Latent Variable Regression and Applications to Planetary Seismic Instrumentation

Alexander Edward Stott

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in Electrical and Electronic Engineering of Imperial College London and the Diploma of Imperial College, October 2018
Abstract

The work presented in this thesis is framed by the concept of latent variables, a modern data analytics approach. A latent variable represents an extracted component from a dataset which is not directly measured.

The concept is first applied to combat the problem of ill-posed regression through the promising method of partial least squares (PLS). In this context the latent variables within a data matrix are extracted through an iterative algorithm based on cross-covariance as an optimisation criterion. This work first extends the PLS algorithm, using adaptive and recursive techniques, for online, non-stationary data applications. The standard PLS algorithm is further generalised for complex-, quaternion- and tensor-valued data. In doing so it is shown that the multidimensional algebras facilitate physically meaningful representations, demonstrated through smart-grid frequency estimation and image-classification tasks.

The second part of the thesis uses this knowledge to inform a performance analysis of the MEMS microseismometer implemented for the InSight mission to Mars. This is given in terms of the sensor’s intrinsic self-noise, the estimation of which is achieved from experimental data with a colocated instrument. The standard coherence and proposed delta noise estimators are analysed with respect to practical issues. The implementation of algorithms for the alignment, calibration and post-processing of the data then enabled a definitive self-noise estimate, validated from data acquired in ultra-quiet, deep-space environment.

A method for the decorrelation of the microseismometer’s output from its thermal response is proposed. To do so a novel sensor fusion approach based on the Kalman filter is developed for a full-band transfer-function correction, in contrast to the traditional ill-posed frequency division method. This algorithm was applied to experimental data which determined the thermal model coefficients while validating the sensor’s performance at tidal frequencies $1 \times 10^{-5}$ Hz and in extreme environments at $-65$ C.

This thesis, therefore, provides a definitive view of the latent variables perspective. This is achieved through the general algorithms developed for regression with multidimensional data and the bespoke application to seismic instrumentation.
Acknowledgements

I owe the opportunity to complete this thesis to many people and would like to leave them a word of thanks. First, I would like to thank my supervisors, Tom Pike and Danilo Mandic, for the chance to be involved which such projects and their help. To my Mum, Dad, Sister and Grandad (to whom this thesis is dedicated), I am deeply grateful for everything you have done for me, your encouragement and support. Yanita, you have been a huge source of inspiration and support. During my time in London I have made a series of fantastic friends including Mike, Loek, Sophie, Seb, Jon, Jonny, Connel, Yihan, Tony, Ryan, Mike, Fraser, Christophe, Anthony, Wilhelm, Charlie, Constantinos, John, Chris, Cat, Ilia, Tricia, Steph, Michael, Enrico and many others who have provided counterbalance and pushed me forwards. I would also like to thank many colleagues and friends from Imperial and other organisations who I have collaborated with and had many stimulating discussions. Particularly the InSight mission science team, NASA, CNES and the UK space agency. Finally, I would like to acknowledge EPSRC for providing financial support.
Dedication

This thesis is dedicated to the memory of Sydney Ashworth Stott 1926-2016
Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.
Nomenclature

$\text{Eig}_{\max}\{\cdot\}$ The largest eigenvector of the argument

$(\cdot)^*$ Conjugate

$(\cdot)^+$ Generalised inverse operator

$(\cdot)^H$ Hermitian transpose

$(\cdot)^T$ Transpose

$A$ A matrix

$a$ A vector

$C$ Field of complex numbers

$\mathbb{H}$ Field of quaternion numbers

$\otimes$ Kronecker product

$\mathbb{R}$ Field of real numbers

$\mathcal{A}$ A tensor

$\text{Tr}\{\cdot\}$ Trace operator

$a$ A scalar

$E[\cdot]$ Statistical expectation operator

$i, j, k$ Imaginary numbers
## Contents

Abstract 1

Acknowledgements 3

Copyright declaration 7

1 Introduction 21
   1.1 Motivation and perspective 21
   1.2 Problem formulation 23
   1.3 Associated publications and reports 26
   1.4 Thesis contributions and organisation 26
   1.5 Statement of Originality 28

2 Background theory 29
   2.1 Outline 29
   2.2 Linear regression 29
      2.2.1 The generalised inverse 30
      2.2.2 The fundamental theorem of linear algebra and the SVD 31
      2.2.3 The SVD, generalised inverses and ill-posed regression 33
      2.2.4 The calculation of eigenvectors 34
      2.2.5 Statistics and probability distributions 35
   2.3 Multidimensional Signal Processing 35
      2.3.1 Complex-valued signal processing 36
      2.3.2 Quaternion-valued signal processing 39
      2.3.3 Tensor Notation and Preliminaries 42
   2.4 Frequency domain analysis 46

I Part A 49

3 Latent Variable Regression 50
   3.1 Introduction to latent variables 50
      3.1.1 Latent Variables in a matrix 51
      3.1.2 Principal component analysis: covariance matrix diagonalisation 53
   3.2 Regression with latent variables 53
   3.3 The method of partial least squares 54
      3.3.1 The NIPALS algorithm for PLS 55
      3.3.2 Other methods of PLS 57
      3.3.3 Analysis of the PLS result 58
      3.3.4 The core components of a PLS algorithm 60
      3.3.5 Example: PLS decomposition of circulant matrices 60
   3.4 Total Least Squares 62

3.5 Optimality of latent variable regression .............................. 63
3.5.1 The Concept of Forwards and Backwards Regression ............ 64
3.6 Online PLS .......................................................... 66
3.6.1 An Online PLS Algorithm ....................................... 67
3.6.2 Adaptive Estimation of the OL-PLS Variables ................... 68
3.6.3 Simulations and Analysis ....................................... 70
3.7 Discussion of practical implementations of OLS, PLS and OL-PLS ... 71
3.8 Conclusion .......................................................... 72

