	●	◑	●	◑	●	◑	◑	◑
Parabolic equation	◑	●	○	○	◑	●	◑	◑

Figure 2.34 Context of applicability of the five most common ocean acoustic propagation models. Shallow waters are generally considered below 200 m. Black circles represent a modelling approach that is physically applicable and computationally feasible. Two-coloured circles indicate limitations in accuracy or speed. White circles indicate that the model is neither applicable nor practical. RD and RI refer to *range-dependent* and *range-independent* environments. The limit of applicability for low frequency and high frequency is 500 Hz (from Etter, 2013)

Normal modes, wavenumber integration and energy flux methods are preferred for range independent environments, since they provide an accurate solution within a reasonable time. For strong range-dependence, the parabolic equation and ray tracing methods are more appropriate. Normal mode and parabolic equation methods perform better at low frequencies. For high frequency simulations, ray tracing and energy flux are commonly used. For an efficient calculation of the pressure time-series, the last two methods are also preferred. Normal mode, wavenumber integration and parabolic equation models are solved in the frequency domain, which means that in order to generate a solution in the time domain by Inverse Fourier Transform (IFFT) a large number of frequencies must be modelled, making the process computationally demanding.

Ray Tracing

The Ray Tracing (RT) method solves the wave equation in the high frequency domain. It is assumed that the acoustic pressure can be expressed as a phasor of the form $p = p_o e^{j\varphi}$, where p_o is the pressure magnitude and φ the phase. The solution generates two formulas when applied to the wave equation: the *eikonal-equation*, a non-linear partial differential equation used to determine the ray path; and the *transport equation*, used to determine the amplitude p_o. The physical implication of the geometrical acoustic approximation is that the change in amplitude, sound speed, and direction of the ray over a wavelength has to be small (Wang et al, 2014). The ray tracing solution is limited to frequencies above $f_{ray} \approx 10c/h_w$ and is more suitable for deep waters (Etter, 2013). Here, c is the speed of sound and h_w the depth of the water column.

The ray paths can be traced with the ray tracing method. The sound levels are calculated by effectively increasing the width of each ray (e.g., using Gaussian beam approximation) and then summing the contribution of all rays near the receiver.

The theory can be applied to range-dependent environments and can handle horizontally stratified waters and complicated bathymetries. The directivity of the source can be easily integrated by weighting the amplitude of the rays launched at different angles. The ray-tracing simulations are also computationally efficient.

The ray interaction with the seafloor is generally described by a reflection factor. The sound waves that penetrate the seabed are refracted, attenuated, and a portion of the energy returns to the water column. The acoustic propagation within the sea bottom is not modelled, but the spatial offset of the rays that return after being transmitted into the seafloor can be included in the model by applying the so-called *beam displacement* method (Etter, 2013). A reflection factor can also be defined for the sea surface to include the effect of roughness in the attenuation of the specular reflection.

The main disadvantages of the ray tracing method include (1) the inability to handle diffraction, (2) unrealistic representation of caustics, due to pressure singularities that occur in the convergence regions, (3) drop in accuracy below ~200 Hz where diffraction and seafloor penetration dominate, (4) the need for a large number of rays in shallow waters, and (5) poor performance in surface ducts (Wang et al, 2014).

Normal Modes

The Normal Mode (NM) method solves the wave equation by separating it in vertical components or *standing waves*, and horizontal *travelling components*. The vertical solutions are the *eigenvalues* or normal modes. The horizontal solutions can be neglected in range-independent environments. In range-dependent environments the *adiabatic mode* or *coupled mode* theories can be applied, with the first one being only suitable for moderate range dependence. The normal mode solution is better suited to low frequencies ($f < 1$ kHz), as the number of modes increases linearly with frequency.

Only a limited number of modes can propagate. Those that cannot propagate, called *leaky modes*, may be caused by steep normal incidence angles lower than the critical angle, which are typical at short ranges, and by an increase in attenuation with distance in elastic seabeds. Models that calculate *propagating modes* exclusively are faster, but their accuracy is compromised in the near range, where leaky modes abound; models that search for all modes are more accurate, but relatively slow for long range simulations.

The NM method includes *dispersion effects*, that is, it accounts for the frequency dependence of the phase velocity. One disadvantage of the NM solution is that the computational effort increases with the number of layers and complexity of the geo-acoustic model (Etter, 2013)

Parabolic Equation

In general, the Parabolic Equation (PE) method neglects the backward-propagating sound, and computes the sound field by solving the equation from the source step-by-step up to the receiver. Two-way PE solutions are also available, but are less common. There are two types of PE methods according to the split-step solution adopted, based on the Fast Fourier Transform (Tappert, 1977) and the Padé expansion (Collins, 1993). The PE is derived for the frequency domain, but time domain solutions are also available (Collins, 1988). The method is suitable for range dependent environments, and is equally used in shallow and deep waters.

The parabolic equation is limited to frequencies below 1 kHz. The step size decreases as frequency increases, and for frequencies higher than 1 kHz the required computational effort makes the use of the PE impractical. The split-step Padé approximation handles wide angles of emission, and with larger apertures (higher number of Padé terms) the computational effort increases. An elastic seabed can also make the simulation computationally demanding. The standard PE also has inherent phase errors that will compromise the accuracy of the solution at long ranges.

Wavenumber Integration

The Wavenumber Integration (WI) or fast-field method provides an exact solution of the wave equation, which inherently includes propagating, leaky, and evanescent modes. The standard model is range-independent. Range-dependent versions have been introduced recently, but these are not publicly available and are computationally intensive. (Etter, 2013; Wang et al, 2014).

Multipath Expansion

The Multipath Expansion (ME) method, sometimes referred as Wentzel-Kramers-Brillouin (WKB) approximation, solves the vertical component of the normal mode equation. The acoustic field integral is expressed in terms of an infinite set of integrals, each associated with a particular ray-path family. The method accounts for first-order diffraction and caustics. Like ray tracing models, the ME method is better suited for deep waters and mid-high frequencies (Etter, 2013).

Energy Flux

The Energy Flux (EF) method is a hybrid solution that lies in between ray tracing and normal modes. The method, first developed by Weston (1968; 1980), divides the channel into four regions: spherical spreading, cylindrical spreading, mode stripping and single mode. The EF method is fast and handles diffraction, but generally is only applicable to simple environments and performs an incoherent computation of the sound field, thus high spatial frequency interference is neglected (Wang et al, 2014).

Finite Difference Method

The acoustic propagation methods discussed above are efficient within their domain of applicability (i.e., frequency region, water depth, range-dependence). The gain in computational speed is achieved at the expense of generality, which is lost through the various assumptions and approximations used for the governing equation. As a result, the models lose their ability to characterise important physical phenomena such as *reverberation* and *scattering*.

The Finite Difference (FD) method is one of several numerical solutions capable of solving complex propagation problems (two-way wave equation, fluid-elastic interaction, range-dependence) based in the *direct discretisation* of the governing equations. The Finite Elements (FE) and Boundary Elements (BE) methods also belong to this group. The FD method is particularly popular in seismic exploration, due to its ability to characterise complex inhomogeneous environments.

The main disadvantage of these methods (FD, FE, BE) is their high computational cost, as they need to discretise the environment with sufficient grid resolution to characterise the spatial and temporal variations of the signal. Their use in underwater acoustics is limited to short-range scattering problems (e.g., scattering from the ice cover), short-range fluid-elastic interactions in geometrically complex seabeds (e.g., nearfield impact pile-driving radiation), hybrid numerical models, and benchmarking.

The FD method discretises the governing differential equation by locally approximating the *differential operators* by *difference operators* (Jensen et al., 2011). Each differential operator in the equation is approximated by a Taylor series expansion applied to the terms in the difference operator. The expression for the approximated governing equation is obtained for each point in the gridded space. Combined with the boundary conditions, this leads to a linear system of equations from which the solution of each point in the grid can be obtained.

The discretisation of the differential operators introduces *discretisation* and *numerical stability* errors, associated with the model's *convergence* and *stability*. These errors are related and depend on the grid size.

Finite Element Method

The Finite Element (FE) method solves the wave equation for space and time by gridding the entire environment with sub-wavelength resolution. The elements in the grid can take different forms according to the number of degrees of freedom (DOF) considered (e.g., pyramid or hexahedron for three-dimensional elements). The connexion between elements in the grid provides a linear system of equations from which a solution can be obtained for each element in terms of its DOF. The main difference with FD is that FE discretises the *physical domain*, while FD discretises the *governing differential equation*.

The large number of grid elements required by FE makes this method only viable for short-range acoustic propagation problems (e.g., in pile driving). The FE method is popular in engineering and physics, due to the availability of efficient finite-element schemes for the wave equation, heat equation, and other equations used in structural mechanics and fluid dynamics (Jensen et al., 2011).

Boundary Elements Method

The Boundary Element (BE) method reformulates the governing differential equation in terms of the boundary conditions. The BE method is used for characterising the radiation of a finite object (closed surface) placed in an infinite volume where is exposed to an acoustic field. The method discretises the *Green's theorem*, which expresses the field in the propagation environments in terms of the *sources in the environment* (volume integral) and the *displacement field and its derivative* on the boundary of the object (surface integral).

The BE method is typically used in combination with a traditional, more efficient model for solving scattering problems. The conventional model addresses the general propagation problem, while the BE method accounts for the scattering produced by a rough sea surface, buried objects, or geometric abnormalities in the seabed or ice canopy. This type of hybrid model is more efficient than the FD and FE methods, as they need to discretise the environment with a grid of sub-wavelength resolution, even across regions where the properties of the environment are constant.

Chapter 3

Sound Field Processing

For the purpose of acoustic field characterisation (AFC), having the right set of tools that can help identify, locate and quantify target sound events based on audio and position data is essential. Compared to the wide availability of open-access and commercial software for underwater sound propagation modelling (Wang et al, 2014; Etter, 2013), the tools used for the generation of sound field maps from survey data are, most frequently, *internal* to the company providing the service and *rarely shared*. Any critical input from a third party is then limited to assessing the suitability of the methodology and likeliness of the results, preventing any improvement on the processing approach itself. This chapter presents the general features and processing modules of the Automated Batch Analysis and Characterisation of Underwater Sounds (ABACUS) software, developed integrally by the author as a versatile, open-access tool to process sound field maps associated with individual acoustic sources. The software is written in MATLAB and, to our knowledge, represents the first fully-documented, open-access tool for AFC.

In this chapter we carry out an in-depth revision of the software with reference to small sections of code and details about inputs, outputs, routines, and their interaction. This approach will provide the reader with all the necessary information to critically assess the software's structure, functionalities, and design decisions. However, we also acknowledge that this might hinder the delivery of a general picture about the software capabilities and potential. To ensure that the extent of the work is reliably presented while providing a clear message, the chapter is organised as follows. The background, motivation and design objectives are presented in Section 3.1. Then, Section 3.2 summarises the general features of ABACUS, in an attempt to outline its main outcomes. The software's structure, execution, and supported data are described in Section 3.3, which lays the ground for a thorough description of the core processing modules, addressed in Section 3.4. A processing example with ABACUS is presented in Section 3.5 to offer an intuitive approach to the software, its execution, features, and possibilities.

The core processing modules are central to the software, but these are by no means the only key elements. The software interacts with input and output *databases* to share information between core modules, and employs custom *libraries* for extra functionalities. Databases and libraries are discussed in the Appendices.

3.1 Introduction

The underwater soundscape is filled with sounds from natural, biological, and human origin. The contribution of anthropogenic noise at sea, in particular, has increased considerably in the last century, with low frequency levels raising at an approximate rate of 3.3 dB per decade within the period 1950-2007 (Frisk, 2012). The noise produced by marine industrial activities, such as shipping, pile driving and seismic operations, have a negative impact on the marine fauna, most of which relies on sound for communication, navigation and foraging. Noise can have a direct physiological impact on the animals, as well as disrupt their daily activities through masking, altering their individual behaviour and causing long-term population effects such as displacement or migration (Southall et al., 2019; Popper et al., 2014; NRC, 2005).

The problem of anthropogenic noise at sea and its impact on marine fauna has attracted the attention of scientists, governments and regulators in recent years, who are making efforts to better understand the issue and put in place appropriate regulations and guidelines. Currently, there exist a number of signed International Agreements and country-specific guidelines (Erbe, 2013). Two relevant examples are the Marine Strategy Framework Directive (MSFD) from the European Parliament (2008) and the technical guidance for assessing the effects of human-made noise on marine mammals (NMFS, 2018). While the International Agreements are a key step for the recognition of the noise as an environmental stressor and present global targets and recommendations to meet them, country-specific guidelines detail the procedures for mitigating the impact of noise for those targets to be reached. All types of mitigation involve some form of control of the acoustic emissions or the exposure of the marine animals to the noise.

Controlling the emissions and their impact on the marine fauna is achieved through careful planning, by selecting the period within which the activity will take place (seasonal area restrictions), the source to be employed (power, layout, source type), the operational details (soft-start, installation method, noise abatement), and the real-time monitoring strategy (safety-zone with MMO/PAM). Some guidelines set generic safety zones for different source types and marine mammal species, but a mitigation strategy is most effective when based on safety zones estimated from sound field measurements. In this sense, having the appropriate tools to process the audio and position data gathered in an AFC survey is essential to obtain the acoustic metrics and receiving locations of the recorded sound events produced by the target source. The processed data can then be used to generate a map of sound levels from which the propagation losses, source output, and safety zones can be estimated. A discussion of the importance of using measurement-based safety zones, along with an analysis of the contributing operational, environmental, and biological factors, is presented in a conference paper (Blondel et al., 2016) and a poster (Jiménez et al., 2016) prepared by the author in collaboration with Dr Philippe Blondel, the thesis' first supervisor.

There are a few basic features that a suitable processing tool must have for the generation of sound field maps. These include the automatic detection of band-limited transient events, full control of the duration of the processing window, fast single and multi-band filtering, support for multiple position formats, synchronisation of audio and position data, computation of navigation parameters (distance, bearing, heading, speed), and the ability to deal with large datasets. To the knowledge of the author, the only free-access software capable of all this (with limitations) is PAMGUARD[6]. The open-source software, funded by the IRFC at its early stages and later by the E&P Sound and Marine Life Joint Industry Program (JIP) from the IOGP, was developed to address the limited capabilities of existing PAM software. As such, PAMGUARD's greatest strength lies on its ability to automatically identify, classify, locate, and analyse cetacean vocalisations. However, the software falls short when it comes to detecting and processing signals of other nature and returning the appropriate acoustic and navigation parameters for each detection. Despite its modular structure, which makes the software extremely versatile, in its current state PAMGUARD is not suitable for sound field map generation. This is due to the following:

- Its detectors cannot be used to trigger the broadband and multi-band processing modules in order to compute the broadband and in-band acoustic metrics for each detection.
- There is no support for seismic P190 files, which contain the position of the airgun array and seismic vessel.
- It does not apply synchronisation of audio and position from source and receiver at the time of each detection.
- The shortest processing window length is limited to 1 s, when typical transient signals have durations of the order of 50-500 ms.
- Navigation parameters such as distance and bearing are not computed for each detection. These are key to the fitting of level-range regression curves to the data, from which propagation losses, source levels, and safety zones can be estimated.

During the period of 2014-2021, the author worked at Seiche Ltd as an underwater acoustics researcher, dedicating most of his work to the study of the sound field produced by airgun arrays and impact pile driving. The lack of any free-access software created specifically for AFC soon became evident. This technology gap motivated the author to develop a set of tools in MATLAB that could process and analyse the audio and position data collected during seismic and offshore windfarm projects. After including a graphical interface, this set of tools became what the company called the SPL Toolbox, a piece of internal software that would soon be used for most AFC and SSV projects and became one of the company's "flagship products". The SPL Toolbox was not designed for recurrent updates and was only conceived as a "wrapper" of a set of

[6] https://www.pamguard.org

useful tools developed by the author over the years. Despite its potential, the SPL Toolbox was only intended for internal use, preventing the scientific community from benefiting and contributing to it. This encouraged the author to develop *from zero* a new software for sound field map generation. The new software improved on the ideas applied to the SPL Toolbox and implemented them into a modular structure more suitable for recurrent updates and for the addition of new features. The new software is called ABACUS and has been developed integrally as part of this PhD thesis.

ABACUS stands for *Automated Batch Analysis and Characterisation of Underwater Sounds*. The software, developed in MATLAB, was designed with four ideas in mind: simplicity, modularity, efficiency, and open-access.

Regarding *simplicity*, ABACUS is coded as an interconnected set of functions, scripts and configuration files, it can be run with a few lines of code through the MATLAB script `runAbacus.m`, and only computes what is necessary to produce the required set of acoustic and navigation metrics for each detection. Unlike the SPL Toolbox, it has no GUI.

In terms of *modularity*, ABACUS is organised into six modules and three main stages (import, detect, process). The software dedicates one module to each relevant group of actions. The corresponding modules for each stage are (1) *import* audio, (2) *import* receiver position, (3) *import* source position, (4) *detect* target events, (5) *process* acoustic metrics, and (6) *process* navigation metrics.

In terms of *efficiency*, ABACUS has been specifically designed to batch-process large audio datasets, eliminating the need to process files individually. The computational speed of MATLAB is limited compared to lower-level languages (e.g., C, Java), but efforts have been made to optimise the code to ensure that the processing times are reasonable.

In terms of *open-access*, the code and user guide from ABACUS are distributed under MIT license[7]. The licence contains the copyright and permission notice and guarantees that the software is free of charge and any person with a copy of the software has unrestricted rights over it, subject to the condition of including the license file. The MIT licence is used for its simplicity and unrestricted quality, as at this stage the code is made available for the immediate benefit of its users and not for the benefit of the software itself. The software and the user guide, including configuration files, templates, sample data and examples, can be downloaded[8] from GitHub. The user guide is a 68-page document that contains all the details about the core modules, libraries, raw data, I/O databases, file formats, and execution.

[7] https://mit-license.org

[8] https://github.com/Guille340/ABACUS ; https://github.com/Guille340/ABACUS_Example

3.2 General Features

Below are listed some of the most relevant features in ABACUS. The list introduces the capabilities of the software before these are discussed in the next sections. The points are organised according to the topic and presented in the following order: general (gen), audio import (aud), source import (sou), receiver import (rec), audio detection (det), audio processing (pro), and navigation processing (nav).

- Modular structure (gen). The software comprises six modules that interact with each other through databases. The modules are: audio import, audio processing, audio detection, receiver import, source import, and navigation processing.
- Multiple libraries (gen). Various libraries have been developed to perform some of the calculations required by the modules, such as reading, resampling and filtering audio, reading position data, or estimating covariance matrices. The routines are organised as independent packages as they can also be applied outside the ABACUS environment.
- Non-graphical user interface (gen). The user interface is based on JSON[9] configuration files. There are three files for general settings and one or more files associated with each module. Templates are available for all configuration files.
- Intuitive batch-processing (gen). Individual channels and target signals from multiple audio files can be automatically processed, and all in one "click". This is particularly useful for the processing of large audio and navigation datasets.
- Inputs and outputs stored in databases (gen). A database is a folder comprising one or more files with information that is relevant for the software and the user. A database may contain input information (Configuration Database), intermediate data (Audio Database, Navigation Database, Detection Database), and results (Acoustic Database). Intermediate databases are essential for accelerating the processing as they contain data that is recurrently used by the software. The configuration and results produced by all six modules on each audio file are stored as an Acoustic Database, a lightweight MATLAB file (.mat) containing all the necessary information for sound field map generation.
- Processing block approach (gen). All the databases are saved in a folder known as a "processing block". This allows the user to easily retrieve the configuration information and results from a processing session. This is important for the traceability of processing decisions and the reproducibility of results.
- All calculations come from audio and position files (gen). The audio and position files collected during an AFC survey are all that is required to calculate the parameters needed to produce the sound field maps.

[9] JavaScript Object Notation is an open-source file format used for data exchange.

- Tables for supplementary information (gen). The software uses tables in comma-separated format (.csv) to store additional information required by some of the modules. The nature and volume of the data makes the CSV10 files more convenient than the JSON files.
- Logging (gen). The events that occur during the software execution are displayed on MATLAB's command window. These include actions and warnings. With the purpose of allowing the user to gauge the duration of a processing step, times are displayed next to each log entry and progress bars are showed for computationally demanding tasks involving recursion or loops.
- Efficient audio reading and resampling (aud). A low-memory solution has been developed to avoid importing an entire audio file for it to be read, filtered and resampled, which may cause memory and speed issues, as sound files are typically of the order of one Gigabyte. With this solution, the audio file is read in small blocks, which are filtered or resampled as required. The individual blocks are then merged by applying overlap-save reconstruction. The functions support WAV11 and RAW12 audio formats.
- Multiple position format supported (rec, sou). The software can read source and receiver position information from GPS13, AIS14 and P19015 files produced by various platforms (PAMGUARD, SeicheSSV16, SM4M17).
- Various transient detection algorithms available (det). There are four detectors to choose from: moving average, slice, constant rate, and Neyman-Pearson (NP). The detectors are suitable for short, band-limited target events from sources such as air guns, pile drivers, sub-bottom profilers, or sparkers. The NP detector is the most sophisticated of all and works with transients and continuous signals.
- Neyman-Pearson Likelihood Ratio Test (det). The detector comprises three algorithms of varying complexity: energy detector, estimator-correlator in white Gaussian noise (WGN), and estimator-correlator in coloured Gaussian noise (CGN). Chapter 4 describes the algorithms in detail.
- Multi-source detection (det). Multiple individual target sources can be detected within each file using different detection and band-limiting settings. The algorithms take advantage of the different dominant frequencies and times of occurrence of the events produced by the different sources to accurately detect them.

[10] *Comma Separated Values* is a file format where individual entries are delimited by comma
[11] *Wave* audio format developed by Microsoft and IBM for storing uncompressed audio data
[12] *Raw* audio format for storing headerless, uncompressed audio data
[13] *Global Positioning System* is a satellite-based navigation service owned by the US government
[14] *Automatic Identification System* is a tracking system for the exchange of vessel information
[15] *P190* is a data exchange format used to store data from a seismic survey
[16] *SeicheSSV* is a legacy software developed by Seiche for recording and processing SSV data
[17] The *SM4M* is an underwater acoustic recorder from Wildlife Acoustics (discontinued product)

- Mirroring detections among channels (det). The position and length of each detection can be replicated from one channel to the rest. This is particularly useful when the time and spectral character of a sound event is similar at the locations of the various receivers (e.g., adjacent hydrophones in an array). Copying the detection windows over from one hydrophone to the rest has two advantages: (1) to speed up the detection process, and (2) to allow comparisons between the audio metrics of the same events recorded in different channels.
- Segmentation for ambient noise analysis (det). For the analysis of ambient noise and continuous soundscapes, audio files are divided into segments of equal duration. The slicing process is fast as it does not require a detection algorithm.
- Pre-detection filtering (det). Bandpass filtering can be applied within the detection module to increase noise rejection and improve detection rates.
- Adjusted processing window (pro). The processing window is fitted to each detection by using its cumulative energy curve.
- Pre-processing conditioning (pro). DC-offset correction and bandpass filtering is applied to the detected events before processing their acoustic metrics. The bandpass filtering enhances the signal-to-noise ratio (SNR) of the broadband metrics.
- Processing broadband and in-band acoustic metrics (pro). The broadband and in-band acoustic metrics are processed for each detection and associated noise segment. The peak, peak-to-peak, RMS, and exposure metrics are computed for the broadband values. Only the last two metrics are used for the band values.
- Pre-filtering zero padding (pro). Solution adopted to minimise the errors associated with the filtering of narrowband signals of short duration. Zero-padding is a well-documented digital signal processing technique employed in spectral analysis for its ability to enhance the spectrum through frequency interpolation. In this thesis, zero-padding is used instead as a tool for improving the accuracy of the output of a band-pass filter applied to short, band-limited signals. The approach is discussed in Appendix B.
- Time offset correction (pro). The timestamp in audio files is typically retrieved from the internal clock of the acquisition unit (henceforth referred to as 'PC time'), which may deviate from the UTC time for various reasons (different time zone, daylight saving time enabled, or precision of the oscillator that generates the clock signal). The timestamp is corrected for synchronising the detections (PC time) with the navigation data (UTC time).
- Calculation of navigation parameters (nav). Receiver and source navigation data are computed for each detection, including "absolute" parameters (latitude, longitude, course, speed) and "relative" parameters (source-to-receiver distance, source-to-receiver bearing, and source directivity angle).

3.3 Software Structure

3.3.1 Flow Chart

Figure 3.1 shows the general flow diagram of ABACUS. The software is built around the interaction of three elements: the raw data (green), the databases (orange), and the computational modules (blue).

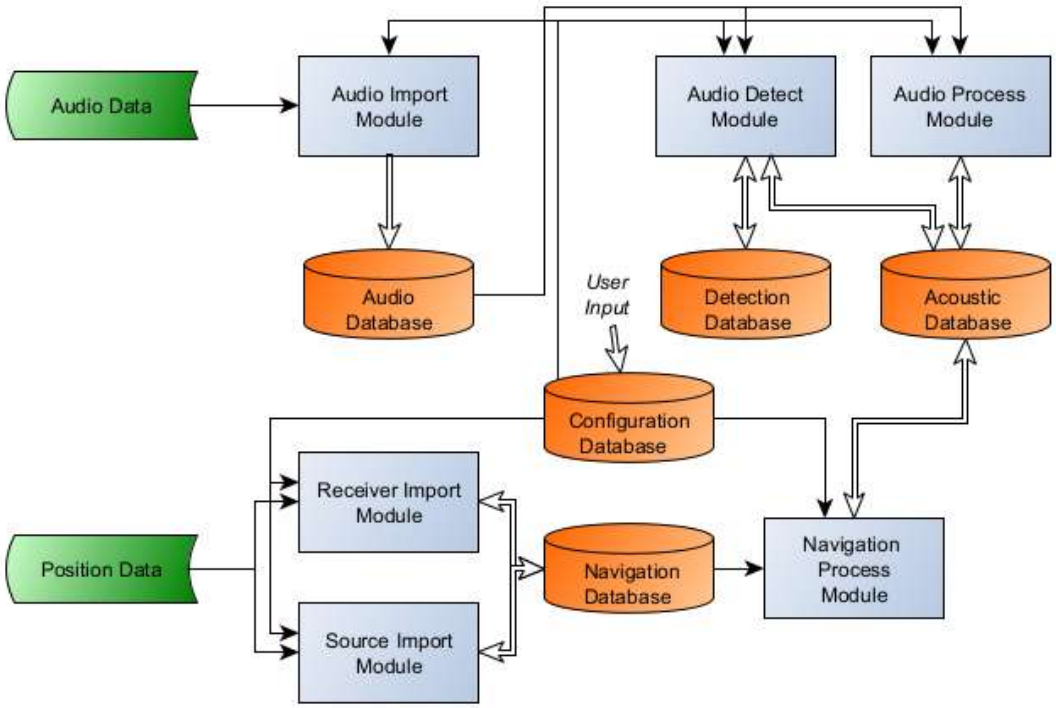

Figure 3.1 Flow chart describing the connections between the data (green), modules (blue) and databases (orange) in ABACUS. Black arrows represent the exchange of information between parts of the software. White arrows represent changes in a database. Double arrows indicate a two-way information exchange.

The process starts with the *raw data*, that is, the audio and position files recorded during an AFC survey. The audio data comprises various sets of multi-channel files, where each set is collected with one data acquisition unit and each channel is linked to one hydrophone. The position data from the sources and receivers is given in the form of positioning files. For further details about the raw data, see Sub-Section 3.3.3.

A graphical interface is not included in ABACUS. Instead, the software employs JSON scripts that the user populates with configuration information. There is one or more JSON configuration templates for each module and three templates for the definition of global settings. The populated configuration files are saved in a Configuration Database folder, accessed by the software to set the parameters of the various modules.

Each module generates a *database* and populates it with configuration and output information. There are five types of databases: configuration, audio, navigation, detection, and acoustic. The audio, navigation, and detection databases contain data that is recurrently used by ABACUS and whose computation is time-consuming. These intermediate databases dramatically increase the efficiency of the software by storing frequently used information in a format that MATLAB can access quickly (.mat). All the databases are stored in a folder known as a *processing block*. For a description about the processing block see Sub-Section 3.3.4. For details about the databases, see Appendix A.

The audio and position data follow two independent processing paths. The *audio data* is first sent through the Audio Import module, which reads and resamples the digital waveform from each selected channel and stores the result in an individual Audio Database file (.mat). Then, the Audio Detect module runs the detection algorithm on each selected Audio Database file to obtain the time limits of each detection, which it stores in an Acoustic Database file (.mat). For the Neyman-Peason algorithm, the module produces files (.mat) containing the performance curves of the detector and information about the target signal and noise (raw scores, covariance, eigendata), all of which are stored in a Detection Database folder. Once the target events and background noise segments have been identified, the Audio Process module locates the detections in each selected Audio Database file and processes their band and broadband acoustic metrics (peak, peak-to-peak, RMS, exposure), which it then saves into an Acoustic Database.

The *position data* of the sources and receivers is processed separately with the Source Import and Receiver Import modules. The modules can read position data (GPS, AIS, P190) from files generated by various supported platforms (PAMGUARD, SeicheSSV, SM4M). For fixed receivers and sources (moored, anchored) a GPS fix can be specified in the configuration files. The raw position information (time, latitude, longitude) is read and saved into a single Navigation Database file (.mat). The Navigation Process module then uses the timestamps of the position sentences (UTC) in the Navigation Database and the time-offset-corrected timestamps of the detections to calculate the "absolute" and "relative" navigation parameters for each detected event. The results are stored in the corresponding Acoustic Databases. By this point, the acoustic and navigation parameters for each detection have been computed, and all the necessary information to generate a map of sound levels for the source is available in the Acoustic Databases. It is not strictly necessary to provide position data, as there will be occasions when only the acoustic parameters are of interest (e.g., ambient noise analysis).

3.3.2 Program Files

There are two folders and two files that are required for running ABACUS: 'Dependencies (Core)', 'Dependencies (External)', `runAbacus.m` and `root.json`. These, along with the LICENSE.txt file, are kept together in the same folder. Of the four, `root.json` is the only file that can be modified by the user.

The folder 'Dependencies (Core)' contains all the MATLAB routines (.m) of ABACUS core modules. The routines are organised in folders, each corresponding to a different module. There are also folders dedicated to configuration templates, configuration examples, and general tools. For a description of the core modules, see Section 3.4.

The folder 'Dependencies (External)' contains a number of MATLAB libraries used by the core modules and organised in individual folders. These packages have been developed by the author as standalone tools to be used in a wide range of applications. Among these libraries, there are solutions for efficient reading, filtering and resampling of large audio files, single-band and multi-band digital and FFT filtering, covariance matrix estimation, reading navigation data (GPS, AIS, P190), and calculating secondary navigation parameters (distances and bearings). Details about the different libraries can be found in Appendix B.

`runAbacus.m` is the file from which the software modules are run. Each section in this file is identified with '%' followed by the title. The sections are organised by order of execution. Note, however, that the processing of the audio and position data follow independent paths and either path can be run first. If the position is not relevant, the position-related commands do not need to be run. Figure 3.2 shows the file content.

The script starts with the 'Add Libraries' and 'Storage Directory' sections. 'Add Libraries' adds the paths of all files in 'Dependencies (Core)' and 'Dependencies (External)' and their subfolders to the MATLAB's search path, so that all routines are available to be run when called. 'Storage Directory' reads the parent directories in `root.json`. The various modules can be executed afterwards by selecting the corresponding lines and pressing F9. Before running any module, the configuration files must be populated.

The `root.json` file is a JSON structure with the fields `"audio"`, `"position"` and `"block"`, representing the parent directories of the audio files, position files, and processing block. ABACUS uses these directories to find the audio, position, and database files. Figure 3.3 shows an example of the content in `root.json`.

A typical MATLAB structure with multiple field-value pairs is easily defined in JSON within curly brackets (*{}*), with the field names are enclosed in quotation marks (*""*), followed by colon (*:*), the value of the field, and a comma for separation. Strings representing paths use double-backslash (`\\`) rather than single-backslash (`\`), a reserved special symbol in JSON.

```
% Add Libraries
addpath(genpath(fullfile(pwd,'Dependencies (Core)')))
addpath(genpath(fullfile(pwd,'Dependencies (External)')))

% Storage Directory
root = jsondecode(fileread(fullfile(pwd,'root.json')));

% Import Audio
AudImpConfig = readAudioImportConfig(root);
audioImportFun(root,AudImpConfig)

% Detect Audio
updateAcousticDatabases(root)
AudDetConfig = readAudioDetectConfig(root);
audioDetectFun(root,AudDetConfig)

% Process Audio
updateAcousticDatabases(root)
AudProConfig = readAudioProcessConfig(root);
audioProcessFun(root,AudProConfig)

% Import Receivers
RecImpConfig = readReceiverImportConfig(root);
receiverImportFun(root,RecImpConfig)

% Import Sources
SouImpConfig = readSourceImportConfig(root);
sourceImportFun(root,SouImpConfig)

% Process Navigation
updateAcousticDatabases(root)
NavProConfig = readNavigationProcessConfig(root);
navigationProcessFun(root,NavProConfig)
```

Figure 3.2 Content of runAbacus.m script.

```
{
"audio": "F:\\PROJECT_XXX\\Survey Data\\Audio",
"position": "F:\\PROJECT_XXX\\Survey Data\\Position",
"block": "C:\\PROJECT_XXX\\Buoy1"
}
```

Figure 3.3 Example of the content in `root.json`. Every time the data or project folder change location, update the paths that accompany the fields “audio”, “position” and “block”.

3.3.3 Audio and Position

Data Acquisition

The purpose of an AFC survey is to measure the sound field, so that the propagation losses in the area, the output of the emitting source, or the safety zones for the marine species of concern can be estimated. During an AFC survey, a sound mapping strategy is adopted to gather a representative sample of audio and position data. A typical

deployment comprises several static (moored) or dynamic (autonomous, drifting) data acquisition units that collect audio and position while the target sound source is active.

Each data acquisition unit is equipped with one or more pre-amplified hydrophones with different gains and one or more devices for recording receiver position (GPS, satellite tracker). The position of the target source is obtained from the navigation system in the source's platform (GPS, AIS, P190). If the source or receiver are static (moored, anchored), a GPS fix is generally enough to determine their position throughout the survey. The output of each hydrophone is sent through a DAQ system that digitises the signal. The DAQ system then communicates with the recording software, which writes the output from each hydrophone into a channel of an audio file. Several multi-channel audio files will typically be created by each recording unit during the course of a survey.

Audio Data

The audio files can be recorded with any data acquisition platform, but there are a few aspects that must be considered to guarantee the quality, compatibility, and completeness of the dataset. We discuss these below.

Every audio file contains as many channels as hydrophones. The maximum bit rate of the DAQ system, the bit resolution, and the maximum frequency of interest determines the number of channels that the data acquisition unit can store. A typical, inexpensive approach for recording a wide range of sound levels consists in using a different preamplifier gain for each hydrophone in a *coincident pair* (i.e., close-by sensors encapsulated together). The goal is to expand the dynamic range of the system, so that the high- and low-pressure levels that will occur during the survey can be registered without risking saturation or low SNR. A dedicated pair of coincident hydrophones is placed at each depth of interest. A common way of preventing local acoustic features (e.g., shadowing effects) from causing *bias* in the audio data is to measure at more than one depth.

The recording system must guarantee a minimum quality for the data. For this purpose, the system must record audio in uncompressed format, with a sampling frequency that covers the dominant spectral region of all signals of interest (twice the maximum frequency of the target signal), and a bit resolution that provides a sufficient dynamic range (96 dB for 16 bit). ABACUS currently supports uncompressed RAW (headerless) and WAV formats with a 16-bit resolution and no restrictions in sampling rate.

To allow synchronisation of audio and position data, the recording software must name each audio file with the PC timestamp of its first sample. The naming format can take any form. Examples of formats are '*yyyymmdd_HHMMSS_FFF' for PAMGuard, 'yyyymmdd_HHMMSS_FFF*' for SeicheSSV[18] or '*yyyymmdd_HHMMSS' for the SM4M[19] system.

[18] www.seiche.com

[19] www.wildlifeacoustics.com

There are two types of error that a recording system may experience when writing into an audio file. The first is a *corrupt header*, which may occur when the writing to an audio file suddenly stops (e.g., power cut). ABACUS is capable of recovering corrupt WAV files using the 'Read Audio' library (see Appendix B). The second type of error is *data loss*, which happens when the recording software is unable to register the entire data flow from the DAQ into the audio file, losing packages in the process. This can have a negative impact on the synchronisation between audio and position (see Sub-Section 3.4.7 for details).

The audio files are stored in the `"audio"` parent directory in `root.json`. The selection of specific audio files for the Audio Import, Audio Detect, Audio Process, and Navigation Process modules is done through the partial paths and folders included in `audioPaths.json` (see Appendix A).

Position Data

A few recommendations and compatibility aspects need to be considered for recording position data used in ABACUS.

ABACUS supports several position file formats with content based on three different positioning protocols: GPS, AIS, and P190. The Global Positioning System (GPS) is a satellite-based navigation system that provides geolocation and time information to a GPS receiver using a combination of four satellites. The AIS and P190 protocols rely on GPS receivers for position data, and provide further information relevant to their particular application. The Automatic Identification System (AIS) is a compulsory positioning system at sea for any vessel above a certain size, and it is employed for traffic information and collision avoidance. A transceiver is used to transmit information of the vessel to which the device is fitted, and to receive information from surrounding vessels. The P190 is an exchange data format specifically designed to store time, position and information relevant to a seismic survey. P190 files are generated by specialised equipment in the seismic vessel.

GPS and AIS sentences[20] use the NMEA protocol, a standard data format for communication between marine electronic equipment. GPS and AIS devices use this protocol, but the format of the position files is ultimately determined by the recording software that reads and decodes the NMEA sentences. ABACUS can read GPS files from SeicheSSV (.gpstext) and PAMGuard (.csv), and AIS files from PAMGuard (.csv). All the navigation and processed data produced by PAMGuard is stored in an SQLite[21] (.sqlite3) database. Direct access to SQLite databases is not supported by the current version of ABACUS, and the GPS and AIS data in the database need to be exported as a .csv table. This is a trivial operation that can be easily performed with any software

[20] *Sentences* are formatted lines of ASCII characters containing navigation information

[21] SQLite is a lightweight, efficient SQL library for storing and managing databases locally. SQL (Structured Query Language) is a programming language for managing *relational* databases

capable of working with SQLite databases, such as SQLiteStudio[22]. The Receiver Import and Source Import modules read the receiver and source position data from supported GPS, AIS, and P190 files (see sub-sections 3.4.5 and 3.4.6).

The Navigation Process module treats the imported position data to obtain a series of navigation parameters for each receiver and source, which it then assigns to each detection in the audio files. The module has three main goals: (1) to correct the spatial offset between each source or receiver and the corresponding positioning device, (2) to "smooth out" the position data, and (3) to synchronise audio and position for each event detected. For details about the Navigation Process module, see Sub-Section 3.4.7.

The position files are stored in the `"position"` parent directory in `root.json`. The selection of specific position files for the Receiver Import and Source Import modules is done through the partial paths and folders in the configuration files `receiverImportConfig*.json` and `sourceImportConfig*.json` (see Appendix A).

3.3.4 Processing Block

A *processing block* is a folder where all the databases for a particular processing scenario are stored. There are five databases in a processing block: Configuration, Audio, Detection, Position, and Acoustic.

The Configuration Database folder stores the populated JSON scripts that define the settings of the various core modules. The Audio, Detection, and Navigation database folders contain large files with pre-processed data created to improve the efficiency of recurrent processing tasks. Acoustic Database files (.mat) are small and contain the results and configuration data from each module and are the primary output of the software. These contain all the information that is needed to analyse, represent and interpret the sound field from target acoustic sources.

A processing block accepts any number of sources and receivers, but it should be limited to a single recording platform. One platform comprises data from one or more sensors (channels) and sound sources (e.g., moored buoy with two hydrophones recording piling blows and ADD pings). Different processing blocks could also be created for specific periods to conveniently process data from long campaigns.

The databases are stored in the `"block"` parent directory in `root.json`, saved in folders 'configdb', 'audiodb', 'detectiondb', 'positiondb', and 'acousticdb'. Except for 'configdb', which is populated by the user, the rest of folders are generated automatically by the relevant modules. Details about each database are given in the Appendix A.

[22] www.sqlitestudio.pl

3.4 Core Modules

3.4.1 General Module Structure

All six core modules are designed around two main functions: `read<MODULE>Config.m`, and `<MODULE>Fun.m`, where `<MODULE>` is the module's name. The first, reads and verifies the content in the module-specific configuration files. The second, communicates with the databases and executes the main routines of the module in accordance with the imported settings. Some modules incorporate additional functions for extra processing or database verification, such as `preProcessNeymanPearson.m` or `updateAcousticDatabases.m` (see Figure 3.2).

The `read<MODULE>Config.m` function reads the module-specific configuration files stored in the 'configdb' folder under the `"block"` directory in `root.json`. A two-step verification process is carried out on each JSON file: first, the individual parameter values are validated and stored in a single-element structure, then the structure is checked for any incompatible combination of parameters or omission of necessary information. If any of the values or their combination are inconsistent, the structure is "flagged" and the appropriate warnings are displayed. The structures generated for each module-related configuration file are combined into a multi-element structure.

`<MODULE>Fun.m` takes the multi-element structure returned by `read<MODULE>Config.m` and runs its specific routines on each of its elements. Any "flagged" element in the configuration structure is ignored. The function interacts with the databases in the `"block"` directory creating new database files and reading and updating their content.

ABACUS employs unique source and receiver names to link the results from all its modules. A relationship between channel and receiver, and sampling rate and source is established through the global configuration files `channelToReceiver.json` and `resampleRateToSource.json`, described in the Appendix A. The `updateAcousticDatabases.m` function, run before the Audio Detect, Audio Process, and Navigation Process modules, verifies that the receiver-channel and the source-sample rate relationships in the Acoustic Databases are consistent with those in `channelToReceiver.json` and `resampleRateToSource.json`. Differences in these relationships may result in the modification or deletion of elements in the Acoustic Databases. Careful planning during the configuration stage will prevent the need for any later modification to the content of the two JSON files.

The `preProcessNeymanPearson.m` function generates a number of files required by the Neyman-Pearson detection algorithm. Further details are given in the Appendix A.

3.4.2 Audio Import

The Audio Import module reads, resamples and stores data from audio files. The idea behind this module is to increase the efficiency of the frequent reading and processing of audio information. This is done in two ways: (1) through the storage of audio data in a format that MATLAB can access faster (.mat), and (2) by using resampling to eliminate redundant information for the specific acoustic source. The MAT audio files, referred to as Audio Database files, are used by the Audio Detect and Audio Process Modules. Figure 3.4 shows the general structure of the Audio Import module.

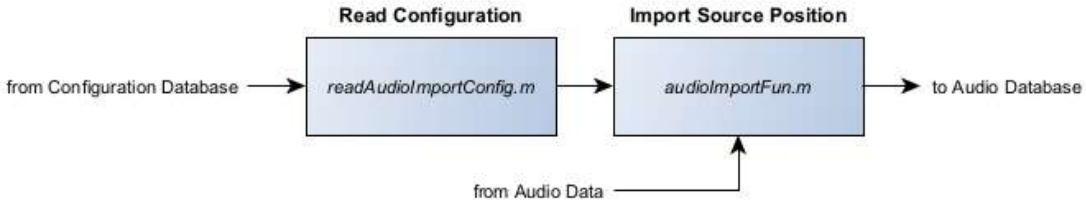

Figure 3.4 Flow diagram of Audio Import module.

The `readAudioImportConfig.m` function reads the content from the `audioImportConfig*.json` files stored in the 'configdb' folder and returns a multi-element configuration structure. The user must create one configuration file for each channel-sampling rate combination to be analysed. For details about the configuration templates, see Appendix A.

`audioImportFun.m` reads and resamples the audio files listed in the global configuration file `audioPaths.json` in accordance with the settings given in each element of the configuration structure. The original audio files are stored in the `"audio"` directory in `root.json`. Combined audio data and configuration information are saved as a MAT file within the Audio Database folder 'audiodb'. One such file is created for each selected audio file, audio channel, and new sampling rate combination. If the MAT file already exists, no reading or resampling takes place. For details about the Audio Database files, see Appendix A.

The module makes use of the 'Read Audio' and 'Resampling Audio' packages. A description of these libraries can be found in Appendix B.

3.4.3 Audio Detect

The Audio Detect module determines the start and end of any event of interest that may be found in an audio file. For efficiency, the function runs on the Audio Database files associated with the original audio files listed in `audioPaths.json`. Figure 3.5 shows the general structure of the Audio Detect module.

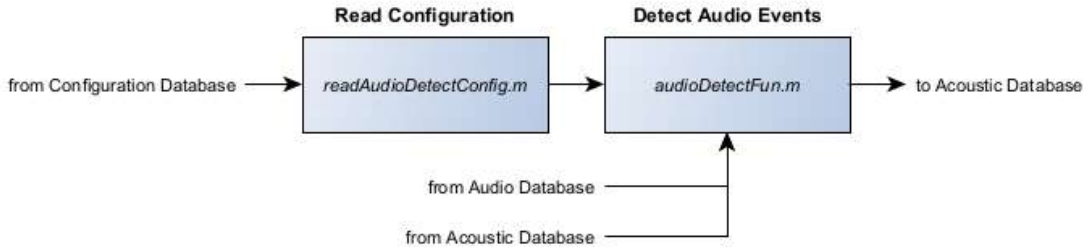

Figure 3.5 Flow diagram of Audio Detect module.

The `readAudioDetectConfig.m` function reads the content from `audioDetectConfig*.json` files stored in the 'configdb' folder and returns a multi-element configuration structure. The user must create one configuration file for each receiver-source combination to be analysed. The module includes five detection algorithms: `'MovingAverage'`, `'Slice'`, `'ConstantRate'`, `'Mirror'`, and `'NeymanPearson'`. The detectors are described in the sub-subsections below. For details about their configuration templates, see Appendix A.

`audioDetectFun.m` first locates the Audio Database files associated with the original audio files listed in the global configuration script `audioPaths.json`. For each selected Audio Database file, the function calculates the time limits of every detection and associated background noise, relative to the start of the audio file. The relative time limits are computed using the detection algorithm and settings given in each element of the configuration structure. For details about the Audio Database files, see Appendix A.

`audioDetectFun.m` creates an Acoustic Database file per listed audio file and stores it in the 'acousticdb' folder. The configuration information and detection results from each receiver-source combination are saved as one element of sub-structures `AudDetConfig` and `AudDetData` in the corresponding Acoustic Database file. If the receiver-source combination already exists in the Acoustic Database file, the old detection data is replaced with the new results and any associated processing data is discarded. For details about the Acoustic Database files, see Appendix A.

The `'MovingAverage'` and `'NeymanPearson'` detectors call the 'FFT Filtering (Single)' package to bandpass-filter each segment before running the detection algorithm. The filter aims at improving the detection performance by limiting the analysis to the dominant frequency region of the target signal. The `'NeymanPearson'` detector also employs the 'Covariance Estimation' package to obtain a reliable estimate of the covariance matrices of the target signal and noise in a *high-dimensional context* (i.e., the number of training observations is comparable to or lower than the number of variables). For details about these two libraries, refer to Appendix B.

The above flow chart changes for the `'NeymanPerson'` detector. In this case, additional calculations need to be carried out with `preProcessNeymanPearson.m` before running `audioDetectFun.m`. The function computes the raw scores and covariance matrices of

the target signal and noise, the eigendata, and the performance curves. This information is stored in the Detect Database folder 'detectiondb' as individual MAT files. The files are later used by the Neyman-Pearson (NP) detector to calculate the *threshold* $\gamma'(\mathbf{x})$ and *test statistic* $T(\mathbf{x})$ for each audio segment. An audio segment is classified as "detection" if $T(\mathbf{x}) > \gamma'(\mathbf{x})$. Figure 3.6 shows the module structure for the `'NeymanPerson'` option. The Detect Database files are described in the Appendix A.

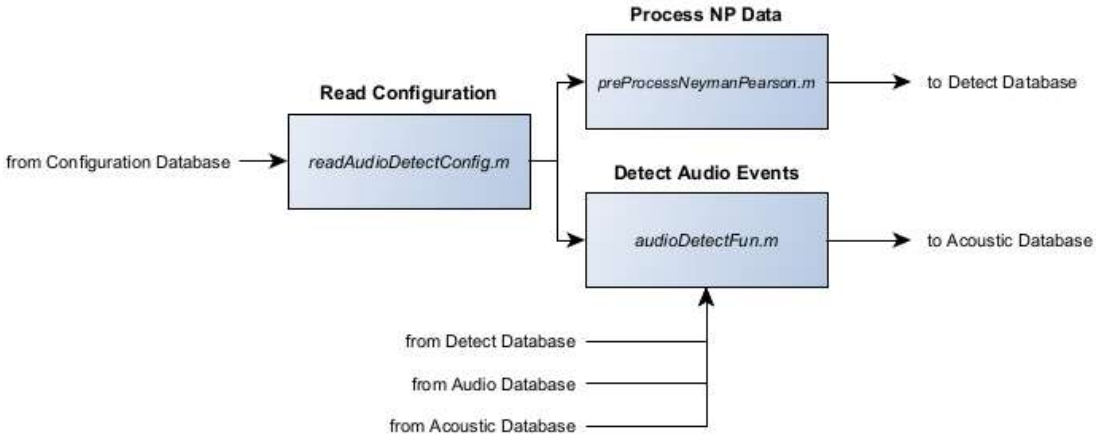

Figure 3.6 Flow diagram of Audio Detect module when the Neyman-Pearson detector is used.

Moving Average

The `'MovingAverage'` detector divides the audio file into segments of fixed duration and classifies as "detection" any segment with a root-mean-square (RMS) amplitude that is larger than the RMS amplitude of the previous segment multiplied by a given factor. The detector also applies a band-pass filter before processing the RMS value to maximise the signal-to-noise ratio (SNR). The windows that delimit the detections are then centred, shifting their start position to a specific number of seconds before the global energy peak. The detector also returns the limits for the background noise windows, placed immediately before each detection window.

Constant Rate

The `'ConstantRate'` detector uses an external 'Pulse' table to locate the sound events. The table contains the relative time of the first pulse and the pulse interval for a number of audio files (see the Appendix A). This solution is suitable when the firing rate of the source is virtually constant (e.g., sub-bottom profiler). The solution requires the user to extract the relative time and pulse interval from each file that needs to be analysed. This approach has been proved useful when a conventional detector is unable to identify the events from a low-energy engineering source with a steady firing rate. The detector returns the limits of the target events and associated noise segments.

Slice

The `'Slice'` detector is not a "detector" strictly speaking, as it does not need to access the audio data to determine the position of the target events. The algorithm simply divides the audio file into segments of fixed duration. This solution is suitable for the analysis of continuous sounds and soundscapes where there is no unique or dominant target source (e.g., ambient sound). The algorithm does not return the limits of background noise windows, since all segments are treated as target sound.

Mirror

The `'Mirror'` solution copies the detection information from a specified receiver into the current receiver. This is a convenient approach for comparing the detections from two or more nearby sensors, as it ensures that the number of detections and their time location is identical. The approach also contributes to improving the overall efficiency of the detection process, as it does not require a detection algorithm to run for all the sensors. The channel to be "mirrored" should be the one with the highest SNR, as the detections computed from that channel will be the most reliable.

Neyman-Pearson

The `'NeymanPearson'` detector divides the audio file into segments of fixed duration and applies one of three available Neyman-Pearson algorithms to each segment. The individual detections are then grouped into larger windows. Using shorter detection windows with subsequent "grouping" is advantageous for the detection of target events of varying duration (e.g., source measured at different distances). The three available algorithms are the energy detector (`'ed'`), the estimator-correlator in white Gaussian noise (`'ecw'`), and the estimator-correlator in coloured Gaussian noise (`'ecc'`).

For the analysis of each segment, all three methods use a target probability of false alarm α to determine the threshold $\gamma'(\mathbf{x})$ from the probability function of false alarm $P_{FA}(\gamma')$ and the variance of the signal and noise. A lower limit to the SNR can be defined to avoid classifying as "detection" any segment with a SNR below that limit. This is a useful parameter, as segments with a SNR < -10 dB are generally of no interest due to the large uncertainty of their broadband and in-band energy metrics. The SNR of the segments can be improved through *resampling* or *filtering*. Neither type of processing affects the Audio Database files, as these are only applied temporarily to enhance the performance of the detector.

The estimator-correlators rely on *training data*[23] to build the covariance matrices that will be used for the computation of the performance curves $P_{FA}(\gamma')$ and test statistics

[23] The training data is a collection of *representative* and *high-quality* observations of the target acoustic event. The *observations* are equal-length audio clips used to build the covariance matrices of the target signal and noise, employed by the Neyman-Pearson estimator-correlators.

$T(\mathbf{x})$. The training data for the target signal and noise is stored in a specified directory, under sub-folders 'signal' and 'noise'. A covariance estimator is used to compute the covariance matrix of the target signal (`'ecw'`), or the covariance matrices of the target signal and noise (`'ecc'`). There are nine covariance estimators available: `'oas'`, `'rblw'`, `'param1'`, `'param2'`, `'corr'`, `'diag'`, `'stock'`, `'looc'`, and `'sample'`. Note that no training data is required for the energy detector to compute the test statistic $T(\mathbf{x})$ and detection threshold $\gamma'(\mathbf{x})$ for each segment, and the number of variables (samples) and the variance of the signal and noise are sufficient.

3.4.4 Audio Process

The Audio Process module computes the broadband and in-band acoustic metrics of the events detected by the Audio Detect module. It also calculates the offset-corrected timestamps (UTC) of those events for them to be later synced with the navigation data. Figure 3.7 shows the general structure of the Audio Process module.

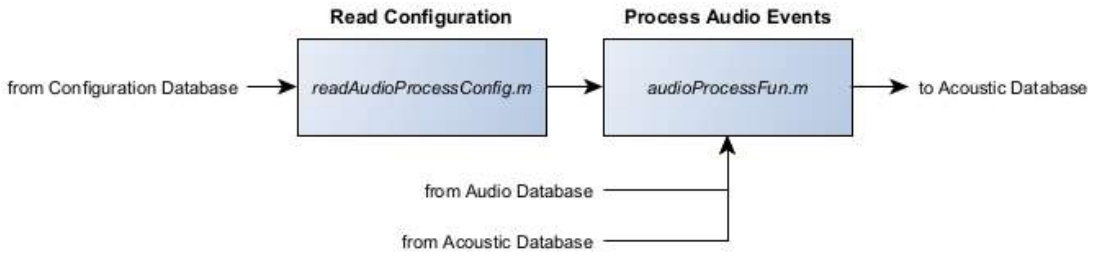

Figure 3.7 Flow diagram of Audio Process module.

The `readAudioProcessConfig.m` function reads the content from the `audioProcessConfig*.json` files stored in the 'configdb' folder and returns a multi-element configuration structure. The user must create one configuration file for each receiver-source combination to be analysed. For details about the configuration templates, see Appendix A.

`audioProcessFun.m` first locates the Audio Database and Acoustic Database files associated with the original audio files listed in the global configuration script `audioPaths.json`. For each selected Audio Database file, the function calculates the offset-corrected timestamps and acoustic metrics of every detection and background noise segments found within the corresponding Acoustic Database file. The processing is done in accordance with the settings given by each element in the configuration structure. For details about the Audio Database files, see Appendix A.

The configuration information and processed results from each receiver-source combination are saved as one element of the sub-structures `AudProConfig` and `AudProData` found in the Acoustic Database file. If the receiver-source combination already exists in the Acoustic Database file, the old processed data is replaced with the new results. For details about the Acoustic Database files, see the Appendix A.

The Audio Process module calls the 'Digital Filtering (Single)' and 'FFT Filtering (Bank)' packages to limit the bandwidth of the audio segment before processing its acoustic metrics, and to compute the acoustic metrics per frequency band. The single digital filter aims at improving the accuracy of the broadband metrics by eliminating any energy contribution outside the dominant frequency region of the target signal. For details about these two libraries, refer to Appendix B.

The correction of the offset between the PC and UTC times is of key importance when the acoustic data needs to be represented in space, as is required for acoustic field characterisation. Errors in the timestamp of the detections will affect the location of the source and receivers, distorting the measured sound field.

It is common for an offset to exist between the PC time of the audio files and the UTC time of the position sentences. The offset may be caused by the settings on the operative system of the PC (time zone, daylight saving time), or the hardware's internal clock. A clock drift of the order of 10 µs/s is usual in commercial oscillators (Melvin & Murphy, 2002). When the clock of a recording system has not been set to UTC before measurements take place, having a way to determine the time offset is key. PAMGUARD and SeicheSSV record the PC and UTC times with each position sentence. The user can use this information to create a 'Time Offset' table (`timeOffset.csv`), a standard file used by ABACUS to compensate for any time offset. The file contains the difference between PC and UTC times at specific moments. For details about this table, see Appendix A.

A less desirable situation is the loss of audio data. *Data loss* happens when the recording software is unable to register the entire data flow from the DAQ into the audio file, losing packages in the process. This is typically the result of a *bottleneck* effect on the audio stream caused by some element in the hardware-software chain. Some level of data loss can always be expected but it only occurs sporadically and involves small data packages. When this is not the case, the quality of the data will be compromised. In this situation it is particularly important that the software incorporates ways to deal with the data loss, such as inserting "filling" samples during bottleneck periods. The use of "filling" samples will ensure that each sample's timestamp is accurately calculated based on the timestamp of the first sample in the audio file, the sampling rate, and the number of samples. When the software fails to do so, the synchronisation between audio and position becomes inaccurate, causing an "spatial compression" of the sound field.

3.4.5 Receiver Import

The Receiver Import module reads time and position data from one or more receivers, and populates a Navigation Database file with that information. The position data may be introduced manually or read from files with a supported format. Figure 3.8 shows the general structure of the Receiver Import module.

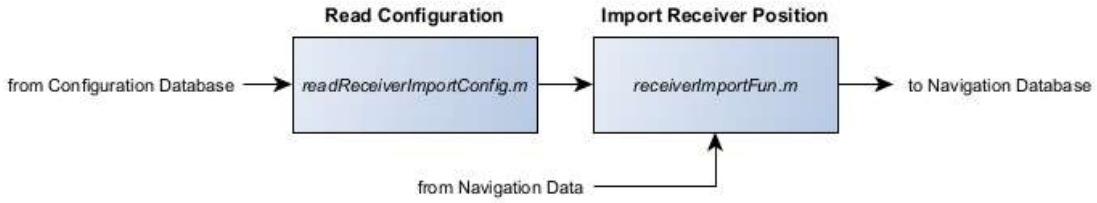

Figure 3.8 Flow diagram of Receiver Import module.

The Navigation Database is an intermediate file introduced with two goals in mind: (1) to improve the efficiency of the recurrent reading of position information by creating a central database written in a format that the software can access quickly (*.mat), and (2) to compile the position information for it to be tested and verified before further processing takes place. The file is stored under the 'navigationdb' folder in the `"block"` directory. For details about the Navigation Database, see Appendix A.

The `readReceiverImportConfig.m` function reads the content from the `receiverImportConfig*.json` files stored in the 'configdb' folder and returns a multi-element configuration structure. The user must create one configuration file for each receiver name. Each sensor in a data acquisition platform represents a receiver. There are two receiver categories: static (`'fixed'`) and dynamic (`'towed'`). These two receiver types are described in the sub-sections below. There is one dedicated configuration template for each category. For details about the configuration templates, see Appendix A.

`receiverImportFun.m` imports the time and position data for each receiver, as indicated by the corresponding element in the configuration structure. The position data may consist of a *single fix*, read directly from the configuration structure, or of *multiple time-position* entries, read from the specified files stored in the `"position"` directory.

The function then populates (or creates if it doesn't exist) the Navigation Database file. The configuration information and the time, position, and metadata from each receiver are saved in the corresponding element of the sub-structures `RecImpConfig` and `RecImpData` in the Navigation Database file. If the name of the receiver already exists in the Navigation Database file, the position data is not imported. An existing receiver can be deleted with function `deleteReceivers.m`.

The Receiver Import module calls the 'Read GPS', 'Read AIS', and 'Read P190' packages to import time and position data from navigation files. The libraries currently support four navigation file formats: GPS (.gpstext), GPS (.csv), AIS (.csv), and P190 (.p190). The .gpstext and .csv extensions refer to the software platforms PAMGUARD and SeicheSSV, as described in Sub-Section 3.3.3 ('Position Data'). For details about the three libraries, refer to Appendix B.

Fixed

The `'fixed'` category refers to a receiver that is static. The position of such receiver can be determined through a single *location fix*[24] (`"latitude"`, `"longitude"`). No time needs to be given, as the position is assumed constant for the period covered by the processing block. Examples of fixed receiver are a moored recording buoy or an anchored recording vessel.

Towed

The `'towed'` category refers to a receiver attached to a moving platform. The time and position of the receiver is determined from positioning files (GPS, AIS, P190). The receiver may be towed or attached to the carrying platform. If a distance exists between the position data and the receiver location, the latter can be estimated by introducing an offset in the horizontal plane and by specifying the type of connection to the platform (`'soft'` for a cable, or `'hard'` for a solid attachment). Examples of `'towed'` receiver are a vessel-towed hydrophone array or a drift recording buoy.

3.4.6 Source Import

The Source Import module reads time and position data from one or more sources, and populates a Navigation Database file with that information. The position data may be introduced manually (static source) or read from supported files. Figure 3.9 shows the general structure of the Source Import module.

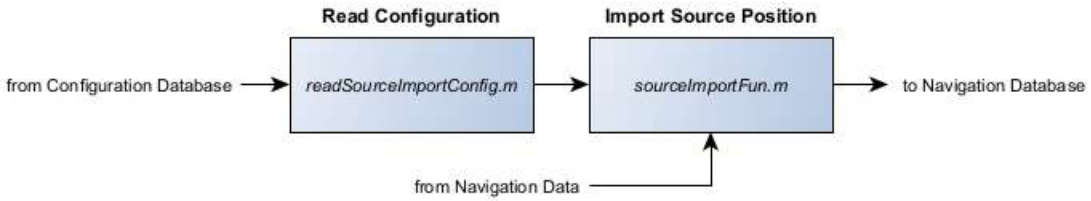

Figure 3.9 Flow diagram of 'Source Import' module.

The `readSourceImportConfig.m` function reads the content from the `sourceImportConfig*.json` files stored in the 'configdb' folder and returns a multi-element configuration structure. The user must create one configuration file for each source name. There are four source categories: static (`'fixed'`), dynamic (`'towed'`), single ship (`'vessel'`), or multiple ships (`'fleet'`). The four source types are described in the sub-sections below. There is one dedicated configuration template for each source category. For details about the configuration templates, see Appendix A.

[24] A point defined by a latitude-longitude coordinate

`sourceImportFun.m` imports the time and position data for each source, in accordance with the corresponding element in the configuration structure. The position data may consist of a *single fix*, read directly from the configuration structure, or of *multiple time-position* entries, read from the specified files in the `"position"` directory.

The function then populates (or creates if it does not exist) the Navigation Database file. The configuration information and the time, position, and metadata from each source are saved in the corresponding element of the sub-structures `SouImpConfig` and `SouImpData` in the Navigation Database. For the `'fleet'` category, the configuration and results are saved in the Navigation Database sub-structures `VesImpConfig` and `VesImpData`. If the source name already exists in the Navigation Database file, the position data is not imported. A source can be deleted with function `deleteSources.m`. Similarly, the `'fleet'` category can only be imported once. In order to read new multi-vessel data, the entire fleet category must be removed from the Navigation Database using the function `deleteVessels.m`.

Like the Receiver Import module, the Source Import module calls the 'Read GPS', 'Read AIS', and 'Read P190' packages to import time and position data from navigation files. For details about the three libraries, refer to Appendix B.

Fixed

The `'fixed'` category refers to a source that remains static. The position of this type of source can be determined through a single location fix (`"latitude"`, `"longitude"`). No time input is required with it. Examples of fixed sources are drills and impact piles.

Towed

The `'towed'` category refers to a source attached to a moving platform. The time and position of the source are determined from positioning files (GPS, AIS, P190). The source may be towed or attached to the moving platform. If a distance exists between the position data and the source location, the latter can be estimated by introducing a horizontal offset and by specifying the type of connection to the platform (`'soft'` for a cable, or `'hard'` for a solid attachment). Examples of `'towed'` source are a vessel-towed seismic airgun array or a hull-mounted sonar.

Vessel

The `'vessel'` category refers to an individual vessel acting as target source. The time and position of the vessel are determined from positioning files (GPS, AIS, P190). This category is best suited for ship noise analysis. It accepts vessel metadata such as length, beam, draft and gross tonnage. Unlike the `'towed'` category, no horizontal offset can be applied, as the vessel itself represents the source.

Fleet

The `'fleet'` category refers to a group of vessels. The time and position of each vessel are determined from AIS positioning files. Like the `'vessel'` category, no horizontal offset can be applied.

Vessels in this category are considered *secondary sources*. As opposed to a target source, a secondary source cannot be selected for audio processing, but its navigation data will be processed and synchronised with the detections from all of the target sources. Importing the position data from a vessel fleet is useful in two ways: (1) it provides an image of the activity in the area at key moments, and (2) it helps assess the proximity of individual vessels to target sources and their masking potential.

The module accepts a 'Vessel Database' file (`vesseldb.csv`) containing the MMSI[25] number of specific vessels, their name, and dimension parameters. The table provides additional vessel information that may not be always included in the AIS sentences. This information can help gain a clear view about the operations and the noise contribution from individual vessels. For details about `vesseldb.csv`, refer to Appendix A.

3.4.7 Navigation Process

The Navigation Process module computes the navigation parameters of the receiver, target source and secondary sources at the time of the events detected by the Audio Detect module. The processed navigation parameters comprise *absolute-values* for the source and the receiver (latitude, longitude, speed and course) and *relative-values* between source and receiver (range, bearing, source's directivity angle). Figure 3.10 shows the general structure of the Navigation Process module.

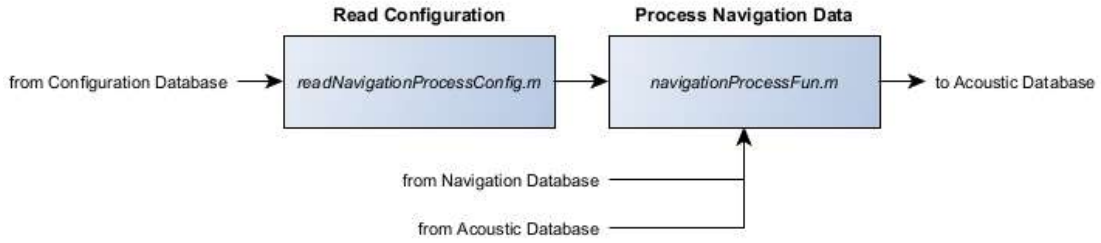

Figure 3.10 Flow diagram of 'Navigation Process' module.

The `readNavigationProcessConfig.m` function reads the content from the `navigationProcessConfig*.json` files stored in the 'configdb' folder and returns a multi-element configuration structure. The user must create one configuration file for each receiver-source combination to be analysed. For details about the configuration template, see Appendix A.

[25] Maritime Mobile Service Identity is a 9-digit unique identifier assigned to a 'maritime object'

`navigationProcessFun.m` first locates the Navigation Database and the Acoustic Database files associated with the original audio files listed in the global configuration script `audioPaths.json`. For each element in the Acoustic Database, the function extracts the position data for the corresponding receiver-source combination from the Navigation Database (for details about the Navigation Database, see Appendix A). The position information of the receiver, source, and "fleet" vessels is treated with a time-averaging window of specified duration. Then, the navigation parameters are calculated at the UTC times of the detections using interpolation. No navigation parameters are calculated for detections falling within a period with no position information longer than a given limit. The smoothing, interpolation method, and maximum position-free period for the receiver-source combination are given by the related element in the configuration structure.

GPS receivers have a high time resolution (one sentence per second) and a moderate horizontal resolution (~10 m for a single receiver). The purpose of the "smoothing" window is to reduce the chance of clustered points, which can affect the accuracy of the estimated course and speed. Clusters are likely to occur with slow-moving platforms.

The configuration information and processed navigation data from each receiver-source combination are saved as one element of sub-structures `"NavProConfig"` and `"NavProData"` in the corresponding Acoustic Database file. If the receiver-source combination already exists in the Acoustic Database file, the old processed navigation data is replaced with the new results. For details about the Acoustic Database files, see Appendix A.

3.5 ABACUS Example

This section presents a short processing example carried out in ABACUS. The intention is to show, in a practical way, how the software is set up and run. In the process, the main functionalities of the software are tested and the potential uses of its output parameters are discussed.

The section starts with a description of the folder structure and the execution steps, presented in Sub-Section 3.5.1. The data acquisition method and the details about the collected audio and position data are summarised in Sub-Section 3.5.2. Then, in Sub-Section 3.5.3, we introduce the settings selected for the processing and justify their choice. To finalise, the content of the output files from the Audio, Detection, Navigation, and Acoustic databases is presented in the form of graphs in Sub-Section 3.5.4.

3.5.1 Structure and Execution

The example is stored in a remote Git[26] repository and can be downloaded[27] from GitHub. The files are open-access and can be used, modified and distributed under the specifications of the MIT license.

The example includes a 'block' folder containing the inputs and outputs from the software, a 'sample_data' folder containing the audio and position data, the MIT license saved as LICENSE.txt, and a README.txt file summarising the content of the folder. The *hidden* '.git' folder is the Git repository where the code and any "committed" changes[28] made along the project's lifetime are stored and tracked.

The 'block' folder contains the sub-folders 'configdb', 'audiodb', 'detectiondb', and 'navigationdb'. 'configdb' is the Configuration Database where the populated tables and JSON configuration scripts (global and module-specific) are stored. 'audiodb' is the Audio Database folder where the audio data extracted from the selected audio files are saved as individual MAT files. 'detectiondb' is the Detection Database folder containing the performance file and ROC[29] curves from a Neyman-Pearson detector. 'navigationdb' is the Navigation Database folder which contains the position data from the sources and receivers, saved in a `navigationdb.mat` file.

The 'sample_data' folder has two sub-folders: 'Data_Audio' and 'Data_Position'. The first includes three uncompressed audio files saved in RAW and WAV formats. The second contains one depth file and four position files with AIS (.csv), GPS (.csv, .gpstext) and seismic (.p190) data.

To run the example, ABACUS must be first downloaded. The paths in the `root.json` file need to be updated manually to point at the location of the audio data (`*\\sample_data\\Data_Audio`), the position data (`*\\sample_data\\Data_Position`), and the processing block (`*\\block`). The different modules are run with the `runAbacus.m` script. Both `root.json` and `runAbacus.m` files can be found under the software's root directory.

3.5.2 Audio and Position Data

The sample audio and position data in the 'sample_data' folder is part of a dataset recorded by the author during a sound source verification (SSV) project that took place in a shallow-water, tidal-bay area in Southcentral Alaska during the ice-melting season in 2014. The sound field mapping was carried out in the context of an ocean-bottom

[26] *Git* is a distributed version control system for tracking changes to files

[27] https://github.com/Guille340/ABACUS_Example.git

[28] "Commit" is the action of recording a snapshot of the current project into the Git repository

[29] A *Receiver Operating Characteristic* graph plots the probability of detection versus false alarm

node (OBN) marine seismic survey. Measurements were gathered for the verification of a pre-survey sound field model and its associated marine mammal safety zones.

Three airgun arrays were measured at two locations before the OBN survey started. The sound field mapping strategy consisted in the use of six unmoored data acquisition buoys deployed upstream from locations separated by 1 km and arranged in a horizontal line. The buoys were deployed from a small passenger vessel and allowed to drift downstream while gathering audio and position data. A wide range of angles and distances were covered as the buoys moved past the source.

Each data acquisition buoy was equipped with: (1) a *localisation system* comprising a strobe light, radar reflector, and satellite tracker, (2) the *electronics box* containing the battery, GPS unit, and PC with the recording software (SeicheSSV), (3) the *float* to provide the floatation, stability and support for the entire data acquisition system, and (4) the *wet leg*, a two-metre metal cylinder designed to protect the hydrophone array. Two pre-amplified hydrophones were placed at the end of the array, each with a different gain to extend the dynamic range of the system and, in that way, minimise the chance of saturation or low signal-to-noise ratio.

The data in the 'sample_data' folder belongs to measurements taken over two days during the operation of a 1760 in^3 seismic airgun array from a location with a water depth of 50 m. The airgun array emitted at an approximate rate of one pulse every 25-29 s. The buoy used for the recordings was equipped with two coincident hydrophones (2 cm separation) with nominal sensitivities of -194.6 and -182.1 dB re 1V/µPa and a nearly-flat response between 10 Hz and 5 kHz. The least sensitive hydrophone was stored in the left channel of the audio files.

The 'Data_Audio' folder includes three non-consecutive audio files of 150 s duration, saved in RAW and WAV formats, each containing six airgun pulses. There is a time gap between consecutive audio files of approximately 50 minutes. The 50-minute separation was selected to ensure that different sound levels and source-receiver distances were represented. The data was recorded at 16-bit with a sampling rate of 48 kHz.

From all the files in the 'Data_Position' folder, only the GPS (.gpstext) and seismic (.p190) files are used. The AIS (.csv), GPS (.csv) and depth (.csv) files are only included for reference, as examples of tables extracted from a PAMGUARD's sqlite3 database.

3.5.3 Configuration

General

There are three general configuration files in the 'configdb' folder: `audioPaths.json`, `channelToReceiver.json`, and `resampleRateToSource.json`. These files can affect multiple modules.

The paths of the six audio files, relative to the "audio" directory in root.json, are listed in the audioPaths.json script. In this example, only the RAW audio files are used. The WAV audio files are ignored by placing an exclamation mark "!" at the start of their relative paths, as shown in Figure 3.11.

```
[
"RAW\\20140406_231556_3304312_7200000.raw2int16",
"RAW\\20140407_000639_8163269_7200000.raw2int16",
"RAW\\20140407_005723_3022226_7200000.raw2int16",
"!WAV\\20140406_231556_3304312_7200000.wav",
"!WAV\\20140407_000639_8163269_7200000.wav",
"!WAV\\20140407_005723_3022226_7200000.wav"
]
```

Figure 3.11 General configuration script audioPaths.json.

One source and two receivers are considered: the 1760 in^3 airgun array ('Airgun'), and the buoy's low-sensitivity and high-sensitivity hydrophones ('Buoy_H1', 'Buoy_H2'). The receiver names are linked to their respective channels (1, 2) through the channelToReceiver.json script, and the source name is linked to the sampling rate of the resampled audio files (24000 Hz) through the resampleRateToSource.json script (see Figure 3.12 and Figure 3.13). Note that different sampling rates may be used for different sources. The audio data, originally sampled at 48 kHz, is decimated by 2.

```
{
"channel": [1,2],
"receiverName": ["Buoy1_H1","Buoy1_H2"]
}
```

Figure 3.12 General configuration script channelToReceiver.json.

```
{
"resampleRate": [ 24000],
"sourceName":   ["Airgun"]
}
```

Figure 3.13 General configuration script resampleRateToSource.json.

Audio Import

There are two configuration files available for this module: audioImportConfig_RAW_01.json for channel 1, and audioImportConfig_RAW_02.json for channel 2. Both scripts contain instructions to read and resample (decimate by 2) the RAW audio files in the "audio" directory. The scripts are identical except for the "channel".

```
{
"audioFormat": "RAW",
"channel": 1,
"resampleRate": 24000,
"sampleRate": 48000,
"bitDepth": 16,
"numChannels": 2,
"endianness": "l"
}
```

Figure 3.14 Configuration script `audioImportConfig_RAW_01.json`.

Audio Detect

There are six configuration files available for this module, two per detector: `audioDetectConfig_NeymanPearson_01.json` and `audioDetectConfig_NeymanPearson_02.json`, `audioDetectConfig_MovingAverage_01.json`, `audioDetectConfig_MovingAverage_02.json`, `audioDetectConfig_ConstantRate_01.json`, and `audioDetectConfig_ConstantRate_02.json`. The scripts ending in `01` and `02` contain the detection instructions for the receivers `'Buoy1_H1'` and `'Buoy1_H2'`.

The example can be run on any detector by simply moving the related scripts from the 'detectors' sub-folder to the 'configdb' folder. Note that 'configdb' can only contain scripts from one type of detector. The settings for the three detectors are summarised below. The results in Sub-Section 3.5.4 correspond to the Neyman-Pearson detector.

The *Neyman Pearson* detector divides the audio file into segments of 0.5 s duration. The energy detection algorithm is then applied to each of those segments. For the purpose of improving the SNR, the audio data is resampled at 2 kHz. Segments with a SNR lower than -6 dB are ignored. The algorithm uses the pre-computed performance curves and a target probability of false alarm of $\mathbf{10^{-4}}$ to calculate the detection threshold $\boldsymbol{\gamma'(\mathbf{x})}$ and the test statistic $\boldsymbol{T(\mathbf{x})}$ for each segment $\mathbf{x}$. A segment is classified as "detection" if $\boldsymbol{T(\mathbf{x}) > \gamma'(\mathbf{x})}$. The individual detections are grouped into windows of a minimum duration of 2 s, which represent the final "detected" segments. The configuration file is shown in Figure 3.15.

The *moving average* detector uses two consecutive windows of 3 s duration. The segments are first processed with a lowpass filter with a 1 kHz cutoff frequency. The segment within the leading window is classified as "detection" if its RMS amplitude is at least 1.3 times the RMS amplitude of the segment from the trailing window. The detection window is then shifted to start 0.7 s before the waveform's peak. The configuration file is shown in Figure 3.16.

The *constant rate* detector employs a 3 s detection window placed at constant intervals as specified in the `pulseTable.csv` file. The 'Pulse' table provides the start of the first

pulse and the pulse repetition interval for each audio file to be processed for detections. The configuration file is shown in Figure 3.17.

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"detector": "NeymanPearson",
"DetectParameters": {
       "detectorType": "ed",
       "kernelDuration": 0.5,
       "windowDuration": 2,
       "windowOffset": [],
       "rtpFalseAlarm": 1e-4,
       "detectorSensitivity": 1,
       "minSnrLevel": -6,
       "cutoffFreqs": [0,1000],
       "trainFolder": "",
       "estimator": "",
       "resampleRate": 2000
       }
}
```

Figure 3.15 Configuration script `audioDetectConfig_NeymanPearson_01.json`.

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"detector": "MovingAverage",
"DetectParameters": {
       "windowDuration": 3,
       "windowOffset": 0.7,
       "threshold": 1.3,
       "cutoffFreqs": [0,1000]}
}
```

Figure 3.16 Configuration script `audioDetectConfig_MovingAverage_01.json`.

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"detector": "ConstantRate",
"DetectParameters": {
      "windowDuration": 3,
      "windowOffset": 1.5,
      "fileName": "pulseTable.csv"}
}
```

Figure 3.17 Configuration script `audioDetectConfig_ConstantRate_01.json`.

Audio Process

There are two configuration files for this module: `audioProcessConfig_01.json` for receiver `'Buoy1_H1'` and `audioProcessConfig_02.json` for receiver `'Buoy1_H2'`.

The broadband and third-octave band metrics are processed within the frequency range of 20 Hz to 12 kHz for each detection. The true limits of a detection are determined by the central 90% energy interval of its cumulative energy curve. The detection timestamps are corrected by applying the offsets in table `timeOffset.csv`.

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"freqLimits": [20, 12000],
"bandsPerOctave": 3,
"cumEnergyRatio": 0.9,
"audioTimeFormat": "yyyymmdd_HHMMSS_FFF*",
"timeOffset": [],
"tags": ["full","rampup"]
}
```

Figure 3.18 Configuration script `audioProcessConfig_01.json`.

Receiver Import

The module has two configuration files: `receiverImportConfig_Towed_01.json` for `'Buoy1_H1'` and `receiverImportConfig_Towed_02.json` for `'Buoy1_H2'`.

The position of the receivers is extracted from the file `"GPS (SeicheSsv).gpstext"` stored in the `"position"` folder. The hydrophones are placed at 2 m below the water surface and share horizontal position with the data acquisition buoy.

```
{
"receiverCategory": "towed",
"receiverName": "Buoy1_H1",
"receiverOffset": [0,0],
"receiverOffsetMode": "soft",
"positionPaths": "GPS (SeicheSsv).gpstext",
"positionFormat": "GPS",
"positionPlatform": "SeicheSsv",
"vesselId": [],
"mmsi": [],
"depth": -2
}
```

Figure 3.19 Configuration script `receiverImportConfig_Towed_01.json`.

Source Import

The module includes one configuration file, `sourceImportConfig_Towed_01.json`.

The central position of the airgun array is extracted from the `"P190 (Seismic).p190"` file stored in the `"position"` folder by providing the identifier of the vessel (`1`) and the source (`1`). The airguns in the array emit from a depth of 2 m below water surface.

```
{
"sourceCategory": "towed",
"sourceName": "Airgun",
"sourceOffset": [0,0],
"sourceOffsetMode": "hard",
"positionPaths": "P190 (Seismic).p190",
"positionFormat": "P190",
"positionPlatform": "Seismic",
"vesselId": 1,
"sourceId": 1,
"mmsi": [],
"depth": -2
}
```

Figure 3.20 Configuration script `sourceImportConfig_Towed_01.json`.

Navigation Process

The module includes two configuration files: `navigationProcessConfig_01.json` for `'Buoy1_H1'` and `navigationProcessConfig_02.json` for `'Buoy1_H2'`.

The Navigation Process module computes the “absolute” navigation parameters of the receivers and the source (position, course, speed) and the “relative” navigation

parameters between source and receivers (range, bearing, directivity angle) at the time of each detection. The position data is "smoothed" with a 10 s window. A nearest-neighbour interpolation is used to estimate the position of the source and the receiver given the detection timestamp. No restrictions are effectively imposed on how large a gap between position sentences may be for interpolation to be allowed.