4 Widely Linear Complex Partial Least Squares 74
4.1 Introduction ......................................................... 74
4.1.1 What should a complex-valued partial least squares be? ........ 75
4.2 A widely linear complex-valued PLS algorithm .................... 75
4.2.1 Alternative WL-CPLS algorithms .............................. 79
4.3 Analysis of the WL-CPLS result .................................... 80
4.3.1 The WL-CPLS with respect to the PLS core-components ....... 80
4.3.2 The augmented matrix generalised inverse: WL-CPLS for a regularised widely linear estimation ................... 81
4.3.3 WL-CPLS as a covariance matrix diagonalisation transform ... 83
4.3.4 The convergence of the WL-CPLS algorithm .................... 87
4.4 Simulation results .................................................. 89
4.4.1 Prediction MSE for a varying SNR in $Y$ ....................... 89
4.4.2 Prediction MSE for a varying number of WL-CPLS components . 90
4.4.3 Prediction MSE for a varying SNR in $X$ and $Y$ ............... 91
4.5 Distributed frequency estimation in power grids .................. 92
4.5.1 Frequency estimation: Problem specification .................. 92
4.5.2 Balanced multiple node case .................................. 93
4.5.3 Unbalanced multiple node case ................................ 94
4.5.4 Exploiting redundancy with a WL-CPLS solution .............. 94
4.5.5 Simulation comparison of estimators ........................... 95
4.6 Data properties and the WL-CPLS performance .................... 96
4.7 Conclusion .......................................................... 96

5 Quaternion and Tensor Partial Least Squares 98
5.1 Introduction ........................................................ 98
5.2 The Quaternion NIPALS algorithm ................................ 99
5.2.1 Quaternion covariance matrix diagonalisation .................. 101
5.2.2 The convergence of the WL-QPLS algorithm ................... 102
5.2.3 The orthogonality of the WL-QPLS model residuals ............ 103
5.3 Multilinear PLS-regression algorithms ............................ 104
5.3.1 The HOPLS algorithm ......................................... 104
5.3.2 A NIPALS algorithm for tensor-variate PLS regression ....... 106
5.4 Comparison of the quaternion and Tensor PLS algorithms ...... 111
5.4.1 Core components of the multidimensional NIPALS PLS algorithms 111
5.4.2 The isomorphism between quaternions and tensors ............ 113
5.4.3 WL-QPLS as a tensor-variate PLS ............................. 113
5.4.4 Comparison of the WL-QPLS and HONIPALS cross-covariance structures ................................................. 115
5.4.5 Comparison of HONIPALS and GHOPLS cross-covariance structures 116
5.5 Application of multidimensional PLS algorithms for image classification 118
5.6 Data properties and the MD-PLS algorithms performance ........ 120
5.7 Conclusion .......................................................... 121
II Part B

6 Self-Noise determination of a microseismometer for planetary science

6.1 Introduction .............................................. 123
6.2 The microseismometer outputs and response .............................................. 124
6.2.1 Transfer-function correction .............................................. 126
6.3 Seismometer Self-Noise Determination .............................................. 126
6.3.1 Seismometer testing literature review .............................................. 127
6.3.2 Coherence noise testing .............................................. 129
6.3.3 Delta noise testing .............................................. 130
6.3.4 Bias and Variance of the Noise Estimators .............................................. 131
6.4 Simulation Analysis of Noise Testing Methods .............................................. 132
6.4.1 The effects of transfer-function correction .............................................. 132
6.4.2 Error analysis: estimate performance and prediction .............................................. 135
6.5 Estimation of the microseismometer noise: Theoretical perspectives .............................................. 138
6.5.1 Experimental setup and background signal considerations .............................................. 138
6.5.2 Noise Model of the microseismometer .............................................. 141
6.5.3 Synthetic simulation of the noise model .............................................. 141
6.5.4 Practical self-noise determination: the accuracy limits for a given site .............................................. 143
6.5.5 Expected results and interpretation of self-noise determination results .............................................. 143
6.6 Noise campaigns and data acquisition .............................................. 145
6.7 The Self-Noise analysis of the microseismometer: algorithmic developments .............................................. 147
6.7.1 Geometric Alignment .............................................. 147
6.7.2 Transfer-Function Correction .............................................. 150
6.7.3 Spectrogram and ranked noise estimation .............................................. 156
6.8 The self-noise analysis of the microseismometer: experimental results .............................................. 160
6.8.1 Site noise floor .............................................. 160
6.8.2 Self-noise determination of the flight model units .............................................. 162
6.8.3 High frequency noise: QM BFO .............................................. 162
6.8.4 Low frequency noise: QM Vert 2017 Oxford .............................................. 163
6.8.5 Deep space observatory .............................................. 163
6.9 Conclusions: Summary of microseismometer self-noise analysis .............................................. 175
6.9.1 Future work - noise testing development .............................................. 178

7 Full-Band Signal Extraction: Temperature Decorrelation

7.1 Introduction .............................................. 180
7.2 The microseismometer thermal model .............................................. 182
7.3 The Kalman Filter .............................................. 183
7.4 Full-Band Sensor Fusion Algorithms .............................................. 185
7.4.1 Sensor fusion for transfer-function correction .............................................. 185
7.4.2 Full-band temperature decorrelation .............................................. 186
7.4.3 Implementation of the K-TF and K-TF+T methods .............................................. 187
7.5 Experimental Results .............................................. 189
7.5.1 Experimental setup .............................................. 189
7.5.2 Mass position and velocity output fusion .............................................. 192
7.5.3 Validation of the microseismometer thermal compensation and temperature decorrelation .............................................. 192
7.6 Conclusion .............................................. 199

8 Conclusion

8.1 Summary of results .............................................. 201
8.1.1 Part A: LV regression with multidimensional methods .............................................. 201
8.1.2 Part B: The performance analysis of the NASA InSight microseis-
meter ................................................................. 203
8.2 Discussion of method approaches .................................. 204
8.3 Future Work .......................................................... 208
8.4 Summary and closing remarks ...................................... 208

Bibliography ......................................................... 209
List of Tables

2.1 Quaternion multiplication ............................................. 39
4.1 Comparison of the real-valued NIPALS and WL-CPLS algorithms .... 80
5.1 Comparison of two-way PLS and HOPLS algorithms ................. 106
5.2 Comparison of the multidimensional PLS algorithms ............... 112
5.3 Coffee image classification ......................................... 119
6.1 Noise method simulation parameters ................................. 133
6.2 Noise testing datasets used .......................................... 146
6.3 Flight model nominal transfer-functions ............................. 156
7.1 Datasets used ......................................................... 191
7.2 Thermal Model Parameters and Fit Quality .......................... 197
8.1 Table of algorithms used in thesis and potential applications .... 207
List of Figures