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"smoothWindow": 10,
"maxTimeGap": 86400,
"interpMethod": "nearest"
}
```

Figure 3.21 Configuration script `navigationProcessConfig_01.json`.

3.5.4 Output Data

In this sub-section the output data from the various software modules is represented and discussed. The goal is to demonstrate the application of this processed information and its use for the characterisation of the sound field and the estimation of the source level, propagation losses, and safety zones.

Audio Database

Each MAT file in the Audio Database folder contains the resampled audio data from one channel of a selected audio file, along with the audio settings and contextual information, such as the name of the audio file and its length in samples. Figure 3.22 shows the waveform of channels 1 (top) and 2 (bottom) for the three non-consecutive audio files, separated by vertical lines. The represented timestamps correspond to the first sample of audio files extracted directly from their file names.

The audio data is stored as floating-point samples. In this case, the samples are saved as 16-bit signed integer values (`int16`). The samples can take values between -32768 and 32767 and are represented as "U" units. The digital waveform can be converted to pressure if the voltage resolution of the audio interface and the sensitivity response of the data acquisition system are known (Merchant et al., 2015; Robinson et al., 2014).

Note that each audio file contains six airgun pulses with lower amplitudes for the later recordings. The audio data was purposefully selected to cover a reasonable range of sound levels and distances from the source, so that the information could be used to obtain a reliable estimate of the propagation loss, source level and safety zones (see sub-subsection "Acoustic Database").

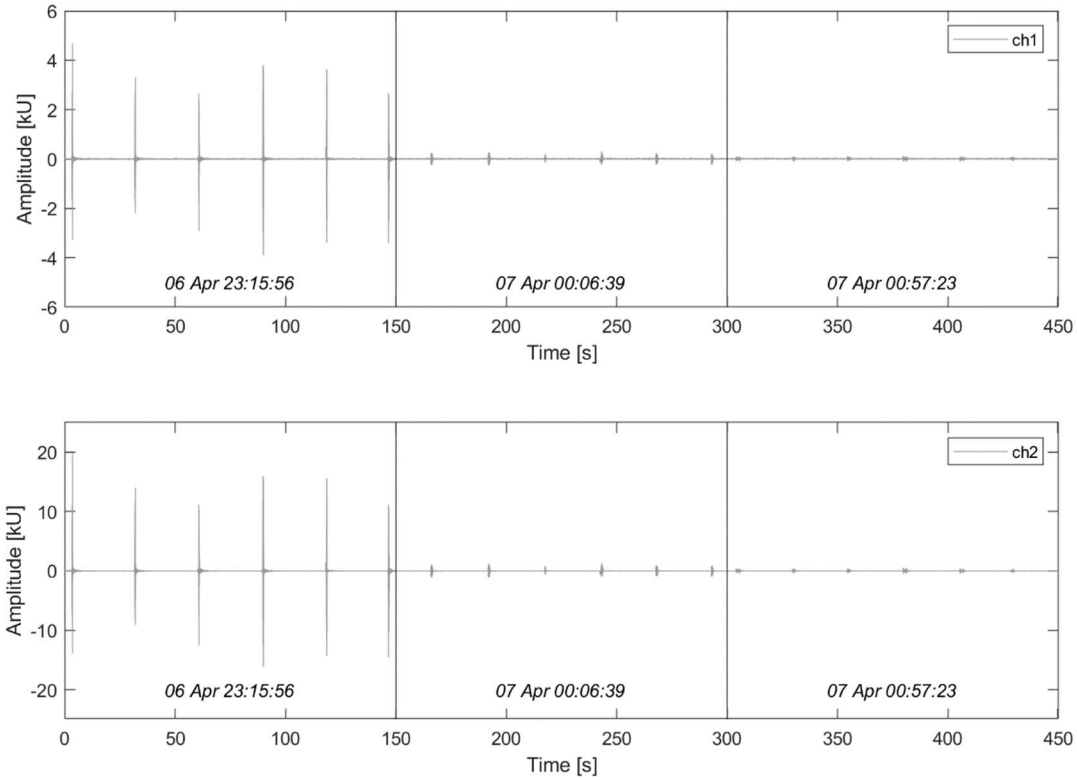

Figure 3.22 Digital waveform of the three unique Audio Database files in the 'audiodb' folder for channels 1 (top) and 2 (bottom), in kU. The vertical lines indicate a change between files.

Navigation Database

The MAT Navigation Database file stores the raw positioning data (date, time and geographic location) of the receivers, sources, and vessels, along with the configuration information used to read the original GPS, AIS or P190 files.

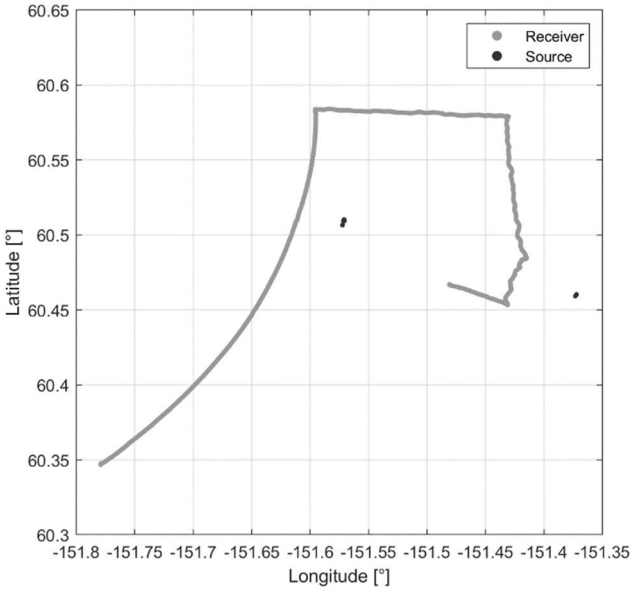

Figure 3.23 Geographic position of the source (airgun) and the recording platform (buoy1).

In this example, the file contains the positioning data for the `'Airgun'` and receivers `'Buoy1_H1'` and `'Buoy1_H2'`. Since the two receivers belong to the same hydrophone array, their position information is identical. Figure 3.23 shows the position of the source (black) and the buoy along its trip from the vessel until its recovery (grey).

Detection Database

The Detection Database folder is only generated when a Neyman-Pearson detector is selected. The 'detectiondb' folder contains one or more of the following files: the matrix of raw scores, the covariance matrix, the eigen data, and the performance data. The example uses the *energy detector* algorithm, which unlike the estimator-correlators, only generates a performance data file (.mat). The *performance data* file is a multi-element structure where each element corresponds to one of 101 SNR levels between -50 and 50 dB. Every element in the structure contains the probability density function for the null and alternative hypotheses, the probability functions of false alarm and detection, the corresponding values of the test statistic (representing the x-axis), and a number of contextual parameters such as type of NP algorithm, SNR level and number of variables.

These probability curves are needed for the calculation of the detection threshold $\gamma'(\mathbf{x})$, obtained for each audio segment $\mathbf{x}$ based on a target probability of false alarm. But the probability curves, represented in Figure 3.24, can also be used for assessing the performance of a detector. For example, one can use the curves to make an informed decision about the target probability of false alarm that will give the best balance between *missed* and *false alarm* detections within each context. The receiver operating characteristic (ROC) shown in Figure 3.25 offers all the relevant information for this purpose, as it can help identify a target probability of false alarm that results in an acceptable detection probability for a range of expected SNR levels.

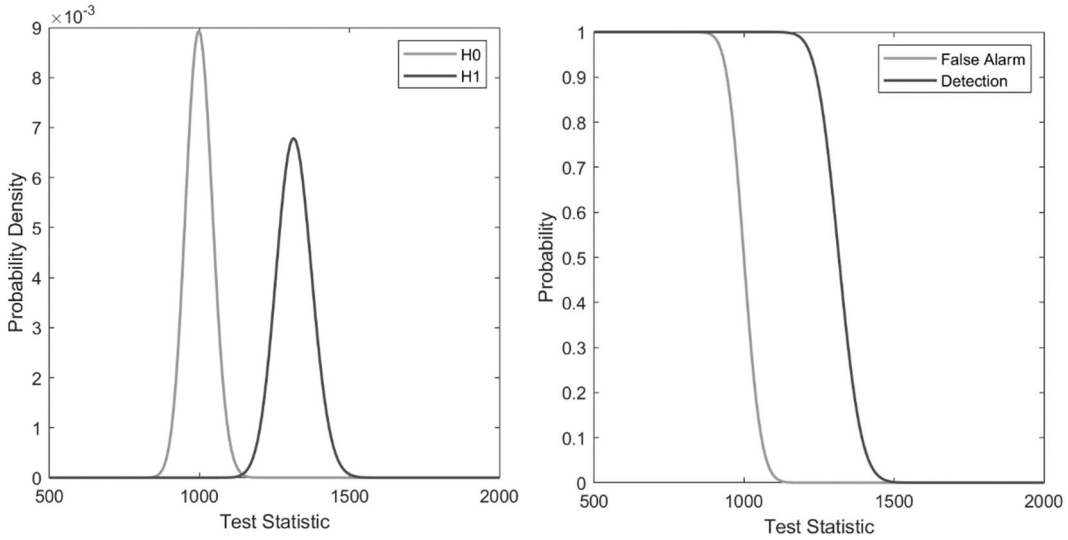

Figure 3.24 Probability density functions for the null and alternative hypotheses (left) and probability functions of false alarm and detection (right), for an energy detector, a SNR of -5 dB and a reference noise variance $\sigma_w^2 = 1$.

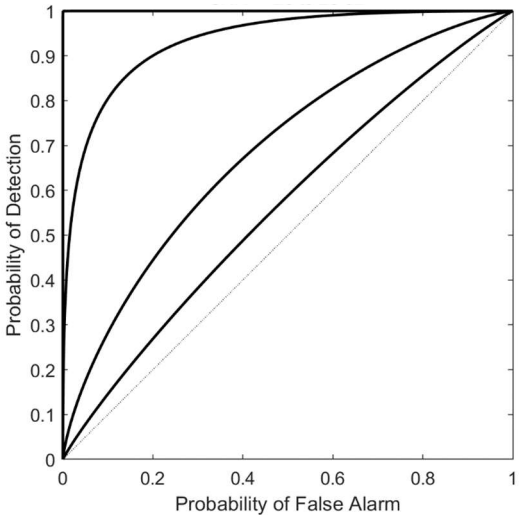

Figure 3.25 Receiver operating characteristic (ROC) of an energy detector. From closest to furthest from the diagonal line, the curves correspond to SNR levels of -20, -15, -10 and -5 dB.

Acoustic Database

While the Audio, Detection, and Navigation databases are used to store intermediate information used by ABACUS across its various processing steps, the Acoustic Database contains the final results. These results can be used for the characterisation and analysis of the sound field and the recording activities. Below we show a few examples.

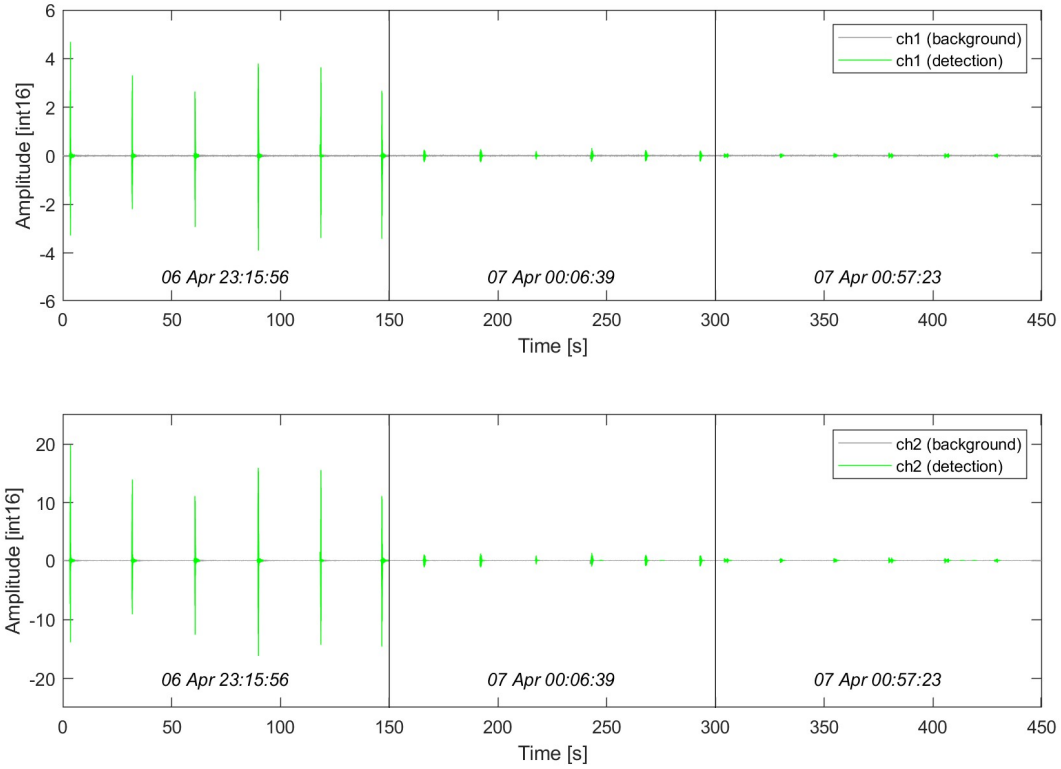

Figure 3.26 Digital waveform of the three unique Audio Database files in the 'audiodb' folder for channels 1 (top) and 2 (bottom), in kU. The green sections represent the detections.

The limits of each detection, determined by the Audio Detect module and stored in the Acoustic Database files, can be displayed over the audio waveform. This is a straightforward way of assessing the validity of the settings used in the detection algorithm. Figure 3.26 represents the waveform of the three processed audio files, with detections from the Neyman-Pearson energy detector shown in green. The results prove the suitability of the energy detector for the detection of acoustic events of acceptable signal-to-noise ratio (> -6 dB) over stationary background noise.

The Navigation Process module computes the "absolute" navigation parameters of the receivers and source (i.e., position, speed, and course) for each detected event and stores the results in Acoustic Database files. This information can be displayed to evaluate the spatial coverage of the recorded target events that will be used to produce the map of sound pressure levels. The speed and course of the receiver, source or surrounding vessels is not only useful for the general analysis of the activities in the area, but it can also provide an insight into potential sound masking (e.g., a vessel passing a receiver at high speed, or a hydrophone being towed from a fast-moving platform). In Figure 3.27, the left image represents the transit paths of the buoy (grey) and the airgun array (black), with their locations at detection instants shown as blue and red. The right image is a close-up view of the track followed by the buoy during the first audio file. The length and orientation of the purple vectors represent the speed and course of the receiver.

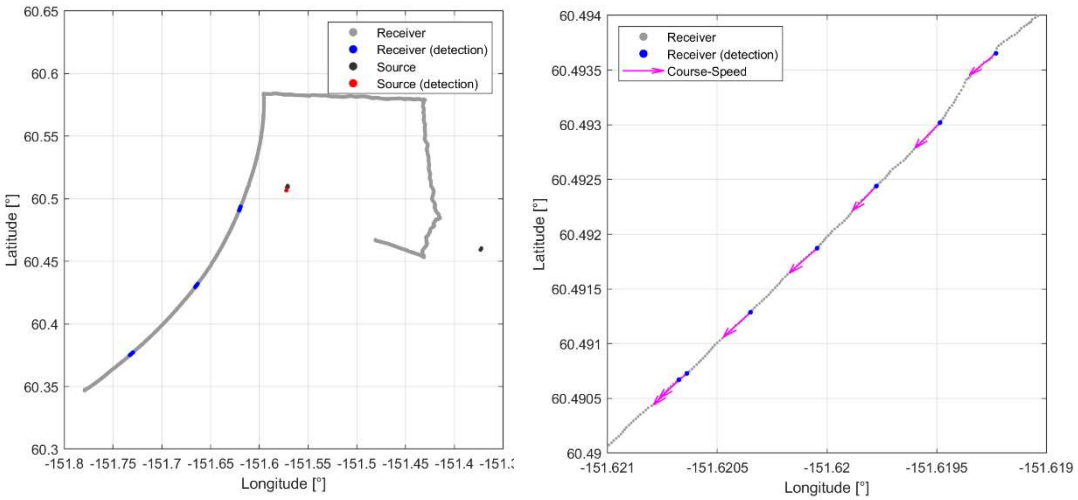

Figure 3.27 Geographic position of the recording buoy and the airgun array on the night of the 6th April 2014, with the corresponding positions at the detection times shown in blue and red (left). Close-up view of the topmost cluster of receiver locations at the detection times, with the speed and course represented by purple vectors (right).

The 'Navigation Process' module also computes "relative" parameters between the source and the receivers. These parameters are the *range* or source-receiver distance, the *bearing* or source-receiver angle, and the source's *directivity angle* or angle between the bearing and the source's heading. These parameters can be combined with the acoustic metrics generated by the Audio Process module (RMS, exposure, peak, peak-

to-peak) to produce some form of sound level map based on one or more spatial parameters (receiver's depth, range, and bearing). Most commonly, the receiver depth remains constant and the sound levels are displayed against range (1D plot) or against range and bearing (2D plot).

The first case is shown in Figure 3.28, where the sound pressure levels from all bearings are represented against the source-receiver distance for the two channels. Slight sound level differences between the channels may be expected due differences in the sensitivity response of the hydrophones, the SNR of the recorded signals, and the detection limits which were obtained independently for each channel.

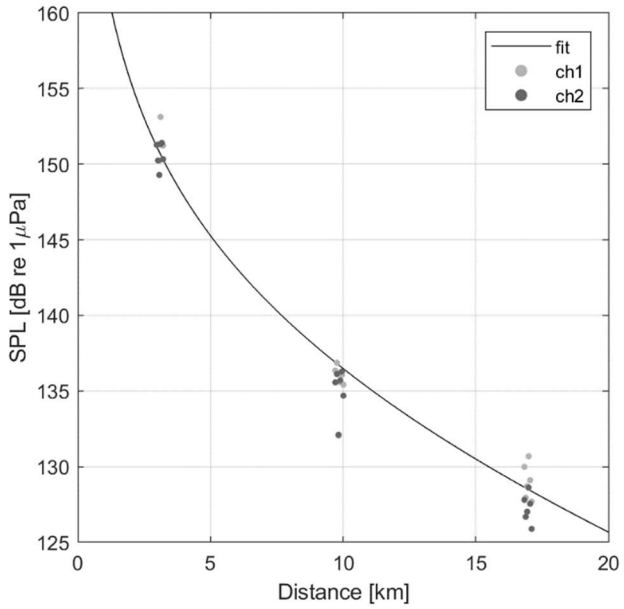

Figure 3.28 Sound pressure level (RMS) as a function of range. The scattered data corresponds to the individual detections, and the continuous line the non-linear regression curve of the form $SL - A\log r - B(r-1)$, where $SL = 230.4$ dB re 1µPa, $A = 22.5$ and $B = 4.1 \cdot 10^{-4}$ dB/m.

This type of representation is of great importance for the estimation of the propagation loss in the area, the source level, and the safety zones for marine species. The simplest way of calculating these parameters is by fitting to the data a regression curve of the form $SL - A\log r - B(r-1)$, where SL is the source level, A is the geometric spreading factor, B is the attenuation factor, and r is the range (Urick, 1983). Such curve is fitted to the data using non-linear optimisation, which results in a source level of 230.4 dB re 1µPa, a geometric spreading factor of 22.5, and an attenuation factor of $4.1 \cdot 10^{-4}$ dB/m. The propagation loss is defined by the last two parameters.

Note that, in this example, the value of the geometric spreading factor is larger than the largest value expected in free field conditions ($A = 20$), corresponding to *spherical spreading*. Larger values of A can be observed in deep waters and at short distances from the source, where the *Lloyd mirror* effect dominates ($A = 40$), but in this example a value of $A = 22.5$ has no physical justification. That being said, it is common practice

within the underwater acoustics community to use unconstrained values of A and B to achieve a better fit to the data (Jiménez-Arranz et al., 2020a). In this sense, large values of A are explained by the fact that both A and B account for the losses associated with *multipath propagation* (i.e., seabed absorption and sea surface scattering).

The safety range can be estimated from a similar graph. This time, the appropriate metrics (RMS or exposure) are treated with the auditory weighting of the marine mammal species of concern and the safety zone is obtained as the range at which the corresponding target level threshold is last exceeded (Southall et al., 2019).The band levels are necessary to analyse and interpret the spectral content of the target signal and background noise, and to process particular regions of the spectrum. The background noise spectrum may be studied separately (e.g., for ambient noise characterisation) or used to evaluate its impact on individual frequency bands of the target signal, as shown in Figure 3.29. It should be noted that the SNR is lower for the low-sensitivity channel (left). This is a typical response of an airgun pulse, characterised by a low-frequency content with the peak energy occurring at a few hundreds of Hertz.

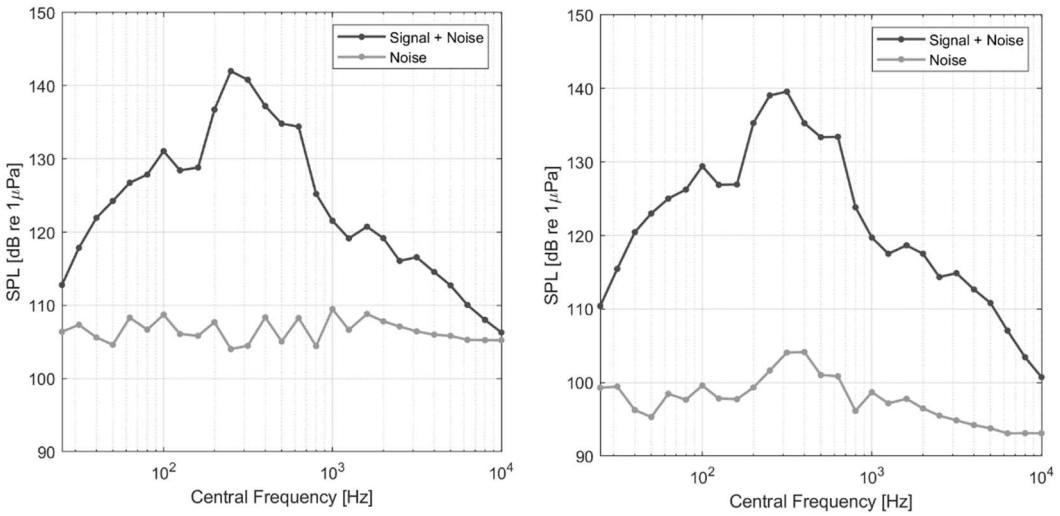

Figure 3.29 Average third-octave band SPL frequency spectrum of pulses emitted by a 1760 in^3 airgun array (black) and of ambient noise (grey), for channels 1 (left) and 2 (right).

The acoustic metrics in the Acoustic Database files are stored as digital values within the range of -32768 and 32767 for audio data recorded at 16-bit. The digital amplitudes of the spectral bands can be converted to sound pressures by multiplying them by the voltage resolution v_{res} of the audio interface (in V) and then dividing the result by the system's sensitivity at the frequency of the band S_f (in V/μPa). If v_{fsr} and n represent the voltage full scale range and the number of bits of the DAQ, and D is the RMS digital value of the band, the corresponding pressure in μPa RMS is calculated as follows

$$p = \frac{D \cdot v_{fsr}}{S_f 2^n} \tag{3.1}$$

Chapter 4

A Neyman-Pearson Algorithm for Detecting Seismic Pulses

This chapter presents the theory, performance, limitations, implementation, and application of three relatively simple but effective statistical detectors: the energy detector, the estimator-correlator in white Gaussian noise (WGN), and the estimator-correlator in "coloured" Gaussian noise (CGN). The algorithms use a Likelihood Ratio Test (LRT) approach and are implemented as part of ABACUS software. Here we introduce the theory of the three statistical detectors in a way that is approachable, to then expand on the details of their performance and implementation. Our final goal is to study their behaviour when applied to the low-frequency transients of seismic air gun pulses, which are of particular interest for this thesis.

The chapter comprises two parts. The first part introduces the problem and theoretical background, whilst the second focuses on the analysis, implementation and application of the Neyman-Pearson (NP) detectors. The *first part* starts with the introduction in Section 4.1, followed by Section 4.2 which presents the nomenclature and general concepts of statistical detection theory. Section 4.3 discusses the various aspects affecting the performance of the detectors, to give further insight into the way they work and to help understand the choices made in their design. The estimator-correlator employs the covariance matrices of the target signal and noise. The difficulty to accurately estimate a covariance matrix when the number of training observations is lower than the number of variables ($M < N$) is an obstacle to overcome when implementing an estimator-correlator. Section 4.4 introduces a few concepts about covariance matrix decomposition that are key to understand some of the formulation introduced in later sections. The most relevant methods available for covariance estimation in high dimensions ($M < N$) are reviewed in Section 4.5. The theory of the three versions of Neyman-Pearson LRT detector for simple hypothesis testing is presented in Sections 4.6, 4.7, and 4.8. The theory follows the formulation from Kay (1998) with added demonstrations, figures, and details to provide a more intuitive understanding of the possibilities of each detector.

The *second part* encompasses the new work carried out to analyse, implement and apply the statistical NP detectors in the context of underwater sounds. In Section 4.9 we

investigate the relation between the test statistic and the detection threshold for various generic target and non-target signals, to reveal the behaviour and limitations of each detector when applied to audio data. In Section 4.10 we propose using resampling, filtering, and window length selection to improve the performance of a NP detector, to then demonstrate how these solutions reshape the probability density functions. In Section 4.11 we use these ideas to derive a general formulation for the energy detector, which extends the application of the original algorithm to filtered observations. In Section 4.12 we propose applying the log-likelihood covariance estimation technique with shrinkage and cross-validation from Theiler (2012) for estimating a high-dimensional covariance matrix using a customised target covariance matrix for filtered data. We then discuss why this is a more suitable approach than some of the most widely used covariance estimators. In Section 4.13 we describe how the three Neyman-Pearson detectors are implemented in ABACUS, putting emphasis on the processing steps and design decisions. In Section 4.14 we analyse the performance of the energy detector and the estimator-correlators on real seismic air gun data.

4.1 Introduction

The ability to automatically decide if an event of interest has taken place in noise-contaminated data is of great importance in areas such as radar, sonar, speech, biomedicine, or seismology. The majority of research related to detection methods has been conducted within the areas of statistical modelling (Maximum Likelihood Estimators, Hidden Markov Models), image processing, neural networks, and simulated annealing (Lampert & O'Keefe, 2010).

In their simplest form, detection algorithms look for a target event in a time-domain signal. The performance of this type of detector can be improved for non-stationary signals by incorporating an extra dimension (e.g., time-frequency). Spectrogram correlators (Altes, 1980), Gabor representation (Friedlander & Porat, 1989), and Wigner-Ville Distribution (Boashash & O'Shea, 1990) are some of the most commonly used time-frequency detectors.

The large number of detection methods that currently exist can make it difficult for the researcher to select a suitable detector with an acceptable level of complexity. That is, a detector that will neither underperform nor require a considerable effort to implement. It is possible to narrow the search by identifying the nature of the target signal and the noise (random or deterministic, Gaussian or non-Gaussian, white or "coloured") and by determining what information is available for them. Below, a number of typical detection scenarios are listed with examples of suitable detectors and references.

- Random signal of known PDF[30] (white Gaussian) in noise of known PDF (white Gaussian). A suitable pick is the *energy detector*, the Neyman-Pearson's

[30] Probability Density Function

Likelihood Ratio Test (LRT) for this scenario (Tuteur, 1967; Kay, 1998; Urkowitz, 1967).

- Deterministic signal with known PDF in noise with known PDF. A related detector is the optimal matched filter (Turin, 1960).
- Random signal with known PDF in noise with known PDF. Related detectors are the Neyman Pearson's LRT, referred to as *estimator-correlator* (Tuteur, 1967; Kay, 1998), the Maximum Likelihood (ML) detector (Kay, 1998), the Maximum A Posteriori (MAP) detector (Kay, 1998), and the Bayes Risk (Kay, 1998).
- Deterministic signal with unknown parameters in noise with known PDF. Related detectors are the spectrogram correlator (Altes 1980; Xian et al., 2014); Gabor representation (Friedlander & Porat, 1989); Wigner-Ville Distribution (WVD) and Cross Wigner-Ville Distribution (XWVD) (Boashash & O'Shea, 1990; Oliveira & Barroso, 2000); Generalised Likelihood Ratio Test (GLRT) (Kay, 1998; Porat & Friedlander, 1986; Urkowitz, 1967); and wavelets (Bailey et al., 1998).
- Random signal with unknown parameters in noise with known PDF. A related detector is the Generalised Likelihood Ratio Test (GRLT) (Kay, 1998).

In many cases, the characteristics of the noise are known, greatly simplifying the detection process. For simplicity, Gaussian noise is generally assumed. If the noise is *non-Gaussian*, the performance of the detector will be affected, unless the detector is modified to minimise the effect of the *outliers* (sparse, high amplitude events that make the noise deviate from its Gaussian character). The LRT, GLRT, and Rao test are used by Kay (1998) to design detectors in non-Gaussian noise for deterministic signals with or without unknown parameters.

The complex theory behind most detectors and the large number of methods available are obstacles when it comes to deciding which method is the most appropriate for the task in hand. Applying a detection algorithm directly to the data can be an effective way of deciding if a detection solution is suitable, but this approach is impractical due to the lack of free software available to perform the tests. To the knowledge of the author, PAMGuard[31] is the only free software package in the field of underwater sound with detection functionality, consisting in a band-limited LRT method (Hood, 2009).

Given the importance that a reliable detector has in underwater sound source characterisation, and given the intricate theory behind most detectors, the lack of practical advice for their implementation, and the scarcity of freely-available detection software, we decided to study and implement a simple but effective detection algorithm.

[31] https://www.pamguard.org/

4.2 Detection Theory

The goal of detection theory is to decide when an event of interest occurs. Detection theory is also referred to as hypothesis testing or decision theory.

The work presented in this chapter is centred on *binary hypothesis testing*. Binary hypothesis testing refers to a *detection* problem, which consists in deciding between two possible hypotheses, as opposed to a *classification* or multiple hypothesis problem where a decision is made between three or more hypotheses.

A *simple hypothesis testing* approach is followed in the design of the detectors presented in this section. In simple hypothesis testing the PDF of each hypothesis is completely known, as opposed to *composite hypothesis testing* where the PDF of each hypothesis is partially known.

In statistical detection theory, detectors for simple hypothesis testing fall within two categories: *classical*, based on the Neyman-Pearson theorem; and *Bayesian*, based on the minimisation of the Bayes Risk. An example of a Neyman-Pearson detector is the Likelihood Ratio Test (LRT). A Bayesian detector is more elaborate as it incorporates prior knowledge of the probability of occurrence of the various hypotheses. From more to less generic, examples of Bayesian detectors include the Bayes Risk, the Maximum A Posteriori probability (MAP), and the Maximum Likelihood (ML).

The detectors described in the following section use the LRT criteria. This method is suitable for the automatic detection of target events in underwater sound recordings. In a typical sound source verification campaign, the start and end of a source operation and its emission rate are either unknown or known but susceptible to change at short notice. Therefore, using a Bayesian detector in sound field mapping studies is generally impractical and the LRT detector becomes the obvious choice. It is interesting to note that bioacoustics and underwater acoustic processing software such as RAVEN[32] and PAMGuard use band-limited LRT detectors.

Let's now have a look at the detection problem from the perspective of an LRT detector. Given a N-variable observation $\mathbf{x} = \{x[0], x[1], \dots x[N]\}$, the detection problem is defined as $\mathcal{H}_0$: $\mathbf{x} = \mathbf{w}$ and $\mathcal{H}_1$: $\mathbf{x} = \mathbf{s} + \mathbf{w}$, where $\mathbf{s}$ and $\mathbf{w}$ are the signal and noise vectors, $\mathcal{H}_0$ is the noise-only or *null* hypothesis, and $\mathcal{H}_1$ is the signal plus noise or *alternative* hypothesis. A detector is affected by two types of errors: type I or *false alarm*, if the detector decides $\mathcal{H}_1$ when $\mathcal{H}_0$ is true; or type II or *missed*, if the detector decides $\mathcal{H}_0$ when $\mathcal{H}_1$ is true.

The objective of an LRT detector is to maximise the *probability of detection* P_D subject to a target *probability of false alarm* α. Maximising P_D is equivalent to minimising the *probability of miss* P_M, since $P_D = 1 - P_M$. Considering that $\Pr\{\mathcal{H}_a, \mathcal{H}_b\}$ is the

[32] https://ravensoundsoftware.com

probability of deciding $\mathcal{H}_a$ when $\mathcal{H}_b$ has occurred, the probability of detection P_D and probability of false alarm P_{FA} are defined as

$$P_{FA} = \Pr\{\mathcal{H}_1, \mathcal{H}_0\} \tag{4.1}$$

$$P_D = \Pr\{\mathcal{H}_1, \mathcal{H}_1\} \tag{4.2}$$

The central problem addressed in detection theory is to find a sufficient statistic or *decision function* $T(\mathbf{x})$ that can be mapped into a decision by comparing $T(\mathbf{x})$ to a threshold γ. The *mean* is an example of a decision function for a deterministic signal of zero variance and varying DC offset in white Gaussian noise.

A *decision rule* is used to derive the decision function $T(\mathbf{x})$ and determine the *critical region*. The critical region is the sample space that defines when the alternative hypothesis $\mathcal{H}_1$ is accepted (i.e., decide that the signal is present). In a Neyman-Pearson detector the *likelihood ratio test* represents the optimal decision rule. The Neyman-Pearson theorem states that in order to maximise P_D for a given $P_{FA} = \alpha$, decide $\mathcal{H}_1$ if the LRT outcome $L(\mathbf{x})$ is higher than the threshold γ. $L(\mathbf{x})$ indicates the likelihood of $\mathcal{H}_1$ versus $\mathcal{H}_0$ for each variable in $\mathbf{x}$.

$$L(\mathbf{x}) = \frac{p(\mathbf{x};\, \mathcal{H}_1)}{p(\mathbf{x};\, \mathcal{H}_0)} > \gamma \tag{4.3}$$

$p(\mathbf{x};\, \mathcal{H}_1)$ and $p(\mathbf{x};\, \mathcal{H}_0)$ are the PDF of $\mathbf{x}$ for the alternative and null hypotheses. The critical region is given by $L(\mathbf{x}) > \gamma$. The threshold γ is found by applying the condition $L(\mathbf{x}) > \gamma$ to $P_{FA} = \alpha$.

$$P_{FA} = \Pr\{L(\mathbf{x}) > \gamma;\, \mathcal{H}_0\} = \int_{\{\mathbf{x}:\, L(\mathbf{x}) > \gamma\}} p(\mathbf{x};\, \mathcal{H}_0) d\mathbf{x} = \alpha \tag{4.4}$$

Note that there are many critical regions (or values of γ) that satisfy $P_{FA} = \alpha$, as each variable in $\mathbf{x}$ will have its own PDF. The NP theorem tells how to choose the best critical region given $p(\mathbf{x};\, \mathcal{H}_1)$, $p(\mathbf{x};\, \mathcal{H}_0)$, and α. In practice, the LRT will be used to find the appropriate decision statistic or *test statistic* $T(\mathbf{x})$ to provide a single statistical value for the vector $\mathbf{x}$, resulting in a single threshold γ'. The probability of false alarm can be expressed as a function of $T(\mathbf{x})$ and the updated threshold γ'.

$$P_{FA} = \Pr\{T(\mathbf{x}) > \gamma';\, \mathcal{H}_0\} = \int_{\gamma'}^{\infty} p(T(\mathbf{x});\, \mathcal{H}_0) dT \tag{4.5}$$

Then, it follows that the probability of detection is

$$P_D = \Pr\{T(\mathbf{x}) > \gamma';\, \mathcal{H}_1\} = \int_{\gamma'}^{\infty} p(T(\mathbf{x});\, \mathcal{H}_1) dT \tag{4.6}$$

Note that P_{FA} and P_D are the right-tail probabilities of PDF of the test statistic for the null and alternative hypotheses $p(T(\mathbf{x});\, \mathcal{H}_0)$ and $p(T(\mathbf{x});\, \mathcal{H}_1)$. For simplicity, we will refer to $p(T(\mathbf{x});\, \mathcal{H}_0)$ and $p(T(\mathbf{x});\, \mathcal{H}_1)$ as $p_0(T)$ and $p_1(T)$. Note that $T(\mathbf{x})$ is the observed test statistic while T represents the test statistic variable. The latter encompasses all possible values of $T(\mathbf{x})$ or γ'.

The last step in the detection process is the calculation of threshold γ'. Given a target probability of false alarm α, γ' can be easily obtained from the $P_{FA}(T)$ curve. The threshold γ' is then compared to the test statistic of each data segment $\mathbf{x}$ to be decided upon: the detector decides $\mathcal{H}_1$ if $T(\mathbf{x}) > \gamma'$, and $\mathcal{H}_0$ otherwise.

The probability functions of false alarm and detection $P_{FA,D}(T)$, or the corresponding density functions for the null and alternative hypotheses $p_{0,1}(T)$, characterise the performance of a NP detector. The next section discusses various aspects of detection performance that can help the reader understand the way in which the three NP detectors work. Some of the design choices made for the detection algorithms are based on these few concepts.

4.3 Aspects of Detection Performance

4.3.1 Deflection Coefficient

The performance of a Neyman-Pearson detector is fully characterised by the PDF of the test statistic for the null and alternative hypotheses $p_{0,1}$. The performance improves as the area overlap between the two probability density curves becomes smaller. Similarly, given a probability of false alarm P_{FA}, a larger probability of detection P_D is indicative of a better detection performance.

The concept of separation between probability functions is used in the *deflection coefficient*, given below, to quantify the performance of a detector

$$d = \frac{\mu_1 - \mu_0}{\sigma_0} \tag{4.7}$$

This parameter depends on the mean μ and standard deviation σ of $p_{0,1}(T)$. A larger value of d is associated with a better detection performance.

4.3.2 Energy and Variance as Test Statistics

The *energy* and *variance* are both valid test statistics for a NP detector. The energy of a zero-mean observation is a scaled version of the variance. A variance-based PDF can be derived from an energy-based PDF by dividing the x-axis and multiplying the y-axis by the number of variables N. It is important to note that the deflection coefficient,

hence the performance of a detector, does not vary when the PDF is scaled since μ_1, μ_0 and σ_0 are all multiplied by the same constant.

The three energy detectors presented in sections 4.6, 4.7, 4.8 use as test statistic three different forms of a weighted sum of squares. A variance-like test statistic, where the sum of squares is normalised by N, has not been considered for consistency with Kay's NP formulation and terminology.

That being said, the variance can be a more intuitive statistic when it comes to interpreting the performance of a detector, as the PDFs do not experience scaling as a result of a change in N. It is easier to visually identify changes in the deflection coefficient from variance-based PDFs. In line with this, the axes of the energy-based PDF curves included in this section are adjusted to compensate for the scaling caused by N.

4.3.3 Parameters Contributing to the Detection Performance

The probability densities $p_{0,1}(T)$ are strongly influenced by three parameters: the signal-to-noise ratio SNR, the number of variables N, and the number of observations M.

The SNR is the relation between the variances of the signal and noise, expressed in decibels as $20\log(\sigma_s/\sigma_w)$. A larger value of SNR produces an overall increase in the separation between $p_0(T)$ and $p_1(T)$. This results in a larger deflection coefficient in which the *difference of means* (numerator) dominates. The signal-to-noise ratio is the most critical parameter in the performance of a detector, since any signal with a sufficient SNR can be detected without the need for a sophisticated detection algorithm.

In an audio signal, each sample of an observed waveform represents a variable. A larger number of variables N results in a smaller overlap between $p_0(T)$ and $p_1(T)$, but unlike the SNR, the performance is improved through the reduction of the *relative standard errors* σ_0/μ_0 and σ_1/μ_1. In practice, N is conditioned by the duration τ and sampling frequency f_s of the target signal. A value of N larger than $\tau \cdot f_s$ can be selected, but at a cost in detection performance due to a lower SNR. The effect of N on the detection performance is discussed and compared to the effect of the SNR in Sub-Section 4.3.5.

The number of variables N not only has an effect on the performance of a detector, but also on its formulation. The asymptotic or large sample theory considers that a statistical problem can be confidently evaluated in the limit $N \to \infty$ when the number of variables N is large. Under this assumption, the derivation of analytical expressions can be greatly simplified. One important consequence of using a large N is the convergence of the signal's probability distribution to a Gaussian form, as described by the *central limit theorem*. The central limit theorem and its application to the NP detectors is discussed in Sub-Section 4.3.6.

The number of observations M is the last key parameter. It only affects the performance of the estimator-correlator, a detector that uses the covariance information of its target

signal and noise to enhance the detection performance. Incorporating this new information comes at a cost, as the covariance matrix must be accurate for the detector to behave as expected. The commonly used *sample covariance matrix* (SCM) is a poor estimator when the number of observations M is comparable to or lower than the number of variables N. This leads to the use of alternative covariance estimators that can minimise the error in the covariance matrix. This issue is discussed in Sub-Section 4.3.7.

There is a fundamental difference between the number of variables and the signal-to-noise ratio: whereas N is a parameter of the detector, SNR is a feature of the observation. The SNR cannot be directly controlled, but it can be maximised if there is previous knowledge of the temporal and spectral characteristics of the signal and noise. In Section 4.10 we discuss how resampling, filtering, and selecting an appropriate length for the observation can be used to eliminate redundant information and, in that way, help improve the detection performance and processing speed.

4.3.4 Visualisation of Detection Performance

Three types of graphs are used throughout the chapter to support the analysis of the performance of the three NP detectors. The graphs show the relationship between two of the following variables: the probability functions of detection and false alarm $P_{D,FA}(T)$, the signal-to-noise ratio SNR, and the detection threshold γ'. Ultimately, the performance of the detector will depend upon *how different* the PDFs are under each hypothesis.

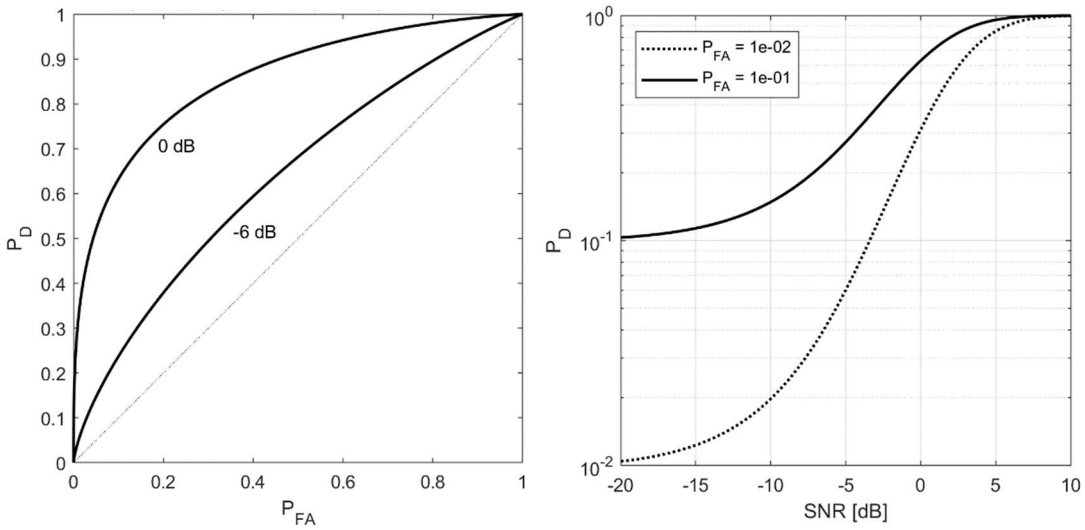

Figure 4.1 Receiver Operation Characteristic (ROC) for two SNR values (left) and detection probability dependence on SNR for two false alarm probabilities (right). Curves computed with a NP energy detector applied to a vector of length $N = 10$.

A common way of representing the performance is by plotting $P_D(T)$ against $P_{FA}(T)$ for different values of γ', what is known as the *Receiver Operating Characteristic* (ROC) (see Figure 4.1, left). In a ROC, a 45° line represents the worst-possible detector, one

that randomly decides between either hypothesis. Any curve above the 45° line indicates that the detector is capable of making a decision using the information in data vector $\mathbf{x}$. The further the inflection point in the ROC curve is from the 45° line, the better the detection performance, as $P_{FA} \to 0$ and $P_D \to 1$. Each curve is computed for a specific SNR and N combination. The ROC is particularly useful for selecting a target probability of false alarm α that results in an acceptable detection probability for a low, reference SNR.

Another useful graph consists in plotting P_D against SNR for a given value of P_{FA} and N (see Figure 4.1, right). P_D converges to unity for large values of SNR: the lower the SNR at which convergence occurs, the better the detection performance. Poor detection performance occurs below a certain SNR, where P_D converges to P_{FA}. A NP detector is designed to behave optimally given a target probability of false alarm $P_{FA} = \alpha$. Once α has been selected, this representation can help identify the SNR below which the performance of the detector is compromised.

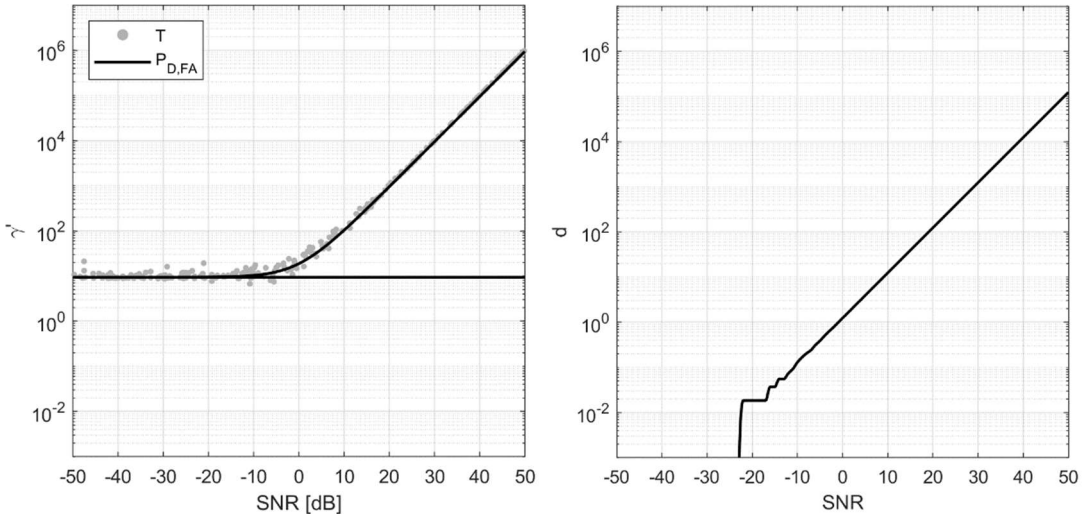

Figure 4.2 Dependence of false alarm and detection thresholds with SNR for target probabilities $P_D = P_{FA} = 0.5$ (left) and dependence of deflection coefficient with SNR (right). The grey dots represent the test statistic of a simulated Gaussian random process of variance $\sigma_s^2 + \sigma_w^2$. The curves are computed with a NP energy detector applied to a vector of length $N = 10$.

The graph in Figure 4.2 (left) has been specifically designed for this chapter, as a tool for comparing the predicted performance of a detector with results from real data. The thresholds $\gamma_D{}'$ and $\gamma_{FA}{}'$ associated with the fixed target probabilities $P_D = P_{FA} = 0.5$ are plotted against the SNR. The detection performance improves as the ratio $\gamma_D{}'/\gamma_{FA}{}'$ increases with SNR. For the target probabilities $P_D = P_{FA} = 0.5$, $\gamma_{D,FA}{}'$ coincide with the mean values $\mu_{1,0}$ of the probability density curves. The values of the test statistic $T(\mathbf{x})$, calculated over observations of signal plus noise (grey dots), are centred in the $\gamma_D{}'(SNR)$ curve ($P_D = 0.5$) and spread according to the standard deviation σ_1 of $p_1(T(\mathbf{x}))$, with an asymmetry characteristic of the Chi-squared distribution.

The greatest advantage of this type of graph is that the predicted curves $\gamma_{D,FA}{}'(SNR)$ can be represented along with the values of the test statistic computed from multiple vector observations. This offers a way of verifying the real performance of a detector, which can be affected by the spread of the outcomes of the test statistic (e.g., low number of variables), by an inaccurate estimate of the covariance matrix, or by a shorter duration of the signal contained in the data vector. These three factors are discussed in Sub-Section 4.3.5, Sub-Section 4.3.7, and Section 4.10.

Figure 4.2 (right) shows the deflection coefficient d as a function of SNR. The equivalence between $\gamma_{D,FA}{}'$ for $P_{D,FA} = 0.5$ and $\mu_{1,0}$ explains why the deflection coefficient increases with SNR with a gradient identical to that of $\gamma_D' - \gamma_{FA}{}'$ (see Figure 4.2, left). In a similar way to the deflection coefficient, $\gamma_D{}'/\gamma_{FA}{}'$ can be used for characterising the detection performance. As a relative metric, $\gamma_D{}'/\gamma_{FA}{}'$ is not affected by the scaling of $p_{1,0}$ experienced by the energy test statistic and caused by a change in the number of variables N. However, the deflection coefficient is a more complete performance metric as it accounts for the standard deviation σ_0 as well as the means $\mu_{1,0}$.

4.3.5 Asymptotics

Figure 4.3 presents the PDFs of the energy detector for the null and alternative hypotheses $p_{0,1}$ for two different vector lengths ($N = 10$ left, $N = 100$ right). There are two features that stand out when comparing the two images: (1) the overlap between the two PDF, and (2) the shape of the PDF. The effect of N on the shape of the PDF is discussed in Sub-Section 4.3.6.

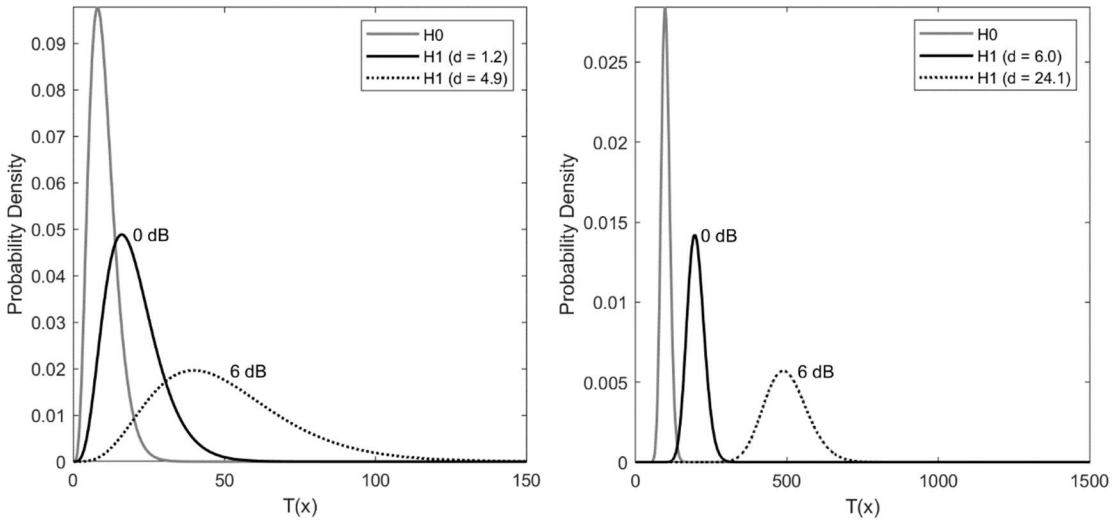

Figure 4.3 Probability density function of the *energy detector* for the null hypothesis (grey) and alternative hypothesis (black) for the cases of vector length $N = 10$ (left) and $N = 100$ (right). The alternative hypothesis is shown for signal-to-noise ratios of 0 dB and 6 dB.

The overlap between p_0 and p_1 is smaller for larger values of N. Effectively, a larger record length N incorporates additional information that is used to build a stronger

statistic $T(\mathbf{x})$, which results in a more confident decision. A larger N reduces the standard deviation $\sigma_{0,1}$ of the PDFs relative to their mean values $\mu_{0,1}$, thus increasing the deflection coefficient and improving the detection performance. For the energy detector in Section 4.6, if we alter the record length from N_a to N_b, the mean μ_a and variance σ_a^2 of the PDF become $\mu_a \cdot k$ and $\sigma_a^2 \cdot k$, where $k = N_b/N_a$. Hence, the deflection coefficient increases by $\sqrt{k}$. Note that this scaling rule only holds when the signal length N_s matches the record length N.

The performance can also be improved by increasing the SNR. In Figure 4.3, the solid and dotted black lines represent the probability density of the alternative hypothesis p_1 for SNR of 0 dB and 6 dB. The comparison between the two curves shows that an increase in SNR results in the scaling (expansion) of p_1 and in an increase in the deflection coefficient. For the energy detector, if the signal-to-noise ratio changes from SNR_a to SNR_b, the mean $\mu_{1,a}$ and variance $\sigma_{1,a}^2$ of the alternative hypothesis become $\mu_{1,a} \cdot k$ and $\sigma_{1,a}^2 \cdot k^2$, where $k = \left(1 + 10^{SNR_b/10}\right)/\left(1 + 10^{SNR_a/10}\right)$.

In a detection problem, the SNR is typically low and a large data length is essential for mapping a vector into the right decision. Whilst a larger N will improve the detection performance, in practice its maximum value is limited by the target signal. At first, using a N value larger than the length of the target signal N_s, or using a sampling rate larger than twice the signal's maximum frequency, may appear as reasonable ways of improving the detection performance. However, these will in fact have the opposite effect. There is only so much that can be done to enhance the performance a detector given a target signal and noise. Still, there are a few ideas that can applied to optimise the performance of a detector, as discussed in Section 4.10.

4.3.6 Gaussian vs Non-Gaussian

If $\mathbf{x} = \{x_1, x_1, \ldots x_N\}$ is a vector of N independent and identically distributed (IID) normal random variables with variance σ_x^2 and zero mean, then the energy test statistic $T(\mathbf{x}) = \sum_i^N x_i^2$ follows a *central chi-squared* distribution with N degrees of freedom (DOF). Formally, this can be expressed as $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma_x^2\mathbf{I})$ and $T(\mathbf{x}) \sim \sigma_x^2 \mathcal{X}_N^2$, where $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ is the multivariate normal distribution with vector of means $\boldsymbol{\mu} = \{\mu_1, \mu_2, \ldots \mu_N\}$ and covariance matrix $\mathbf{C}$, $\mathcal{X}_N^2$ is the central chi-squared distribution of N degrees of freedom, $\mathbf{0}$ is a vector of zeroes, and $\mathbf{I}$ is the identity matrix.

In Figure 4.4, the PDFs in the left image ($N = 10$) have the characteristic non-symmetric shape of a chi-squared distribution, but on the right image ($N = 100$) the PDFs are more symmetric and similar to a Gaussian distribution. This effect is described by the central limit theorem, which states that independent random variables will converge to a normal distribution as N approaches infinite. In other words, the initially non-Gaussian statistic $T(\mathbf{x})$ can be approximated to a Gaussian distribution if N is large and $\mathbf{x}$ consists of IID variables. For the test statistic $T(\mathbf{x})$ with central Chi-squared distribution

of N degrees of freedom, the mean and variance of the equivalent Gaussian function for large N are $\sigma_x^2 N$ and $\sigma_x^4 2N$.

But how large is a "large" value of N? Figure 4.4 compares the central Chi-squared PDF (solid line) to its equivalent Gaussian PDF (dashed line) for two sample lengths. The RMS error, referred to the maximum of the Chi-squared curve and expressed as percentage, is 9.7% for $N = 10$ (left image) and 3% for $N = 100$ (right image). For $N \geq 100$, the equivalent Gaussian function of mean $\sigma_x^2 N$ and variance $\sigma_x^4 2N$ can be considered as a reasonable approximation to the Chi-squared PDF. In practice, N is much larger than 100 due to the large audio sampling rates (typically $f_s > 24\,\text{kHz}$) and signal durations (0.1-1 s), and a Gaussian distribution can be confidently assumed.

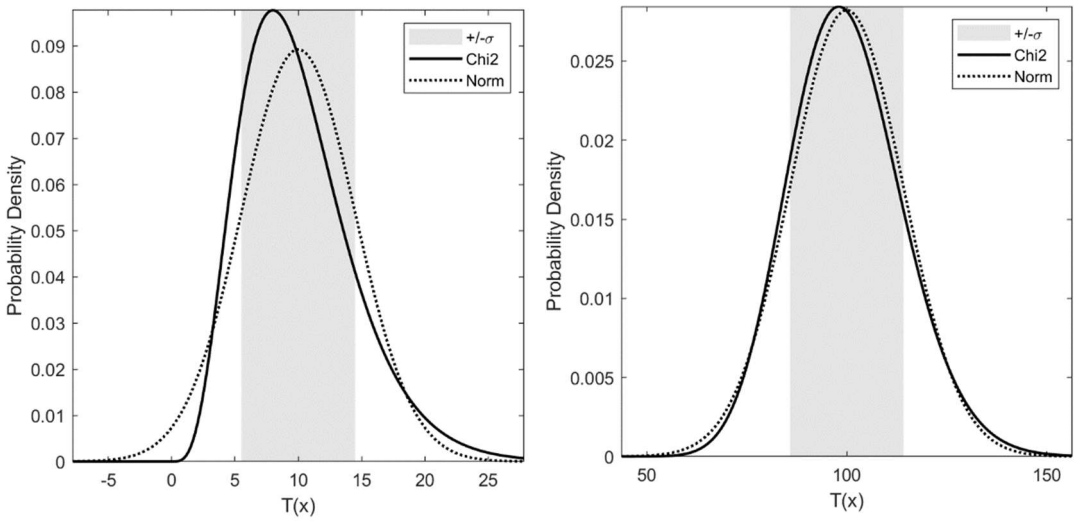

Figure 4.4 Probability density function of the *energy detector* for $\mathbf{x} \sim \mathcal{N}(0,1)$ for the cases of vector length $N = 10$ (left) and $N = 100$ (right). The central Chi-squared PDF of $T(\mathbf{x})$ is shown as a solid line and the equivalent Gaussian PDF $\mathcal{N}(N, 2N)$ is shown as a dashed line. The standard deviation $\pm\sqrt{2N}$ is delimited by a grey polygon centred at the mean N.

4.3.7 The High-Dimensional Problem

The *covariance matrix* and its inverse (the *precision matrix*) play a fundamental role in multivariate analysis. They are at the core of a wide range of statistical tools, such as principal component analysis, linear discriminant analysis, and detection theory. The covariance matrix is a critical component in the formulation of the estimator correlator. As such, estimating it accurately is essential to guarantee a good detection performance. But what is the covariance matrix and why calculating it may not be straightforward?

The covariance matrix is a *square*, *symmetric*, and *positive-definite* matrix that contains the variances (diagonal) and covariances (off-diagonal) of a vector of N variables. Given an N-variable random process defined by column vector $\mathbf{x} = \{x_1, x_2, \ldots x_N\}^T$, the true or *population* covariance matrix (PCM) $\boldsymbol{\Sigma}$ is

$$\mathbf{\Sigma} = E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T] \tag{4.8}$$

where $\boldsymbol{\mu} = \{E(x_1), E(x_2), \dots E(x_N)\}^T$ is the vector of means and $E(\cdot)$ is the expectation. The simplest and most frequently used estimator of the PCM is the sample covariance matrix (SCM) $\mathbf{S}$. In an audio signal, $\boldsymbol{\mu}$ can be assumed to be zero. For a $[N, M]$ *matrix of raw scores* $\mathbf{X}$ containing M zero-mean observations of $\mathbf{x}$, the SCM can be expressed in matrix form as

$$\mathbf{S} = \frac{1}{M}\mathbf{X}\mathbf{X}^T \tag{4.9}$$

The SCM is the optimal estimator in the classical *low-dimensional* scenario of low N and $M \gg N$. It has the advantages of being *unbiased* (i.e., same expected value as the PCM) and easy to compute. But when the number of variables exceeds the number of observations, $\mathbf{S}$ is singular (non-invertible) and a "noisy" unreliable estimator of the PCM. This limitation of the SCM, what we refer to as the *high-dimensional problem*, has motivated an entire line of research aimed at finding alternative methods for estimating the PCM in a high-dimensional setting.

Nowadays, the technological advance has contributed to a massive increase in the number of variables and observations that can be collected for a dataset. Disciplines such as bioinformatics, climate studies, risk management, or audio processing consistently work with high-dimensional data. For example, audio signals tend to be sampled at frequencies of the order of tens of kHz and have durations of the order of hundreds of milliseconds, leading to thousands of variables. Gathering the large number of observations that is necessary to obtain a well-conditioned, reliable SCM in this context is generally impractical. Fortunately, from the early 2000's there has been a rapid growth in the field of high-dimensional covariance estimation for statistics and machine learning (Pourahmadi, 2013), and currently there is a wide range of algorithms suitable for practically any type of data. The most relevant methods for covariance and precision matrix estimation are discussed in Section 4.5. The formulation and concepts introduced later in this chapter will benefit from a summary of the properties of the covariance matrix and its spectral and Cholesky decomposition. These topics are addressed in the following section.

4.4 The Covariance Matrix and its Decomposition

4.4.1 Properties of the Covariance Matrix

The covariance matrix is *square* of dimensions $[N, N]$, *symmetric* (i.e., equal to its transpose, that is, $\mathbf{\Sigma} = \mathbf{\Sigma}^T$ or $\varsigma_{ij} = \varsigma_{ji}$ with $\mathbf{\Sigma} = [\varsigma_{ij}]$), and *nonnegative definite* (i.e., its eigenvalues are positive or zero). Any estimator must respect all these three properties.

There are two other properties that are desirable: being *well-conditioned* and *structured*. A *well-conditioned* matrix is *full rank* and not close to singular, meaning that its inverse (or precision matrix) exists and can be obtained with high accuracy. An example of a badly conditioned covariance estimator is the SCM computed from a number of vector observations M that is comparable to (but larger than) the number of variables N (for $M < N$ the SCM is singular).

A covariance matrix can also be highly *structured*. If that structure or general pattern is known, it will be easier to select a suitable and effective covariance estimator. Some of the most common covariance structures are scaled identity, autoregressive, compound symmetry, Toeplitz, variance components, spatial, and unstructured (IBM, 2021). Heterogeneous versions, where the entries in the main diagonal are different, exist for the first four patterns. The scaled identity, the heterogeneous scale identity or *diagonal*, and the spatial structures (e.g., linear, exponential) are typically occurring patterns. Sometimes, covariance matrices are *unstructured* or their pattern is unknown, in which case an estimator that makes no structural assumptions is preferred (e.g., shrinkage).

4.4.2 Eigenvalue and Eigenvector Decomposition

The spectral or *eigen* decomposition is the name given to the process of breaking down a square matrix into its characteristic vectors (or *eigenvectors*) and their weightings (or *eigenvalues*). The eigen decomposition of the covariance matrix is of special importance in multivariate analysis and is at the core of statistical tools such as principal component analysis (PCA).

A square matrix $\mathbf{A}$ has a N-variable eigenvector $\mathbf{v}$ and scalar eigenvalue λ if

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \tag{4.10}$$

The eigenvalue is obtained from the N-degree polynomial equation $|\mathbf{A} - \lambda\mathbf{I}| = 0$. The associated eigenvector is obtained by solving the system of N equations $(\mathbf{A} - \lambda\mathbf{I})\mathbf{v} = 0$.

In practice, a $[N, N]$ square matrix has N pairs of eigenvectors and eigenvalues. The general form of equation (4.10) for the entire set of eigenvectors and eigenvalues is

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \tag{4.11}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots \lambda_N)$ is the *diagonal matrix* or matrix of eigenvalues and $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_p\}$ is the *modal matrix* or matrix of eigenvectors.

If $\mathbf{A}$ is symmetric, as are the covariance and precision matrices, then the eigenvalues are *real*, the eigenvectors are *mutually orthogonal*, and the modal matrix is *orthogonal*. In an orthogonal matrix, the inverse is equal to its transpose ($\mathbf{V}\mathbf{V}^T = \mathbf{I}$).

From (4.11), $\mathbf{A}$ can be written as a function of its modal and diagonal matrices.

$$\mathbf{A} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\boldsymbol{T}} = \sum_{i=1}^{p} \lambda_i \mathbf{v}_i \mathbf{v}_i^T \tag{4.12}$$

This expression is known as the *spectral decomposition* of $\mathbf{A}$, and shows that a symmetric matrix can be expressed as the sum of weighted contributions from its eigenvectors.

Another consequence of (4.11) is that a symmetric matrix can be diagonalised using its modal matrix, as in

$$\boldsymbol{\Lambda} = \mathbf{V}^T\mathbf{A}\mathbf{V} \tag{4.13}$$

In a covariance matrix, *diagonalisation* is equivalent to decorrelating all its variables. Thus, in the transformed space (eigen space) all variables are independent, and each is characterised by a unique vector (eigenvector) and weighting (eigenvalue).

If a function $f(\cdot)$ is applied to a symmetric matrix $\mathbf{A}$, the computation can be greatly simplified by using its spectral decomposition.

$$f(\mathbf{A}) = \mathbf{V}f(\boldsymbol{\Lambda})\mathbf{V}^T \tag{4.14}$$

The advantage is that applying a function to a diagonal matrix is equivalent to modifying its diagonal elements by that same function, $f(\boldsymbol{\Lambda}) = \mathrm{diag}\big(f(\lambda_1), f(\lambda_2), \ldots f(\lambda_p)\big)$. For example, $\mathbf{A}^{-1} = \mathbf{V}\boldsymbol{\Lambda}^{-1}\mathbf{V}^T$ or $\log(\mathbf{A}) = \mathbf{V}\log(\boldsymbol{\Lambda})\mathbf{V}^T$. From the latter it can be deduced that the logarithm of a positive definite symmetric matrix is *real symmetric*.

The ratio between the maximum and minimum eigenvalue in $\boldsymbol{\Lambda}$ is known as the *condition number*. This term indicates how sensitive a matrix is to roundoff errors when operations are applied to it. A matrix with a large condition number is poorly-conditioned and close to singular.

4.4.3 Modified Cholesky Decomposition

Like the eigen decomposition, the Cholesky decomposition is a form of covariance matrix reparameterisation. It expresses the covariance matrix $\boldsymbol{\Sigma}$ as a product of a lower diagonal matrix $\mathbf{C} = \left[c_{ij}\right]$ and its transpose $\mathbf{C}^T$ ($\boldsymbol{\Sigma} = \mathbf{C}\mathbf{C}^T$). A more common form is the *modified Cholesky decomposition* (MCD) given by

$$\boldsymbol{\Sigma} = \mathbf{L}\mathbf{D}\mathbf{L}^T \tag{4.15}$$

or

$$\boldsymbol{\Sigma}^{-1} = \mathbf{T}^T\mathbf{D}^{-1}\mathbf{T} \tag{4.16}$$

$$\mathbf{D} = \mathbf{T}\boldsymbol{\Sigma}\mathbf{T}^T \tag{4.17}$$

where $\mathbf{D} = \mathbf{D}_0^2$ and $\mathbf{D}_0 = \mathrm{diag}(c_{11}, c_{22}, \ldots c_{NN})$ are diagonal matrices, and $\mathbf{L} = \mathbf{C}\mathbf{D}_0^{-1}$ and $\mathbf{T} = \mathbf{L}^{-1}$ are lower triangular matrices. The similarity between (4.15) and (4.12) is evident, but unlike the orthogonal modal matrix $\mathbf{V}$ from the spectral decomposition, the entries in $\mathbf{T}$ are *unconstrained*.

One can arrive at $\mathbf{T}$ and $\mathbf{D}$ through a linear least square regression of each variable in a naturally ordered random vector $\mathbf{y}$ based on its predecessors (Pourahmadi, 2013). Hence, regression methods can be used to compute the precision and covariance matrices from the estimated Cholesky factors $\widehat{\mathbf{T}}$ and $\widehat{\mathbf{D}}$. However, the MCD can only be applied when the variables in $\mathbf{y}$ follow a *natural order*, as in time-dependent or spectrogram data.

An important consequence of the MCD is that the precision and covariance matrices estimated with (4.16) are *always positive-definite*.

Examples of the application of the modified Cholesky decomposition to the estimation of a high-dimensional covariance matrix can be found in Wu & Pourahmadi (2003), Huang et al (2006), Bickel & Levina (2008a), Levina et al (2008), Rothman et al (2008) and Besson et al (2020).

4.4.4 Latent Factor Models

Factor analysis is a common dimensionality-reduction tool used for identifying the main unobservable or *latent* factors on which data depends, and is typically employed in financial applications.

In a factor model, the covariance matrix is decomposed as follows

$$\mathbf{\Sigma} = \mathbf{L}\mathbf{\Lambda}\mathbf{L}^T + \mathbf{\Psi} \tag{4.18}$$

where $\mathbf{L}$ is a $[M, N]$ matrix, and $\mathbf{\Lambda}$ and $\mathbf{\Psi}$ are usually diagonal matrices of dimensions $[N, N]$ and $[M, M]$. The spectral decomposition, modified Cholesky decomposition, and principal component analysis (PCA) are specific types of factor models.

Factor models have been applied to the estimation of high-dimensional covariance matrices by Fan et al (2008) and Fan et al (2013).

4.5 Methods for High-Dimensional Covariance Estimation

4.5.1 Covariance Estimation Methods

The methods available for estimating covariance matrices can be divided into two broad groups: (1) those making structural assumptions about the covariance matrix, and (2) those that don't (Huang, 2016). The first group comprises a wide variety of solutions that include banding, tapering, thresholding, maximum likelihood, and sparse Gaussian graphical models. Shrinkage algorithms dominate the second group. In the first group

the concept of *sparsity* plays a central role. The sparsity principle assumes that only a small number of parameters are nonzero.

Many of these methods employ some form of decomposition of the covariance and precision matrices (see Section 4.4) that allow regression techniques to be applied to the complex task of covariance estimation. In Sub-Section 4.5.3 we describe the most common forms of linear regression used for this purpose. The various methods for covariance and precision matrix estimation are presented next, starting with *shrinking* algorithms and continuing with solutions from the first group.

Any estimator of the population covariance and precision matrices should be *positive-definite*, *well-conditioned*, *efficient*, and *accurate* (Schafer & Strimmer, 2005). The positive-definite condition is a major obstacle to estimating covariance matrices in the high-dimensional setting (Pourahmadi, 2013), and meeting that condition often involves adding extra steps and complexity to an existing algorithm. There is still a reasonable number of available positive-definite (PD), well-conditioned (WC) estimators with varying degrees of accuracy or performance. In the next sub-section, we revise the concept of *loss* and *risk* functions and their role in assessing the performance of a covariance estimator.

4.5.2 Estimator Performance: Loss, Risk and Bias-Variance Tradeoff

The performance of a covariance matrix estimator is given by its *loss* and *risk* functions. Both parameters measure how close the estimation is to the true value. But while the *loss* indicates how well the estimator performs against the *training data*, the *risk* informs about the estimator's performance based on validation or *test data*. The risk is the expectation of the loss over several tests. Low loss with high risk is a sign of *overfitting*, meaning that the estimator performs well with the training data but poorly with the test data. A good estimator features low risk for a suitable risk function.

There are several loss functions and associated risk functions to choose from. Which one to pick depends on the type of data. Two of the most common loss functions are the Kullback-Liebler divergence of the true and estimated covariance matrices $L_1(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma})$, commonly referred to as Stein's or *entropy* loss, and the Euclidean or Frobenius norm of the difference between the true and estimated covariance matrices $L_2(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma})$, also referred to as *quadratic* loss (Haff, 1980; Stein, 1961).

$$L_1(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma}) = \mathrm{tr}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}) - \log|\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1}| - N \tag{4.19}$$

$$L_2(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma}) = \mathrm{tr}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Sigma}^{-1} - I)^2 \tag{4.20}$$

The corresponding risk functions are $R_i(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma}) = E[L_i(\widehat{\boldsymbol{\Sigma}}, \boldsymbol{\Sigma})]$ with $i = 1,2$. The function $\mathrm{tr}(\mathbf{A})$ represents the *trace* or sum of the diagonal entries of a square matrix $\mathbf{A}$.

The *mean squared error* (MSE) is another important measure of the quality of a covariance estimator. Given a vector $\boldsymbol{\beta} = \{\beta_1, \beta_2, \dots \beta_P\}^T$, there exists an estimator $\widehat{\boldsymbol{\beta}} = \{\hat{\beta}_1, \hat{\beta}_2, \dots \hat{\beta}_P\}^T$ that is a function of the matrix of observations $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_M\}$ with individual multivariate observations $\mathbf{x}_j = \{x_{1j}, x_{2j}, \dots x_{Nj}\}$. The MSE between $\boldsymbol{\beta}$ and $\widehat{\boldsymbol{\beta}}$ is given by (Lebanon, 2010)

$$\mathrm{MSE}(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}) = E\left(\left\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right\|^2\right) = \mathrm{tr}\left[\mathrm{var}(\widehat{\boldsymbol{\beta}})\right] + \left\|\mathrm{bias}(\widehat{\boldsymbol{\beta}})\right\|^2 \tag{4.21}$$

$$\mathrm{tr}\left[\mathrm{var}(\widehat{\boldsymbol{\beta}})\right] = \sum_{j=1}^{P}\left[E(\hat{\beta}_j^2) - E(\hat{\beta}_j)^2\right] \tag{4.22}$$

$$\left\|\mathrm{bias}(\widehat{\boldsymbol{\beta}})\right\|^2 = \sum_{j=1}^{P}\left[E(\hat{\beta}_j) - \beta_j\right]^2 \tag{4.23}$$

where $\mathrm{var}(\widehat{\boldsymbol{\beta}})$ is the covariance matrix of $\widehat{\boldsymbol{\beta}}$ and $\|\mathbf{a}\|^2 = \sum_j a_j^2$ is the squared Frobenius norm of a vector $\mathbf{a}$. The trace term $\mathrm{tr}\left[\mathrm{var}(\widehat{\boldsymbol{\beta}})\right]$ represents the sum of variances of the P variables in the estimation vector. $\left\|\mathrm{bias}(\widehat{\boldsymbol{\beta}})\right\|^2$ is the sum of squared biases from all variables in $\widehat{\boldsymbol{\beta}}$. Note that the above formulation is also applicable to matrices (i.e., $\boldsymbol{\beta} \equiv \mathrm{vec}(\boldsymbol{\Sigma})$).

Hence, the MSE is the combination of the estimator's *variance* and *squared bias*. The sample covariance matrix (SCM) is an *unbiased estimator* of the population covariance matrix (PCM), but it involves considerable risk due to the increase in variance when $N \gg M$. A good estimator would reduce the excessive variance at the expense of adding a small amount of bias. The search for the optimum balance between bias and variance that minimises the risk for an estimator is what is known as the *bias-variance tradeoff*.

4.5.3 Penalised Regression

The estimation of high-dimensional covariance and precision matrices can be reduced to solving a sequence of regularised or *penalised* regressions (Pourahmadi, 2013). Despite not being a covariance estimation method in itself, penalised regression is used by most modern algorithms to tackle the difficult problem of high-dimensional covariance estimation. The aim of regularised regression is to minimise the discrepancies between the observations and the model subject to a constraint.

Linear Least Squares

The linear least squares (LLS) is a *non-penalised* regression approach that forms the basis for other regularised regression techniques. Its goal is to minimise the sum of squared differences between a linear fit and the system's response to a number of multivariate observations. Let $\mathbf{y} = \{y_1, y_2, \dots y_M\}$ be the *vector of responses* to the M observations of N-variable vectors in the $[N, M]$ *matrix of observations* $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_M\}$,

$\boldsymbol{\beta} = \{\beta_1, \beta_2, \ldots \beta_N\}$ the *vector of regression coefficients*, and $\boldsymbol{\varepsilon} = \{\varepsilon_1, \varepsilon_2, \ldots \varepsilon_M\}$ the *error vector*. The LLS regression equation, associated loss function and its solution are given by (Geer, 2005)

$$\mathbf{y} = \mathbf{X}^T\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad \text{Regression Equation} \tag{4.24}$$

$$L(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}^T\boldsymbol{\beta}\|_2^2 \quad \text{Loss Function} \tag{4.25}$$

$$\boldsymbol{\beta} = \left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)^{-1}\mathbf{X}\mathbf{y} \quad \text{Regression Vector} \tag{4.26}$$

where $\|\mathbf{a}\|_2^2 = \sum_i |a_i|^2$ is the squared Frobenius norm. Note that the product $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ represents the scaled SCM. In the high-dimensional setting, the LLS method fails since the SCM is not invertible, as can be deduced from (4.26). Therefore, some form of regularisation is needed.

Ridge Regression

The *ridge regression* (RR), developed by Hoerl & Kennard (1970), addresses the limitation of the LLS regression by adding a penalty term

$$L(\boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}^T\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 \quad \text{Loss Function} \tag{4.27}$$

$$\boldsymbol{\beta} = \left(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}\mathbf{y} \quad \text{Regression Vector} \tag{4.28}$$

The penalty term comprises the so-called *ridge penalty* $\|\boldsymbol{\beta}\|_2^2$ corresponding to the squared Frobenius or L_2 norm of the regression vector $\boldsymbol{\beta}$ (i.e., sum of its squared values), and the *tuning parameter* λ. The RR approach is equivalent to minimising $\|\mathbf{y} - \mathbf{X}^T\boldsymbol{\beta}\|_2^2$ subject to $\|\beta_{RR}\|_2^2 < c$, where c is a constant. This has the effect of shrinking towards zero the regression coefficients with the lowest contribution to the response. However, as close to zero as some of the regression coefficients may be, none are set to zero when using RR. The ability to induce sparsity (i.e., to make the smallest coefficients zero) is desirable when only a small number of variables contribute to the outcome. The lasso regression overcomes this limitation.

Lasso Regression

The *lasso regression* (LR), due to Tibshirani (1996), induces sparsity by applying an alternative penalty term.

$$L(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \quad \text{Loss Function} \tag{4.29}$$

The *lasso penalty* $\|\boldsymbol{\beta}\|_1$ is the L_1 norm of the regression vector $\boldsymbol{\beta}$, that is, the sum of its absolute values. The LR approach is equivalent to minimising $\|\mathbf{y} - \mathbf{X}^T\boldsymbol{\beta}\|_2^2$ subject to $\|\boldsymbol{\beta}\|_1 < c$. The lasso regression applies shrinkage (like the RR), but also *forces to zero* the smallest regression coefficients, with λ controlling the sparsity.

The lasso solution has no closed form, unlike the LLS and RR. The *coordinate descent algorithm*, a method commonly used to compute the lasso, iteratively minimises $L(\boldsymbol{\beta})$ over each regression coefficient until convergence (Friedman et al, 2007; Friedman et al, 2010). Thresholding operators, such as *soft* or *hard* thresholding, are fundamental to this process.

The lasso is known to shrink larger values unnecessarily. There are a few alternatives that address this problem, such as adaptive lasso (Zou, 2006), elastic net (Zou & Hastie, 2005) and Smoothly Clipped Absolute Deviation or SCAD (Fan & Li, 2001). The *adaptive lasso* applies a weighted version of L_1 with regression coefficients weighted based on the data. In the *elastic net* the penalty is a linear combination of the ridge and lasso penalties. The *SCAD* uses an optimal penalty to obtain a sparse, unbiased, and continuous covariance estimator.

4.5.4 Shrinking

The *condition number* of the SCM tends to be larger than in the PCM in the high-dimensional setting ($N > M$). Simply put, the largest eigenvalues of the PCM are overestimated while the smallest are underestimated. Shrinking methods work by narrowing the eigenvalues of the SCM towards those of a target covariance matrix without altering the eigenvectors.

There are three main classes of shrinkage estimators: those regularising the eigenvalues of the SCM, those regularising the Cholesky factor of the SCM, and those using a linear combination of the SCM and a target matrix (Besson et al., 2020). The principle is the same: to minimise a suitable risk function to arrive at the estimator.

The first class, known as *Steinian shrinkage* estimators, were proposed by Stein (1956; 1975). These estimators are of the form $\widehat{\boldsymbol{\Sigma}} = \mathbf{V}\widehat{\mathbf{D}}\mathbf{V}^T$, where $\widehat{\mathbf{D}} = \mathrm{diag}(\hat{\boldsymbol{\lambda}})$ is the diagonal matrix of estimated eigenvalues, and $(\mathbf{V}, \mathbf{D})$ are the modal and diagonal matrices of the SCM. Each estimated eigenvalue $\hat{\boldsymbol{\lambda}}_i$ is a non-linear function of the SCM eigenvalues $\boldsymbol{\lambda}$. The method produces negative and non-monotone eigenvalues $\hat{\boldsymbol{\lambda}}_i$, but a procedure suggested by Stein helped overcome this problem (Lin & Perlman, 1984).

The second class of shrinkage estimators regularises the modified Cholesky factor of the SCM. These estimators are of the form $\widehat{\boldsymbol{\Sigma}} = \mathbf{L}\widehat{\mathbf{D}}\mathbf{L}^T$, where $\widehat{\mathbf{D}} = \mathrm{diag}(\hat{\mathbf{d}})$ and $\hat{\mathbf{d}}$ is the regularisation vector that minimises a specific risk function. Expressions are available for various risk functions (Selliah, 1964; Eaton & Olkin, 1987; Besson et al, 2020).

Shrinkage estimators of the third class form the largest group among the three classes and were first proposed by Ledoit & Wolf (2003b). These estimators are of the form $\widehat{\boldsymbol{\Sigma}} = \delta\mathbf{Q} + (1-\delta)\mathbf{S}$, where $\mathbf{Q}$ is a structured target covariance matrix, $\mathbf{S}$ is the SCM, and δ is the shrinkage parameter or *intensity*, found by minimising the selected risk function. The intensity can be found *analytically* (Ledoit & Wolf, 2004), or in a *data-driven* manner using, for example, cross-validation (Friedman, 1989; Warton, 2008), bootstrap,

or Markov Chain Monte Carlo (MCMC). The identity and scaled identity are the most commonly used target covariance matrices. Other typical shrinking targets are included in Schafer & Strimmer (2005). Selecting an appropriate shrinking target $\mathbf{Q}$ is crucial for the accuracy of the covariance estimate.

Among some of the most influential works from Ledoit and Wolf is the derivation of a simple lineal shrinkage formula for the calculation of the optimal value of the intensity parameter by minimising the risk associated with the Frobenius norm (Ledoit & Wolf, 2004), or their contribution with the approach for non-linear shrinkage of the eigenvalues of the SCM (Ledoit & Wolf 2012; 2015; 2021). The non-linear relation between the eigenvalues of the SCM and PCM motivated their research on non-linear methods, which make no structural assumption on the covariance matrix.

4.5.5 Banding

In naturally ordered data, such as time series, strong correlations tend to appear only between nearby variables. This means that the largest entries in a covariance matrix will be those closest to the main diagonal. *Banding* takes advantage of this covariance pattern by limiting the estimation to a number of diagonals. The banding estimator is given by

$$\widehat{\boldsymbol{\Sigma}} = \boldsymbol{\mathcal{B}}(\mathbf{S}) = \left[s_{ij}\mathbf{1}(|i-j| \leq l)\right] \tag{4.30}$$

where $\mathbf{1}(\mathcal{A})$ is the *indicator function* of subset $\mathcal{A}$, $\mathbf{S} = \left[s_{ij}\right]$ is the SCM, and l is the banding parameter indicating the number of diagonal pairs in the band. The banding function $\boldsymbol{\mathcal{B}}(\mathbf{S})$ sets to zero all entries in the SCM except those within the specified band.

Unlike shrinkage, banding affects both the eigenvalues and eigenvectors of $\widehat{\boldsymbol{\Sigma}}$. This method is *simple* and *computationally feasible*; only the calculation of the banding parameter can be demanding. The banding estimator is *well-conditioned* but *not positive-definite* necessarily. Banding the lower triangular matrix $\mathbf{T}$ of the modified Cholesky decomposition of the precision matrix, as in (4.16), is a way of ensuring positive-definiteness (Wu & Pourahmadi, 2003; Bickel & Levina, 2008a; Levina et al., 2008). Levina et al. (2008) propose a form of *adaptive banding* that incorporates a nested Lasso penalty to achieve a more natural form of banding. Another option to guarantee positive-definiteness is to use *tapering* (see next sub-section).

4.5.6 Tapering

Tapering is a form of smooth banding where the off-diagonal entries in the covariance matrix are weighted gradually towards zero. Like banding, this method requires the variables to be naturally ordered, it affects both the eigenvalues and eigenvectors, and is simple and computationally feasible. The tapering estimator is given by

$$\widehat{\boldsymbol{\Sigma}} = \mathbf{S} \circ \mathbf{W} = \left[s_{ij} w_{ij}\right] \tag{4.31}$$

where $\mathbf{W} = \left[w_{ij}\right]$ is the *tapering matrix* and the $\circ$ operator represents the entry-wise matrix multiplication or *Schur* product.

Compared to banding, a key advantage of tapering is that the estimator $\widehat{\boldsymbol{\Sigma}}$ is PD if the tapering matrix $\mathbf{W}$ is also PD. This is explained by the fact that the entry-wise product of two PD matrices is also PD. A tapering matrix is PD if its off-diagonal entries decrease monotonically towards zero. A usual choice for $\mathbf{W}$ is the *trapezoidal matrix*, characterised by unity weights up to diagonal l_h and zero weights from diagonal l, with a linear decay in between. The l value is the *tapering parameter*.

Examples of this approach can be found in Furrer & Bengtsson (2007), Bickel & Levina (2008a), and Cai et al (2010).

4.5.7 Thresholding

In many applications the data is not naturally ordered, and an estimation method capable of inducing sparsity in a way that is consistent with the covariance structure is needed. Thresholding is *permutation-invariant*, meaning that is not affected by the order of the variables. The thresholding estimator is given by

$$\widehat{\boldsymbol{\Sigma}} = \left[s_{ij} \mathbf{1}\left(\left|s_{ij}\right| \geq \lambda\right)\right] \tag{4.32}$$

where $\mathbf{1}(\mathcal{A})$ is the *indicator function* of subset $\mathcal{A}$, $\mathbf{S} = \left[s_{ij}\right]$ the SCM, and λ the thresholding parameter. The estimator sets to zero all entries in the SCM lower than λ. The method is suitable for the estimation of large covariance matrices, which tend to be sparse with an irregular pattern of zeroes.

The thresholding approach is *computationally efficient*, preserves *symmetry*, and affects both the eigenvalues and eigenvectors of the estimator, but does *not ensure positive definiteness*. Several authors have proposed algorithms for PD thresholding estimators using lasso penalised regressions. Rothman (2012) applies a logarithmic barrier to enforce PD. Xue et al. (2012) present an efficient alternating direction algorithm for computing the log-barrier estimator from Rothman (2012).

Different forms of thresholding are available for enforcing sparsity. *Universal thresholding* applies the same tuning parameter λ to all entries of the estimator, while *adaptive thresholding* applies a different threshold for each entry. The work from El Karoui (2008) and Bickel & Levina (2008b) are examples of universal thresholding using a *hard threshold*, like the one in (4.32). The individual thresholds in the adaptive thresholding approach depend on the variance of the individual entries of the SCM, the number of observations M, the number of variables N, and a tuning parameter found through cross-validation, as described in Cai & Liu (2011). Compared to universal thresholding,

adaptive thresholding leads to more natural sparsity patterns. A broader class of universal thresholding functions known as *generalised thresholding* and introduced by Rothman et al (2009) combine thresholding with shrinkage to widen the range of thresholding options. Common thresholding functions such as hard thresholding, soft thresholding, adaptive lasso, and SCAD are special cases covered in generalised thresholding.

4.5.8 Maximum Likelihood

Maximum likelihood estimators arrive at the covariance matrix by maximising the multivariate Gaussian PDF of M independent observations. Let $\mathbf{x} = \{x_1, x_2, \dots x_N\}$ be a vector of N Gaussian variables, $\boldsymbol{\mu} = \{E(x_1), E(x_2), \dots E(x_N)\}$ the mean vector, and $\boldsymbol{\Sigma}$ the covariance matrix. The corresponding multivariate Gaussian PDF for this single observation is given by

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N |\boldsymbol{\Sigma}|}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] \tag{4.33}$$

If now we consider M independent observations of vector $\mathbf{x}$, the PDF becomes

$$p(\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_M) = \frac{1}{\sqrt{(2\pi)^{MN} |\boldsymbol{\Sigma}|^M}} \exp\left[-\frac{1}{2}\sum_{m=1}^{M}(\mathbf{x}_m-\boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_m-\boldsymbol{\mu}_m)\right] \tag{4.34}$$

For the set of vector samples $\mathbf{x}_m$ $(m = 1,2 \dots M)$, the *maximum likelihood estimate* is the covariance matrix that maximises (4.34). The covariance matrix is unknown but constrained to a family of feasible $\boldsymbol{\Sigma}$ (Burg et al, 1982).

A more common version of (4.34) is the so-called *negative log-likelihood*.

$$L(\boldsymbol{\Sigma}, \mathbf{S}) = \ln|\boldsymbol{\Sigma}| + \mathrm{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S}) \tag{4.35}$$

One can easily arrive at (4.35) from (4.34) by taking the natural logarithm, removing the constant term, scaling by $M/2$, and applying the definition of matrix trace $\mathrm{tr}(\cdot)$ and sample covariance matrix $\mathbf{S}$ (Burg et al., 1982). Note that the expression assumes *zero-mean*. The covariance estimator $\widehat{\boldsymbol{\Sigma}}$ can be defined through parameterisation by also relying on available data and structural constraints imposed on $\boldsymbol{\Sigma}$. The covariance estimate is obtained by finding the optimal parameter vector that minimises $L(\boldsymbol{\Sigma}, \mathbf{S})$.

Examples of the application of maximum likelihood on covariance estimation include Won & Kim (2006), who use a constraint on the condition number, and Theiler (2012), who applies a leave-one-out cross-validation approach to separately compute a shrinkage covariance estimate and the combined likelihood from all observations from which he obtains the optimal shrinkage coefficient.

A more general form of the maximum likelihood estimator adds a penalty to the objective function $L(\boldsymbol{\Sigma}, \mathbf{S})$. For example, Huang et al. (2006) propose the reparameterising the

covariance matrix through modified Cholesky decomposition, to then apply a regression algorithm based on a penalised negative log-likelihood with ridge and lasso penalties. Rothman et al. (2008) present a similar approach with lasso penalty for estimating the precision matrix. Witten & Tibshirani (2009) propose a general penalised log-likelihood method for various penalty scenarios (least-squared, ridge, lasso, and elastic net) using the Scout Procedure.

4.5.9 Sparse Gaussian Graphical Models

Sparse Gaussian Graphical Models (SGGM) analyse the conditional dependence among pairs of random variables to estimate a sparse version of the precision matrix (Huang, 2016). The justification for not regularising the covariance matrix directly is that the sparsity of the precision matrix relates to the conditional dependence of pairs of variables given the rest (Pourahmadi, 2013). In particular, a zero entry in the precision matrix indicates that the corresponding variable pair (i, j) is *conditionally independent*.

Covariance matrices estimated by finding zeros in its inverse are called *covariance selection models*, an idea first introduced by Dempster (1972). This approach is motivated by the fact that in some problems the precision matrix has many zeroes in its off-diagonal entries.

In SGGM, variables are represented as *nodes,* joined by edges when the linked pair is conditionally dependent. A SGGM performs *nodewise regression* to find what entries of the $[N, N]$ precision matrix are zero. This is done by fitting N separate regression models, one per node, where each variable is regressed from the rest. Therefore, precision matrix estimation is reduced to a sequence of N regression models of varying order. The idea of nodewise regression with lasso penalty to induce sparsity was first proposed by Meinshausen and Buhlmann (2006).

Penalised regression methods for sparse estimation of the precision matrix are central to SGGM. The application of penalised likelihood for SGGM was proposed by Yuan and Lin (2007), but their method had limited applicability ($N < 10$). The fastest SGGM algorithm is the Graphical Lasso or Glasso (see next sub-section).

4.5.10 The Graphical Lasso

The Graphical Lasso or Glasso is a SGGM that induces sparsity by fitting the lasso regression model to each variable using the rest of variables as predictors. The method was proposed by Friedman et al (2008) and is based on the nodewise regression approach from Meinshausen & Bauhlmann (2006), but improves over it by applying a *coordinate descent algorithm*, a fast optimisation technique due to Banerjee et al. (2008).

The method estimates the precision matrix by solving a lasso regression problem column-by-column. The optimisation process is repeated until convergence. The algorithm

aims at maximising the log-likelihood with lasso penalty through its subgradient equation. The column-wise regression process is made possible by partitioning the matrices in the subgradient equation (precision estimate $\widehat{\mathbf{\Sigma}}^{-1}$, penalty $\mathbf{\Gamma}$, sample covariance $\mathbf{S}$) into four submatrices. From a graphical perspective, predicting a column of the precision matrix estimate is equivalent to determining the edges between one node and the rest.

The Glasso is more computationally efficient than competing algorithms and produces a positive-definite estimation of the precision matrix. However, cross-validation may be used to choose the tuning parameter, which will increase the computational cost of the algorithm.

4.6 Energy Detector

The simplest of the Neyman-Pearson detectors is the *energy detector*. Unlike the estimator-correlator, presented in sections 4.7 and 4.8, in the energy detector the target signal and noise do not need to be known in advance, and only their variances are needed for the algorithm. Still, some knowledge about the time and frequency response of the target signal and noise is beneficial, as it can be used to improve the performance of the energy detector through resampling, filtering, and window length selection. In Section 4.10 we demonstrate how these three solutions can be used to improve the performance of a NP detector, and in Section 4.11 we present an improved version of the energy detector that incorporates filtering. Below is presented the formulation of the standard energy detector, following Kay's (1998) derivation, with various aspects of its performance discussed at the end.

In the energy detector, the signal and the noise are assumed to be zero mean, *wide sense stationary* (WSS) white Gaussian random processes (WGS, WGN) of *independent and identically distributed* (IID) variables with variances σ_s^2 and σ_w^2. Formally, this is expressed as $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \sigma_s^2\mathbf{I})$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2\mathbf{I})$, where $\mathbf{s}$ and $\mathbf{w}$ are the signal and noise vectors, $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ is the multivariate normal distribution with *mean vector* $\boldsymbol{\mu}$ and *covariance matrix* $\mathbf{\Sigma}$, $\mathbf{0}$ is a vector of zeroes (i.e., zero mean), and $\mathbf{I}$ is the identity matrix. In a WSS process the mean and variance are time-independent. The IID condition indicates that the samples are uncorrelated and have the same PDF.

From the above, it follows that the distributions of a zero-mean vector $\mathbf{x} = \{x_1, \dots x_N\}^T$ for each hypothesis are

$$\mathcal{H}_0 : \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2\mathbf{I}) \tag{4.36}$$

$$\mathcal{H}_1 : \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma_t^2\mathbf{I}) \tag{4.37}$$

where σ_w^2 and $\sigma_t^2 = \sigma_s^2 + \sigma_w^2$ are the variances of $\mathbf{x}$ under the null hypothesis $\mathcal{H}_0$ (noise only) and alternative hypothesis $\mathcal{H}_1$ (signal plus noise).

The multivariate Gaussian probability distribution for the column vector $\mathbf{x}$ is given by

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N \det \boldsymbol{\Sigma}}} \exp\left[-\frac{1}{2}\mathbf{x}^T \boldsymbol{\Sigma}^{-1}\mathbf{x}\right] \tag{4.38}$$

The critical region is determined by the likelihood ratio test in (4.3), which depends on the probability distributions of $\mathbf{x}$ for the alternative and null hypotheses, given in (4.36) and (4.37). By combining the equations, we see that the NP detector decides $\mathcal{H}_1$ if

$$L(\mathbf{x}) = \frac{\mathcal{N}(\mathbf{0}, \sigma_s^2\mathbf{I} + \sigma_w^2\mathbf{I})}{\mathcal{N}(\mathbf{0}, \sigma_w^2\mathbf{I})} > \gamma \tag{4.39}$$

If we now apply (4.38) to (4.39), the critical region can be rewritten as

$$\begin{aligned} L(\mathbf{x}) &= \frac{\frac{1}{[2\pi\sigma_t^2]^{N/2}} \exp\left[-\frac{1}{2\sigma_t^2}\sum_{n=0}^{N-1} x^2[n]\right]}{\frac{1}{(2\pi\sigma_w^2)^{N/2}} \exp\left[-\frac{1}{2\sigma_w^2}\sum_{n=0}^{N-1} x^2[n]\right]} \\ &= \left(\frac{\sigma_t^2}{\sigma_w^2}\right)^{-\frac{N}{2}} \exp\left[-\frac{1}{2}\left(\frac{1}{\sigma_t^2} - \frac{1}{\sigma_w^2}\right)\sum_{n=0}^{N-1} x^2[n]\right] > \gamma \end{aligned} \tag{4.40}$$

By taking the Napierian logarithm at both sides of $L(\mathbf{x}) > \gamma$ and rearranging terms we arrive at the expression for the test statistic

$$T(\mathbf{x}) = \sum_{n=0}^{N-1} x^2[n] > \gamma' \tag{4.41}$$

The updated threshold γ' is a function of the previous threshold γ. The expression for γ' is irrelevant for the calculations and is not included here. The test statistic $T(\mathbf{x})$ is an estimate of the *energy* of the zero-mean vector $\mathbf{x}$, hence the name for the detector. Alternatively, the *variance* can be used as the test statistic by simply dividing by N at both sides of the inequality.

$T(\mathbf{x})$ is a statistic with a central Chi-squared PDF of N variables or *degrees of freedom* (DOF). Formally, $T(\mathbf{x}) \sim \sigma_i^2 \mathcal{X}_N^2$, where σ_i^2 is the variance of $\mathbf{x}$ under hypothesis i and $\mathcal{X}_N^2$ is the central Chi-squared distribution with N DOF. If the PDF of $\mathcal{X}_N^2$ is expressed as $p_{\chi_N^2}(T)$, then the PDFs of $T(\mathbf{x})$ for the two hypotheses are

$$p(T(\mathbf{x});\, \mathcal{H}_0) = \sigma_w^{-2} p_{\chi_N^2}(\sigma_w^{-2} T) \tag{4.42}$$

$$p(T(\mathbf{x});\, \mathcal{H}_1) = \sigma_t^{-2} p_{\chi_N^2}(\sigma_t^{-2} T) \tag{4.43}$$

Note that $T(\mathbf{x}) \sim \sigma_i^2 \mathcal{X}_N^2$ is equivalent to $T(\mathbf{x})\sigma_i^{-2} \sim \mathcal{X}_N^2$. As seen in the Section 4.2, the performance of a detector is characterised by the probability functions of false alarm $P_{FA}(T)$ and detection $P_D(T)$. If $Q_{\chi_N^2}(y)$ represents the right-tail probability of the central

Chi-squared PDF of N-DOF for the statistic $y = \sum_{m=0}^{N} {x_m}^2$ with $x_m \sim \mathcal{N}(0,1)$, P_{FA} and P_D for a threshold γ' are calculated from $Q_{\chi_N^2}$ as follows

$$P_{FA} = \Pr\{T(\mathbf{x}) > \gamma';\ \mathcal{H}_0\} = Q_{\chi_N^2}(\sigma_w^{-2}\gamma') \tag{4.44}$$

$$P_D = \Pr\{T(\mathbf{x}) > \gamma';\ \mathcal{H}_1\} = Q_{\chi_N^2}(\sigma_t^{-2}\gamma') \tag{4.45}$$

Due to the large audio sampling rates and signal durations typically found in underwater sound recordings (>24 kHz, 0-0.1 s), the number of variables in $\mathbf{x}$ tends to be large. The central limit theorem can be used to simplify (4.44) and (4.45), rewriting $P_{D,FA}(T = \gamma')$ as a function of the right-tail normal distribution Q. Given that the *mean* and *variance* of the N-DOF central Chi-squared statistic $y = \sum_{m=0}^{N} {x_m}^2$ are $\sigma^2 N$ and $\sigma^4 2N$, where $x_m \sim \mathcal{N}(0, \sigma^2)$ and σ^2 is the variance of $\mathbf{x}$, (4.44) and (4.45) can be approximated to

$$P_{FA} = \Pr\{T(\mathbf{x}) > \gamma';\ \mathcal{H}_0\} = Q\left(\frac{\gamma' - \sigma_w^2 N}{\sigma_w^2\sqrt{2N}}\right) \tag{4.46}$$

$$P_D = \Pr\{T(\mathbf{x}) > \gamma';\ \mathcal{H}_1\} = Q\left(\frac{\gamma' - \sigma_t^2 N}{\sigma_t^2\sqrt{2N}}\right) \tag{4.47}$$

The signal-to-noise ratio SNR in decibels is given by $10\log(\sigma_s^2/\sigma_w^2)$. From (4.44) and (4.45) and assuming that N and σ_w^2 are fixed, $P_{FA}(T)$ does not change with SNR, while the $P_D(T)$ curve is scaled ("expands") as SNR increases. The result is an increase in γ_D' for higher values of SNR. Also, as σ_w^2 increases so do the thresholds $\gamma_{D,FA}'$, producing an equal "expansion" in the $P_{FA}(T)$ and $P_D(T)$ curves. Therefore, an increase in σ_w^2 has no effect on the detection performance, since the relation between P_{FA} and P_D remains unaltered for any given SNR.

A simple way of visualising the behaviour of P_{FA} and P_D with SNR is to represent the dependence of γ' with SNR for fixed target probabilities of false alarm α and detection $(1-\beta)$. Figure 4.5 shows this dependence for $\alpha, \beta = 0.5$, where β is the *target probability of miss*. As expected, $\log {\gamma_{FA}}'$ is constant while $\log {\gamma_D}'$ increases linearly with SNR.

It can be concluded that the ability of the *energy detector* to map a vector $\mathbf{x}$ of length N into the correct decision depends exclusively on the ratio between signal and noise variances. Figure 4.6 shows the dependence of P_D with SNR for four target probabilities of false alarm and $N = 100$. Figure 4.7 represents the receiver operating characteristic for four SNR values and $N = 100$. A higher value of P_D/P_{FA} is associated with a better detection performance. Both figures show the expected increase of P_D/P_{FA} with SNR.

Now, the question of mapping an observation with the energy detector still remains. Given a target probability of false alarm α and assuming that the noise variance σ_w^2 is known, the threshold can be obtained *analytically* from $P_{FA}(T)$ as $\gamma' = \sigma_w^2 Q_{\chi_N^2}^{-1}(\alpha)$, or numerically by locating the target probability of false alarm α in $P_{FA}(T)$. The observation $\mathbf{x}$ is classified as detection if $T(\mathbf{x}) > \gamma'$.

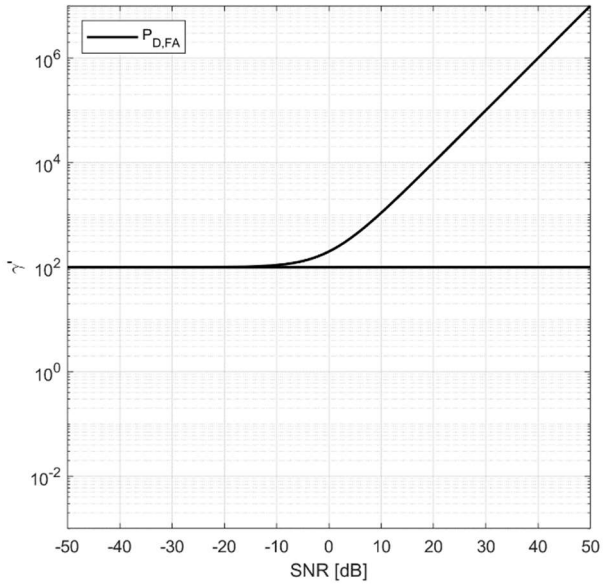

Figure 4.5 Detection threshold of *energy detector* as a function of the signal-to-noise ratio for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. Top and bottom (flat) curves correspond to $P_D$and P_{FA}, respectively. $N = 100$, $\sigma_w^2 = 1$.

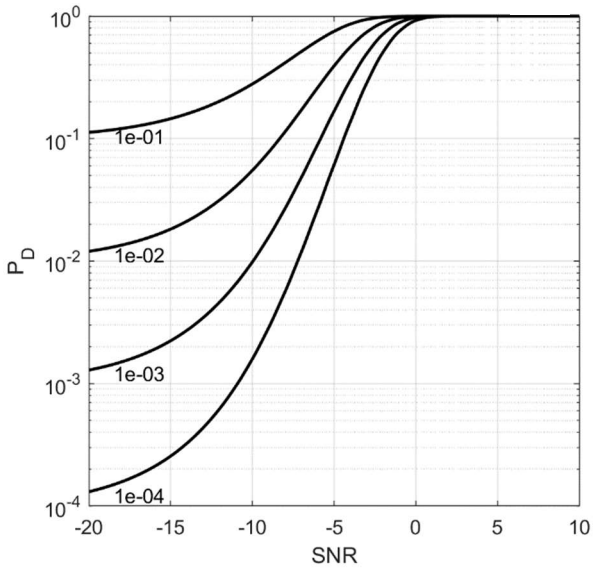

Figure 4.6 Probability of detection P_D of the *energy detector* as a function of the signal-to-noise ratio SNR for four target probability of false alarms α. The values of α are shown below their respective curves. $N = 100$, $\sigma_w^2 = 1$.

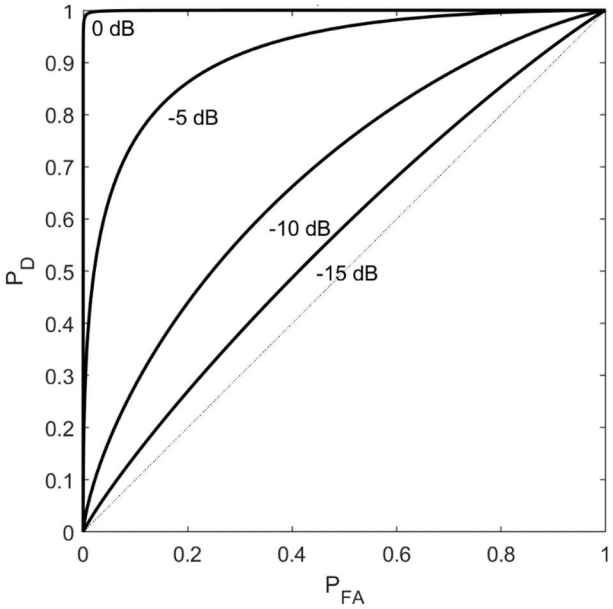

Figure 4.7 Receiver operating characteristic ROC of the *energy detector* for four signal-to-noise ratios SNR. The values of SNR are shown by their respective curves. $N = 100$, $\sigma_w^2 = 1$.

4.7 Estimator-Correlator with White Gaussian Noise

The estimator-correlator is a more general form of energy detector where the covariance of the target signal is known. It is assumed that the noise is a zero mean, WSS, white Gaussian random process of IID variables with variance σ_w^2, and the signal is a zero mean, Gaussian random process with covariance matrix $\mathbf{\Sigma}_s$. The probability distributions for the signal $\mathbf{s}$ and noise $\mathbf{w}$ are $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_s)$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2\mathbf{I})$, where $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ is a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$. The expression for $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ is given by (4.38).

From the assumed probability distributions of the signal $\mathbf{s}$ and the noise $\mathbf{w}$, it follows that the distributions of $\mathbf{x}$ for each hypothesis are

$$\mathcal{H}_0 \colon \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma_w^2\mathbf{I}) \tag{4.48}$$

$$\mathcal{H}_1 \colon \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_s + \sigma_w^2\mathbf{I}) \tag{4.49}$$

Before continuing with the definition of the problem, we should outline how the covariance matrix of the target signal $\mathbf{\Sigma}_s$ is calculated in practice. It is, in fact, the *normalised covariance matrix* $\bar{\mathbf{\Sigma}}_s$ that we aim to find in the first place. $\mathbf{\Sigma}_s$ is related to $\bar{\mathbf{\Sigma}}_s$ by

$$\mathbf{\Sigma}_s = \sigma_s^2 \bar{\mathbf{\Sigma}}_s \tag{4.50}$$

Hence, $\mathbf{\Sigma}_s$ can be easily obtained for an observation $\mathbf{x}$ if the signal variance σ_s^2 is available. Assuming that the noise variance σ_w^2 is known, the signal variance σ_s^2 of an observation can be estimated by subtracting σ_w^2 from the signal-plus-noise variance σ_t^2.

In the algorithms presented in Section 4.13, the noise variance σ_w^2 is obtained from "quiet" inter-pulse periods within the audio file.

The elements in the diagonal of a covariance matrix are the variances of the corresponding variables. The variance of the target signal is the expectation of the diagonal elements in the covariance matrix, that is, $\sigma_s^2 = E(\text{diag}\{\boldsymbol{\Sigma}_s\})$. Therefore, $\sigma_s^2 = 1$ for $\breve{\boldsymbol{\Sigma}}_s$.

The normalised covariance matrix $\breve{\boldsymbol{\Sigma}}_s$ is calculated from a number of *zero-mean*, "noise-free" ($SNR \gg 10$ dB) N-variable observations of the target signal using an appropriate covariance estimation method (see Section 4.5). To arrive at $\breve{\boldsymbol{\Sigma}}_s$, all observations are normalised by their standard deviation σ_t^2.

Let's continue with the definition of the detection problem for the estimator-correlator in WGN. The critical region is determined by the likelihood ratio test in (4.3). The Neyman-Pearson detector decides $\mathcal{H}_1$ if

$$L(\mathbf{x}) = \frac{\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_s + \sigma_w^2\mathbf{I})}{\mathcal{N}(\mathbf{0}, \sigma_w^2\mathbf{I})} > \gamma \tag{4.51}$$

After applying the expression of the multivariate Gaussian distribution in (4.38) to (4.51), we have

$$\begin{aligned} L(\mathbf{x}) &= \frac{\frac{1}{[(2\pi)^N \det(\boldsymbol{\Sigma}_s + \sigma_w^2\mathbf{I})]^{1/2}} \exp\left[-\frac{1}{2}\mathbf{x}^T(\boldsymbol{\Sigma}_s + \sigma_w^2\mathbf{I})^{-1}\mathbf{x}\right]}{\frac{1}{(2\pi\sigma_w^2)^{N/2}} \exp\left[-\frac{1}{2}\mathbf{x}^T\left(\frac{\mathbf{I}}{\sigma_w^2}\right)\mathbf{x}\right]} \\ &= \frac{\sigma_w^N}{[\det(\boldsymbol{\Sigma}_s + \sigma_w^2\mathbf{I})]^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{x}^T\left[(\boldsymbol{\Sigma}_s + \sigma_w^2\mathbf{I})^{-1} - \frac{\mathbf{I}}{\sigma_w^2}\right]\mathbf{x}\right\} > \gamma \end{aligned} \tag{4.52}$$

By taking the Napierian logarithm at both sides of $L(\mathbf{x}) > \gamma$ and transferring the data-independent terms to the right side of the inequality, we arrive at

$$T(\mathbf{x}) = \mathbf{x}^T\left[(\boldsymbol{\Sigma}_s + \sigma_w^2\mathbf{I})^{-1} - \frac{\mathbf{I}}{\sigma_w^2}\right]\mathbf{x} > \gamma' \tag{4.53}$$

If we apply the matrix inversion lemma to (4.53) as described in Kay (1998, p. 145), we arrive at the final expression for the test statistic

$$T(\mathbf{x}) = \mathbf{x}^T\hat{\mathbf{s}} = \sum_{n=0}^{N-1} x[n]\hat{s}[n] > \gamma' \tag{4.54}$$

$$\hat{\mathbf{s}} = \boldsymbol{\Sigma}_s(\boldsymbol{\Sigma}_s + \sigma_w^2\mathbf{I})^{-1}\mathbf{x} \tag{4.55}$$

The expression for the updated threshold γ' is irrelevant for the calculations and is not included here. The test statistic $T(\mathbf{x})$ represents the *correlation* of data vector $\mathbf{x}$ with an *estimate* of the signal $\hat{\mathbf{s}}$, hence the name *estimator-correlator*.

The test statistic $T(\mathbf{x})$ is a quadratic sum and as such is a non-Gaussian process. $T(\mathbf{x})$ is a sum of Chi-squared random variables of one degree of freedom scaled by different weights (i.e., $x[n]\hat{s}[n]\sim\mathcal{X}_1^2$). Unlike the energy detector, the test statistic of the estimator-correlator cannot be expressed as a scaled Chi-squared of N degrees of freedom. The *canonical form* is a practical, more convenient expression for $T(\mathbf{x})$, which can be used to derive the formulas for computing the PDFs for the two hypotheses. The canonical form of the test statistic is (Kay, 1998)

$$T(\mathbf{y}) = \mathbf{y}^T\boldsymbol{\Lambda}_s(\boldsymbol{\Lambda}_s + \sigma_w^2\mathbf{I})^{-1}\mathbf{y} \tag{4.56}$$

$$\mathbf{y} = \mathbf{V}_s^T\mathbf{x} \tag{4.57}$$

where $\boldsymbol{\Lambda}_s$ is the *diagonal matrix* or matrix of eigenvalues of $\boldsymbol{\Sigma}_s$, $\mathbf{V}_s$ is the *modal matrix* or matrix of eigenvectors of $\boldsymbol{\Sigma}_s$, and $\mathbf{y}$ is the *decorrelated* version of $\mathbf{x}$ where each variable is independent of the rest. The columns $\mathbf{v}_{s,j}$ in $\mathbf{V}_s = \left[\mathbf{v}_{s,0}\ \mathbf{v}_{s,1}\ \ldots\ \mathbf{v}_{s,N-1}\right]$ are the *eigenvectors* of $\boldsymbol{\Sigma}_s$. The diagonal elements $\lambda_{s,j}$ in $\boldsymbol{\Lambda}_s = \mathrm{diag}\left(\lambda_{s,0}, \lambda_{s,1}, \ldots \lambda_{s,N-1}\right)$ are the *eigenvalues* of the eigenvectors in $\mathbf{V}_s$.

The diagonal and modal matrices are obtained through eigendecomposition of the covariance matrix $\boldsymbol{\Sigma}_s$ using the expression $\boldsymbol{\Lambda}_s = \mathbf{V}_s^T\boldsymbol{\Sigma}_s\mathbf{V}_s$. Since $\boldsymbol{\Sigma}_s$ is symmetric, the eigenvalues $\lambda_{s,j}$ are *real*, the eigenvectors $\mathbf{v}_j$ are *orthonormal*, and the modal matrix $\mathbf{V}_s$ is *orthogonal*. The inverse of an *orthogonal* matrix is equal to its transpose ($\mathbf{V}_s^{-1} = \mathbf{V}_s^T$). The orthogonality of $\mathbf{V}_s$ was applied to the standard form of the test statistic in (4.54) to derive the canonical expression in (4.56).

It can be easily demonstrated by eigendecomposition that if $\breve{\boldsymbol{\Lambda}}_s$ and $\breve{\mathbf{V}}_s$ are the diagonal and modal matrices of the normalised covariance matrix $\breve{\boldsymbol{\Sigma}}_s$, then $\boldsymbol{\Lambda}_s = \sigma_s^2\breve{\boldsymbol{\Lambda}}_s$ and $\mathbf{V}_s = \breve{\mathbf{V}}_s$. The direct implication of these two equivalences is that the test statistic $T(\mathbf{y})$ in (4.56) can be computed if $\breve{\mathbf{C}}_s$ and σ_s^2 are known. The normalised covariance matrix $\breve{\mathbf{C}}_s$ needs to be calculated only once, and the approximate signal variance σ_s^2 can be obtained from the data vector $\mathbf{x}$ if the noise variance σ_w^2 is known.

Making use of the diagonal character of $\boldsymbol{\Lambda}_s$ and $\sigma_w^2\mathbf{I}$, the canonical form of the test statistic can be rewritten as

$$T(\mathbf{y}) = \sum_{n=0}^{N-1}\frac{\lambda_{sn}}{\lambda_{sn} + \sigma_w^2}y^2[n] \tag{4.58}$$

The term $\lambda_{sn}(\lambda_{sn} + \sigma_w^2)^{-1}$ represents the weights in the transformed space. Each weighting coefficient indicates how much larger than the noise is the signal along the direction of eigenvector $\mathbf{v}_n$. In this canonical form, the data vector $\mathbf{x}$ is linearly transformed into $\mathbf{y}$ (decorrelation) and sent through a *weighted* energy detector. The vector $\mathbf{y}$ consists of uncorrelated random variables of different variances, which result in different weights being applied to $y^2[n]$.

The diagonal matrix of the normalised covariance is $\breve{\mathbf{\Lambda}}_s = \mathrm{diag}\left(\breve{\lambda}_{s,0}, \breve{\lambda}_{s,1}, \ldots \breve{\lambda}_{s,N-1}\right)$. Since $\mathbf{\Lambda}_s = \sigma_s^2 \breve{\mathbf{\Lambda}}_s$, we have that $\mathbf{\Lambda}_s = \mathrm{diag}\left(\sigma_s^2 \breve{\lambda}_{s0}, \sigma_s^2 \breve{\lambda}_{s1}, \ldots \sigma_s^2 \breve{\lambda}_{sN-1}\right)$. Hence, the test statistic can be conveniently expressed as a function of the eigenvalues from the normalised covariance matrix $\breve{\lambda}_{sn}$ and the signal-to-noise ratio σ_s^2/σ_w^2

$$T(\mathbf{y}) = \sum_{n=0}^{N-1} \frac{\breve{\lambda}_{sn}(\sigma_s^2/\sigma_w^2)}{\breve{\lambda}_{sn}(\sigma_s^2/\sigma_w^2) + 1} y^2[n] \tag{4.59}$$

The probability density functions of the test statistic for the two hypotheses can be calculated numerically with the following expressions. These PDFs are derived from the canonical form of $T(\mathbf{x})$ as described in Kay (1998, p. 183).

$$p(T(\mathbf{y});\, \mathcal{H}_0) = \mathcal{F}_{-t}^{-1}\left\{\prod_{n=0}^{N-1} \frac{1}{\sqrt{1 - 2j\alpha_n\omega}}\right\} \tag{4.60}$$

$$\alpha_n = \frac{\lambda_{sn}\sigma_w^2}{\lambda_{sn} + \sigma_w^2} \tag{4.61}$$

$$p(T(\mathbf{y});\, \mathcal{H}_1) = \mathcal{F}_{-t}^{-1}\left\{\prod_{n=0}^{N-1} \frac{1}{\sqrt{1 - 2j\lambda_{sn}\omega}}\right\} \tag{4.62}$$

where $\mathcal{F}_{-t}^{-1}\{f(\omega)\}$ is the inverse Fourier transform evaluated at $-t$

$$\mathcal{F}_{-t}^{-1}\{x(t)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} x(t) e^{-j\omega t} d\omega \tag{4.63}$$

By inserting (4.60) and (4.62) into (4.5) and (4.6), we arrive at the expressions for the right-tail probabilities of false alarm and detection

$$P_{FA} = \Pr\{T(\mathbf{y}) > \gamma'';\, \mathcal{H}_0\} = \int_{\gamma''}^{\infty}\int_{-\infty}^{\infty} \frac{1}{2\pi}\prod_{n=0}^{N-1} \frac{1}{\sqrt{1 - 2j\alpha_n\omega}} e^{-j\omega t} d\omega\, dt \tag{4.64}$$

$$P_D = \Pr\{T(\mathbf{y}) > \gamma'';\, \mathcal{H}_1\} = \int_{\gamma''}^{\infty}\int_{-\infty}^{\infty} \frac{1}{2\pi}\prod_{n=0}^{N-1} \frac{1}{\sqrt{1 - 2j\lambda_{sn}\omega}} e^{-j\omega t} d\omega\, dt \tag{4.65}$$

Interpreting the dependence of $P_{FA}(T)$ and $P_D(T)$ on SNR directly from (4.64) and (4.65) is not straightforward. Still, the behaviour of $P_{FA}(T)$ and $P_D(T)$ with SNR can be analysed through the dependence of thresholds $\gamma'_{D,FA}$ on SNR for fixed target probabilities of false alarm α and detection $(1-\beta)$. Figure 4.8 shows this dependence for $\alpha, \beta = 0.5$. Similar to the energy detector (see Figure 4.5), the threshold curves $\gamma'_{D,FA}(SNR)$ converge at low SNR values and diverge at high SNR values where $\gamma'_{D,FA}$ reaches a plateau. Another aspect in common with the energy detector is that as the noise variance σ_w^2 increases, the threshold $\gamma'_{D,FA}$ increases by the same amount. This scaling effect has no

impact on the detection performance, since the relation between P_{FA} and P_D remains the same for any given SNR and threshold $\gamma'_{D,FA}$.

There are two major differences between figures Figure 4.8 and Figure 4.5: (1) the threshold $\log \gamma'_{D,FA}$ decreases linearly with SNR for low SNR values, and (2) the type of target signal has an effect on P_{FA}. Only the second observation has an impact on the detection performance. The larger the difference between the target signal and the WGN, the greater the ability of the detector to map the data vector $\mathbf{x}$ into the correct decision. In particular, a strongly "coloured" target signal will reduce the P_{FA} for a given P_D, thus improving the detection performance. Interestingly, the shape of the $P_D(T)$ curve changes with the type of target signal, but this has a minor effect on γ' for $\beta \approx 0.5$ (Figure 4.9, right).

It can be concluded that the estimated performance of the estimator-correlator depends exclusively on the SNR and on how different the target signal is from WGN. Figure 4.10 shows the dependence of P_D with SNR for four target probabilities of false alarm and $N = 100$, for the case of a WGS filtered in one-octave band centred at half the Nyquist frequency. In Figure 4.11, the corresponding receiver operating characteristic is represented for four SNR values. Both figures show the expected increase in P_D/P_{FA} with SNR. The detection performance improves with respect to Figure 4.6 and Figure 4.7.

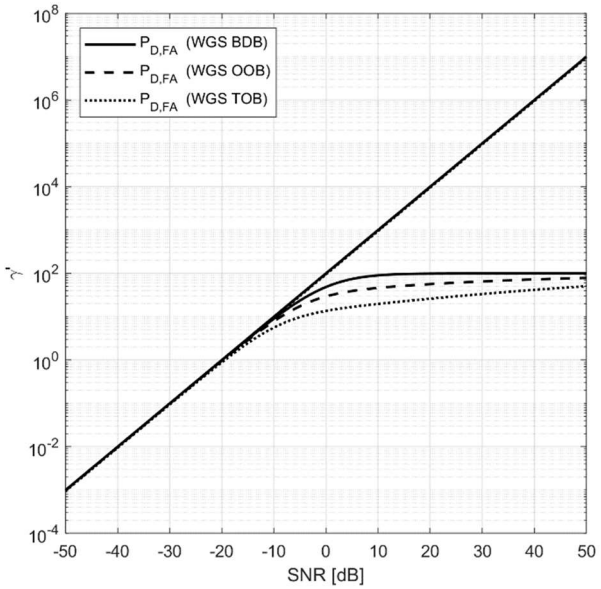

Figure 4.8 Detection threshold of *estimator-correlator* in WGN as a function of the signal-to-noise ratio for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. Three signals are represented: broadband WGN (solid lines), one-octave filtered WGN (dashed lines) and third-octave WGN (dotted lines). Top (diagonal) and bottom (flattening) curves correspond to $P_D$and P_{FA}, respectively. Note that the three $\gamma_D(SNR)$ curves are coincident. $N = 100$, $\sigma_w^2 = 1$.

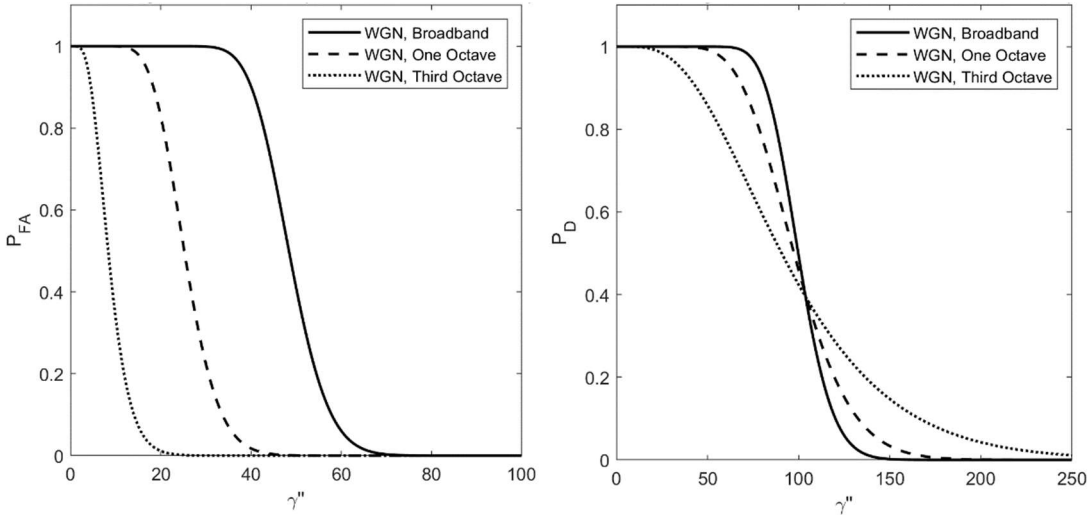

Figure 4.9 Right-tail probability of false alarm (left) and detection (right) as a function of the threshold γ' for three types of target signal: white Gaussian noise (solid line), one octave band-filtered white Gaussian noise (dashed line), one third-octave bandpass-filtered white Gaussian noise (dotted line). $N = 100$, $\sigma_w^2 = 1$, $SNR = 0$ dB.

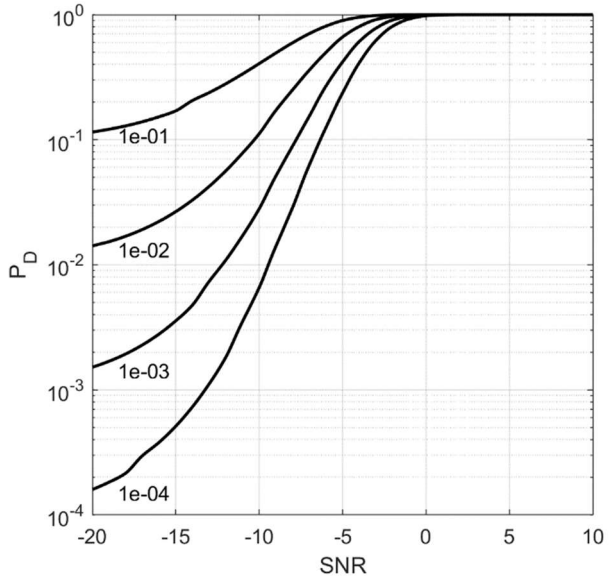

Figure 4.10 Probability of detection P_D of the *estimator-correlator* in WGN as a function of the signal-to-noise ratio for four target probabilities of false alarm α. The values of α are shown below their respective curves. One-octave filtered WGS. $N = 100$, $\sigma_w^2 = 1$.

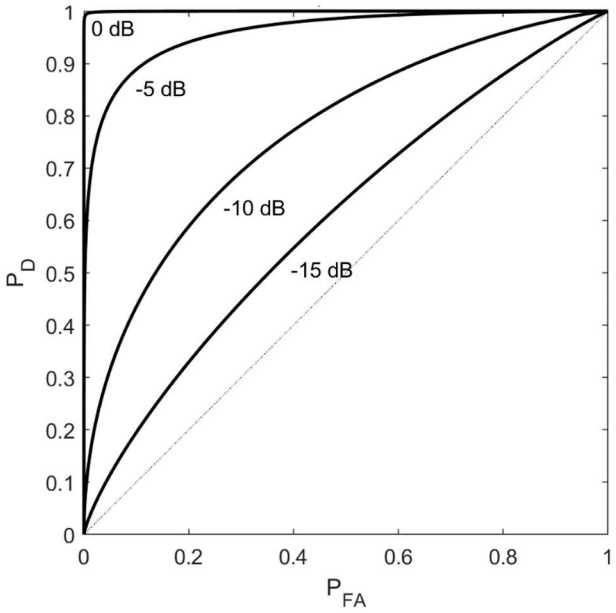

Figure 4.11 Receiver operating characteristic ROC of the *estimator-correlator* in WGN for four signal-to-noise ratios. The values of SNR are shown below their respective curves. One-octave filtered WGS. $N = 100$, $\sigma_w^2 = 1$.

4.8 Estimator-Correlator with "Coloured" Gaussian Noise

The estimator-correlator from the previous section assumed white Gaussian noise. If the background noise has a "coloured" frequency spectrum that considerably deviates from a flat response, the white noise assumption will not be longer applicable. In such scenario, using the detector from Section 4.7 will tend to produce high false alarm rates (Barthelemy & Willet, 1993). In Section 4.9 we analyse the behaviour of all three NP detectors with target and non-target signals and noise, and prove how under typical background noise conditions and low SNR the estimator-correlator in WGN tends to produce high false-alarm rates.

Improving the performance of the estimator-correlator in the presence of "coloured" or *correlated* Gaussian noise (CGN) requires the addition of *pre-whitening*. The pre-whitening operation is easily integrated in the formulation of the estimator-correlator by replacing the noise variance σ_w^2 with the covariance matrix of the correlated noise $\mathbf{\Sigma}_w$ before the likelihood ratio test. The pre-whitening consists of two stages (Stergiopoulos, 2001): (1) a *pre-whitening filter* that applies a linear transform to the data vector $\mathbf{x}$ to compensate for the response of the CGN, and (2) a *signal correction* stage that corrects the target signal alone.

In this general version of the estimator-correlator, the covariance matrices of the target signal and background noise are known. The target signal and noise are assumed to be zero mean, Gaussian random processes with covariance matrices $\mathbf{\Sigma}_s$ and $\mathbf{\Sigma}_w$. Therefore,

the probability distributions for the signal $\mathbf{s}$ and noise $\mathbf{w}$ are $\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_s)$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_w)$. The expression for the multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is given by (4.38). The discussion from previous section about the covariance matrix of the target signal and its calculation is applicable to the covariance matrix of the correlated noise.

From the above, it follows that the distributions of $\mathbf{x}$ for each hypothesis are

$$\mathcal{H}_0 \colon \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_w) \tag{4.66}$$

$$\mathcal{H}_1 \colon \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_w) \tag{4.67}$$

The Neyman-Pearson detector decides $\mathcal{H}_1$ if

$$L(\mathbf{x}) = \frac{\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_w)}{\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_w)} > \gamma \tag{4.68}$$

By applying the expression of the multivariate Gaussian distribution, we arrive at

$$\begin{aligned} L(\mathbf{x}) &= \frac{\frac{1}{[(2\pi)^N \det(\boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_w)]^{1/2}} \exp\left[-\frac{1}{2}\mathbf{x}^T(\boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_w)^{-1}\mathbf{x}\right]}{\frac{1}{[(2\pi)^N \det(\boldsymbol{\Sigma}_w)]^{1/2}} \exp\left[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}_w^{-1}\mathbf{x}\right]} \\ &= \frac{[(2\pi)^N \det(\boldsymbol{\Sigma}_w)]^{1/2}}{[\det(\boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_w)]^{1/2}} \exp\left\{-\frac{1}{2}\mathbf{x}^T[(\boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_w)^{-1} - \boldsymbol{\Sigma}_w^{-1}]\mathbf{x}\right\} > \gamma \end{aligned} \tag{4.69}$$

If we now take the Napierian logarithm at both sides of $L(\mathbf{x}) > \gamma$ and transfer the data-independent terms to the right side of the inequality, we arrive at

$$T(\mathbf{x}) = \mathbf{x}^T[(\boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_w)^{-1} - \boldsymbol{\Sigma}_w^{-1}]\mathbf{x} > \gamma' \tag{4.70}$$

By applying the matrix inversion lemma to (4.70) as described in Kay (1998, p. 145), we arrive at the final expression for the test statistic

$$T(\mathbf{x}) = \mathbf{x}^T\boldsymbol{\Sigma}_w^{-1}\hat{\mathbf{s}} > \gamma' \tag{4.71}$$

$$\hat{\mathbf{s}} = \boldsymbol{\Sigma}_s(\boldsymbol{\Sigma}_s + \boldsymbol{\Sigma}_w)^{-1}\mathbf{x} \tag{4.72}$$

The test statistic $T(\mathbf{x})$ represents the *correlation* of data vector $\mathbf{x}$ with a *pre-whitened* version of the *estimate* of the signal $\hat{\mathbf{s}}$. The term $\boldsymbol{\Sigma}_w^{-1}$ is the *pre-whitener* and $\boldsymbol{\Sigma}_w^{-1}\hat{\mathbf{s}}$ is the pre-whitened estimate of the signal.

The *canonical form* of (4.71), necessary to derive the formulas for computing the PDFs of the test statistic for the two hypotheses, can be obtained in a way similar to (4.56) using the orthogonality property of the modal matrix of a symmetric matrix. Kay (1998, p. 179) explains how to arrive at the canonical form by diagonalising $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_w$ in a single linear transformation. $T(\mathbf{x})$ is first expressed as a function of matrices $\mathbf{A}$ and $\mathbf{B}$

$$T(\mathbf{x}) = \mathbf{x}^T \mathbf{A}\mathbf{B}(\mathbf{B} + \mathbf{I})^{-1}\mathbf{A}^T\mathbf{x} \tag{4.73}$$

$$\mathbf{A} = \mathbf{V}_w \mathbf{\Lambda}_w^{-1/2} \tag{4.74}$$

$$\mathbf{B} = \mathbf{A}^T \mathbf{\Sigma}_s \mathbf{A} \tag{4.75}$$

where $\mathbf{V}_w$ and $\mathbf{\Lambda}_w$ are the modal and diagonal matrices of $\mathbf{\Sigma}_w$, and $\mathbf{B}$ is a pre-whitened version of the covariance matrix of the target signal.

Note that $\mathbf{\Sigma}_w$ and $\mathbf{B}$ are symmetric, hence their modal matrices $\mathbf{V}_w$ and $\mathbf{V}_B$ are orthogonal. If $\mathbf{\Lambda}_B$ is the diagonal matrix of $\mathbf{B}$, by making use of the orthogonality of $\mathbf{V}_B$ we arrive at the canonical form of the test statistic

$$T(\mathbf{y}) = \mathbf{y}^T \mathbf{\Lambda}_B (\mathbf{\Lambda}_B + \mathbf{I})^{-1}\mathbf{y} \tag{4.76}$$

$$\mathbf{y} = \mathbf{V}_B^T \mathbf{A}^T \mathbf{x} \tag{4.77}$$

where $\mathbf{y}$ is the *decorrelated* version of $\mathbf{x}$. It is worth noting that the matrix $\mathbf{D} = \mathbf{A}\mathbf{V}_B$ simultaneously decorrelates the signal ($\mathbf{D}^T \mathbf{\Sigma}_s \mathbf{D} = \mathbf{\Lambda}_B$) and the noise ($\mathbf{D}^T \mathbf{\Sigma}_w \mathbf{D} = \mathbf{I}$).

As shown in Section 4.7, it can be demonstrated through eigendecomposition that if $\breve{\mathbf{\Lambda}}_s$ and $\breve{\mathbf{V}}_s$ are the diagonal and modal matrices of the normalised covariance matrix of the target signal $\breve{\mathbf{\Sigma}}_s$, where $\mathbf{\Sigma}_s = \sigma_s^2 \breve{\mathbf{\Sigma}}_s$, then $\mathbf{\Lambda}_s = \sigma_s^2 \breve{\mathbf{\Lambda}}_s$ and $\mathbf{V}_s = \breve{\mathbf{V}}_s$. This same principle is applicable to the coloured noise ($\mathbf{\Lambda}_w = \sigma_w^2 \breve{\mathbf{\Lambda}}_w$ and $\mathbf{V}_w = \breve{\mathbf{V}}_w$ for $\mathbf{C}_w = \sigma_w^2 \breve{\mathbf{\Sigma}}_w$) and the matrix $\mathbf{B}$ ($\mathbf{\Lambda}_B = \sigma_B^2 \breve{\mathbf{\Lambda}}_B$ and $\mathbf{V}_B = \breve{\mathbf{V}}_B$ for $\mathbf{B} = \sigma_s^2 \mathbf{A}^T \breve{\mathbf{\Sigma}}_s \mathbf{A}$, where $\mathbf{A} = \sigma_w^{-1} \mathbf{V}_w \breve{\mathbf{\Lambda}}_w^{-1/2}$ and $\sigma_B^2 = \sigma_s^2 / \sigma_w^2$ is the signal-to-noise energy ratio).

As a result, the test statistic $T(\mathbf{y})$ in (4.76) can be computed if $\breve{\mathbf{\Sigma}}_s$, $\breve{\mathbf{\Sigma}}_w$, σ_s^2 and σ_w^2 are known. The normalised covariance matrix of the target signal $\breve{\mathbf{\Sigma}}_s$ needs to be calculated only once using a previously built training dataset. The signal variance σ_s^2 can be obtained from the data vector $\mathbf{x}$ if the noise variance σ_w^2 is known, as is typically the case. The normalised noise covariance matrix $\breve{\mathbf{\Sigma}}_w$ needs to be computed *once* from a "quiet" period of ambient noise representative of the noise during the period of activity. Sometimes, $\breve{\mathbf{\Sigma}}_w$ needs to be computed more than once, depending on the stability of the normalised noise spectrum throughout the survey. The noise variance σ_w^2 can be obtained from "quiet" intervals in between target events.

Since $\mathbf{\Lambda}_B$ is a diagonal matrix, the canonical form from (4.76) can be rewritten as

$$T(\mathbf{y}) = \sum_{n=0}^{N-1} \frac{\lambda_{Bn}}{\lambda_{Bn} + 1} y^2[n] \tag{4.78}$$

In (4.78), the data vector $\mathbf{x}$ is linearly transformed into a vector $\mathbf{y}$ of uncorrelated random variables of different variance, which is sent through a *weighted* energy detector.

For the estimator-correlator in CGN, the expressions of the probability density functions of $T(\mathbf{y})$ for the null and alternative hypotheses $p_{0,1}$ are

$$p(T(\mathbf{y});\, \mathcal{H}_0) = \mathcal{F}_{-t}^{-1}\left\{\prod_{n=0}^{N-1} \frac{1}{\sqrt{1-2j\alpha_n\omega}}\right\} \tag{4.79}$$

$$\alpha_n = \frac{\lambda_{Bn}}{\lambda_{Bn}+1} \tag{4.80}$$

$$p(T(\mathbf{y});\, \mathcal{H}_1) = \mathcal{F}_{-t}^{-1}\left\{\prod_{n=0}^{N-1} \frac{1}{\sqrt{1-2j\lambda_{Bn}\omega}}\right\} \tag{4.81}$$

where $\mathcal{F}_{-t}^{-1}\{f(\omega)\}$ is the inverse Fourier transform evaluated at $-t$, as in (4.63). We derived these expressions by applying to the canonical form of (4.76) the same procedure used for the estimator-correlator in WGN (Kay, 1998, p. 183).

Hence, the expressions for the probabilities of false alarm and detection are

$$P_{FA} = \Pr\{T(\mathbf{y}) > \gamma'';\, \mathcal{H}_0\} = \int_{\gamma''}^{\infty}\int_{-\infty}^{\infty} \frac{1}{2\pi}\prod_{n=0}^{N-1} \frac{1}{\sqrt{1-2j\alpha_n\omega}} e^{-j\omega t} d\omega\, dt \tag{4.82}$$

$$P_D = \Pr\{T(\mathbf{y}) > \gamma'';\, \mathcal{H}_1\} = \int_{\gamma''}^{\infty}\int_{-\infty}^{\infty} \frac{1}{2\pi}\prod_{n=0}^{N-1} \frac{1}{\sqrt{1-2j\lambda_{Bn}\omega}} e^{-j\omega t} d\omega\, dt \tag{4.83}$$

The dependence of the performance curves in (4.82) and (4.83) on the SNR can be easily analysed through the function $\gamma'(SNR)$ for fixed target probabilities of false alarm α and detection $(1-\beta)$, as shown in Figure 4.12 for $\alpha, \beta = 0.5$. Like the estimator-correlator in WGN (see Figure 4.8), the threshold curves from P_{FA} and P_D converge at low SNR values and diverge at high SNR values. But unlike the previous two detectors, the threshold γ' does not depend on the noise variance σ_w^2.

Figure 4.12 presents $\gamma'(SNR)$ for three types of signal (flat, one-octave, and third-octave band-filtered white Gaussian process) and two types of noise (one-octave and third-octave band-filtered white Gaussian process, shown on the *left* and *right* images, respectively). By comparing the two images, one can see that both the known target signal and noise affect the performance of the detector. In particular, the larger the difference between the target signal and the coloured noise, the better the performance of the detector for any given SNR when compared to the estimator-correlator in WGN. In Figure 4.12, this improved behaviour appears as a "left-displacement" of the $\gamma'(SNR)$ curves, or effectively, a larger ratio P_D/P_{FA} for any given SNR.

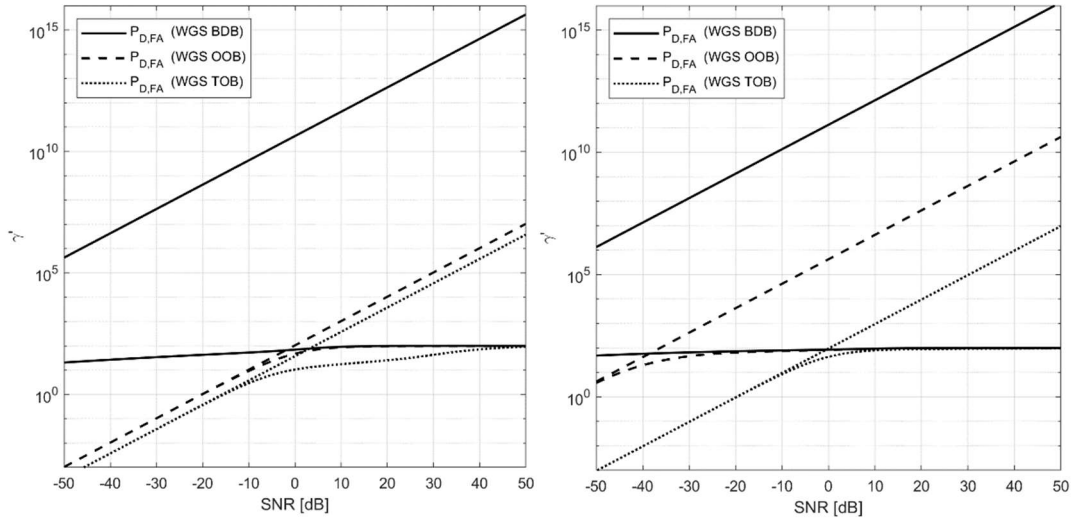

Figure 4.12 Detection threshold of *estimator-correlator* in CGN as a function of the signal-to-noise ratio for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. The CGN is WGN filtered in one-octave (left) and third-octave (right) bands centred at half the Nyquist frequency ($f_s/4$). Three signals are represented: broadband WGS (solid lines), one-octave filtered WGS (dashed lines) and third-octave filtered WGS (dotted lines). Top (diagonal) and bottom (flattening) curves correspond to P_D and P_{FA}, respectively. $N = 100$, $\sigma_w^2 = 1$.

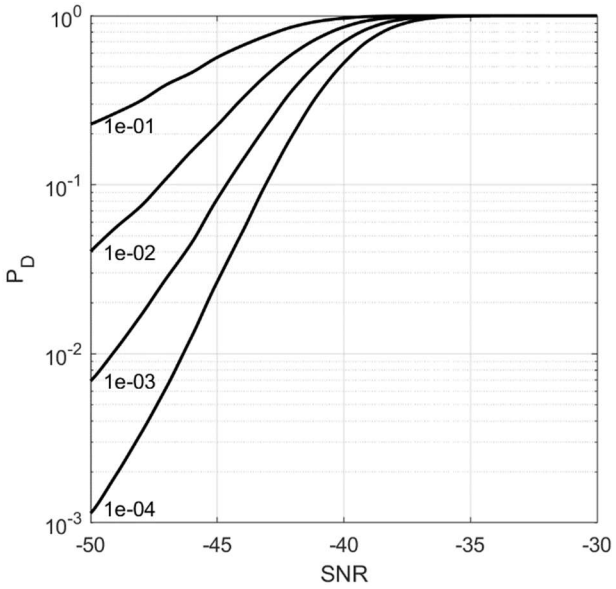

Figure 4.13 Probability of detection P_D of the *estimator-correlator* in CGN as a function of the SNR for four target probabilities of false alarm α. The values of α are shown below their respective curves. One-octave filtered WGS, third-octave filtered WGN. $N = 100$, $\sigma_w^2 = 1$.

It is interesting to note that for the case of a target signal and noise with matching covariance matrices, the $\gamma'(SNR)$ curves from the estimator-correlator in CGN and the estimator-correlator in WGN (for a WGS) are all identical. This scenario represents the worst-possible performance of the estimator-correlators, which will behave as simple energy detectors. This observation confirms that it is in fact the difference between the target signal and the noise what determines the performance of an estimator-correlator.

In summary, the estimated performance of the estimator-correlator in CGN depends exclusively on the SNR and how different the target signal is from the noise. Figure 4.13 shows $P_D(SNR)$ for four target probabilities of false alarm and $N = 100$ for one-octave filtered WGS. Figure 4.14 represents the corresponding ROC curves for four SNR values. Both figures show the expected increase of P_D/P_{FA} with SNR. The detection performance is improved with respect to Figure 4.6 and Figure 4.7, and Figure 4.10 and Figure 4.11.

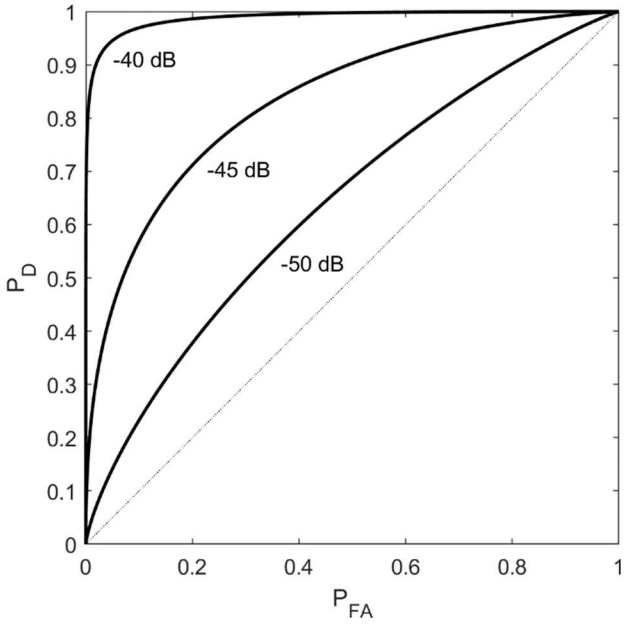

Figure 4.14 Receiver operating characteristic ROC of the *estimator-correlator* in CGN for four signal-to-noise ratios. The values of SNR are shown below their respective curves. One-octave filtered WGS, third-octave filtered WGN. $N = 100$, $\sigma_w^2 = 1$.

4.9 Behaviour of Detectors with Target and Non-Target Signals

The performance of the energy detector and the two estimator-correlators has been briefly discussed in the previous three sections. The detection performance is fully characterised by the probability density functions of $T(\mathbf{x})$ for the null and alternative hypotheses $p_{0,1}(T)$. These PDF are simulated from a small number of known parameters, namely, the number of variables N, the signal-to-noise ratio SNR, and the normalised covariance matrices of the target signal and noise $\breve{\boldsymbol{\Sigma}}_{s,w}$. The $p_{0,1}(T)$ curves provide information about how well the detector is expected to perform, but they do not give insight into how the detector may respond to target and non-target signals and noise.

In this section we aim to provide a more complete understanding of the practical behaviour of the NP detectors. We will do so by comparing the test statistic $T(\mathbf{x})$ of simulated observations of signal-plus-noise ("alternative hypothesis") to the detection threshold γ'_D, for a range of SNR values and a fixed target probability of false alarm α and detection $(1-\beta)$. Put simply, we will assess the practical performance of the detectors by means of a $\gamma'(SNR)$ graph with an overlaying scatter plot of $T(\mathbf{x})$.

The results in the following sub-sections are based on three types of N-variable observations, created by filtering simulated white Gaussian vectors with one of the following frequency responses: 'flat' or unfiltered, 'var1' or bandpass, 'var2' or bandstop (see Figure 4.15). The $p_{0,1}(T)$ curves are computed for various of the nine possible combinations of target signal and noise. The test statistics $T(\mathbf{x})$ are calculated over multiple observations, with the signal and noise simulated independently using one of the three frequency responses ('flat', 'var1', 'var2'), before they are added together.

These three frequency responses are purposefully selected to study the detector's behaviour in singular and extreme scenarios. The *flat* response agrees with the white Gaussian assumption for the signal and noise in the energy detector, and for the noise in the estimator-correlator in WGN. The *bandpass* response is included to simulate the band-limited character of signals and noise found in practice. The *bandstop* response, as a pseudo-complementary version of the bandpass response, can help evaluate the detector's behaviour when the character of the *observed* signal (or noise) is nearly opposite to that of the *target* signal (or noise).

For clarity, the analysis is split into a "small signal" region ($SNR < 0$ dB) and a "large signal" region ($SNR > 0$ dB). In the following sub-sections, the behaviour of the detector on the large signal region (left) and small signal region (right) is shown from Figure 4.16 to Figure 4.23. For the images on the left, we considered three signal types ('flat', 'var1', 'var2') and one noise type matching the target noise. For the images on the right, we considered three noise types ('flat', 'var1', 'var2') and one signal type matching the target signal. In this way, the effect of the *non-target* signal or noise is confined to the SNR region of interest.

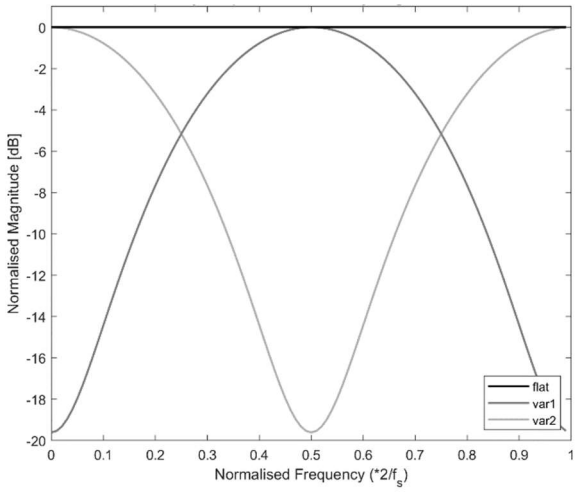

Figure 4.15 Normalised power spectral response of the filters used to generate the white Gaussian vectors that represent the target and non-target signals and background noise.

4.9.1 Energy Detector

In the energy detector, the target signal and noise are assumed to be white Gaussian (WGS, WGN). Figure 4.16 shows $\gamma'_{D,FA}(SNR)$ for the target signal and noise along with $T(\mathbf{x})$ for observations of signal plus noise. The test statistic $T(\mathbf{x})$ follows $\gamma'_D(SNR)$ for all three types of signal and noise ('flat', 'var1', 'var2').

Now, if the detector assumes that the signal and noise are white, then why do the values of the test statistic calculated over "coloured" observations agree with $\gamma'_D(SNR)$? Although this may be unintuitive, from (4.41) we know that the test statistic in the energy detector does not depend on the type of signal or noise, only on their variances. More specifically, given the length of the observation N and noise variance σ_w^2, $T(\mathbf{x})$ only depends on the SNR.

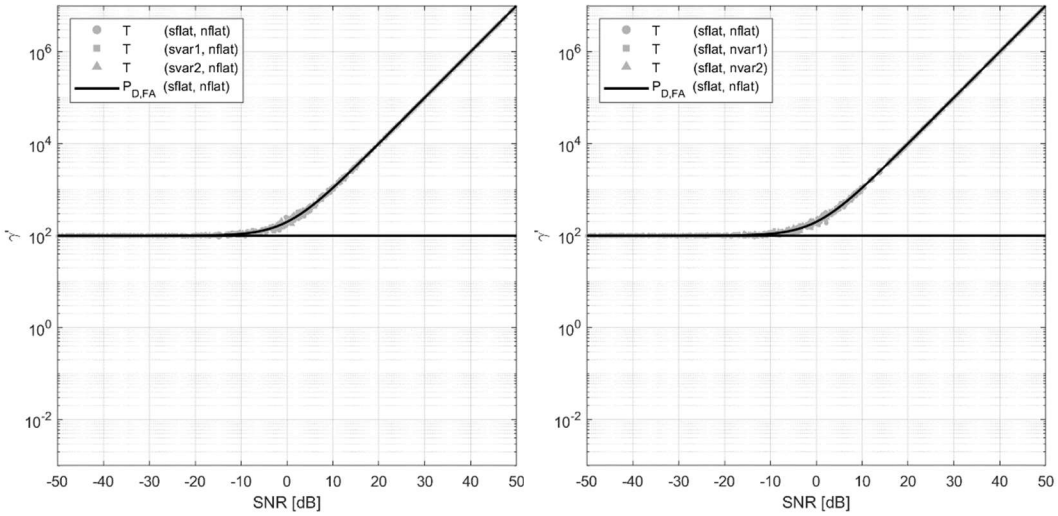

Figure 4.16 Detection threshold of *energy detector* as a function of SNR for target probabilities of false alarm and miss $\alpha, \beta = 0.5$. The black curves correspond to P_D (top) and P_{FA} (bottom) for the case of WGS and WGN. The test statistic (grey circles) is calculated over simulated $\mathbf{x}$ vectors of Gaussian signal plus noise for three types of signal ('flat', 'var1', 'var2') and WGN (left) and for three types of noise ('flat', 'var1', 'var2') and WGS (right). $N = 100$, $\sigma_w^2 = 1$.

4.9.2 Estimator-Correlator in WGN

WGS and WGN

Let's start by analysing the behaviour of this detector in the most basic scenario, in which the target signal and noise are both white Gaussian (WGS, WGN). Figure 4.17 shows $\gamma'_{D,FA}(SNR)$ along with $T(\mathbf{x})$ for three different combinations of signals and noise. The test statistic of the observations follows $\gamma'_D(SNR)$ regardless of the type of signal and noise.

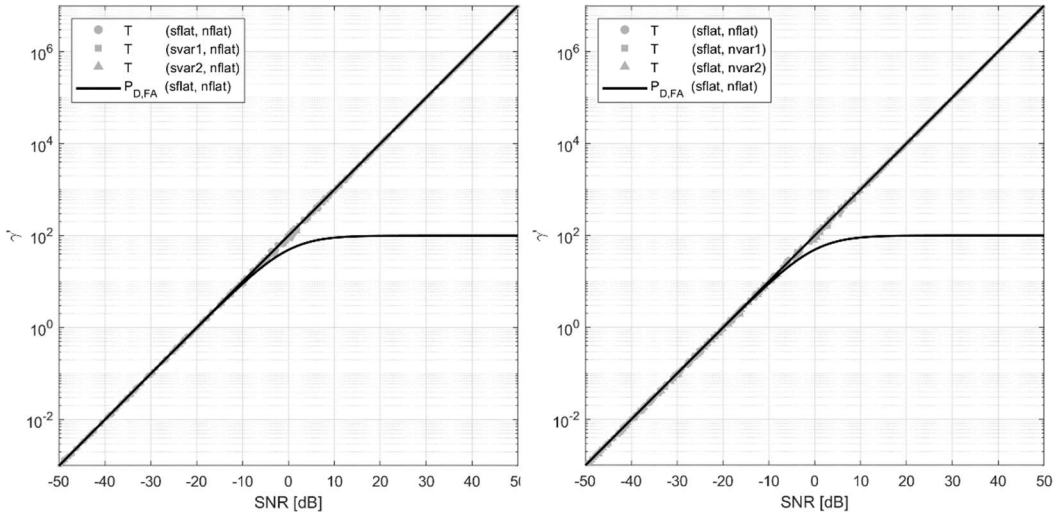

Figure 4.17 Detection threshold of *estimator-correlator in WGN* as a function of SNR for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. The black curves correspond to P_D (top) and P_{FA} (bottom) for the case of WGS and WGN. The test statistic (grey circles) is calculated over simulated $\mathbf{x}$ vectors of Gaussian signal plus noise (alternative hypothesis) for three types of signal ('flat', 'var1', 'var2') and WGN (left) and for three types of noise ('flat', 'var1', 'var2') and WGS (right). $N = 100$, $\sigma_w^2 = 1$.

This behaviour is identical to that of the energy detector. Thus, if the target signal is white Gaussian, there is no benefit in using the estimator correlator in WGN over the energy detector. If we look at the expression of the test statistic in (4.54), we can easily demonstrate that for a target WGS ($\mathbf{\Sigma}_s = \sigma_s^2\mathbf{I}$), and given N and σ_w^2, for the signal-plus-noise observation the test statistic $T(\mathbf{x})$ only depends on the SNR

$$T(\mathbf{x}) = \frac{\sigma_s^2}{\sigma_t^2}\sum_{n=0}^{N-1} x_n^2 = \sigma_w^2 N\left(\frac{\sigma_s^2}{\sigma_w^2}\right) \tag{4.84}$$

CGS and WGN

In the estimator-correlator in WGN only the noise is assumed white Gaussian. Let's see how the estimator-correlator behaves when the signal is coloured (CGS) and the noise is white (WGN). In this scenario, the detector is expected to show its full potential.

Figure 4.18 shows $\gamma'_{D,FA}(SNR)$ for a bandpass white Gaussian target signal ('var1') and white Gaussian target noise ('flat'), along with $T(\mathbf{x})$ for three different combinations of signals and noise. The test statistics of the non-target signals ('flat', 'var2') do no longer follow the $\gamma'_D(SNR)$ curve in the "large signal" region (Figure 4.18, left). The same happens with the non-target noise ('var1', 'var2') in the "small signal" region (Figure 4.18, right).

In the "large signal" region, and as it should be expected, the test statistic for the target signal and noise ('svar1, nflat') matches $\gamma'_D(SNR)$. The observations containing non-

target signals ('sflat, nflat' and 'svar2, nflat') are lower than $\gamma'_D(SNR)$, particularly in the vicinity of $SNR = 0$ dB, with 'svar2, nflat' giving the lowest test statistic. It appears that the more different the observation is from the target signal, the lower $T(\mathbf{x})$ will be. This can be proved by inspecting (4.58). In that equation, $T(\mathbf{y}) = \sum_{n=0}^{N-1} k(\lambda_{sn}) y^2[n]$ with $\mathbf{y} = \mathbf{V}_s^T \mathbf{x}$ and k being the *weighting curve* ($k \approx 0$ for $n = 0$, $k \approx 1$ for $n = N - 1$). Note that for eigenvalues sorted in increasing order, the eigenvectors in $\mathbf{V}_s$ are ordered from less to more statistically similar to the target signal. Thus, $\mathbf{y}$ will be smaller if the observation $\mathbf{x}$ is not well correlated with the dominant (signal-like) eigenvectors in $\mathbf{V}_s$.

In the "small signal" region, the observations containing non-target noise ('svar1, nvar1' and 'svar1, nvar2') deviate from $\gamma'_D(SNR)$. The test statistic for the observations of the type 'svar1, nvar1' are higher than $\gamma'_D(SNR)$. This can be explained by $\mathbf{x}$ being well-correlated with the dominant (signal-like) eigenvalues, as opposed to what would happen if $\mathbf{x}$ was white Gaussian. On the other hand, the test statistic for the observations of the type 'svar1, nvar2' are lower than $\gamma'_D(SNR)$. In this case, $\mathbf{x}$ is well-correlated with the least-dominant (signal-unlike) eigenvalues and badly-correlated with the dominant (signal-like) eigenvalues.

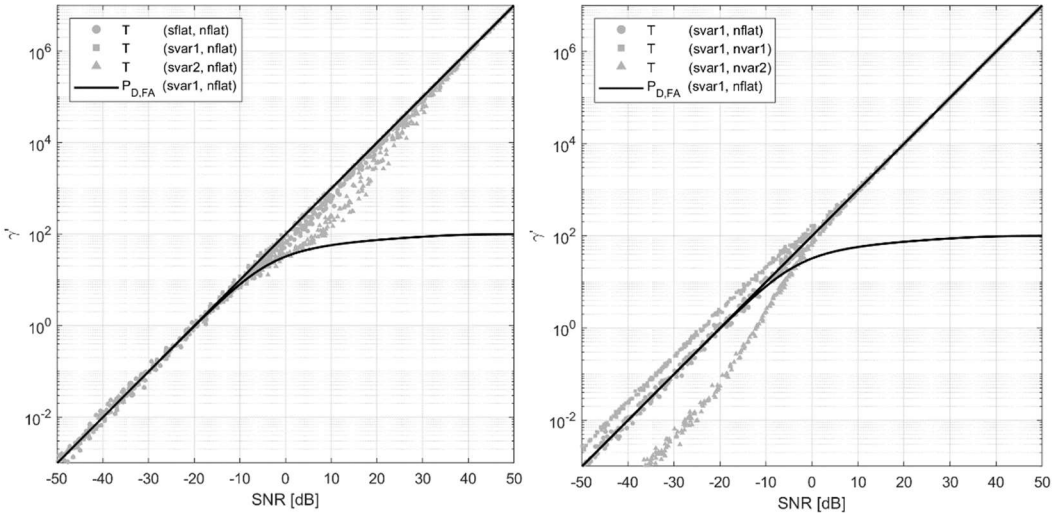

Figure 4.18 Detection threshold of *estimator-correlator in WGN* as a function of SNR for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. The black curves correspond to P_D (top) and P_{FA} (bottom) for the case of bandpass ('var1') signal and WGN. The test statistic (grey circles) is calculated over simulated $\mathbf{x}$ vectors of Gaussian signal plus noise (alternative hypothesis) for three types of signal ('flat', 'var1', 'var2') and WGN (left) and for three types of noise ('flat', 'var1', 'var2') and bandpass signal (right). $N = 100$, $\sigma_w^2 = 1$.

In summary, if $SNR > 0$ the estimator-correlator in WGN will penalise observations that are statistically different from the target signal. In that sense, the more "coloured" the target signal is (i.e., further from a white response), the greater the benefit of using the estimator-correlator in WGN over the energy detector. If $SNR < 0$, the estimator-correlator in WGN will penalise observations that are statistically different from the target signal, but will boost those that are similar to it. The latter situation can be

problematic, as it can increase the false alarm rate when the SNR is low. For this reason, we do not recommend using the estimator-correlator in WGN in underwater sound processing (low-frequency dominated noise) or when filtering is used (coloured noise). The estimator-correlator in CGN is a more effective detector for these applications.

4.9.3 Estimator-Correlator in CGN

WGS and WGN

First, let's analyse the behaviour of estimator-correlator in CGN for the basic scenario of white Gaussian target signal and noise (WGS, WGN). From the discussion in the previous two sub-sections, it should be expected that the estimator-correlator in CGN will behave as the energy detector if the target signal and noise are set as white Gaussian (WGS, WGN). This is what happens in practice.

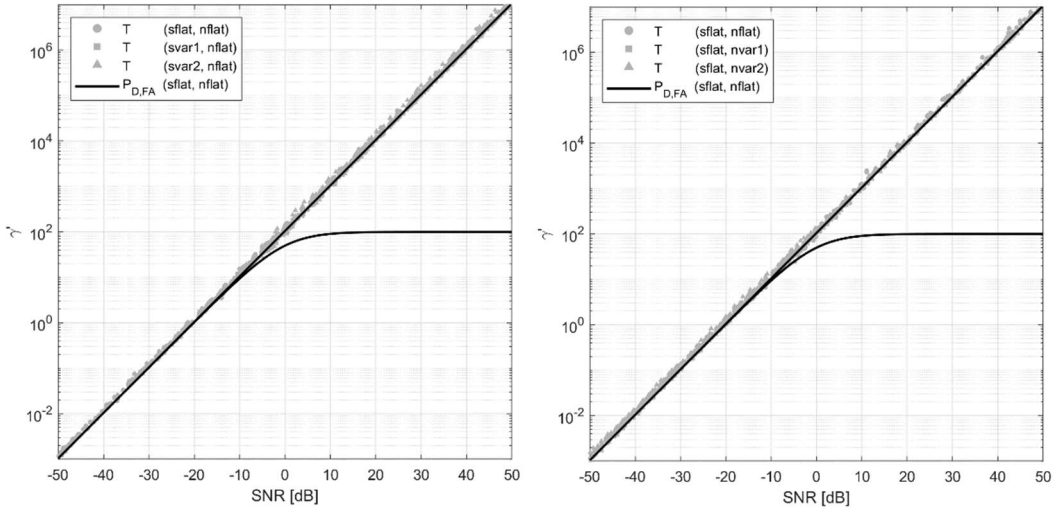

Figure 4.19 Detection threshold of *estimator-correlator in CGN* as a function of SNR for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. The black curves correspond to P_D (top) and P_{FA} (bottom) for the case of WGS and WGN. The test statistic (grey circles) is calculated over simulated $\mathbf{x}$ vectors of Gaussian signal plus noise (alternative hypothesis) for three types of signal ('flat', 'var1', 'var2') and WGN (left) and for three types of noise ('flat', 'var1', 'var2') and WGS (right). $N = 100$, $\sigma_w^2 = 1$.

The proof can be found by replacing $\mathbf{\Sigma}_s$ and $\mathbf{\Sigma}_w$ with their white Gaussian versions ($\sigma_s^2\mathbf{I}$ and $\sigma_w^2\mathbf{I}$) in the $T(\mathbf{x})$ formula from (4.71). The result can be compared with (4.41) and (4.84). For this scenario, equation (4.71) becomes

$$T(\mathbf{x}) = \frac{\sigma_s^2}{\sigma_w^2\sigma_t^2}\sum_{n=0}^{N-1} x_n^2 = N\left(\frac{\sigma_s^2}{\sigma_w^2}\right) \tag{4.85}$$

CGS and WGN

Here, we analyse the behaviour of estimator-correlator in CGN for the scenario of a coloured target signal in white Gaussian noise (CGS, WGN). By setting the noise as white Gaussian, the estimator-correlator in CGN will behave as an estimator-correlator in WGN. This can be seen by comparing Figure 4.20 with Figure 4.18.

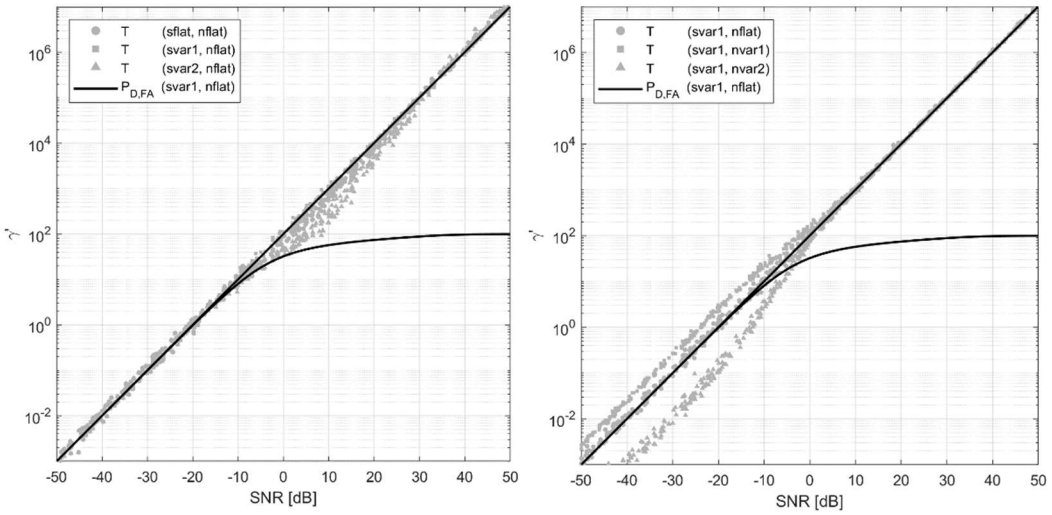

Figure 4.20 Detection threshold of *estimator-correlator in CGN* as a function of SNR for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. The black curves correspond to P_D (top) and P_{FA} (bottom) for the case of bandpass ('var1') signal and WGN. The test statistic (grey circles) is calculated over simulated $\mathbf{x}$ vectors of Gaussian signal plus noise (alternative hypothesis) for three types of signal ('flat', 'var1', 'var2') and WGN (left) and for three types of noise ('flat', 'var1', 'var2') and bandpass signal (right). $N = 100$, $\sigma_w^2 = 1$.

Further proof can be found by inserting the white Gaussian form of $\boldsymbol{\Sigma}_w$ (i.e., $\sigma_w^2\mathbf{I}$) into the $T(\mathbf{x})$ formula from (4.71) and then compare the result with (4.54). For this scenario, equation (4.71) becomes

$$T(\mathbf{x}) = \sigma_w^{-2}\mathbf{x}^T\boldsymbol{\Sigma}_s(\boldsymbol{\Sigma}_s + \sigma_w^2\mathbf{I})^{-1}\mathbf{x} \tag{4.86}$$

WGS and CGN

In this case, we analyse the behaviour of the estimator-correlator in CGN when the target signal and noise are white and coloured Gaussian processes (WGS, CGN). This scenario is represented in Figure 4.21, where $\gamma'_{D,FA}(SNR)$ is shown along $T(\mathbf{x})$ for three different combinations of signals and noise. $\gamma'_{D,FA}(SNR)$ is computed from a white Gaussian target signal ('flat') and bandstop Gaussian target noise ('var2').

In the "large signal" region (Figure 4.21, left), $T(\mathbf{x})$ is higher than $\gamma'_D(SNR)$ for the non-target observations 'svar1, nvar2', and lower than $\gamma'_D(SNR)$ for the non-target observations 'svar2, nvar2'.

How can the observation of a non-target signal produce a test statistic that exceeds the detection threshold $\gamma'_D(SNR)$? Should not the largest score come from an observation of the target signal? Let's look at the expression (4.78) for the test statistic in the eigen space to try to answer these questions. If we modify (4.78) to consider a white Gaussian target signal by setting $\mathbf{\Sigma}_s = \sigma_s^2\mathbf{I}$ (hence, $\mathbf{V}_B = \mathbf{I}$ and $\boldsymbol{\lambda}_B = \sigma_s^2\lambda_w^{-1}$) we have that

$$T(\mathbf{y}) = \sum_{n=0}^{N-1} \frac{\sigma_s^2}{\sigma_s^2 + \lambda_{wn}} y^2[n] \tag{4.87}$$

$$\mathbf{y} = \mathbf{\Lambda}_w^{-1/2}\mathbf{V}_w^T\mathbf{x} \tag{4.88}$$

where $\mathbf{V}_w^T\mathbf{x}$ is the prewhitened (i.e., noise-compensated) observation. For eigenvalues sorted in ascending order, the eigenvectors in $\mathbf{V}_w$ that are statistically similar to the target noise appear in the last columns. Therefore, $\mathbf{V}_w^T\mathbf{x}$ increases towards the end for observations similar to the noise. However, $\mathbf{\Lambda}_w^{-1/2}$ attenuates the last elements in $\mathbf{V}_w^T\mathbf{x}$ and enhances the first ones, making $\mathbf{y}$ larger for noise-differing observations. The weighting term $\sigma_s^2/(\sigma_s^2 + \lambda_{wn})$ decreases towards the end, also making $T(\mathbf{y})$ larger for observations that are different from the target noise. In other words, for a WGS target signal and CGN target noise, the more different the observation is from the target noise, the larger $T(\mathbf{y})$ will be. Therefore, it is possible to find observations that are statistically less similar to the target noise than the target signal is. This means that the detection rate for some non-target signals can be larger than for the target WGS itself.

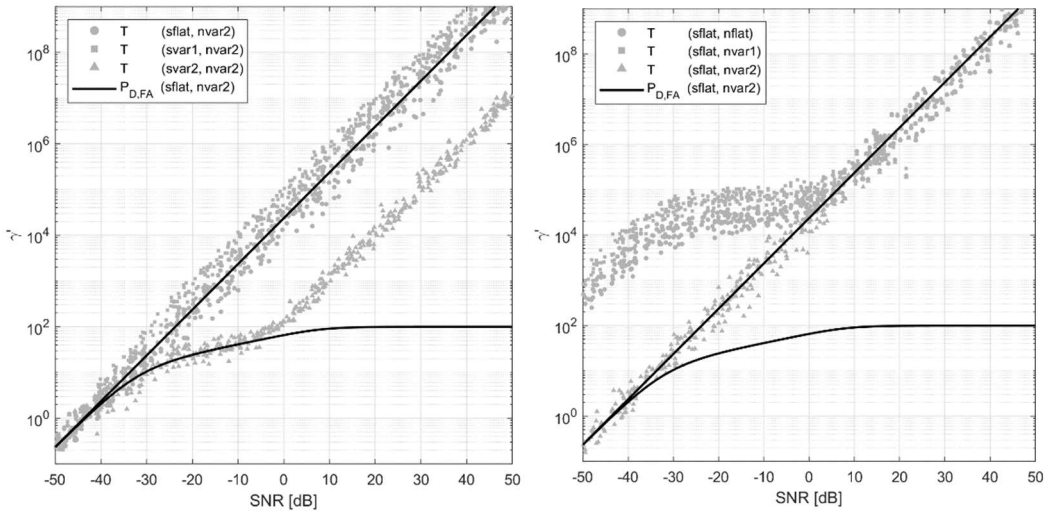

Figure 4.21 Detection threshold of *estimator-correlator in CGN* as a function of SNR for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. The black curves correspond to P_D (top) and P_{FA} (bottom) for the case of WGS and bandstop ('var2') noise. The test statistic (grey circles) is calculated over simulated $\mathbf{x}$ vectors of Gaussian signal plus noise (alternative hypothesis) for three types of signal ('flat', 'var1', 'var2') and bandstop noise (left) and for three types of noise ('flat', 'var1', 'var2') and WGS (right). $N = 100$, $\sigma_w^2 = 1$.

In the "small signal" region (Figure 4.21, right), the test statistic for the non-target observations 'sflat, nflat' and 'sflat, nvar1' are higher than $\gamma'_D(SNR)$. The same reasoning made for the "large signal" region can be applied here. The less similar the observation is to the target noise, the larger the test statistic. In underwater sound processing, this is not an issue necessarily, since the background noise tends to be stable for long periods and can be accurately characterised. An increase in false alarm rates may only occur if the noise covariance matrix is built from local training data and applied over a large period of observations where the background noise varies considerably.

CGS and CGN

In the estimator-correlator in CGN, both the target signal and noise can be coloured Gaussian processes. Let's analyse the behaviour of the detector for this particular scenario (CGS, CGN). Figure 4.22 shows $\gamma'_{D,FA}(SNR)$ along with $T(\mathbf{x})$ for three different combinations of signals and noise. $\gamma'_{D,FA}(SNR)$ is computed from a bandpass Gaussian target signal ('var1') and bandstop Gaussian target noise ('var2'). As with the other two detectors, $T(\mathbf{x})$ deviates from $\gamma'_{D,FA}(SNR)$ for non-target signals and noise.

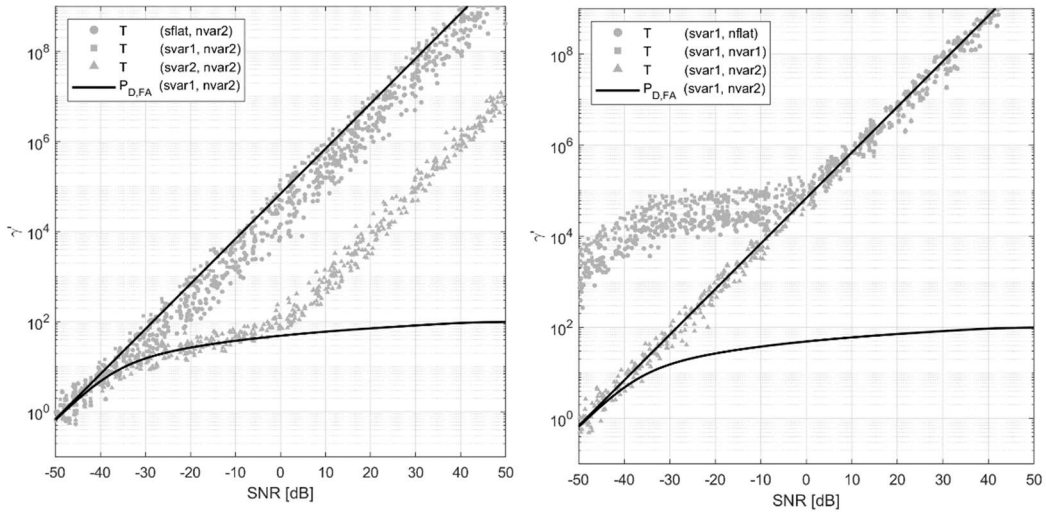

Figure 4.22 Detection threshold of *estimator-correlator in CGN* as a function of SNR for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. The black curves correspond to P_D (top) and P_{FA} (bottom) for the case of bandpass ('var1') signal and bandstop ('var2') noise. The test statistic (grey circles) is calculated over simulated $\mathbf{x}$ vectors of Gaussian signal plus noise (alternative hypothesis) for three types of signal ('flat', 'var1', 'var2') and bandstop noise (left) and for three types of noise ('flat', 'var1', 'var2') and bandpass signal (right). $N = 100$, $\sigma_w^2 = 1$.

In the "large signal" region (Figure 4.22, left), the test statistics for the non-target observations 'sflat, nvar2' and 'svar2, nvar2' are lower than $\gamma'_D(SNR)$, with the lowest values given by 'svar2, nvar2'. We can interpret the behaviour of the estimator-correlator in CGN by breaking down the expression (4.78) for the test statistic in the eigen space $T(\mathbf{y})$. Vector $\mathbf{y}$ is the product of two terms: $\mathbf{V}_B^T$, the transpose of the modal matrix of the *prewhitened signal*; and $\mathbf{A}^T\mathbf{x}$, the *prewhitened observation*. $\mathbf{A}^T$ decorrelates the

noise from observation **x**. From previous analysis, we know that vector $\mathbf{A}^T\mathbf{x}$ is larger the less similar **x** is to the target noise, and that the largest amplitudes in $\mathbf{A}^T\mathbf{x}$ appear at the beginning. $\mathbf{V}_B^T$ decorrelates the signal from the prewhitened (noise-decorrelated) observation, and makes **y** larger the more similar **x** is to the target signal, with the largest amplitudes in **y** appearing towards the end. Hence, we can say that the test statistic in the estimator-correlator in CGN is larger the more similar the observation is to the target signal and the less similar it is to the target noise.

In the "small signal" region (right image), the test statistics for the non-target observations 'svar1, nflat' and 'svar1, nvar1' are higher than $\gamma_D'(SNR)$, with the highest values corresponding to 'svar1, nvar1'. These results confirm that the largest scores are linked to observations that are, statistically speaking, "far from the target noise" and "close to the target signal". As discussed in the previous example (Figure 4.21, right), the high stability of the background noise in underwater acoustic recordings and the precise knowledge of the statistical features of the background noise make the possibility of a high false alarm rate ($E\{T\} > \gamma_D'$) *unlikely* in the context of low SNR.

CGS and CGN with Similar Covariance Matrices

A special case of the previous example happens when the target signal is equal (or very similar) to the target noise ($\mathbf{\Sigma}_s \approx \mathbf{\Sigma}_w$). Figure 4.23 shows the behaviour of the detector for this particular case, where the target signal and noise are considered bandpass Gaussian processes ('var1'). The test statistics for *all* non-target observations are higher than $\gamma_D'(SNR)$ across all SNR.

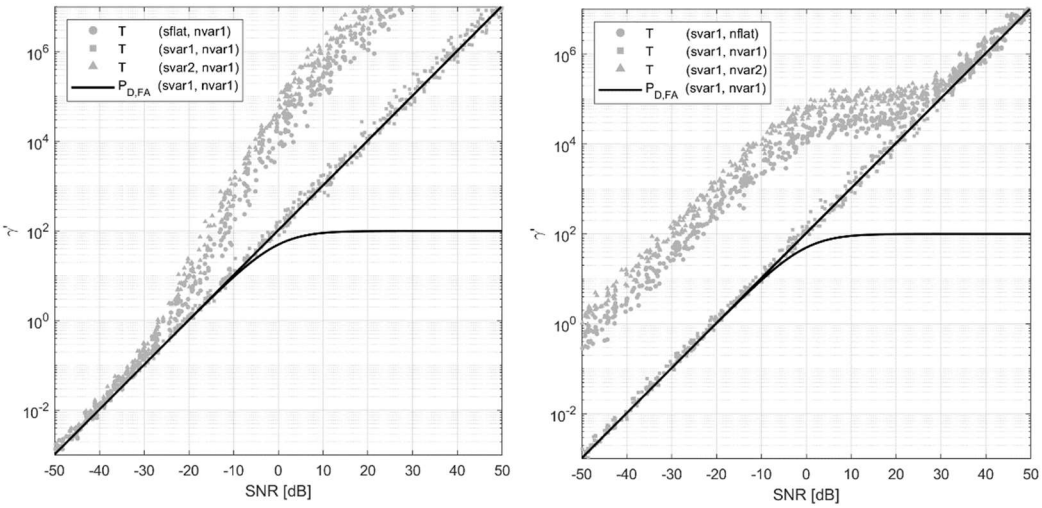

Figure 4.23 Detection threshold of *estimator-correlator in CGN* as a function of SNR for target probabilities of false alarm $\alpha = 0.5$ and miss $\beta = 0.5$. The black curves correspond to P_D (top) and P_{FA} (bottom) for the case of bandpass ('var1') signal and noise. The test statistic (grey circles) is calculated over simulated **x** vectors of Gaussian signal plus noise (alternative hypothesis) for three types of signal ('flat', 'var1', 'var2') and bandpass noise (left) and for three types of noise ('flat', 'var1', 'var2') and bandpass signal (right). $N = 100$, $\sigma_w^2 = 1$.

We can explore the behaviour of the test statistic in this context by replacing $\mathbf{\Sigma}_s$ and $\mathbf{\Sigma}_w$ in (4.78) with $\sigma_s^2\breve{\mathbf{\Sigma}}$ and $\sigma_w^2\breve{\mathbf{\Sigma}}$, where $\breve{\mathbf{\Sigma}}$ is the normalised covariance matrix of the target signal and noise (both identical). The resulting expression is

$$T(\mathbf{y}) = \sum_{n=0}^{N-1} \frac{\sigma_s^2}{\sigma_s^2 + \sigma_w^2} y^2[n] \tag{4.89}$$

$$\mathbf{y} = \sigma_w^{-1}\breve{\mathbf{\Lambda}}^{-1/2}\mathbf{V}^T\mathbf{x} \tag{4.90}$$

where $\mathbf{V}$ and $\breve{\mathbf{\Lambda}}$ are the modal and diagonal matrices of $\breve{\mathbf{\Sigma}}$. The prewhitened observation $\mathbf{V}^T\mathbf{x}$ is larger towards the last values if $\mathbf{x}$ is statistically similar to the target $\breve{\mathbf{\Sigma}}$. However, $\breve{\mathbf{\Lambda}}^{-1/2}$ penalises the last (target-like) values and enhances the first (target-unlike) ones, making $\mathbf{y}$ larger the less similar $\mathbf{x}$ is to $\breve{\mathbf{\Sigma}}$. Put simply, the less similar the observations are to the target signal (or noise), the larger $T(\mathbf{y})$ will be compared to γ'_D. This confirms that, for the case of "coloured" target signal and noise (CGS, CGN) with $\mathbf{\Sigma}_s = \mathbf{\Sigma}_w$, the mean value of the test statistic always exceeds the detection threshold ($E\{T\} > \gamma'_D$).

Assuming that the background noise is stable and well characterised, the main consequence of having a similar "coloured" target signal and noise is that the detection rate will be lower for the target signal than for non-target signals of similar amplitude. In practice, there are always differences between the background noise and the target signal. Nevertheless, one common way in which the target signal and noise can become similar is through filtering. Strong filtering can make two distinct coloured Gaussian processes (i.e., target signal and noise) statistically similar by making their spectral responses alike. On one hand, filtering can help improve the SNR of target signal observations and reduce the SNR of non-target signal observations; but on the other hand, filtering can hinder the ability of the detector to penalise observations that are similar to the target noise or different from the target signal.

In general, using filtering on an estimator-correlator in CGN is always beneficial. This is true insofar as the filter's bandwidth and central frequency are carefully selected to maximise noise rejection, while allowing the filter to exploit the differences between the target signal and the noise.

4.10 Maximising Detection Performance and Speed

The performance of an energy detector is determined by the length N and signal-to-noise ratio SNR of the data record $\mathbf{x}$. The number of observations M of the target signal and noise also play an important role in the performance of an estimator-correlator, as these are used for computing the corresponding covariance matrices. In this section we discuss how to optimise N and SNR to maximise the detection performance and processing speed. The influence of M in the estimation of covariance matrices was already examined in Sub-Section 4.3.7.

We suggest three approaches for improving the detection performance and computational speed: adjusting the record length N, filtering, and resampling.

4.10.1 Window Length

When deriving the test statistic using the LRT in (4.3), we define the alternative hypothesis $\mathcal{H}_1$ as "signal plus noise" across the entire length of the record or processing window, that is $N = N_s$. Still, we may decide to use a record length N different from the length of the target signal N_s for computing the decision statistic of a data segment. Let's discuss the effect that selecting a value of N different from N_s has on the detector's performance.

Choosing a window length $N > N_s$ might seem reasonable in the first place, as increasing the number of variables should have a positive effect on the detection performance. However, more noise samples are included as N becomes larger than N_s, reducing the SNR and increasing the chance of a 'miss'. Figure 4.24 shows $p_0(T)$ (grey) and $p_1(T)$ (black) of an energy detector for two record lengths ($N = 100$ left, $N = 500$ right), for a fixed length of the target signal ($N_s = 100$), and an original SNR of $6\,\text{dB}$ ($N = N_s$ case). As N becomes larger, we can see an improvement in $p_0(T)$ but a deterioration of the performance as $p_1(T)$ converges to $p_0(T)$. This causes a reduction in the deflection coefficient, from $d = 24$ with $SNR = 6\,\text{dB}$ to $d = 12$ with $SNR = -1\,\text{dB}$.

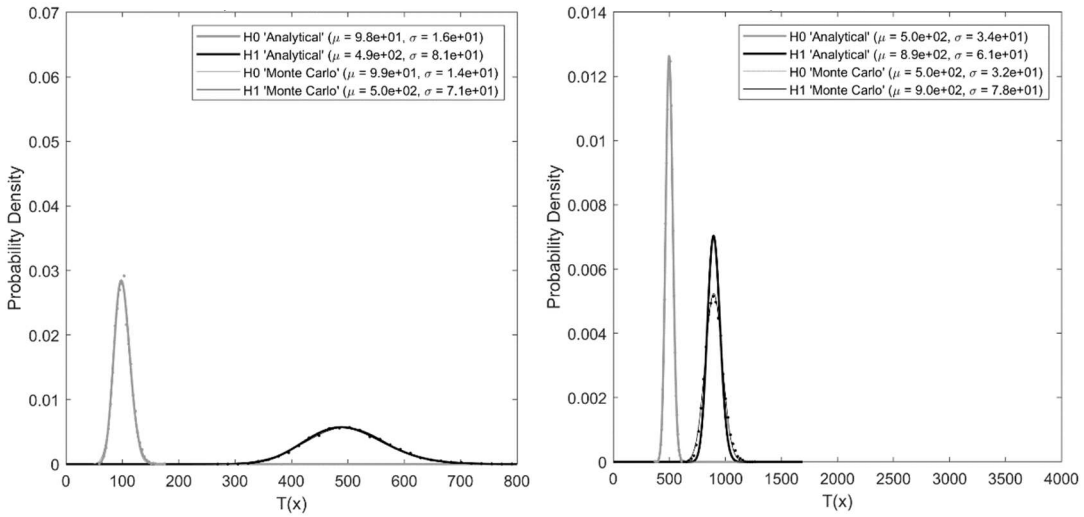

Figure 4.24 Probability density function of the *energy detector* for the null hypothesis (grey) and alternative hypothesis (black) for the cases of vector length $N = 100$ (left) and $N = 500$ (right) for $SNR = 6\,\text{dB}$ and fixed length of the target signal $N_s = 100$. The PDF are obtained from the analytical expressions (thick lines) and from Monte Carlo (MC) simulations (thin lines + dots). The deflection coefficients for the analytical and MC simulations are: $d_{A,MC} = [24.1, 27.7]$ (left), $d_{A,MC} = [11.7, 12.4]$ (right).

Now, choosing a window length $N < N_s$ will compromise the detection performance due to the lower number of variables available for processing the decision statistic. It can

also lead to more than one positive detection within the duration of the target event. Although this might not sound ideal, with this approach there is a higher chance of data records containing signal from start to end, in line with the LRT assumption ($N_s = N$). Additionally, a grouping scheme for consecutive detections can be adopted, therefore allowing target events of different length to be accurately detected.

The effect that selecting a different window length N has on the probability density functions $p_{0,1}(T)$ can be quantified by finding the relationship that exists between the means and between the variances of $p_{0,1}(T)$ for the *original* (N) and *new* (N') window length. This can be done by evaluating $p_{0,1}(T)$ from simulated sequences of "signal plus noise" and "noise only" for various window lengths N, where the signal and noise are white Gaussian, the SNR and N_s are fixed, and the noise is N-samples long ($N > N_s$). For this purpose, the PDF of the null and alternative hypotheses of the energy detector are computed using the analytical expressions in (4.42) and (4.43) and Monte-Carlo simulations. Let q be the ratio between the energies of a data vector computed for the *new* and *original* window lengths. From the calculated probability density functions $p_{0,1}(T)$, it can be easily verified that

$$\begin{aligned} \mu_0' &= q_w\, \mu_0 \\ \sigma_0' &= \sqrt{q_w}\, \sigma_0 \end{aligned} \qquad \text{for } \mathcal{H}_0 \tag{4.91}$$

$$\begin{aligned} \mu_1' &= q_t\, \mu_1 \\ \sigma_1' &= \frac{q_t}{\sqrt{q_w}}\, \sigma_1 \end{aligned} \qquad \text{for } \mathcal{H}_1 \tag{4.92}$$

where 0 and 1 subscripts represent the alternative and null hypotheses ($\mathcal{H}_1$, $\mathcal{H}_0$). The energy ratios of the "noise" q_w and "signal plus noise" q_t are given by

$$q_w = \frac{N'}{N} \tag{4.93}$$

$$q_t = \frac{N_s\sigma_s^2 + N'\sigma_w^2}{N_s\sigma_s^2 + N\sigma_w^2} = q_w \frac{1 + \frac{N_s}{N'}10^{SNR/10}}{1 + \frac{N_s}{N}10^{SNR/10}} \tag{4.94}$$

where σ_s and σ_w are the standard deviations of the signal and noise in the data vector, and SNR is the signal-to-noise ratio of the original data vector. The above expressions assume that $N' \geq N \geq N_s$, hence $q_w \geq 1$ and $q_w \geq q_t \geq 1$. Note that $q_t \approx 1$ for large SNR and $q_t \approx q_w$ for small SNR.

The PDF obtained from Monte-Carlo simulations are virtually identical to those from the analytical expressions in (4.42) and (4.43). As the window length becomes larger ($N' \gg N$), $p_1(T)$ converges to $p_0(T)$. This convergence occurs at short window lengths if SNR is already low. This behaviour, which can be deduced from (4.91) and (4.92), can be explained more intuitively by the fact that if the window length is increased

dramatically ($N \gg N_s$), the data vector will be eventually dominated by the background noise, assuming that no additional target events are introduced when increasing N.

From the above, and to prevent the selected window length to negatively affect the detection performance, we recommend using a N value equal to the length of the longest expected target event ($N = N_{s,max}$), or alternatively, a value of N several times shorter than the longest target event if a grouping scheme is to be implemented ($N < N_{s,max}$). While the first option offers better overall detection performance for signal events close to $N_{s,max}$, the second option is preferred if the duration of the target events is expected to vary.

4.10.2 Filtering

The target signal and noise generally have a different frequency spectrum. The filtering aims at taking advantage of that difference by removing from a data record any energy outside the dominant frequency band of the target signal. In most cases, this eliminates a reasonable portion of background noise energy while marginally reducing the energy of the signal, hence improving the signal-to-noise ratio.

A *white* random process has a constant power spectral density (PSD) across all frequencies. This type of process is also referred to as *uncorrelated* since its amplitude at any given instant is only correlated with itself, and its auto-correlation is the impulse or Dirac delta function. A non-white or *coloured* random process is characterised by a PSD that varies with frequency and by an auto-correlation function different from the Dirac delta, reason why it is also known as a *correlated* process. An uncorrelated random sequence can be converted into a correlated one through filtering (Viswanathan, 2020). The type of probability distribution is not altered by this linear transformation, only the spectrum of the sequence; therefore, a white Gaussian noise record will still be Gaussian after filtering.

The *estimator-correlator in WGN* and the *energy detector* assume that the noise or the signal and noise are white Gaussian processes. Filtering may not be a possibility in the first place for these two detectors, as it will transform the uncorrelated sequences into correlated ones, and the analytical expressions of their PDFs will be no longer applicable. Below we study the effect that filtering has on the PDF of the *energy detector* and how the PDFs seen in practice differ from those derived from the analytical expressions in (4.42) and (4.43). Despite the differences that exist, results show that a simple relationship can be established between the simulated and empirical PDF. This relationship offers the possibility of correcting the simulated PDF so that filtering can be applied to the energy detector. The details about the improvement of the energy detector through filtering are discussed in Section 4.11.

The *estimator-correlator in CGN*, on the other hand, can handle correlated sequences. Filtering can therefore be applied without any correction or update in the formulation.

But using filtering with this detector is not exempt of certain limitations. The estimator-correlator in CGN already uses spectral information to generate a larger score (decision statistic) for sequences with a PSD "different from the noise" and "similar to the signal". However, using a narrow bandpass filter to preserve the highest energy region of the target signal will have a detrimental effect on the performance of the detector, as there is a higher risk of benefiting non-target signals with spectral characteristics different from the noise (for details about this constraint, refer to 'CGS and CGN with Similar Covariance Matrices' in Sub-Section 4.9.3). Filtering is still a valid option for this type of detector, as long as the filter is limited to removing non-target signals while conserving the main spectral features of the target signal.

Let's examine what happens to $p_{0,1}(T)$ when a data sequence containing a band-limited signal and broadband noise is filtered. Figure 4.25 shows $p_0(T)$ (grey) and $p_1(T)$ (black) of an energy detector *without filtering* (left) and *with filtering* (right), for a record length $N = N_s = 100$, and an original SNR of 6 dB. The calculations are made with a FIR low-pass filter (LPF) of order 8 and a cut-off frequency f_c equal to half the Nyquist frequency $f_s/2$ (i.e., $f_c = f_s/4$). The noise is uncorrelated (WGN) and the signal is band-limited to $f_c = f_s/4$ (CGS). The thick lines represent the PDF from the analytical expressions in (4.42) and (4.43), and the thin lines are Gaussian curves fitted to Monte-Carlo simulations (dots). For the latter, the decision statistic is computed from 7500 simulations of signal plus noise, *unfiltered* (left) and *filtered* with the aforementioned LPF (right).

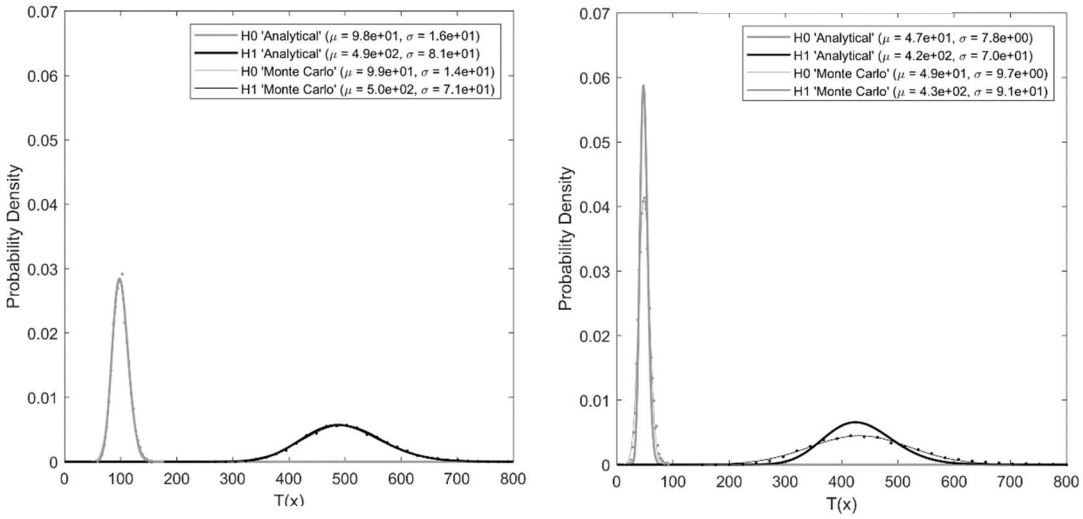

Figure 4.25 Probability density function of the *energy detector* for the null hypothesis (grey) and alternative hypothesis (black) for the cases *no filtering* (left) and *LPF filtering* with $f_c = f_s/4$. Unfiltered $SNR = 6$ dB and fixed record length $N = N_s = 100$. The PDF are obtained from the analytical expressions (thick lines) and from Monte Carlo (MC) simulations (thin lines + dots). The deflection coefficients for the analytical and MC simulations are: $d_{A,MC} = [24.1, 27.7]$ (left), $d_{A,MC} = [48.1, 39.6]$ (right).

In Figure 4.25 one can see that the filtering lowers the mean of the two probability curves $\mu_{0,1}$, but reduces μ_0 by a larger factor. There is also a reduction in the standard

deviations $\sigma_{0,1}$. The outcome is an overall improvement in the detection performance from the reduction of noise energy within the data sequence.

Note, however, that the Monte-Carlo simulations produce a PDF with a larger standard deviation than the PDF generated with the analytical expressions (4.42) and (4.43). In the example of the figure, the deflection coefficient changes from $d = 24\ (27.7)$ with $SNR = 6$ dB for the unfiltered vector to $d = 48.1\ (39.6)$ with $SNR = 9$ dB for the filtered vector, with the value in brackets corresponding to the Monte-Carlo simulations.

The effect that filtering has on the probability density functions $p_{0,1}(T)$ can be quantified by finding the relationship that exists between the means and between the variances of $p_{0,1}(T)$ *before* and *after* filtering. This can be done by evaluating $p_{0,1}(T)$ from simulated sequences of fixed length ($N = N_s$), fixed SNR, band-limited signal, and white Gaussian noise for a LPF with various cutoff frequencies. Doing so for the analytical expressions and Monte-Carlo simulation will allow us to establish a relationship between their PDFs. Let q be the ratio between the energies of the *filtered* and *unfiltered* data vector. The following relations can be easily derived from the PDFs computed with (4.42) and (4.43) for the various filter responses. The apostrophe ($'$) identifies the filtered data; no apostrophe is used for the parameters of the original, non-filtered data.

$$\begin{aligned} \mu_0' &= q_w\, \mu_0 \\ \sigma_0' &= q_w\, \sigma_0 \end{aligned} \qquad \text{for } \mathcal{H}_0, \text{analytical} \tag{4.95}$$

$$\begin{aligned} \mu_1' &= q_t\, \mu_1 \\ \sigma_1' &= q_t\, \sigma_1 \end{aligned} \qquad \text{for } \mathcal{H}_1, \text{analytical} \tag{4.96}$$

The subscripts 0 and 1 represent the alternative and null hypotheses ($\mathcal{H}_1$, $\mathcal{H}_0$). The energy ratios of the noise q_w, signal q_s, and signal plus noise q_t are lower than or equal to one and are given by

$$q_w = \frac{\sigma_w'^2}{\sigma_w^2} \tag{4.97}$$

$$q_s = \frac{\sigma_s'^2}{\sigma_s^2} \tag{4.98}$$

$$q_t = \frac{\sigma_s'^2 + \sigma_w'^2}{\sigma_s^2 + \sigma_w^2} = \frac{q_s 10^{SNR/10} + q_w}{10^{SNR/10} + 1} \tag{4.99}$$

where σ_s and σ_w are the standard deviations of the signal and noise in the data vector, and SNR is the signal-to-noise ratio of the original data vector. In the above expressions, $\sigma_w' \leq \sigma_w$ and $\sigma_s' \leq \sigma_s$, hence $q_w \leq 1$, $q_s \leq 1$. Since the whole purpose of filtering is to optimise the detection performance by reducing the amount of noise energy while conserving the signal energy, it can be assumed that $q_w \leq q_t \leq q_s$. Note that $q_t \approx q_s$ for large SNR and $q_t \approx q_w$ for small SNR.

Similarly, for the Monte-Carlo simulations

$$\begin{aligned} \mu_0' &= q_w\,\mu_0 \\ \sigma_0' &= \sqrt{q_w}\,\sigma_0 \end{aligned} \qquad \text{for } \mathcal{H}_0, \text{Monte} - \text{Carlo} \tag{4.100}$$

$$\begin{aligned} \mu_1' &= q_t\,\mu_1 \\ \sigma_1' &= \frac{q_t}{\sqrt{q_w}}\,\sigma_1 \end{aligned} \qquad \text{for } \mathcal{H}_1, \text{Monte} - \text{Carlo} \tag{4.101}$$

The relations (4.100) and (4.101) show that μ_0 is reduced more than μ_1, as $q_w < q_t$. They also show that, in practice, band-limiting a white Gaussian data sequence increases the standard deviation of its PDF relative to the mean ($\sigma_{0,1}/\mu_{0,1}$) by a factor of ${q_w}^{-1/2}$, something that is not seen in (4.95) and (4.96). Comparing the deflection coefficients before and after filtering confirms, as may be expected, that a better improvement in performance is achieved if the original SNR is low (smaller $\mu_1 - \mu_0$), and more noise energy is removed while conserving the energy of the signal (smaller q_w/q_t). There will always be a degree of improvement in detection performance as long as $q_w < q_t$. If this condition is not met, as a result of a larger reduction of signal energy from filtering compared to the reduction of noise energy, the detection performance will deteriorate.

It is also interesting to note that, for the energy detector, the PDF of a data sequence processed by any two filters will not vary if both filters cause the same energy reduction $q_{w,t}$. From here, we refer to such filters as *equivalent*. This is convenient, as any suitable filter (e.g., lowpass, bandpass) can be used to minimise the contribution from noise, and the simple relations from (4.95) to (4.101) will still hold.

Of particular use is also the simple relationship that exits between the standard deviations of the PDF from the analytical expressions (AE) and those computed from the Monte-Carlo (MC) simulations ($\sigma'_{0,1\,(MC)} = q_w^{-1/2} \cdot \sigma'_{0,1\,(AE)}$), which allows us to apply a straightforward correction to the PDF of an energy detector when filtering is used (see Section 4.11).

4.10.3 Resampling

Reducing the bandwidth and number of samples through decimation or resampling is a useful resource for speeding up the processing and representation of data. In practice, a resampling factor is selected so that the new sampling frequency is close to double the maximum frequency of the signal of interest. This incidentally improves the signal-to-noise ratio by removing the noise-dominated energy from the high-end of the spectrum. From the perspective of detection performance, it is the ability of resampling to increase the SNR that is of most interest. In this sub-section, we analyse the effect of resampling on the PDF of the energy detector for the null and alternative hypotheses $p_{0,1}(T)$.

Let's examine what happens to $p_{0,1}(T)$ when we resample a data sequence consisting of a band-limited signal and broadband noise. Figure 4.26 shows $p_0(T)$ (grey) and $p_1(T)$

(black) from an energy detector after *filtering* (left) and *resampling* (right), for a record length $N = N_s = 100$ and an original SNR of 6 dB. As before, the thick lines represent the PDF from the analytical expressions in (4.42) and (4.43), and the thin lines are Gaussian-fitted Monte-Carlo (MC) simulations. The example presented here is identical to that in the previous sub-section, except that in this case the simulated data is resampled to half of the original sampling frequency ($f_s' = f_s/2$) rather than reducing its bandwidth by two through low-pass filtering ($f_c = f_s/4$). Note that the anti-alias/recovery filter used for *resampling* and the LPF used for *filtering* have a very similar frequency response, with the same cutoff frequency and filter order.

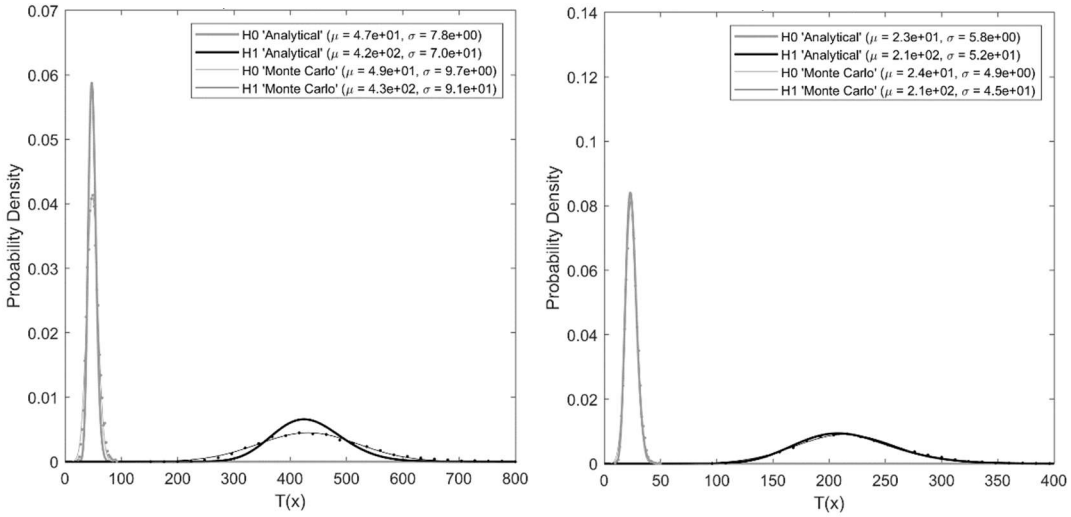

Figure 4.26 Probability density function of the *energy detector* for the null hypothesis (grey) and alternative hypothesis (black) for the cases of *LPF filtering* with $f_c = f_s/4$ (left) and *resampling* by 2 (right). Unfiltered $SNR = 6$ dB and record length before resampling of $N = N_s = 100$. The PDF are obtained from the analytical expressions (thick lines) and from Monte Carlo (MC) simulations (thin lines + dots). The deflection coefficients for the analytical and MC simulations are: $d_{A,MC} = [48.1, 39.6]$ (left), $d_{A,MC} = [31.9, 39.0]$ (right).

There are two things that stand out when looking at the PDFs of the MC simulation (thin line) for the *resample data* (right): they are practically identical to the PDFs from the analytical expressions (thick lines), and they are a scaled version of the PDFs of the MC simulations for the *filtered data* (left). The scaling constant is given by the *resampling factor*, that is, the ratio between the sampling rate *after* and *before* resampling. Since $p_0(T)$ and $p_1(T)$ are scaled by the same value, the deflection coefficient (and detection performance) of the filtered and resampled sequences are effectively the same. In the example, the deflection coefficients for the filtered and resampled vectors based on the MC simulations are $d = 39.6$ and $d = 39.0$. Both improve the SNR from 6 dB to 9 dB.

The effect that resampling has on the probability density functions $p_{0,1}(T)$ can be quantified by finding the relationship that exists between the means and between the variances of $p_{0,1}(T)$ *before* and *after* resampling. This can be done by evaluating $p_{0,1}(T)$

from simulated sequences of fixed length ($N = N_s$), fixed SNR, band-limited signal, and white Gaussian noise for various resampling factors. Let q be the ratio between the energies of the *original* and *resampled* data vector, and R the resampling factor ($R < 1$). The following relations can be easily derived from PDFs computed with (4.42) and (4.43) and from PDFs obtained from MC simulations, for various resampling factors. The parameters of the resampled data are accompanied by apostrophe ($'$), while no apostrophe is used for the parameters of the original data.

$$\begin{aligned} \mu_0' &= q_w\,\mu_0 \\ \sigma_0' &= \sqrt{R q_w}\,\sigma_0 \end{aligned} \qquad \text{for } \mathcal{H}_0 \tag{4.102}$$

$$\begin{aligned} \mu_1' &= q_t\,\mu_1 \\ \sigma_1' &= \frac{\sqrt{R}\,q_t}{\sqrt{q_w}}\,\sigma_1 \end{aligned} \qquad \text{for } \mathcal{H}_1 \tag{4.103}$$

The subscripts 0 and 1 represent the alternative and null hypotheses ($\mathcal{H}_1$, $\mathcal{H}_0$). The energy ratios of the noise q_w, signal q_s, and signal plus noise q_t are given by

$$q_w = \frac{R\sigma_w'^2}{\sigma_w^2} \tag{4.104}$$

$$q_s = \frac{R\sigma_s'^2}{\sigma_s^2} \tag{4.105}$$

$$q_t = R\,\frac{\sigma_s'^2 + \sigma_w'^2}{\sigma_s^2 + \sigma_w^2} = \frac{q_s 10^{SNR/10} + q_w}{10^{SNR/10} + 1} \tag{4.106}$$

where σ_s and σ_w are the standard deviations of the signal and noise in the data vector, and SNR is the signal-to-noise ratio of the original data vector. The above expressions assume that $\sigma_w' \le \sigma_w$ and $\sigma_s' \le \sigma_s$, hence $q_w \le R$ and $q_s \le R$. Since the purpose of resampling is to limit the frequency content of a data vector to approximately that of the target signal, it can also be assumed that $q_w \le q_t \le q_s$. Note that $q_t \approx q_s$ for large SNR and $q_t \approx q_w$ for small SNR.

The relations in (4.102) and (4.103) are notably similar to those in (4.100) and (4.101). By replacing q_w, q_s and q_t with their expressions one can see that $\mu_{0,1}'$ and $\sigma_{0,1}'$ *after resampling* are equal to $\mu_{0,1}'$ and $\sigma_{0,1}'$ *after filtering* multiplied by the resampling coefficient R. This confirms that, in an energy detector, the PDFs of a resampled data vector are a scaled version of the PDFs of the data vector processed with a filter *equivalent* to the antialias/recovery filter. The deflection coefficients of the two scenarios (resampling and filtering) are identical, and the same performance aspects discussed in Sub-Section 4.10.2 are applicable here.

We can then say that resampling has two main advantages when compared to filtering: (1) it offers the same detection performance as an *equivalent* filter but with a lower number of variables, which can help improve the computational speed, and (2) the PDFs

can be directly estimated with the analytical expressions from (4.42) and (4.43), eliminating the need for any correction to the PDFs. Hence, when it comes to optimising the detection performance, resampling is preferable to low-pass filtering.

4.11 Improving the Energy Detector's Performance with Filtering

The *energy detector* is a simple and effective solution for detecting events when there is limited information about the target signal and noise. This simplicity turns into a disadvantage when covariance data is available. The detector is unable to distinguish between target and non-target events, as it does not incorporate statistical information about the target signal and noise other than their variances (see Section 4.6).

If the target signal and noise are known, we may want to find a way of enhancing the performance of the energy detector while taking advantage of its simplicity. As discussed in Section 4.10, improvements in detection performance can be achieved through filtering, resampling, or by selecting an appropriate length for the processing window. Of the three, only filtering alters the probability densities of false alarm and detection $p_{0,1}(T)$ in a way that cannot be predicted directly with expressions (4.42) and (4.43). In this section we analyse the effect of filtering on the $p_{0,1}(T)$ and present a modified, general version of (4.42) and (4.43) that is valid for filtered and unfiltered observations.

4.11.1 Simple Correction for the PDF Equations of the Energy Detector

In the previous section we have proved that, after filtering, a simple relationship exists between the mean $\mu'_{0,1\,(MC)}$ and standard deviation $\sigma'_{0,1\,(MC)}$ of the PDFs obtained from Monte-Carlo simulations, and the mean $\mu'_{0,1\,(AE)}$ and standard deviation $\sigma'_{0,1\,(AE)}$ of the PDFs computed with the analytical expressions (4.42) and (4.43). In particular

$$\mu'_{0,1\,(MC)} = \mu'_{0,1(AE)} \tag{4.107}$$

$$\sigma'_{0,1\,(MC)} = \frac{\sigma'_{0,1\,(AE)}}{\sqrt{q_w}} \tag{4.108}$$

The mean values of the two PDFs (MC, AE) are identical, and the standard deviations are related by the scale factor $\sqrt{q_w}$, where q_w is the energy ratio of the noise given by (4.97). It appears that we only need to apply a simple correction to $p_{0,1\,(AE)}$ to arrive at $p_{0,1\,(MC)}$. Recall that the PDFs obtained from MC simulations represent the real behaviour of the energy detector.

4.11.2 Proposed General Form for the PDF Equations of the Energy Detector

While the mean and variance of a white Gaussian data vector are sufficient to calculate its PDF, the test statistic of the energy detector $T(\mathbf{x})$ follows a Chi-squared distribution

$\mathcal{X}^2_{\widetilde{N}}$, which requires the variance and length of the data vector for its computation. To arrive at the PDF of the energy detector that verifies the relations (4.107) and (4.108), we need to find the expressions of the corrected variance $\widetilde{\sigma'}^2_{w,t}$ and length $\widetilde{N}'$ and apply them to the expressions for the Chi-squared PDFs (4.42) and (4.43). Knowing that the mean and variance of a Chi-squared PDF for a N-variable Gaussian vector $\mathbf{x}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})$ are $N\sigma^2$ and $2N\sigma^4$, the mean and standard deviation of $p_{0,1}(T)_{(AE)}$ after filtering are

$$\mu'_{0\,(AE)} = \sigma'^2_w N' \tag{4.109}$$

$$\sigma'_{0\,(AE)} = \sigma'^2_w \sqrt{2N'} \tag{4.110}$$

$$\mu'_{1\,(AE)} = \sigma'^2_t N' \tag{4.111}$$

$$\sigma'_{1\,(AE)} = \sigma'^2_t \sqrt{2N'} \tag{4.112}$$

where $\sigma'^2_{w,t}$ are the variances of the observation for the null and alternative hypotheses and N' is the length of the observation, all after filtering. Similarly, the mean and standard deviation of the "real" (Monte-Carlo) PDFs can be expressed as a function of the *corrected* variance $\widetilde{\sigma'}^2_{w,t}$ and length $\widetilde{N}'$ of the observation

$$\mu'_{0\,(MC)} = \tilde{\sigma}'^2_w \widetilde{N}' \tag{4.113}$$

$$\sigma'_{0\,(MC)} = \tilde{\sigma}'^2_w \sqrt{2\widetilde{N}'} \tag{4.114}$$

$$\mu'_{1\,(MC)} = \tilde{\sigma}'^2_t \widetilde{N}' \tag{4.115}$$

$$\sigma'_{1\,(MC)} = \tilde{\sigma}'^2_t \sqrt{2\widetilde{N}'} \tag{4.116}$$

The corrected variance $\widetilde{\sigma'}^2_{w,t}$ and length $\widetilde{N}'$ are obtained for a Chi-squared PDF curve fitted to Monte-Carlo simulations of $T(\mathbf{x})$ for various filter bandwidths using unconstrained optimisation. The results showed that $\widetilde{\sigma'}^2_{w,t}$ and $\widetilde{N}'$ are given by

$$\widetilde{\sigma'}^2_{w,t} = \frac{\sigma'^2_{w,t}}{q_w} \tag{4.117}$$

$$\widetilde{N}' = \text{round}\{N' q_w\} \tag{4.118}$$

One can verify the relations (4.107) and (4.108) by inserting (4.117) and (4.118) into (4.113) to (4.116). Thus, the general form of probability density functions (4.42) and (4.43) for the energy detector, applicable to *filtered* and *unfiltered* observations, is

$$p(T(\mathbf{x});\, \mathcal{H}_0) = \tilde{\sigma}'^{-2}_w p_{\chi^2_{\widetilde{N}}}(\tilde{\sigma}'^{-2}_w T) \tag{4.119}$$

$$p(T(\mathbf{x});\, \mathcal{H}_1) = \tilde{\sigma}'^{-2}_t p_{\chi^2_{\widetilde{N}}}(\tilde{\sigma}'^{-2}_t T) \tag{4.120}$$

Assuming that the filtering improves the detection performance ($q_w < 1$), it holds that $\widetilde{\sigma'}^2_{w,t} > \sigma'^2_{w,t}$ and $\widetilde{N}' < N'$. The result is an "expansion" of $p_{0,1}(T)$ *over the mean* $\mu_{0,1}$ compared to the original expressions (4.42) and (4.43). It is worth noting that $\widetilde{\sigma'}^2_{w,t}$ and $\widetilde{N}'$ are only derived for the purpose of correcting (4.42) and (4.43), and the new expressions (4.119) and (4.120) represent the PDFs for the original parameters $\sigma'^2_{w,t}$, N', and $SNR = 10\log(\sigma'^2_s/\sigma'^2_w)$.

4.11.3 Calculating the Noise Energy Ratio q_w

There still remains the question of what value of q_w should be used to compute the probability density functions in (4.119) and (4.120). The noise energy ratio q_w can be estimated with high accuracy using its definition in (4.97), where the numerator and denominator are, respectively, the mean variances of multiple *filtered* and *unfiltered* simulated WGN vectors. However, filtering a large number of vectors can be computationally expensive.

An approximation that gives comparable results consists in calculating q_w from the filter's bandwidth Δf, in particular

$$q_w \approx \frac{2\Delta f}{f_s} \tag{4.121}$$

where f_s is the sampling rate and $\Delta f = f_2 - f_1$, with f_1 and f_2 being the bottom and top cutoff frequencies of the filter.

Both methods assume that the filtered noise is flat within the bandpass region of the filter. Ambient underwater noise is characterised by a low-frequency spectrum with a slow roll-off (-6 dB/oct from 20 Hz to 50 kHz). For typical filter bandwidths of one-octave, "flatness" within (f_1, f_2) is a reasonable assumption.

For a coloured, band-limited signal and noise the standard expressions (4.42) and (4.43) fail to predict $p_{0,1}(T)$ accurately. If filtering is applied, the corrected expressions (4.119) and (4.120) will always outperform (4.42) and (4.43) at predicting $p_{0,1}(T)$, irrespectively of the spectral response of the signal and noise.

4.11.4 Application of the General Form of the Energy Detector

Figure 4.27 shows how the *general* analytical expressions (4.119) and (4.120) compare to the Monte Carlo simulations of the energy test statistic for *filtered* observations. On the left, $p_{0,1}(T)$ are represented for the case of a LPF with cutoff frequency $f_s/4$. This is the same example as in Figure 4.25 (right), except that here the general analytical expressions $p_{0,1}(T)$ are used. On the right, $p_{0,1}(T)$ are represented for a LPF with cutoff frequency $f_s/8$. Note how, in both images, the analytical curves match the Monte-Carlo simulations.

The importance of using filtering and estimating $p_{0,1}(T)$ with (4.119) and (4.120) becomes evident from Figure 4.27. The probability density curves $p_{0,1}(T)$ of filtered observations, or more generally, band-limited observations, are more "damped" than those estimated with the standard analytical expressions (4.42) and (4.43). Using the last two expressions with "coloured" or filtered observations would result in a lower detection threshold and a potentially higher rate of false alarms.

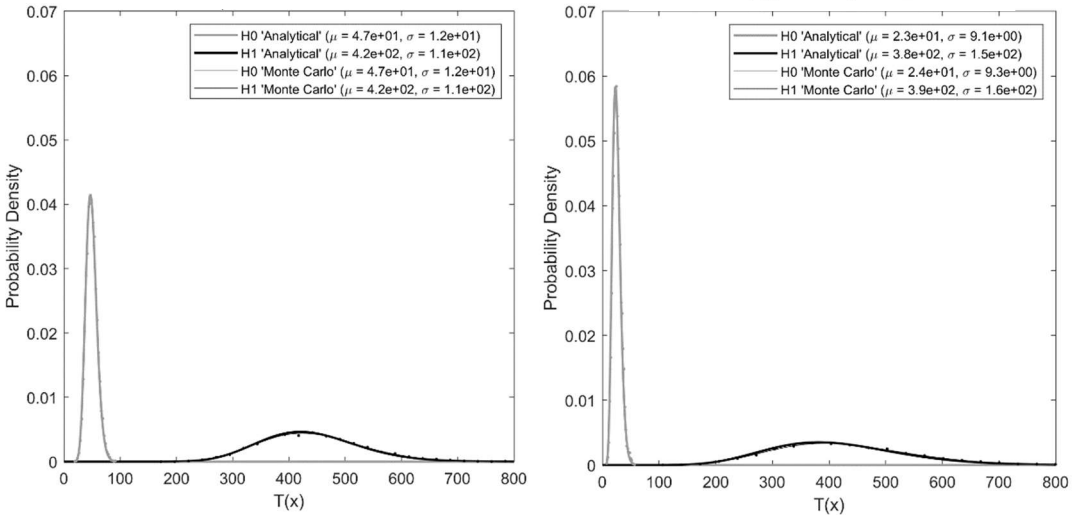

Figure 4.27 Probability density function of the *energy detector* for the null hypothesis (grey) and alternative hypothesis (black) for the cases of *LPF filtering* with $f_c = f_s/4$ (left) and *LPF filtering* with $f_c = f_s/8$ (right). Unfiltered $SNR = 6$ dB and record length before resampling of $N = N_s = 100$. The PDFs are obtained from the *general* analytical expressions (thick lines) and from Monte Carlo (MC) simulations (thin lines + dots). The deflection coefficients for the analytical and MC simulations are: $d_{AE,MC} = [31.8, 31.3]$ (left), $d_{AE,MC} = [39.6, 40]$ (right).

4.12 Estimating the Covariance of Filtered Signals with LOOC

4.12.1 Solutions to the High-Dimensional Problem

The high-dimensional problem was first introduced in Sub-Section 4.3.7. The problem is described as the loss of accuracy in the covariance matrix estimate when the number of observations M is lower than the number of variables N. In particular, the sample covariance matrix (SCM) converges to an optimal covariance estimator as M grows larger than N, but it becomes a poor estimator when M is comparable to or lower than N. In an estimator-correlator, the covariance matrix plays a key role in the calculation of the probability density functions $p_{0,1}(T)$ and test statistic $T(\mathbf{x})$. As a result, a poor covariance estimate will have a negative impact on the detector's performance.

Figure 4.28 shows the SCM built from identically distributed "coloured" Gaussian vectors with $N = 100$ variables and equal variance, for two different sets of observations. The image on the left corresponds to a near-perfect estimate of the covariance matrix

($M/N = 10$), where the SCM behaves as an optimal estimator. The image on the right shows a sub-optimal estimate of the covariance matrix ($M/N = 0.5$) that is "noisy" and badly conditioned. The latter is characterised by a "clustered" structure that deviates from the Toeplitz pattern of the population covariance matrix (PCM).

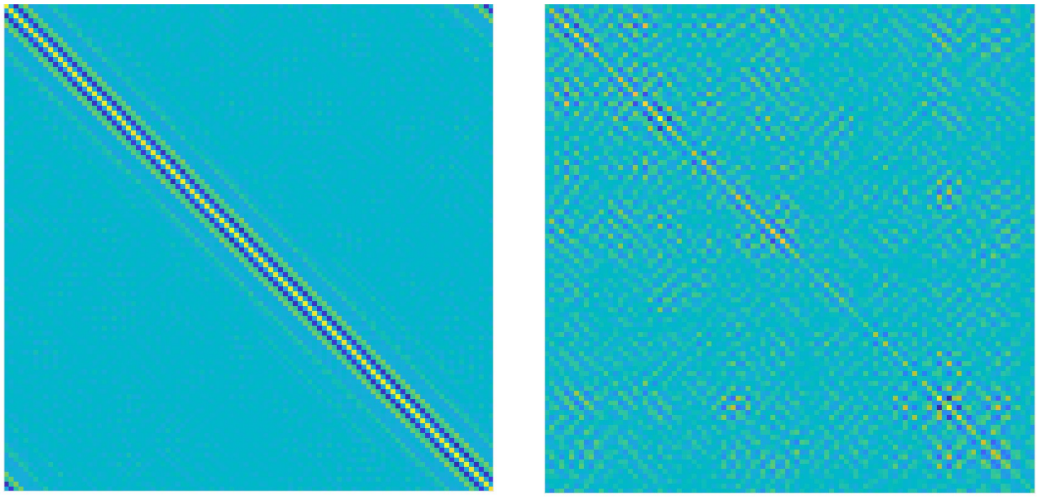

Figure 4.28 Covariance matrices computed from $M = 1000$ (left) and $M = 10$ (right) observations of white Gaussian noise with $N = 100$ variables filtered in an octave band centred at $f_s/4$.

It is evident from Figure 4.28 that using a limited number of observations M can have a negative effect on the covariance estimate. But how does M affect the performance of the NP detector? What is the lowest value of M/N that we can use before the performance of the detector is compromised? Figure 4.29 shows the threshold curves $\gamma'_{D,FA}(SNR)$ and test statistics $T(\mathbf{x})$ for the estimator-correlator in CGN. The detector is designed with the following sample covariance matrices: (1) a *signal covariance matrix* built from multiple one-third octave Gaussian vectors processed with a one-octave filter, and (2) a *noise covariance matrix* built from multiple white Gaussian vectors processed with a one-octave filter. The test statistics are calculated for observations of "noise only" (squares) and "signal plus noise" (circles), where the noise is white Gaussian and the signal is a third-octave band-limited Gaussian process. We will refer to these two test statistics as $T_w(\mathbf{x})$ and $T_t(\mathbf{x})$. The band-limited character of the signal and the filtering stage are included to produce observations similar to those found in practice. Note that the statistical properties of the *observed* and *target* ("covariance training") signal, and those of the observed and target noise, are purposefully identical.

For the optimal covariance estimate on the left image ($M/N = 10$), $T_{t,w}(\mathbf{x})$ match the $\gamma'_{D,FA}(SNR)$ curves. This behaviour is expected for a reliable covariance estimate. For the sub-optimal covariance estimate on the right image ($M/N = 0.1$), $T_t(\mathbf{x})$ diverges from $\gamma'_D(SNR)$ in the "small signal" region ($SNR < 0$ dB), while $T_w(\mathbf{x})$ diverges from $\gamma'_{FA}(SNR)$ throughout ($\forall SNR$). A value of M/N a few times lower than 1 is enough to produce an obvious mismatch between $\gamma'_{D,FA}(SNR)$ and $T_{t,w}(\mathbf{x})$. A positive bias in $T_{t,w}(\mathbf{x})$, like the one in the image, will cause higher false alarm rates.

Fortunately, the SCM is not the only covariance estimator available. In Section 4.5 we reviewed some of the most important methods that exist for estimating the covariance matrix in a high-dimensional setting. But what solution is best? Different estimators work better with different types of data, and choosing the most suitable is not straightforward. The number and complexity of the available estimators make it difficult to test them before deciding for one. A quick look into the pros and cons of the various methods can help discard those that are clearly unsuitable, but only implementation and testing can help choose the best estimator for the application.

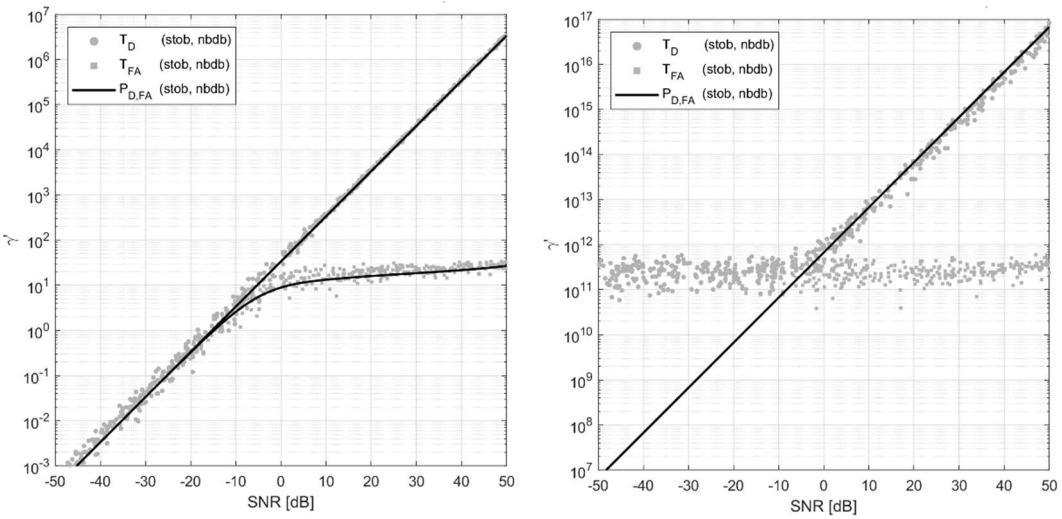

Figure 4.29 Detection threshold of *estimator-correlator in CGN* as a function of SNR for target probabilities of false alarm and miss $\alpha, \beta = 0.5$. The black curves represent $P_{D,FA}$ for a third-octave Gaussian signal and white Gaussian noise, both filtered in one octave band centred at $f_s/4$. The test statistic (grey) is calculated over simulated observations of "noise" and "signal plus noise", where the signal is a third-octave band-limited Gaussian process and the noise is white Gaussian. The observations have $N = 100$ variables and are filtered in a one-octave band centred at $f_s/4$. The SCM is used as covariance estimator and is computed from $M = 1000$ (left) and $M = 10$ (right). Noise variance $\sigma_w^2 = 1$.

With this in mind, we implemented several simple estimators that gave promising results when applied to different types of data. In particular, we coded and tested four *shrinkage estimators* with available analytical expressions: one that used a *scaled identity* target matrix (Ledoit & Wolf, 2004), two approximations to the latter based on the Rao-Blackwell theorem (RBLW) and an iterative method (OAS) (Chen et al, 2010), and two estimators designed for portfolio optimisation based on less conventional target covariance matrices (Ledoit & Wolf, 2003a; 2003b). We also tested two variations of the estimator from Ledoit & Wolf (2004) that used a *diagonal* target matrix, and a two-parameter target matrix with constant values for the diagonal and off-diagonal entries, both found at Ledoit's website (http://ledoit.net). For convenience, we will refer to the seven shrinkage estimators as 'param1', 'rblw', 'oas', 'stock', 'corr', 'diag', and 'param2'.

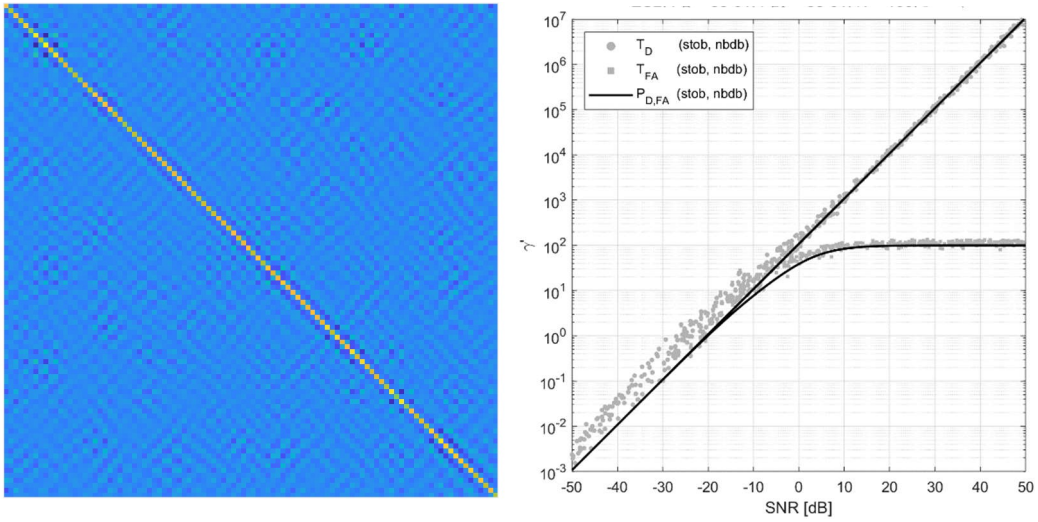

Figure 4.30 Covariance matrix of target signal and noise estimated with 'param1' method (Ledoit & Wolf, 2004) (left) and corresponding detection threshold of the estimator-correlator in CGN as a function of SNR for target probabilities $\alpha, \beta = 0.5$ (right). The covariance matrix is computed from $M = 10$ observations of white Gaussian noise with $N = 100$ variables filtered in one octave band centred at $f_s/4$. The test statistic (grey) is calculated over observations of "noise" and "signal plus noise" with $N = 100$ variables, where the signal is third-octave Gaussian and the noise is white Gaussian, both filtered in one octave band centred at $f_s/4$. Variance $\sigma_w^2 = 1$.

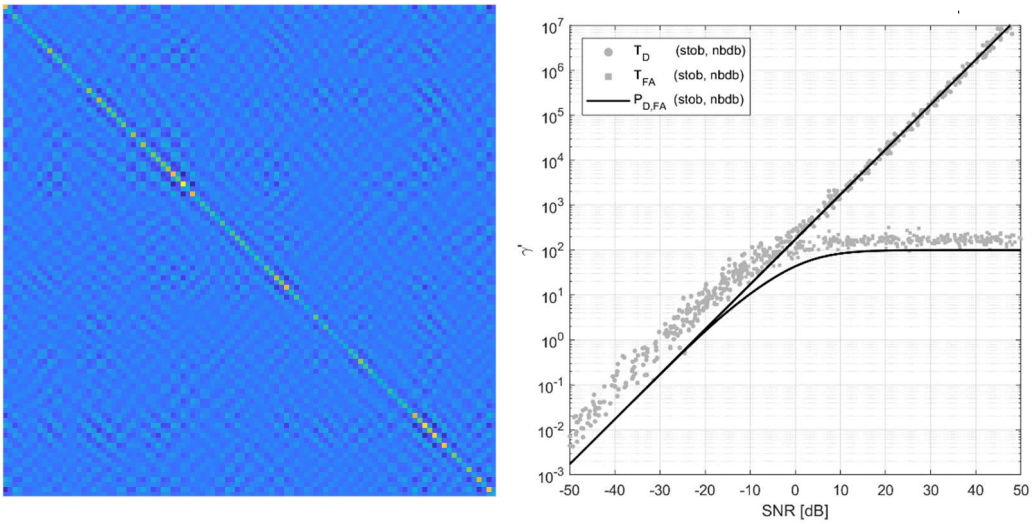

Figure 4.31 Covariance matrix of target signal and noise estimated with 'corr' method (Ledoit & Wolf, 2004) (left) and corresponding detection threshold of the estimator-correlator in CGN as a function of SNR for target probabilities $\alpha, \beta = 0.5$ (right). The covariance matrix is computed from $M = 10$ observations of white Gaussian noise with $N = 100$ variables filtered in one octave band centred at $f_s/4$. The test statistic (grey) is calculated over observations of "noise" and "signal plus noise" with $N = 100$ variables, where the signal is third-octave Gaussian and the noise is white Gaussian, both filtered in one octave band centred at $f_s/4$. Variance $\sigma_w^2 = 1$

Figure 4.30 and Figure 4.31 show the covariance matrix of the target noise estimated with 'param1' and 'corr' methods (left) along with the corresponding performance graph for the estimator-correlator in CGN (right). The results from the 'oas' and 'rblw', 'diag'

and 'stock', and 'param2' are very similar to those from 'param1' (Figure 4.30), 'corr' (Figure 4.31), and the SCM (Figure 4.29), respectively. None of the seven shrinkage methods are particularly effective at finding a reliable covariance matrix for filtered audio data in high-dimensions ($M \ll N$). Additionally, all methods (but 'param2') result in a fair match between $\gamma'_{D,FA}(SNR)$ and $T_{t,w}(\mathbf{x})$, at the expense of an alteration in the performance curves $\gamma'_{D,FA}(SNR)$ if compared to the optimal estimator (Figure 4.29, left).

The poor results from all seven methods can be justified by the naïve assumptions made on the structure of the covariance matrix. In a filtered signal, the covariance matrix is characterised by a wide, high-amplitude central band accompanied by multiple narrow, low-amplitude diagonal bands (see Figure 4.28, left). This pattern indicates a strong correlation between nearby variables, and a small (but non-negligible) correlation between variables far apart. The narrower the filter, the wider the central band.

A better estimation could be obtained with methods that make more suitable structural assumptions. Banding and tapering keep the central band but remove useful off-diagonal information. Estimators that incorporate *sparsity*, such as thresholding methods or the Graphical Lasso, can provide a better approximation to the real covariance matrix. However, for $M \ll N$ these sparse estimators can set many entries to zero, distorting the characteristic Toeplitz pattern in the covariance matrix of a filtered signal.

All these solutions make generic assumptions about the structure of the covariance matrix. But what if we knew the covariance structure to a greater detail? In that case, we could use a shrinkage estimator to find the optimal combination between the SCM and a known, unrestricted target covariance matrix. However, the majority of shrinkage estimators rely on a few basic target covariance structures that allow them to derive analytical expressions for the shrinkage coefficient. The question is: How do we calculate the shrinkage coefficient if the target covariance matrix does not follow a standard pattern? The solution is to use some form of optimisation; in practice, this is done with *cross validation*. Theiler (2012) proposes a computationally efficient shrinkage estimator that uses Leave-One-Out Cross Validation (LOOCV) to compute the shrinkage coefficient based on Hoffbeck & Landgrebe's (1996) manipulation of the likelihood function. Theiler referred to this estimator as the Leave-One-Out Covariance or *LOOC*.

In the following sub-subsection we describe the theory behind Theiler's method. In Sub-Section 4.12.3 we justify using a *filtered covariance matrix* as shrinkage target. In Sub-Section 4.12.4, the LOOC estimator is used to compute the covariance matrix of a coloured target signal and noise in a high-dimensional setting.

4.12.2 Covariance Shrinkage with LOOC

The idea behind cross validation is to avoid *overfitting*. Overfitting occurs when a model that is trained on a dataset performs poorly when tested on new data. For that reason,

a dataset is generally split into two or even three subsets, each with a different goal: *training* and *validating* (and/or *testing*) the model.

For the shrinkage estimator, the purpose of cross-validation is to find the shrinkage coefficient that produces the most likely covariance matrix based on the available observations. This is known as *parameter-tuning*. For the case of Leave-One-Out Cross-Validation (LOOCV), all observations *but one* are used to compute the covariance matrix for a given shrinkage coefficient, and the *remaining* observation is used to evaluate how likely that observation is to occur given the estimated covariance matrix. This is a two-step *training* and *validation* approach. The process is repeated with all the *independent* observations, to then obtain their combined probability for the specific shrinkage coefficient. The optimal shrinkage coefficient is the one that maximises the combined probability. This method is described in Theiler (2012), who presents the formulas to compute the shrinkage coefficient in an efficient way given a set of observations and a target covariance matrix. The algorithm is summarised below.

The probability of a N-variable, zero-mean Gaussian vector $\mathbf{x}$ of covariance matrix $\mathbf{\Sigma}$ is given by (4.38). This same expression can be used to describe the likelihood of observing a zero-mean, N-variable column vector $\mathbf{x}_m = \{x_1, x_2, \dots x_N\}^T$ given the estimated precision matrix or *shrinkage estimate* $\widehat{\mathbf{\Sigma}}_{\bar{m}}^{-1}$.

$$p(\delta, \mathbf{x}_m) = (2\pi)^{-N/2} \left|\widehat{\mathbf{\Sigma}}_{\bar{m}}\right|^{-1/2} \exp\left[-\frac{1}{2}\mathbf{x}_m^T \widehat{\mathbf{\Sigma}}_{\bar{m}}^{-1} \mathbf{x}_m\right] \tag{4.122}$$

$$\widehat{\mathbf{\Sigma}}_{\bar{m}} = \mathbf{S}_{\bar{m}}(1-\delta) + \delta\mathbf{F} \tag{4.123}$$

$$\mathbf{S}_{\bar{m}} = \frac{1}{M-1}\sum_{i=1, i\neq m}^{M} \mathbf{x}_i \mathbf{x}_i^T = \frac{M}{M-1}\mathbf{S} - \frac{1}{M-1}\mathbf{S}_m \tag{4.124}$$

$$\mathbf{S} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_m \mathbf{x}_m^T \tag{4.125}$$

$$\mathbf{S}_m = \mathbf{x}_m \mathbf{x}_m^T \tag{4.126}$$

where δ is the shrinkage coefficient, $\mathbf{F}$ is the target covariance matrix, $\mathbf{S}_{\bar{m}}$ is the SCM built from all observations except for $\mathbf{x}_m$, $\mathbf{S}$ is the SCM built from all M observations, and $\mathbf{S}_m$ is the SCM matrix built from the $\mathbf{x}_m$ observation. In line with the idea of cross-validation, different data is used for *training* (e.g., computation of $\widehat{\mathbf{\Sigma}}_{\bar{m}}$) and *verification* (e.g., calculation of likelihood of $\mathbf{x}_m$ given $\widehat{\mathbf{\Sigma}}_{\bar{m}}$).

The likelihood of $\mathbf{x}_m$ given $\widehat{\mathbf{\Sigma}}_{\bar{m}}$ is more commonly expressed as the negative logarithm of (4.122). This is known as the *negative log-likelihood*.

$$\mathcal{L}(\delta, \mathbf{x}_m) = -\log p(\delta, \mathbf{x}_m) = \frac{N}{2}\log 2\pi + \frac{1}{2}\log\left|\widehat{\mathbf{\Sigma}}_{\bar{m}}\right| + \frac{1}{2}\mathbf{x}_m^T \widehat{\mathbf{\Sigma}}_{\bar{m}}^{-1} \mathbf{x}_m \tag{4.127}$$

The likelihood in (4.122) is not a robust parameter when it comes to estimating the tuning coefficient δ, as it only accounts for one observation. A stronger parameter is given by the mean negative log likelihood of all observations.

$$\begin{aligned}\mathcal{L}\left(\delta, \mathbf{x}_{1,} \dots \mathbf{x}_{M}\right) &= \frac{1}{M}\sum_{m=1}^{M}\mathcal{L}(\delta, \mathbf{x}_{m}) \\ &= \frac{N}{2}\log 2\pi + \frac{1}{2M}\sum_{m=1}^{M}\left[\log\left|\widehat{\boldsymbol{\Sigma}}_{\bar{m}}\right| + \mathbf{x}_{m}^{T}\widehat{\boldsymbol{\Sigma}}_{\bar{m}}^{-1}\mathbf{x}_{m}\right]\end{aligned} \tag{4.128}$$

It is worth noting that $\mathcal{L}\left(\delta, \mathbf{x}_{1,} \dots \mathbf{x}_{M}\right)$ is equivalent to M^{-1} times the negative logarithm of the combined probability $p\left(\delta, \mathbf{x}_{1,} \dots \mathbf{x}_{M}\right)$.

The computation of the shrinkage coefficient δ is a simple optimisation problem where δ is selected to minimise $\mathcal{L}\left(\delta, \mathbf{x}_{1,} \dots \mathbf{x}_{M}\right)$, that is

$$\delta = \operatorname{argmin}\left\{\mathcal{L}\left(\delta, \mathbf{x}_{1,} \dots \mathbf{x}_{M}\right)\right\} \tag{4.129}$$

In order to compute $\mathcal{L}\left(\delta, \mathbf{x}_{1,} \dots \mathbf{x}_{M}\right)$ for a given δ, the determinant $\left|\widehat{\boldsymbol{\Sigma}}_{\bar{m}}\right|$ and inverse matrix $\widehat{\boldsymbol{\Sigma}}_{\bar{m}}^{-1}$ need to be calculated for each value of m. This requires a total computational effort of $\mathcal{O}(MN^3)$.

This number of calculations can be prohibitive, but simple manipulation can help reduce it. Hoffbeck and Landgrebe (1996) showed that the terms $\log\left|\widehat{\boldsymbol{\Sigma}}_{\bar{m}}\right|$ and $\mathbf{x}_{m}^{T}\widehat{\boldsymbol{\Sigma}}_{\bar{m}}^{-1}\mathbf{x}_{m}$ in (4.128) can be expressed as a function of a matrix $\mathbf{G}_{\delta}$, which only needs to be calculated once for a given δ. The method is summarized below.

First, the covariance estimator $\widehat{\boldsymbol{\Sigma}}_{\bar{m}}$ is decomposed as follows

$$\widehat{\boldsymbol{\Sigma}}_{\bar{m}} = \mathbf{G}_{\delta} - \beta\mathbf{x}_{m}\mathbf{x}_{m}^{T} \tag{4.130}$$

$$\mathbf{G}_{\delta} = M\beta\mathbf{S} - \delta\mathbf{F} \tag{4.131}$$

$$\beta = \frac{1-\delta}{M-1} \tag{4.132}$$

After applying the Sherman-Morrison formula to the precision covariance estimator $\widehat{\boldsymbol{\Sigma}}_{\bar{m}}^{-1}$, and multiplying the result by $\mathbf{x}_{m}^{T}$ on the left and $\mathbf{x}_{m}$ on the right, one arrives at the following expression for the term $\mathbf{x}_{m}^{T}\widehat{\boldsymbol{\Sigma}}_{\bar{m}}^{-1}\mathbf{x}_{m}$

$$\mathbf{x}_{m}^{T}\widehat{\boldsymbol{\Sigma}}_{\bar{m}}^{-1}\mathbf{x}_{m} = \frac{\mathbf{r}_{m}}{1-\beta\mathbf{r}_{m}} \tag{4.133}$$

$$\mathbf{r}_{m} = \mathbf{x}_{m}^{T}\mathbf{G}_{\delta}^{-1}\mathbf{x}_{m} \tag{4.134}$$

The determinant of the covariance estimator $\left|\mathbf{G}_{\delta} - \beta\mathbf{x}_{m}\mathbf{x}_{m}^{T}\right|$ can be written in a more convenient form by applying Sylvester's determinant theorem. The theorem states that given two matrices $\mathbf{A}$ and $\mathbf{B}$ of dimensions $[p, q]$ and $[q, p]$ and the identity matrix $\mathbf{I}$,

the relation $|\mathbf{I}+\mathbf{AB}| = |\mathbf{I}+\mathbf{BA}|$ is satisfied. By writing $\left|\widehat{\boldsymbol{\Sigma}}_{\bar{m}}\right|$ as $|\mathbf{G}_{\delta}| \cdot \left|\mathbf{I}-\beta\mathbf{G}_{\delta}^{-1}\mathbf{x}_m\mathbf{x}_m^T\right|$ and taking $\mathbf{A} = \beta\mathbf{G}_{\delta}^{-1}\mathbf{x}_m$ and $\mathbf{B} = \mathbf{x}_m^T$, the determinant $\left|\widehat{\boldsymbol{\Sigma}}_{\bar{m}}\right|$ becomes

$$\left|\widehat{\boldsymbol{\Sigma}}_{\bar{m}}\right| = |\mathbf{G}_{\delta}|(1-\beta\mathbf{r}_m) \tag{4.135}$$

The identities (4.133) and (4.135) are replaced in (4.128) to arrive at the more efficient expression from Hoffbeck and Landgrebe for the mean negative log likelihood

$$\mathcal{L}(\delta, \mathbf{x}_1, \ldots \mathbf{x}_M) = \frac{N}{2}\log 2\pi + \frac{1}{2}\log|\mathbf{G}_{\delta}| + \frac{1}{2M}\sum_{m=1}^{M}\left[(1-\beta\mathbf{r}_m) + \frac{\mathbf{r}_m}{1-\beta\mathbf{r}_m}\right] \tag{4.136}$$

This expression is equivalent to (4.128), except that in (4.136) the computational effort is reduced to $\mathcal{O}(N^3) + \mathcal{O}(MN^2)$, where $\mathcal{O}(N^3)$ corresponds to $|\mathbf{G}_{\delta}|$ and $\mathcal{O}(MN^2)$ to the M instances of $\mathbf{r}_m$. Theiler (2012) proposes two approximations to (4.136). We do not consider these as their computational improvement for large N is small.

4.12.3 The Shrinkage Target

The shrinkage algorithm described in the previous sub-section would not be complete without a suitable target covariance matrix $\mathbf{F}$ or *shrinkage target*. The type of data to be processed for detections can help determine which target matrix is most appropriate. But choosing a good shrinkage target, one that is a fair statistical representation of the expected observations, is not straightforward. One could opt for a shrinkage target with a *generic* covariance pattern (e.g., diagonal matrix) or instead select one with a more complex or *specific* structure (e.g., Toeplitz). While the first may provide a reasonable covariance estimate compatible with a wide range of data, the second will give an estimate that is optimal for a smaller data group. But what option is best in practice?

Underwater sound data typically consists of continuous low-frequency noise and discrete band-limited sound events. The scaled identity matrix is not the best shrinkage target for this type of data. The most obvious reason is that the scaled identity matrix represents the covariance of a white Gaussian process, while in practice signals and noise are coloured. This is particularly true when filtering is applied, for example, for improving the SNR of the target events before processing the data for detections. If we look at the behaviour of a shrinkage covariance estimator, shrinkage coefficients tend to be larger in a high-dimensional setting ($M \ll N$) and smaller in a low-dimensional setting ($M \gg N$). In other words, the covariance estimator will approximate the shrinkage target when $M \ll N$, but it will be close to the SCM when $M \gg N$. Choosing the scaled identity matrix as the shrinkage target has two effects: (1) if $M \ll N$, the estimator-correlator in CGN will behave like an energy detector (see Sub-Section 4.9.3), making the use of covariance matrices and complex processing unnecessary, and (2) if $M \approx N$, the LOOC algorithm will have enough information to recognise that the scaled identity matrix is not a fair statistical representation of the coloured observations, and the covariance estimator will approximate the SCM, which is sub-optimal in this case.

As an alternative to the scaled identity matrix, we propose using as shrinkage target the SCM of a "low-dimensional" ($M \gg N$) set of white Gaussian *training* observations processed with the same filter and filtering algorithm used on the *detection* observations. We will refer to this shrinkage target as the *filtered target covariance*. Under background noise conditions typically met in underwater acoustic recordings, the proposed shrinkage target outperforms the scaled identity matrix in the high-dimensional ($M \ll N$) and near high-dimensional ($M \approx N$) settings. Note that all shrinkage targets perform well in the low-dimensional setting ($M \gg N$), where the LOOC estimator approximates the SCM.

Let's explain why the filtered target is a better shrinkage target in practice than the scaled identity matrix. In underwater acoustics, the background noise is characterised by a low frequency spectrum with a -6 dB/oct roll-off from 500 Hz, with most of the energy concentrated below 100 Hz (Wenz, 1962). When the data is filtered to improve the SNR of the target signal (see Sub-Section 4.10.2), the spectrum of the filtered noise becomes approximately flat within the passband region. In this scenario, the *filtered target* covariance is a fair statistical representation of the filtered background noise. The result is a covariance estimate $\widehat{\mathbf{\Sigma}}$ that approximates the filtered target $\mathbf{F}$ across the entire range of dimensionality ratios M/N. Put simply, the shrinkage algorithm with the proposed filtered target is a near-optimal covariance estimator for the background noise.

A near-optimal covariance estimator for the target signal may not be possible to obtain. The reason is that any filter applied to the data should be wider than the dominant spectral region of the target signal, to guarantee that the target signal and noise are not statistically identical, and in that way avoid a dramatic increase in the false alarm rate (see 'CGS and CGN with Similar Covariance Matrices' in Sub-Section 4.9.3). Therefore, the filter target covariance cannot be used as a reliable representation of the filtered signal due to the "coloured" response of the signal within the filter's band. In practice, as it will be demonstrated in the next sub-section, having a near-optimal covariance estimator for the background noise is enough to ensure that the estimator-correlator in CGN works correctly in the "low signal" region, where inaccuracies in the covariance estimate cause the greatest impact.

In the next sub-section we study the behaviour of the estimator-correlator in CGN with Theiler's covariance estimator. The results from the SCM, LOOC estimator with scaled identity matrix, LOOC estimator with filtered target, and Ledoit & Wolf (2004) 'param1' method are compared.

4.12.4 Application of LOOC Estimation

Figure 4.32, Figure 4.33 and Figure 4.35 show the behaviour of the estimator-correlator in CGN for four covariance estimators in *high*, *near-low*, and *low* dimensional settings ($M/N = [0.1, 1, 10]$). Each figure includes four $\gamma'(SNR)$ graphs, one per covariance estimator. From left to right and top to bottom, the images correspond to the SCM, the 'param1' method from Ledoit and Wolf (2004), the LOOC method with filtered target,

and the LOOC method with scaled identity matrix. In all four images, the same type of signal and noise is used to simulate the *training* and *detection* vectors. The signal is a third-octave band-limited Gaussian process and the noise is white Gaussian; both are treated with a one-octave bandpass filter. The threshold curves of detection $\gamma'_D(SNR)$ and false alarm $\gamma'_{FA}(SNR)$ are calculated for target probabilities of false alarm and miss $\alpha, \beta = 0.5$. The tests statistics are computed over simulations of "noise" (T_w, squares) and "signal plus noise" (T_t, circles).