1.1 Latent variable representation of signals recorded from SEIS. ........................................ 23
1.2 Thesis structure. .............................................................................................................. 28
2.1 The four fundamental subspaces, $\mathcal{R}(\mathbf{A})$ and $\mathcal{N}(\mathbf{A}^T)$ exist in $\mathbb{R}^n$ and $\mathcal{R}(\mathbf{A}^T)$ and $\mathcal{N}(\mathbf{A})$ exist in $\mathbb{R}^m$. .......................................................... 32
2.2 Demonstration of the power method iteration ................................................................. 34
2.3 The mode-n unfoldings of a third order tensor [1]. .......................................................... 43
2.4 Tucker decomposition of a tensor. .................................................................................... 45
2.5 SVD of a matrix. .............................................................................................................. 45
3.1 The matrix $\mathbf{X}$ is created from $N$ samples/realisations of an $m$-variate process generated from some distribution $\mathcal{D}(\theta)$. The variable, $\mathbf{x}$, can be seen as a sliding window through the matrix $\mathbf{X}$. ................................................. 51
3.2 The matrix factorisation (3.1) transforms the coordinate basis from the assumed identity matrix, $\mathbf{I}$, to that given by the matrix $\mathbf{P}$. This can be viewed as a sum of rank-1 outer products. Note that the $\mathbf{t}_i \in \mathbb{R}^{N \times 1}$ presented here is not the entity but a realisation of $N$ samples of one component of the new entity $\mathbf{t} \in \mathbb{R}^{1 \times r}$. ................................................................. 52
3.3 PLS for de-noising sinusoids. .......................................................................................... 62
3.4 OL-PLS error in predicting $\mathbf{Y}$. .................................................................................... 71
4.1 Geometric view of WL-CPLS regression. .......................................................... 78
4.2 Proportion of the variance (in %) explained by each component in the SUT and WL-CPLS transform, $VE$ in (4.21). ................................................................. 85
4.3 Prediction MSE for $\mathbf{Y}$ (as a percentage of total variance) against the SNR in $\mathbf{Y}$ ........................................................................................................ 90
4.4 Performance, in terms of MSE, of the prediction of $\mathbf{Y}$ (as a percentage of total variance) for a varying number of WL-CPLS components, in both training and test data. ................................................................. 90
4.5 Prediction MSE against the noise level in $\mathbf{Y}$ (as a percentage of total variance), with noise also present in $\mathbf{X}$. ........................................................................................................ 91
4.6 Prediction obtained from the Moore-Penrose pseudoinverse against the noise level in $\mathbf{Y}$ (as a percentage of total variance), with noise also present in $\mathbf{X}$. ................................................................. 92
4.7 Comparison of MSE for an unbalanced 3-phase power grid frequency estimator, for a range of SNR. ........................................................................................................ 95
5.1 The MSEs of widely linear and tensor regression estimates calculated for a range of SNRs and different numbers of PLS components, as a ratio of the power in the dependent variables. ................................................................. 115
5.2 A selection of the satellite images taken of Brazilian Coffee scenes dataset and classified as either containing a coffee plantation or a non-coffee zone. The intraclass variance is known to be high for this dataset as the images cover different regions and so terrains and crop types. The multidimensional filters used in this application offer an alternative approach to segment such information. .................................................. 120

6.1 The schematic shows (in cross-section) the MEMS sensor, connectivity to the functional blocks of the electronics and the packaging of the sensor in its enclosure. \( RT_{enc} \) and \( RT_{die} \) indicate the locations of the resistance thermometers. ................................................................. 125

6.2 The top two plots show the frequency response of the velocity channel \( (V/m/s) \) and mass position channel \( (V/m/s^2) \). The bottom plot shows both the responses to acceleration \( (V/m/s^2) \). .................................................. 125

6.3 Test scenario .......................................................... 127

6.4 Simulation to show the residual correlation in a CPSD between two uncorrelated sources. Calculated with the Welch method averaging over two Hanning windows with 50% overlap, where the data are assumed to be sampled at \( f_s = 100 \text{ Hz} \). ...................................................... 128

6.5 Simulation 1: Perfect correction. PSDs calculated using the Welch method averaging over two Hanning windows with sampling rate defined as \( f_s = 1 \text{ Hz} \). ...................................................... 133

6.6 Simulation 2: Incorrect estimated gain correction. PSDs calculated using the Welch method averaging over two Hanning windows with sampling rate defined as \( f_s = 1 \text{ Hz} \). ...................................................... 134

6.7 Simulation 3: Incorrect estimated corner frequency. PSDs calculated using the Welch method averaging over two Hanning windows with sampling rate defined as \( f_s = 1 \text{ Hz} \). ...................................................... 135

6.8 Simulation 4: Incorrect estimated Damping. PSDs calculated using the Welch method averaging over two Hanning windows with sampling rate defined as \( f_s = 1 \text{ Hz} \). ...................................................... 135

6.9 Mean noise estimation bias with gain change. .................................................. 136

6.10 The noise determination accuracy as function of SNR. ........................................ 138

6.11 Flow-chart of the signals in the self-noise determination process. ......................... 139

6.12 Demonstration of source coupling for coincidence testing. The parallel component in the ground dynamics can be rejected by the noise determination algorithms, however, the orthogonal, non-linear and non-dynamic contributions are incoherent excess injections. .................................................. 140

6.13 Seismic noise testing and incoherent source contributions. .................................. 141

6.14 The noise model of the microseismometer .......................................................... 142

6.15 Accuracy of self-noise determination depending on frequency ............................ 144

6.16 The alignment angle determination as function of input bandwidth \( f_{\text{bottom}} \leq f \leq f_{\text{top}} \) for Algorithm 18 for BFO dataset. .................................................. 149

6.17 The alignment angle determination as function of input bandwidth \( f_{\text{bottom}} \leq f \leq f_{\text{top}} \) for Algorithm 18 for SP3 CNES dataset. .................................................. 150

6.18 Microseismometer self-noise estimation on the QM BFO dataset for a range of alignment errors, \( \Delta \theta \), in the horizontal plane. The noise determination was calculated using the Welch periodogram averaged over eight Hanning windows with 50% overlap. The change in \( \Delta \theta_{NE} \) is shown to add in a component of the microseismic peak, showing the injection of the ambient into the noise curve. .................................................. 151
6.19 Microseismometer self-noise estimation on the QM BFO dataset for a range of alignment errors, $\Delta \theta$, in the vertical plane. The noise determination was calculated using the Welch periodogram averaged over eight Hanning windows with 50% overlap. .......................................................... 152

6.20 Microseismometer self-noise estimation on the SP3 CNES dataset for a range of alignment errors, $\Delta \theta$, in the horizontal plane. The noise determination was calculated using the Welch periodogram averaged over eight Hanning windows with 50% overlap. .......................................................... 153