A poor match between $T_{t,w}$ and $\gamma'_{D,FA}$ is a sign of an inadequate covariance estimator. For instance, if T_t is consistently smaller than γ'_D, the rate of *miss* will increase. Similarly, the rate of *false alarms* will increase if T_w is consistently larger than γ'_{FA}. Note, however, that the quality of a covariance estimator is not only determined by how close $T_{t,w}$ is to $\gamma'_{D,FA}$, but also by how well the detection performance is preserved. The ratio γ'_D/γ'_{FA} is one way of quantifying the detection performance for a given SNR. For example, let's think about a shrinkage covariance estimator with a scaled identity target $\check{\mathbf{F}} = \mathbf{I}$. In the high-dimensional setting ($M \ll N$), where $\widehat{\boldsymbol{\Sigma}} \approx \mathbf{F}$, the scaled identity matrix turns the estimator-correlator in CGN into a simple energy detector. In this situation, $T_{t,w}$ will match $\gamma'_{D,FA}$, but at the expense of a deterioration in the detection performance γ'_D/γ'_{FA}. It is worth noting that a close match between $T_{t,w}$ and $\gamma'_{D,FA}$ will also occur with non-target signals and noise, as a result of the estimator-correlator losing its ability to distinguish between target and non-target signals and noise as its behaviour becomes closer to that of the energy detector.

The optimal covariance estimator is given by the SCM in a low-dimensional setting (see Figure 4.35, top-left). A good covariance estimator will lead to a detection performance similar to that of the SCM in low-dimensions ($M \gg N$) and a reasonable match between $T_{t,w}$ and $\gamma'_{D,FA}$, for any dimensionality ratio M/N. In the following sub-subsections, we compare the behaviour of the estimator-correlator in CGN for four covariance estimators in three different dimensionality scenarios ($M/N = [0.1, 1, 10]$).

High-Dimensional Setting ($M \ll N$)

The SCM in the high-dimensional setting is an example of a poor covariance estimator. Figure 4.32 shows $\gamma'_{D,FA}$ and $T_{t,w}$ for $M/N = 0.1$. In this scenario, where $M \ll N$, the SCM (top-left image) is a "noisy" estimator. The disagreement between γ'_{FA} and T_w ($\forall\, SNR$), or between γ'_D and T_t in the "small signal" region ($SNR < 0$ dB), is evident. It is also clear the difference with the detection performance curves $\gamma'_{D,FA}$ from the optimal covariance estimator (see Figure 4.29, left). These results show that the SCM is a poor covariance estimator for high dimensions and alternative solutions are needed.

The one-parameter, analytical estimator from Ledoit & Wolf (2004) produces a dramatic improvement in the detector's behaviour, compared to the SCM (see Figure 4.32, top-right). The graph shows a reasonable agreement between $\gamma'_{D,FA}$ and $T_{t,w}$, but some deterioration in performance compared to the optimal estimator. The results from the

LOOC method with scaled identity target (bottom-right image) are similar, but with a slightly better agreement between $\gamma'_{D,FA}$ and $T_{t,w}$, particularly in the "large signal" region. The LOOC method with filtered target (bottom-left image) shows the best agreement between $\gamma'_{D,FA}$ and $T_{t,w}$, with a detection performance similar to that of the two alternative estimators.

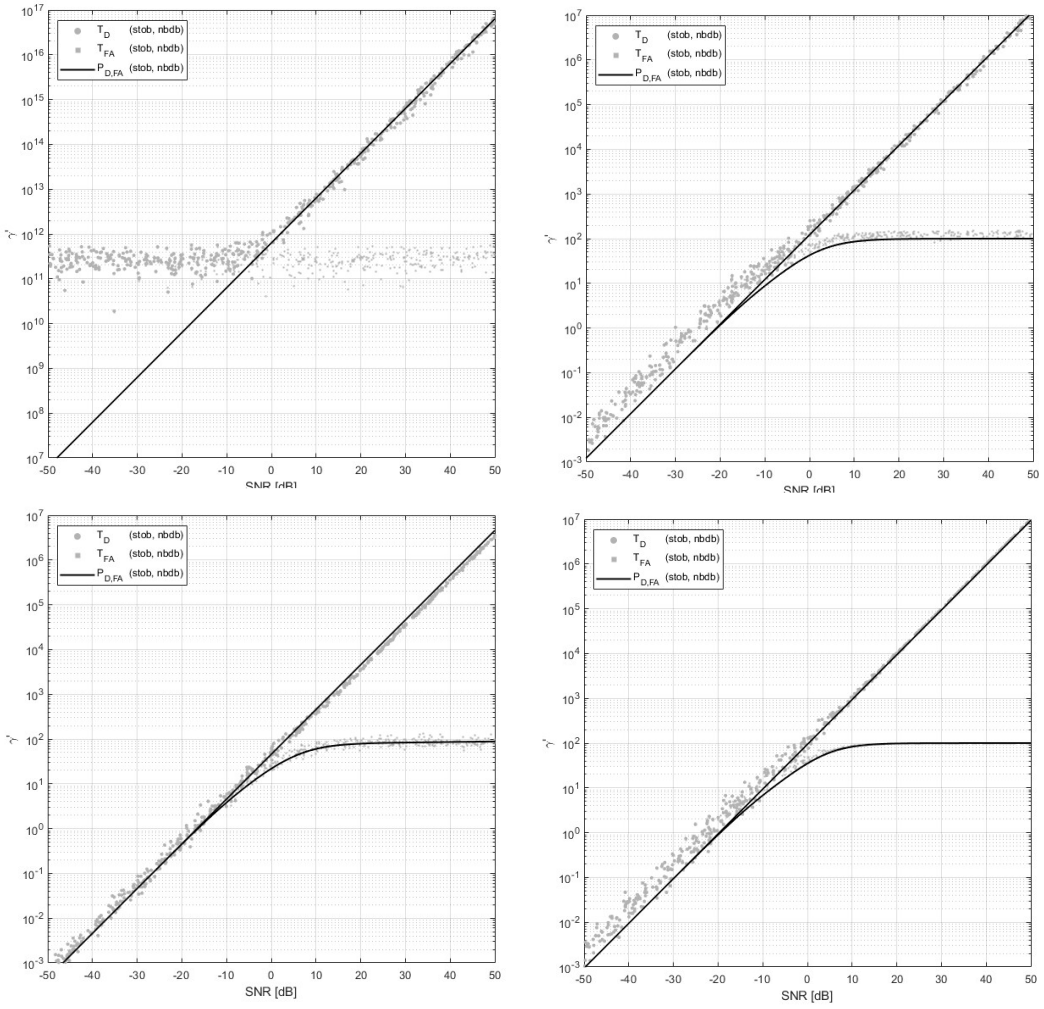

Figure 4.32 Detection threshold of *estimator-correlator in CGN* as a function of SNR for target probabilities of false alarm and miss $\alpha, \beta = 0.5$. The black curves represent $P_{D,FA}$ for a Gaussian target signal with a third-octave band-limited spectrum and white Gaussian noise, both filtered in one octave band centred at $f_s/4$. The test statistic is calculated over simulated *detection* observations of "noise" T_w (squares) and "signal plus noise" T_t (circles), where the signal is a third-octave band-limited Gaussian process and the noise is white Gaussian. The *detection* observations have $N = 100$ variables and are filtered in one octave band centred at $f_s/4$. The estimated covariance matrices for the target signal and noise are computed from $M = 10$ *training* observations ($M/N = 0.1$) using the SCM (top left), the 'param1' method (top right), the LOOC method with one-octave filtered target (bottom left), and the LOOC method with scaled identity target (bottom right). Noise variance $\sigma_w^2 = 1$.

This is a key result, as it shows that using the LOOC method with a target covariance matrix that is statistically representative of the observed noise, but not necessarily of the observed signal, is sufficient to guarantee an acceptable behaviour of the detector. When we say that the target covariance is "statistically representative" of the noise (but not the signal) we are simply stating that the true covariance matrix of the observed noise (after filtering) $\mathbf{\Sigma}_w$ is close to the filtered target covariance **F.** In the example, **F** is generated from one-octave filtered white Gaussian vectors. Filtering the observations in one octave means that the original white noise is one-octave band-limited ($\mathbf{\Sigma}_w \approx \mathbf{F}$), while the original third-octave filtered signal is not altered ($\mathbf{\Sigma}_s \neq \mathbf{F}$).

There is one downside to the LOOC approach with filtered target: when $M \ll N$, the normalised covariance estimators for the target signal and noise are notably similar ($\check{\mathbf{\Sigma}}_s \approx \check{\mathbf{F}}$, $\check{\mathbf{\Sigma}}_w \approx \check{\mathbf{F}}$). This makes the estimator-correlator in CGN behave as a *filtered energy detector* with higher detection rates for certain non-target signals (see 'CGS and CGN with Similar Covariance Matrices' within Sub-Section 4.9.3). This deterioration in detection performance is the price paid for using a limited number of training vectors, which will compromise the accuracy of the covariance matrix regardless of the estimator.

Near Low-Dimensional Setting ($M \approx N$)

Let's now see how the detector behaves in a near low-dimensional setting ($M \approx N$). In this scenario, more training data is available to build the covariance matrices but the amount of information is not enough to ensure optimality for the SCM. Figure 4.33 shows $\gamma'_{D,FA}$ and $T_{t,w}$ for $M/N = 1$. The SCM (top-left image) is full rank, still $\gamma'_{D,FA}$ and $T_{t,w}$ are not well matched, proving that only when the number of training vectors greatly exceeds the number of variables ($M \gg N$) it is safe to use the SCM. Note, however, that the performance curves $\gamma'_{D,FA}$ are practically identical to those from the optimal covariance estimator (Figure 4.35, top-left).

For $M \approx N$, one may expect the alternative covariance estimators to improve over the SCM, given the larger amount of training information available, but this is not necessarily the case for all of them. The one-parameter, analytical estimator from Ledoit & Wolf (2004) or 'param1' method (top-right image) provides a good match between $\gamma'_{D,FA}$ and $T_{t,w}$, but with a worse detection performance compared to the optimal estimator. On the other hand, the results from the LOOC method with scaled identity matrix (bottom-right image) are notably similar to those from the SCM (top-left image), except for a slight deterioration in detection performance for high SNR. Unlike these two alternative estimators, the LOOC method with filtered target (bottom-left image) improves over the SCM, with results that are comparable to those from the optimal covariance estimator.

But what is causing the differences between the LOOC method with scaled identity target and the 'param1' method, if both shrinkage solutions use the same shrinkage target? Tests show that the 'param1' method returns shrinkage coefficients that are

considerably larger than those obtained by cross-validation, producing a covariance estimate that is close to the identity matrix, thus making the estimator-correlator in CGN behave as an energy detector.

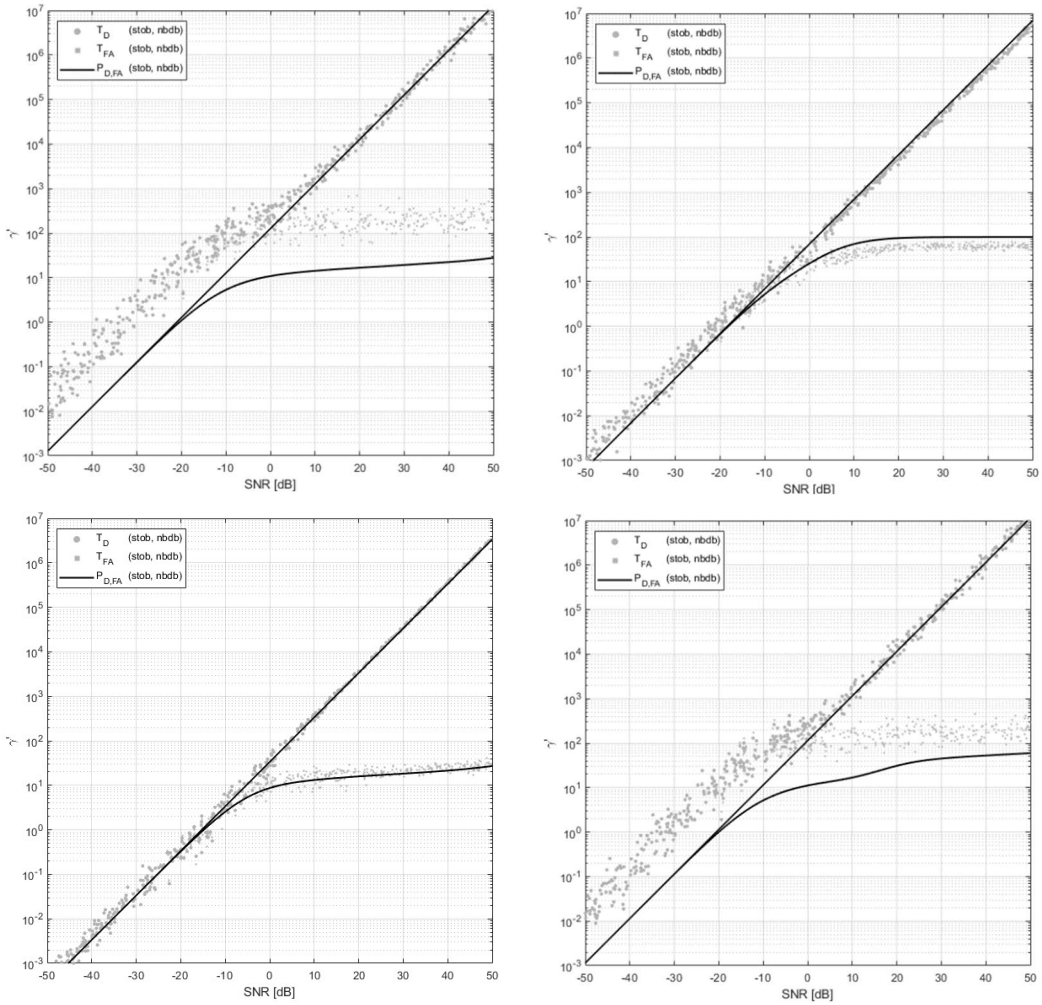

Figure 4.33 Detection threshold of *estimator-correlator in CGN* as a function of SNR for target probabilities of false alarm and miss $\alpha, \beta = 0.5$. The black curves represent $P_{D,FA}$ for a Gaussian target signal with a third-octave band-limited spectrum and white Gaussian noise, both filtered in one octave band centred at $f_s/4$. The test statistic is calculated over simulated *detection* observations of "noise" T_w (squares) and "signal plus noise" T_t (circles), where the signal is a third-octave band-limited Gaussian process and the noise is white Gaussian. The *detection* observations have $N = 100$ variables and are filtered in one octave band centred at $f_s/4$. The estimated covariance matrices for the target signal and noise are computed from $M = 100$ *training* observations ($M/N = 1$) using the SCM (top left), the 'param1' method (top right), the LOOC method with one-octave filtered target (bottom left), and the LOOC method with scaled identity target (bottom right). Noise variance $\sigma_w^2 = 1$.

Figure 4.34 shows the evolution of the shrinkage coefficient δ with the dimensionality ratio M/N for the three alternative covariance estimates of the target signal (left) and target noise (right). The shrinkage coefficient decreases from a near-unit value as M/N

increases. The more similar the shrinkage target is to the SCM when $M > N$, the larger the shrinkage coefficient should be.

In the LOOC method with scaled identity matrix (dark grey line), the rapid decrease in the shrinkage coefficient on both images proves that the scaled identity matrix is not a fair statistical representation of the target signal and noise. It is also clear that the shrinkage coefficient is unrealistically large for the 'param1' method, turning the estimator-correlator into something closer to an energy detector. In the LOOC method with filtered target (black line), the shrinkage coefficient for $\check{\mathbf{\Sigma}}_w$ (right image) remains close to unity, due to the similitude between target noise and filtered target ($\check{\mathbf{S}}_w \approx \check{\mathbf{F}}$).

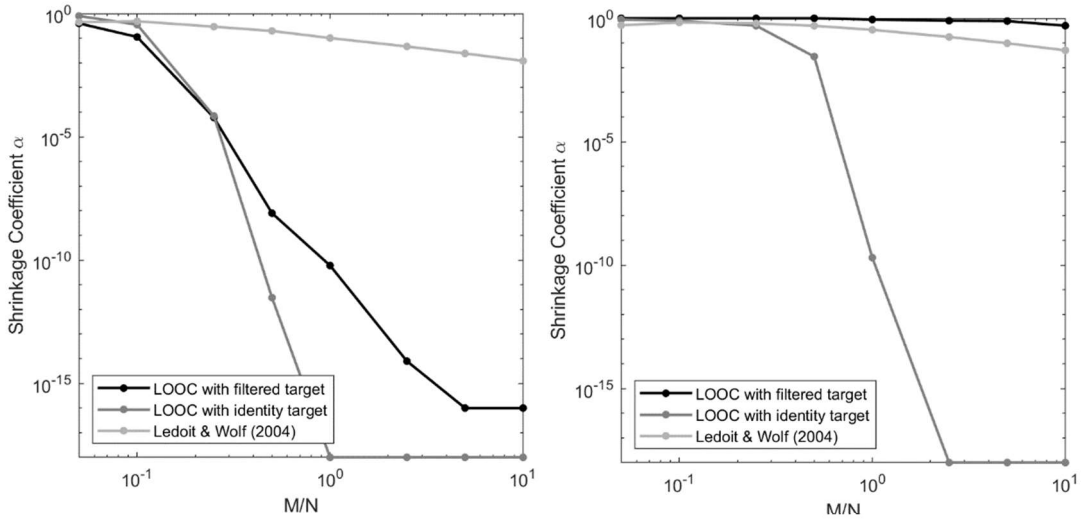

Figure 4.34 Dependence of shrinkage coefficient δ on dimensionality ratio M/N for three types of shrinkage estimator: LOOC with filtered target (black), LOOC with scaled identity target (dark grey), and analytical one-parameter method from Ledoit & Wolf (light grey). The results correspond to the covariance estimates of the *target signal* (left) and *target noise* (right) used in figures Figure 4.32, Figure 4.33 and Figure 4.35.

Low-Dimensional Setting ($M \gg N$)

In a low-dimensional setting ($M \gg N$) any shrinkage covariance estimator should be expected to converge to the optimal SCM. Figure 4.35 shows $\gamma'_{D,FA}$ and $T_{t,w}$ for $M/N = 10$. The SCM (top-left image) is the optimal covariance estimator, a "gold reference" for any good estimator. The results for the LOOC method with filtered target (bottom-left image) are virtually identical to those from the optimal estimator. For the LOOC method with scaled identity target (bottom-right image), the match between $\gamma'_{D,FA}$ and $T_{t,w}$ is excellent, except for a slight deterioration in performance at high SNR. The 'param1' method (top-right image) shows the largest deterioration in performance and a comparatively poor match between $\gamma'_{D,FA}$ and $T_{t,w}$.

The extent by which the 'param1' method alters the performance of the estimator-correlator in CGN, even in a low-dimensional setting, shows that this shrinkage

covariance estimator is not well suited for the calculation of large and complex covariance matrices typically found in audio recordings. The cross-validation approach from Theiler provides more realistic covariance estimates which better preserve the optimal detection performance of the estimator-correlator in CGN. Theiler's LOOC method has also proved to be an excellent covariance estimator under typical underwater noise conditions when used with our proposed filtered target covariance.

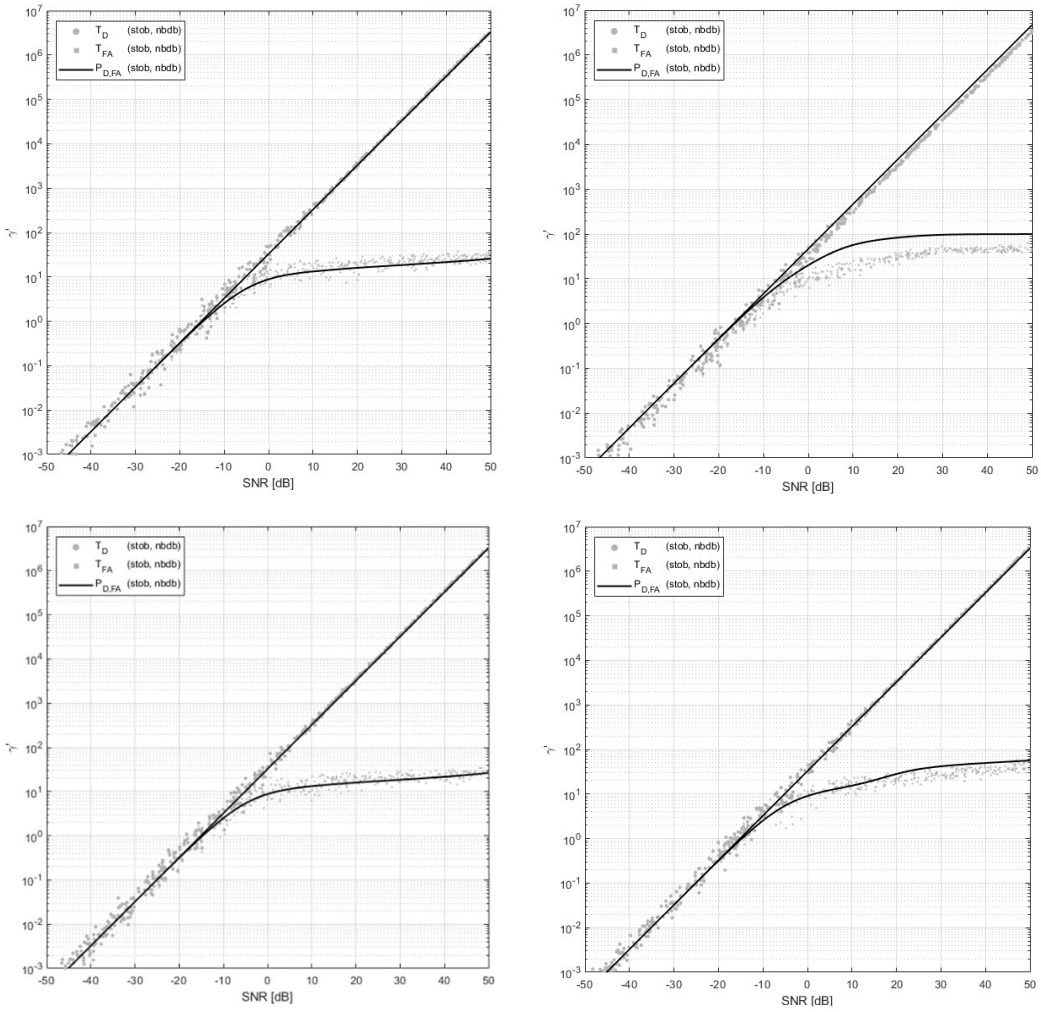

Figure 4.35 Detection threshold of *estimator-correlator in CGN* as a function of SNR for target probabilities of false alarm and miss $\alpha, \beta = 0.5$. The black curves represent $P_{D,FA}$ for a Gaussian target signal with a third-octave band-limited spectrum and white Gaussian noise, both filtered in one octave band centred at $f_s/4$. The test statistic is calculated over simulated *detection* observations of "noise" T_w (squares) and "signal plus noise" T_t (circles), where the signal is a third-octave band-limited Gaussian process and the noise is white Gaussian. The *detection* observations have $N = 100$ variables and are filtered in one octave band centred at $f_s/4$. The estimated covariance matrices for the target signal and noise are computed from $M = 1000$ *training* observations ($M/N = 10$) using the SCM (top left), the 'param1' method (top right), the LOOC method with one-octave filtered target (bottom left), and the LOOC method with scaled identity target (bottom right). Noise variance $\sigma_w^2 = 1$.

4.13 Implementation of the NP Detectors in ABACUS

In the previous sections, we have analysed the theory and potential applicability of classical decision theory to the detection of target events in audio recordings. Our motivation comes from the need that currently exists in the fields of underwater acoustics and noise impact assessment for a quick, effective, and open-access detection algorithm. The aim was to develop a solution that could automatically detect target acoustic events spread across hours of underwater sound data without having to manually select a detection threshold.

With this in mind, we have developed a Neyman-Pearson (NP) detection algorithm in MATLAB. The code is part of the underwater sound analysis software ABACUS, presented in the previous chapter. The software can be downloaded from GitHub[33].

The NP algorithm can be split into three modules or processing stages: (1) *pre-processing*, for the one-off calculation of parameters required by the detection algorithm, (2) *audio processing*, which applies segmentation and audio data conditioning for improving detection performance and speed and, (3) *detection*, which uses pre-computed data to map every conditioned audio segment as detection ("signal plus noise") or non-detection ("noise"). The three stages are described in the following sub-subsections.

4.13.1 Pre-Processing

The pre-processing module computes the detection performance curves ($p_{0,1}$, $P_{D,FA}$) and eigendata. The detection algorithm needs this information to calculate the *detection threshold* and *test statistic* for each conditioned audio segment.

Running this module can be time-consuming, particularly for the estimator-correlator, for which raw scores, covariance, and modal matrices need to be calculated. This process can take several minutes. Nevertheless, having defined the detection parameters, the pre-processing module only needs to run once. Figure 4.36, Figure 4.37 and Figure 4.38 present the flow diagrams of this stage for the three NP detectors.

Figure 4.36 Flow diagram of the *pre-processing* module for the energy detector.

Figure 4.37 Flow diagram of the *pre-processing* module for the estimator-correlator in WGN.

[33] https://github.com/Guille340/ABACUS

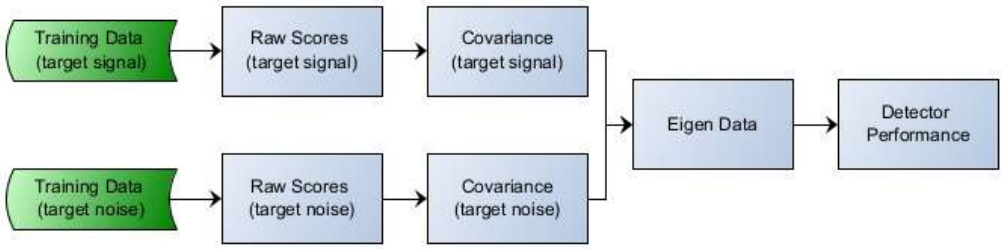

Figure 4.38 Flow diagram of the *pre-processing* module for the estimator-correlator in CGN.

The "Detector Performance" module computes the probability densities of the null and alternative hypotheses and the probabilities of detection and false alarm ($p_{0,1}(T)$, $P_{FA,D}(T)$). The detection thresholds γ' are calculated from $P_{D,FA}(T)$ for a given probability of false alarm α. There is one pair of performance curves and one detection threshold for each observation, that is, for each combination of variances of target signal and noise. However, computing $p_{0,1}(T)$ for each audio segment can be extremely inefficient. Luckily, a set of normalised $\check{p}_{0,1}(T)$ curves can be computed for a range of likely SNR (e.g., -50 dB to 50 dB) and a reference noise variance $\sigma_w^2 = 1$. Then, $\check{p}_{0,1}(T)$ only needs to be scaled by the noise variance σ_w^2 to arrive at the $p_{0,1}(T)$ curves (and corresponding threshold γ') for the observation. Due to the formulation chosen for the test statistic in sections 4.6, 4.7, and 4.8, scaling is only needed for the energy detector and estimator-correlator in WGN.

The normalised performance curves $\check{p}_{0,1}(T)$ for the energy detector can be easily calculated with the central Chi-squared distribution, as in (4.42) and (4.43). The number of variables (DOF) and the cutoff frequencies of the filter are the only parameters needed to compute these curves (see Section 4.11). This makes the calculation of $\check{p}_{0,1}(T)$ notably quicker compared to the estimator-correlator (a few seconds versus a few tens of seconds). The pre-processing for the energy detector is the simplest and fastest of all three NP algorithms, as it does not rely on covariance matrices or eigendata, as shown in Figure 4.36.

More challenging is the calculation of the normalised performance curves $\check{p}_{0,1}(T)$ for the estimator-correlator. The formulas for computing the probabilities of false alarm and detection for *WGN* (see (4.64) and (4.65)) and *CGN* (see (4.82) and (4.83)) include a double infinite integral. An exact solution is not possible, but an accurate estimate of $p_{0,1}(T)$ and $P_{D,FA}(T)$, within a small tolerance, can be obtained by approximating to *finite integrals* the infinite integrals in the *original* and *transformed* domains. The term within the outer integral is the probability density function $p(t)$ and represents the original domain, where the independent variable t is the *test statistic*. The term within the inner integral represents the transformed domain. For simplicity, we will refer to the transformed domain as *spectrum* and to its independent variable as *frequency* (f). The reason why the infinite integrals can be approximated to finite integrals is that the finite product within the inner integral (i.e., the spectrum) is a Gaussian-like curve centred at $f = 0$, and the term within the outer integral (i.e., the PDF) is a Chi-squared

curve defined for $t > 0$. Both reach a near-zero amplitude above a particular value of f and t. We can take advantage of this and define the integration limits of the inner and outer integrals f_{max} and t_{max} as the values of f and t at which the amplitudes of the spectrum and test statistic are k times their respective maxima.

The relative amplitude k can be selected to achieve a certain accuracy for the PDF. A larger k produces a more accurate PDF but results in longer processing times ($k = 10^4$ for a tolerance of $\varepsilon = 10^{-5}$ in the PDF). The integration limits $(-f_{max}, f_{max})$ and $(0, t_{max})$ determine the resolution of the PDF (t_{res}) and the spectrum (f_{res}), given by

$$t_{res} = 1/(2f_{max}) \tag{4.137}$$

$$f_{res} = 1/t_{max} \tag{4.138}$$

Note that using $f_{res} > 1/t_{max}$ will induce periodicity in the estimated PDF. Note also that by limiting the inner integral to $(-f_{max}, f_{max})$ we are essentially modulating the spectral response by a rectangular function, producing *leakage* in the original domain, thus widening the PDF and introducing secondary lobes. The impact of leakage will be larger the more energy is rejected from the Gaussian-like spectrum by limiting it to $|f_{max}|$. Similarly, a low value of t_{max} will have a negative effect on the accuracy of the PDF. In particular, it will produce *aliasing* in the original domain, making the rejected portion of the PDF overlap within the selected range $(0, t_{max})$. Choosing an appropriate (large-enough) value for k will virtually eliminate these effects.

From (4.137) and (4.138), the number of points for the PDF ($2f_{max}/f_{res}$) and the spectrum (t_{max}/t_{res}) is $2f_{max}t_{max}$. Since f_{max} and t_{max} are inversely proportional, the number of points is consistently low for all SNR. This makes the estimation of the performance curves feasible, as the number of operations is manageable even when the integration limit of either of the integrals is large.

The estimator-correlator in WGN includes three extra calculations that make the process more computationally involved: (1) the matrix of raw scores, (2) the covariance matrix, and (3) the eigendata for the *target signal* (see Figure 4.37). For the estimator-correlator in CGN, the matrix of raw scores and covariance matrix for the *target noise* are also processed (see Figure 4.38).

The "Raw Scores" module generates the *matrix of raw scores* from a set of training observations. The training data for the target signal comprises a number of high-SNR audio clips of the target event, manually extracted from available datasets. The training data for the target noise can be automatically extracted from audio files containing only background noise. The noise training data must be statistically representative of the noise found during the active period of the target signal. The module collates the training observations and ensures that they all have the same length and sampling rate.

The "Covariance" module computes the *covariance matrix* from the matrix of raw scores by applying one of nine available shrinkage estimators: 'param1', 'param2', 'rblw', 'oas', 'stock', 'corr', 'diag', 'looc', or 'sample'. The last two refer to Theiler's LOOC method and SCM. The *training* observations in the raw scores' matrix are first resampled and filtered in the same way as the *detection* observations (see next sub-section). Then, the selected shrinkage algorithm is applied to the processed raw scores to obtain the covariance estimate.

The "Eigen Data" module generates the *eigendata* from the covariance matrices. The eigendata comprises the eigenvalues and eigenvectors that will later be used to compute the performance curves and test statistics. In particular, for the case of the estimator-correlator in WGN, the module returns the normalised eigenvalues and eigenvectors of the signal covariance matrix $(\check{\boldsymbol{\lambda}}_s, \mathbf{V}_s)$. For the estimator-correlator in CGN, the eigendata contains the normalised eigenvalues and eigenvectors of the noise covariance matrix and the combined matrix $\mathbf{B}$ $(\check{\boldsymbol{\lambda}}_w, \mathbf{V}_w, \check{\boldsymbol{\lambda}}_B, \mathbf{V}_B)$.

4.13.2 Audio Processing

The audio processing module divides the audio file into *detection* observations which are then processed to enhance their SNR. This module also estimates the variance of the noise and signal, needed to later determine the detection threshold $\gamma'(\mathbf{x})$ and test statistic $T(\mathbf{x})$. The flow diagram is presented in Figure 4.39. The same calculation steps are applied to the three NP detectors.

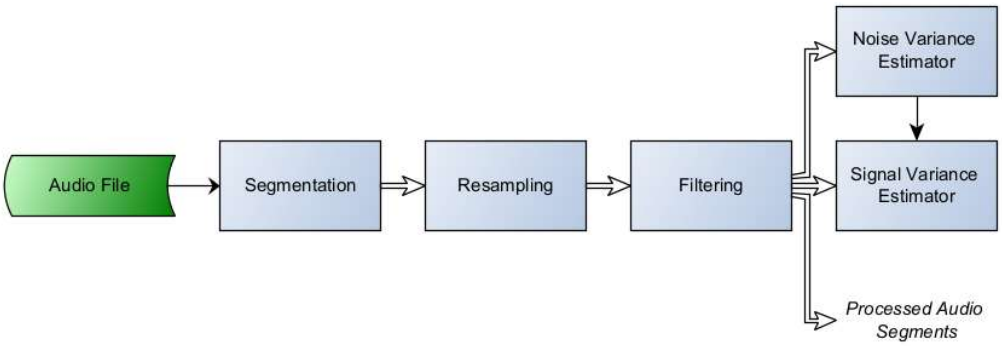

Figure 4.39 Flow diagram of the *audio processing* module. The same processing is applied to the energy detector and the estimator-correlator (WGN, CGN).

In the "Segmentation" stage, the audio file is split into segments of specified duration. The "Resampling" and "Filtering" stages then reduce the sampling rate and limit the bandwidth of the observation to improve the SNR, with a resulting positive impact on the performance and speed of the detector.

The "Noise Variance Estimator" module calculates the variance of the noise in the audio file. The algorithm assumes that the audio file contains more sections of "noise" than it does of "signal plus noise". In underwater recordings for sound source characterisation

this is a fair assumption, since target events are generally short with duty cycles significantly smaller than 0.5. With this premise, the most frequent amplitudes will correspond to the background noise. The algorithm computes the histogram of variances from all segments in the audio file, to then estimate the noise variance as the variance in the histogram associated with the maximum count. However, a somewhat larger value is recommended to account for the standard deviation of the Gaussian-like curve (e.g., 10% of maximum count).

The "Signal Variance Estimator" module calculates the variance for each segment by subtracting the estimated noise variance from the variance of the observation. The uncertainty of this estimation is large for $SNR \ll 0\,\mathrm{dB}$. This results in estimated SNR values that are very low or close to $-\infty$. This is not an issue in sound source characterisation, the main focus of this thesis, as observations with $SNR < -10\,\mathrm{dB}$ do not contain enough information of the target signal for them to be useful in practice.

4.13.3 Detection Algorithm

In the "Detection" module, the detection threshold $\gamma'(\mathbf{x})$ and test statistic $T(\mathbf{x})$ are calculated for each processed audio segment $\mathbf{x}$ and compared to map the observation into the right decision ("detection" for $T(\mathbf{x}) \geq \gamma'(\mathbf{x})$ or "noise" for $T(\mathbf{x}) < \gamma'(\mathbf{x})$). The flow chart for the energy detector and estimator-correlators is shown in Figure 4.40 and Figure 4.41.

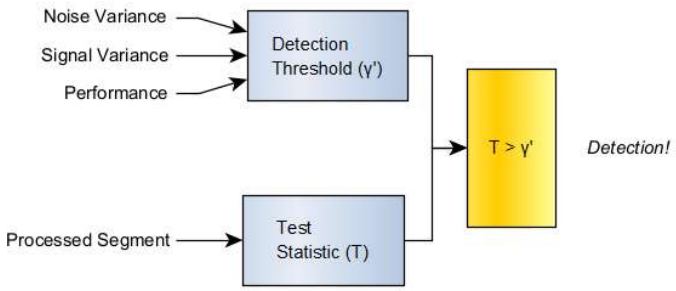

Figure 4.40 Flow diagram of the *detection* module for the energy detector.

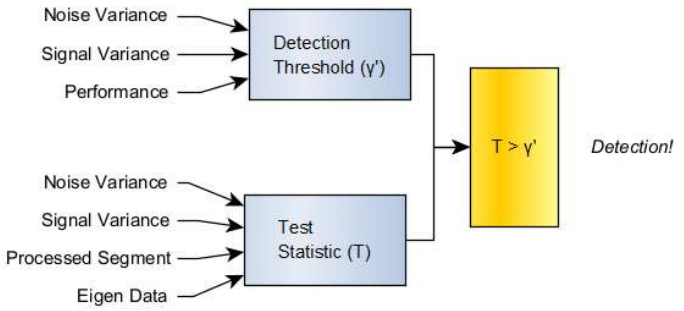

Figure 4.41 Flow diagram of the *detection* module for the estimator-correlator (WGN, CGN).

In the "Detection Threshold" module, the threshold $\gamma'(\mathbf{x})$ for each observation is obtained from the appropriate $P_{FA}(T)$ curve by simply looking for the value of T linked to the selected target probability of false alarm α. The appropriate $P_{FA}(T)$ curve is the one corresponding to the SNR that is closest to the SNR of the observation. Recall that in the pre-processing module the normalised performance curves $\check{p}_{0,1}(T)$ and $\check{P}_{FA,D}(T)$ were computed for a range of SNR and reference noise variance $\sigma^2_{w,ref} = 1$. For the energy detector and estimator-correlator in WGN, $P_{FA}(T)$ is obtained by scaling the corresponding $\check{P}_{FA}(T)$ curve by the noise variance σ^2_w. This is equivalent to scaling by σ^2_w the normalised threshold itself $\check{\gamma}'$, extracted from $\check{P}_{FA,D}(T)$.

In the "Test Statistic" module, the decision statistic for each observation $T(\mathbf{x})$ is calculated with equations (4.41), (4.56) and (4.76), for the energy detector and the estimator-correlators (WGN, CGN). Note that while the energy detector only needs the observation itself for calculating $T(\mathbf{x})$, the estimator-correlators also need the eigendata (i.e., normalised eigenvalues and eigenvectors), and the variance of the signal and noise associated with the observation.

4.14 Performance of the NP Detectors with Airgun Pulses

In the previous sections we have analysed the theory behind the energy detector and the estimator-correlators, explained their benefits and limitations, added improvements, and described their implementation. The work would not be complete without looking into the way these three detectors behave when applied to real data. In this section, we present the results from the three NP detectors run on a 250 s audio clip containing nine airgun pulses, as shown in Figure 4.42, Figure 4.43, and Figure 4.44. The clip is part of a set of recordings made by the author during a sound source verification (SSV) project that took place during the free-ice season, in a shallow, high-tidal bay in the south region of Alaska.

The purpose of the project was to characterise the acoustic output from various air gun array configurations in two distinct locations, previous to an Ocean Bottom Node (OBN) seismic survey that would take place in the area. The final goal was to establish a noise mitigation plan that could help protect the local marine mammals from the acoustic impact of these activities. In this project, six data acquisition buoys were allowed to drift downstream, recording audio and position while the airgun array was emitting from a fixed location. The buoys were equipped, from top to bottom, with a mast fitted with a radar reflector and a strobe light, a data acquisition box ('Peli-case'), the float, and a pair of 2-m long hydrophone cables protected by a metal cylinder solidly attached to the float. The hydrophones at the end of the cables were enclosed in a three-pole cage, only partially protected to minimise the interaction of the enclosure with the sound field.

The audio was recorded at a sampling rate 96 kHz, 16-bit resolution, and two channels. The airgun array used during the 250 s recording period had a volume of 1610 in^3 and emitted at an approximate rate of two pulses per minute (~28 s spacing). During this period, the buoy was passing through a shallow water section where debris from near the seabed consistently impacted against the hydrophones and the protective enclosure. The impacts produced short, low-energy spikes characterised by high frequency ringing from the metal enclosure and a low frequency emission from the vibration transmitted to the float. The regularity of these spurious events and their similitude to airgun pulses make the selected audio clip particularly useful for testing the NP algorithms, in particular their ability for detecting target events while rejecting the rest.

The three detection algorithms use a 0.5 s processing window and a target probability of false alarm $\alpha = 10^{-3}$. The data is resampled at 2 kHz and filtered with a IIR high-pass filter with a cutoff frequency of 20 Hz. The resampling and filtering are included to enhance the SNR by eliminating the contribution from the continuous background noise outside the dominant frequency region of the target signal (20 Hz to 1 kHz). For the calculation of the matrix of raw scores and covariance matrix of the *target signal*, 1274 training observations are used. The training data for the target signal consists of 0.5 s audio segments extracted manually from recordings made during the SSV survey. The recordings, acquired with six buoys on three different days, were made for three airgun arrays (440 in^3, 1610 in^3, 1770 in^3) deployed in shallow (50 m) and very shallow (5 m) waters. To ensure that the information built into the detector is not correlated with the data to be analysed, the events in the audio clip are not used as training data. For the calculation of the matrix of raw scores and covariance matrix of the *target noise*, 3022 training observations are used. The training observations for the target noise are 0.5 s segments automatically extracted from a 25-minute recording of continuous background noise, made before the start of the SSV survey.

The LOOC shrinkage estimator is used to compute the covariance matrices of the target signal and noise. The sample covariance matrix could have been used instead, as the number of variables of the processed audio segments ($N = 1000$) is 1.2 to 3 times smaller than the number of training observations. However, the LOOC method is selected as it showed to moderately improve the performance of the detector when $M \approx N$ (see Figure 4.33, bottom-left).

If we look at the results from Figure 4.42, Figure 4.43, and Figure 4.44, we can see some clear differences in performance between the three detectors. The images on the left show the 250 s audio clip, with the detections and noise highlighted as dark and light grey. The images on the right show a $\gamma'(SNR)$ graph depicting the performance of the detector.

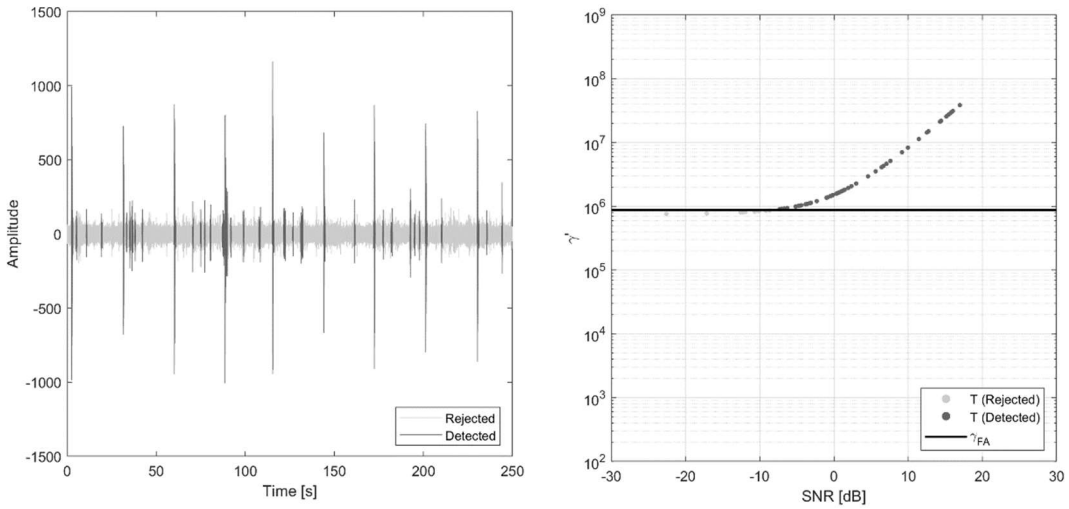

Figure 4.42 Section of audio recording ($f_s = 96\,\text{kHz}$) containing airgun pulses ($\text{T} \approx 28\,s$), continuous ambient noise, and transitory noise events (left image). The section is processed with an *energy detector* using resampling ($f_s = 2000\,\text{Hz}$) and high-pass filtering ($f_c = 20\,\text{Hz}$) for enhancing performance. The detected and rejected segments ($\tau = 0.5\,\text{s}$) are highlighted as dark and light grey, respectively. The corresponding test statistics T and the thresholds γ' for a target probability of false alarm $\alpha = 10^{-3}$ are shown on the right.

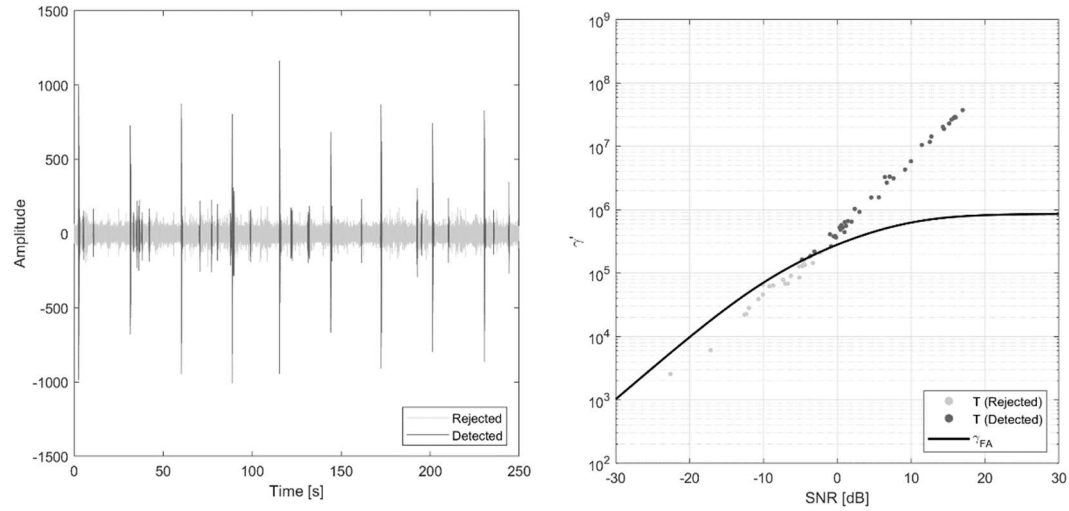

Figure 4.43 Section of audio recording ($f_s = 96\,\text{kHz}$) containing airgun pulses ($\text{T} \approx 28\,s$), continuous ambient noise, and transitory noise events (left image). The section is processed with an *estimator-correlator in WGN* using resampling ($f_s = 2000\,\text{Hz}$) and high-pass filtering ($f_c = 20\,\text{Hz}$) for enhancing performance. The detected and rejected segments ($\tau = 0.5\,\text{s}$) are highlighted as dark and light grey, respectively. The corresponding test statistics T and the thresholds γ' for a target probability of false alarm $\alpha = 10^{-3}$ are shown on the right.

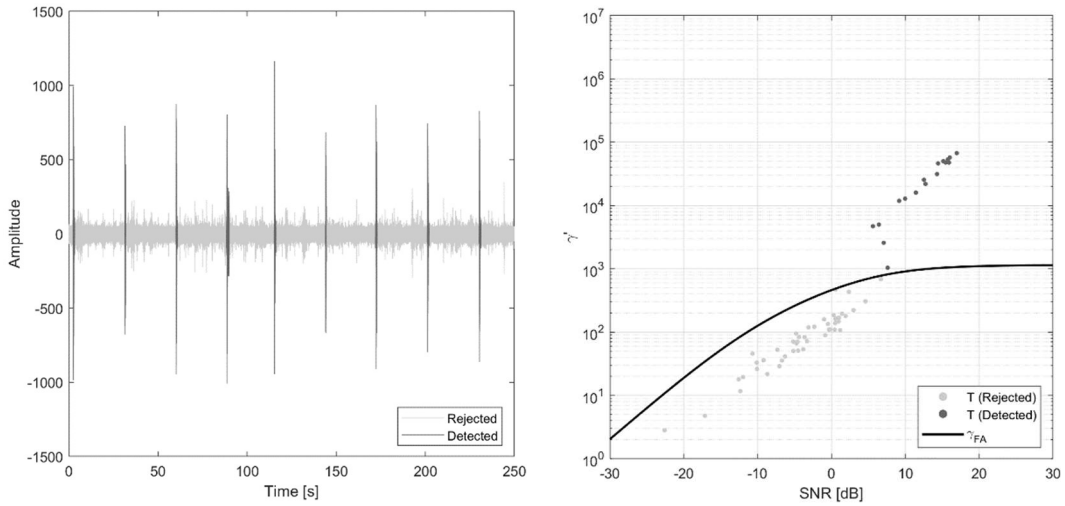

Figure 4.44 Section of audio recording ($f_s = 96\,\text{kHz}$) containing airgun pulses ($\text{T} \approx 28\,s$), continuous ambient noise, and transitory noise events (left image). The section is processed with an *estimator-correlator in CGN* using resampling ($f_s = 2000\,\text{Hz}$) and high-pass filtering ($f_c = 20\,\text{Hz}$) for enhancing performance. The detected and rejected segments ($\tau = 0.5\,\text{s}$) are highlighted as dark and light grey, respectively. The corresponding test statistics T and the thresholds γ' for a target probability of false alarm $\alpha = 10^{-3}$ are shown on the right.

The *energy detector* exhibits the worst performance of the three NP algorithms (see Figure 4.42). The energy detector is capable of rejecting the continuous noise, but many of the transient events caused by the debris are classified as detections. The dominant energy for the airgun pulses and for the spikes produced by the impact of the debris on the buoy is, in both cases, concentrated within the bandpass region of 20 Hz to 1 kHz. Since the detector is unable to distinguish between signals with different spectral responses, the non-target events with higher energy are inevitably mapped as detections.

The *estimator-correlator in WGN* shows a slight improvement in performance with respect to the energy detector (see Figure 4.43). Some of the spikes caused by the debris are ignored, but still several are interpreted as detections. The results are not surprising, since the spectral response of the background noise is far from flat ("white"); rather, it is characterised by a -6 dB/oct roll-off, with an overall level drop of ~30 dB between 20 Hz and 1 kHz. This highlights the importance that the statistical properties of the background noise have on the performance of the detector.

The *estimator-correlator in CGN* shows a noticeable improvement over the other two NP algorithms. In this particular example, the results are remarkable as all nine airgun pulses are detected (including first arrival and reflections) and all of the transient events caused by the debris are ignored. These results demonstrate the importance of using the estimator-correlator in CGN in those cases where the background noise is not "white" and there exist non-target events with frequency content similar to the target signal.

Chapter 5

Seismic Air Gun Model

The air gun array is the most powerful anthropogenic sound source used at sea. Being able to model its acoustic output is an essential part of the noise studies required in the context of Environmental Impact Assessments (EIA). These noise impact and mitigation studies are particularly important during seismic surveys, where larger array configurations are used continuously for periods of weeks. This chapter presents the background theory, features, implementation, and performance of a brand-new calibrated air gun model.

The chapter is organised in ten sections and comprises four parts. The *first part* reviews the theory of bubble dynamics and acoustic radiation for air gun array modelling. It starts with the introduction on Section 5.1, where the background, motivation and objectives are presented. The general physics and formulation of air bubble oscillations in water are discussed in Section 5.2. The small details are presented in Section 5.3, which describes the theoretical advancements that allow the application of the bubble dynamic's theory to the calculation of the acoustic signature of seismic air gun arrays. The initial conditions required for the calculation of the bubble motion parameters are described in Section 5.4. The first part ends with Section 5.5 and Section 5.6, which present the theory of acoustic radiation and interaction of air bubbles.

The *second part*, consisting of Section 5.7, reviews the theory of simulated annealing for the optimisation of complex multivariate problems. The theory is used for the development of a simulated annealing algorithm, later used for the optimisation of the calibration parameters of the air gun array model.

The *third part* presents a brand-new air gun array model developed for the thesis, called RAGAS[34] (Routines for Air Gun Array Simulation). This part consists of two sections. Section 5.8 introduces the air gun array modelling software and its capabilities. Section 5.9 analyses the effects that the various physics and control parameters in RAGAS have on the bubble motion and acoustic signature.

The *fourth part*, consisting of Section 5.10, deals with the calibration of RAGAS.

[34] https://github.com/Guille340/RAGAS_AIRGUN_MODEL

5.1 Introduction

The seismic air gun is the standard acoustic source used for oil and gas exploration at sea. The processing of its reflections provides a stratigraphic image of the seafloor down to depths of several hundreds of metres. Since the introduction of the air gun in the 1960s, many researchers have been involved in the development of models to predict its acoustic signature. These models became an essential part in the design of air gun arrays and helped improve the technology. Air gun arrays produce high sound pressure levels with potential negative effects on marine species, and these models have become an important tool for the assessment of the impact of noise in the marine environment.

In Section 2.7.1 we introduced the models that currently exist for simulating the acoustic signature of air gun arrays. Two companies directly related to the oil and gas industry have developed their own commercial air gun modelling software: Gundalf™ from Oakwood Computing Associates[35], and Nucleus+ from PGS[36] (Goertz et al., 2013). These models are industry standards and are fully calibrated. Unfortunately, their cost and design, tailored to the budgets and needs of geophysical operations, make this software a less attractive option for small companies, research institutions, and individuals. However, these are the most reliable options available for modelling air gun array signatures, and are often the only option accepted for EIA. Other companies and institutions such as JASCO and the Centre for Marine Science and Technology (CMST) have developed their own calibrated air gun array modelling software. In particular, the Airgun Array Source Model (AASM) created by MacGillivrai (2019; 2006) from JASCO, and the CMST Air Gun Array Model (CAGAM) created by Duncan & Gavrilov (2019). The two software packages proved to closely match simulations from Gundalf™ (Ainslie et al., 2016), but are not available for the research community. A fifth application that is open access is AGORA (Sertlek & Ainslie, 2015), but the software fails to provide results as accurate as the other four models. AGORA is the only package that is not explicitly calibrated and simply uses the calibration parameters from the AASM. This situation leaves the small company and researcher with the question of using an expensive, calibrated, but rather cumbersome industry standard, or an open-access but unoptimized model that is not always accepted in EIA projects.

This need for an open-access software that is also accurate, efficient, flexible, and usable motivated us to develop a new air gun array model. This new model is called RAGAS (Routines for Air Gun Array Simulation) and is the result of an in-depth study of the theory of bubble dynamics, air gun array modelling, and multivariate optimisation. It is the hope of the author that this software will serve to encourage new research that can help better understand the impacts that the noise produced by these seismic sources has on marine life.

[35] https://www.gundalf.com

[36] https://www.pgs.com/marine-acquisition/design-a-survey/nucleus

5.2 Air Gun Bubble Oscillation

The sound produced by an air gun is caused by the expansion and collapse of the air bubble released when the air gun is fired. Gilmore (1952) was the first researcher to study the dynamics of air bubbles in water and propose an equation for the bubble motion. Based on this work, Ziolkowsky (1970) published an integral physical model for the acoustic radiation of a single air gun. Gilmore's equation for the bubble motion is of the form

$$\left(1-\frac{u_b}{c_b}\right)r_b\frac{du_b}{dt}+\frac{3}{2}\left(1-\frac{u_b}{3c_b}\right)u_b^2=\left(1+\frac{u_b}{c_b}\right)h_b+\left(1-\frac{u_b}{c_b}\right)\frac{r_b}{c_b}\frac{dh_b}{dt} \tag{5.1}$$

where r_b is the bubble radius, u_b the bubble radial velocity ($u_b = dr_b/dt$), c_b the speed of sound at the bubble wall, and h_b the *specific enthalpy* at the bubble wall. The equation, based on the study of underwater explosions from Kirkwood and Bethe (1942), describes the radial motion of a spherical air bubble oscillating in an infinite, compressible liquid under an irrotational flow condition. The equation offers accurate predictions for bubble wall velocities as large as a significant fraction of the speed of sound (Gilmore, 1952).

In order to solve the equation of motion it is necessary to know h_b and c_b. These quantities can be expressed as a function of the time-dependent pressure $p(t)$ at the bubble wall (Gilmore, 1952)

$$h_b=\int_{p_\infty}^{p_b}\frac{dp}{\rho} \tag{5.2}$$

$$c_b^2=\frac{dp}{d\rho} \tag{5.3}$$

where ρ represents the density. For most liquids, the pressure-density relation for isentropic compression is given by the following experimental formula (Gilmore, 1952)

$$\left(\frac{p+B}{p_\infty+B}\right)=\left(\frac{\rho}{\rho_\infty}\right)^n \tag{5.4}$$

p_∞ is the hydrostatic pressure, ρ_∞ is the density away from the bubble, and B and n are constants ($n = 7.15$ and $B = 304.9$ MPa in water).

By inserting (5.4) into (5.2) and (5.3), h_b and c_b can be expressed as functions of the pressure at the bubble wall p_b.

$$h_b=\frac{p_\infty+B}{\rho_\infty}\cdot\frac{n}{n-1}\left[\left(\frac{p_b+B}{p_\infty+B}\right)^{\frac{n-1}{n}}-1\right] \tag{5.5}$$

$$c_b = c_\infty \left(\frac{p_b + B}{p_\infty + B}\right)^{\frac{n-1}{2n}} \tag{5.6}$$

c_∞ and p_∞ represent the sound speed and pressure that would exist in the water if the bubble was not present (*undisturbed* parameters), as opposed to c_b and p_b which represent the *disturbed* sound speed and pressure at the bubble wall. The sound speed away from the bubble is given by

$$c_\infty = \frac{n(p_\infty + B)}{\rho_\infty} \tag{5.7}$$

The numerator is the bulk modulus in water away from the bubble and can be obtained from (5.4) and the definition of bulk modulus $K = \rho\, dp/d\rho$.

If g is the standard gravity ($g \approx 9.8$ m $\cdot$ s^{-1}) and z_g is the depth of the air gun below the water surface, the hydrostatic pressure p_∞ is given by

$$p_\infty = p_{atm} + \rho_\infty g z_g \tag{5.8}$$

Other authors have published slightly different versions of the Gilmore's equation for the motion of an air bubble in water. An alternative version to (5.1) was first proposed by Keller and Kolodner (1956)

$$\left(1 - \frac{u_b}{c_\infty}\right) r_b \frac{du_b}{dt} + \frac{3}{2}\left(1 - \frac{u_b}{3c_\infty}\right) u_b^2 = \left(1 + \frac{u_b}{c_\infty}\right) \cdot \frac{p_b - p_\infty}{\rho_\infty} + \frac{r_b}{\rho_\infty c_\infty} \frac{dp_b}{dt} \tag{5.9}$$

A summary and derivation of (5.9) can be found in Keller and Miksis (1980) and Prosperetti (1984). The main differences between (5.1) and (5.9) are that, in the latter, the sound speed at the bubble wall is constant ($c_b = c_\infty$), and the sound speed is considered large enough that the second order term c_∞^{-2} can be discarded. From this last approximation, the specific enthalpy is reduced to $h_b = (p_b - p_\infty)/\rho_\infty$.

By discarding the first order terms of the sound speed c_∞^{-1} in (5.9), the Rayleigh-Plesset equation of motion for an air bubble in an incompressible fluid is obtained (Lezzi & Prosperetti, 1985)

$$r_b \frac{du_b}{dt} + \frac{3}{2} u_b^2 = \frac{p_b - p_\infty}{\rho_\infty} \tag{5.10}$$

By multiplying both sides of (5.10) by u_b/c_∞ and inserting the result into (5.9), one will arrive at an equation of the form (Herring, 1941)

$$\left(1 - \frac{2u_b}{c_\infty}\right) r_b \frac{du_b}{dt} + \frac{3}{2}\left(1 - \frac{4u_b}{3c_\infty}\right) u_b^2 = \frac{p_b - p_\infty}{\rho_\infty} + \frac{r_b}{\rho_\infty c_\infty} \frac{dp_b}{dt} \tag{5.11}$$

Among the four equations describing the dynamics of an air bubble in water, the one from Gilmore (1952) is the most general, as it includes sound speed terms of the second

order c^{-2}. The equations from Keller and Kollodner (1956) and Herring (1941) are accurate up to terms of the first order c^{-1}, with Mach numbers $u_b/c < 1$ for (5.9) and $u_b/c < 1/2$ for (5.11). The Rayleigh-Plesset equation is only valid for incompressible fluids, with a Mach number $u_b/c \approx 0$.

5.3 Pressure at the Bubble Wall

The equation of motion of an air bubble in water (see (5.1), (5.9), (5.10), and (5.11)) is solved iteratively to obtain the relative change in velocity u_b and pressure p_b at the bubble wall for each differential time increment. This recursive process starts with a set of initial conditions (see Section 5.4). The pressure p_b must be known in order to solve the equation of motion.

The pressure at the external side of the bubble wall p_b is given by (Prosperetti, 1984)

$$p_b = p_i - \frac{2\sigma_w}{r_b} - \frac{4\mu_w u_b}{r_b} \tag{5.12}$$

where p_i is the internal pressure of the bubble, increased with respect to p_b by the effect of the surface tension σ_w and dynamic viscosity μ_w of the liquid. The tension and viscosity in sea water are functions of temperature and salinity and can be calculated using the respective equations included in Nayar et al. (2014) and ITTC (2011). The pressure inside the bubble results from the combination of the partial pressures of the air released by the air gun p_a and the water vapour originated during the expansion and compression of the bubble p_v (Ziolkowsky, 1970)

$$p_i = p_a + p_v \tag{5.13}$$

The *polytropic relation* $pV^n = \mathrm{cst}$ characterises the relationship between pressure and volume during the compression and expansion of an ideal gas. Ziolkowsky (1970) proposed the use of this relation to express the pressure inside the bubble p_i as a function of the bubble radius r_b. Accordingly, the internal bubble pressure is given by

$$p_i = p_\infty \left(\frac{r_{b0}}{r_b}\right)^{3\gamma} \tag{5.14}$$

where r_{b0} is the equilibrium radius of the bubble and γ is the heat capacity ratio. The *equilibrium bubble radius* is the radius at which the pressure inside the bubble p_i equals the undisturbed hydrostatic pressure p_∞. The *heat capacity ratio* is the ratio of heat capacities of dry air at constant pressure C_p and volume C_V ($\gamma = C_p/C_V$). In real air guns, the experimental value of γ is 1.13, which lies somewhere in between the isothermal ($\gamma = 1$) and adiabatic ($\gamma = 1.4$) expansion (Ziolkowsky, 1970).

In the bubble, the vapour pressure p_v is maximum when the air pressure p_a is minimum, which occurs when the radius reaches its maximum ($\max r_b \leq 2r_{b0}$ under normal

working conditions). For typical air gun operating depths, $p_v < 0.1 p_a$ (Ziolkowsky, 1970). Hence, the partial pressure of water vapour can be neglected in practice ($p_i \approx p_a$).

The equilibrium radius r_{b0} can be obtained from the equation of state for an ideal gas (Prosperetti, 1984)

$$pV = mRT \tag{5.15}$$

where p, V, m, T and R are the pressure in Pa, volume in m^3, mass in kg, temperature in K, and constant of the gas in $J \cdot kg^{-1} \cdot K^{-1}$.

The gas constant R should not be confused with the universal gas constant R_u. The first one is specific of an individual gas or gas mixture, whereas the second one is independent of the gas and has a value of $R_u \approx 8.314$ $J \cdot mol^{-1} \cdot K^{-1}$. Both quantities are related by $R_u = RM$, where M is the molecular mass of the gas, in $kg \cdot mol^{-1}$.

Dalton's law of partial pressures states that the pressure of a gas mixture is equal to the sum of the partial pressures of each individual gas (Brown et al, 2017). A *partial pressure* is the hypothetical pressure that a gas would have if it alone occupied the volume of the mixture at the same temperature. By applying the concept of partial pressure to the equation of state of an ideal gas, we can easily arrive at the expression of the pressure of a gas mixture

$$p_m = \frac{m_m R_m T_m}{V_m} = \frac{T_m}{V_m} \sum_j m_j R_j \tag{5.16}$$

where the subscripts m and j refer to the mixture and the individual gases. From (5.16), the expressions for the constant R_m and molecular weight M_m of a gas mixture can be readily obtained

$$R_m = \frac{\sum_j R_j m_j}{m_m} \tag{5.17}$$

$$M_m = \frac{m_m}{\sum_j m_j / M_j} \tag{5.18}$$

Back to the equilibrium radius, both temperature and pressure inside the bubble equal the conditions outside when the equilibrium is reached ($p_i = p_\infty$, $T_b = T_w$). The volume of a spherical bubble V_b is given by

$$V_b = \frac{4}{3} \pi r_b^3 \tag{5.19}$$

The expression for the equilibrium radius is obtained by applying the conditions at the equilibrium state ($p_i = p_\infty$, $T_b = T_w$) and the volume formula (5.19) to the equation of estate (5.15) (Prosperetti, 1984)

$$r_{0b} = \left(\frac{3m_a R_a T_w}{4\pi p_\infty}\right)^{1/3} \tag{5.20}$$

The subscript a refers to the dry air inside the bubble and w to the surrounding water. The gas constant for air is $R_a = 287$ J · kg^{-1} · K^{-1}. The air mass is $m_a = V_g \rho_a$, where V_g is the volume of the air gun chamber and ρ_a the density of air ($\rho_a = 1.2$ kg · m^3 at 20 °C and 1 atm).

Ziolkowsky (1970) showed that the use of the polytropic relation from (5.14) results in an underdamped waveform when compared to measurements. He noted that the value of γ was related to the heat transfer between the air in the bubble and the water, and that the turbulence in the real bubble may be contributing to the variation in γ. To compensate for the limitation of using a constant value for γ, Ziolkowsky proposed to apply an exponential damping factor to the time response of the bubble radius $r_b(t)$ about the equilibrium radius r_{0b}. The damped bubble radius $r'_b(t)$ is then given by

$$r'_b(t) = (r_b(t) - r_{0b})e^{-\alpha t} + r_{0b} \tag{5.21}$$

where α is the damping constant. This artificial decay affects the pressure p_b and enthalpy h_b of the bubble and the velocity at the bubble wall u_b. The updated parameters for the damped bubble p'_b, h'_b, and u'_b can be calculated as follows

$$p'_b(t) \approx p_\infty \left(\frac{r_{b0}}{r'_b(t)}\right)^{3\gamma} \tag{5.22}$$

$$h'_b(t) = \frac{p_\infty + B}{\rho_\infty} \cdot \frac{n}{n-1}\left[\left(\frac{p'_b(t) + B}{p_\infty + B}\right)^{\frac{n-1}{n}} - 1\right] \tag{5.23}$$

$$u'_b(t) = \frac{dr'_b(t)}{dt} = [u_b(t) - \alpha(r_b(t) - r_{0b})]e^{-\alpha t} \tag{5.24}$$

The polytropic relation in (5.14) is a simple and convenient way of expressing the bubble pressure as a function of the radius r_b. However, this relation ignores physical processes such as *mass flow* and *heat transfer*. A more complete formulation for the bubble pressure p_b requires the definition of differential equations that synthetize the changes experienced by the bubble's mass m_b, temperature T_b, and volume V_b. According to (5.16), the internal pressure of the bubble can be better described by

$$p_i = \frac{m_b R_b T_b}{V_b} \tag{5.25}$$

During the oscillation process, water vapor will be produced and added to the air in the bubble. The mass and gas constant for the mixture can be expressed as functions of the same parameters for the dry air and water vapour

$$m_b = m_a + m_v \tag{5.26}$$

$$R_b = \frac{m_a R_a + m_v R_v}{m_b} \tag{5.27}$$

Subscripts a, v and b refer to the dry air, the water vapour, and the mixture of both gases in the bubble, respectively. In line with (5.13), the internal pressure in the bubble can be written as the sum of the partial pressures from the individual gasses by replacing (5.26) and (5.27) into (5.25)

$$p_i = p_a + p_v = \frac{m_a R_a T_b}{V_b} + \frac{m_v R_v T_b}{V_b} \tag{5.28}$$

The gas constant for water vapour is $R_v = 461$ J · kg^{-1} · K^{-1}.

In the following sub-sections, the differential equations that govern the changes in m_a, m_v, T_b, and V_b, and the associated physical processes are described. Two other processes that affect the dynamic viscosity of water μ_w and the hydrostatic pressure p_∞ are also presented.

5.3.1 Mass Transfer of Air due to Port Throttling

When an air gun is fired, the high pressurised air initially stored in the gun chamber is accelerated through its ports and released into the water, where an air bubble is formed. Laws et al. (1990) provide a semi-empirical equation for the flow rate of compressed air through an orifice

$$\frac{dm_a}{dt} = k_e A_g \sqrt{2\rho_g (p_g - p_i)} \tag{5.29}$$

where k_e is an empirically-determined constant that depends on the orifice design of the port ($k_e < 1$), A_g is the total port area, ρ_g and p_g are the density and pressure of air inside the gun chamber, and p_i is the pressure inside the bubble. ρ_g is given by m_g/V_g, where m_g and V_g represent the mass and volume of the air in the chamber. In general, the air-flow rate from the air gun dm_a/dt controls the amplitude of the initial bubble peak. The bubble oscillation period is dominated by the amplitude and shape of this peak.

Since the ports are in direct contact with the interior of the bubble, it can be assumed that the mass transfer is driven by the difference between the chamber pressure and the pressure inside the bubble ($p_b = p_i$). A more complete version of (5.29) is proposed by Graaf et al. (2012), but due to its simplicity the formulation from Laws et al. is presented here.

The constant k_e has a noticeable effect upon the amplitude of the primary bubble peak. MacGillivrai (2006), in his formulation of the throttling process, employs two empirical

parameters that replace the $k_e A_g \sqrt{2}$ factor in (5.29). These are the port throttling constant τ and the volume exponent β. The formulation proposed by MacGillivray is

$$\frac{dm_a}{dt} = \tau V_g^{\beta} \sqrt{\rho_g (p_g - p_i)} \tag{5.30}$$

The higher rate of mass injection dm_a/dt in larger guns explains MacGillivray's decision to introduce the empirical parameter β. In his tests, the addition of the β parameter substantially improved the agreement between model and observations.

From the equation of state for an ideal gas, the chamber pressure p_g can be written as

$$p_g = \frac{m_g R_a T_g}{V_g} \tag{5.31}$$

where m_g, T_g, and V_g are the mass, temperature, and volume of air in the gun chamber. The chamber volume V_g is a constant. Typical volumes for an air gun vary from 0.3 to 13 l (Landrø & Amundsen, 2010).

As the air is injected in the bubble, the mass of air in the gun chamber decreases at a rate dm_a/dt. The following condition is satisfied

$$m_g + m_a = m_{g0} \tag{5.32}$$

where m_{g0} is the total or initial mass of air in the gun chamber. m_{g0} depends on the *nominal* or initial pressure in the gun chamber p_{g0} and on the initial temperature of the enclosed air T_{g0}.

$$m_{g0} = \frac{p_{g0} V_g}{R_a T_{g0}} \tag{5.33}$$

The most common nominal pressure p_{g0} in air guns used in seismic exploration is 2000 psi (≈ 14 MPa), but values can be as low as 1500 psi (≈ 10 MPa).

Firing Efficiency

The firing mechanism of an air gun is not fully efficient, and a fraction of the air may remain in the chamber once the ports close after the air gun is fired. This reduced amount of mass transferred to the bubble results in a shorter bubble period. The air gun efficiency η is the ratio between the air mass released $m_a|_{t=\infty}$ and the initial air mass in the chamber m_{g0} ($\eta = m_a|_{t=\infty}/m_{g0}$), and is determined empirically (MacGillivray, 2006). The throttling process will stop ($dm_a/dt = 0$) when the mass of air in the gun reaches the value

$$m_g = m_{g0}(1 - \eta) \tag{5.34}$$

Effective Air Temperature in the Gun

The very high nominal pressure within the chamber of an air gun causes a departure from the ideal gas laws. This anomaly can be taken into account by replacing the initial temperature T_{g0} in (5.33) with an initial *effective* temperature T'_{g0} (Laws et al., 1990)

$$T'_{g0} = T_{g0}\left(1 + \frac{p_{g0}}{p_c}\right) \tag{5.35}$$

where p_c is a constant ($p_c = 139$ MPa). T'_{g0} is larger than T_{g0} by approximately 10%. The initial temperature has an important effect on the bubble oscillation period. A value near ambient ($T_{g0} = T_w$) is in good agreement with the experiments (Laws et al., 1990).

Relation between Temperatures in the Bubble and in the Gun Chamber

The temperatures inside the air gun chamber and the bubble are not the same. The temperature of the gas changes in the throttling process, as the gas passes through a section that reduces the flow rate, moving from a higher-pressure to a lower-pressure region. This phenomenon occurs exclusively in real gases, and is known as the *Joule-Thomson expansion*. The Joule-Thomson process is *adiabatic* (i.e., there is no heat transfer). The temperature change between the gas in the chamber and the bubble is given by (Laws et al, 1990)

$$\Delta T = k_{JT}(p_g - p_i) \tag{5.36}$$

where k_{JT} is the Joule-Thomson coefficient, in $\mathrm{K \cdot Pa^{-1}}$. Since the throttling process is considered adiabatic and there is no heat exchange with the surroundings, the formation of water vapour can be ignored. Therefore, the internal pressure of the bubble is due exclusively to dry air ($p_i = p_a$).

The Joule-Thomson coefficient describes the variation of temperature with pressure at constant enthalpy (i.e., $(\partial T/\partial p)_H$), and its value can be calculated as follows (Ansermet & Brechet, 2019)

$$k_{JT} = \frac{V_g}{m_g c_p}(\alpha_{te} T_g - 1) \tag{5.37}$$

where α_{te} is the coefficient of thermal expansion, defined as $V^{-1}(\partial V/\partial T)_p$. For temperatures between 0 and 40 °C, $3 \cdot 10^{-3} < \alpha_{te} < 4 \cdot 10^{-3}$.

The Joule-Thomson effect depends on the interaction of gas molecules. When a low-pressure gas expands, the inter-molecule distance increases and the attractive forces of the molecules withdraw energy from the system, reducing the temperature. Conversely, if a highly pressurised gas expands the repulsive forces of the molecules provide energy to the system as they separate, increasing the temperature. Hence, a positive Joule-Thomson coefficient indicates that the gas cools as it expands (notice the positive

temperature-pressure gradient of the curves of constant enthalpy at low pressures and temperatures in Figure 5.1), and a negative value results in the gas heating with expansion (see the negative gradient of the isenthalpic curves on the right side of the inversion curve in Figure 5.1).

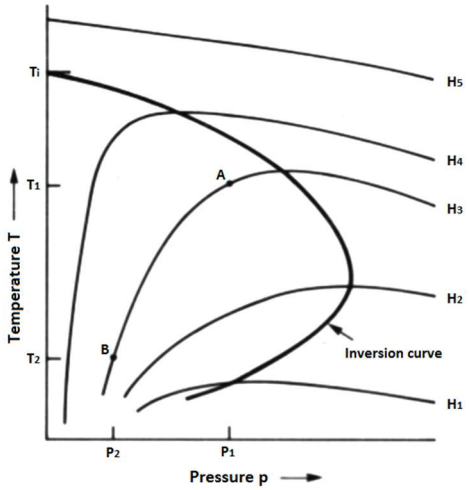

Figure 5.1 Curves of equal enthalpy (isenthalps) and inversion curves. The gradient of the isenthalps $(\partial T/\partial p)_H$ is by definition the Joule-Thomson coefficient k_{JT} (after Mandl, from https://theory.physics.manchester.ac.uk)

The variation in the bubble temperature is derived from the first law of thermodynamics (see Sub-Section 5.3.4). The temperature in the gun chamber T_g can be related to the bubble temperature T_b by the Joule-Thomson temperature difference ΔT

$$T_g = T_b + \Delta T \tag{5.38}$$

Extended Formulation for GI Guns

The formulas for the transfer rate of air mass during the throttling process are applicable to the standard air gun with a single chamber (G-gun). A more recent type of air gun, the GI-gun, consists of two air chambers: the *generator* and the *injector*. An air gun array is designed to reduce the bubble oscillations by carefully selecting the volume of its air guns, to ensure that bubbles of different resonant frequency interfere with each other and the primary peak dominates (Dragoset, 2000). The GI-gun technology was developed to reduce the amplitude of the bubble oscillations, producing a signature of similar quality to that of large and complex air gun arrays.

A GI-gun releases air from one chamber, the *generator*. As the bubble approaches its maximum radius, extra air is injected by a second chamber, the *injector*, hindering further compression or expansion of the bubble (see Figure 5.2, right).

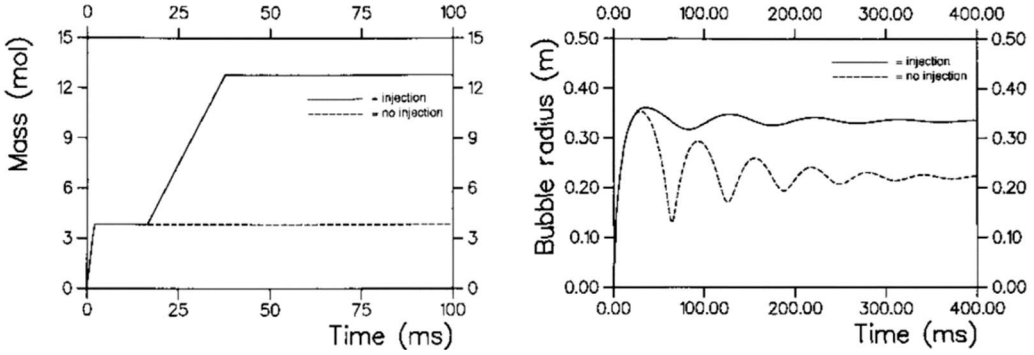

Figure 5.2 Mass of air inside the bubble from the two injectors of a 45 in^3 GI gun fired at 6 m depth (left) and its effect on the bubble radius (right) as a function of time (from Landrø, 1992).

The air-flow rate from the generator and injector chambers is given by (Landrø, 1992; Li et al., 2010)

$$\frac{dm_{aG}}{dt} = \tau_G V_{gG}^{\beta_G} \sqrt{\frac{m_{gG}}{V_{gG}}\left(p_{gG} - p_i\right)}, \quad \text{for } 0 \le t \le \Delta t_G \tag{5.39}$$

$$\frac{dm_{aI}}{dt} = \tau_I V_{gI}^{\beta_I} \sqrt{\frac{m_{gI}}{V_{gI}}\left(p_{gI} - p_i\right)}, \quad \text{for } t_I \le t \le t_I + \Delta t_I \tag{5.40}$$

where G and I refer to parameters from the generator and the injector chamber, respectively. The terms $\Delta t_{G,I}$ indicate the generator and injector periods, that is, the elapsed time between the firing of a chamber and the instant when the air inside has been effectively released. The initial mass in the injector chamber m_{g0G} is typically about twice the initial mass in the generator chamber m_{g0G}.

The optimal cancellation of bubble oscillations is achieved with the selection of appropriate values for the injection start time t_I and period Δt_I. The optimal values vary with injector volume and gun depth. The injection start time t_I should be decreased and the injection period Δt_I increased as the injector volume increases. Optimal injection start times and periods are typically in the range of 10-30 ms for typical single air gun volumes of 25 to 125 in^3. Optimal bubble suppression is achieved when the amount of air in the injector is twice that of the generator, condition referred to as *true GI mode*. However, good bubble-rejection can still be achieved in *harmonic mode*, when the amount of air in the generator and injector are the same (Landrø, 1992).

5.3.2 Mass Transfer of Water Vapour

At the bubble wall, the surrounding water experiences evaporation and condensation. The mass transfer rate of water vapour from heated surrounding water into the interior

of the bubble is known as *non-equilibrium mass transfer* rate. This process is characterised by the equation (Laws et al, 1990)

$$\frac{dm_v}{dt} = \frac{aA_b(p_{v,eq} - p_v)}{\sqrt{2\pi R_v T_s}} \tag{5.41}$$

where a is a constant (≈ 0.01), A_b is the surface area of the bubble in m^3 ($A_b = 4\pi r_b^2$), R_v is the gas constant for water vapour ($R_v = 461$ J $\cdot$ kg^{-1} $\cdot$ K^{-1}), T_s is the temperature at the bubble wall in K, $p_{v,eq}$ is the *equilibrium* or saturated vapour pressure in Pa, and p_v is the partial pressure of water vapour in Pa (p_v corresponds to the right term of the sum in (5.28)). Note that (5.41) accounts for both evaporation ($dm_v/dt > 0$) and condensation ($dm_v/dt < 0$). Maximum condensation rates are reached when the bubble radius is small and the pressure is high, whereas maximum evaporation rates are reached as the bubble approaches its maximum radius (Cook et al., 1997).

The mass transfer of water vapour has also an effect on the bubble temperature, and causes a continuous flow of heat between the bubble interior and the surrounding water (see Sub-Section 5.3.5).

The mass transfer rate law of (5.41) is just an approximation. Fujikawa and Akamatsu (1980) produced a detailed formulation for the mass and heat transfer caused by evaporation-condensation processes in the bubble wall. Their formula for the mass transfer rate of water vapour is

$$\frac{dm_v}{dt} = \frac{\alpha_M A_b}{\sqrt{2\pi R_v}}\left[\frac{p_{v,eq}}{\sqrt{T_{wi}}} - \Gamma\frac{p_v}{\sqrt{T_{bi}}}\right] \tag{5.42}$$

where α_M is the accommodation coefficient for evaporation condensation (≈ 0.04), Γ a correction factor, and T_{wi} and T_{bi} the temperatures of liquid and vaporised water at the phase interface. The *phase interface* is the thin non-equilibrium region at the bubble wall where the changes in the state of matter occur (i.e., liquid-gas transitions). The coefficient α_M is the ratio of the vapour molecules sticking to the phase interface to those impinging on it.

The saturated vapour pressure $p_{v,eq}$ is the pressure at which the number of water molecules entering the enclosed air space of the bubble, as a result of the evaporation process, equals the number of molecules returning to the surrounding liquid. The saturated vapour pressure is a function of the water temperature at the phase interface T_{wi} and is given by the Clausius-Clapeyron equation (Fujikawa and Akamatsu, 1980)

$$p_{v,eq} = p_c \exp\left[\frac{JL_v}{R_v T_c}\left(1 - \frac{T_c}{T_{wi}}\right)\right] \tag{5.43}$$

where p_c and T_c are the pressure and temperature for water-steam at the critical point ($p_c = 22.1$ MPa and $T_c = 647.3$ K). The parameter J is the mechanical equivalent of heat and represents the amount of work used in the production of heat (in SI notation

$J = 1$). The specific latent heat of vaporization L_v is the heat required to complete the phase change from liquid to vapour of a unit mass without change in temperature, and is expressed in $J \cdot kg^{-1}$.

Cook et al. (1997) further simplify (5.43) by assuming $T_c \gg T_{wi}$. This approximation results in the following expression

$$p_{v,eq} = p_c \exp\left(\frac{-L_v}{R_v T_{wi}}\right) \tag{5.44}$$

On each side of the interface between the bubble and the water there exists one thin thermal boundary layer with a thickness of $\approx 100\ \mu m$ (see Sub-Section 5.3.5), where the evaporation-condensation and heat conduction processes occur (see Figure 5.3, left).

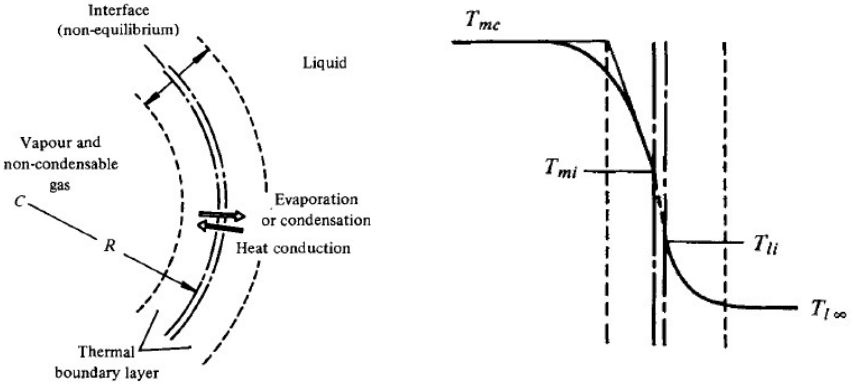

Figure 5.3 Model of the non-equilibrium region in the bubble-water interface (left) and temperature profile from inside the bubble to the surrounding water through the thermal boundary layer (right) (from Fujikawa and Akamatsu, 1980)

The properties inside the bubble and in the surrounding water can be considered spatially uniform, but not within the narrow double-layer separating the two environments. The boundary layer acts as a transition region with a temperature gradient. The temperatures T_{bi} and T_{wi} are defined within the boundary layer and their values lie somewhere in between the temperature inside the bubble T_b and the temperature of the surrounding water T_w (see Figure 5.3, right). The formulation given by Fujikawa and Akamatsu (1980) to calculate T_{bi} and T_{wi} is rather complex, and other authors opt for simpler solutions. Laws et al. (1990), in their approximate equation for the vapour flow rate (5.41), use the term T_s which represents the average of T_b and T_w. Cook et al. (1997) and Graaf et al. (2012) assume that $T_{bi} = T_b$ and $T_{wi} = T_w$. Either of these two alternatives for defining the temperatures at the phase change interface is preferred to the complex original formulation from Fujikawa and Akamatsu.

The correction factor Γ compensates for the flow of vapour mass towards or away from the phase interface, and is expressed as follows:

$$\Gamma = e^{-\Omega^2} - \Omega\sqrt{\pi}\left(1 - \frac{2}{\sqrt{\pi}}\int^{\Omega} e^{-x^2} dx\right) \tag{5.45}$$

$$\Omega = \frac{1}{p_v}\frac{dm_v}{dt}\left(\frac{R_v T_b}{2}\right)^{1/2} \tag{5.46}$$

The equation for the correction factor is relatively complicated. Cook et al. (1997) and Graaf et al. (2012) use a value of $\Gamma = 1$ in their studies. This assumption is valid under conditions of equilibrium or when the deviation from equilibrium is small.

The results of the model from Graaf et al. (2012) show a small impact of the mass transfer rate of water vapour on the radius, temperature, and oscillation period of the bubble when compared to other damping mechanisms.

5.3.3 Variation of Bubble Volume and Effect of Air Gun Body

The variation of the bubble volume with time is a function of the bubble radius r_b and the radial velocity u_b. The expression is obtained as the time differential of (5.19)

$$\frac{dV_b}{dt} = 4\pi r_b^2 u_b \tag{5.47}$$

The presence of the air gun may also affect the bubble dynamics, as suggested by Schulze-Gattermann (1972). The air gun is considered a solid sphere of radius r_{gun} and volume V_g situated at the centre of the oscillating bubble. Note that V_{gun} does not refer to the volume of air in the gun chamber, but rather to the space occupied by the gun inside the bubble. The new volume of gas in the bubble after accounting for the air gun presence is given by

$$V_b = \frac{4\pi r_b^3}{3} - V_{gun} \tag{5.48}$$

where r_b is the *extended* bubble radius, computed with

$$r_b = \left[\frac{3}{4\pi}\left(V_b + V_{gun}\right)\right]^{1/3} \tag{5.49}$$

The volume of gas in the bubble V_b for a given radius r_b is now smaller than before, when the presence of the air gun was ignored. As a result, the bubble pressure is effectively increased. The smaller pressure difference between the bubble and the surrounding water during expansion reduces the intensity and velocity during the compression phase, increasing the minimum radius at every collapse of the bubble and reducing the oscillation period (Graaf et al, 2012).

The concentrical geometry assumed by Schulze-Gattermann is justified by the slow ascension of the bubble (see Sub-Section 5.3.8), which requires a few oscillation periods

to travel a distance comparable to the dimensions of the cross section of the air gun. After those first few oscillation periods the bubble detaches from the air gun, making the use of (5.49) questionable. Considering the presence of the airgun in the calculation of the bubble radius is certainly beneficial, as it offers a more realistic estimate of the bubble radius when the system produces the largest acoustic output.

5.3.4 Variation of Bubble Temperature

The differential of temperature in the bubble is controlled by various mechanisms, and it can be derived from the first law of thermodynamics for an open, quasi-static thermodynamic system (Landrø, 1992). This law applies the principle of conservation of energy to a thermodynamic process, and states that the change in internal energy U equals the balance between the heat transferred into the system Q and the work done by it W. Expressed in terms of differentials, the first law of thermodynamics for an *open*, *diathermal* and *deformable* system is given by (Ansermet & Brechet, 2019)

$$dU = \delta Q + \delta W + \delta C \tag{5.50}$$

where dU is the differential of the internal energy, and δQ, δW, δC are the infinitesimal increments of heat transferred to the system, work performed on the system, and matter transferred to the system through convection. These are all energy quantities, measured in joules (J). It is assumed that the system is not exposed to forces and torques, which would modify the translational and rotational kinetic energies of and around the mass.

For a pressure p and volume V, the infinitesimal mechanical work performed on a gas is expressed as $\delta W = -pdV$. The matter transfer can result in mechanical work and heat transfer, and δC will be included as part of δW and δQ. For a gas, (5.50) can be more generally written as (Ansermet & Brechet, 2019)

$$dU = \delta Q - pdV \tag{5.51}$$

The added heat can be expressed as a function of the enthalpy H as $\delta Q = dH - Vdp$. From the definition of specific heat capacities at constant pressure c_p and volume c_v, the enthalpy H and internal energy U are given by

$$H = mc_pT \tag{5.52}$$

$$U = mc_vT \tag{5.53}$$

where m is the mass in kg and T is the temperature in K. The specific heat capacities c_p and c_v are properties of the material and are expressed in $\mathrm{JK^{-1}kg^{-1}}$. These should not be confused with the specific heats $C_{p,V}$, which are properties of the sample and are related to the specific heat capacities by $C_{p,V} = c_{p,V}m$.

By inserting (5.52) and (5.53) in (5.51), and applying the heat-enthalpy relation, one will arrive at the modified form of the first law of thermodynamics for an open, diathermic, deformable system with no external mechanical forces or torques (Landrø, 1992)

$$dU + pdV = hdm + \delta Q' \tag{5.54}$$

where dU is the differential internal energy, pdV is the differential work done by the bubble, h is the specific enthalpy in $\mathrm{J \cdot kg^{-1}}$, hdm is the added heat due to matter exchange, and $\delta Q'$ is the added heat due to conduction. The expressions for these terms, as obtained from the full equation, are

$$dU = c_v m dT + c_v T dm \tag{5.55}$$

$$h = c_p T \tag{5.56}$$

$$\delta Q' = c_p m dT - V dp \tag{5.57}$$

Note that hd_m, which accounts for the transfer of mass, originates from δQ. Hence, a different term $\delta Q'$ is used to represent the heat transfer due to conduction exclusively.

The differential variation of temperature inside the bubble with time for the open, quasi-static system can be obtained from (5.54)–(5.56) by isolating the temperature differential dT, dividing by the time differential dt at both sides of the equation, and particularising the mass, volume, temperature, pressure, and heat capacities for the bubble

$$\frac{dT_b}{dt} = \frac{R_b T_b \frac{dm_b}{dt} + \frac{dQ'}{dt} - p_b \frac{dV_b}{dt}}{m_b c_{V,b}} \tag{5.58}$$

R_b is the specific gas constant of the air in the bubble given by the difference of specific heat capacities ($R_b = c_{p,b} - c_{V,b}$). According to (5.58) the temperature variation inside the bubble is controlled by mass injection, volume expansion or contraction, and heat transfer.

The mass in the bubble m_b is equal to the combination of mass from dry air m_a and mass from water vapour m_v, as shown in (5.26).

The specific heat capacity of the ideal gas mixture at constant pressure or volume $c_{pV,m}$ is given by the sum of the specific heat capacities $c_{pV,i}$ of the individual gases weighted by their mass fraction m_i/m_m (Tsilingiris, 2008)

$$c_{pV,m} = \frac{\sum_i \ c_{pV,i} m_i}{m_m} \tag{5.59}$$

Applied to the gas mixture in the bubble, (5.59) becomes

$$c_{pV,b} = \frac{c_{pV,a} m_a + c_{pV,v} m_v}{m_b} \tag{5.60}$$

The specific heat capacities for dry air and water vapour are $c_{(p,V),a} = (1005, 718)$ $J \cdot kg^{-1} \cdot K^{-1}$ at 20 °C and $c_{(p,V),v} = (1970, 1500)$ $J \cdot kg^{-1} \cdot K^{-1}$ at 100 °C (Engineering Toolbox, 2001).

The heat transfer term dQ'/dt is neglected by Landrø (1992). Instead, he introduces an effective specific heat $c_{V,ef} \approx 5c_V$ to compensate for the fact that no heat or mass transfer between the bubble and the water was considered. Following the approach of Laws et al. (1990) and MacGillivray (2006), the heat transfer term is retained here.

The term dQ'/dt can be used to model two important processes occurring as the bubbles expands and collapses: (1) the heat transfer between bubble and water Q_w, and (2) the heat transfer due to phase change Q_v. The heat transfer ratio dQ'/dt is then given by

$$\frac{dQ'}{dt} = -\frac{dQ'_w}{dt} - \frac{dQ'_v}{dt} \tag{5.61}$$

The negative sign convention indicates that the heat energies Q_w and Q_v flow outside the system, from the bubble into the water.

By inserting (5.26), (5.60) and (5.61) into (5.58), we arrive at the following expression for the rate of temperature variation in the bubble

$$\frac{dT_b}{dt} = \frac{R_b T_b \frac{dm_a}{dt} + R_b T_b \frac{dm_v}{dt} - \frac{dQ'_w}{dt} - \frac{dQ'_v}{dt} - p\frac{dV_b}{dt}}{m_a c_{V,a} + m_v c_{V,v}} \tag{5.62}$$

From (5.62), one can see that the variation in temperature depends on five differential functions: the mass transfer of air due to port throttling dm_a/dt (Sub-Section 5.3.1), the mass transfer of water vapour from evaporation-condensation dm_v/dt (Sub-Section 5.3.2), the bubble expansion-compression rate dV_b/dt (Sub-Section 5.3.3), the conductive heat transfer between bubble and water dQ'_w/dt (Sub-Section 5.3.5), and the heat transfer from evaporation-condensation dQ'_v/dt (Sub-Section 5.3.6).

5.3.5 Heat Transfer between Bubble and Water

The heat conduction between the bubble interior and the surrounding water takes place in the thermal boundary layer (see Figure 5.3). The rate of heat conduction depends on the thickness δ and the thermal conductivity κ of the boundary layer, the surface area of the bubble A_b, and the difference in temperature across the bubble wall ΔT, and is given by the classical heat transfer formula for *conduction* (Laws et al., 1990)

$$\frac{dQ'_w}{dt} = \Delta T A_b \frac{\kappa}{\delta} \tag{5.63}$$

The heat flows towards the coolest reservoir, the surrounding water, in accordance with the sign convention from (5.61). Hence, $\Delta T = T_{hot} - T_{cold}$ is in fact $\Delta T = T_b - T_w$. There

will be periods when the temperature of the gas will be lower than that of the surrounding water, and the heat will flow towards the bubble ($dQ_w/dt < 0$).

The expansion and contraction of the bubble is relatively slow compared to the speed of sound (Mach number $u_b/c_b \ll 1$) but fast enough to generate a constant turbulence near its surface (Laws et al., 1990). The eddies formed contribute to increase the heat transfer, but also the effective viscosity of the surrounding water (see Sub-Section 5.3.7). The turbulence introduces additional damping to the oscillations.

Due to the turbulent character of the motion, the surface of the bubble is not perfectly smooth. This surface roughness causes an effective increase of the interface area. In their model, Laws et al. (1990) employ an effective area of the form

$$A_{b,eff} = k_{rou} 4\pi r_b^2 \tag{5.64}$$

where k_{rou} is an empirical constant determined to account for the increased interface area ($k_{rou} \approx 10$). The surface area A_b in (5.42) and (5.63) can be replaced with $A_{b,eff}$ to include the effect of the rough interface.

The heat transport occurs across the thermal boundary layer. The thickness of this layer is given by Thomas (1979)

$$\delta = 8 r_b R_e^{-3/4} P_r^{-1/2} \tag{5.65}$$

where R_e and P_r are the Reynolds and Prandtl numbers for the water surrounding the bubble wall.

The Reynolds number is a dimensionless quantity used to predict the flow pattern of a fluid. More specifically, $R_e < 2300$ indicates that the flow is *laminar*, and $R_e > 4000$ corresponds to a *turbulent* flow. In between these two values is the *transient* flow region. The Reynolds number represents the ratio of inertial to viscous forces. Particularised to the case of the air bubble in water, the Reynolds number is a function of the radius r_b and velocity u_b of the bubble, and the density ρ_w and dynamic viscosity μ_w of the water

$$R_e = \frac{2 r_b u_b \rho_w}{\mu_w} \tag{5.66}$$

The Prandtl number is a dimensionless number defined as the ratio of viscous to thermal diffusion rate. If the specific heat capacity at constant pressure is $c_{p,w}$, the Prandtl number is of the form

$$P_r = \frac{c_{p,w} \mu_w}{\kappa} \tag{5.67}$$

We could not find a clear mention in the literature about the thermal conductivity κ that should be used in the context of heat transfer between the air bubble and water. Herring (1950) stated that the outward heat transfer is a function of the thermal

conductivity of the gas. Liu et al. (2011) obtained a value for the heat transfer coefficient κ/δ of 4000 $W \cdot m^{-2} \cdot K^{-1}$, validated experimentally. Graaf et al. (2012) modelled two extreme cases corresponding to the thermal conductivity of water κ_w (0.6 $W \cdot m^{-1} \cdot K^{-1}$ at 20°C) and dry air κ_a (0.025 $W \cdot m^{-1} \cdot K^{-1}$ at 20°C).

Based on the assumption that the heat transfer between bubble and water is controlled by the gas in the bubble (Herring, 1950), the expression from Tsilingiris (2008) for the thermal conductivity of a gas mixture can be used to compute $\kappa = \kappa_b$.

A more practical solution was proposed by MacGillivray (2006), who integrated the parameters κ, δ and k_{rou} from (5.63) into a single empirical constant, here called ε and expressed in $W \cdot m^{-2} \cdot K^{-1}$. The expression for the conductive heat transfer between bubble and water becomes

$$\frac{dQ'_w}{dt} = \varepsilon \Delta T A_b \tag{5.68}$$

The best-fit value in MacGillivray's study for ε is approximately 22000 $W \cdot m^{-2} \cdot K^{-1}$, which corresponds to a heat transfer coefficient $\kappa/\delta = 2200$ $W \cdot m^{-2} \cdot K^{-1}$ for $k_{rou} = 10$. This value of κ/δ is of the same order as the optimal heat transfer coefficient found by Liu et al. (2011).

The heat transfer between bubble and water results in *heat loss* ($dQ'_w/dt > 0$) for high bubble temperatures (small radius, compression phase) and *heat gain* ($dQ'_w/dt < 0$) for low bubble temperatures (large radius, expansion phase). As the bubble expands, its temperature falls below the temperature of the water, causing an inward heat flow that increases the bubble temperature. Conversely, during compression the bubble temperature becomes larger than the temperature of the water, causing heat loss and a temperature decrease in the bubble. The contribution of heat conduction to the increase in temperature during expansion and the temperature reduction during compression allows the bubble to grow and shrink further, reducing the damping (see Sub-Section 5.9.3).

5.3.6 Heat Transfer from Condensation-Vaporisation

The mass transfer of water vapour into the bubble is also a mechanism of heat loss. A certain amount of energy, called *latent heat*, is required to transform the water in contact with the bubble surface into vapour (see Sub-Section 5.3.2). The energy used for the phase change does not remain at the bubble when the vapour condenses, since the heat will flow to the coolest reservoir, the surrounding water. Hence, the latent heat is lost from the system from the transformation of water into gas during the evaporation process. The heat flow due to vapour mass transfer is given by (Laws et al, 1990; Cook et al, 1997)

$$\frac{dQ'_v}{dt} = \begin{cases} L_v \dfrac{dm_v}{dt} & \text{for } dm_v/dt \geq 0 \\ 0 & \text{for } dm_v/dt < 0 \end{cases} \tag{5.69}$$

where L_v is the specific latent heat in $\mathrm{J} \cdot \mathrm{kg}^{-1}$ and dm_v/dt is the mass flow of water vapour. The identity $dm_v/dt \geq 0$ refers to the evaporation period, and $dm_v/dt < 0$ to the condensation period.

The heat loss due to water vaporisation reduces the maximum and minimum bubble radius and its oscillation period. Its contribution is small compared to that of the conventional heat transfer between bubble and water (see Sub-Section 5.9.4). Graaf et al. (2012) arrive at the same conclusion.

5.3.7 Turbulence and Effective Viscosity

The flow of the bubble produced by an air gun is expected to be not completely irrotational, and in this situation the viscous effects become important (Landrø, 1992). In the water, the bubble moves at a speed large enough for a turbulent flow to be present at the bubble wall. The turbulence enhances the heat transfer processes between bubble and water by increasing the effective area of the bubble (see Sub-Section 5.3.5). But it also causes an increase in the viscosity of the surrounding water which favours the damping of the bubble oscillations. The standard way of accounting for the additional energy loss caused by the turbulent flow is to use an effective viscosity for the water. The *effective dynamic viscosity* $\mu_{w,e}$ is a function of the standard dynamic viscosity of water μ_w and the Reynolds number and is given by (Laws et al. 1990)

$$\mu_{w,e} = \mu_w(1 + 0.02R_e) \tag{5.70}$$

The Reynolds number from (5.66) is calculated using the standard dynamic viscosity of the water μ_w and inserted in (5.70) to compute the effective dynamic viscosity $\mu_{w,e}$. The latter can be used in (5.12) and (5.67) to account for the effect of turbulence on the bubble pressure p_b and Prandtl number P_r. An empirical parameter k_μ can be introduced to replace the 0.02 factor that accompanies the Reynolds number to gain an extra degree of control over the damping of the bubble motion during calibration.

5.3.8 Rise of the Bubble in Water

The lower density of air makes the bubble rise through the water as it expands and compresses. If z_b is the depth of the bubble, g the acceleration of gravity in $\mathrm{m} \cdot \mathrm{s}^{-2}$ and r_b the bubble radius, the vertical velocity of the bubble is given by (MacGillivray, 2006)

$$\frac{dz_b}{dt} = \frac{2g}{r_b^3} \int_0^t r_b^3 dt \tag{5.71}$$

The rise of the bubble introduces time dependence in the hydrostatic pressure p_∞. The bubble depth z_b replaces the depth of the air gun in (5.8). Another factor that would add time dependence to p_∞ is the sound pressure generated at the bubble wall by the other bubbles in an air gun array (see Section 5.6).

5.4 Initial Conditions

The time dependent radial velocity and pressure at the bubble wall $u_b(t)$ and $p_b(t)$ are determined by resolving iteratively the Gilmore's equation of the bubble motion (5.1) or one of its approximate alternatives (5.9), (5.10) and (5.11). In each iteration, the time differential is determined for u_b, p_b and other quantities such as bubble mass m_b, temperature T_b, volume V_b, radius r_b and depth z_b. These differentials are multiplied by the selected time resolution Δt and then added to the last accumulated value to obtain the value for the current iteration. For example, if $(n-1)$ and (n) indicate the previous and current iterations, the bubble mass is $m_{b,(n)} = m_{b,(n-1)} + \Delta t \cdot dm_{b,(n)}/dt$. The initial conditions determine the $(n-1)$ value in the first iteration for the various parameters of interest. The preferred initial conditions for the air gun model are

$$V_{b0} = V_g \tag{5.72}$$

$$r_{b0} = \left[\frac{3\left(V_{b0} + V_g\right)}{4\pi}\right]^{1/3} \tag{5.73}$$

$$z_{b0} = z_g \tag{5.74}$$

$$p_{\infty 0} = p_{atm} + \rho_\infty g z_{b0} \tag{5.75}$$

$$p_{b0} = p_{\infty 0} \tag{5.76}$$

$$c_{b0} = c_\infty \tag{5.77}$$

$$u_{b0} = \frac{p_{\infty 0} - p_{b0}}{\rho_\infty c_\infty} = 0 \tag{5.78}$$

$$T_{g0} = T_w\left(1 + \frac{p_{g0}}{p_c}\right) \tag{5.79}$$

$$T_{b0} = T_{g0} \tag{5.80}$$

$$m_{v0} = 0 \tag{5.81}$$

$$m_{b0} = m_{a0} = \frac{p_{b0} V_{b0}}{R_a T_{b0}} \tag{5.82}$$

where V_{b0}, r_{b0}, z_{b0}, p_{b0}, u_{b0} and m_{b0} are the initial volume, radius, depth, pressure, radial velocity and mass of the gas bubble, $p_{\infty 0}$ and c_{b0} the initial pressure and sound

speed of water at the bubble wall, m_{v0} and m_{a0} the initial mass of water vapour and dry air in the bubble, and T_{g0} and T_{b0} the initial temperature inside the gun chamber and bubble.

The initial conditions are the same as those proposed by MacGillivray (2006), except for the bubble radius, which here is twice as large due to the presence of the air gun body (see Sub-Section 5.3.3).

5.5 Radiated Acoustic Field

The instantaneous acoustic pressure $p(t)$ and particle velocity $u(t)$ can be calculated at any location in the water using the equations proposed by Gilmore (1952) and summarised by Ziolkowsky (1970). These equations are functions of the pressure p_b and radial velocity u_b in the bubble wall. Gilmore provides two methods for the calculation of $p(t)$ and $u(t)$: (1) a *first-order* approximation, only accurate to Match terms of the first order (u_b/c_∞), and (2) a *second-order* approximation that uses the Kirkwood-Bethe hypothesis to provide a more accurate solution. Below are listed the sets of equations for the first and second order approximations. For details regarding the derivation see Gilmore (1952), Ziolkowsky (1970) or Keller and Kollodner (1956).

The *first-order* approximation for the calculation of the instantaneous acoustic pressure $p(t)$ and particle velocity $u(t)$ at a distance r from the centre of the bubble is described by the following set of equations

$$p(r,t) = \rho_\infty \left[\frac{\partial \Phi}{\partial t} - \frac{1}{2}\left(\frac{\partial \Phi}{\partial r}\right)^2\right] \tag{5.83}$$

$$u(r,t) = -\frac{\partial \Phi}{\partial r} \tag{5.84}$$

$$\frac{\partial \Phi}{\partial t} = \frac{f'(\xi)}{r} \tag{5.85}$$

$$\frac{\partial \Phi}{\partial r} = -\frac{f(\xi)}{r^2} - \frac{f'(\xi)}{r c_\infty} \tag{5.86}$$

$$f(\xi) = -\frac{r_b^2}{c_\infty}\left[h_b + \frac{u_b^2}{2} - u_b c_\infty\right] \tag{5.87}$$

$$f'(\xi) = r_b\left[h_b + \frac{u_b^2}{2}\right] \tag{5.88}$$

where Φ is the velocity potential and ξ the time function ($\xi = t - r/c_\infty$). The negative term in (5.83) is called the *afterflow* pressure and is only important at small distances from the bubble (Keller and Kollodner, 1956). Since the enthalpy is a state function,

fully defined by its value at the initial and final states, the specific enthalpy of the bubble is given by $h_b = (p_b - p_\infty)/\rho_\infty$.

The *second-order* approximation is given by the following set of equations

$$p(r,t) = \rho_\infty \left(\frac{y}{r} - \frac{u^2}{2}\right) + \frac{\rho_\infty}{2c_\infty^2}\left(\frac{y}{r} - \frac{u^2}{2}\right)^2 \tag{5.89}$$

$$u(r,t) = \frac{y}{c_\infty r} + \frac{K_3 y^2}{c_\infty^3 r^2}\left(1 - \frac{y}{c_\infty^2 r} + \frac{K_3^2 y^4}{2c_\infty^8 r^4}\right) \tag{5.90}$$

$$y = r_b\left(h_b + \frac{u_b^2}{2}\right) \tag{5.91}$$

$$K_3 = \frac{c_\infty^3 r_b^2 u_b}{y^2}\left(1 - \frac{u_b^2}{2c_\infty^2}\right) - \frac{c_\infty^2 r_b}{y}\left(1 - \frac{u_b}{c_\infty}\right) \tag{5.92}$$

The parameters u_b, r_b, h_b, p_b, on which the two solutions depend, are functions of time. By evaluating $f(\xi)$ and $f'(\xi)$ at each time step for the first-order solution, or y and K_3 for the second-order solution, the time-dependent pressure $p(t)$ and radial velocity $u(t)$ can be obtained for any distance r from the centre of the bubble.

The radius and velocity of the bubble produced by an airgun are large enough for them to have an effect on the arrival time of the pressure wave. If t_b is the time axis selected for the computation of the bubble parameters, the time axis for the resulting values of $p(t)$ and $u(t)$ is given by (Gilmore, 1952)

$$t = t_b + \left(\frac{r - r_b}{c_\infty}\right) \qquad \text{for first order approximation} \tag{5.93}$$

$$t = t_b + \left(\frac{r - r_b}{c_\infty}\right)\left(1 - \frac{u_b r_b}{r c_\infty}\right) \quad \text{for second order approximation} \tag{5.94}$$

The first order approximation in (5.83)-(5.88) is only accurate when the liquid velocity is considerably lower than the speed of sound. For the large wall velocities associated with an air gun bubble this approximation is only applicable at locations far away from the bubble wall, that is, at points outside the *high velocity region* (Gilmore, 1952). The first order approximation is certainly worth considering for far field simulations, as this option is less computationally demanding than the second order approximation. However, the improvement in speed attained by using (5.83)-(5.88), as opposed to (5.89)-(5.92), may be limited if compared to the time spent in the computation of the bubble dynamics. This is particularly true if complex physical phenomena have been modelled to obtain a more realistic representation of the bubble motion.

5.6 Bubble Interactions

The theory presented so far described how to model the motion of the air bubble generated by an air gun and its radiated sound field. However, these sources are rarely often used individually. Instead, a number of air guns of different volumes are combined in an array to produce a more desirable acoustic signature. Arrays are designed to generate a far field acoustic signature in the vertical direction that is short and impulsive, with high amplitude, broad bandwidth and a smooth low-frequency response (Giles & Johnston, 1973; Dragoset, 2000).

Measuring the signature of an array directly is technically challenging due to the large water depths required to capture the waveform in the far field of the source. The output of an acoustic source in the far field is of particular interest, as within that region the wave has developed a spherical wavefront and its phase spectrum, directivity and attenuation with distance are fully characterised. The *far field distance* can be approximated to the ratio between the square of the source's largest dimension and the wavelength (L^2/λ), and therefore is considerably smaller for a single air gun than it is for the array. Ziolkowsky et al. (1982) proposed a method to compute the far field signature of an air gun array using near field measurements. The method showed that the acoustic signature of an array could be obtained from far field measurements of the notional signature of each individual source. If the separation between the air guns in the array were sufficiently large, the notional signature from the individual sources could be combined using equations in Section 5.5 to obtain the pressure and particle velocity at any point in the far field. In practice, the sources are so close to each other that their radiated pressure can affect the motion of nearby bubbles. The existence of these *bubble interactions*, which occur when the source separation D is shorter than the wavelength ($D < \lambda$), prevents the estimation of the notional signature of the array based on a simple *superposition* of signatures from individual air guns. These interactions become more important at low frequencies and short source separations.

Following the analysis from Safar (1976) and Sinclair and Bhattacharya (1980), Ziolkowsky et al. (1982) proposed a solution to estimate the signature of the array including bubble interactions, based on the idea that the bubble motion is driven by the pressure difference between the bubble and the surrounding water. The solution consists in treating the interactions from nearby sources as a time-dependent modulation of the hydrostatic pressure at the location of the affected bubble.

This time-dependent, effective hydrostatic pressure $p_{e,i}(t)$ is different for each bubble and is equal to the sum of the standard hydrostatic pressure p_∞ and the total sound pressure from the interactions $p_{s,i}(t)$ at bubble i. The latter accounts for the acoustic pressure from the surrounding sources and ghost reflections. For the air gun i, the effective pressure is given by (Ziolkowsky et al., 1982; Li et al., 2011)

$$p_{e,i}(t) = p_\infty + p_{s,i}(t) \tag{5.95}$$

$$p_{s,i}(t) = \sum_{j \neq i} p_{ij}(t) \tag{5.96}$$

The parameter $p_{ij}(t)$ is the time-dependent acoustic pressure at bubble i due to bubble j, and can be expressed as

$$p_{ij}(t) = p_j(r_{ij}, \xi_{ij}) - \frac{r_{ij}}{r'_{ij}} p_j(r_{ij}, \xi'_{ij}) \tag{5.97}$$

$$\xi_{ij} = t - \frac{r_{ij}}{c_\infty} \tag{5.98}$$

$$\xi'_{ij} = t - \frac{r'_{ij}}{c_\infty} \tag{5.99}$$

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{5.100}$$

$$r'_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i + z_j)^2} \tag{5.101}$$

where (x_i, y_i, z_i) and (x_j, y_j, z_j) are the coordinates of i and j bubbles, r_{ij} and r'_{ij} are the travel distances between bubble i and j for the direct and ghost reflection, ξ_{ij} and ξ'_{ij} are the time functions for the direct and ghost reflections between bubble i and j, with r_{ij}/c_∞ and r'_{ij}/c_∞ being their respective delays. The second term on the right in (5.97) corresponds to the ghost reflection, which is equal to the direct signal $p_j(r_{ij}, \xi_{ij})$ with corrected amplitude and different time delay (a reflection factor $R = -1$ is assumed for the sea surface).

By replacing p_∞ with $p_{e,i}(t)$ in the equations of motion of the bubble, one can obtain the notional signatures of the individual sources with interactions. The superposition principle can be applied to these updated signatures to compute the notional signature of the array at any point in the space.

The effective pressure affects the enthalpy, and thus the dynamics of each bubble. The calculation of the acoustic output of an air gun array consists of four steps

1. An iterative solution of the equation of the bubble motion for each unique air gun.
2. Calculation of the radiated sound field for all bubble combinations $p_{ij}(t)$ and the total effective hydrostatic pressure at each bubble $p_{e,i}(t)$.
3. Iterative solution of the equation of bubble motion for each air gun, now including interactions ($p_\infty \rightarrow p_{e,i}(t)$).
4. Calculation of the radiated sound at a given point in the space for each air gun including interactions. The notional signature of the array is obtained by *linear superposition* of the individual acoustic signatures.

5.7 Parameter Optimisation with Simulated Annealing

5.7.1 Introduction to Simulated Annealing

The concept of *simulated annealing* was first introduced by Kirkpatrick et al. (1983), and two years later by Černý (1985). The two papers draw an analogy between statistical mechanics and combinatorial optimisation. The field of *statistical mechanics* studies the behaviour of a system of many degrees of freedom in thermal equilibrium, while *combinatorial optimisation* refers to the techniques employed to find the minimum of a multivariate function where the set of possible solutions is finite. The algorithm can be applied to problems defined in a continuous solution space through discretisation.

Simulated annealing is inspired by the *annealing process* in condensed matter physics by which the temperature is raised to a maximum value for a solid to melt, to then decrease in a controlled way. During the cooling-down period, the particles randomly rearrange until they reach a minimum energy *state* where they form a highly structured pattern (Aarts et al, 1997). When a particle changes its energy estate from E_i to E_j by "jumping" into another position, the system experiences a small change in energy $\Delta E = E_j - E_i$. These energy transitions are governed by the temperature of the system.

Metropolis et al (1953) introduced a statistical procedure to describe the behaviour of atoms in thermal equilibrium at a specific temperature. The system naturally progresses towards lower energy estates ($\overline{\Delta E} < 0$), but high energy ($\Delta E > 0$) transitions will occur during the process, specially at higher temperatures. The algorithm works as follows: from a current energy state E_i, the algorithm finds a subsequent state E_j by applying a small random displacement to an atom through a *perturbation mechanism*. The simulated transition is accepted straightaway if $\Delta E \leq 0$ and used as a reference state for the next transition. But if $\Delta E > 0$, the acceptance is treated probabilistically (Kirkpatrick et al., 1983). Hence, for the annealing process, the Metropolis algorithm defines the *probability of acceptance* of an energy transition ΔE at temperature T as

$$P_a(\Delta E) = \exp(-\Delta E / k_B T) \quad \text{for } \Delta E > 0 \tag{5.102}$$

$$P_a(\Delta E) = 1 \quad \text{for } \Delta E \leq 0 \tag{5.103}$$

where k_B is the Boltzmann constant and $\exp(-\Delta E / k_B T)$ is the Boltzmann or Gibbs distribution. A simple way of determining the acceptance of a transition when $\Delta E > 0$ is to compare $P_a(\Delta E)$ with a random uniformly generated number r between (0,1), and retain the transition if $P_a(\Delta E) > r$; if rejected, the original energy state is used for the calculation of the next transition (Kirkpatrick et al., 1983). The three basic steps implicit in the Metropolis algorithm (*perturbation* of the energy state E_i, and *evaluation* and *acceptance/rejection* of the new energy state E_j) are repeated until the system reaches the *ground state* ($T = 0$). In this process, the temperature is lowered following a pre-established pattern, referred to as *cooling schedule*.

There are a number of non-linear optimisation methods that improve on the classic simulated annealing approach. One example is the Adaptive Simplex Simulated Annealing (ASSA) from Dosso et al. (2001), a hybrid solution that combines a variant of the Fast Simulated Annealing (FSA) with the Downhill Simplex (DHS), allowing for efficient downhill moves while avoiding local minima. The ASSA adaptively controls the "jump" size based on previous search information. Results from fifteen geoacoustic inversion tests showed a clear gain in computational speed and accuracy for the ASSA when compared to the FSA and grid search methods. The method, originally developed for geoacoustic inversion, can be applied to any optimisation problem.

5.7.2 Implementation of the Classic Simulated Annealing Algorithm

But how does all this relate to combinatorial optimisation? The relationship becomes apparent by establishing an analogy between the terms in the annealing process and those in a combinatory optimisation problem. In multivariate optimisation, the *cost* (or objective) *function* is used to evaluate the "goodness" of a potential solution or *multivariate design vector* $\mathbf{x} = \{x_1, x_2, \ldots x_M\}$. In the annealing process, the *energy* represents the cost ($E \rightarrow C$, $\Delta E \rightarrow \Delta C$), the *state* is the test solution and the *ground state* the optimal solution. The term $k_B T$ is a control parameter with the same units as the cost function and is commonly referred to as *temperature* ($k_B T \rightarrow c$ or T). When the algorithm reaches the ground state, the system is said to be *frozen*.

A flow diagram of the standard simulated annealing algorithm is shown in Figure 5.4 and Figure 5.5. The process starts with the definition of an *initial temperature* T_0 and *initial design vector* $\mathbf{x_0}$. A *perturbation algorithm* $G(\mathbf{x})$ adds a small distortion to the previous vector $\mathbf{x}_i$ to obtain the new one, $\mathbf{x}_j$. The *objective function* $C(\mathbf{x})$ calculates the cost C_j associated with the new vector. Then, the *acceptance criteria* defined by the Metropolis algorithm decides what vector is to be retained ($\mathbf{x}_i$ or $\mathbf{x}_j$). The *cooling schedule* determines the evolution of the temperature T with time. This function controls the probability of acceptance of vector $\mathbf{x}_j$ when it causes an increase in cost ($\Delta C > 0$) and plays a critical role on the performance of the algorithm. The "perturbation → evaluation → acceptance/rejection" cycle is repeated multiple times for each temperature step until the *stop criteria* is reached. Reached that point, the design vector with lowest cost becomes the solution. The seven modules are described in detail in Appendix C.

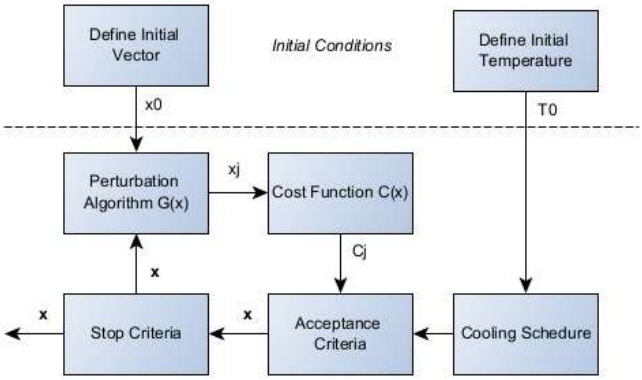

Figure 5.4 Flow diagram describing a standard simulated annealing algorithm.

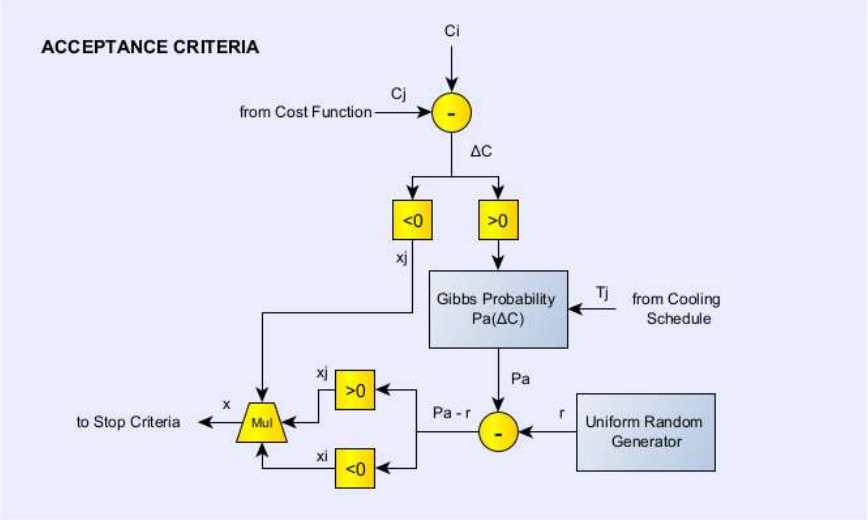

Figure 5.5 Flow diagram of the *acceptance criteria* in a standard simulated annealing algo-

From the acceptance criteria illustrated in Figure 5.5, one can see that the probability of accepting a vector that reduces the cost ($\Delta C < 0$ or "good jump") is very high, but there is still a small, non-zero probability that a transition into a higher-cost vector ($\Delta C > 0$ or "bad jump") is accepted. If only "good jumps" were considered, the algorithm would rapidly arrive at the stop condition before reaching the optimal solution. The high temperature value at the beginning of the search results in a high probability of acceptance of high-cost transitions, allowing the algorithm to visit different regions and in that way reduce the risk of getting stuck at a local minimum. As time progresses, the "jumps" are small, limiting the transitions to a small area until no more changes are allowed. These two stages are referred to as the *exploration* and *tuning* phases. This is an important feature of simulated annealing, which can be seen as a form of "divide-and-conquer" approach where the gross features of the system are revealed at the start to then focus on the finer details towards the end of the process (Kirkpatrick et al, 1983).

Another great advantage of simulated annealing compared to iterative improvement algorithms such as gradient-based methods is that it allows occasional transitions to more expensive solutions at high temperature (Skiena, 2008), giving the algorithm the chance to escape local minima at the beginning of the process. In addition, simulated annealing can outperform practically any other algorithm when long-enough processing times are allowed (Aarts et al, 1997), while revealing more than one good solution in a single execution. It is also easy to implement the algorithm and to define boundary conditions and constraints on the cost function and solution space (Ingber, 1993). The main negative aspects of simulated annealing are the long processing times required to find an optimal solution and the difficulty to fine-tune it to specific problems, which can easily lead to misuse and misinterpretations (Ingber, 1993; Aarts et al, 1997).

There are a few decisions that need to be made for the implementation and configuration of the simulated annealing algorithm (Johnson et al., 1989). The solution space $\mathcal{X}$, the initial solution $\mathbf{x_0}$, the cost function $C(\mathbf{x})$, and the perturbation algorithm $G(\mathbf{x})$ are problem-dependent parameters, while the initial temperature T_0, the cooling ratio α, the number of iterations per temperature N_T, and the stop criteria are generic parameters. The various stages of the algorithm and the recommended choice for their design parameters are discussed in the following sub-sections.

5.8 RAGAS – Routines for Air Gun Array Simulation

5.8.1 Introduction

In Sub-Section 2.7.1 we introduced the five air gun array modelling software applications that currently exist. Two of them, Gundalf™ and Nucleus+, are industry standards that require a costly license and include features specifically designed for seismic exploration and geophysical applications. CAGAM and AASM are non-commercial proprietary software from the CMST and JASCO. The only software that is freely available is AGORA. Gundalf™, CAGAM, AASM and AGORA (v.1, v.2) were compared during a workshop held in Dublin in 2016 (Ainslie et al., 2016). The models were benchmarked under three scenarios and substantial differences were observed in the spectrum at frequencies above 1 kHz. The mismatch was attributed to differences in the initial rise time of the primary pulse. The validation of the models was identified as a priority.

The workshop offered an opportunity for the developers of the software to interpret the differences in their results, question their approach and justify the model performance with supporting evidence. Three years later, two papers came out describing in detail the method followed for the implementation of CAGAM (Duncan & Gavrilov, 2019) and AASM (MacGillivrai, 2019; 2006), for which no peer-reviewed paper existed previously. No publication followed for AGORA, for which a conference paper already existed (Sertlek & Ainslie, 2015).

The largest mismatch in the acoustic signature was observed between AGORA and the rest of the models, with an amplitude difference for the primary peak and reflection of ~6 dB. As discussed in Sub-Section 2.7.1, AGORA applies the calibration parameters from the first calibrated version of AASM but it is not explicitly calibrated against a particular dataset or benchmarked model. The use of parameters that are not optimised for the model is the most likely cause of differences between AGORA and the rest of the models. The lack of calibration and errors found in the original code (AGORA v.1) motivated us to develop a new calibrated air gun array model with improved physics and features better suited for noise impact assessment.

The new model developed for this thesis is called RAGAS (Routines for Air Gun Array Simulation) and takes its name from the melodic patterns used in classical Indian music, resembling the sequential emission of pulses of different dominant frequencies in an air gun array. In the next three subsections we present the general features of the software and its general structure.

5.8.2 General Features

Below are listed the most relevant features of RAGAS

- Simple configuration through JSON files.
- Array layout through CSV file.
- Input management and error control routines.
- Two bubble dynamics models available: multi-physics and polytropic. The multi-physics model incorporates several physical phenomena suitable for the simulation of clusters and GI guns (Laws et al., 1990; Landrø, 1992). The polytropic model uses the polytropic relation from Gilmore (1952) and a radius-dependent damping coefficient (Ziolkowsky, 1970).
- Four bubble acceleration algorithms available: Gilmore (1952), Keller & Kollodner (1956), Rayleigh-Plesset (Prosperetti & Lezzi, 1985), and Herring (1941).
- The various physical phenomena included in the multi-physics model can be individually disabled to assess their contribution to the bubble motion.
- Ghost interaction can be enabled (or disabled) for the bubble interactions.
- Interactions can be enabled (or disabled) for the calculation of the signature of the air gun array.
- The ghost reflection can be enabled (or disabled) for the calculation of the signature of the air gun array.
- Two acoustic radiation formulas available: first order and second order.

- Calculation of multiple acoustic metrics from the air gun array signature. The metrics include the strength (PBR37) and bubble period, and the amplitudes and times of the primary pulse and bubble.
- Representation of the normalised (1 m) array signature, spectrum, horizontal directivity and vertical directivity for the specified azimuth, elevation and frequencies.
- Acoustic signature filtered with zero-phase bandpass filter. The software uses the filtering libraries developed for ABACUS by the author (see Chapter 3).
- Storage of bubble dynamics results for optimum speed.
- Calibration of model parameters using measurements from the Svein Vaage dataset (Mattson et al., 2012; Mattson, 2010) of two models of single air gun (Bolt, G-Gun) for two receiver locations and several chamber pressures and volumes.

5.8.3 Software Structure

RAGAS has been designed with five goals in mind: flexibility, efficiency, precision, usability and open-access. *Flexibility* in the sense that all the necessary options are available to simulate any specific scenario (e.g., environment parameters, physics and algorithms). The *efficiency* is achieved by avoiding unnecessary processing steps, optimising the code and storing recurrent data. MATLAB is less efficient than other programming languages, but the nature of the algorithms and the code optimisation result in processing times of the order of a few seconds for a large array. The *precision* is achieved through the selection of appropriate algorithms and the calibration of the model parameters. Creating a realistic model is certainly the most delicate and difficult part of the design process and careful selection of the physics and empirical (calibration) parameters of the model is key. The *usability* is guaranteed by producing clean, well-documented code and creating functions that can be easily integrated with other software. Having a reliable *open-access* model will encourage new research from institutions or individuals that cannot afford a license. This is an important aspect of the development of RAGAS, and it is hoped that it will boost the publication of independent research that can shed new light on the impacts of seismic surveys on marine life.

Another aspect of the *usability* of RAGAS is the generation of outputs that can be used in the context of Environmental Impact Assessments (EIA). Industry standards such as Gundalf™ and Nucleus+ are designed for geophysicists, who are interested in the vertical signature of the source with ghost reflection. However, in EIA applications it is the sound that travels horizontally that is of most interest. Additionally, sound propagation models used in EIA to predict the range-dependent sound levels already account for reflections from all surfaces, making the ghost reflection from the source model redundant. The researcher who needs to simulate the sound field produced by an air gun

[37] *Pulse to Bubble Ratio* is the amplitude relation between the primary peak and the bubble

array for EIA purposes typically resorts to Gundalf™ or Nucleus+ to obtain the acoustic signature of the array. From the author's experience, a modelling request from Gundalf™ or Nucleus+ for a specific array layout always consists of a far-field signature in the vertical direction with ghost reflection and two frequency-dependent directivity graphs for the crossfire (E-W) and endfire (N-S) directions. Obtaining the free-field (i.e., ghost-free) acoustic signature of the array in the direction of interest ($\theta > \theta_c$) using this information is simply not possible, except in the very exceptional case where all the sources in the array are identical. The researcher is then left with a vertical, ghost-affected far-field signature, roughly corrected in amplitude to account for the vertical directivity of the array. This output is not a fair representation of the true free-field acoustic signature of an air gun array at a given direction.

With this in mind, let's introduce the general structure of the software. Figure 5.6 represents the different function modules, databases, configuration files of the software and their interaction. RAGAS can be split into three categories: (1) the *mechanical routines* dedicated to the modelling of the bubble motion, (2) the *databases* consisting of two bubble motion databases (with and without bubble interactions), and (3) the *acoustic routines* that compute the air gun signatures at the specified elevations and azimuths from the bubble motion databases to produce four types of graphs (horizontal directivity, vertical directivity, time signature and source level spectrum).

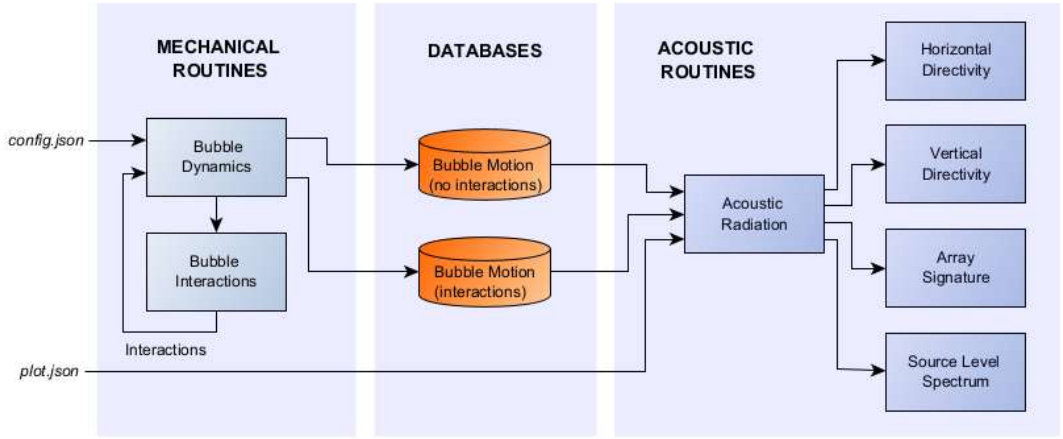

Figure 5.6 Flow chart of RAGAS code. Core function modules are shown as *blue rectangles*, and databases as *orange pies*. The global modules are shown as light blue rectangles.

There are two general functions in RAGAS: `runRagas.m` and `plotRagas.m`. The first function runs the mechanical routines and generates the bubble motion databases based on the content of the settings file `config.json`. The second function uses the information in the bubble motion databases and the settings in `plot.json` to compute the acoustic signature at a number of points, to then produce four acoustic graphs (time signature, source level spectrum, and horizontal and vertical directivities). The settings `config.json` and `plot.json`, the array layout `array.csv`, and the bubble motion databases are stored in a folder specified in the file `root.json` saved in the software's

root directory. RAGAS is organised in two folders: 'Core', which contains the function modules shown in Figure 5.6, and 'Libraries', which includes additional packages used by RAGAS. Those packages are the 'Digital Filtering (Single)', 'FFT Filtering (Single)' and 'FFT Filtering (Bank)' developed by the author and described in Appendix B.

A description of the seven function modules and bubble motion databases is given below.

Bubble Dynamics

There are two main functions in the *Bubble Dynamics* module for the computation of bubble motion parameters: `multiphysicsDynamics.m` and `polytropicDynamics.m`.

`multiphysicsDynamics.m` computes the bubble dynamics using (1) the method described by Gilmore (1952) and Ziolkowsky (1970) for the calculation of the acceleration at the bubble wall du_b/dt, and (2) the thermodynamic approach from Landrø (1992) to establish a relationship between the bubble radius r_b and the pressure p_b. The function incorporates the multiple physics described by Laws et al. (1990), shown to improve the agreement between model and simulations for clustered air guns (< 1 m separation). The function is also compatible with two-chamber generator-injector (GI) guns (Landrø, 1992). The approach from Laws et al. (1990) is intended to provide an additional damping, recognised by Ziolkowsky (1970) to be needed on the original formulation from Gilmore (1952) to compensate for the overly resonant behaviour of the bubble pulses.

The physics included in this function are the following: (1) the flow of air mass dm_a/dt due to the throttling process, (2) the flow of vapour mass dm_v/dt through condensation-vaporisation at the bubble interface, (3) the alteration of the bubble temperature T_b during the throttling process due to the Joule-Thomson effect, (4) the heat transfer dq_w/dt caused by heat conduction through the bubble wall, (5) the heat loss through latent heat dq_v/dt due to condensation-vaporisation at the bubble interface, (6) the effective viscosity $\mu_{w,eff}$ due to turbulent eddies caused by the relatively large velocity at the bubble wall, (7) the bubble raise dz_b/dt due to its buoyancy, and (8) the presence of the air gun body V_{gun} inside the bubble. Every single of them can be disabled through an input argument to test their contribution to the bubble dynamics.

`polytropicDynamics.m` computes the bubble dynamics using (1) the method described by Gilmore (1952) and Ziolkowsky (1970) for the calculation of the acceleration at the bubble wall du_b/dt and (2) the polytropic relation from (5.14) proposed by Gilmore to establish a link between the bubble radius r_b and the pressure p_b. This formulation does not allow the implementation of physics that can be used to account for the complex processes that occur inside the bubble. The only parameters not included in the original formulation from Gilmore and considered for this function are the air gun presence V_{gun}, the air gun efficiency η, and the effective viscosity $\mu_{w,eff}$. In this case, the efficiency η is used to calculate the equilibrium radius r_{0b}, unlike `multiphysicsDynamics.m` where η takes part in the transfer of air mass during port throttling.

There are three consequences to the neglect of the physical processes inside the bubble: (1) the model is underdamped, (2) the model is not suitable for cluster simulations, and (3) the model cannot be applied to GI guns. Ziolkowsky (1970) proposed a solution for the first limitation that consisted in applying exponential attenuation to the simulated time response of the bubble radius $r_b(t)$, as shown in (5.21).

`multiphysicsDynamics.m` accepts an `Interactions` structure containing the pressure experienced by each bubble as a result of the acoustic radiation from surrounding bubbles and ghost reflections (if bubble interactions are enabled). An input `Calibration` structure can also be provided with specific values for the calibration parameters (τ, β, η, ε, k_μ). `polytropicDynamics.m` also accepts an `Interactions` and a `Calibration` structure. In this case, the calibration parameters are (γ, α, η, k_μ).

The two functions return a `Bubble` structure containing five sub-structures: (1) `Config`, with the settings from `config.json`, (2) `Array`, containing the layout of the array and the parameters of its air guns (pressure, volume, radius) (3) `Interactions`, with the acoustic pressure at the source location caused by surrounding bubbles, (4) `Calibration`, with the calibration parameters for each source, and (5) `Dynamics`, containing the time-dependent bubble parameters for each source. `Interactions`, `Array`, `Calibration` and `Dynamics` are multi-element substructures where each element corresponds to one source in the array. The main time-dependent parameters in the `Dynamics` structure include: the time axis; the mass, pressure, and temperature in the bubble and inside the gun; the enthalpy, velocity, radius, volume, and depth of the bubble; and the water density, sound speed and hydrostatic pressure. The equilibrium radius is also included. `polytropicDynamics.m` only generates outputs for the bubble pressure, enthalpy, velocity, radius and depth, and the water density, sound speed and pressure.

Bubble Interactions

This module contains only one function: `bubbleInteractions.m`. The function receives a `Bubble` structure and returns a multi-element `Interactions` structure. Each element in `Interactions` includes the interacting pressure at the affected source, and the identifiers of the affected source and emitting sources.

The user can choose whether to include or ignore the ghost reflections as sources for bubble interaction. Initially, it may seem unnecessary to have an option to enable or disable the ghost for the interactions. Surely, if the ghost is to be included in the acoustic signature of the array, should not also be considered for the bubble interactions? This is not always the case and there is a practical application for it. In sound propagation modelling we do not want to include the ghost reflection as part of the air gun signature, as it is already accounted for by the propagation algorithm. However, we do want to include the effect of the ghost on the bubble dynamics, since the propagation model has no knowledge of the underlying behaviour of the source. In other words, if we wanted

to generate an acoustic signature for it to be used in a sound propagation model, the ghost should be enabled for bubble interactions but disabled from the acoustic signature.

Acoustic Radiation

This module contains one main function: `bubbleRadiation.m`. The function calls several subroutines, but for simplicity we will only describe its general functionality.

`bubbleRadiation.m` receives a `Bubble` structure, a vector of receiver positions and a flag indicating whether the ghost is to be included in the acoustic signature. The function calls the first-order or second-order approximation algorithm to compute the radiated acoustic field from the bubble motion parameters. The time response of the acoustic pressure and particle velocity at 1 m from the source and relevant metrics are calculated.

`bubbleRadiation.m` returns a `Radiation` structure containing four substructures: (1) `Config`, with the settings of the model, (2) `Array`, specifying the layout of the air gun array and the parameters of the individual guns, (3) `Signature`, containing the radiation parameters for each receiver position, and (4) `Metrics`, with the relevant metrics for each signature. The `Signature` structure includes the time axis, pressure and particle velocity at 1 m, receiver position, array's central position, and the distance, elevation, and azimuth between the receiver and the array. The `Metrics` structure includes the signature strength (PBR), the bubble period, and a number of parameters for the *primary* pulse and the *bubble*, such as the positive peak, negative peak, peak-to-peak pressures, and the broadband and third-octave band RMS pressures.

Plotting

There are four main functions in this module for representing the results from the acoustic radiation simulations: `plotSignature.m`, `plotSpectrum.m`, `plotHorizontalDirectivity.m`, and `plotVerticalDirectivity.m`.

`plotSignature.m` displays the time-dependent signature of the array at 1 m at the selected elevation and azimuth. Figure 5.7 shows an example of the output of this function for a 4-gun 230 in^3 array at 20° elevation and 0° azimuth. The elevation is referred to the horizontal and the azimuth is 0° for the y-direction of the array's layout. The red and cyan circles represent the position of the primary peak and first bubble pulse.

`plotSpectrum.m` displays the third-octave band spectrum or the power spectral density (PSD) of the array at 1 m at the selected elevation and azimuth. The sound pressure level (SPL RMS) or the sound exposure level (SEL) can be represented. Figure 5.8 shows the TOB and PSD spectra of the signature from Figure 5.7.

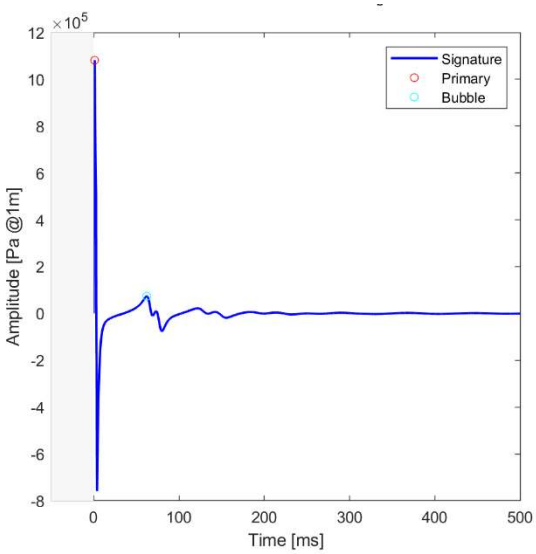

Figure 5.7 Far field acoustic signature of a 230 in^3 air gun array (2x 45 in3, 2x 70 in3) at 1 m, 20° elevation and 0° azimuth. $PBR = 14.7$ dB, $\tau_b = 61$ ms.

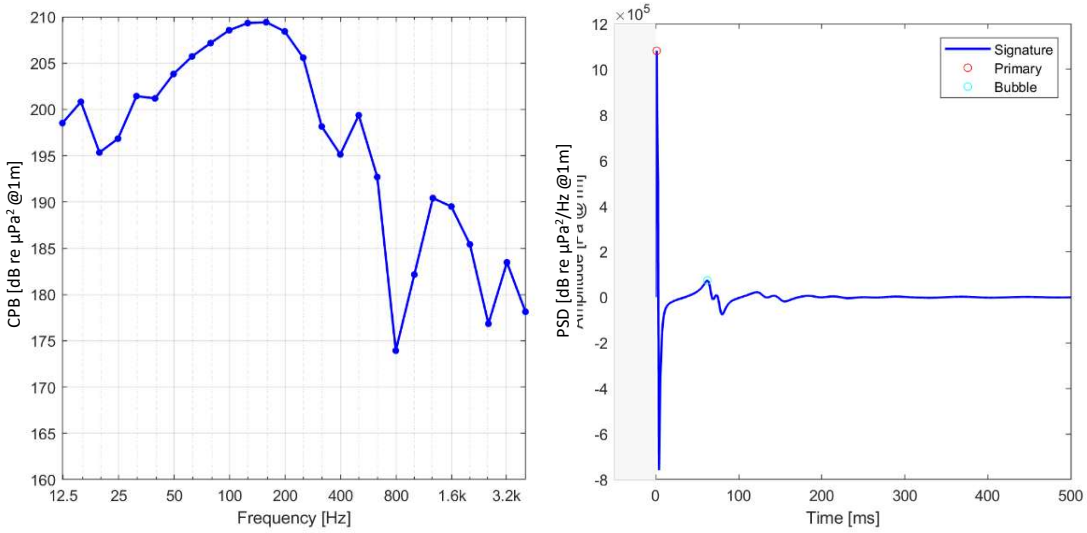

Figure 5.8 Far field TOB spectrum (left) and PSD (right) a 230 in^3 air gun array (2x 45 in3, 2x 70 in3) at 1 m, 20° elevation and 0° azimuth. Source level RMS.

`plotHorizontalDirectivity.m` displays the directional pattern of the air gun array at the horizontal plane that crosses the receiver position. The directivity is a value between 0 and 1, where 1 corresponds to the direction of maximum pressure. The directivity can be represented for the broadband signal or a number of frequencies. Figure 5.9 shows the horizontal directivity of a 4-gun 230 in^3 array at 20° elevation for the central TOB frequencies of 50, 100 and 200 Hz.

`plotVerticalDirectivity.m` displays the directional pattern of the air gun array at the vertical plane that crosses the receiver position. Figure 5.10 shows the vertical directivity of a 4-gun 230 in^3 array at 0° azimuth for the broadband signal.

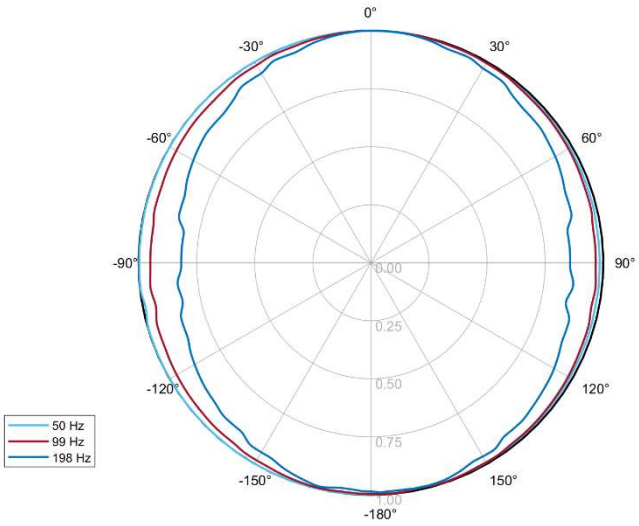

Figure 5.9 Horizontal directivity of a 230 in^3 air gun array (2x 45 in3, 2x 70 in3) at 20° elevation. The directivity is represented for central TOB frequencies of 50, 100 and 200 Hz.

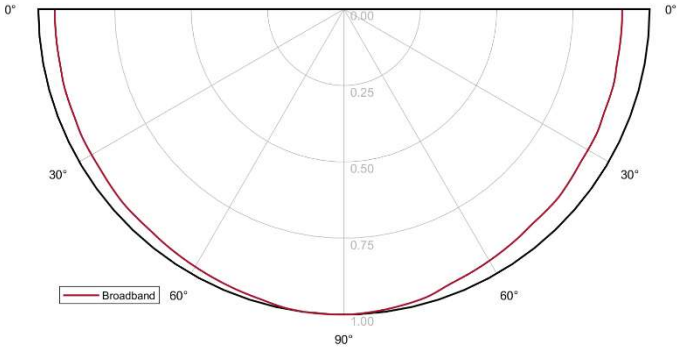

Figure 5.10 Vertical directivity of a 230 in^3 air gun array (2x 45 in3, 2x 70 in3) at 0° azimuth.

5.9 Effects of Model Parameters on the Bubble Motion

In this section, we study the effects that the differential magnitudes described in Section 5.3 have on the bubble radius and acoustic signature. This analysis is intended to provide a better understanding of the contribution of the various physics considered in the *multi-physics* model to the bubble dynamics and radiation (see 'Bubble Dynamics'). For this purpose, the differential magnitudes are "enabled" and "disabled" or, where possible, the associated control parameters and magnitudes are modified. This will help not only gain a better insight into the effects of the contributing physics, but also into the degree by which the calibration parameters alter the bubble response. The results can help identify the calibration parameters that are most relevant, and in that way simplify the calibration process. In particular, the analysis can help identify those physical processes that have a less significant contribution to the bubble motion and acoustic radiation and can be safely discarded without a negative impact on the model's performance. The section ends with two subsections where air gun clusters and GI guns are discussed.

In the figures included in the next sub-sections, all the acoustic signatures are generated in the far field (at 1000 m distance from the source) in the vertical direction ($\theta = 90°$) and under free-field conditions (i.e., no ghost reflection), and are normalised to 1 m. Unless stated otherwise, the black curve represents a reference scenario defined by the following parameters: port-throttling factor $\tau = 0.1$, port throttling exponent $\beta = 0.5$, air gun efficiency $\eta = 0.8$, heat transfer coefficient $\varepsilon = 20000\ \mathrm{W\,m^{-2}\,K^{-1}}$, air gun volume $V_g = 250\ \mathrm{in^3}$, and air gun pressure $p_g = 2000\ \mathrm{psi}$. In this scenario, all the physics are included except for the air gun presence ($V_{gun} = 0$) and the bubble raise ($z_b = z$).

5.9.1 Mass Transfer Rate of Air dm_a/dt

The mass transfer rate of dry air due to port throttling depends on five constants, as shown by equations (5.30) and (5.34): the air gun pressure p_g and volume V_g, the port throttling factor τ and exponent β, and the air gun efficiency η. In this sub-section we look at the effect of these five parameters on the bubble motion and radiation and compare the results of some of them within the initial *release* phase. The bubble motion at the beginning of the emission has been noticed to have a large influence on the rest of the oscillation process, hence our interest in having a closer look at this region.

Air Gun Volume V_g

The volume in the air gun chamber has a significant effect on the amplitude and period of the bubble motion, as shown in Figure 5.11. The reduced momentum associated with the smaller mass of low air gun volumes will result in a lower bubble expansion under the same environmental conditions. The lower pressure difference between bubble and water at the start of the compression phase will limit how small the bubble radius can be. This also causes a shorter bubble period. Overall, lower air gun volumes have smaller equilibrium radius, smaller radius about equilibrium, and shorter bubble periods.

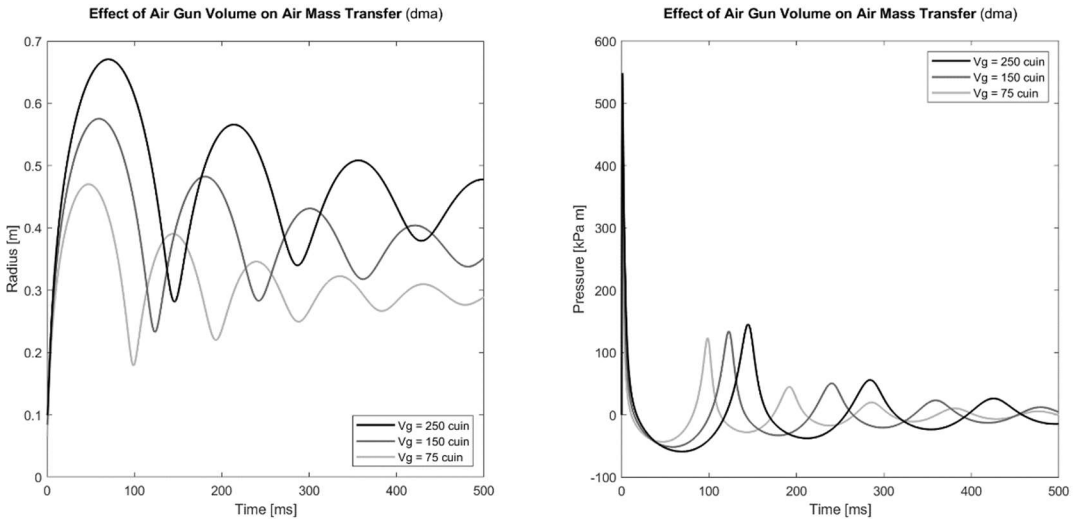

Figure 5.11 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for three air gun chamber volumes: 250 in^3 (black), 100 in^3 (grey), and 75 in^3 (light grey).

Air Gun Pressure p_g

The pressure in the air gun chamber has an effect similar to the volume V_g. Lower air gun pressures result in smaller bubble radius, radiated pressure and bubble period. The smaller pressure difference between the air gun and water causes a lower bubble speed. Note, however, that the release time is similar for all gun pressures.

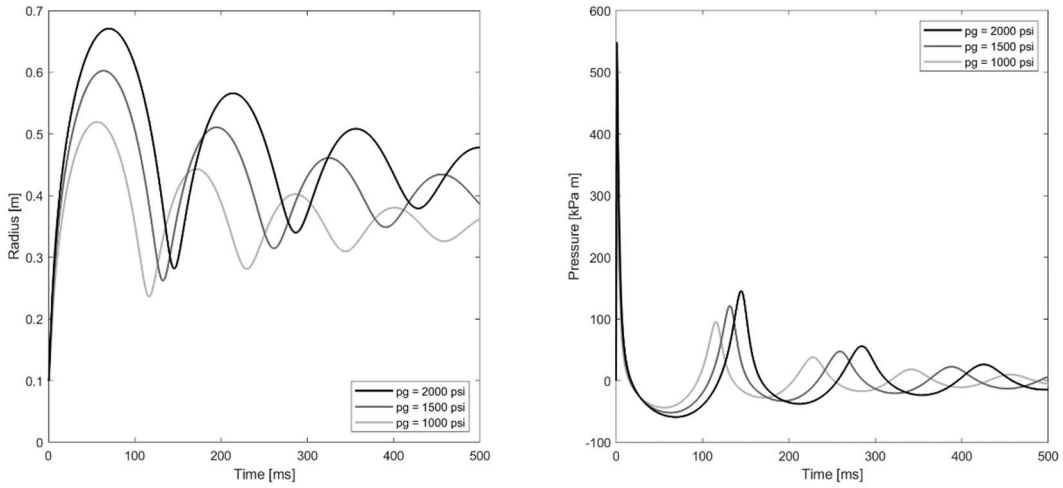

Figure 5.12 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for three air gun chamber pressures: 2000 psi (black), 1500 psi (grey), and 1000 psi (light grey).

The similarities between Figure 5.11 and Figure 5.12 cannot be justified only by the higher momentum transferred into the bubble from the pressure difference between air gun and water. In Figure 5.12 the equilibrium radius r_{0b} decreases at lower air gun pressures. This is easily explained by the larger air mass that can be stored in the air gun at higher pressures. Since the air mass m_a is the only air gun parameter that r_{0b} depends on, as shown in (5.20), it becomes clear that lower pressures are linked to a smaller amount of air in the chamber. This is justified by (5.31) and the modelling assumption that the air gun is submerged long enough for its temperature to be similar to the temperature of the water, in which case the enclosed air mass becomes proportional to the pressure in the gun.

Comparison between Air Gun Pressure p_g *and Volume* V_g

The air gun volume V_g and pressure p_g have been shown to have a similar effect on the overall time response of the bubble radius and acoustic signature. Let's now look closer into the actual differences within the initial *release* phase. Larger gun pressures produce a *faster air release*, while larger volumes do not have a significant effect on the release gradient (i.e., bubble wall velocity u_b). Additionally, from Figure 5.11 and Figure 5.12 one could see that larger gun volumes and pressures result in a larger bubble radius. Interestingly, something that could not be clearly observed in those figures is how lower volumes do not have a significant effect on the amplitude of the primary acoustic pulse while lower pressures can affect this amplitude considerably. This is consistent with the

notion that the source strength is proportional to the air gun pressure p_g and to the third root of the volume $V_g^{1/3}$ (Dragoset, 2000).

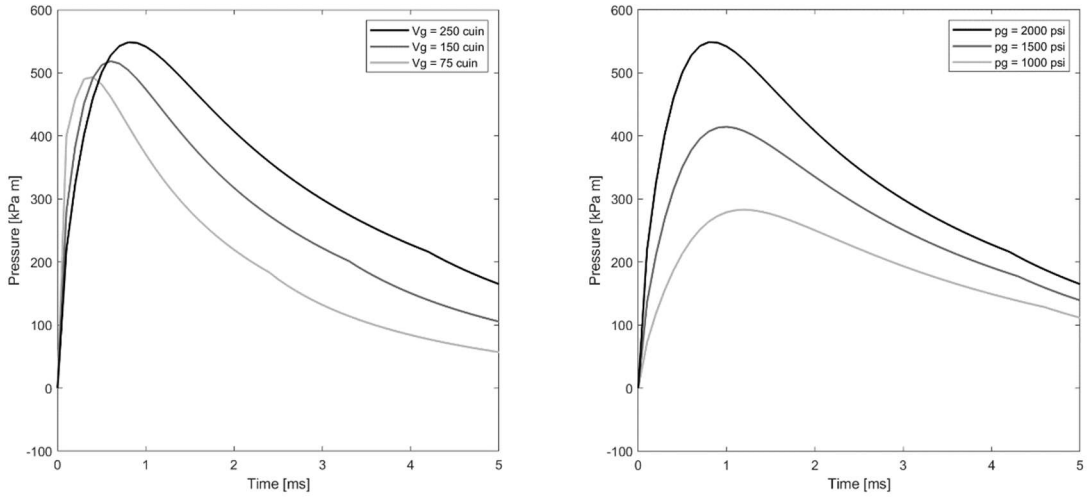

Figure 5.13 Time evolution of far-field pressure at 1 m for three air gun chamber volumes (left) and pressures (right). Close-up view of the initial phase (0-5 ms).

Port Throttling Exponent β

The port throttling exponent β modifies the air gun volume to produce a factor that controls the mass transfer rate. Since β is a positive number lower than 1 and the volume (in m^3) is lower than 1, larger values of β will reduce the mass transfer rate. The result is a noticeable reduction of the bubble radius about equilibrium and a reduction in the amplitudes of the primary pulse and bubbles. No relevant effect on the bubble period can be observed within this range of values (see Figure 5.14).

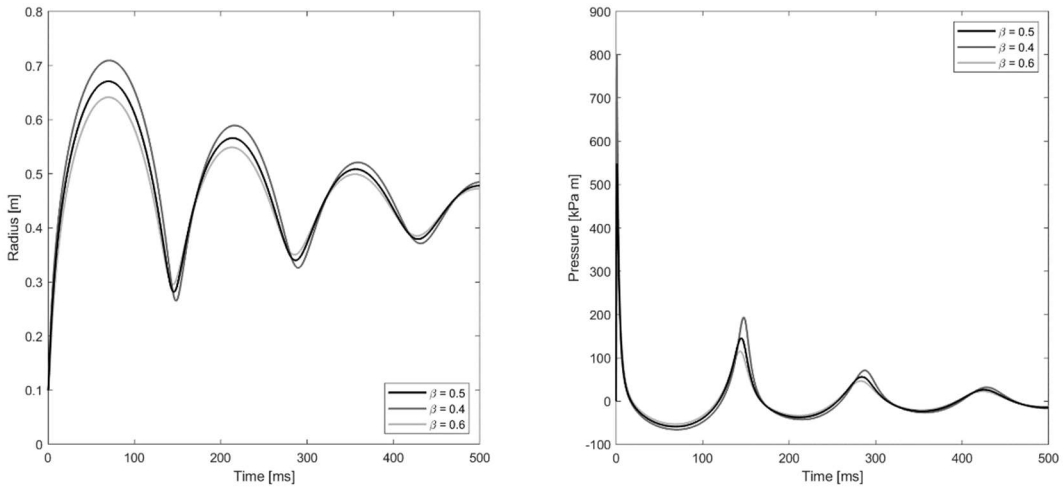

Figure 5.14 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for three values of the exponent β: 0.5 (black), 0.4 (grey), and 0.6 (light grey).

Port Throttling Factor τ

The port throttling factor τ directly controls the magnitude of the transfer rate of air mass dm_a/dt. Thus, lower values of τ result in lower air mass transfer rates. The effect of lowering τ is the same as that of increasing the values of the port throttling exponent β, namely a reduction in the bubble radius about equilibrium and a reduction in the maximum pressure of the primary pulse and bubbles. Larger values of τ (or lower values of β) will produce a slight increase in bubble period for the range of values considered in practice.

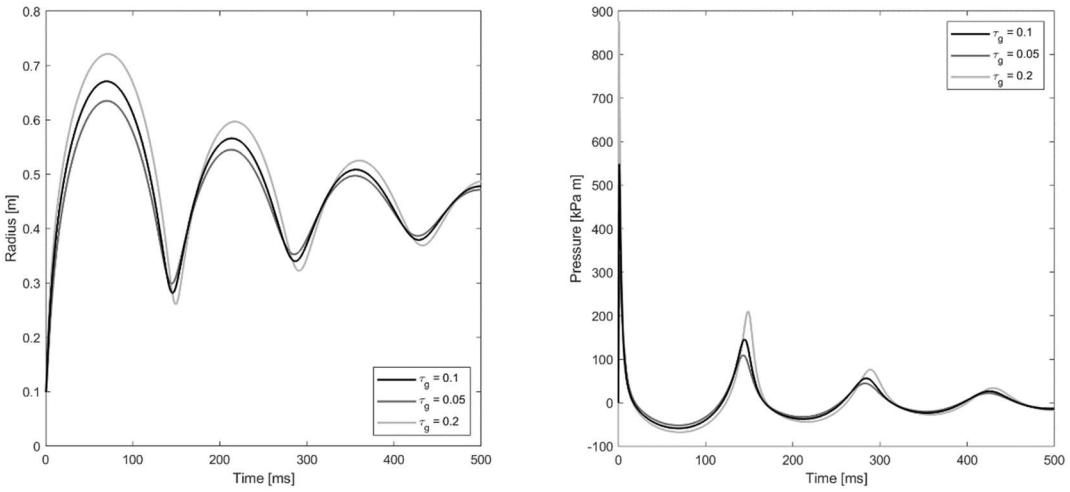

Figure 5.15 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for three values of the factor τ_0: 0.1 (black), 0.05 (grey), and 0.2 (light grey).

Air Gun Efficiency η

The efficiency of the air gun η determines the fraction of air that is released from the air gun chamber. When η is lower than unity, the release of air stops prematurely. Lower values of η will cause a reduction in the bubble period and the equilibrium radius r_{0b}. By the time the air gun has released all the air that its efficiency allows, the initial release phase has already passed. The consequence is that η will not have a significant effect on the magnitude of the acoustic signature (see next point).

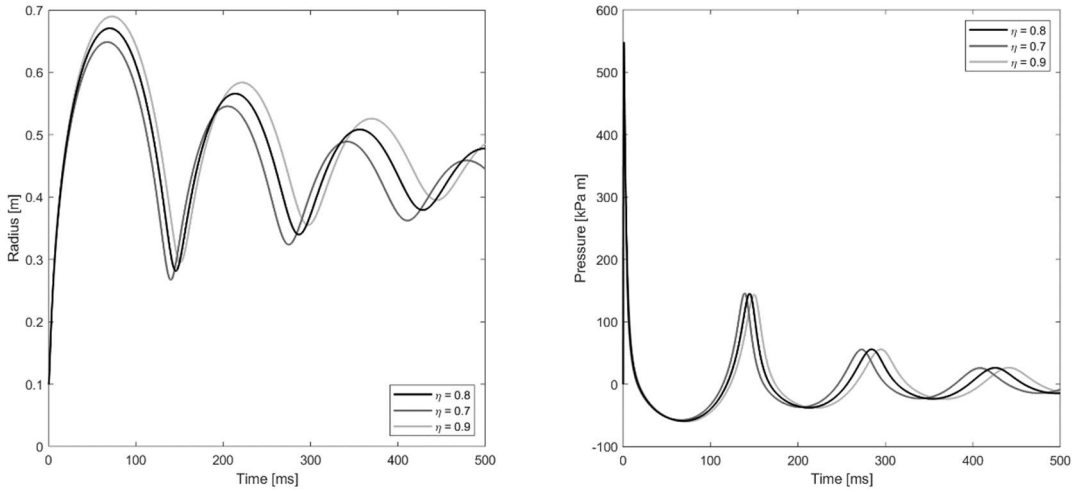

Figure 5.16 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for three values of air gun efficiency η: 0.1 (black), 0.05 (grey), and 0.2 (light grey).

Comparison between Port-Throttling Factor τ and Air Gun Efficiency η

The port throttling factor τ and the air gun efficiency η have a significantly different effect on the acoustic signature during the initial phase. Lower values of τ result in a slower air release and lower acoustic pressures. The τ factor accounts for the contribution of the design and dimensions of the air gun exhaust port, which will determine the release speed of the enclosed air depending on its characteristics. The air gun efficiency η has no effect on the amplitude of the primary pulse, since the end of the air release phase occurs after the acoustic pressure reaches its maximum.

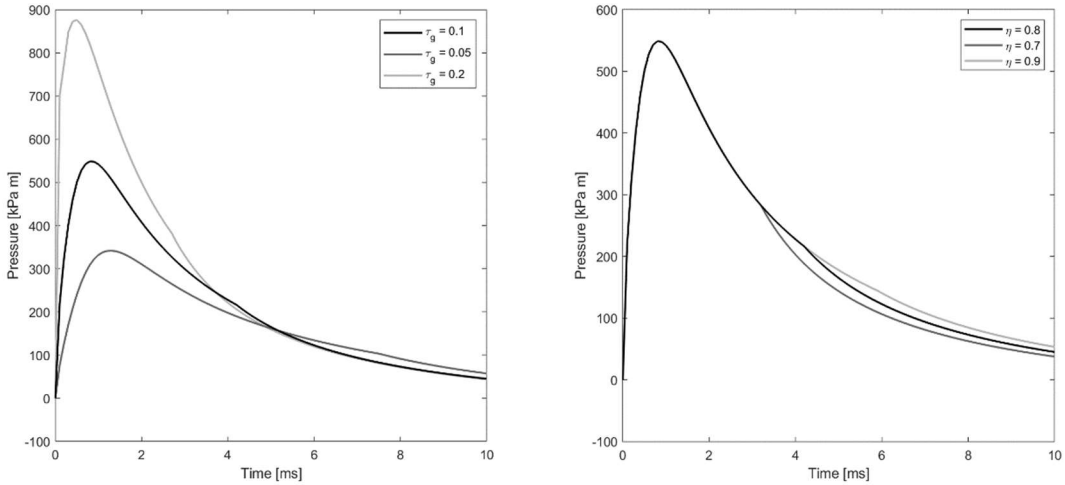

Figure 5.17 Time evolution of far-field pressure at 1 m for three port throttling factors τ (left) and three air gun efficiencies (right). Close-up view of the initial phase (0-10 ms).

5.9.2 Mass Transfer Rate of Water Vapour dm_v/dt

The effect of the mass transfer rate of water vapor due to condensation-vaporisation at the bubble wall is shown in Figure 5.18. No significant differences exist for the time responses of the bubble radius and acoustic radiation between the cases of "enabled" and "disabled" mass transfer of water vapour.

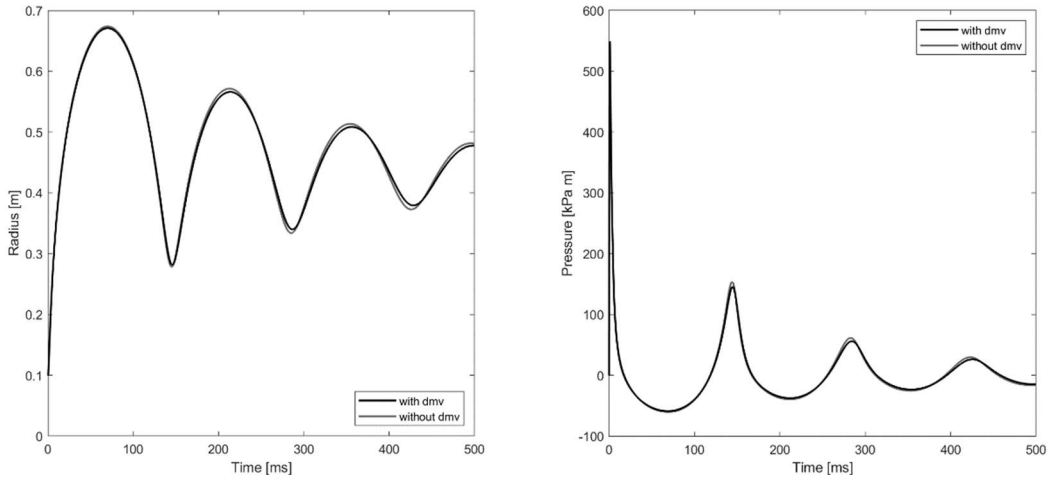

Figure 5.18 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) with mass transfer of water vapour (black) and without it (grey).

5.9.3 Heat Transfer Rate through Conduction dQ'_w/dt

Figure 5.19 shows the effect of the heat transfer between bubble and water through conduction for three values of heat transfer coefficient ε. A larger value of ε increases the heat transfer dQ'_w/dt, as shown by (5.68).

A positive value for dQ'_w/dt represents the transfer of heat from the bubble into the water, and therefore indicates a loss in energy. A larger value of dQ'_w/dt affects the rate of change in temperature dT_b/dt, according to (5.62). During the *compression phase*, the positive heat transfer rate (energy loss) contributes to lowering the temperature in the bubble. That lower bubble temperature during compression hinders the ability of the bubble to counteract the compression process, which further reduces the bubble radius. Then, during the *expansion phase* the heat transfer rate becomes negative (energy gain) and a raise in bubble temperature occurs. The increase in bubble temperature enhances the ability of the bubble to grow due to its higher internal energy.

With this in mind, if the heat transfer coefficient increases, one should expect lower bubble temperatures during compression and higher temperatures during expansion, with the resulting increase in bubble radius about equilibrium. This effect can be clearly observed in Figure 5.19.

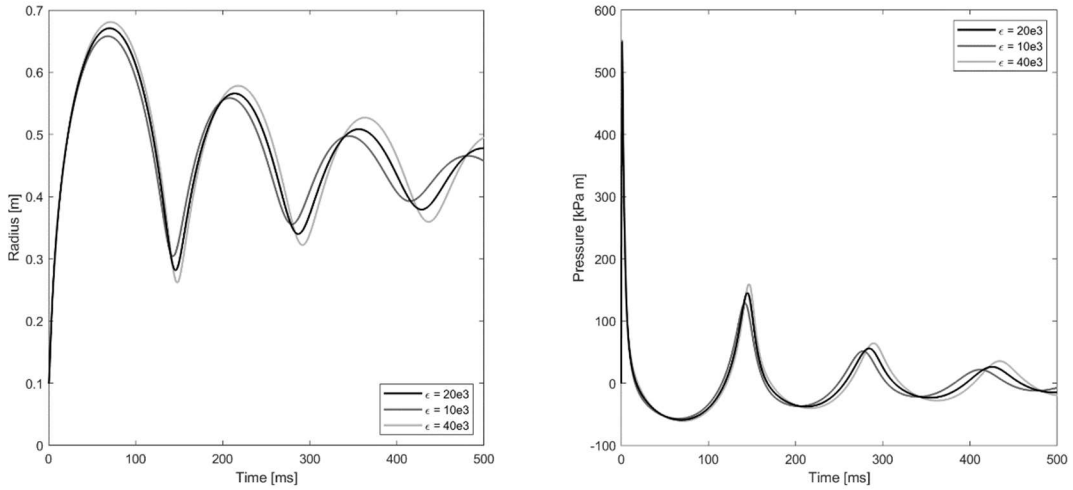

Figure 5.19 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for three values of heat transfer coefficient ε: 20 (black), 10 (grey), and 40 (light grey) kW · m^{-1} · K^{-1}.

5.9.4 Heat Transfer Rate from Latent Heat dQ_v/dt

The effect of the heat loss due to vaporisation (latent heat) dQ_v/dt on the bubble radius and acoustic radiation is shown in Figure 5.20. Like the mass transfer rate of water vapour dm_v/dt (see Sub-Section 5.9.2), controlled by the same processes of condensation-vaporisation, the heat loss dQ_v/dt has no significant effect on the responses of the bubble radius and acoustic radiation.

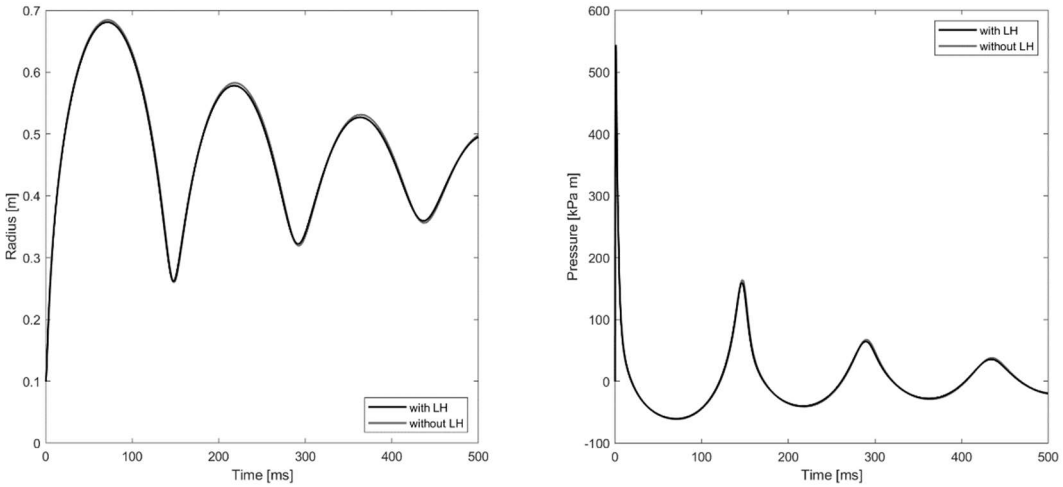

Figure 5.20 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) with heat loss through latent heat (black) and without it (grey).

5.9.5 Temperature Change from Joule-Thomson Effect dT_b/dt

The effect of the change in bubble temperature as a result of the Joule-Thomson effect (JTE) during port throttling is shown in Figure 5.21. The JTE appears to have a critical effect on the bubble dynamics. For the typical pressures used for seismic air guns, the JTE causes the pressure drop between the gun chamber and the bubble to be accompanied by a drop in the temperature of the released air. The cooler air injected into the bubble decreases the bubble temperature during expansion. The throttling process is fast and lasts only a few tens of milliseconds. During this initial release period the temperature in the bubble is considerably lower when applying the JTE. As a result, the pressure in the bubble during the release phase is dramatically reduced, limiting the ability of the bubble to expand further.

The significant contribution of the JTE within the initial release phase has a defining effect on the motion of the bubble for the remaining time. The bubble radius about equilibrium is reduced, and so is the bubble period.

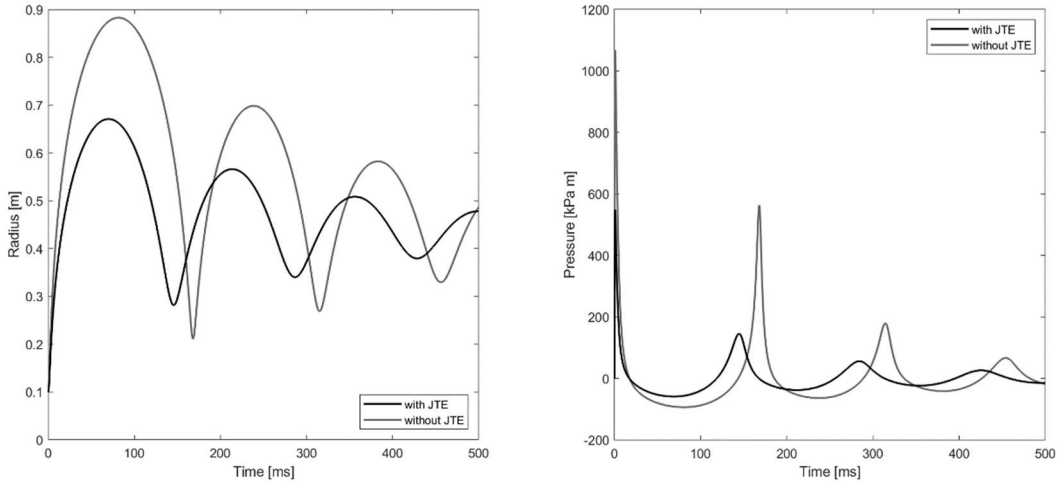

Figure 5.21 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) with Joule-Thomson effect (black) and without it (grey).

5.9.6 Bubble Raise Speed dz_b/dt

The effect of the raise speed of the air bubble in water on its radius and acoustic radiation is shown in Figure 5.22. As the bubble raises, the hydrostatic pressure that the bubble is exposed to decreases. This, in turn, increases the equilibrium radius r_{0b}. This reduction in r_{0b} produces a slight decrease in the acoustic pressure as the bubble approaches the sea surface. When the bubble reaches the surface, the radiation stops. In this particular example the curves stop at ~300 ms, which is the time it takes for the bubble to travel from a 2 m depth to the sea surface.

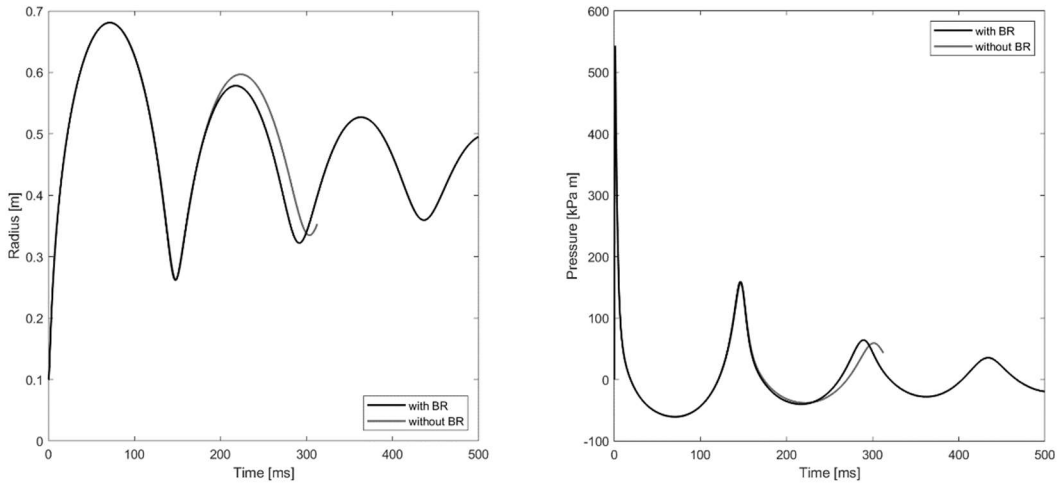

Figure 5.22 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) with bubble raise (black) and without it (grey). Source depth $z = 2\,\mathrm{m}$.

5.9.7 Effective Viscosity

The effect of the effective viscosity $\mu_{w,e}$ on the bubble radius and acoustic radiation is shown in Figure 5.23. The dynamic viscosity mainly affects the bubble pressure p_b. A larger value for the dynamic viscosity μ_w reduces the pressure in the bubble. The effect of μ_w is greater during the expansion phase, during which the bubble velocity u_b is large and the bubble radius r_b is small. The lower bubble pressure contributes to the damping of its oscillations, which translates into a smaller radius about equilibrium and a lower amplitude in the acoustic signature.

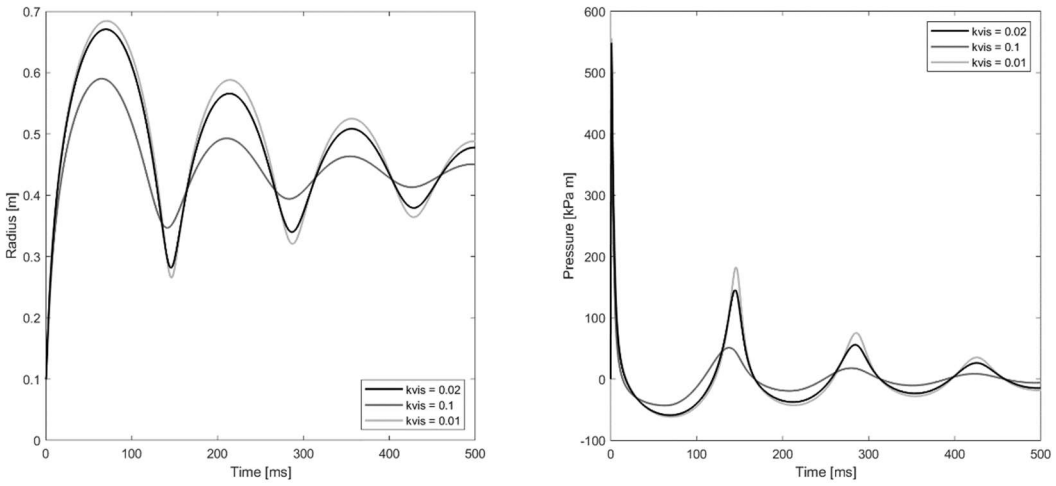

Figure 5.23 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for three values of effective viscosity factor k_μ: 0.02 (black), 0.1 (grey), and 0.01 (light grey) kg m^{-1} s^{-1}.

The constant k_μ is used as a control parameter for the effective viscosity $\mu_{w,e}$ (see Sub-Section 5.3.7). A larger value of k_μ increases $\mu_{w,e}$, damping the bubble motion as a result, as discussed above. This damping effect can be clearly seen in Figure 5.23 by comparing the results for three different values of k_μ. An increase in dynamic viscosity has no noticeable effect on the bubble period.

5.9.8 Air Gun Presence V_{gun}

The effect of the air gun presence on the bubble motion and acoustic radiation is shown in Figure 5.24. During the first several tens of milliseconds the bubble can be considered to be *static* (i.e., it does not experience a significant displacement towards the sea surface). Within this period, a portion of the air gun body occupies the centre of the bubble. When the air is released, the same radius encloses a smaller mass of air. The volume of gas in the bubble V_b for a given radius r_b is now smaller than before, when the presence of the air gun was ignored. As a result, the bubble pressure is effectively increased. The minimum radius of the bubble and the equilibrium radius r_{0b} are increased, and the bubble period experiences a slight reduction.

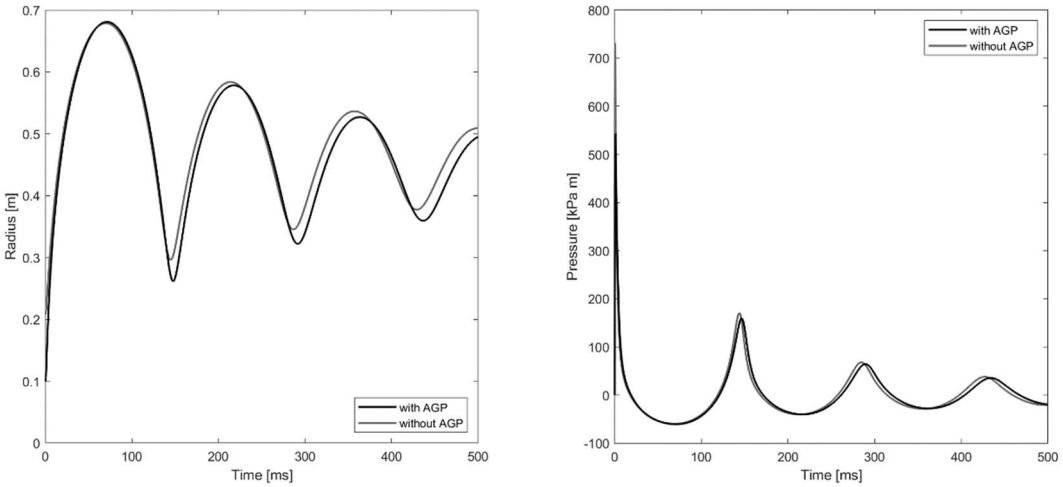

Figure 5.24 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) with air gun presence (black) and without it (grey).

5.9.9 Generator-Injector

A GI air gun can be modelled in RAGAS through the definition of two chamber volumes, one for the generator (V_{gG}) and one for the injector (V_{gI}). Figure 5.25 shows an example of a GI gun with a 45 in^3 generator and a 150 in^3 injector. This type of air gun can be adapted to reduce the volume and release duration of the injector with the use of volume reducers inserted in the chamber and exhaust port reducers. The injection time is controlled electronically. In this example, we present the simulated bubble motion and acoustic signature of the 45G-150I GI gun for the three conventional modes:

(1) *gun mode*, where only the generator is used (45G-0I), (2) *harmonic mode*, where the volumes of the generator and injector match (45G-45I), and (3) *true GI mode*, where the injector has a larger volume than the generator (45G-150I).

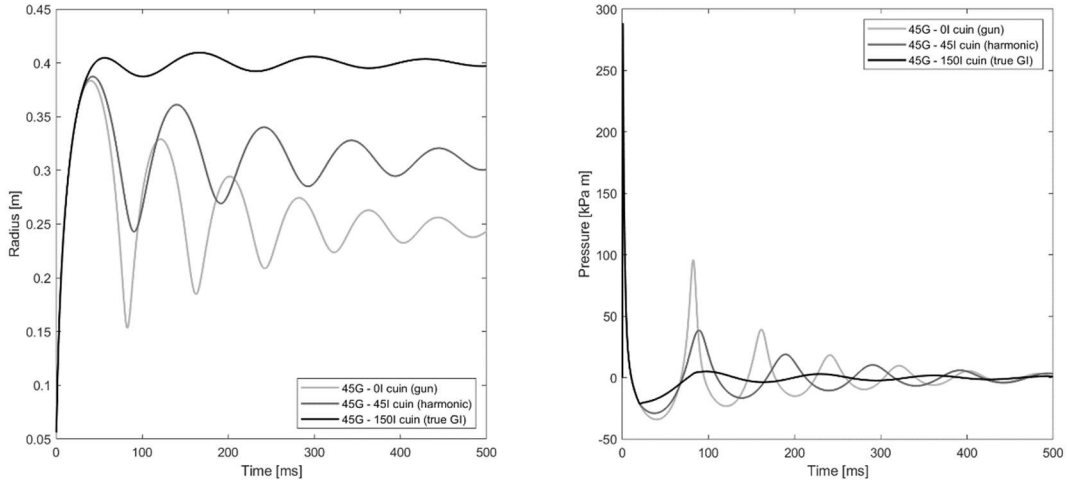

Figure 5.25 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for a 45 in^3 GI gun for three modes: *gun* (45G - 0I) (black), *harmonic* (45G – 45I) (grey), and *true* (45G – 150G) (light grey). Injection time 20 ms and injection period 40 ms.

One can see how the injector reduces the bubble oscillations. The best results are obtained with the true GI mode. In this mode, the bubble radius practically stabilises after the injection. The stabilisation of the bubble radius dramatically reduces the amplitude of the oscillations in the acoustic signature, as is the final purpose of GI guns.

5.9.10 Air Gun Clusters

The multi-physics bubble dynamics algorithm included in RAGAS can model air gun clusters. A cluster is a combination of two to four air guns placed close together. The short spacing between them makes their bubbles coalesce so that they can produce a more efficient acoustic signature than that of a single air gun (Dragoset, 2000). Note, however, that the formulation from Laws et al. (1990) for clustered air guns, and employed in RAGAS, does not support *bubble coalescence* (i.e., merging). Still, strong interactions between nearby bubbles will occur without coalescence taking place.

Figure 5.26 and Figure 5.27 show the bubble radius and acoustic signature of a cluster consisting of a 45 in^3 and a 20 in^3 air gun with 0.5 m separation. In the first figure, the bubble interactions have been disabled to better appreciate the effect of *bubble synchrony*. In Figure 5.26, the two bubbles oscillate independently, each characterised by a different equilibrium radius, maximum radius about equilibrium and bubble period. In Figure 5.27 the bubble interactions are enabled, and one can see how the smaller bubble adapts to match the motion of the largest bubble. More details about the synchrony between the bubbles in a seismic cluster and the use of multi-physics to obtain a realistic simulation of their signature can be found in Laws et al. (1990).

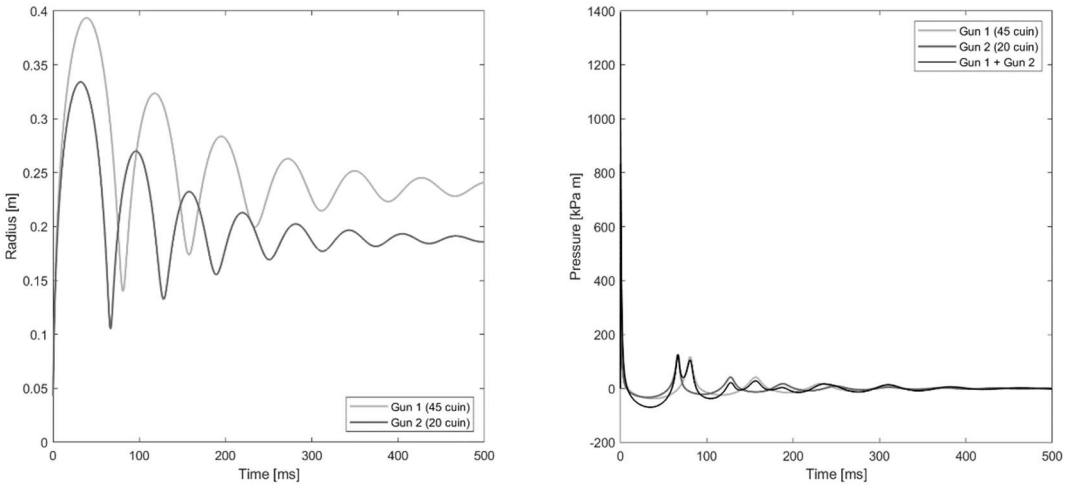

Figure 5.26 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for an air gun cluster with guns of 45 in^3 (light grey) and 20 in^3 (grey) separated 0.5 m. The acoustic signature of the two guns combined is shown in black. No bubble interactions.

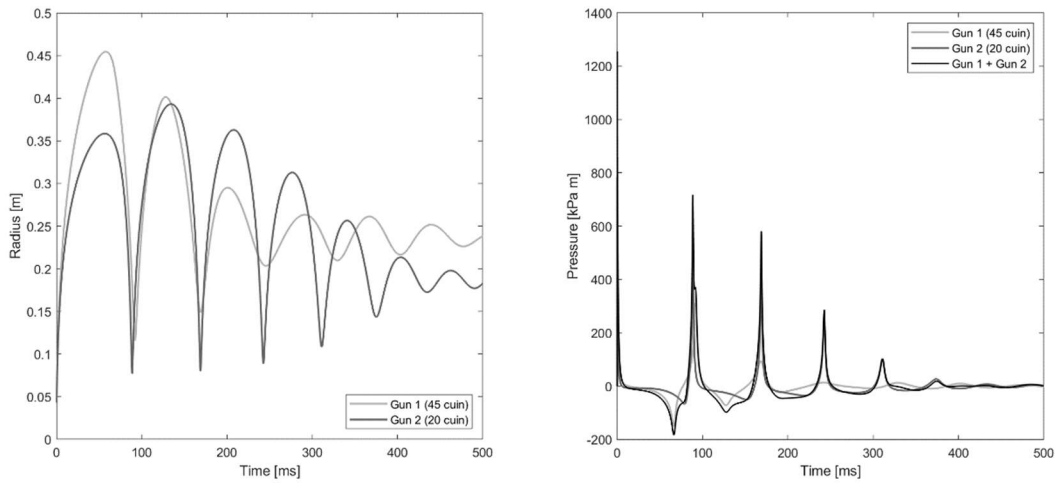

Figure 5.27 Time evolution of bubble radius (left) and far-field pressure at 1 m (right) for an air gun cluster with guns of 45 in^3 (light grey) and 20 in^3 (grey) separated 0.5 m. The acoustic signature of the two guns combined is shown in black. Bubble interactions included.

5.10 Calibration of RAGAS with Simulated Annealing

5.10.1 Svein Vaage Dataset

The Svein Vaage dataset is the most complete dataset to date of far-field and near-field measurements of individual air guns and clusters. The data was recorded as part of a project commissioned by E&P Sound and Marine Life, JIP (Mattson, 2010; 2012). The project was conducted to measure the acoustic signatures of the most common air guns used by the oil and gas industry at frequencies up to 50 kHz. The motivation of the

project was to create a dataset that could be used for the improvement and development of seismic air gun modelling software capable of predicting air gun signatures at frequencies higher than 1 kHz, for which the understanding of the physical processes is limited.

Measurements were made over a range of volumes, pressures and depths for each tested air gun model using 19 hydrophones and 1 accelerometer. Two hydrophone arrays were used: one on a side at distances from 15 to 20 m from the source, and the other centred below the air gun at distances of 30 m and 100 m. Near-field hydrophones were also deployed at known positions close to the source. The tests were conducted in an area with a water depth of over 400 m, which ensured free-field conditions (i.e., 500 ms reflection-free signatures).

5.10.2 Calibration Methodology

The calibration of RAGAS was performed on the multi-physics model considering interactions from bubbles and ghost reflections for the computation of the bubble dynamics. The air gun presence and the bubble raise were disabled for the calibration. The calculation of the acoustic signature included the ghost reflection.

Instabilities arose for some modelling scenarios when the air gun presence was considered, even in the simplest configuration of the multi-physics model and on the polytropic model. These instabilities require further investigation. The bubble raise was disabled due to the large simulated surfacing speed of the bubble which stopped the signal after 300 ms, before the bubble collapse was actually seen in the measured signature. The omission of these two model features do not affect its quality and performance because the rest of modelled physics and control parameters have the ability to adjust, when combined, every aspect of the signature (see Section 5.9).

Five control parameters have been optimised during the calibration process: the port throttling factor of the generator chamber τ_G and exponent β, the heat transfer coefficient ε, the air gun mass transfer efficiency η, and the effective viscosity factor k_μ.

Two air gun models have been selected for calibration: a Bolt 1500LL from Teledyne, and a G-Gun source from Sercel. The acoustic signatures are extracted from the Svein Vaage dataset from channels 0 and 13. Channel 0 corresponds to the second hydrophone from the bottom of the vertical corner array, placed at position x = 10.8 m, y = 9.8 m and z = 14.5 m, at a distance of 19.2 m from the source. Channel 13 corresponds to the top hydrophone in the vertical centre array, placed at position x = 0 m, y = 0 m, z = 30 m at a distance of 28 m from the source. For all the selected signatures, the source is placed at position x = 0 m, y = 0 m, z = 2 m. We have selected two signatures for the Bolt model, one per target channel, for a 150 in^3 and 2000 psi air gun. For the G-Gun, we have selected 8 signatures for each channel corresponding to the following configurations: 60 in^3 (500, 1000, 2000 psi), 150 in^3 (500, 1000, 2000 psi), 250 in^3 (500,

1000 psi). Out of the 50 to 70 shots recorded for each measured air gun configuration in the Svein Vaage dataset, we have taken the 40th shot for the calibrations, for being the one that showed to be accurately measured across all the selected configurations.

For the optimisation of the control parameters, the simulated annealing (SA) code developed for this thesis was used (see Section 5.7). The optimisation process consisted in two basic steps: (1) selecting, loading and conditioning the measured signature from the database, and (2) running the SA code. The SA code calls a *cost function* that returns a cost value for an input calibration vector. In this case, the input is the five-element vector $[\tau_G, \beta, \varepsilon, k_\mu, \eta\,]$.

The *cost function* is at the core of the optimisation algorithm. We have opted for the Napierian logarithm of the mean square error between the conditioned measured and modelled signatures. At each evaluation of the cost function, the acoustic signature is simulated for the specific source properties, environment conditions, receiver position and various other model-related settings. The simulated signature is then filtered and aligned with the measured signature to compute the cost. Multiple evaluations can be performed in a second, making the simulated annealing process relatively efficient for this particular optimisation problem.

The SA code was configured with an initial temperature $T_0 = 0.5$, previously estimated with (C.8), a minimum temperature $T_{min} = 0.005$, a cooling factor $\alpha = 0.9$, and a maximum number of transitions accepted within a temperature step $\mathsf{MaxAccept} = 20$. The rest of parameters for the stop criteria are irrelevant since $\mathsf{MaxAccept}$ and T_{min} became the stopping conditions in all SA executions.

5.10.3 Calibration Results

The history of the SA process for one of the measured signatures is represented in Figure 5.28. The top left image shows the evolution of the cost with each iteration and is represented along with the temperature. The rest of cost-variable graphs represent the convergence towards the value of each variable.

A good indicator of the convergence of the algorithm to the optimal solution is a fluctuating or scattered cost pattern at the start of the SA process with a slow progressive decay towards the end (see top-left image). In a cost-variable graph, a "U-like" shape is a good sign that the algorithm has converged to an optimal value for the depicted variable. The variable numbers 1, 3, 4, 5 and 6 correspond to $[\tau_G, \beta, \varepsilon, k_\mu, \eta\,]$. The variable 2, which corresponds to the port-throttling factor of the injector in a GI gun (τ_I), is not included here as it was set as constant.

The calibrated control parameters, averaged over the values obtained for each selected air gun signature used for optimisation, are shown in Table 5.1. The exact values of the calibrated control parameters are listed in Table 5.2.

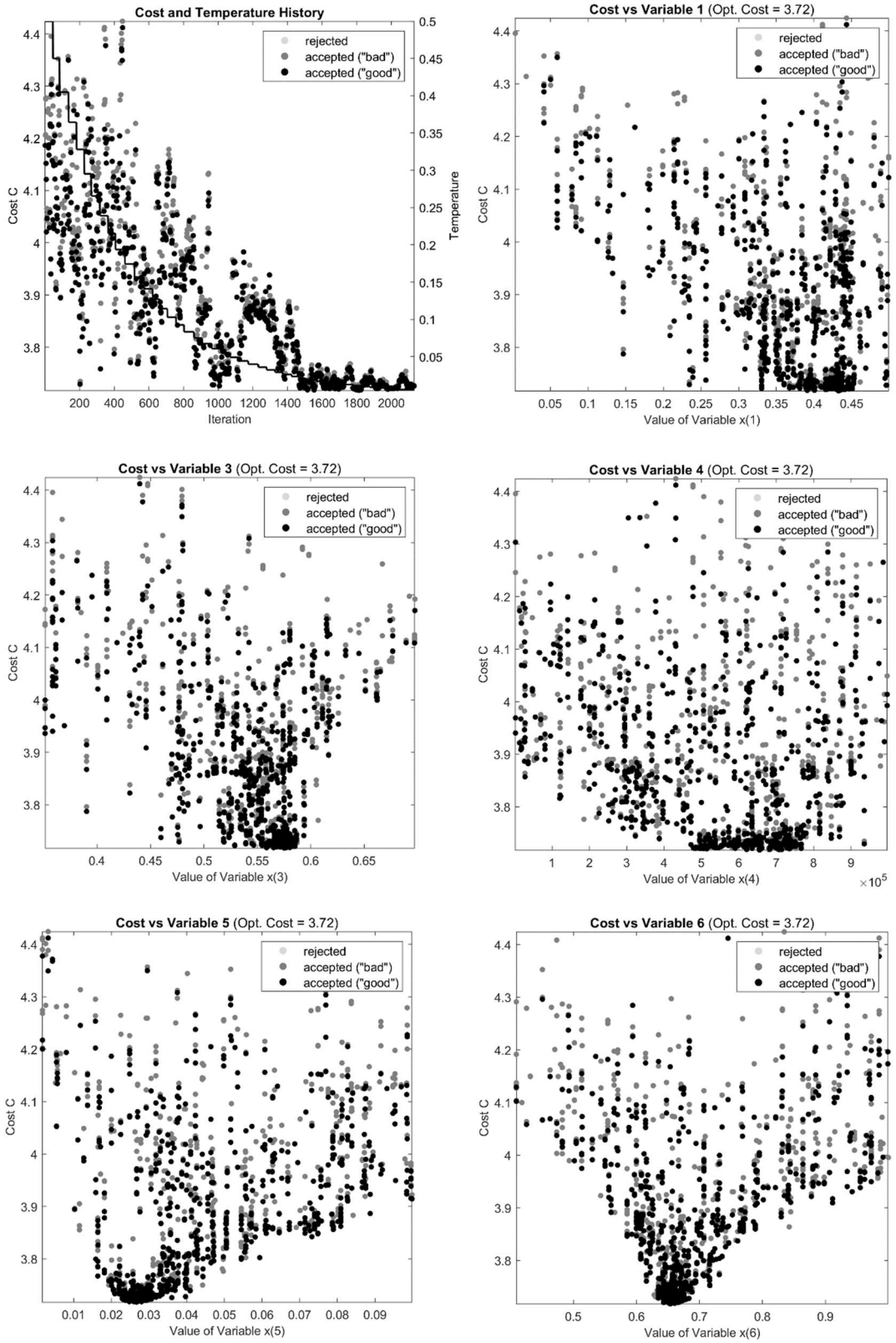

Figure 5.28 Calibration history of Bolt gun (150 in^3, 2000 psi, 2 m depth) for channel 13. From left to right and top to bottom: (1) cost and temperature history, and cost-variable graph for variables (2) τ_g, (3) β, (4) ε, (5) k_μ, (6) η.

Table 5.1 Calibration parameters for the Bolt and G-Gun averaged over the values in Table 5.2

Gun Model	τ_0	β	ε [W m^{-2} K^{-1}]	k_μ	η
Bolt	0.40	0.57	675606	0.027	0.65
G-Gun	0.27	0.54	450636	0.012	0.80

Table 5.2 Exact calibration parameters for air gun signatures optimised with simulated annealing

Gun Model	Channel	V_g [in^3]	p_g [psi]	τ_0	β	ε [W m^{-2} K^{-1}]	k_μ	η
Bolt	0	150	2000	0.411	0.550	766467	0.028	0.625
Bolt	13	150	2000	0.433	0.575	655539	0.027	0.654
G-Gun	0	60	500	0.399	0.576	818400	0.002	0.765
G-Gun	13	60	500	0.388	0.629	951227	0.011	0.753
G-Gun	0	60	1000	0.298	0.534	721171	0.002	0.782
G-Gun	13	60	1000	0.282	0.588	684200	0.008	0.805
G-Gun	0	60	2000	0.244	0.565	586824	0.013	0.850
G-Gun	13	60	2000	0.398	0.631	439120	0.012	0.872
G-Gun	0	150	500	0.381	0.623	323315	0.004	0.775
G-Gun	13	150	500	0.172	0.534	443756	0.010	0.770
G-Gun	0	150	1000	0.234	0.553	328516	0.010	0.791
G-Gun	13	150	1000	0.059	0.358	354625	0.015	0.794
G-Gun	0	150	2000	0.459	0.660	247205	0.013	0.834
G-Gun	13	150	2000	0.432	0.638	388819	0.023	0.840
G-Gun	0	250	500	0.413	0.626	257883	0.014	0.805
G-Gun	13	250	500	0.102	0.370	272368	0.011	0.812
G-Gun	0	250	1000	0.158	0.453	301823	0.013	0.768
G-Gun	13	250	1000	0.090	0.393	289987	0.020	0.779

5.10.4 Calibrated Model

This sub-section presents a comparison between the measurement and simulation for all the signatures selected from the Svein Vaage dataset. The measured signature (black) is represented along with the simulations from RAGAS, run with the average (light grey) and exact (grey) calibration parameters from Table 5.1 and Table 5.2.

The model shows very good agreement with the measurements when both the exact (grey) and average (light grey) calibration parameters are used. The worst agreement happens with the amplitude of the primary peak when the average calibration parameters are used.

The behaviour of the calibrated version of RAGAS is very good overall and shows that this model could be used as a reliable prediction tool for air gun array signatures. Still, better results could have been found by letting the SA run for longer periods with more demanding settings. Additional research will be necessary to identify a way to improve the matching of the positive peak. The initial conditions were identified to be of critical importance for the evolution of the bubble motion, and this can a potential way of improving the accuracy of the model.

Bolt Gun

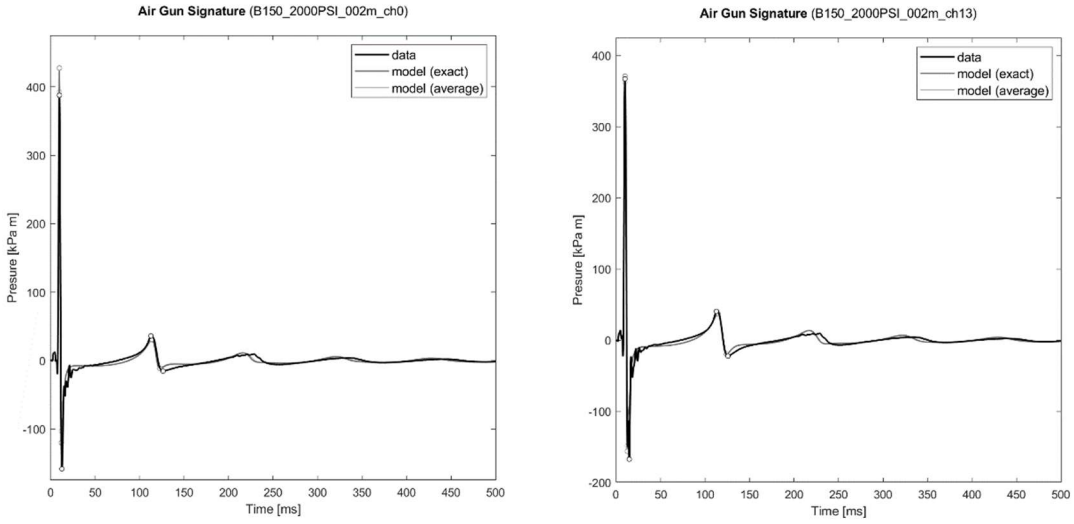

Figure 5.29 Far field acoustic signature at 1 m of Bolt gun (150 in^3, 2000 psi, 2 m depth) for channels 0 (left) and 13 (right). Comparison between data (black), and simulation with exact (grey) and averaged (light grey) calibration parameters.

G-Gun

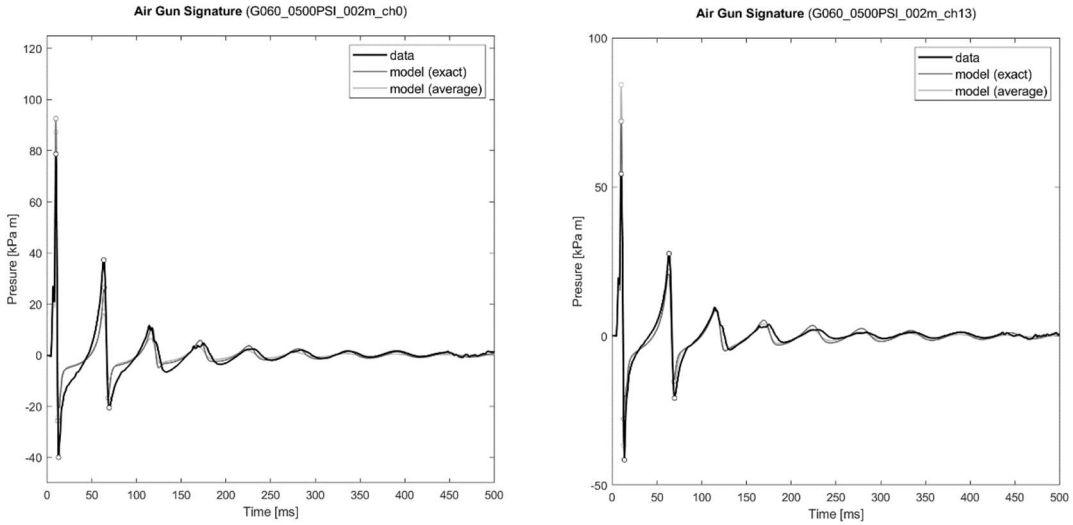

Figure 5.30 Far field acoustic signature at 1 m of G-Gun (60 in^3, 500 psi, 2 m depth) for channels 0 (left) and 13 (right). Comparison between data (black), and simulation with exact (grey) and averaged (light grey) calibration parameters.

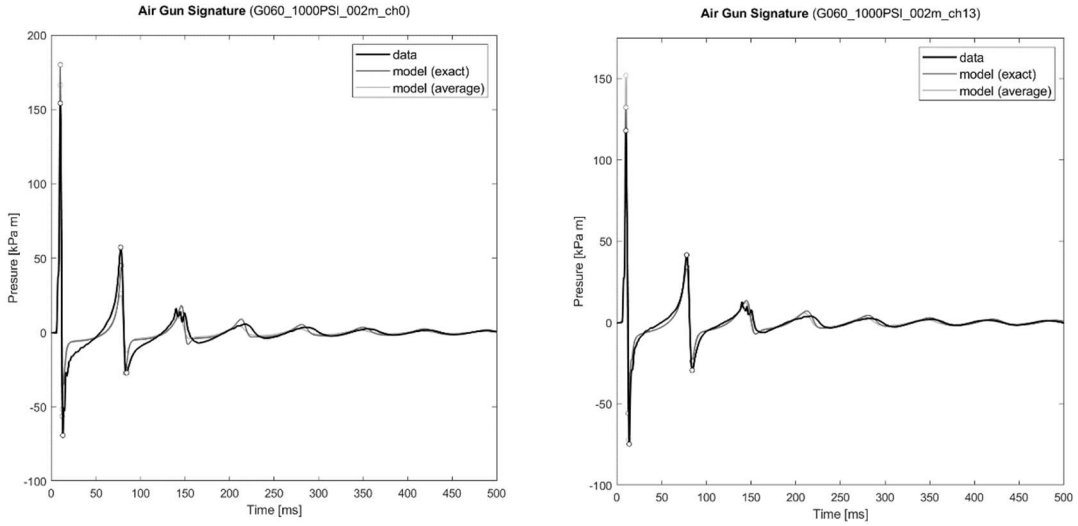

Figure 5.31 Far field acoustic signature at 1 m of G-Gun (60 in^3, 1000 psi, 2 m depth) for channels 0 (left) and 13 (right). Comparison between data (black), and simulation with exact (grey) and averaged (light grey) calibration parameters.

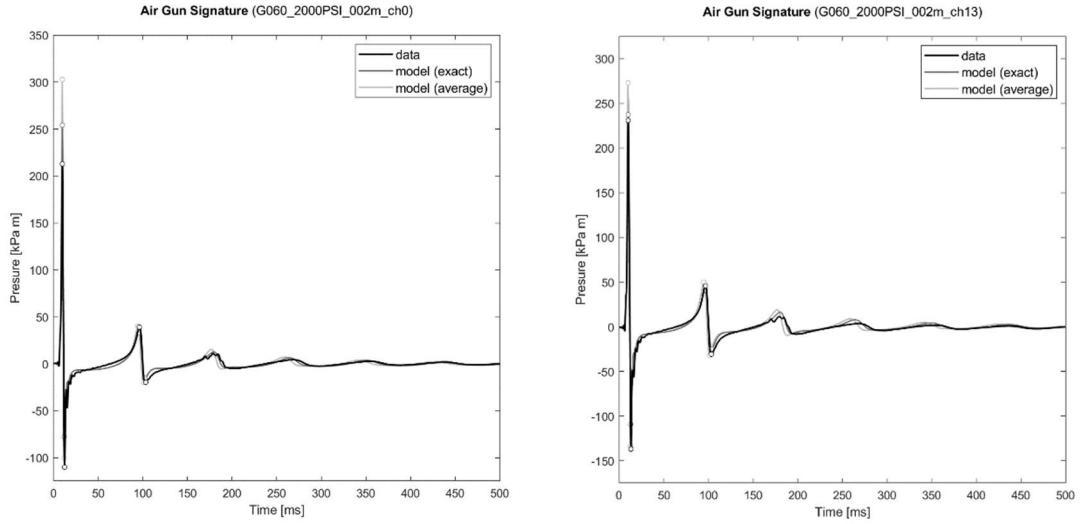

Figure 5.32 Far field acoustic signature at 1 m of G-Gun (60 in^3, 2000 psi, 2 m depth) for channels 0 (left) and 13 (right). Comparison between data (black), and simulation with exact (grey) and averaged (light grey) calibration parameters.

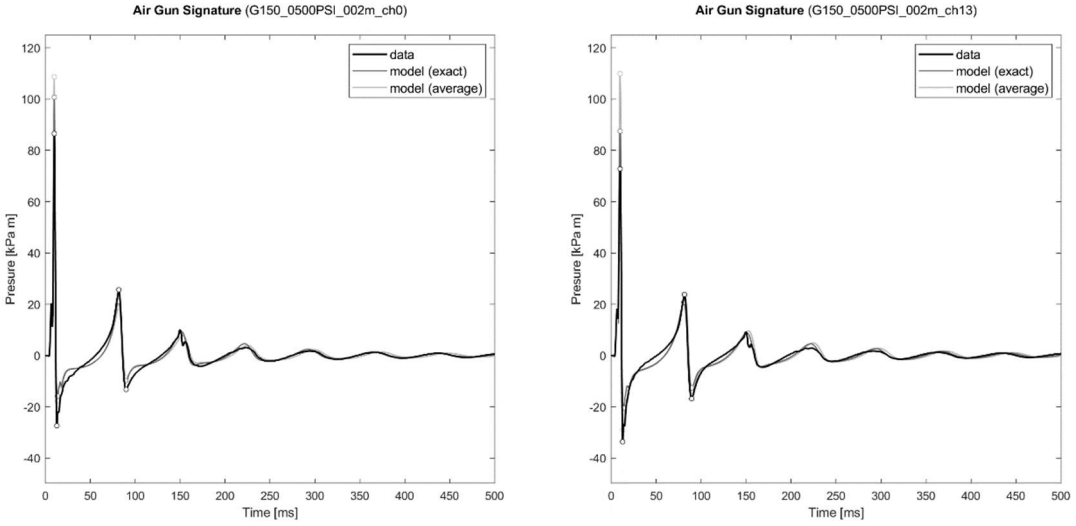

Figure 5.33 Far field acoustic signature at 1 m of G-Gun (150 in^3, 500 psi, 2 m depth) for channels 0 (left) and 13 (right). Comparison between data (black), and simulation with exact (grey) and averaged (light grey) calibration parameters.

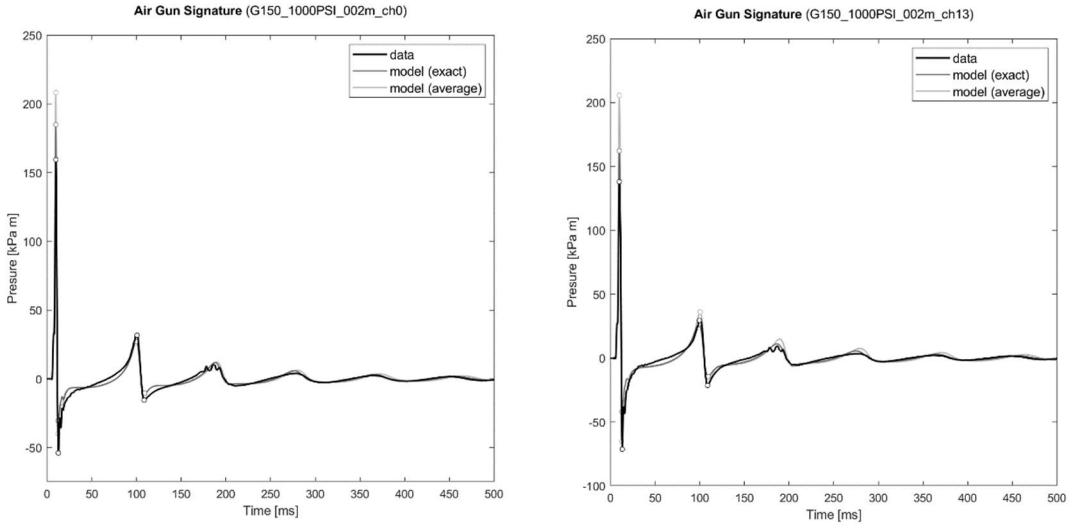

Figure 5.34 Far field acoustic signature at 1 m of G-Gun (150 in^3, 1000 psi, 2 m depth) for channels 0 (left) and 13 (right). Comparison between data (black), and simulation with exact (grey) and averaged (light grey) calibration parameters.

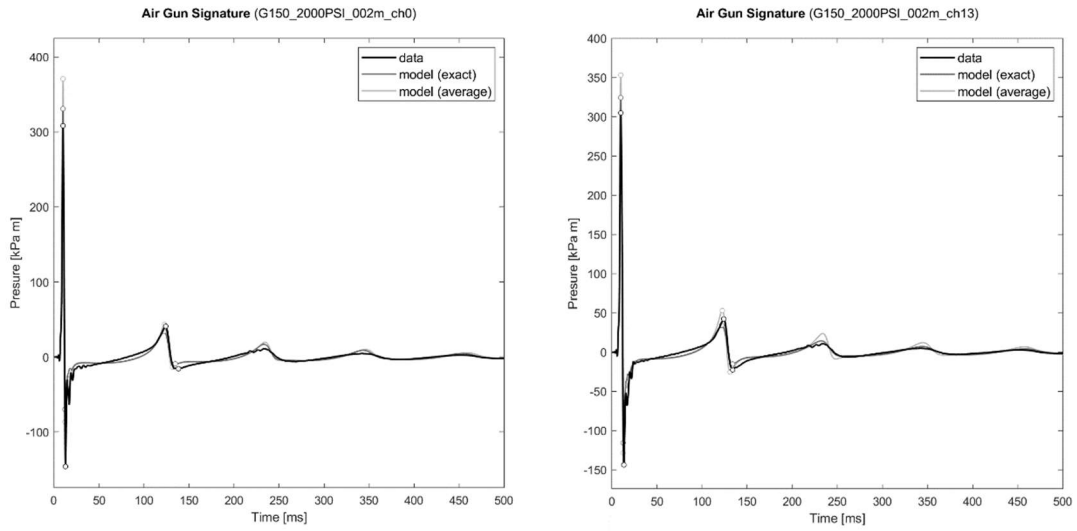

Figure 5.35 Far field acoustic signature at 1 m of G-Gun (150 in^3, 2000 psi, 2 m depth) for channels 0 (left) and 13 (right). Comparison between data (black), and simulation with exact (grey) and averaged (light grey) calibration parameters.

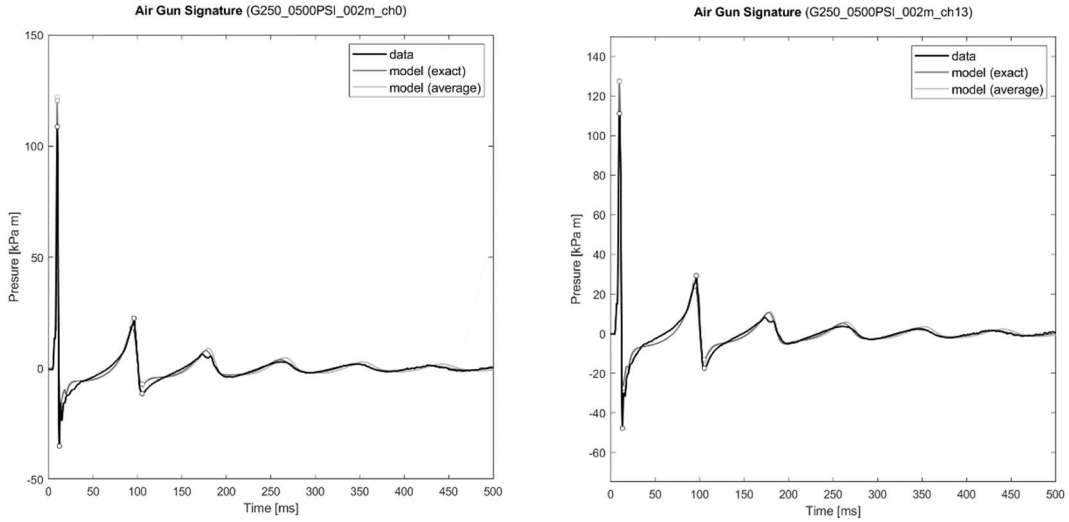

Figure 5.36 Far field acoustic signature at 1 m of G-Gun (250 in^3, 500 psi, 2 m depth) for channels 0 (left) and 13 (right). Comparison between data (black), and simulation with exact (grey) and averaged (light grey) calibration parameters.

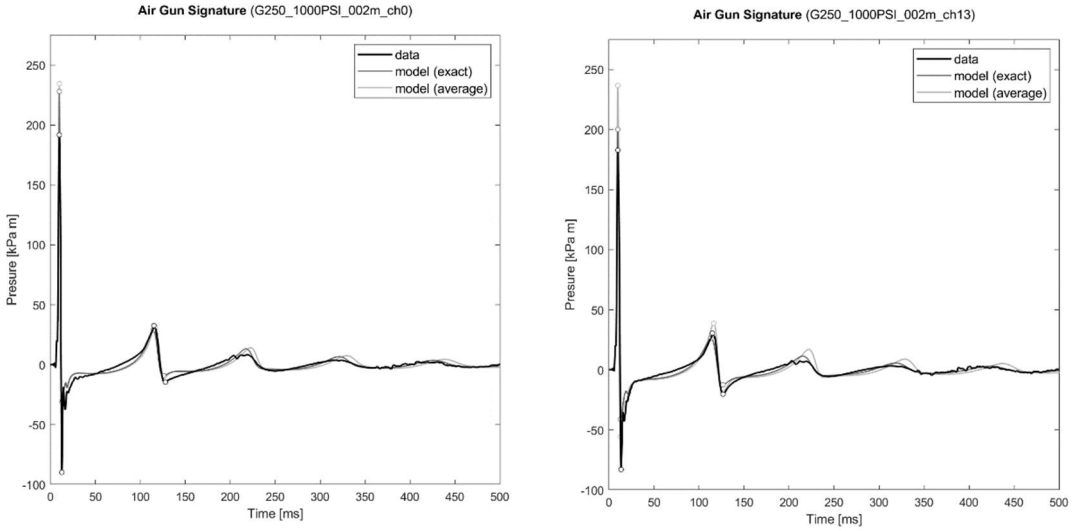

Figure 5.37 Far field acoustic signature at 1 m of G-Gun (250 in^3, 1000 psi, 2 m depth) for channels 0 (left) and 13 (right). Comparison between data (black), and simulation with exact (grey) and averaged (light grey) calibration parameters.

Chapter 6

Conclusions

In this thesis, we have developed a set of tools that can be used to characterise the sound field and the source. The tools include:

i. A complete sound field processing package (ABACUS) with all the functionalities that are required for generation of sound pressure level maps from raw audio and position data.

ii. A statistical Neyman-Pearson detector capable of identifying the target signal with a minimum false alarm rate even in conditions of low SNR and non-purely Gaussian noise ("peaky" noise).

iii. A calibrated air gun array model (RAGAS) that incorporates multiple physics and the ability to model clusters and GI guns.

New contributions have been made during the research process, aside from the end software tools. The Neyman-Pearson detector has been designed from classical statistical theory, not initially intended to work on high-dimensional vectors such as those from audio waveforms, sampled at frequencies of tens of kilohertz. It has been proved that a Likelihood Ratio Test (LRT) estimator-correlator can be successfully implemented for high-dimensional signals by using covariance estimation theory. An improved LRT energy detector has also been proposed for the detection of band-limited signals, which applies a bandpass filter to increase the SNR and a later correction to compensate for the white Gaussian signal and noise (WGS, WGN) assumptions.

Another contribution has been the implementation of a filtering technique that improves the accuracy of band levels at low frequencies for signals of short duration. The filtering method applies zero-padding to address an undesired energy "leakage" effect that occurs when the product of the filter's bandwidth Δf and the signal duration Δt is close to 10 (< 10). The zero padding compensates for the "leakage" effect, allowing for a lower value of $\Delta f \Delta t \approx 1$. Below that value the uncertainty principle dramatically reduces the accuracy of the filtered signal. This finding is particularly useful for the band spectral analysis of signals of short duration like those from air gun arrays or pile driving, as it allows the addition of lower frequency bands in the spectral representation. No mention about this effect could be found in the literature on signal processing.

A smaller software package has been developed for the optimisation of multivariate problems with simulated annealing. The package has been implemented as part of the calibration process for RAGAS.

The three software packages: ABACUS[38] (with NP algorithm), RAGAS[39] and the simulated annealing[40] code are available at GitHub under permissive MIT license.

[38] https://github.com/Guille340/ABACUS.git

[39] https://github.com/Guille340/RAGAS_AIRGUN_MODEL.git

[40] https://github.com/Guille340/SIMULATED_ANNEALING.git

References

Aarts, E. H. L., Korst, J. H. M., and Laarhoven, van, P. J. M. (1997). Simulated annealing. In E. H. L. Aarts, & J. K. Lenstra (Eds.), *Local search in combinatorial optimization* (pp. 91-120). Chichester: Wiley-Interscience.

Ainslie M. A., McColm J. G. (1998). "A simplified formula for viscous and chemical absorption in sea water", *J. Acoust. Soc. of Am*, 103(3), 1671-1672.

Ainslie, M. A., Halvorsen, M. B., Dekeling, R. P. A., Laws, R. M., Duncan, A. J., Frankel, A. S., Heaney, K. D., Küsel, E. T., MacGillivray, A. O., Prior, M. K., Sertlek, H. O., and Zeddies, D. G. (2016), "Verification of airgun sound field models for environmental impact assessment", Proc. Mtgs. Acoust., 27 (070018), pp. 13. doi: 10.1121/2.0000339

Ainsworth, S. E. (2001). The reflection and scattering of sound from the seabed. University of Bath.

Altes, R. A. (1980) "Detection, estimation and classification with spectrograms", J. Acoust. Soc. Am., 67, 1232-1246. doi: 10.1121/1.384165

American National Standards Institute [ANSI] (2004). *ANSI S1.11: Specification for octave, half-octave, and third-octave band filter sets*. ANSI. [online]. Available at: https://webstore.ansi.org/Standards/ASA/ ansiasas1112004r2009

Ansermet, J. P., and Brechet, S. D. (2019). *Principles of Thermodynamics*. (First Ed.) Cambridge University Press, pp. 529.

Bailey, T. C., Sapatinas, T., Powell, K. J., Krzanowski, P. W. (1998). "Signal detection in underwater sound using wavelets", *Journal of the American Statistical Association*, 93(441), 73-83. doi: 10.2307/2669604

Banarjee, O., El Ghaoui, L., d'Aspremont, A. (2007) "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data", *Journal of Machine Learning Research,* 9, 485-516.

Barthelemy, A. C., and Willett, P. K. (1993) "Prewhitening of colored noise fields for detection of threshold sources". In: Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, 1-3 Nov 1993, Pacific Grove, CA (USA), 243-247. doi: 10.1109/ACSSC.1993.342509

Ben-Ameur, W. (2004). "Computing the initial temperature of simulated annealing", *Computational Optimization and Applications*, 29, 369-385.

Besson, O., Vincent, F., and Gendre, X. (2020). "A Stein's approach to covariance matrix estimation using regularization of Cholesky factor and log-Cholesky metric", *Statistics and Probability Letters*, 167, p. 9. doi: 10.1016/j.spl.2020.108893

Bickel, P. J., and Levina, E. (2008b). "Covariance regularization by thresholding", *Ann. Statist.*, 36(6), 2577-2604. doi: 10.1214/08-AOS600

Bickel, P. J., and Levina, E. (2008a). "Regularized estimation of large covariance matrices", *Ann. Statist.*, 36(1), 199-227. doi: 10.1214/009053607000000758

Blondel, P., Jiménez, G., and Wyatt, R. (2016) "Seismic surveys in complex environments: Effects of environmental variability on sound propagation and mitigation practice". *Proc. Inst. Acoust.*, 38 (3), pp. 16-23. Acoustics and Environmental Variability, Fluctuations and Coherence, 12-13 December 2016, Cambridge (UK), pp. 16-23.

Boashash, B., and O'Shea, P. (1990) "A methodology for detection and classification of some underwater acoustic signals using time-frequency analysis techniques". IEEE Transactions On Acoustics, Speech and Signal Processing, 38 (11), 1829-1841. doi: 10.1109/29.103085

Borsani, J. F., Faulkner, R., Merchant, N.D. (2015). *Impacts of noise and use of propagation models to predict the recipient side of noise*. Report prepared under contract ENV.D.2/FRA/2012/0025 for the European Commission. Centre for Environment, Fisheries & Aquaculture Science, UK.

Brekhovskikh, L. M. (1980). *Waves in layered media*. (2nd Ed.) Academic Press

Brekhovskikh, L. M. and Lysanov, Y. P (1991). *Fundamentals of ocean acoustics*. 2nd Ed., Berlin: Springer-Verlag

British Standards Institution [BSI] (2014). *BS EN 61260-1:2014 Electroacoustics – Octave-band and fractional-octave-band filters, Part 1: Specifications*. ISO [online]. https://shop.bsigroup.com/products/electroacoustics-octave-band-and-fractional-octave-band-filters-specifications/standard

Brown, T. E., LeMay, H. E., Bursten, B. E., Murphy, C., Woodward, P., and Stoltzfus, M. E. (2017) *Chemistry: The central science*. 14th ed. Pearson, p. 1094.

Burg, J. P., Luenberger, D. G., and Wenger, D. L. (1982) "Estimation of structured covariance matrices", *Proceedings of the IEEE*, 70(9), 963-974. doi: 10.1109/PROC.1982.12427

Caldwell, J., Dragoset, W. (2000). "A brief review of seismic air-gun arrays", *The Leading Edge*, 19(8),

Cai, T. T., Zhang, C. H., and Zhou, H. H. (2010) "Optimal rates of convergences for covariance matrix estimation", *Ann. Statist.*, 38(4), 2118-2144

Cai, T., and Liu, W. (2011) "Adaptive thresholding for sparse covariance matrix estimation", *Journal of the American Statistical Association*, 106(494), 672-684. doi: 10.1198/jasa.2011.tm10560

Černý, V. (1985). "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm", *J. Optim. Theory Appl.*, 45(1), 41-51.

Chen, C.-T. and Millero, F.J. (1977) "Speed of sound in seawater at high pressures" *J. Acoust. Soc. Am.*, 62, 1129–35.

Chen, Y., Wiesel, A., Eldar, Y. C., and Hero, A. O. (2010) "Shrinkage algorithms for MMSE covariance estimation", *IEEE Transactions On Signal Processing*, 58(10), 5016-5029. doi: 10.1109/TSP.2010.2053029

Coates, R. (2006). *Foundation course: basic underwater acoustics. The SONAR course*. Seiche Ltd.

Collins, M. D. (1993). "A split-step Pade solution for the parabolic equation method", *J. Acoust. Soc. Am.*, 93, 1936-1942.

Collins, M. D. (1988). "The time-domain solution of the wide-angle parabolic equation including the effect of sediment dispersion", *J. Acoust. Soc. Am.*, 84(6), 2114-2125.

Cook, J. A., Gleeson, A. M. and Roberts, R.M. (1997). "A spark-generated bubble model with semi-empirical mass transport", J. Acoust. Soc. Am., 101 (4), 1908-1920

Coram, A., Gordon, J., Thompson, D., and Northridge, S. (2014). *Evaluating and assessing the relative effectiveness of non-lethal measures, including Acoustic Deterrent Devices, on marine mammals*. Scottish Government, pp. 145.

Crum, L. A., and Mao, Y. (1996). "Acoustically enhanced bubble growth at low frequencies and its implications for human diver and marine mammal safety", *J. Acoust. Soc. Am.*, 99(5), 2898-2907.

Dempster, A. P. (1972) "Covariance selection", *Biometrics*, 28(1), 157-75. doi: 10.2307/2528966

Dosso, S. E., Wilmut, M. J., Lapinski, A. S. (2001). "An adaptive-hybrid algorithm for geoacoustic inversion", *IEEE J Oceanic Eng*, 26(3), 324-336. doi: 10.1109/48.946507

Dragoset, B. (2000). "Introduction to air guns and air-gun arrays", *The Leading Edge*, 19(8), 892-897.

Duncan, A. J., and Gavrilov, A. N. (2019). “The CMST airgun array model – A simple approach to modelling the underwater sound output from seismic airgun arrays”, *IEEE Journal of Oceanic Engineering*, 44(3), 589-597.

Dunkin, J. W. (1965). “Computation of modal solutions in layered, elastic media at high frequencies”, *Bulletin of the Seismological Society of America*, 55 (2), 335-358.

Eaton, M. L., and Olkin, I. (1987) “Best equivariant estimators of a Cholesky decomposition”, *Ann. Statist.*, 15(4), 1639-1650. doi: 10.1214/aos/1176350615

Engineering ToolBox (2001). *The Engineering Toolbox* [online] Available at: https://www.engineeringtoolbox.com [Accessed: 30 Jun 2022].

Erbe, C. (2013). “International regulation of underwater noise”, *Acoustics Australia*, 41(1), 12-19.

Etter, P. C. (2013). *Underwater acoustic modeling and simulation*. 4th Ed., Boca Raton, Florida: CRC Press

European Parliament (2008), “Directive 2008/56/EC of the European Parliament and of the Council of 17 June 2008 establishing a framework for community action in the field of marine environmental policy (Marine Strategy Framework Directive)”, *Official Journal of the European Union* L 164, 19-40.

Fan, J, Li, R. (2001). “Variable selection via nonconcave penalized likelihood and its oracle properties”. *J Am Stat Assoc*, 96, 1348–1360. doi: 10.1198/016214501753382273

Fan, J., Fan, Y., and Lv, J. (2006). “High dimensional covariance matrix estimation using a factor model”, *Journal of Econometrics,* 147(1), 186-197. doi: 10.1016/j.jeconom.2008.09.017

Fan, J., Liao, Y., and Mincheva, M. (2013). “Large covariance estimation by thresholding principal orthogonal complements”, *J. R. Statist. Soc. B*, 75(4), 603-680. doi: 10.1111/rssb.12016

Findlay, C. R., Aleynik, D., Farcas, A., Merchant, N. D., Risch, D., and Wilson, B. (2021). “Auditory impairment from acoustic seal deterrents predicted for harbour porpoises in a marine protected area”, *J. Appl. Ecol.*, 0, 1-12. doi: 10.1111/1365-2664.13910

Fisher F. H., Simmons V. P. (1977). "Sound absorption in seawater", *J. Acoust. Soc. of Am*, 62, 558-564.

Fofonoff, N. P., and Millard, R. C. (1983). *Algorithms for computation of fundamental properties of seawater*. Paris, Unesco.

Francois R. E., Garrison G. R. (1982a). "Sound absorption based on ocean measurements: Part I:Pure water and magnesium sulfate contributions", *J. Acoust. Soc. of Am.*, 72(3), 896-907.

Francois R. E., Garrison G. R. (1982b). "Sound absorption based on ocean measurements: Part II:Boric acid contribution and equation for total absorption", *J. Acoust. Soc. of Am*, 72(6), 1879-1890.

Friedlander, B., and Porat, B. (1989) "Detection of transient signals by the Gabor representation". IEEE Transactions On Acoustics, Speech and Signal Processing, 32 (2), pp. 169-180. doi: 10.1109/29.21680

Friedman, J. H (1989) "Regularized discriminant analysis", *Journal of the American Statistical Association*, 84(406), 166-176. doi: 10.1080/01621459.1989.10478752

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007). "Pathwise coordinate optimization", *The Annals of Applied Statistics,* 1(2), 302-332. doi: 10.1214/07-AOAS131

Friedman, J., Hastie, T., and Tibshirani, R. (2008). "Sparse inverse covariance estimation with the graphical Lasso". *Biostatistics*, 9(3), 432–441. doi: 10.1093/biostatistics/kxm045

Friedman, J., Hastie, T., and Tibshirani, R. (2010). "Regularization paths for generalized linear models via coordinate descent", *Journal of Statistical Software*, 33(1), 1-22.

Frisk, G. V. (2012). "Noiseonomics: The relationship between ambient noise levels in the sea and global economic trends", *Sci. Rep.*, 2(437), pp. 4. doi: 10.1038/srep00437.

Fujikawa, S. and Akamatsu, T. (1980). "Effects of the non-equilibrium condensation of vapour on the pressure wave produced by the collapse of a bubble in a liquid", J. Fluid Mech., 97(3), 481-512

Furrer, R., and Bengtsson, T. (2007) "Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants", *Journal of Multivariate Analysis,* 98(2), 227-255. doi: 10.1016/j.jmva.2006.08.003

Graaf, K. L., Penesis, I. and Brandner, P. A. (2012). "Comparison of the Rayleigh-Plesset and Gilmore equations and additional aspects for the modelling of seismic airgun bubble dynamics", 18th Australasian Fluid Mechanics Conference, Launceston, Australia

Giles, B. F., and Johnston, R. C. (1973) "System approach to air-gun array design", *Geophysical Prospecting*, 21, 77-101.

Gilmore, F. R. (1952). *The growth or collapse of a spherical bubble in a viscous compressible liquid*, California Institute of Technology, report 26-4.

Geer, S. A. (2005). Least squares estimation. In: Everitt, B. S., and Howell, D. C. (eds.), *Encyclopedia of Statistics in Behavioural Science,* Vol. 2, Chichester: John Wiley & Sons, 1041-1045. doi: 10.1002/9781118445112.stat06609

Goertz, A., Wisløff, J. F., Drossaert, F., and Ali, J. (2013). "Environmental source modelling to mitigate impact on marine life", *First Break*, 31(11), 59-64. doi: 10.3997/1365-2397.31.11.71795

Goold, J. C., and Fish, P. J. (1998). "Broadband spectra of seismic survey air-gun emission, with reference to dolphin auditory thresholds", *J. Acoust. Soc. Am.*, 103(4), 2177-2184.

Gordon, J., Gillespie, D., Potter, J., Frantzis, A., Simmonds, M. P., Swift, R., and Thompson, D. (2004). "A review of the effect of seismic surveys on marine mammals", *Marine Technology Society Journal*, 37(4), 16-34.

Grotzinger, J., and Jordan, T. (2010). *Understanding the earth.* 6th Ed., W. H. Freeman, pp. 680.

Haff, L. R. (1980). "Empirical Bayes estimation of the multivariate normal covariance matrix", *Ann. Statist.*, 8(3), 586-597. doi: 10.1214/aos/1176345010

Hajek, B. (1988). "Cooling schedules for optimal annealing", *Mathematics of Operations Research*, 13(2), 311-329.

Hamilton, E. L. (1975). "Sound attenuation as a function of depth in the sea floor", *J. Acoust. Soc. Am.* 59 (3), 528-535

Hamilton, E. L. (1980) "Geoacoustic modelling of the sea floor", *J. Acoust. Soc. Am.* 68 (5), 1313-1340

Haskell, N. A. (1953). "The dispersion of surface waves on multilayered media", *Bulletin of the Seismological Society of America*, 43(1), 86-103.

Herring, C. (1941). *Theory of the Pulsations bf the Gas Bubble Produced by an Underwater Explosion*, OSRD Report No. 236

Herring, C. (1950) "Theory of the pulsations of the gas bubble produced by an underwater explosion", in Underwater Explosion Research, 2.

Hoerl, A. E., and Kennard, R. W. (1970). "Ridge regression: Biased estimation for nonorthogonal problems", *Technometrics*, 12(1), 55-67. doi: 10.2307/1267351

Hoffbeck, J. P., and Landgrebe, D. A. "Covariance matrix estimation and classification with limited training data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7), 763-767. doi: 10.1109/34.506799

Hood, J. (2009). Mitigation and monitoring: Passive Acoustic Monitoring (PAM) software development – Detection, classification and localisation capabilities. Final report no. 2008-004, v. 1.1, prepared by Akoostix Inc. for the Oil & Gas Producers (OGP), pp. 46.

Huang, J. Z., Liu, N., Pourahmadi, M., Liu, L. (2006). "Covariance matrix selection and estimation via penalised normal likelihood", *Biometrika*, 93(1), 85-98. doi: 10.1093/biomet/93.1.85

Huang, N. (2016). *Estimation of covariance, correlation and precision matrices for high-dimensional data*. Ph. D. thesis, London School of Economics and Political Science. Available at: http://etheses.lse.ac.uk/3371/

IBM (2021). *Covariance structures*. [online] https://www.ibm.com/docs/en/spss-statistics/26.0.0?topic=statistics-covariance-structures [Accessed 14 Apr 2022].

Ingber, L. (1993). "Simulated Annealing: Practice versus Theory", *Mathl. Comput. Modelling*, 18(11), 29-57.

International Organization for Standardization (2017). *Underwater Acoustics – Terminology*. ISO 18405:2017.

ITTC. (2011). "ITTC Recommended Procedures - Fresh water and seawater properties", 26th International Towing Tank Conference

James, W., and Stein, C. (1961). Estimation with quadratic loss. In Neyman, J. (eds.), *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, *Volume 1: Contributions to the Theory of Statistics*, Statistical Laboratory of the University of California, Berkeley, 20th Jun – 30th Jul, 361-379.

Jensen, F. B., Kuperman, W. A., Porter, M. B. and Schmidt, H. (2011). *Computational ocean acoustics*. 2nd Ed., New York: Springer Science + Business Media

Jiménez, G., Hedgeland, D., Cook, S., Banda, N., Johnston, P., and Oliver, E. (2019) *Sound level analysis for an offshore oil and gas drilling activity*. [Poster Presentation] 5th International Conference on the Effects on Noise on Aquatic Life, 7-12 July, Den Haag, The Netherlands. doi: 10.13140/RG.2.2.17494.32321

Jiménez, G., Blondel, P., Heath, B., and Wyatt, R. (2016). *Seismic surveys in complex environments – Analysing the variability of exclusion zones with field measurements and models*. [Poster Presentation] 4th International Conference on the Effects on Noise on Aquatic Life, 10-16 July, Dublin, Ireland. doi: 10.13140/RG.2.2.22946.91841

Jiménez-Arranz, G., Hedgeland, D., Cook, S., Banda, N., Johnson, P., and Oliver, E. (2019). "Acoustic Characterisation of a Mobile Offshore Drilling Unit". Proc. Mtgs. Acoust., 37(1), 11 p. Paper for *The Effects of Noise On Aquatic Life, Den Haage (Netherlands)*, 7th 12th July 2019. doi: 10.1121/2.0001193

Jiménez-Arranz, G., Banda, N., Cook, S., and Wyatt, R. (2020). *Review on existing data on underwater sounds produced by the oil and gas industry*. Technical report P783, prepared by Seiche Ltd for E&P Sound & Marine Life (JIP), pp. 182. [online] Available

at: https://gisserver.intertek.com/JIP/DMS/_ProjectReports/Cat1/JIP-Proj1.4.2_Review_on_Noise_from_Industrial_Sources.pdf (Accessed 21 May 2022).

Jiménez-Arranz, G., Banda, N., Cook, S., and Wyatt, R. (2020). *Review on existing data on underwater sounds from pile driving activities*. Technical report P784, prepared by Seiche Ltd for E&P Sound & Marine Life (JIP), pp. 82. [online] Available at: https://gisserver.intertek.com/JIP/DMS/ProjectReports/Cat1/JIP-Proj1.4.2_Review_on_Pile_Driving.pdf (Accessed 21 May 2022)

Johnson, D., Aragon, C., Mcgeoch, L., Schevon, C. (1989). "Optimization simulated annealing: an experimental evaluation; Part I, graph partitioning". *Oper. Res.*, 37(6), 865–892.

El Karoui, N. (2008) "Operator norm consistent estimation of large-dimensional sparse covariance matrices", *Ann. Statist.*, 36(6), 2717-2756. doi: 10.1214/07-AOS559

Kay, S. M. (1998). *Fundamentals of statistical signal processing. Vol 2: Detection theory.* 1st edn. New Jersey: Prentice Hall.

Keller, J.B., Kolodner, I.I. (1956). "Damping of underwater explosion bubble oscillations", *J Appl. Phys.* 27(10), 1152-1161

Keller, J.B., Miksis, M. (1980). "Bubble oscillations of large amplitude", *J. Acoust. Soc. Am* 68(2), 628-633

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. (1983). "Optimization by simulated annealing", *Science*, 220(4598), 671-680.

Kirkwood, J. G. and Bethe, H. A. (1942) *The pressure wave produced by an underwater explosion*. OSRD Report No. 588. Washington, DC: Office of Scientific Research and Development.

Kuttruff, H. (2009). *Room acoustics*. London & New York: Spon Press

Lampert, T., and O'Keefe, S. (2010) "A survey of spectrogram track detection algorithms". Applied Acoustics, 71, 87-100. doi: 10.1016/j.apacoust.2009.08.007

Landrø, M. (1992). "Modelling of GI gun signatures", Geophysical Prospecting 40(7), pp. 721-747

Landro, M., and Amundsen, L. (2010). "Marine seismic sources – Part I: Air-guns for non experts", *Geo ExPro*, 7(1), 32-34.

Laws, R. M., Hatton, L. and Haartsen, M. (1990). "Computer modelling of clustered airguns", First Break 8(9), pp. 331-337

Lebanon, G. (2010). “Bias, variance, and MSE of estimators”, teaching notes. [online] Available at: http://theanalysisofdata.com/notes/estimators1.pdf [Accessed 15 Apr 2022].

Ledesma, S., Aviña, G., Sánchez, R. (2008). Practical consideraction for simulated annealing implementation. In Ming Tan, C. (Ed.). *Simulated Annealing* (pp. 401-420). Vienna: I-Tech Education and Publishing.

Ledoit, O., and Wolf, M. (2003a) “Honey, I shrunk the sample covariance matrix”, UPF Economics and Business Working Paper No. 691. doi: 10.2139/ssrn.433840

Ledoit, O., and Wolf, M. (2003b) “Improved estimation of the covariance matrix of stock returns with an application to portfolio selection”, *Journal of Empirical Finance*, 10, 603-621. doi: 10.1016/S0927-5398(03)00007-0

Ledoit, O., and Wolf, M. (2004) “A well-conditioned estimator for large-dimensional covariance matrices”, *Journal of Multivariate Analysis*, 88(2), 365-411. doi: 10.1016/S0047-259X(03)00096-4

Ledoit, O., and Wolf, M. (2004) “A well-conditioned estimator for large-dimensional covariance matrices”, *Journal of Multivariate Analysis*, 88(2), 365-411. doi: 10.1016/S0047-259X(03)00096-4

Ledoit, O., and Wolf, M. (2012) “Nonlinear shrinkage estimation of large-dimensional covariance matrices”, *Ann. Statist.,* 40(2), 1024-1060. doi: 10.1214/12-AOS989

Ledoit , O., and Wolf, M. (2015) “Spectrum estimation: A unified framework for covariance matrix estimation and PCA in large dimensions”, *Journal of Multivariate Analysis,* 139, 360-384. doi: 10.1016/j.jmva.2015.04.006

Ledoit , O., and Wolf, M. (2021) “Shrinkage estimation of large covariance matrices: Keep it simple, statistician?”, *Journal of Multivariate Analysis*, 186. doi: 10.1016/j.jmva.2021.104796

Lee, C. Y., and Lee, D. (2014). “Determination of initial temperature in fast simulated annealing”, *Comput Optim Appl*, 58, 503-522.

Leroy, C.C., Robinson, S.P., and Goldsmith, M.J. (2008). “A new equation for the accurate calculation of sound speed in all oceans” *J. Acoust. Soc. Am.*, 124, 2774–82.

Lévesque, D., and Piché, L. (1992). “A robust transfer matrix formulation for the ultrasonic response of multi-layered absorbing media”, *J. Acoust. Soc. Am.*, 92(1), 452-467.

Levina, E., Rothman, A., and Zhu, J. (2008). “Sparse estimation of large covariance matrices via a nested lasso penalty”, *Ann. Statist.,* 2(1), 245-263. doi: 10.1214/07-AOAS139

Li, G., Cao, M., Chen, H. and Ni, C. (2010). "Modelling air gun signatures in marine seismic exploration considering multiple physical factors", Appl. Geophys. 7(2), 158-165

Lin, S. P., and Perlman, M. D. (1984*). A Monte Carlo comparison of four estimators of a covariance matrix*. Technical report no. 44, Department of Statistics, University of Washington, pp. 27.

Liu, Y., Ni, C. Z., Chen, H. L., Liu, J., Ye, Y., Niu, H., and Cao, M. Q. (2011). "Parameter optimization in air-gun type-dependent signature modelling", in 73rd EAGE Conference & Exhibition incorporating SPE EUROPEC, Vienna, Austria.

Lurton, X. (2010). *An introduction to underwater acoustics, principles and applications*. Berlin, Heilderberg: Springer-Verlag

MacGillivray, A. O. (2006). *An acoustic modelling study of seismic airgun noise in Queen Charlotte Basin*, published master's thesis, University of Victoria, Canada.

MacGillivrai, A. O. (2019). "An airgun array source model accounting for high-frequency sound emissions during firing – Solutions to the IAMW source test cases", *IEEE Journal of Oceanic Engineering*, 44(3), 582-588.

Mackenzie, K.V. (1981). "Nine-term equation for sound speed in the oceans". *J. Acoust. Soc. Am.*, 70, 807–12.

Madsen, P. T. (2005). "Marine mammals and noise: Problems with root mean square sound pressure levels for transients", *J. Acoust. Soc. Am.*, 117(6), 3952-3957.

Mattson, A. (2010). *JIP Svein Vaage broadband airgun study*. Final Report, PGS Geophysical, pp. 27.

Mattson, A., Parkes, G.E., and Hedgeland, D. (2012). "Svein Vaage broadband air gun study", Adv. Exp. Med. Biol., 730, 469-471. 10.1007/978-1-4419-7311-5_106

Medwin, H. (1975). "Speed of sound in water: A simple equation for realistic parameters" *J. Acoust. Soc. Am.*, 58, 1318–19.

Meinshausen, N., and Bühlmann (2006) "High-dimensional graphs and variable selection with the Lasso", *Ann. Statist.*, 34(3), 1436-1462. doi: 10.1214/009053606000000281

Melvin, H., and Murphy, L. (2002). "Time synchronization for VoIP Quality of Service", *IEEE Internet Computing*, 6(3), 57-63. doi: 10.1109/MIC.2002.1003132

Merchant, N. D., and Robinson, S. P. (2020). "Abatement of underwater noise pollution from pile-driving and explosions in UK waters", report of the UKAN workshop, 12 November 2019, Royal Society, London, pp. 31. doi: 10.6084/m9.figshare.11815449

Merchant, N. D., Fristrup, K. M., Johnson, M. P., Tyack, P. L., Witt, M. J., Blondel, P., and Parks, S. E. (2015). "Measuring acoustic habitats". *Methods Ecol. Evol.*, 6(3), 257-265. doi: 10.1111/2041-210X.12330

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). "Equation of state calculations by fast computing machines", *Journal of Chemical Physics*, 21(6), 1087-1092.

Millero, F.J. and Li, X. (1994) Comments on "On equations for the speed of sound in seawater" *J. Acoust. Soc. Am.* 95(5), 2757-2759

Millero, F. J., & Poisson, A. (1981). "International one-atmosphere equation of state of seawater" *Deep Sea Research Part A, Oceanographic Research Papers*, *28(6),* 625-629.

Mitnik, L. M. and Dubina, V. A. (2005). "Synoptic-scale, mesoscale and fine-scale oceanic features in the Japan/East Sea: study with ERS-1/2 SAR and Envisat ASAR". IGARSS '05. Proceedings of the IEEE International, 7, Seoul, Korea, 4788-4791

National Marine Fisheries Services (2018). *2018 Revision to: Technical guidance for assessing the effects of anthropogenic sound on marine mammal hearing (version 2.0) – Underwater threshold for onset permanent and temporary threshold shifts*. NOAA Technical Memorandum NMFS-OPR-59.

National Research Council (2005). *Marine mammal populations and ocean noise: Determining when noise causes biologically significant effects*. Washington DC, USA: National Academies Press, pp. 142.

Nayar, K. G., Panchanathan, D., McKinley, G. H. and Lienhard, J. H. (2014). "Surface tension of seawater", J. Phys. Chem. Ref. Data, 43(4)

Oliveira, P. M., and Barroso, V. (2000) "Data driven underwater transient detection based on time-frequency distributions", OCEANS 2000 MTS/IEEE Conference and Exhibition, Providence, RI (USA), 11-14 September 2000, pp. 1059-1063. doi: 10.1109/OCEANS.2000.881741

OGP (2011). "An overview of marine seismic operations", OGP Publications, 448, pp. 46

Popper, A. N., Hawkins, A. D., Fay, R. R., Mann, D. A., Bartol, S., Carlson, T. J., Coombs, S., Ellison, W. T., Gentry, R. L., Halvorsen, M. B., Løkkeborg, S., Rogers, P. H., Southall, B. L., Zeddies, D. G., and Tavolga, W. N. (2014). Sound Exposure Guidelines for Fishes and Sea Turtles. Technical report ASA S3/SC1.4 TR-2014. ASA Press/Springer.

Porat, B., and Friedlander, B. (1986). "Adaptive detection of transient signals", IEEE Transactions On Acoustics, Speech, and Signal Processing, 34(6), 1410-1418. doi: 10.1109/TASSP.1986.1164977

Pourahmadi, M. (2013). *High-dimensional covariance estimation*. 1st edn. New Jersey: John Wiley & Sons.

Proakis, J. G, and Manolakis, D. G. (2007). *Digital signal processing – Principles, algorithms, and applications*. 4th ed., New Jersey: Prentice Hall. pp. 1084.

Prosperetti, A., (1984) "Bubble phenomena in sound fields," Ultrasonics 22(2), 69–82

Prosperetti, A., and Lezzi, A. (1985). "Bubble dynamics in a compressible liquid – Part 1. First-order theory", *J. Fluid Mech.*, 168, 457-478. doi: 10.1017/S0022112086000460

Racca, R. G., and Scrimger, J. A. (1986). *Underwater acoustic source characterization of air and water guns*. Technical Report No. 06SB 97708-5-7055, Defence Research Establishment Pacific.

Ridgway, S. H., and Howard, R. (1982). "Dolphins and the bends", *Science*, 216(651), 650-651.

Robinson, S.P., Lepper, P. A. and Hazelwood, R.A. (2014). *Good Practice Guide No. 133 – Underwater Noise Measurement*. A report from the National Physics Laboratory (NPL) National for Marine Scotland, The Crown Estate, pp. 97.

Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. (2008). "Sparse permutation invariant covariance estimation", *Electronic Journal of Statistics*, 2, 494-515. doi: 10.1214/08-EJS176

Rothman, A. J., Levina, E., and Zhu, J. (2009) "Generalized thresholding of large covariance matrices", *Journal of the American Statistical Association*, 104(485), 177-186. doi: 10.1198/jasa.2009.0101

Rothman, A. J. (2012) "Positive definite estimators of large covariance matrices", *Biometrika,* 99(3), 733-740. doi: 10.1093/biomet/ass025

Safar, M. H. (1976). "Efficient design of air-gun arrays", *Geophysical Prospecting*, 24, 773-787.

Schäfer, J., and Strimmer, K. (2005). "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics", *Statistical Applications in Genetics and Molecular Biology*, 4(1), article 32. doi: 10.2202/1544-6115.1175

Schulze-Gattermann, R. (1972) "Physical aspects of the airpulser as a seismic energy source", *Geophysical Prospecting*, 20, 155–192

Seibold, E., and Berger, W. (2017). *The sea floor – An introduction to marine geology*. 4th Ed., Switzerland: Springer International Publishing.

Selliah, J.B. (1964). Estimation and Testing Problems in a Wishart Distribution. Technical report no. 10, Department of Statistics, Stanford University.

Sertlek, O., Ainslie, M. A. (2015). "Airgun source model (AGORA): Its application for seismic surveys sound maps in the Dutch North Sea", Conference Proceedings of the Underwater Acoustics Conference (UAC) 2015.

Sinclair, J. E., and Bhattacharya, G. (1980). "Interaction effects in marine seismic source arrays", *Geophysical Prospecting*, 28, 323-332.

Skiena, S. S. (2008). *The Algorithm Design Manual*. (2nd Ed.). Springer-Verlag, London.

Southall B, et al. (2007). "Marine mammal noise exposure criteria: initial scientific recommendations", Aquatic Mammals, 33(4), 411-521

Southall, B. L., Finneran, J. J., Reichmuth, C., Nachtigall, P. E., Ketten, D. R., Bowles, A. E., Ellison, W. T., Nowacek, D. P., and Tyack, P. L. (2019). "Marine mammal noise exposure criteria: Updated scientific recommendations for residual hearing effects". Aquatic Mammals, 45(2), 125-232. doi: 10.1578/AM.45.2.2019.125

Stein, C. (1956). Inadmissibility of the usual estimator of the mean of a multivariate normal distribution. In: Neyman J, editor. *Proceedings of the Third Berkeley Symposium on Mathematical and Statistical Probability. Volume 1: Contribution to the Theory of Statistics*. University of California Press, 197–206. doi: 10.1525/9780520313880-018

Stein, C. (1975). "Estimation of a covariance matrix". *Rietz Lecture, 39th Annual Meeting IMS*. Atlanta, GA.

Stergiopoulos, S. (2001). *Advanced signal processing handbook: Theory and implementation for radar, sonar, and medical imaging real-time systems*. Florida (USA): CRC Press.

Stone, C. J. (2003). *The effect of seismic activity on marine mammals in UK waters, 1998-2000*. JNCC Report No. 323, pp. 78.

Tappert, F.D. (1977). The parabolic approximation method. In: Keller, J.B., Papadakis, J.S. (eds) Wave Propagation and Underwater Acoustics. Lecture Notes in Physics, vol 70. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-08527-0_5

Theiler, J. (2012) "The incredible shrinking covariance estimator". In: SPIE Defense Security and Sensing Symposium, 23rd April 2012, Baltimore, MD (USA), pp. 12.

Thomas, R. M. (1979). "Condensation of steam on water in turbulent motion", Int. J. of Multiphase Flow, 5(1), pp.1-15

Thomson, W. T. (1950). "Transmission of elastic waves through a stratified solid medium", *American Institute of Physics*, 21, 89-93.

Tibshirani, R. (1996). "Regression shrinkage and selection via the Lasso", *J. R. Statist. Soc. B*, 58(1), 267-288. doi: 10.1111/j.2517-6161.1996.tb02080.x

Tsilingiris, P. T. (2008). "Thermophysical and transport properties of humid air at temperature range between 0 and 100 °C", *Energy Conversion and Management*, 49, 1098-1110.

Turin, G. L. (1960). "An introduction to matched filters", *IEEE Transactions On Information Theory*, 6(3), 311-329. doi: 10.1109/TIT.1960.1057571

Tuteur, F. B. (1967) "On the detection of transiting broadband targets in noise of uncertain level", IEEE Transactions in On Communication Technology, 15(1), 61-69. doi: 10.1109/TCOM.1967.1089538

Urick, R.J. (1983) *Principles of Underwater Sound*. 3rd Ed., McGraw-Hill, New York, pp. 384.

Urkowitz, H. (1967). "Energy detection of unknown deterministic signals", *Proceedings of the IEEE*, 4, 523-531. doi: 10.1109/PROC.1967.5573

Verfuss, U.K., Aniceto, A. S., Biuw, M., Fielding, S., Gillespie, D., Harris, D., Jiménez, G., Johnston, P., Plunkett, R., Sivertsen, A., Solbø, A., Storvold, R., and Wyatt, R. (2016) *Literature review: Understanding the current state of autonomous technologies to improve/expand observation and detection of marine species*. [online] Report prepared by SMRU Consulting for The Joint Industry Programme (JIP) on E&P Sound and Marine Life, Rep. SMRUC-2015-015, JIP ref. 5.4. Available at: https://gisserver.intertek.com/JIP/DMS/ProjectReports/Cat5/JIP-Proj5.4_Final%20LiteratureReview.pdf [Accessed 13 Feb 2021]

Verfuss, U.K., Sinclair, R.R. & Sparling, C.E. (2019a). A review of noise abatement systems for offshore wind farm construction noise, and the potential for their application in Scottish waters. Scottish Natural Heritage Research Report No. 1070.

Verfuss, U.K., Aniceto, A.S., Harris, D.V, Gillespie, D., Fielding, S., Jiménez, G., Johnston, P., Sinclair, R., Sivertsen, A., Solbø, Storvold, R., Biuw, M., and Wyatt, R. (2019b). "A review of unmanned vehicles for the detection and monitoring of marine fauna". Mar. Pollut. Bull. 140, p. 17-29. doi: 10.1016/j.marpolbul.2019.01.009

Viswanathan, M (2020). Wireless communication systems in Matlab. 2nd ed. Viswanathan, M. (ed.).

Wang, L., Heaney, K., Pangerc, T., Theobald, P., Robinson, S., Ainslie, M. (2014). Review of underwater acoustic propagation models. NPL report AC 12, pp. 30.

Warton, D. I. (2008) "Penalized normal likelihood and ridge regularization of correlation and covariance matrices", Journal of the American Statistical Association, 103(481), 340-349. doi: 10.1198/016214508000000021

Weir, C. R., and Dolman, S. J. (2007). "Comparative review of the regional marine mammal mitigation strategies implemented during industrial seismic surveys, and

guidance towards a worldwide standard", Journal of International Wildlife Law and Policy, 10, 1-27.

Weston, D. E. (1968). "Sound Focusing and beaming in the interference field due to several shallow-water modes", J. Acoust. Soc. Am., 44, 1706-1712

Weston, D. E. (1980). "Acoustic flux formulas for range-dependent ocean ducts", J. Acoust. Soc. Am., 68 (1), 269-281.

Witten, D., and Tibshirani, R. (2009) "Covariance-regularized regression and classification for high-dimensional problems", *J. R. Stat. Soc. Series. B Stat. Methodol.*, 71(3), 615-636. doi: 10.1111/j.1467-9868.2009.00699.x

Won, J. H., and Kim, S. J. (2006) "Maximum likelihood covariance estimation with a condition number constraint". In: 40th Asilomar Conference on Signals, Systems and Computers, 29th Oct to 1st Nov 2006, Pacific Grove, CA (USA), pp. 5. doi: 10.1109/ACSSC.2006.354997

Wu, W. B., and Pourahmadi, M. (2003). "Nonparametric estimation of large covariance matrices of longitudinal data", *Biometrika*, 90(4), 831-844. doi: 10.1093/biomet/90.4.831

Xian, Y., Nolte, L., Tantum, S., Liao, X., and Zhang, Y. (2014). "On marine mammal acoustic detection performance bounds", *J. Acoust. Soc. Am.*, 135(4). doi: 10.48550/arXiv.1510.05520

Xinchao, Z. (2011). "Simulated annealing algorithm with adaptive neighborhood", *Applied Soft Computing*, 11, 1827-1836.

Xue, L., Ma, S., and Zou, H. (2012) "Positive-definite l1-penalized estimation of large covariance matrices", *Journal of the American Statistical Association,* 107 (500), 1480-1491. doi: 10.1080/01621459.2012.725386

Yuan, M., and Lin, Y. (2007) "Model selection and estimation in the Gaussian graphical model", *Biometrika*, 94(1), 19-35. doi: 10.1093/biomet/asm018

Zou, H., and Hastie, T. (2005) "Regularization and variable selection via the elastic net", *J. R. Statist. Soc. B*, 67(2), 301-320. doi: 10.1111/j.1467-9868.2005.00503.x

Zou, H. (2006) "The Adaptive Lasso and its oracle properties", *Journal of the American Statistical Association*, 101 (476), 1418-1429, DOI: 10.1198/016214506000000735

Ziolkowski, A. (1970). "A method for calculating the output pressure waveform from an air gun", Geophys. J. R. Astr. Soc. 21(2), 137-161.

Ziolkowski,, A., Parkes, G., Hatton, L. and Haugland, T. (1982). "The signature of an air gun array: computation from near-field measurements including interactions", Geophysics 47(10), 1413-1421

Appendix A

Databases

This appendix presents an in-depth summary of the content and usage of the five databases used in ABACUS. The databases contain intermediate data used by the software for communicating between its different modules, as well as input and output information that works as interface with the user. The appendix is organised into five sections, each dedicated to a specific database (Configuration, Audio, Detection, Navigation, and Acoustic). The content is expected to serve as a reference for the reader to consult, as well as a way of providing a more complete view of the work without disrupting the main message within the core chapters.

A.1 Configuration Database

The Configuration Database folder contains global, module-specific, and setting-specific configuration files. These are stored in folder 'configdb' under the `"block"` directory in `root.json`. Details about all configuration scripts are given below.

A.1.1 Global

Global configuration files affect all modules in ABACUS. These files simplify the configuration process by removing the need to individually define settings that are common to all modules, which could lead to mistakes. There are three global configuration files: `audioPaths.json`, `channelToReceiver.json` and `resampleRateToSource.json`.

`audioPaths.json` contains a list of relative paths and folders pointing at specific audio files. The paths and folders are relative to the `"audio"` directory in `root.json`. The list is used by the Audio Import, Audio Detect, Audio Process, and Navigation Process modules to determine what audio files, audio databases, or acoustic databases need to be processed. Listed paths and folders can be ignored by adding at the start of the string one supported "commenting" character (!, #, %). Figure A.1 shows an example.