6.21 Microseismometer self-noise estimation on the SP3 CNES dataset for a range of alignment errors, $\Delta \theta$, in the vertical plane. The noise determination was calculated using the Welch periodogram averaged over eight Hanning windows with 50% overlap. .......................................................... 154

6.22 Microseismometer self-noise estimation on the SP1 CNES dataset for a range of alignment errors, $\Delta \theta$, in the horizontal plane. The noise determination was calculated using the Welch periodogram averaged over eight Hanning windows with 50% overlap. .......................................................... 155

6.23 Microseismometer self-noise estimation on the SP1 CNES dataset for a range of alignment errors, $\Delta \theta$, in the vertical plane. The noise determination was calculated using the Welch periodogram averaged over eight Hanning windows with 50% overlap. .......................................................... 156

6.24 Transfer-function between corrected reference and corrected microseismometer signals for SP3 CNES dataset. .......................................................... 157

6.25 Transfer-function between corrected reference and corrected microseismometer signals for QM BFO dataset. .......................................................... 157

6.26 Microseismometer calibration waveform and corrected response in velocity. 158

6.27 The transfer-function estimate to the calibration waveform for SP2. ........ 158

6.28 The transfer-function estimate to the calibration waveform for SP3. .... 159

6.29 The two sensor noise test setup. .......................................................... 161

6.30 Prediction of the bias of self-noise determination in SP3 CNES dataset. .. 162

6.31 Prediction of the bias of self-noise determination in QM BFO dataset. .... 162

6.32 SP1 CNES dataset: The microseismometer rPSD, reference rPSD, reference noise along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%. .................. 163

6.33 SP1 CNES dataset: Spectrograms of the microseismometer signal, reference noise and microseismometer noise. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometer, reference and the delta noise is low-pass filtered with a corner frequency at 1 Hz. .................. 164

6.34 SP1 CNES dataset: The ranked-noise estimate (6.24) for $i = 1, 2, \ldots, 32$ of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth 0.01 – 1 Hz. Calculated from the 800s period STFT (400s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%. .................. 165

6.35 SP2 CNES dataset: The microseismometer rPSD, reference rPSD, the reference noise along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%. .................. 165
6.36 SP2 CNES dataset: Spectrograms of the microseismometer signal, reference noise and microseismometer noise. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometer, reference and the delta noise is low-pass filtered with a corner frequency at 1 Hz.

6.37 SP2 CNES dataset: The ranked-noise estimate (6.24) for $i = 1, 2, \ldots, 32$ of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth $0.01 - 1$ Hz. Calculated from the 800s period STFT (400s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%.

6.38 SP3 CNES dataset: The microseismometer rPSD, reference rPSD, the reference noise along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%.

6.39 SP3 CNES dataset: Spectrograms of the microseismometer signal, reference noise and microseismometer noise. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometer, reference and the delta noise is low-pass filtered with a corner frequency at 1 Hz.

6.40 SP3 CNES dataset: The ranked-noise estimate (6.24) for $i = 1, 2, \ldots, 32$ of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth $0.01 - 1$ Hz. Calculated from the 800s period STFT (400s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%.

6.41 QM BFO dataset: The microseismometer rPSD, reference rPSD, the reference noise along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%.

6.42 QM BFO dataset: Spectrograms of the microseismometer signal, reference noise and microseismometer noise. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometer, reference and the delta noise is low-pass filtered with a corner frequency at 1 Hz.

6.43 QM BFO dataset: The ranked-noise estimate (6.24) for $i = 1, 2, \ldots, 32$ of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth $0.01 - 1$ Hz. Calculated from the 800s period STFT (400s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%.

6.44 QM Vert 2017 dataset: The microseismometer rPSD, reference rPSD along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%.

6.45 SP2/3 Space event dataset: Spectrograms of the microseismometer signals. Calculated for 100s period STFT at 10s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometers is low-pass filtered with a corner frequency at 1 Hz.

6.46 SP2/3 Space event dataset: The ranked-noise estimate (6.24) for $i = 1, 2, \ldots, 32$ of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth $0.01 - 1$ Hz. Calculated from the 200s period STFT (20s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%.
6.47 SP2/3 Space long period dataset: self-noise rPSD, calculated using the Welch method averaging over two Hanning windows overlapping by 50%, of each microseismometer signal. ................................. 173

6.48 SP2/3 Space long period dataset: Spectrograms of the microseismometer signals. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometers is low-pass filtered with a corner frequency at 0.5 Hz. ............................................. 174

6.49 A summary of the microseismometer self-noise determinations on terrestrial datasets. ......................................................... 177

6.50 A summary of the self-noise determination from the SP2 FM sensor in cruise. 177

6.51 A summary of the self-noise determination from the SP3 FM sensor in cruise. 177

7.1 PSD of the thermometer temperatures in each dataset and their respective low-pass filter stop-band frequencies. The two FM datasets have a lower acquisition noise floor. ................................. 190

7.2 K-TF synthesised microseismometer acceleration, $a_{\mu S}$, time series and rPSD compared to $a_{VEL}$ and $a_{POS}$ for QM Vert 2017 dataset. ................................. 192

7.3 QM Vert 2017. First pane: Self-noise determination from the $a_{VEL}$ and temperature decorrelated, $a_{\mu S} - a_{\mu ST}$, signals. Second pane: Improvement ratio between self-noise before and after temperature decorrelation. Third Pane: acceleration time series for the reference, $a_{REF}$, and decorrelated microseismometer acceleration with a comparison of the K-TF+T synthesised acceleration and thermal model fit inset. Fourth pane: $T_{die}$ and $T_{enc}$ signals, Fifth pane: $\Delta T$ signal. ................................. 193

7.4 FM Vert. First pane: rPSD of the $a_{VEL}$, and temperature decorrelated, $a_{\mu S} - a_{\mu ST}$, signals. Second pane: Improvement ratio between rPSDs before and after temperature decorrelation. Third Pane: time series for the decorrelated microseismometer acceleration compared to the K-TF+T synthesised acceleration and thermal model fit with the K-TF+T synthesised acceleration and mass position inset. Fourth pane: $T_{die}$ and $T_{enc}$ signals. Fifth pane: $\Delta T$ signal. ................................. 195