```
[
"RAW\\20140406_183314_4589586_7200000.raw2int16",
"RAW\\20140406_183544_4589586_7200000.raw2int16",
"!RAW\\20140406_183814_4589586_7200000.raw2int16",
]
```

Figure A.1 Example of the content in `audioPaths.json`.

ABACUS uses unique source and receiver names to link the results from all its modules. `channelToReceiver.json` and `resampleRateToSource.json` define the relationship between channel and receiver, and new sampling rate and source. Figure A.2 and Figure A.3 show examples for these two files.

```
{
"channel": [1,2],
"receiverName": ["Buoy1_H1","Buoy1_H2"]
}
```

Figure A.2 Example of the content in `channelToReceiver.json`.

```
{
"resampleRate": [ 24000, 96000],
"sourceName":   ["Airgun", "Sparker"]
}
```

Figure A.3 Example of the content in `resampleRateToSource.json`.

Defining these channel-receiver and resampling-source relationships requires some careful planning, as changing these settings will force the software to modify elements in the Acoustic Databases (for details see Sub-Section 3.4.1).

A.1.2 Audio Import

There are two module-specific configuration files in Audio Import. The templates are named `audioImportConfig_<FMT>.json`, where `<FMT>` is the supported audio format (`RAW`, `WAV`). The user must create an `audioImportConfig*.json` file for each `channel-resampleRate` combination to be processed. Figure A.4 and Figure A.5 show examples for the two configuration files.

Both files include the audio format (`"audioFormat"`), the channel to read from (`"channel"`), and the new sampling rate (`"resampleRate"`). The template for the `RAW` format also contains the audio settings needed to read the RAW data correctly. These are the sampling rate in Hertz (`"sampleRate"`), the bit resolution (`"bitDepth"`), the number of channels (`"numChannels"`), and the byte order (`"endianness"`). In the `WAV` format, the audio settings are extracted directly from its file's header.