7.5 FM Horiz. First pane: rPSD of the $a_{VEL}$, and temperature decorrelated, $a_{\mu S} - a_{\mu ST}$, signals. Second pane: Improvement ratio between rPSDs before and after temperature decorrelation. Third Pane: time series for the decorrelated microseismometer acceleration compared to the K-TF+T synthesised acceleration and thermal model fit with the K-TF+T synthesised acceleration and mass position inset. Fourth pane: $T_{die}$ and $T_{enc}$ signals. Fifth pane: $\Delta T$ signal. ................................. 196

7.6 QM Vert 2016. First pane: Self-noise determination from the $a_{VEL}$ and temperature decorrelated, $a_{\mu S} - a_{\mu ST}$, signals. Second pane: Improvement ratio between self-noise before and after temperature decorrelation. Third Pane: acceleration time series for the reference, $a_{REF}$, and decorrelated microseismometer acceleration with a comparison of the K-TF+T synthesised acceleration and thermal model fit inset. Fourth pane: $T_{die}$ and $T_{enc}$ signals. Fifth pane: $\Delta T$ signal. ................................. 198
7.7 QM Vert Cold. First pane: Self-noise determination from the $a_{VEL}$ and temperature decorrelated, $a_{\mu S} - a_{\mu ST}$, signals. Second pane: Improvement ratio between self-noise before and after temperature decorrelation. Third Pane: acceleration time series for the K-TF+T synthesised acceleration, thermal model fit and temperature decorrelated signals with the reference, $a_{REF}$, and the decorrelated microseismometer acceleration inset. Fourth pane: $T_{die}$ and $T_{enc}$ signals. Fifth pane: $\Delta T$ signal. . . . . . . . . . . . . . . . . 200
Chapter 1

Introduction

Accurate and minute measurement seems to the non-scientific imagination, a less lofty and dignified work than looking for something new. But nearly all the grandest discoveries of science have been but the rewards of accurate measurement and patient long-continued labour in the minute sifting of numerical results.

Lord Kelvin

1.1 Motivation and perspective

Human nature drives us to attempt to answer a fundamental question: how special are we? From a geophysical perspective this means how special is our planet Earth? Much scientific endeavor has focused on delineating the processes which led its creation and shaped its features, e.g. plate tectonics, the magnetic field and the atmosphere. In this vein, the internal structure of the Earth has been determined over the last century or so [2]. A remaining line of inquiry is then to understand what caused the Earth to form and evolve this way? To truly delve into this question, we must place our knowledge of Earth in the context of other planets. To provide such information, the NASA InSight mission landed on Mars on the 26th November 2018, following an entry into the atmosphere, a descent to the surface and landing known collectively as the “seven minutes of terror”. The mission is for the Interior Exploration of Mars, using Seismic Investigations, Geodesy and Heat Transport (InSight) and aims, for the first time, to resolve the inner structure and geophysical processes at play on another planet [3]. In turn, this allows us to compare the Earth and hence develop Earth science theories and our understanding of planetary formation/differentiation [4].

In order to begin to answer such questions, a series of objectives for the NASA InSight mission have been derived as [2]:

1. Determine the size, composition and physical state of the core,
2. Determine the thickness and structure of the crust,
3. Determine the composition and structure of the mantle,
4. Determine the thermal state of the interior,
5. Measure the rate and distribution of internal seismic activity,
6. Measure the rate of impacts on the surface.

These can be deduced from the study of seismic waves and so the mission must obtain such data on Mars. To this end, the InSight mission carries the Seismic Experiment for Interior Structure (SEIS) payload [2]. This contains a Very Broad Band (VBB) seismometer and our MEMS microseismometer as a Short Period (SP) sensor [5]. The experiment will detect the seismic signals on Mars made by marsquakes and impacts. Resolving the nature of these recorded waves then enables the properties of the material they travelled through to be established, and thus, study the internal structure of Mars [6].

In order to successfully implement the tools of seismology on recorded data, the protagonist must be sure that it contains no extraneous components [6] [7]. As a result, the performance of the sensor must be understood for implementation in the harsh, extreme Martian environment. This entails the determination of the instrument self-noise and the sensor’s response to external environmental sources [8] [9]. This is of particular relevance for the InSight mission as it is deployed in a harsh environment with minimal resources available to improve the installation. The details of this problem are made clear through consideration of the Viking mission to Mars in 1976 [10]. This mission carried the first seismometer experiment to another planet but the data return was unfortunately found to show minimal seismic signals that could be used to gain information about the planet interior [10] [11]. This was because the installation was on top of a lander and consequently coupled extremely well to the wind forcing. The impact of this is further shown from comparison to the more successful Apollo experiments from 1969-77, where a network of seismometers was deployed by astronauts on the Moon [12]. The efficacy of this data was much greater than that from the Viking mission both due to the increased deployment scope and the lack of atmospheric coupling. As a result, the SEIS experiment has been made the prime focus of this discovery class mission and, hence, the sensors are to be placed on the ground under a wind and thermal shield (WTS) [2]. Despite such efforts, the environmental injections are still expected to play a part [7]. As a result, much of the seismic information is hidden from view, either not directly observable or obscured by other processes. It is this problem which motivates the methodologies used and developed in this thesis.

Figure 1.1 shows that the recorded signal is a mixture of seismic, instrument and environmental contributions, that is, the original signal is modelled as a sum of several contributions. These constituent components of the observed data are termed latent variables. For application to the problem posed by the InSight mission, the latent variable approach is a data analysis methodology. In this sense, the separation of the latent components would be performed through careful integration of prior knowledge and physical understanding into the processing. Such an approach is encouraged by the inclusion of the
Auxiliary Payload Sensor Suite (APSS) which will record pressure, wind and temperature. This means that external contributions can be correlated to the seismic data on top of physical isolation.

Latent variables can also be regarded purely in a rigorous mathematical framework, where they are separated based on theoretical properties. The concept is therefore extremely powerful and can be applied in many scenarios. Such an approach is of a more general interest to modern applications [13] [14]. Increasing computing power has made data, and the ability to process it, available in high Volume, Veracity, Velocity and Variety; the four V’s of big data [1]. As a result, technological developments in signal processing are often focused on extracting information in the same way that latent variable methods intend to. Furthermore, large datasets often contain a rich structure. Employing techniques that offer a natural representation for the structure will also yield dividends. This can be catered for through multidimensional methods, where a single variable has multiple components and the algebra serves to provide a physically meaningful portrayal of the data [15] [16]. These concepts are readily applied to solve the difficulties faced by space missions, where resources are limited and so more processing is required compared to terrestrial analogues.

This thesis is focused on two main goals: (i) to equip latent variable regression methods with the ability to process multidimensional data and (ii) to provide a performance analysis, using the latent variable framework, of the MEMS microseismometer for the InSight mission to Mars. Therefore, the latent variable approach is considered from two angles. The first part approaches the problem from a general mathematical perspective, where the goal is to use theoretical properties to separate the latent variables. The second part is application specific and aims to utilise prior information to discover the latent contributions. This is an important task, highlighted by the effort and expense which has gone into the installation of the InSight seismometers and more so after a wait of 42 years for another attempt at planetary seismology.

1.2 Problem formulation

The first seismometer was developed in China in 132AD to determine the direction of a incoming, destructive Earthquake [17]. Only in the last hundred years though, have sophisticated seismological tools been developed, owing to significant instrumentation ad-
vances [6]. These aim to analyse recorded waveforms from seismic “events” and, in turn, provide an understanding of the Earth and its processes. This is a type of imaging for far below the Earth’s crust, thousands of kilometres deeper than from any human exploration. The tools achieve this imaging from resolving the affect that the propagation medium has on the recorded waveform. Seismology is, therefore, an investigative method of exploration. In the same vein, space exploration is at the core of human endeavour. Human and robotic spaceflight missions have led to many scientific and technological developments including the more mundane such as freeze-drying [18], to the discovery that Mars harboured a watery environment potentially suitable for life [19]. To this end, employing the tools of seismology on seismic data from Mars, acquired by the InSight mission, allows the exploration deep inside another planet for the first time.

As discussed in the prequel, the quality of the interior structure provided by seismological tools relies on the quality of the ground motion determination. Therefore, the InSight microseismometer’s ability to measure this ground motion must be established. This ability is quantified by the sensor’s sensitivity; both the raw sensing limit and its sensitivity to external factors such as pressure, temperature and wind [20]. The sensors noise floor sets the lowest limit of the ground motion that the instrument is able to record, and hence whether the mission is able achieve its science goals [3] [2]. Moreover, the external factors are especially important as the sensor will be deployed in an environment unlike any on Earth. A typical seismometer installation isolates the sensor from any external contribution inside a vault underground. This allows the sensor to be well coupled to the Earth itself. However, this is not possible for the InSight mission as there are not enough resources available for deployment to create such a vault. These issues then pose difficulties in both the testing and available post-processing methods of the microseismometer [7] [8] [9].

A seismometer’s self-noise is usually determined from a comparison to a colocated sensor with a similar or better performance [21]. Consider that the microseismometer is a novel technological development designed for operation on Mars [22] [23]. Therefore, its analysis requires rigorous testing in ways unlike terrestrial seismometers, owing to the deployment difficulties. For example, the gravity on Mars is around a third of that on Earth, so a vertical inertial sensor designed for Mars cannot function in its nominal configuration on Earth. Moreover, the microseismometer performance must be examined in extreme environments [7]. On top of the difficulties posed by the testing, signal processing developments are required in order to provide a “virtual vault” and so offer post-processing protection from external signals, equivalent to what would be provided by a physical vault. In doing so, the microseismometer can also be useful for ancillary, environmental and atmospheric science goals.

Latent variable methods are techniques that take recorded data and extract hidden components (either not directly observed or obscured by other contributions) within it, often producing a physically meaningful decomposition of the original recording. Clearly, these methods have a place in the performance analysis of the microseismometer. The noise testing is a regression problem where the microseismometer is to be predicted by the reference seismometer with the difference being the self-noise [21]. The components within
1.2. Problem formulation

the microseismometer signal that correspond to external factors such as temperature can
be seen as hidden components within the data and so are a latent variable.

Such approaches have also been applied to more general regression problems where
the goal is to predict one set of data from another [24]. Modern regression applications
often involve extremely high dimensional datasets, due to technological advancement in
sensors and computing making both more available. For example, using data from high-
density networks and recent machine learning applications with images. The structure
offered by such datasets usually yields an informative analysis. In such cases, however,
the variables are often highly correlated, indeed colinear, which renders the regression ill-
posed [25]. This type of problem is ideal for latent variable methods as the decomposition
naturally finds the constituent parts and lead to a regularisation. The results presented
in this thesis will focus on the method of partial least squares, which is one such latent
variable regression algorithm that solves these computational tractability issues [26]. In
this case the hidden nature of the latent variable refers to the data be more appropriately
represented in alternative coordinates and not being directly observed.

The rich structure eluded to in high dimensional datasets can also be exploited through
multidimensional methods, in particular: complex-, quaternion- and tensor-based signal
processing [16] [27] [28] . These offer the ability to treat a multivariate source as a single
variable and impose an algebra with which to deal with it. The representation provided
by this algebra is used to delineate a relationship between the variables. For example,
complex and quaternion numbers represent a 2D or 4D rotation as a scalar multiplication.
On the other hand, the tensor approach generalises linear algebra to a multilinear form
and, therefore, yields alternative datastructures and directional decompositions. Owing to
such qualities, these multidimensional techniques have found applications in topics such as:
(i) communications [29] [30], (ii) power grid frequency estimation [31] [32] and (iii)
image classification [33] [34].

As stated the research in this thesis has two main goals:

1. to equip latent variable regression methods with the ability to process multidimensional data,

2. To provide a performance analysis, using the latent variable framework, of the MEMS
microseismometer for the InSight mission to Mars.

These are in the main motivated by: (i) the requirements/science goals of the InSight
mission, (ii) the ability of the latent variable concept to create physically meaningful
solutions and (iii) the increasing availability of data and the importance that must be
placed on how its qualities govern the way in which it should be processed.

To achieve these goals, the thesis will first examine the partial least squares algorithm
for latent variable regression and then extend it for complex, quaternion and tensor-valued
data. The proposed algorithms are analysed and compared in order to elucidate the similari-
ties and differences of each and so determine for what data each can be used. Following
this, the performance analysis of the NASA InSight microseismometer is presented, first,
in terms of its self-noise (or sensing limits) and then in terms of its thermal response.
1.3 Associated publications and reports

Peer-reviewed publications:


1.4 Thesis contributions and organisation

The central contributions of this thesis are:

1. Developments and insights into latent variable regression with partial least squares type algorithms
   - The concept of latent variable regression is explored in terms of the PLS algorithm and the generalised inverse. In doing so the limitations and optimallity of a PLS type algorithm are established. Furthermore, the Online PLS (OL-PLS) algorithm is introduced for streaming data, instead of the usual block type data.
     - Relevant publications: “An online NIPALS algorithm for Partial Least Squares”
2. A partial least squares algorithm for complex-valued widely-linear regression

- A PLS algorithm for complex-valued data is proposed, WL-CPLS. This yields a regularised widely-linear regression taking into account full complex second-order statistics. As such, the WL-CPLS provides an extension of latent variable regression for multidimensional methods and is applied for the frequency estimation of smart grids.