```
{
"audioFormat": "RAW",
"channel": 1,
"resampleRate": 48000,
"sampleRate": 48000,
"bitDepth": 16,
"numChannels": 2,
"endianness": "l"
}
```

Figure A.4 Example of RAW template for the configuration file `audioImportConfig*.json`.

```
{
"audioFormat": "WAV",
"channel": 1,
"resampleRate": 96000
}
```

Figure A.5 Example of WAV template for the configuration file `audioImportConfig*.json`.

A.1.3 Audio Detect

There are five module-specific configuration files in Audio Detect, one per detection algorithm. The templates are named `audioDetectConfig_<DET>.json`, where `<DET>` is the type of detector (`MovingAverage`, `Slice`, `NeymanPearson`, `ConstantRate` and `Mirror`). The user must create one `audioDetectConfig*.json` for each `receiverName-sourceName` combination to be processed. Examples of the configuration files for the five detectors are shown in Figure A.6 to Figure A.10.

The first four templates have the following fields in common: the name of the source ("sourceName"), the name of the receiver ("receiverName"), the type of detector ("detector") and its specific parameters ("DetectParameters"). The Mirror option is not a detector "per se", but a solution to replicate available detection data on other channels. As such, it does not include the "detector" and "DetectParameters" fields. The fields in "DetectParameters" are specific for each type of detector. These fields are described below for the four detector templates.

The MovingAverage detector includes the duration of the detection window in seconds ("windowDuration"), the time difference in seconds between the start of the processing window and the energy peak of the event ("windowOffset"), the minimum ratio between the RMS of the processed segment and the RMS of the background noise ("threshold"), and the cutoff frequencies of the bandpass filter in Hertz ("cutoffFreqs"). The "detector" field is always set as MovingAverage.

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"detector": "MovingAverage",
"DetectParameters": {
      "windowDuration": 1000,
      "windowOffset": 500,
      "threshold": [],
      "cutoffFreqs": [20,10000]}
}
```

Figure A.6 Example of configuration file audioDetectConfig_MovingAverage.json.

The Slice detector includes the duration of the detection window in seconds ("windowDuration"). The algorithm only requires the window duration to identify the segments to be processed, which are consecutive and of equal duration. The "detector" field is always set as Slice.

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"detector": "Slice",
"DetectParameters": {
      "windowDuration": 1000}
}
```

Figure A.7 Example of configuration file audioDetectConfig_Slice.json.

The ConstantRate detector includes the duration of the detection window in seconds ("windowDuration"), the time difference in seconds between the start of the processing window and the energy peak ("windowOffset"), and the name of the 'Pulse' table used to determine the location of the periodically-occurring target events ("fileName"). The "detector" field is always set as ConstantRate.

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"detector": "ConstantRate",
"DetectParameters": {
       "windowDuration": 5000,
       "windowOffset": 2500,
       "fileName": "pulseTable.csv"}
}
```

Figure A.8 Example of configuration file `audioDetectConfig_ConstantRate.json`.

The `NeymanPearson` detector includes the detection algorithm (`"detectorType"`), the duration of the detection window in seconds (`"kernelDuration"`), the minimum duration of the target event in seconds (`"windowDuration"`), the time difference in seconds between the start of the processing window and the energy peak of the event (`"windowOffset"`), the target probability of false alarm for which to obtain the detection threshold γ' (`"rtpFalseAlarm"`), the sensitivity factor used to artificially increase the threshold γ' to help minimise the number of false alarms produced by high-level, non-target signals (`"detectorSensitivity"`), the lower SNR limit for any segment to be classified as "detection" (`"minSnrLevel"`), the cutoff frequencies of the bandpass detection filter used to enhance the SNR and improve the detection performance (`"cutoffFreqs"`), the directory containing the training data for the target signal and noise (`"trainFolder"`), the type of covariance estimator (`"estimator"`), and the provisional new sampling rate used to improve detection performance for low-frequency target signals (`"resampleRate"`). The `"detector"` field is always set as `NeymanPearson`.

```
{
"receiverName": "Buoy1_H2",
"sourceName": "Airgun",
"detector": "NeymanPearson",
"DetectParameters": {
       "detectorType": "ecc",
       "kernelDuration": 0.5,
       "windowDuration": 1,
       "windowOffset": 0.5,
       "rtpFalseAlarm": 0.001,
       "detectorSensitivity": 1,
       "minSnrLevel": -6,
       "cutoffFreqs": [20,8000],
       "trainFolder": "C:\\AIRGUN PULSES",
       "estimator": "sample",
       "resampleRate": 16000
       }
}
```

Figure A.9 Example of configuration file `audioDetectConfig_NeymanPearson.json`.

The `Mirror` solution includes the name of the receiver from which to copy the detection data (`"mirrorReceiver"`). This is a general field, not part of `"DetectParameters"`.

```
{
"receiverName": "Buoy1_H2",
"sourceName": "Airgun",
"mirrorReceiver": "Buoy1_H1"
}
```

Figure A.10 Example of configuration file `audioDetectConfig_Mirror.json`.

A.1.4 Audio Process

There is only one module-specific configuration file in Audio Process. The template is named `audioProcessConfig.json`. The user must create one `audioProcessConfig.json` file for each `receiverName-sourceName` combination to be processed. Figure A.11 shows an example of this configuration template.

The template includes the name of the receiver (`"receiverName"`), the name of the source (`"sourceName"`), the frequency limits in Hertz for the broadband and band metrics (`"freqLimits"`), the number of bands per octave to determine the bandwidth and central frequencies of the bandpass filters in the bank (`"bandsPerOctave"`), the energy ratio used for calculating the time limits of a processed segment based on its cumulative energy curve (`"cumEnergyRatio"`), the format of the timestamp in the audio files used to determine the absolute time of their first sample (`"audioTimeFormat"`), the time difference in seconds between the PC and UTC times (`"timeOffset"`), and special words that can be later assigned to sections of the audio data to help classify and interpret their content (`"tags"`).

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"freqLimits": [20, 20000],
"bandsPerOctave": 3,
"cumEnergyRatio": 0.9,
"audioTimeFormat": "yyyymmdd_HHMMSS_FFF*",
"timeOffset": [],
"tags": ["full","rampup"]
}
```

Figure A.11 Example of configuration file `audioProcessConfig.json`.

A.1.5 Receiver Import

There are two module-specific configuration files in Receiver Import. The templates are named `receiverImportConfig_<CAT>.json`, where `<CAT>` is the receiver category (`Fixed`, `Towed`). The user must create one `receiverImportConfig*.json` file for each receiver to be processed. Examples of the configuration files for the two receiver categories are shown in Figure A.12 and Figure A.13.

The two templates have two fields in common: the category and the name of the receiver ("receiverCategory", "receiverName"). The rest of the fields are described below.

For the fixed category, the template includes the receiver's geographic position in degrees ("latitude", "longitude") and the depth below water surface in metres ("depth"). The "receiverCategory" field is always set as fixed.

```
{
"receiverCategory": "fixed",
"receiverName": "Buoy1_H1",
"latitude": 44.2032,
"longitude": -4.5368,
"depth": -2
}
```

Figure A.12 Example of configuration file receiverImportConfig_Fixed.json.

For the towed category, the template includes the (x,y) horizontal offset of the receiver in metres relative to the positioning device ("receiverOffset"), the type of connection between the receiver and the moving platform ("receiverOffsetMode"), the paths of the positioning files relative to the "position" directory in root.json ("positionPaths"), the positioning protocol ("positionFormat"), the platform used to record the position data ("positionPlatform"), the index of the vessel in a P190 file ("vesselId"), the Maritime Mobile Service Identity (MMSI) number of the vessel found in AIS sentences ("mmsi"), and the receiver depth in metres below water surface ("depth"). The "receiverCategory" field is always set as towed.

```
{
"receiverCategory": "towed",
"receiverName": "Buoy1_H1",
"receiverOffset": [0,0],
"receiverOffsetMode": "soft",
"positionPaths": "GPS (SeicheSsv).gpstext",
"positionFormat": "GPS",
"positionPlatform": "SeicheSsv",
"vesselId": [],
"mmsi": [],
"depth": -2
}
```

Figure A.13 Example of configuration file receiverImportConfig_Towed.json.

A.1.6 Source Import

There are four module-specific configuration files in Source Import. The templates are named sourceImportConfig_<CAT>.json, where <CAT> is the source category (Fixed, Towed, Vessel, Fleet). The user must create one sourceImportCon-

fig*.json file for each source to be processed. Examples of the configuration files for the four source categories are shown in Figure A.14 to Figure A.17.

The four templates have two fields in common: the category and the name of the source ("sourceCategory", "sourceName"). The rest of the fields are described below.

For the fixed category, the template includes the source's geographic position in degrees ("latitude", "longitude") and the depth below water surface in metres ("depth"). The "sourceCategory" field is always set as fixed.

```
{
"sourceCategory": "fixed",
"sourceName": "Airgun",
"latitude": 44.1890,
"longitude": -4.5045,
"depth": -2
}
```

Figure A.14 Example of configuration file sourceImportConfig_Fixed.json.

For the towed category, the template includes the (x,y) horizontal offset of the source in metres relative to the positioning device ("sourceOffset"), the type of connection between the source and the moving platform ("sourceOffsetMode"), the paths of the positioning files relative to the "position" directory in root.json ("positionPaths"), the positioning protocol ("positionFormat"), the platform used to record the position data ("positionPlatform"), the index of the vessel ("vesselId") and the seismic source ("sourceId") in a P190 file, the MMSI number of the vessel found in AIS sentences ("mmsi"), and the source depth in metres below water surface ("depth"). The "sourceCategory" field is always set as towed.

```
{
"sourceCategory": "towed",
"sourceName": "Airgun",
"sourceOffset": [0,0],
"sourceOffsetMode": "hard",
"positionPaths": "P190 (Seismic).p190",
"positionFormat": "P190",
"positionPlatform": "Seismic",
"vesselId": 1,
"sourceId": 1,
"mmsi": [],
"depth": -2
}
```

Figure A.15 Example of configuration file sourceImportConfig_Towed.json.

For the vessel category, the template includes the paths of the positioning files relative to the "position" directory ("positionPaths"), the positioning protocol ("positionFormat"), the platform used to record the position data ("positionPlatform"),

the vessel index in a P190 file ("vesselId"), the vessel MMSI number ("mmsi"), and the vessel name ("vesselName"), length in metres ("vesselLength"), width in metres ("vesselBeam"), draft in meters ("vesselDraft"), and size in tonnes ("vesselGrossTonnage"). The "sourceCategory" field is always set as vessel.

```
{
"sourceCategory": "vessel",
"sourceName": "Isabella Rose",
"positionPaths": "AIS (PAMGuard).csv",
"positionFormat": "AIS",
"positionPlatform": "PamGuard",
"vesselId": [],
"mmsi": 367614560,
"vesselName": "Isabella Rose",
"vesselLength": 45,
"vesselBeam": 11,
"vesselDraft": [],
"vesselGrossTonnage": []
}
```

Figure A.16 Example of configuration file sourceImportConfig_Vessel.json.

For the fleet category, the template includes the paths of the positioning files relative to the "position" directory in root.json ("positionPaths"), the positioning protocol ("positionFormat"), and the platform used to record the position data ("positionPlatform"). The fields "sourceCategory", "positionFormat", and "positionPlatform" are always set as fleet, AIS, and PamGuard.

```
{
"sourceCategory": "fleet",
"sourceName": "fleet",
"positionPaths": "AIS (PamGuard).csv",
"positionFormat": "AIS",
"positionPlatform": "PamGuard"
}
```

Figure A.17 Example of configuration file sourceImportConfig_Fleet.json.

A.1.7 Navigation Process

There is one module-specific configuration file in Navigation Process, named NavigationProcessConfig.json. The user must create one NavigationProcessConfig.json file for each sourceName-receiverName combination to be processed. An example of the configuration file is shown in Figure A.18.

The template includes the name of the receiver ("receiverName"), the name of the source ("sourceName"), the duration in seconds of the "smoothing" window used to reduce the chance of clustered positions ("smoothWindow"), the maximum duration in seconds with no position data that is allowed for the navigation parameters of an event

detected within that period to be processed ("maxTimeGap"), and the interpolation method used to estimate the position at the time of the detections ("interpMethod").

```
{
"receiverName": "Buoy1_H1",
"sourceName": "Airgun",
"smoothWindow": 20,
"maxTimeGap": 300,
"interpMethod": "linear"
}
```

Figure A.18 Example of configuration file navigationProcessConfig*.json.

A.1.8 Tables

There are three CSV tables related to settings of specific modules: the 'Pulse' table for the ConstantRate detector in the Audio Detect module, the 'Time Offset' table for the "multiple time offset" option in the Audio Process module, and the 'Vessel Database' table for the fleet source category in the Source Import module. The tables start with a one-line header containing the field names, followed by the data lines containing the field values. The field names and values are comma-separated. All three tables are stored in the 'configdb' folder under the "block" directory in root.json.

The 'Pulse' table is used by the ConstantRate detector to locate sound events emitted at a steady rate. The header has three fields: the time of the first pulse in seconds relative to the start of the audio file ("FirstPulse_s"), the spacing in milliseconds between consecutive pulses ("PulseInterval_ms"), and the name of the audio file, including extension ("AudioName"). The 'Pulse' table has to be populated manually and requires the user to inspect the audio files with an audio editor to extract the relevant parameters. An example is shown in Figure A.19.

```
FirstPulse_s,PulseInterval_ms,AudioName
17.710,28680,20140406_183314_4589586_7200000.raw2int16
11.527,28660,20140406_183544_4589586_7200000.raw2int16
5.066,27500,20140406_183814_4589586_7200000.raw2int16
```

Figure A.19 Example of a 'Pulse' table for a 'ConstantRate' detector.

The 'Time Offset' table is used by the Audio Process module to correct the difference between the detection timestamps, dependent on the PC clock, and the UTC timestamps from the navigation data. The header has two fields: the timestamp of the detection in local PC time ("Timestamp_yyyymmddTHHMMSS"), and the corresponding PC-UTC difference in seconds ("TimeOffset_s"). The 'Time Offset' table has to be populated manually and requires the user to find the time offset for several PC timestamps. This information can be obtained from the GPS and AIS files generated by PAMGUARD or SeicheSSV. The table takes the name timeOffset.csv.

As opposed to a single time offset, a 'Time Offset' table can compensate for time deviations from UTC that vary between audio files or within them. This solution is particularly important in large audio datasets from long campaigns, since conventional PC clock oscillators can introduce cumulative time errors of ~1 s per day (10µs/s). In general, one data line per audio file is sufficient, since a noticeable offset should not be expected for typical file durations. An example of this table is shown in Figure A.20.

```
Timestamp_yyyymmddTHHMMSS,TimeOffset_s
20140406T183314,3590
20140406T183544,3590
20140406T183814,3590
```

Figure A.20 Example of a 'timeOffset.csv' file for variable PC time offset correction in the 'Audio Process' module.

The 'Vessel Database' table provides the specifications of a list of relevant vessels. When the source category `fleet` is selected in a `sourceImportConfig*.json` file, the Source Import module reads the time and position data for a number of vessels from an AIS navigation file(s). The 'Vessel Database' determines what vessels to read from the AIS file(s) and provides the specifications for each of them. The header has six fields: the vessel's MMSI number (`"mmsi"`), name (`"vesselName"`), length in metres (`"vesselLength"`), width in metres (`"vesselBeam"`), draft in metres (`"vesselDraft"`), and size in tonnes (`"vesselGrossTonnage"`). The 'Vessel Database' table has to be populated manually and requires the user to find vessel specifications through its unique MMSI number. This can be done through online maritime tracking resources[41]. The file is named `vesseldb.csv`.

Creating a 'Vessel Database' is not strictly necessary, but is a useful addition in a context where the activity of surrounding vessels needs to be monitored (e.g., to monitor the operations, or to assess the acoustic masking potential from passing vessels). An example of this table is shown in Figure A.21.

```
mmsi,vesselName,vesselLength,vesselBeam,vesselDraft,vesselGrossTonnage
218582000,MSC Charleston,324.9,42.8,,89954
236483000,Sanco Star,80,16,,3953
239962000,Makronissos,244,42,,57062
259645000,Artemis Arctic,74.3,18,,3947
```

Figure A.21 Example of a 'vesseldb.csv' file for `'fleet'` category in the 'Source Import' module.

[41] www.marinetraffic.com/
www.vesselfinder.com

A.2 Audio Database

The Audio Database folder stores MAT files containing the settings and resampled data from the audio files stored under the "audio" directory in root.json. Audio Database files are created by the Audio Import module for each audio file in audioPaths.json and selected channel, and are saved in the 'audiodb' folder under the "block" directory.

One Audio Database file is generated for each audio file, channel, and new sampling rate combination. The database takes its name from the audio file it originates from, followed by the channel and new sampling rate, such as in <AUDIO_NAME>_ch<CHANNEL>_fr<RESAMPLE_RATE>.mat.

Reading and processing audio files in MATLAB can be particularly slow (1.4 MB/s with an Intel Core i5-4300U @ 1.9 GHz). Acoustic Database files can dramatically increase the speed of recurrent reading and processing in two ways: (1) by resampling data at a lower sampling rate, sufficient to characterise the target source, and (2) by saving that data in a format that MATLAB can access quickly (.mat).

An Audio Database file is a MATLAB structure with two single-element substructures: AudImpConfig and AudImpData.

```
AudioDatabase = struct('AudImpConfig',AudImpConfig,[],...
    'AudImpData',AudImpData);
```

Figure A.22 Initialised AudioDatabase structure.

AudImpConfig contains the audio file specifications, transferred directly from the corresponding configuration file audioImportConfig*.json. It also includes a logical variable that indicates the joint validity of the configuration values ("inputStatus"), the name of the configuration file ("configFileName"), and the number of audio samples ("audioLength").

```
AudImpConfig = struct('inputStatus',[],'configFileName',[],...
    'audioFormat',[],'channel',[],'resampleRate',[],...
    'sampleRate',[],'bitDepth',[],'numChannels',[],...
    'endianness',[],'audioLength',[]);
```

Figure A.23 Initialised AudImpConfig substructure from an Audio Database.

AudImpData contains the absolute path of the audio file ("audioPath"), and the floating-point amplitude of the audio samples from the specified channel ("audioData").