  - **Relevant publications:** “Widely Linear Complex Partial Least Squares for Latent Subspace Regression”
  - **Collaboration:** Sithan Kanna and Danilo Mandic.

3. Partial least squares algorithms for quaternion-valued widely-linear regression and tensor regression

- The use of multidimensional latent variables is further developed for quaternion and tensor-valued data. This leads to a PLS algorithm for quaternion-valued widely-linear regression is proposed and two further PLS algorithms are proposed for tensor-based regression. The algorithms are compared and contrasted in terms of the tensor and quaternion algebras, which leads to an analysis of the qualities of each multidimensional method. The algorithms are applied for pre-processing in image classification.

  - **Relevant publications:** “Multidimensional NIPALS Algorithms for Quaternion and Tensor Partial Least Squares Regression”
  - **Collaboration:** Ilia Kisil, Bruno Scalzo and Danilo Mandic.

4. The self-noise determination of the InSight MEMS microseismometer

- The fundamental self-noise floor of the MEMS microseismometer for the NASA InSight mission is determined from test data. An analysis of the standard coherence testing result is produced along with an analysis of a proposed delta noise method. Algorithms are then introduced for the required processing including: alignment, calibration and a robust low noise estimation. This leads to a rigorous determination of the microseismometer’s self-noise from experimental data, including that obtained from the deep space cruise phase of the mission.

  - **Relevant publications:** “SEIS: The Seismic Experiment for Internal Structure of InSight” and “A silicon seismic package (SSP) for planetary geophysics”
  - **Collaboration:** Constantinos Charalambous and William Pike.

5. Development of a method to decorrelate the microseismometer output from temperature and derivation of the thermal model’s coefficients

- In order to apply the data acquired from the microseismometer on Mars to seismological tools, its response must be decorrelated from external, aseismic
contributions. To this end, we propose a new method to remove the thermal response from the microseismometer’s outputs. In doing so a new sensor fusion approach is developed to counteract the problems of ill-posed transfer-function inversion. Furthermore, a thermal model for the microseismometer is introduced and its parameters determined for the flight units. This permits the analysis of testing in extreme cold environments (−65°C) and establishes the sensor performance at low frequencies.

- **Relevant publications:** "Full-Band Signal Extraction from Sensors in Extreme Environments: The NASA InSight Microseismometer"
- **Collaboration:** Constantinos Charalambous, Tristram Warren and William Pike.

This thesis is organised into two parts. Part A is concerned with the signal processing contributions related to latent variable regression and multidimensional methods, the contributions 1-3 are introduced in Chapters 3-5. Part B is focused on the performance analysis of the microseismometer with contributions 4 and 5 given in Chapters 6 and 7. The thesis structure is shown in Figure 1.2.

### 1.5 Statement of Originality

The content of this thesis are entirely my own work. Contributions from others are mentioned in the text and a list of references is provided. I declare that none of the work has been submitted for another qualification elsewhere.
Chapter 2

Background theory

If we have data, let’s look at data. If all we have are opinions, let’s go with mine.

Jim Barksdale

2.1 Outline

In this chapter the signal processing fundamentals, on which the thesis developments are based, are introduced. Several of these contributions are based on the PLS methodology which is fully reviewed in sequel, Chapter 3. Here, however, the required background for linear regression, generalised inverses, statistics and eigenvectors is introduced in order to frame the PLS discussion. Next, the multidimensional complex-, quaternion- and tensor-valued signal processing methods are presented in Section 2.3. This includes the necessary considerations required to introduce a PLS extension for such data. Finally, a brief overview of spectral estimation is given which is required for the seismometer noise testing in Part B. Note that this chapter outlines the signal processing fundamentals, further background and literature review is provided as necessary.

2.2 Linear regression

Estimation is a common problem in signal processing. Linear regression is commonly employed for such scenarios and forms the backbone for more advanced techniques. Specifically, linear regression is the problem of predicting a matrix of $p$ dependent variables, $\mathbf{Y} \in \mathbb{R}^{N \times p}$, as a linear combination of the columns of a matrix of $m$ independent variables, $\mathbf{X} \in \mathbb{R}^{N \times m}$, through a matrix of coefficients, $\mathbf{B} \in \mathbb{R}^{m \times p}$, described by

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{B}, \quad (2.1)$$
Chapter 2. Background theory

where \( \hat{Y} \) denotes the estimate of \( Y \) and \( N \) denotes the number of observations. The general solution for the regression coefficients, \( B \), has the form

\[
B = X^+ Y, \tag{2.2}
\]

which requires the calculation of the generalised matrix inverse \( X^+ \) [35]. The ordinary least squares (OLS) is the standard method with which to compute the required generalised inverse. The aim is to find the \( \hat{B} \) that minimises the square error between the prediction, \( \hat{Y} \) and the target output, \( Y \). This can be cast into the optimisation problem

\[
\hat{B}_{OLS} = \arg \min ||X \hat{B} - Y||_2, \tag{2.3}
\]

which has the solution

\[
\hat{B}_{OLS} = (X^T X)^{-1} X^T Y. \tag{2.4}
\]

The generalised inverse of \( X \) then has the form

\[
X^+ = (X^T X)^{-1} X^T. \tag{2.5}
\]

Remark 1. The OLS solution requires the calculation of the inverse of the covariance matrix \( X^T X \). Therefore, \( X \) must be full rank for the calculation of \( \hat{B}_{OLS} \) to be computationally tractable.

Remark 2. The model assumes that all the error in the recorded variables is in \( Y \) as the error minimised is the difference between \( Y \) and the prediction from \( X \).