```
AudImpData = struct('audioPath',[],'audioData',[]);
```

Figure A.24 Initialised AudImpData substructure from an Audio Database.

A.3 Detection Database

The Detection Database folder contains four types of MAT file: raw scores, covariance, eigendata, and performance data. The database is generated only for the `NeymanPearson` detector. The files are created by function `preProcessNeymanPearson.m` from the Audio Detect module and saved in the "detectdb" folder under the `"block"` directory in `root.json`. The raw scores, covariance, and eigendata files are generated only for the estimator-correlator algorithms of the Neyman-Pearson detector.

The content of the four Detection Database files is described in the next sub-sections.

A.3.1 Raw Score Data

The raw scores file (.mat) is a single-element structure containing the matrix of raw scores (`"rawScoreMatrix"`) and the settings used to build it. The settings comprise the duration of the training segments in seconds (`"kernelDuration"`), their sampling rate *after* resampling (`"sampleRate"`), the minimum SNR that a training segment must have for it to be considered (`"minSnrLevel"`), and the SNR of the collated training segments (`"snrLevels"`).

```
RawScoreData = struct('kernelDuration',[],'sampleRate',[],...
   'minSnrLevel',[],'snrLevels',[],'rawScoreMatrix',[]);
```

Figure A.25 Initialised `RawScoreData` structure from Detection Database.

A matrix of *raw scores* contains all the information that is necessary to build a covariance matrix. It comprises as many rows as variables N and as many columns as training observations M. The number of variables is equivalent to the number of samples in a training observation. There are two raw scores matrices, one for the target signal and one for the background noise. These are built from training audio segments stored in subfolders 'signal' and 'noise' under the parent folder `trainFolder` given in the configuration file `audioDetectConfig_NeymanPearson.json` (see Sub-Section A.1.3).

The raw scores files are created by function `rawScores.m`. The function compiles the training segments to then *trim* and *resample* them based on the parameters given in the configuration file (`"kernelDuration"`, `"resampleRate"`). Any resampling can be alternatively applied by function `covariance.m` (see Sub-Section A.3.2).

The raw scores files follow the naming convention `<RAWTYPE>_<SOURCE>_<DETECTOR>_FS<FS>_T<T>.mat`, where `<RAWTYPE>` is the type of raw scores file (`RawNoise`, `RawSignal`), `<SOURCE>` is the name of the target source that the training data corresponds to, `<DETECTOR>` is the Neyman-Pearson algorithm (`ECW` for the estimator-correlator in WGN, and `ECC` for the estimator-correlator in CGN[42]), `<FS>` is the detection

[42] WGN and CGN refer to White Gaussian Noise and "Coloured" (or Correlated) Gaussian Noise

sampling rate in Hertz *after* resampling, and <T> is the duration of the processing window in seconds. Note that RawNoise is only required by the ECC algorithm, while RawSignal is required by both ECC and ECW. The energy detector (ED) does not rely on training data, as signal and noise are both assumed white Gaussian.

A.3.2 Covariance Data

The covariance file (.mat) is a single-element structure containing the covariance matrix ("covarianceMatrix") and the settings used to build it. The settings comprise the duration of the training segments ("kernelDuration"), and their sampling rate *after* resampling ("sampleRate").

```
CovarianceData = struct('kernelDuration',[],'sampleRate',[],...
   'covarianceMatrix',[]);
```

Figure A.26 Initialised CovarianceData structure from Detection Database.

A *covariance* matrix contains the variances of all N variables from the M training observations (diagonal entries) and the covariances between every variable pair (off-diagonal entries). It is built from a $[N, M]$ matrix of raw scores and it comprises as many rows and columns as variables. There are two covariance matrices, one for the target signal $\mathbf{\Sigma}_s$ and one for the background noise $\mathbf{\Sigma}_w$.

The covariance files are created by function covariance.m. The function computes the *normalised* covariance matrices from the matrices of raw scores. In practice, we are interested in the normalised covariance matrices $\breve{\mathbf{\Sigma}}_s = \sigma_s^{-2}\mathbf{\Sigma}_s$ and $\breve{\mathbf{\Sigma}}_w = \sigma_w^{-2}\mathbf{\Sigma}_w$, as these are independent of the variance of the training observations. The normalised covariance matrices $\breve{\mathbf{\Sigma}}_{s,w}$ are calculated from training observations that have been previously normalised by their variance $\sigma^2_{s,w}$. Note that the mean of the diagonal entries in $\breve{\mathbf{\Sigma}}_{s,w}$ is 1.

In a high-dimensional setting ($M < N$), the sample covariance matrix (SCM) becomes inaccurate and a better covariance estimator is needed. covariance.m calls the 'Covariance Estimation' package to address this issue (see Appendix B). The library offers a range of *shrinkage* covariance estimators that can help minimise the impact that a noisy covariance matrix estimate may have on the detection performance.

The Audio Detect module can apply provisional *resampling* and *filtering* to the audio segments to be processed for detections. This is done to improve the SNR and, as a result, the performance of the detector. covariance.m applies that same filtering and resampling to the input matrix of raw scores. This is a fundamental step as it ensures that the detector performs correctly by guaranteeing consistency between the training data and the measurements to be processed for detections.

The covariance files follow the naming convention <COVTYPE>_<SOURCE>_<DETECTOR>_FA<FA>_FB<FB>_FS<FS>_T<T>_<ESTIMATOR>.mat, where <COVTYPE> is the

type of covariance file (`CovarianceNoise`, `CovarianceSignal`), `<SOURCE>` is the name of the target source that the training data corresponds to, `<DETECTOR>` is the Neyman-Pearson algorithm (`ECW` for the estimator-correlator in WGN, and `ECC` for the estimator-correlator in CGN), `<FA>` and `<FB>` are the bottom and top cuttoff frequencies of the filter in Hertz, `<FS>` is the detection sampling rate in Hertz *after* resampling, `<T>` is the duration of the processing window in seconds, and `<ESTIMATOR>` is the covariance estimator. `CovarianceNoise` is only required by the `ECC` algorithm, while `CovarianceSignal` is required by both `ECC` and `ECW`. No covariance structure is generated for the energy detector (`ED`).

A.3.3 Eigendata

The eigendata file (.mat) is a single-element structure containing the normalised eigenvalues (`"signalEigenValuesNorm"`, `"noiseEigenValuesNorm"`) and eigenvectors (`"signalEigenVectors"`, `"noiseEigenVectors"`) of specific matrices, along with general settings such as the duration of the training segments (`"kernelDuration"`) and their sampling rate *after* resampling (`"sampleRate"`). The content in the structure depends on the detection algorithm (`ecw`, `ecc`). For the estimator-correlator in WGN (`ecw`), the eigendata file contains the eigenvectors $\mathbf{V}_s$ and eigenvalues $\check{\boldsymbol{\lambda}}_s$ of the *normalised* covariance matrix of the target signal $\check{\boldsymbol{\Sigma}}_s$. For the estimator-correlator in CGN (`ecc`), the eigendata file contains the eigenvectors $\mathbf{V}_w$ and eigenvalues $\check{\boldsymbol{\lambda}}_w$ of the *normalised* covariance matrix of the target noise $\check{\boldsymbol{\Sigma}}_w$, and the eigenvectors $\mathbf{V}_B$ and eigenvalues $\check{\boldsymbol{\lambda}}_B$ of the compound signal-noise matrix $\check{\mathbf{B}} = \check{\mathbf{A}}^T\check{\boldsymbol{\Sigma}}_s\check{\mathbf{A}}$, where $\check{\mathbf{A}} = \mathbf{V}_w\check{\boldsymbol{\Lambda}}_w^{-1/2}$. The function `eigenEquation.m` computes the eigendata for the two estimator-correlators. Note that the scaling factor applied to the covariance matrix does not affect the matrix of eigenvectors (i.e., $\mathbf{V}_{s,w} = \check{\mathbf{V}}_{s,w}$).

```
EigenData = struct('kernelDuration',[],'sampleRate',[],...
    'noiseType',[],'signalEigenVectors',[],...
    'signalEigenValuesNorm',[],'noiseEigenVectors',[],...
    'noiseEigenValuesNorm',[]);
```

Figure A.27 Initialised `EigenData` structure from Detection Database.

A square matrix $\mathbf{A}$ can be expressed as the product $\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{-1}$, where $\mathbf{V}$ is the matrix of eigenvectors (or *modal* matrix) and $\boldsymbol{\Lambda}$ is the matrix of eigenvalues (or *diagonal* matrix). This is commonly known as the *eigen decomposition* of $\mathbf{A}$. In $\mathbf{V}$, each column is an eigenvector, and $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots \lambda_N\}$ with λ_n representing the n^{th} eigenvalue. The modal and diagonal matrices are referred to as the *eigendata*.

The eigendata is of particular interest as it can be used to *diagonalise* a covariance matrix $\boldsymbol{\Sigma}$ through the expression $\mathbf{V}^{-1}\boldsymbol{\Sigma}\mathbf{V}$ and *decorrelate* an observation $\mathbf{x} = \{x_1, x_2, \ldots x_N\}$ with $\mathbf{V}^{-1}\mathbf{x}$. The decorrelation transforms the fraction of energy in $\mathbf{x}$ that is statistically similar to $\boldsymbol{\Sigma}$ into white Gaussian noise. Note that for a symmetric matrix (such as the covariance $\boldsymbol{\Sigma}$), the modal matrix is *orthogonal* ($\mathbf{V}^{-1} = \mathbf{V}^T$).

The eigen decomposition is necessary for the estimator-correlators, since only with the eigendata one can arrive at the expressions of the *test statistic* $T(\mathbf{x})$ and *performance curves* $p_{0,1}(T)$ (Kay, 1998). The energy detector doesn't use covariance matrices or eigen data, as the test statistic is simply the energy of the observation $\mathbf{x}$ and the performance curves are given by the Chi-squared PDF of the N-variable vector $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, where σ^2 is the variance of the null (σ_w^2) or alternative ($\sigma_s^2 + \sigma_w^2$) hypotheses.

The eigen data files follow the naming convention `EigenData_<SOURCE>_<DETECTOR>_FA<FA>_FB<FB>_FS<FS>_T<T>_<ESTIMATOR>.mat`. The terms within pointy brackets '<>' are the same as those in Sub-Section A.3.2. No eigendata structure is generated for the energy detector (`ED`).

A.3.4 Performance Data

The performance file (.mat) is a multi-element structure containing detection performance curves and related contextual information for multiple SNR values and a reference noise variance $\sigma_{w,ref}^2 = 1$. The performance data refers to the probability density functions (PDF) of the null and alternative hypotheses $p_{0,1}(T)$ and the probability functions (PF) of false alarm and detection $P_{FA,D}(T)$. Note that $P_{FA,D}(T)$ is the right-tail probability (RTP) of $p_{0,1}(T)$. The detection performance is fully characterised by $p_{0,1}(T)$ or $P_{FA,D}(T)$, insofar as the covariance matrix estimate is accurate.

The following parameters are included in each element of the performance structure: the type of Neyman-Pearson detection algorithm (`"detectorType"`), the variance of the signal (`"signalVariance"`) and the noise (`"noiseVariance"`) on which the performance curves are based, the corresponding SNR in decibels (`"snrLevel"`), the number of variables or sample length of the audio segment (`"nVariables"`), the normalised bottom and top cuttoff frequencies of the detection filter (`"cutoffFreqns"`), the test statistic axis for the PDF of the null hypothesis and the PF of false alarm (`"axisFalseAlarm"`), the test statistic axis for the PDF of the alternative hypothesis and the PF of detection (`"axisDetection"`), the PDF of the null hypothesis (`"pdfFalseAlarm"`) and alternative hypothesis (`"pdfDetection"`), and the PF of false alarm (`"rtpFalseAlarm"`) and detection (`"rtpDetection"`).

```
PerformanceData = struct('detectorType',[],'signalVariance',[],...
    'noiseVariance',[],'snrLevel',[],'nVariables',[],...
    'cutoffFreqns',[],'axisFalseAlarm',[],'axisDetection',[],...
    'pdfFalseAlarm',[],'pdfDetection',[],'rtpFalseAlarm',[],...
    'rtpDetection',[]);
```

Figure A.28 Initialised `PerformanceData` structure from Detection Database.

Given a target probability of false alarm α, the threshold $\gamma'(\mathbf{x})$ for observation $\mathbf{x}$ is obtained from $P_{FA}(T)$. The observation is classified as "detection" if the test statistic $T(\mathbf{x})$ is larger than the threshold $\gamma'(\mathbf{x})$.

The performance files are created by the function `characterisePerformance.m` using the information in the eigendata files. For the estimator-correlators (`ecw`, `ecc`), the performance curves are calculated from the normalised eigenvalues $\check{\boldsymbol{\lambda}}_B$ and $\check{\boldsymbol{\lambda}}_s$, the signal variance σ_s^2, and the noise variance σ_w^2. The energy detector (`ed`), only requires the number of variables N, the signal variance σ_s^2, the noise variance σ_w^2, and the cutoff frequencies of the detection filter $f_{1,2}$ to calculate the performance curves.

`characterisePerformance.m` calculates $p_{0,1}(T)$ and $P_{FA,D}(T)$ for multiple SNR, from -50 dB to 50 dB in 1 dB steps. Is this limited number of SNR values enough to characterise the performance of a detector? $p_{0,1}(T)$ and $P_{FA,D}(T)$ represent the PDF and PF of a test statistic $T(\mathbf{x})$, where $\mathbf{x}\sim\mathcal{N}(0,\boldsymbol{\Sigma}_w)$ for the null hypothesis and $\mathbf{x}\sim\mathcal{N}(0,\boldsymbol{\Sigma}_s+\boldsymbol{\Sigma}_w)$ for the alternative hypothesis. One can see that $T(\mathbf{x})$ depends on the variance of the observation σ_t^2 $(=\sigma_w^2+\sigma_s^2)$, and so do $p_{0,1}(T)$ and $P_{FA,D}(T)$. Initially, this would mean that the performance curves need to be computed for every combination of signal and noise variances. Fortunately, calculating $p_{0,1}(T)$ for a number of SNR and a reference noise variance $\sigma_{w,ref}^2=1$ is sufficient, as one can arrive at the PDF curves for a specific signal and noise variance by simply scaling the "reference" curves by the noise variance of $\mathbf{x}$. That is, $p_0(T,\sigma_w^2)=\sigma_w^{-2}p_0(T\sigma_w^{-2},1)$ and $p_1(T,SNR,\sigma_w^2)=\sigma_w^{-2}p_1(T\sigma_w^{-2},SNR,1)$. Therefore, a limited number of "reference" performance curves can be used to calculate $\gamma'(\mathbf{x})$ and $T(\mathbf{x})$, insofar as they cover the range of SNR values expected for the data to be treated for detections.

The performance files follow the naming convention `PerformanceData_<SOURCE>_<DETECTOR>_FA<FA>_FB<FB>_FS<FS>_T<T>_<ESTIMATOR>.mat`. The terms within pointy brackets '<>' are the same as those in the previous two sub-sections.

A.4 Navigation Database

The Navigation Database folder contains a MAT file used to store the configuration information and navigation data for the receivers, sources, and vessels. The Navigation Database file is created by the Receiver Import and Source Import modules and saved with the name `navigationdb.mat` in the "navigationdb" folder under the `"block"` directory in `root.json`.

The Navigation Database file is a MATLAB structure containing six multi-element sub-structures and a list of receivers (`"receiverList"`), sources (`"sourceList"`), and vessels (`"vesselList"`) associated with the elements in those sub-structures. The six sub-structures, described below in detail, contain the configuration information of the receivers (`"RecImpConfig"`), sources (`"SouImpConfig"`), and 'fleet' vessels (`"VesImpConfig"`), and the navigation data of the receivers (`"RecImpData"`), sources (`"SouImpData"`), and 'fleet' vessels (`"VesImpData"`).

```
NavigationDatabase = struct(...
   'receiverList',[],'sourceList',[],'vesselList',[],...
   'RecImpConfig',RecImpConfig,'RecImpData',RecImpData,...
   'SouImpConfig',SouImpConfig,'SouImpData',SouImpData,...
   'VesImpConfig',VesImpConfig,'VesImpData',VesImpData);
```

Figure A.29 Initialised "NavigationDatabase" structure.

"RecImpConfig", "SouImpConfig", and "VesImpConfig" contain the import settings for the position data of the receivers, sources, and 'fleet' vessels, copied directly from the corresponding configuration files receiverImportConfig*.json and sourceImportConfig*.json. They also include a logical variable that indicates the joint validity of the settings ("inputStatus"), and the name of the configuration file ("configFileName"). The sub-structures are shown in Figure A.30 to Figure A.32.

```
RecImpConfig = struct('inputStatus',[],'configFileName',[],...
   'receiverCategory',[],'receiverName',[],'receiverOffset',[],...
   'receiverOffsetMode',[],'positionPaths',[],...
   'positionFormat',[],'positionPlatform',[],'vesselId',[],...
   'mmsi',[],'latitude',[],'longitude',[],'depth',[]);
```

Figure A.30 Initialised "RecImpConfig" substructure from Navigation Database.

```
SouImpConfig = struct('inputStatus',[],'configFileName',[],...
  'sourceCategory',[],'sourceName',[],'sourceOffset',[],...
  'sourceOffsetMode',[],'positionPaths',[],'positionFormat',[],...
  'positionPlatform',[],'vesselId',[],'sourceId',[],'mmsi',[],...
  'vesselName',[],'vesselLength',[],'vesselBeam',[], ...
  'vesselDraft',[],'vesselGrossTonnage',[],'latitude',[],...
  'longitude',[],'depth',[]);
```

Figure A.31 Initialised "SouImpConfig" substructure from Navigation Database.

```
VesImpConfig = struct('inputStatus',[],'configFileName',[],...
  'sourceCategory',[],'sourceName',[],'sourceOffset',[],...
  'sourceOffsetMode',[],'positionPaths',[],'positionFormat',[],...
  'positionPlatform',[],'vesselId',[],'sourceId',[],'mmsi',[],...
  'vesselName',[],'vesselLength',[],'vesselBeam',[],...
  'vesselDraft',[],'vesselGrossTonnage',[],'latitude',[],...
  'longitude',[],'depth',[]);
```

Figure A.32 Initialised "VesImpConfig" substructure from Navigation Database.

"RecImpData", "SouImpData", and "VesImpData" contain the following parameters for the receiver, source, and 'fleet' vessel within each element: the absolute paths of the position files ("positionPaths"), the absolute PC times ("pcTick") and UTC times ("utcTick") referred to 01 Jan 0000, the geographic positions in degrees ("latitude", "longitude"), and the depth in metres below water surface ("depth"). The sub-structures are shown in Figure A.33 to Figure A.35.

```
RecImpData = struct('positionPaths',[],'pcTick',[],'utcTick',[],...
  'latitude',[],'longitude',[],'depth',[]);
```

Figure A.33 Initialised "RecImpData" substructure from Navigation Database.

```
SouImpData = struct('positionPaths',[],'pcTick',[],'utcTick',[],...
  'latitude',[],'longitude',[],'depth',[]);
```

Figure A.34 Initialised "SouImpData" substructure from Navigation Database.

```
VesImpData = struct('positionPaths',[],'pcTick',[],'utcTick',[],...
  'latitude',[],'longitude',[],'depth',[]);
```

Figure A.35 Initialised "VesImpData" substructure from Navigation Database.

A.5 Acoustic Database

The Acoustic Database folder stores lightweight MAT files containing the configuration information and results from each module in ABACUS. One Acoustic Database file is created for each selected audio file in `audioPaths.json` and selected channel, and saved in the "acousticdb" folder under the "block" directory. The Acoustic Database takes its name from the audio file it originates from.

An Acoustic Database file contains a MATLAB structure with two multi-element sub-structures: "AcoConfig" and "AcoData".

```
AcousticDatabase = struct('AcoConfig',AcoConfig,'AcoData',AcoData);
```

Figure A.36 Initialised AcousticDatabase structure.

Each element in "AcoConfig" contains the settings for one receiverName-sourceName combination as returned by the modules Audio Import ("AudImpConfig"), Audio Detect ("AudDetConfig"), Audio Process ("AudProConfig"), Receiver Import ("RecImpConfig"), Source Import ("SouImpConfig", "VesImpConfig"), and Navigation Process ("NavProConfig"), along with contextual information such as the name of the associated Audio Database file ("audiodbName"), the extracted channel ("channel"), the sampling rate *after* resampling ("resampleRate"), and the name of the receiver ("receiverName") and the source ("sourceName"). "AudImpConfig", "AudDetConfig", "AudProConfig", "RecImpConfig", "SouImpConfig", "VesImpConfig" and "NavProConfig" are single-element structures with information obtained directly from configuration files, with various additional fields such as a logical variable that indicates the joint validity of the settings ("inputStatus"), the name of the configuration file ("configFileName"), the channel number ("channel"), and the resampling rate ("resampleRate"). For details about the fields in these module-related sub-structures, refer to Section A.1.

```
AcoConfig = struct('audiodbName',[],'channel',[],...
   'resampleRate',[],'receiverName',[],...
   'sourceName',[],'AudImpConfig',AudImpConfig,...
   'RecImpConfig',RecImpConfig,'SouImpConfig',SouImpConfig,...
   'VesImpConfig',VesImpConfig,'AudDetConfig',AudDetConfig',...
   'AudProConfig',AudProConfig,'NavProConfig',NavProConfig);
```

Figure A.37 Initialised `AcoConfig` substructure from Acoustic Database.

```
AudImpConfig = struct('inputStatus',[],'configFileName',[],...
   'audioFormat',[],'channel',[],'resampleRate',[],...
   'sampleRate',[],'bitDepth',[],'numChannels',[],...
   'endianness',[],'audioLength',[]);
```

Figure A.38 Initialised `"AudImpConfig"` substructure in `AcoConfig` multi-structure.

```
RecImpConfig = struct('inputStatus',[],'configFileName',[],...
   'receiverCategory',[],'receiverName',[],'receiverOffset',[],...
   'receiverOffsetMode',[],'positionPaths',[],...
   'positionFormat',[],'positionPlatform',[],'vesselId',[],...
   'mmsi',[],'latitude',[],'longitude',[],'depth',[]);
```

Figure A.39 Initialised `"RecImpConfig"` substructure in `AcoConfig` multi-structure.

```
SouImpConfig = struct('inputStatus',[],'configFileName',[],...
  'sourceCategory',[],'sourceName',[],'sourceOffset',[],...
  'sourceOffsetMode',[],'positionPaths',[],'positionFormat',[],...
  'positionPlatform',[],'vesselId',[],'sourceId',[],'mmsi',[],...
  'vesselName',[],'vesselLength',[],'vesselBeam',[], ...
  'vesselDraft',[],'vesselGrossTonnage',[],'latitude',[],...
  'longitude',[],'depth',[]);
```

Figure A.40 Initialised `"SouImpConfig"` substructure in `AcoConfig` multi-structure.

```
VesImpConfig = struct('inputStatus',[],'configFileName',[],...
  'sourceCategory',[],'sourceName',[],'sourceOffset',[],...
  'sourceOffsetMode',[],'positionPaths',[],'positionFormat',[],...
  'positionPlatform',[],'vesselId',[],'sourceId',[],'mmsi',[],...
  'vesselName',[],'vesselLength',[],'vesselBeam',[],...
  'vesselDraft',[],'vesselGrossTonnage',[],'latitude',[],...
  'longitude',[],'depth',[]);
```

Figure A.41 Initialised `"VesImpConfig"` substructure in `AcoConfig` multi-structure.

```
AudDetConfig = struct('inputStatus',[],'configFileName',[],...
  'channel',[],'resampleRate',[],'receiverName',[],...
  'sourceName',[],'mirrorReceiver',[],'detector',[],...
  'DetectParameters',[]);
```

Figure A.42 Initialised `"AudDetConfig"` substructure in `AcoConfig` multi-structure.

```
AudProConfig = struct('inputStatus',[],'configFileName',[],...
  'channel',[],'resampleRate',[],'receiverName',[],...
  'sourceName',[],'freqLimits',[],'bandsPerOctave',[],...
  'cumEnergyRatio',[],'audioTimeFormat',[],'timeOffset',[],...
  'tags',[]);
```

Figure A.43 Initialised "AudProConfig" substructure in AcoConfig multi-structure.

```
NavProConfig = struct('inputStatus',[],'configFileName',[],...
'channel',[],'resampleRate',[],'receiverName',[],'sourceName',[],...
'smoothWindow',[],'maxTimeGap',[],'interpMethod',[]);
```

Figure A.44 Initialised "NavProConfig" substructure in AcoConfig multi-structure.

Similarly, each element in AcoData contains the results for one receiverName-sourceName combination as returned by the modules Audio Detect ("AudDetData"), Audio Process ("AudProData"), Navigation Process ("RecProData", "SouProData", "VesProData"), and Revision ("RevData"), along with the same contextual information as in AcoConfig. The fields of the single-element structures "AudDetData", "AudProData", "RecProData", "SouProData" and the multi-element structure "VesProData" are described below. The "RevData" structure contains the fields for the analysis, interpretation, and revision of the results in the Acoustic Database files. The functionalities associated with this substructure are to be implemented in a future version of the software, as part of the integration of a graphical user interface (GUI).

```
AcoData = struct('audiodbName',[],'channel',[],'resampleRate',[],...
   'receiverName',[],'sourceName',[],'AudDetData',AudDetData,...
   'AudProData',AudProData,'RecProData',RecProData,...
   'SouProData',SouProData,'VesProData',VesProData,...
   'RevData',RevData);
```

Figure A.45 Initialised AcoData substructure from Acoustic Database.

The "AudDetData" structure includes the following times, in seconds, relative to the start of the audio file: the times of the start, end and absolute maximum of each detected target event ("signalTime1", "signalTime2", "signalTime") and the start and end times of the associated background noise events ("noiseTime1", "noiseTime2").

```
AudDetData = struct('signalTime',[],'signalTime1',[],...
    'signalTime2',[],'noiseTime1',[],'noiseTime2',[]);
```

Figure A.46 Initialised "AudDetData" substructure in AcoData multi-structure.

The "AudProData" structure contains the absolute PC and UTC times of the detections in seconds relative to 00 Jan 0000 ("signalPcTick", "signalUtcTick"), the time difference in seconds between UTC and PC times ("timeOffset"), the timestamp of the audio file ("fileTimeStamp"), the absolute PC time in seconds of the first sample in the audio file relative to 00 Jan 0000 ("fileTick"), the PC times of the start and end of the detections ("signalTime1", "signalTime2") and related noise segments

("noiseTime1", "noiseTime2") in seconds relative to the start of the audio file, the adjusted time limits based on the cumulative energy curves ("signalEnergyTime1", "signalEnergyTime2", "noiseEnergyTime1", "noiseEnergyTime2"), the broadband peak, peak-to-peak, RMS, and exposure metrics of the detections ("signalZ2p", "signalP2p", "signalRMS", "signalExp") and related noise segments ("noiseZ2p", "noiseP2p", "noiseRms", "noiseExp"), the band RMS and exposure metrics of the detections ("signalRmsBand", "signalExpBand") and related noise segments ("noiseRmsBand", "noiseExpBand"), the nominal and exact central frequencies of the bands in Hertz ("nominalFreq", "centralFreq") and their cutoff frequencies ("cutoffFreq1", "cuttoffFreq2").

```
AudProData = struct('signalPcTick',[],'signalUtcTick',[],...
  'timeOffset',[],'fileTimestamp',[],'fileTick',[],...
  'signalTime1',[],'signalTime2',[],'signalEnergyTime1',[],...
  'signalEnergyTime2',[],'noiseTime1',[],'noiseTime2',[],...
  'noiseEnergyTime1',[],'noiseEnergyTime2',[],'signalZ2p',[],...
  'signalP2p',[],'signalRms',[],'signalExp',[],'noiseZ2p',[],...
  'noiseP2p',[],'noiseRms',[],'noiseExp',[],'signalRmsBand',[],...
  'signalExpBand',[],'noiseRmsBand',[],'noiseExpBand',[],...
  'nominalFreq',[],'centralFreq',[],'cutoffFreq1',[],...
  'cutoffFreq2',[]);
```

Figure A.47 Initialised "AudProData" substructure in AcoData multi-structure.

The "RecProData" structure contains the PC and UTC times of the detections in seconds relative to 00 Jan 0000 ("pcTick", "utcTick"), and the corresponding receiver's geographic position in degrees ("latitude", "longitude"), depth in metres ("depth"), course over ground in degrees ("course"), and speed in metres per second ("speed").

```
RecProData = struct('pcTick',[],'utcTick',[],'latitude',[],...
  'longitude',[],'depth',[],'course',[],'speed',[]);
```

Figure A.48 Initialised "RecProData" substructure in AcoData multi-structure.

The "SouProData" structure contains the PC and UTC times of the detections in seconds relative to 00 Jan 0000 ("pcTick", "utcTick"), and the corresponding sources's geographic position in degrees ("latitude", "longitude"), depth in metres ("depth"), course over ground in degrees ("course"), speed in metres per second ("speed"), distance and angle towards the receiver ("sou2recDistance", "sou2recBearing"), heading ("sourceHeading"), and emission angle ("sourceEmitAngle").

```
SouProData = struct('pcTick',[],'utcTick',[],'latitude',[],...
  'longitude',[],'depth',[],'course',[],'speed',[],...
  'sou2recDistance',[],'sou2recBearing',[],'sourceHeading',[],...
  'sourceEmitAngle',[]);
```

Figure A.49 Initialised "SouProData" substructure in AcoData multi-structure.

The "VesProData" structure contains the same fields as "SouProData", but in this case there are multiple vessels as opposed to just one source.

```
VesProData = struct('pcTick',[],'utcTick',[],'latitude',[],...
  'longitude',[],'depth',[],'course',[],'speed',[],...
  'sou2recDistance',[],'sou2recBearing',[],'sourceHeading',[],...
  'sourceEmitAngle',[]);
```

Figure A.50 Initialised "VesProData" substructure in AcoData multi-structure.

Acoustic Database files have all the information that is needed to analyse, represent and interpret the sound field from target acoustic sources. The Audio Detect, Audio Process and Navigation Process modules can interact with these files to retrieve or add information. One Acoustic Database file is generated for each selected audio file and receiverName-sourceName combination. The file takes its name from the audio file it originates from.

Appendix B

Libraries

This appendix describes in detail the MATLAB libraries coded to carry out a number of specialised calculations in ABACUS. The libraries are included as part of ABACUS, but their generic features and independent character make them suitable for a wide range of applications outside the software. The features include reading, filtering and resampling audio data, estimating covariance matrices, and calculating navigation parameters. The appendix is organised into twelve sections, each dedicated to a specific library, plus an additional section where we discuss the use of zero-padding for improving the spectral characterisation of short, band-limited signals (Section B.4).

B.1 Read Audio

The 'Read Audio' package extracts the audio data and header information from audio files in WAV and RAW formats. It contains two functions: `readWavHeader.m`, for reading the header information from a WAV file, and `readAudioFile.m`, for reading audio data from a WAV/RAW audio file. `readAudioFile.m` has the same functionalities as MATLAB's function `audioread.m`, but it is also capable of reading audio data from files with a corrupt header.

If the writing to a WAV audio file suddenly stops, its header may become corrupt. This may be caused by a power loss (e.g., low battery, loose connection) or be the result of software-hardware limitations or incompatibilities (e.g., CPU overload, low-speed flash memory, unstable telemetry link). As a result, some information in the header may be incomplete or incorrect, preventing most audio editing programs to access the audio data. In most cases, the sudden interruption of a recording only affects the WAV header and the last few audio bytes. Being able to recover this information is often critical, as in practice there is rarely a second chance to carry out new recordings.

B.2 Resample Audio

The 'Resample Audio' package resamples audio files in an efficient way using the Fast-Fourier Transform (FFT). The package contains two functions: `resampleAudioFile.m` for resampling the data from an audio file, and `resamplingFactors.m` for calculating the most efficient upsampling and downsampling factors.

`resampleAudioFile.m` is an alternative to MATLAB's function `resample.m` for resampling audio files efficiently using FFT filtering with overlap-save reconstruction. `resampleAudioFile.m` divides an audio file into manageable 50 MB segments. Those segments are filtered in the frequency domain, converted back into time waveforms, and joined together with the *overlap-save* method, using upsampling and downsampling along the process. This "small-segment" approach is faster and more memory-efficient than applying `resample.m` to an entire audio file. In most cases, using `resample.m` is impractical due to the computer's limited physical memory. `resampleAudioFile.m` calls the Read Audio package (see Section B.1).

An upsampling factor increases the processing times and memory usage. `resamplingFactors.m` calculates the optimal upsampling and downsampling factors that will allow `resampleAudioFile.m` to resample the audio data more efficiently, subject to a small error in the target sampling frequency. The aim of this function is to calculate the lowest upsampling and downsampling factors (P,Q) that will result in a sampling frequency $f_s\,P/Q$ equal to the target sampling frequency f_r within a given tolerance.

Error! Reference source not found. shows a conceptual graph for the algorithm in `resampleAudioFile.m`. In this process, the audio file is first divided into segments of

length M. Then, the function designs a low-pass FIR filter of length N' that works both as a *reconstruction filter* and *anti-alias filter*, to be applied after upsampling and before downsampling. The time response of the filter is zero-padded to a length $L' = M' + N' - 1$ and its FFT is calculated. The apostrophe refers to sample lengths from the upsampled data. For every non-overlapping segment in the audio file, the algorithm follows the next steps:

1. A segment of length M is read from the audio file.
2. The segment is upsampled by P and its amplitude increased by that same factor. The length for the upsampled segment is $M' = M \cdot P$.
3. The result is appended to the overlapping portion kept from the previous upsampled segment (*overlap save*), making the signal $L' = M' + N' - 1$ samples long. The last $N' - 1$ samples are reserved for the next segment.
4. The FFT is computed over the upsampled segment of length L'.
5. The FFT of the upsampled segment and recovery/anti-alias filter are multiplied.
6. The IFFT of the latter product is computed.
7. The $N' - 1$ samples affected by the filter's distortion effects are discarded.
8. The result is downsampled by Q and appended to the preceding resampled segments.

Note that the filter causes leading and trailing distortion effects. Both distorted sections can be displaced to the end of the segment by applying a circular shift $(N' - 1)/2$ samples long to the FIR filter. Alternatively, this can be achieved by using the complex conjugate of the filter's FFT. Due to its simplicity, the latter solution is implemented.

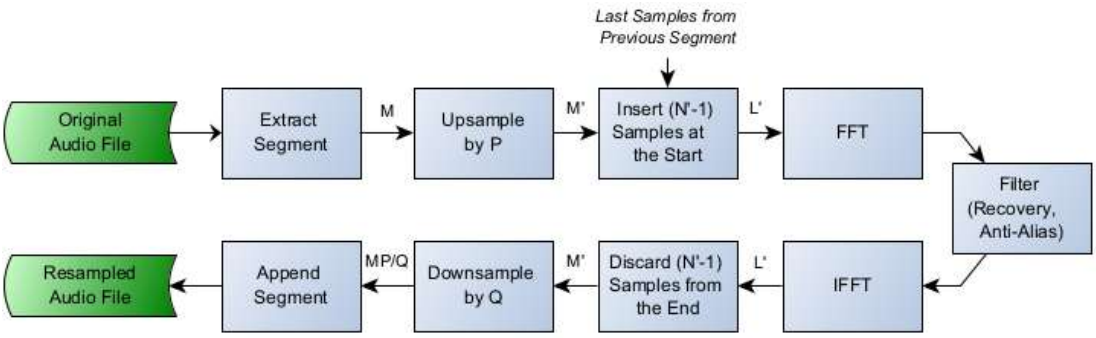

Figure B.1 Flow chart describing the resampling algorithm `resampleAudioFile.m`.

B.3 Filter Audio

The 'Filter Audio' package filters audio files in an efficient way using the Fast-Fourier Transform (FFT). The package contains only the function `filterAudioFile.m`.

`filterAudioFile.m` is an alternative to MATLAB's function `filter.m` for filtering audio files efficiently using the overlap-save method. `filterAudioFile.m` divides an

audio file into manageable 50 MB segments. Those segments are then individually filtered and combined using *overlap-save* reconstruction. This "small-segment" approach is similar to the one used in `resampleAudioFile.m` (see Sub-Section B.2). The solution is faster and more memory-efficient than applying `filter.m` to an entire audio file. `filterAudioFile.m` calls the Read Audio package (see Sub-Section B.1).

The function divides the audio signal into segments of length M and filters each segment with a filter of (a,b) input coefficients. The filtering approach is different depending on the type of filter response (FIR, IIR). For every non-overlapping segment in the audio file, the steps below are followed. **Error! Reference source not found.** and **Error! Reference source not found.** show the conceptual graphs of the filtering algorithm for the FIR (`a` =1) and IIR options.

1. A segment of length M is read from the audio file.
2. The result is appended to the overlapping portion kept from the previous segment (*overlap save*), obtaining a vector $L = M + N - 1$ samples long. The last $N - 1$ samples are reserved for the next segment.
3. The segment is filtered. The approach is different for the FIR or IIR filter options.
   a. *FIR filter*. The FFT of the segment and filter are calculated. The IFFT of their product is computed to obtain the filtered waveform.
   b. *IIR filter*. The segment is filtered using the `filter.m` function and the vector of final conditions from the previously filtered segment.
4. The $N - 1$ samples affected by the filter's distortion effects are discarded.

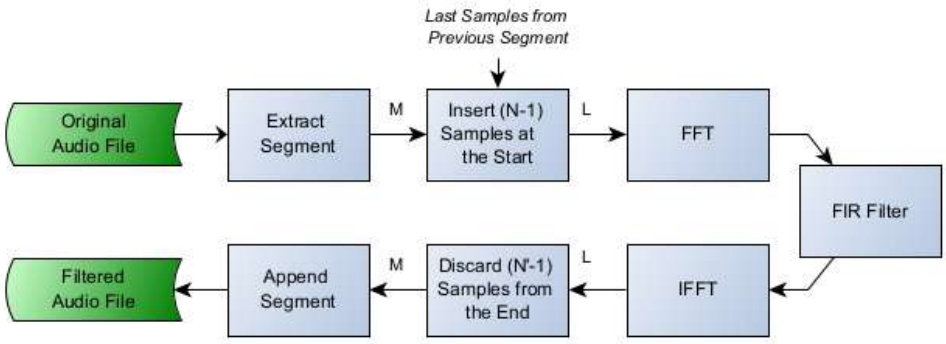

Figure B.2 Flow chart describing the FIR filtering algorithm in `filterAudioFile.m`.

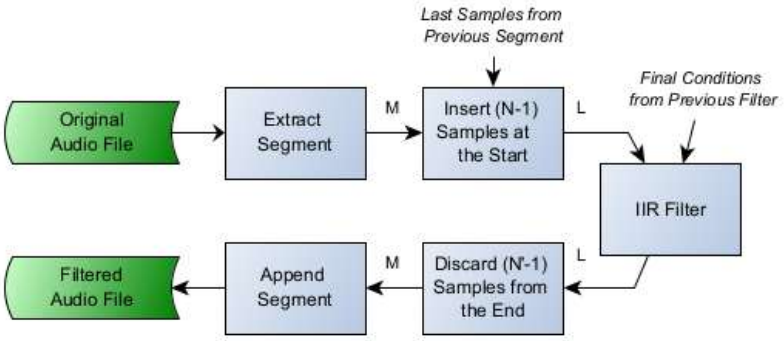

Figure B.3 Flow chart describing the IIR filtering algorithm in `filterAudioFile.m`.

B.4 Zero-Padding before Filtering Short, Band-Limited Signals

The *uncertainty principle* in signal processing states that a signal and its Fourier transform cannot be *highly concentrated*. That is, the product of a signal's duration Δt and bandwidth Δf must be larger than a constant K for the results to be meaningful.

$$\Delta t \cdot \Delta f \geq K \tag{B.1}$$

In the context of signal band analysis, where the bandwidth of each band is proportional to the its central frequency, the lower bands will be more likely to experience accuracy issues related to the uncertainty principle ($\Delta t \cdot \Delta f < K$).

Therefore, trying to filter short signals at low frequency bands will lead to problems. These problems manifest differently in FFT and conventional digital filters. In an FFT, not meeting the condition $\Delta t \cdot \Delta f \geq K$ means that the bandwidth of the filtering band Δf will be smaller than K times the frequency resolution $f_s/N = 1/\Delta t$, where f_s and N represent the sampling rate and the number of samples in time and frequency. In a digital filter, not meeting the uncertainty condition will delay the filtered signal, displacing a portion of the waveform outside the processing window. The end effect in either case is an *energy leakage* on the filtered signal.

In digital filters, the energy leakage is progressive and is characterised by lower energies at lower bands, where group delays are larger. The *group delay* (GD) is the time delay of the envelope of the individual single-frequency components of a signal and is a function of frequency. The GD of a filter is proportional to its order, inversely proportional to its bandwidth, and is also dependent on the filter type (e.g., Butterworth). In the FFT filtering, the leakage appears as fluctuations in the energy of nearby bands when $\Delta t\, \Delta f < K$ and is characterised by the impossibility of computing the band energy when the filter bandwidth is smaller than the spectral resolution ($\Delta t \Delta f < 1$).

In practice, as we will show later, the product $\Delta t \Delta f$ needs to be of the order of $K = 10$ to obtain a band level error of 1 dB. This introduces a limit on the frequency of the lowest band that can be accurately processed. For example, for an impact piling blow of 100 ms duration, the 400 Hz third-octave band is the lowest reliable band. This is

certainly a problem, since the maximum spectral energy from the piling blow occurs at a frequency of a few hundred Hertz (Jimenez-Arranz et al, 2020b). This same argument can be applied to other transient events such as pulses from airgun arrays, sparkers, or sub-bottom profilers (Jimenez-Arranz et al, 2020a).

We propose a simple but effective solution to address the problems associated with the filtering of short, band-limited signals. The solution consists in extending the duration of the signal by applying zero-padding. Adding zeros at the end of a zero-mean process has two effects: (1) it artificially improves the *spectral resolution*, and (2) increases the *length of the segment* to be filtered. These effects have respective implications on the FFT and digital filtering, which we describe below.

Padding the time signal with zeroes results in a better representation of the frequency spectrum through the addition of new spectral samples. Note that this is a form of interpolation in the frequency domain and as such it does not provide new information about the signal (Proakis & Manolakis, 2007). The consequence of this interpolation is that, in a FFT filter, a larger a number of samples will fall within each constant percentage band (CPB) from the filter bank. A band that covers multiple FFT points is less susceptible to errors arising from the integration of spectral samples whose combined bandwidth (multiple of $1/\Delta t$) is not identical to the actual bandwidth of the band. This discrepancy between bandwidths explains the error fluctuations that occur in the lowest bands when $\Delta f \Delta t$ approaches a value of $K = 1$. In this sense, the spectral interpolation resulting from zero-padding minimises any error caused by a mismatch between the bandwidth of the spectral points and the bandwidth of the band.

The increase in window length through zero-padding also prevents a filtered waveform to be displaced out of the processing window when a digital filter (FIR, IIR) is applied. The number of added zeros must be larger than the maximum GD of the frequencies in the passband. The GD of frequencies at the centre of the band tends to be notably smaller (2-5 times) than the GD near the cutoff frequencies.

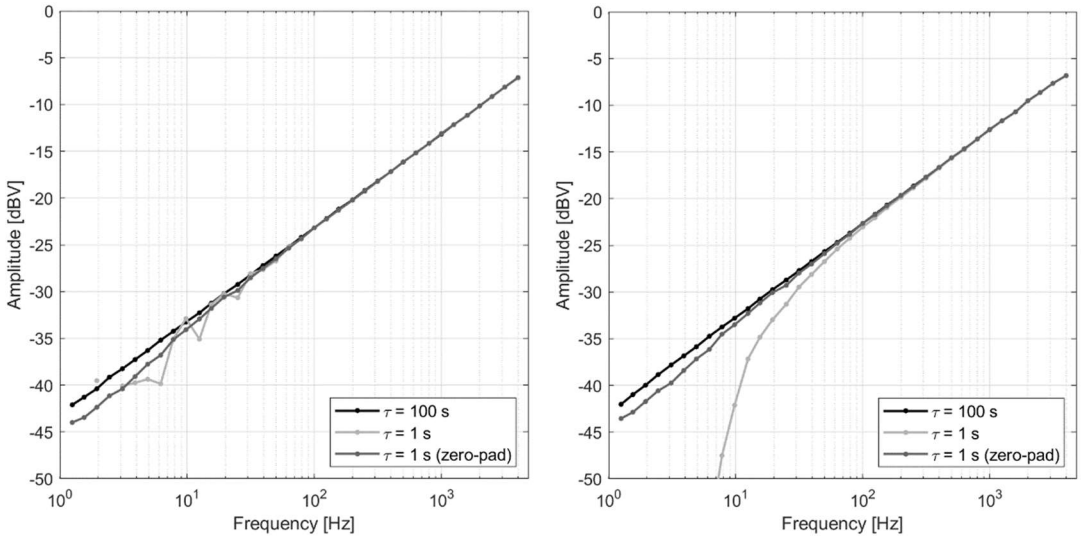

Figure B.4 Third-octave band level spectrum averaged over 100 white Gaussian noise (WGN) segments of duration 100 s (black, "exact" spectrum), and 1 s (light grey) for the case of FFT filtering (left) and digital filtering (right). The 1 s segments are zero-padded to 100 s (dark grey).

Figure B.4 shows the mean level spectrum in third-octave bands calculated over 100 segments of white Gaussian noise (WGN) of durations of 100 s (black) and 1 s (light grey) for the cases of FFT filtering (left) and digital IIR filtering (right). The results are compared with the average spectrum of 1 s segments padded with zeros (dark grey). The black curve represents the "exact" spectrum, which serves as a reference for assessing the accuracy of the spectra from the shorter segments. One can see the "fluctuating" and progressive energy drop below 50 Hz for the FFT and digital filters. Note that the third-octave band centred at 50 Hz has a bandwidth of 11.5 Hz. For the light grey curve, $\Delta f \Delta t$ is equal to 11.5, which is consistent with a value of $K \approx 10$. Now, if the time waveform is padded with zeros, a dramatic improvement over the energy leakage is observed. Note that the dark grey spectral curves that correspond to the zero-padded 1-s segments are virtually identical in both images. A level difference can still be observed between the "exact" and "zero-padded" spectra. That difference is consistent with the uncertainty principle and with how low the product $\Delta t \Delta f$ can be before the accuracy becomes compromised.

In an FFT filtering approach, K spectral points per band are necessary to obtain a result that is minimally accurate in practice ($\Delta f / f_{res} > K$). This can be easily demonstrated by inserting the FFT spectral resolution $f_{res} = 1/\Delta t$ into (B.1). In a digital (FIR, IIR) filtering approach, for results to be accurate the window length must be approximately K times larger than the *minimum* GD of the filter within the passband. The GD within the passband varies with the filter design (see Figure B.6, left). Tests show that the GD is given by $A/\Delta f$, where the constant A takes values from 1 to 5 depending on the frequency and filter type.

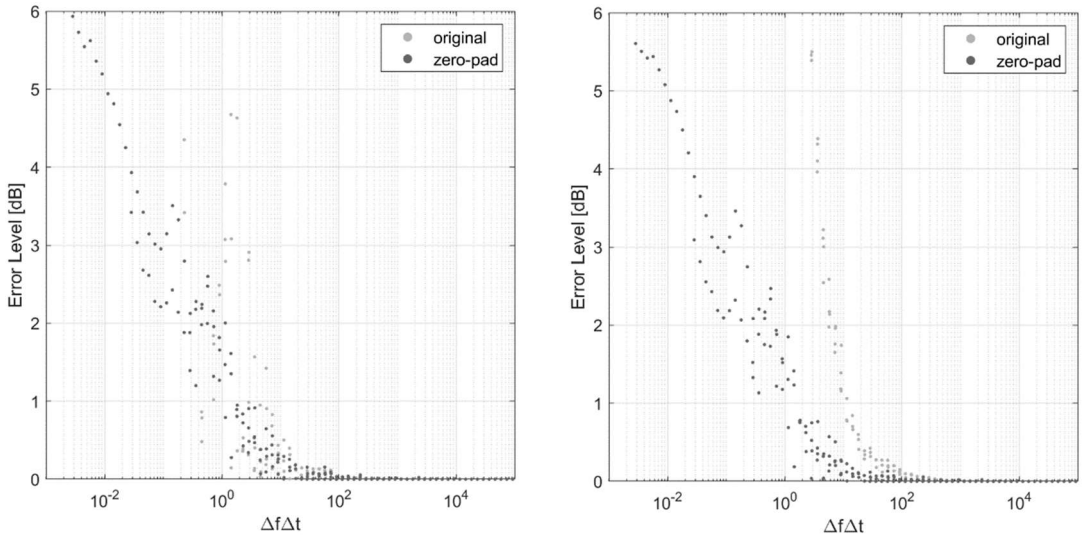

Figure B.5 Error level dependence with the bandwidth-duration product for the FFT filtering (left) and digital IIR filtering (right) methods. The *error level* represents the difference between the band level spectra of the WGN processes of duration 100 s ("exact" spectrum) and 1 s, *before* (grey) and *after* (dark grey) zero-padding to 100 s.

We argue that by using zero-padding, a single FFT point per bandwidth of the target band, or a window duration equal to the inverse of that bandwidth, are sufficient for the filtered energy to be reliably computed within 1 dB accuracy. In other words, padding with zeroes can help reduce the restrictive requirement of $K \approx 10$, observed in practice, to the more reasonable value of $K \approx 1$. Figure B.5 shows the error level ϵ as a function of the product $\Delta t \Delta f$, for the original (light grey) and zero-padded (dark grey) signal, and for FFT filtering (left) and digital IIR filtering (right). In both images, the error level for the original segment (without zero-padding) increases exponentially as $\Delta t \Delta f$ drops below 10. If we established 1 dB as the maximum error level that is considered acceptable, the minimum values of $\Delta t \Delta f$ for that error drop from 10 to 1 when zero-padding is applied. This demonstrates that, by using zero-padding, the inequality of the uncertainty principle is consistent with a value of $K \approx 1$, ten times lower than the value observed in practice when no zero-padding is used.

The error level ϵ in a digital filter is related to its GD. We know that the GD not only depends on the filter's bandwidth but also on its type. Figure B.6 shows the GD as a function of frequency for four types of filter (left), and the error level as a function of bandwidth-duration for the Butterworth and Chebyshev Type II filters, the ones with the lowest and highest passband-averaged GD values (right). One can see that the lower mean GD of the Chebyshev Type-II filter offers a moderate improvement in the error level by reducing the uncertainty constraint from $K = 10$ to $K = 8$.

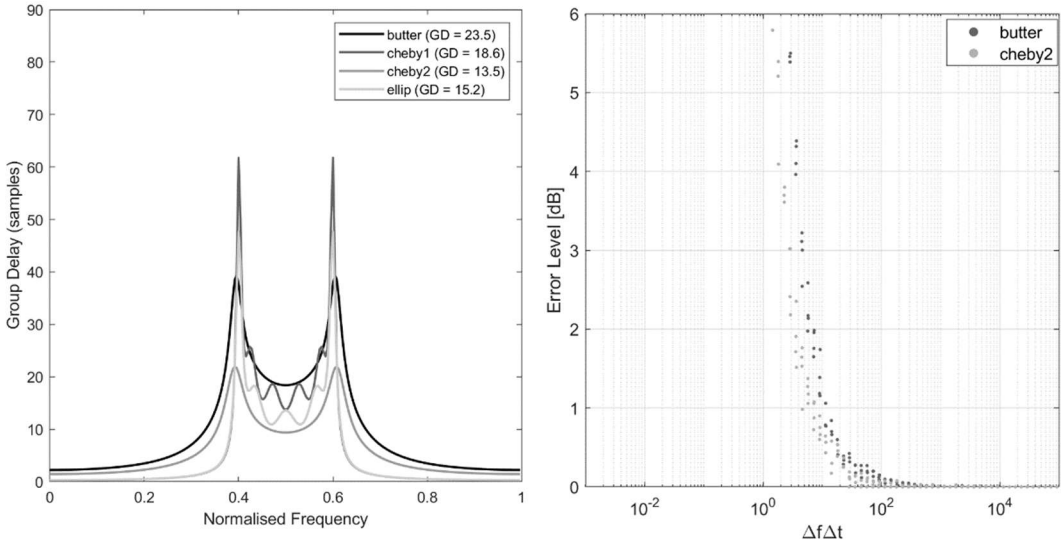

Figure B.6 Group delay with frequency for four filter types (Butterworth, Chebyshev Type I, Chebyshev Type II, and Elliptic) and a normalised bandwidth of 0.2 (left). Comparison of error level with bandwidth-duration factor between filter Butterworth and Chebyshev Type II filters.

One question still remains, namely the amount of zero-padding that is necessary to arrive at the best possible CPB spectrum, subject to the information available in the segment. Given a short signal vector, the optimal amount of zero-padding can be derived from the error level ϵ_0 between the spectral bands of the signal, zero-padded until no substantial change is observed in the spectrum (1e-3 dB or less), and the spectral bands of that same signal padded with varying amounts of zeroes. Here, we use the term ϵ_0 to distinguish this error level, which represents the level difference between the spectral band of two zero-padded signals, from the term ϵ, which refers to the level difference between the spectral bands of a long signal and its shorter zero-padded version. The error level ϵ_0 is expressed as a function of $\Delta t \Delta f$. Figure B.7 shows the result of these calculations when applied to a one-second WGN segment processed in third-octave bands from 1 Hz to 4 kHz using FFT filtering (left) and digital IIR filtering (right).

To obtain an expression for the dependence of ϵ_0 with $\Delta t \Delta f$, a line can be fitted to the data. The function $\log \epsilon_0 = A \log(\Delta t \Delta f) + B$ is used for the regression curve and fitted using non-linear optimisation. The optimisation solver is applied over a set of highly concentrated points to minimise the influence from the outliers. Then, the regression curve is displaced vertically by $\Delta\epsilon_0$ to include the majority of data points, with the intention of obtaining a conservative equation that guarantees a maximum error level for a given $\Delta t \Delta f$ constraint.

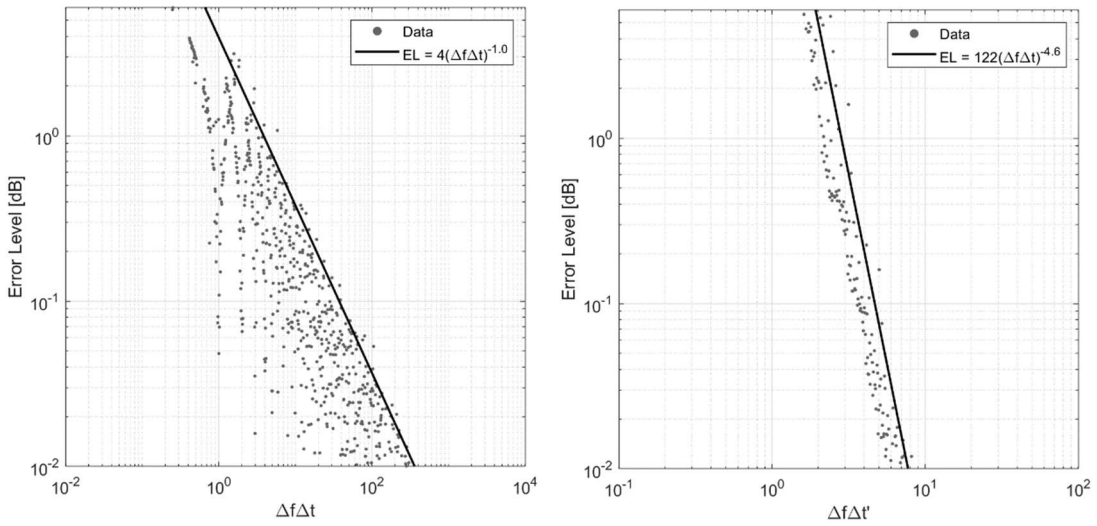

Figure B.7 Error level dependence with the bandwidth-duration product for the FFT filtering (left) and digital IIR filtering (right) methods. In this case, the *error level* represents the difference between the band level spectrum of the WGN process of duration 1 s padded with zeros up to 100 s, and the spectra of the same vector padded with a varying number of zeroes (1-100 s).

The regression equations $\epsilon_0\ (\Delta t \Delta f)$ for the FFT and digital IIR filtering methods are

$$\epsilon_0 \approx \frac{4}{\Delta t \Delta f} \qquad \text{for FFT filtering} \tag{B.2}$$

$$\epsilon_0 \approx \frac{120}{(\Delta t \Delta f)^{4.6}} \qquad \text{for digital filtering} \tag{B.3}$$

The two equations are valid for error levels in the range $0.01 < \epsilon_0 < 10$ dB. For a target error level $\epsilon_0 = 0.1$ dB, the duration of the zero-padded signal is given by

$$\Delta t \approx \frac{40}{\Delta f} \qquad \text{for FFT filtering} \tag{B.4}$$

$$\Delta t \approx \frac{5}{\Delta f} \qquad \text{for digital filtering} \tag{B.5}$$

It is interesting to see that the FFT filtering method requires an amount of zero-padding eight-times larger for the same error level. Note that the 0.1 dB target error ϵ_0 assumed for (B.4) and (B.5) only indicates that adding extra zeros will not produce a substantial change to the energy of the spectral bands. The overall error level ϵ within each band is eventually determined by the uncertainty principle. In general, and according to the results in Figure B.5, which suggest a value of $K = 1$ when zero-padding is used, the accuracy of bands with a bandwidth lower than $1/\Delta t$ will be severely compromised. Knowing that the bandwidth Δf of a CPB band of central frequency f_c can be expressed as $f_c \cdot b$, where $b = 2^{0.5/bpo} - 2^{-0.5/bpo}$ is the band factor and bpo is the number of

bands per octave (e.g., $b = 0.231$ for a third-octave band), the central frequency of the lowest band that can be accurately estimated after applying zero-padding is

$$f_c = \frac{1}{\Delta t \cdot b} \tag{B.6}$$

If no zero-padding was to be applied, the frequency of the lowest band affected by an overall error level ϵ of 1 dB or less would be approximately ten times larger ($K = 10$).

B.5 Digital Filtering (Single)

The 'Digital Filtering (Single)' package includes routines to design, represent, and apply a single digital filter. The package uses IIR filter objects of `digitalFilter` class designed with MATLAB's `designfilt.m`. Lowpass, highpass, bandpass, and arbmag ("arbitrary magnitude") filters are supported.

The package comprises three main functions: `digitalSingleFilterDesign.m` for designing the digital filter, `digitalSingleFilter.m` for filtering a signal using a previously-designed digital filter, and `digitalSingleFilterPlot.m` for representing the response of the digital filter and decimation filters (if applicable). The *filter design* only needs to be executed once but can be more computationally demanding than the *filtering* itself, which may need to be applied multiple times. For this reason, the two processes are implemented as separate functions.

`digitalSingleFilterDesign.m` designs a IIR filter for `digitalSingleFilter.m`. The function returns a structure `DigitalFilter` containing the filtering object and information about the decimator and digital filter. In order to improve the efficiency of the filtering process when applying a lowpass or bandpass filter to a large audio segment, a multi-stage decimation algorithm is implemented. The aim is to ensure the accuracy of the results through the design of simple and stable filters.

`digitalSingleFilter.m` filters an input signal using the `DigitalFilter` structure. The function filters the input vector by calling MATLAB's `filter.m` function. The user can choose to extract the *metrics* (RMS, exposure, peak, peak-to-peak) or the *filtered waveform*. The algorithm decimates by two the input signal multiple times until the Nyquist frequency is closest to the top cutoff frequency of the filter. The decimation improves the overall speed by reducing the number of samples to be filtered.

Zero-phase filtering is supported through `myfiltfilt.m` to compensate for the filter's group delay. `myfiltfilt.m` is a custom version of MATLAB's `filtfilt.m` introduced to avoid the strong leading and trailing oscillations that sometimes occur when filtering with the latter.

`digitalSingleFilter.m` also applies a suitable amount of zero-padding to the signal before filtering to avoid the *energy leakage* that occurs when the product between the

signal's duration and bandwidth is comparable to one or lower (see Sub-Section B.4). The zero-padding will increase the overall length of the filtered waveform, an effect that is not desirable if one is interested in the waveform itself. To get around this limitation, `digitalSingleFilter.m` uses MATLAB's `datawrap.m`. The function divides the filtered vector into segments of length equal to the original (unfiltered) vector, to then add them together. The function does a good job at conserving energy and phase.

`digitalSingleFilterPlot` represents the decimators (red) and filter (magenta) from the structure `DigitalFilter`. A marker on the response curves of the decimator and digital filter designates the half-power point (-3 dB).

B.6 FFT Filtering (Single)

The 'FFT Filtering (Single)' package calculates the RMS and exposure of an input signal within a specific frequency band. The algorithm uses the Fast Fourier Transform (FFT) to compute the power spectrum of the signal. The spectral samples are added within the specified band to obtain the corresponding power, from which the in-band RMS and exposure are calculated.

The package contains two main functions: `fftSingleFilterDesign.m` for designing the FFT filter, and `fftSingleFilter.m` for filtering a signal using a previously-designed FFT filter. For consistency with other filtering packages, the *design* and *filtering* functionalities are kept separate.

`fftSingleFilterDesign.m` designs a digital FFT filter for `fftSingleFilter.m`. The function returns a structure `FftFilter` containing the sampling rate, cutoff frequencies, and normalised cutoff frequencies. `fftSingleFilterDesign.m` only performs error control tasks on the input filter parameters, which it directly assigns to `FftFilter`.

`fftSingleFilter.m` filters an input signal using the `FftFilter` structure. The function is a wrapper of `fftFilter.m`. The functionality of `fftSingleFilter.m` is limited to error control operations and to the calculation of the exposure metric from the filtered RMS value returned by `fftFilter.m`. The latter applies a suitable amount of zero-padding to the signal before the FFT to avoid *energy leakage* (see Sub-Section B.4).

Unlike the conventional FIR-type filtering based on FFT and reconstruction methods (overlap-add, overlap-save), the 'FFT Filtering (Single)' package computes the metrics within a band directly from the spectrum rather than from the time-domain version of the filtered signal. When the metrics of a filtered signal (but not the filtered signal itself) are of interest, the approach used in this package is considerably more efficient.

B.7 FFT Filtering (Bank)

The 'FFT Filtering (Bank)' package calculates the RMS and exposure of an input signal for a set of standard frequency bands extending across a specific frequency range. The algorithm uses the Fast Fourier Transform (FFT) to compute the power spectrum of the signal. The *band RMS* is then calculated by adding the spectral samples within the corresponding frequency band. The *band exposure* is derived from the band RMS. The band metrics are computed directly from the spectrum rather than from the filtered waveform, making the process faster compared to methods based on waveform reconstruction (overlap-save, overlap-add).

The package comprises two main functions: `fftBankFilterDesign.m` for designing the FFT filter bank, and `fftBankFilter.m` for filtering a signal in bands using a FFT filter bank designed previously. For consistency with other filtering packages, the *design* and *filtering* functionalities are kept separate.

`fftBankFilterDesign.m` designs a digital FFT filter bank for `fftBankFilter.m`. The function returns a structure `FftFilterBank` containing the sampling rate, the nominal central frequencies, the standard and normalised central frequencies, and the standard and normalised cutoff frequencies. `fftBankFilterDesign.m` only performs error control tasks on the input parameters and assigns them to `FftFilterBank`.

`fftBankFilter.m` filters an input signal using the `FftFilterBank` structure. The function is a wrapper of `fftFilter.m`, discussed in the previous section. The functionality of `fftBankFilter.m` is limited to error control operations and to the calculation of the exposure metric from the filtered RMS values returned by `fftFilter.m`.

B.8 Digital Filtering (Bank) Package

The 'Digital Filtering (Bank)' package includes routines to design, represent, and apply a standard bank of digital filters. The package uses IIR filter objects of `digitalFilter` class designed with MATLAB's `designfilt.m`. Two types of filters are used: lowpass (LPF) for decimation, and bandpass (BPF) for computing the bands in the bank. The specifications of the bandpass filters meet the ANSI S1.11 standard (ANSI, 2004; BSI, 2014). As with all previous filtering packages, the filter design and filtering functions are kept separate to ensure computational efficiency.

The package comprises four main functions: `digitalBankFilterDesign.m` for designing the digital filter bank, `digitalBankFilter.m` for filtering a signal using a previously-designed digital filter bank, `ansiBankFilterCompliance.m` for determining the class (0, 1, 2) that the designed filter complies with, and `digitalBankFilterPlot.m` for representing the response of the digital filters in the bank and decimation filters.

`digitalBankFilterDesign.m` designs a bank of IIR filters for `digitalBankFilter.m`. The function returns a structure `AnsiFilterBank` containing the filtering objects and information about the decimators and digital filters. Similar to the 'Digital Filtering (Single)' package, multi-stage decimation is implemented to ensure accurate results through the design of simple and stable bandpass filters. This process is particularly useful for the analysis of oversampled low-frequency signals. The function designs a single decimation filter to be applied as many times as needed.

`digitalBankFilter.m` filters an input signal using the `AnsiFilterBank` structure. The function filters the input vector by calling MATLAB's `filter.m` function. The user has the option of extracting the *band metrics* (RMS, exposure, peak, peak-to-peak) or the *filtered waveform per band*. Similar to the `digitalSingleFilter.m` function in the 'Digital Filter (Single)' package, `digitalBankFilter.m` decimates by two the input signal multiple times until the Nyquist frequency is just above the top cutoff frequency of highest bandpass filter to be processed. Multiple sub-octave bands are computed in each decimation step. The muti-stage decimation solution adds further operations to the filtering process, but improves the overall speed by reducing the number of samples to be filtered.

The group delay from each filter is compensated by applying *zero-phase* filtering using `myfiltfilt.m`. This function replaces MATLAB's `filtfilt.m` to avoid the leading and trailing oscillations that may appear when filtering with the latter.

`digitalBankFilter.m` also applies a suitable amount of zero-padding to the signal before filtering to avoid the *energy leakage* that occurs when the product of between the signal's duration and bandwidth is lower than $K = 10$ (see Sub-Section B.4). As in `digitalSingleFilter.m`, MATLAB's function `datawrap.m` is used to compensate for the additional samples introduced by the zero padding, and return a filtered waveform of equal length as the original vector.

`ansiBankFilterCompliance.m` calculates the ANSI S1.11 class that the input bandpass filter is compliant with (0,1,2). The function returns -1 if the input filter does not meet the least restrictive specification (class `2`). `ansiBankFilterCompliance.m` is used by `digitalBankFilterDesign.m` to calculate the necessary filter order based on a target filter class.

`digitalSingleFilterPlot.m` displays the decimators (red) and filters (magenta) form a structure `AnsiFilterBank`. A marker on the response curves of the decimator and digital filter designates the half-power point (-3 dB).

B.9 Covariance Estimation

The 'Covariance Estimation' package contains eight different *shrinkage* algorithms for estimating covariance matrices. All the algorithms in the package belong to a group of

shrinkage estimators based on a linear combination of the sample covariance matrix (SCM) and a target matrix. Formally, the shrinkage estimator is expressed as $\mathbf{\Sigma} = \mathbf{S}(1-\delta) + \mathbf{F}\delta$, where $\mathbf{S}$ is the SCM, $\mathbf{F}$ is the target matrix, and δ is the shrinkage coefficient. This type of estimator was first proposed by Ledoit & Wolf (2003b).

The sample covariance matrix (SCM) is an optimal estimator when the number of *training* observations greatly exceeds their number of variables ($M \gg N$). However, when M is lower than or comparable to N, the SCM becomes a "noisy" estimator and alternative solutions are needed. An inaccurate covariance estimate will have a negative impact on the performance of a Neyman-Pearson estimator-correlator. In particular, it will affect the simulated probability density functions $p_{0,1}(T)$ and the computed test statistic $T(\mathbf{x})$, and how well the first characterises the statistical behaviour of the second.

Fortunately, the SCM is not the only covariance estimator available. The field of covariance estimation in high-dimensions has experienced great progress from the early 2000s, encouraged by advances in the fields of finance, bioinformatics, climate studies, risk management, and audio processing. Among the most popular covariance estimation methods are the banding, tapering, thresholding, maximum likelihood, sparse Gaussian graphical models, and shrinkage (Pourahmadi, 2013). Only the last method makes no structural assumptions about the covariance matrix and can be applied to a wide range of covariance structures with positive results. Here, we briefly describe the various shrinkage algorithms implemented as part of the 'Covariance Estimation' package.

The `covParam1.m` estimator uses a *scaled identity* target matrix. The *scaling factor* is equal to the mean variance of the SCM, that is, the mean of its diagonal entries. The shrinkage coefficient δ is efficiently calculated as the ratio p/c, where p is a simple function of the SCM and c is the squared Frobenious norm of the difference $\mathbf{S} - \mathbf{F}$. The method was proposed by Ledoit & Wolf (2004) and is one of the simplest and more widely used covariance estimators.

The `covRblw.m` and `covOas.m` estimators are approximations to the linear shrinkage estimator from Ledoit & Wolf (2004). Both methods were proposed by Chen et al (2010). The two consist in simple analytical expressions that improve over the previous algorithm when a Gaussian distribution is assumed. `covRblw.m` applies the *Rao-Blackwell* theorem to the method from Ledoit & Wolf (2004), and is therefore denoted as RBLW. `covOas.m`, referred to as oracle approximating shrinkage (OAS), is a closed form expression derived for the limit of an iterative approach which approximates the shrinkage estimator from Ledoit &Wolf (2004). Both algorithms use the scaled diagonal as target matrix. The expression for the shrinkage coefficient is identical in both cases, except for the value of two specific constants. Simulations carried out by Chen et al. demonstrate that the OAS approach can perform better in high-dimensions ($M \ll N$).

The `covStock.m` and `covCorr.m` are designed for the estimation of the covariance matrix of stock returns in portfolio optimisation and were proposed by Ledoit & Wolf (2003a; 2003b). These methods use specialised structured shrinkage targets suitable for

financial applications. The shrinkage coefficient δ is calculated as the ratio $(p-r)/c$. This is the same expression used in Ledoit & Wolf (2004), except for the r parameter. `covStock.m` and `covCorr.m` carry out different elaborate operations on the elements of the matrix of raw scores to obtain the parameter r.

The `covDiag.m` estimator is a variation of the estimator from Ledoit & Wolf (2004) that uses a *diagonal* target matrix and an extra parameter for the shrinkage coefficient. In the diagonal target matrix, the diagonal entries are identical to those in the SCM and the off-diagonal entries are all zero. The shrinkage coefficient δ is calculated as the ratio $(p-r)/c$. The parameter r is obtained as a mid-step from the calculation of p. The algorithm was developed by Olivier Ledoit[43].

The `covParam2.m` estimator is another variation of the estimator from Ledoit & Wolf (2004). In this case, the algorithm uses a two-parameter target matrix with constant values for the diagonal and off-diagonal entries. The shrinkage coefficient δ is calculated as the ratio $(p-r)/c$. The parameter r is obtained from *direct* and *recursive* operations on the matrix of raw scores. The computation of r can be demanding but it can be disabled, since its contribution to δ is generally small. The algorithm was developed by Olivier Ledoit[43].

The `covLooc.m` estimator employs a computationally efficient shrinkage estimator that uses Leave-One-Out Cross Validation (LOOCV) to compute the shrinkage coefficient based on Hoffbeck & Landgrebe's (1996) manipulation of the likelihood function. The method was proposed by Theiler (2012), who referred to this estimator as the Leave-One-Out Covariance or LOOC. We have updated the method to use a shrinkage target different from the proposed scaled diagonal. `covLooc.m` can take any shrinkage target. The function `targetCovariance.m` produces a shrinkage target equal to the optimal covariance matrix of a white Gaussian process treated with a specified filter. The *filtered target* outperforms the *scaled identity target* when used for the estimation of covariance matrices of filtered or band-limited signals.

B.10 Read GPS

The 'Read GPS' package reads the navigation data from one or more supported files. There are two file formats that are currently supported: (1) .gpstext, from SeicheSSV software and (2) .csv, extracted in table form from a PAMGuard's SQLite3 database.

`readgps.m` is the primary function in the package. It reads the GPS data from one or more supported files and returns it in a `GpsData` structure. `readgps.m` can read sentences of RMC, GGA and GSA type, as these are the ones that provide time, latitude and longitude. `GpsData` contains PC and UTC ticks, latitude, longitude, course and

[43] http://ledoit.net

speed, regardless of the file format and selected type of sentence. For .gpstext files, `GpsData` includes additional fields specific for the type of GPS sentence.

In .gpstext files, `readgps.m` calls the function `nmeaChecksum.m` . The function returns a logical value that is `true` if the checksum in the NMEA sentence is correct. Sentences with a `false` checksum are ignored.

B.11 Read AIS

The 'Read AIS' package reads the navigation data from one or more supported files containing decoded AIS data (.csv). The current version of the package does not support AIS files generated with SeicheSSV software (.aistext), since it does not include a functionality to decode AIS sentences. The supported .csv files are extracted in table form from PAMGuard's SQLite3 database.

`readais.m` is the main function in the package. It reads the AIS data from a .csv table extracted from a PAMGuard database and returns it in a `AisData` structure. Regardless of the file format, `AisData` contains PC and UTC ticks, latitude, longitude, course, speed, heading, and vessel specifications. AIS files include information from all vessels in the area. Data from specific vessels can be obtained by giving a list of MMSI numbers.

B.12 Read P190

The 'Read P190' package reads the navigation data from one or more .p190 files created by the navigation system of a seismic vessel. P1-90 is a format used in seismic surveys for logging position and field data. It comprises a *header*, for general information and format specs, and a *main body*, for real time data (line id, geographic/projected coordinates, time, and other parameters for every pulse emitted by the source).

`readp190.m` is the primary function in the package. It returns a `P190Data` structure containing the time, location, and operational information from the source(s) and the seismic vessel(s).

B.13 Distance & Bearing

The 'Distance and Bearing' package contains several routines for calculating (1) the distance and bearing between two points in the earth (*inverse* method) or (2) the final point (latitude, longitude) and bearing given a starting point, distance, and initial bearing (*direct* method).

The package contains six main functions: `pitagorasDirect.m`, `pitagoras.m`, `haversineDirect.m`, `haversine.m`, `vincentyDirect.m`, and `vincenty.m`. Each pair

corresponds to the *direct* and *inverse* versions of three different calculation methods: trigonometry, haversine, and Vincenty.

The *trigonometry* approach assumes a "regionally-flat" earth, thus simple trigonometry can be applied for calculating distances, bearings, latitudes, and longitudes. A constant distance of 111320 m per degree of latitude is used. This value comes from the assumption of a perfectly spherical earth (radius $R = 6378137$ m). The trigonometry method is the least accurate and should not be used for distances larger than 5 km. Since its implementation is particularly easy, it is a good option for spreadsheet formulas and casual calculations. With this method, the initial and final bearings are assumed identical.

The *Haversine* approach assumes a spherical model of the earth. The calculations involved are somewhat more complicated than those in the trigonometry method, but their analytical, non-recursive nature makes the Haversine solution a good option for a spreadsheet. The calculations are slightly more involved than in the trigonometry method, but both are comparable in terms of processing time. Unlike the latter, the Haversine solution computes both the initial and final bearings.

The *Vincenty* approach uses an elliptical model of the Earth. This is a recursive, numerical solution and as such is more involved than the previous two methods. It provides an accuracy of 1 mm when the correct *reference ellipsoid* is used. The function `refEllip.m` provides the required parameters from a wide range of reference ellipsoids (radius at the equator, radius at the poles, and flattening). This is the recommended approach due to its greater accuracy and reasonable computational speed.

Appendix C

The Simulated Annealing Algorithm

This appendix describes in detail the steps involved in the classic simulated annealing (SA) algorithm. The appendix is organised into six sections, each dedicated to a specific module of the SA algorithm developed in MATLAB for the calibration of the air gun array model (see Chapter 5). The sections, organised according to the logical order of the computations, discuss the following topics: (1) *Perturbation Algorithm*, a process designed to generate a new solution vector by applying a small disturbance to the previous vector, (2) *Cost Function*, a set of operations selected to estimate the cost associated with the "jump" between two solution vectors, (3) *Acceptance Criteria*, the conditions used to determine when a solution vector is accepted, (4) *Cooling Schedule*, the temperature decrement pattern that controls the speed at which the algorithm reaches the equilibrium state, (5) *Initial Conditions*, the starting temperature and solution vector, whose effect is critical for the convergence of the algorithm, and (6) *Stop Criteria*, a number of conditions that determine when the execution of the algorithm (or specific internal steps) must end. The *Initial Conditions* and *Stop Criteria* are described last, as these require a general understanding of the concepts introduced in earlier sections.

C.1 Perturbation Algorithm

The *design vector* or test solution $\mathbf{x}$ represents one of the many specific combinations of values for the target set of variables. The M-variable design vector is defined as $\mathbf{x} = \{x_1, x_2, \ldots x_M\}$. The "goodness" of each of these test solutions is evaluated with the *cost function* and accepted or rejected according to the *acceptance criteria*.

The valid set of test solutions is limited to the *solution space*, within which the optimal solution exists. The maximum and minimum values of the variables are the same for all design vectors. These boundaries must be clearly defined to guarantee that no calculations are made outside the solution space.

The *perturbation algorithm* or generator obtains a new design vector by adding a small distortion to an existing vector. The generator can be designed to alter one or multiple variables of the vector at each iteration, with the first option being the most common. The generator includes rules to produce only points within the solution space. There are several ways to perturb a design vector. The amount of perturbation or *jump size* can be constant or vary with temperature. Temperature-dependent generators produce larger jumps at higher temperatures to cover a wider area and small jumps at lower temperatures to limit the search to the neighbouring region. Using the temperature to determine the step size for the perturbation algorithm is down to the user.

C.2 Cost Function

The *cost function* or objective function $C(\mathbf{x})$ aims at providing a figure of merit for the *design vector* $\mathbf{x}$. The design vector that minimises the cost function ($\mathbf{x} = \text{argmin}(C(\mathbf{x}))$) represents the optimal solution. The cost function is problem-specific and choosing one requires full knowledge of the problem (Ledesma et al, 2008). In model optimisation, the mean square error between model and data is commonly used as the objective function. The output of the objective function (i.e., the cost) corresponds to the *energy* in a thermodynamic annealing process.

C.3 Acceptance Criteria

The *acceptance criteria* determines when a design vector is to be accepted as potential solution during the search process. The acceptance criteria for the standard simulated annealing algorithm discussed in this section is given by the Metropolis algorithm (Metropolis et al, 1953). The acceptance criteria work as follows:

1. If the perturbation algorithm generates a new design vector $\mathbf{x}_j$ that has a lower cost than the current design vector $\mathbf{x}_i$ ("good jump", $\Delta C < 0$), $\mathbf{x}_j$ is accepted as a potential solution. The term ΔC represents the *cost change*, which is the difference between the cost of the new and current design vectors ($\Delta C = C(\mathbf{x}_j) - C(\mathbf{x}_i)$).

$$P_a(\Delta C) = 1 \quad \text{for } \Delta C \leq 0 \tag{C. 1}$$

2. If the perturbation algorithm generates a design vector $\mathbf{x}_j$ that has a higher cost than the current one $\mathbf{x}_i$ ("bad jump", $\Delta C \geq 0$), the selection of the design vector is treated probabilistically. At this stage, a number r between 0 and 1 is randomly generated using a uniform distribution and compared to the Boltzmann-type probability of acceptance $P_a(\Delta C) = \exp(-\Delta C/T)$. If $P_a(\Delta C) \geq r$, the new design vector $\mathbf{x}_j$ is accepted, but if $P_a(\Delta C) < r$ the original vector $\mathbf{x}_i$ is used.

$$P_a(\Delta C) = \exp\left(\frac{-\Delta C}{T}\right) \quad \text{for } \Delta C > 0 \tag{C. 2}$$

The acceptance criterium for a "bad jump" depends on the temperature. The probabilistic treatment of high-cost transitions is what makes the simulated annealing different from other classical optimisation algorithms and is what provides the approach its *hill climbing* and *divide-and-conquer* features. Therefore, the evolution of the temperature or *cooling schedule* has an important effect on the performance of the algorithm.

C.4 Cooling Schedule

The *cooling schedule* defines the evolution of temperature with time and is regulated by the following parameters (Skiena, 2008): the initial temperature T_0, the temperature decrement function, and the number of iterations per temperature N_T.

There are four main types of temperature decrement functions (Ledesma et al, 2008): exponential, linear, cyclic and logarithmic. *The exponential cooling* spends a short time at high temperatures and long times at lower temperatures. The linear cooling spends the same time at all temperatures. In the *cyclical cooling* the temperature fluctuates, refining the quality of the solution at each cycle. The *logarithmic schedule*, also referred to as canonical, is an *optimal* cooling schedule proposed by Hajek (1988).

For iteration $n = 0,1,2 \ldots N$ and constant K, the canonical cooling schedule is given by (Johnson et al, 1989)

$$T_n = \frac{K}{\log(n+1)} \tag{C. 3}$$

The value of K is critical for the correct performance of the algorithm. Xinchao (2011) uses a value of $K = T_0$. Despite its proved optimality, the number of iterations that are needed to guarantee that a global optimum is found is generally very large, making it difficult to be applied in practice (Aarts et al, 1997; Johnson et al, 1989).

The *linear cooling schedule* decreases linearly with time and its slope is determined by the total number of temperature steps N.

$$T_n = \left(1 - \frac{n}{N}\right) T_0 \tag{C. 4}$$

In practice, the *geometric schedule* offers the best balance between accuracy and speed (Kirkpatrick et al, 1983; Johnson et al, 1989). This schedule is defined as a decaying exponential function of the form

$$T_{i+1} = \alpha^i T_i \tag{C. 5}$$

where α is the exponential constant ($\alpha = T_1/T_0$) with a value between 0 and 1 (typically $0.8 < \alpha < 0.99$).

Johnson et al. (1989) tested a cooling scheme based on a linear decrease of the probability of acceptance. This approach may not be of much use in practice, but its strong effect on the evolution of cost with time showed how critical the cooling scheme is for the correct performance of the algorithm.

The *number of iterations per temperature* N_T is typically in the range of $100 < N_T < 1000$ (Skiena, 2008). One of the observations from Johnson et al (1989) is that it may not be necessary to spend long times at high temperatures, where a large fraction of the transitions is accepted. One way of avoiding unnecessary iterations at high temperatures is to terminate the execution at a given temperature if the number of accepted transitions reaches a predetermined threshold qN_T, where q is the *cutoff factor*. Therefore, a temperature change will be forced if either N_T transitions have occurred or qN_T transitions have been accepted. As an alternative to the use of a cutoff factor, Johnson et al. applied a lower initial temperature T_0 with similar results.

Creating a suitable cooling schedule requires of some trial and error. The chance of success in finding the optimal solution can be improved by increasing the number of temperatures N, increasing the number of iterations per temperature N_T, choosing appropriate initial and final temperatures T_0 and T_N , and selecting a suitable temperature decrement function (Ledesma et al, 2008).

C.5 Initial Conditions

The *initial design vector* $\mathbf{x_0}$ and the *initial temperature* T_0 are the main initial conditions to be set before starting the simulated annealing process. An *initialisation function* is used to generate an appropriate start vector $\mathbf{x_0}$. This function is problem-dependent and tends to be simple (Ledesma et al., 2008). A random generator may be used to obtain an initial design vector $\mathbf{x_0}$, with the only restriction of keeping $\mathbf{x_0}$ within the limits of the solution space. A non-random initialisation function may be more effective in terms of accuracy and speed, but it requires some knowledge of the structure of the problem (Johnson et al., 1989).

The system starts with a maximum temperature that decreases according to a cooling scheme. This *initial temperature* T_0 has a strong effect on the ratio of accepted transitions, and needs to be carefully selected to ensure a good performance for the algorithm (Ben-Ameur, 2003). There are several methods available in the literature for computing the initial temperature. Kirkpatrick et al. (1983) proposed using the maximal cost difference between any two neighbouring solutions

$$T_0 = \Delta C_{max} \tag{C. 6}$$

Aarts et al (1997) recommended the use of the following expression for T_0

$$T_0 = K\sigma_\infty^2 \tag{C. 7}$$

where σ_∞^2 is the variance of the cost distribution when the temperature is ∞ and K is a constant typically ranging from 5 to 10. Note that in simulated annealing the condition for $T \to \infty$ is equivalent to accepting and evaluating every randomly generated solution. σ_∞^2 is estimated using a few of the randomly generated solutions (Ben-Ameur, 2003).

Johnson et al. (1989) proposed another approach that uses the mean cost increase of strictly positive transitions $\overline{\Delta C}$ and a suitable initial probability of acceptance P_{a0}

$$T_0 = -\frac{\overline{\Delta C}}{\ln P_{a0}} \tag{C. 8}$$

This same formula was independently suggested by Ledesma et al (2008), who chose a value of $P_{a0} = 0.8$. Johnson et al. (1989) used $P_{a0} = 0.4$ to reduce the time spent at higher temperatures, where most transitions are accepted. However, Johnson et al. showed that similar results can be obtained with a larger P_{a0} if the iterations within each temperature are stopped when a specific number of accepted transitions $N_T' = qN_T$ is exceeded. In their tests, the results from the configurations $[P_{a0} = 0.4, q = 1]$ and $[P_{a0} = 0.95, q = 0.125]$ were equivalent.

The simplest method to calculate T_0 is given by XinChao (2011), who replaces $\overline{\Delta C}$ in the equation from Johnson et al. (1989) with the difference between the highest and the minimum cost ΔC_{max} obtained from a number of randomly generated solutions ($T \to \infty$). An exact value of ΔC_{max} would require a high number of perturbations, but in practice a few are enough (10 in Xinchao's tests). Kirkpatrick et al (1983) suggested using the maximum cost difference from a number of randomly generated solutions ΔC_{max} directly as initial temperature. ($T_0 = \Delta C_{max}$).

A more intuitive method is described in Kirkpatrick et al. (1983) and summarised by Ben-Ameur (2003), where a number of transitions are computed for a large initial temperature and the ratio of accepted transitions is compared with a target probability of acceptance P_{a0}. If the acceptance ratio is less than P_{a0}, the temperature is increased. The process is repeated until the acceptance ratio exceeds P_{a0}.

More involved computations of the initial temperature are proposed by Ben-Ameur (2003) and Lee & Lee (2014).

C.6 Stop Criteria

The *stop criteria* are the conditions set to stop the search algorithm and provide a final solution, and to stop the search iterations within the current temperature value. Different algorithms use different stop criteria, but they normally include a condition of each type at least, in particular a minimum temperature T_{min} (*algorithm* stop criteria) and the maximum number of iterations within one temperature N_T (*temperature iteration* stop criteria). Below we describe the most common stop criteria for the algorithm and temperature step.

There are three criteria typically used to stop the search algorithm and provide a final solution to the system.

a) *Minimum temperature* T_{min}: the minimum temperature is selected based on the minimum number of temperature steps Q_{min} that are considered necessary to converge to the optimal solution. The number of temperature steps N is determined by the cooling constant α, the initial temperature T_0 and the minimum temperature T_{min} with

$$N = \ln(T_{min}/T_0)/\ln \alpha \tag{C. 9}$$

For a minimum of 50 steps, using typical values for the initial temperature and cooling constant ($T_0 = 1$, $\alpha = 0.8$), the minimum temperature must be of the order of $T_{min} = 10^{-5}$. More steps will require lower values of T_{min}. At each temperature, the simulation must proceed long enough for the system to reach a steady state (Kirkpatrick et al, 1983).

b) *Minimum cost function value* $C_{min}(s)$: this is the cost value considered sufficient to obtain a reliable solution. C_{min} is normally very low or unrestricted ($-\infty$).

c) *Maximum number of consecutive rejections* M_{rej}: a large number of rejections occur at the lowest temperatures where the algorithm converges to an optimal solution. A value of 1000 is recommended.

There are two common criteria for stopping the search within a given temperature.

a) *Maximum number of iterations per temperature step* N_T: using the same number of iterations within temperature steps increases the likelihood of finding the minimum (Ledesma et al, 2008). A value of 300 is recommended.

Maximum number of successes within one temperature $N_{T,suc}$: this is the number of accepted transitions within a temperature step and is equal to the product of N_T by a cutoff factor q. The values $q = 0.1$ or $N_{T,suc} = 20$ generally work well.