2.2.1 The generalised inverse

The OLS solution for linear regression can be further understood in the context of a generalised inverse which provides a useful framework for understanding the system of equations

\[
Ax = b, \tag{2.6}
\]

akin to the linear regression problem. This equation states that columns of the matrix \( A \in \mathbb{R}^{n \times m} \) are combined by the vector \( x \in \mathbb{R}^m \) to produce the vector \( b \in \mathbb{R}^n \). The goal is to find the vector \( x \) which produces \( y \) which can be achieve through the matrix inverse \( A^{-1} \) as

\[
x = A^{-1} b.
\]

However, this only applies for square and non-singular matrices \( A \) and is therefore highly restrictive when investigating a system of linear equations. On the other hand, the generalised inverse, denoted \( A^+ \), provides the description of a general solution [36] [35]

\[
x = A^+ b + (I - A^+ A)y. \tag{2.7}
\]

where \( A^+ b \) is the particular solution for a given vector \( b \) and \((I - A^+ A)y \) is the complementary solution for any vector \( y \in \mathbb{R}^m \). As a result, this is a general expression describing
both the existence and uniqueness of a solution.

The system of linear equations defined by the matrix $A$ can be viewed as a transform between $\mathbb{R}^m$ and $\mathbb{R}^n$, and its generalised inverse is the provides the reverse mapping. This is further illuminated through examination of the criteria a matrix inverse, $A^{-1}$, must satisfy

$$A^{-1}A = I,$$
$$AA^{-1} = I.$$  

On the other hand, the a generalised inverse, $A^+$, must satisfy the following properties [36]

$$AA^+A = A,$$  
(2.8)

$$A^+AA^+ = A^+.$$  
(2.9)

which are less restrictive. In both cases, these criteria indicate that the matrix and its (generalised) inverse allow the expression of a vector, $x$ and $b$, in their respective dimensions which is possible to obtain as a linear combination of its rows or columns. For the matrix inverse, however, any vector is permitted, i.e. the solution always exists and is unique. Furthermore, a Moore-Penrose pseudoinverse [35] satisfies the conditions

$$\left( AA^+ \right)^H = AA^+, $$  
(2.10)

$$\left( A^+A \right)^H = A^+A, $$  
(2.11)

where $(\cdot)^H$ is the Hermitian transpose, where $A^HA = I$ and $I$ is the identity matrix. If a generalised inverse matrix $A^+$ satisfies these conditions then it is unique for $A$.

### 2.2.2 The fundamental theorem of linear algebra and the SVD

It has been shown that the calculation of a generalised inverse is at the core of the linear regression problem. Moreover, it also provides further insight into the system of equations. The fundamental theorem of linear algebra [37] is also concerned with the equation

$$Ax = b,$$

and how the transform represented by the matrix $A$ defines the potential results $b$ and solutions $x$ including whether they could exist or are unique, as with the generalised inverse. Strang [37] defines this equation in terms of four subspaces:

- The column space of $A$ (the space spanned by the columns of $A$ denoted as $\text{R}(A)$) which defines the possible values of a potential answer $b$,

- The row space of $A$ (the space spanned by the rows of $A$ denoted as $\text{R}(A^T)$) which defines the possible values of a potential solution $x$,

- The column nullspace space of $A$ (the space that cannot be spanned by the rows of $A$ denoted by $\text{N}(A)$) which is defined by the solutions of $Ax = 0$ and shows whether a solution is unique,
Chapter 2. Background theory

Figure 2.1. The four fundamental subspaces, \( R(\mathbf{A}) \) and \( N(\mathbf{A}^T) \) exist in \( \mathbb{R}^n \) and \( R(\mathbf{A}^T) \) and \( N(\mathbf{A}) \) exist in \( \mathbb{R}^m \).

- The row nullspace of \( \mathbf{A} \) (the space that cannot be spanned by the columns of \( \mathbf{A} \) denoted by \( N(\mathbf{A}^T) \)) which is defined by the solutions of \( \mathbf{A}^T \mathbf{y} = 0 \) and shows whether a solution will always exist.

Therefore, the vector space \( \mathbb{R}^n \) is split into the column space, \( R(\mathbf{A}) \), describing the values that can be spanned by the columns of \( \mathbf{A} \) and the row nullspace, \( N(\mathbf{A}^T) \), which describes the values that cannot be produced from the transform represented by \( \mathbf{A} \). The vector space \( \mathbb{R}^m \) is split into the row space, \( R(\mathbf{A}^T) \), which gives the values of a potential solution to (2.6) and the column nullspace, \( N(\mathbf{A}) \), which are the values of \( \mathbf{x} \) that would produce a zero vector and hence can be added to any solution. In other words, this is another version of the general solution (2.7) provided by the generalised inverse.

The four subspaces, therefore, offer an intuitive understanding of the system of linear equations (2.6). As such, it is desirable to derive a basis for these subspaces. This is achieved through the singular value decomposition (SVD) which is a workhorse of signal processing. The SVD of the matrix \( \mathbf{A} \in \mathbb{R}^{n \times m} \) is given as

\[
\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^H,
\]

(2.12)

where \( \mathbf{U} \in \mathbb{R}^{n \times n} \) and \( \mathbf{V} \in \mathbb{R}^{m \times m} \) are unitary matrices and \( \Sigma \in \mathbb{R}^{n \times m} \) is diagonal whose entries are known as the singular values of \( \mathbf{A} \). The number of non-zero singular values is equal to the rank of \( \mathbf{A} \) and the non-zero entries are always positive and are equal to the square root of the eigenvalues of the matrix \( \mathbf{A}^H \mathbf{A} \). Moreover, the matrices \( \mathbf{U} \) and \( \mathbf{V} \) are calculated as the eigenvectors of the matrices \( \mathbf{A} \mathbf{A}^H \) and \( \mathbf{A}^H \mathbf{A} \) respectively.

The matrices \( \mathbf{U} \) and \( \mathbf{V} \) provide the bases of the four subspaces \( R(\mathbf{A}) \), \( R(\mathbf{A}^T) \), \( N(\mathbf{A}) \) and \( N(\mathbf{A}^T) \), shown in Figure 2.1. The first \( r \) columns in \( \mathbf{U} \) (where \( r \) is the rank of \( \mathbf{A} \) and number of non-zero singular values) provide the basis for \( R(\mathbf{A}) \) and the remaining
2.2. Linear regression

$n - r$ columns of $U$ provide the basis of $N(A^T)$ which gives a general form to describe the vector $b$. On the other hand, the first $r$ columns of $V$ provide a basis for $R(A^T)$ and the remaining $m - r$ provide a basis for $N(A)$ which gives a general form to describe the vector $x$. Note that the subspaces $R(A)$ and $N(A^T)$ are orthogonal as are the subspaces $R(A^T)$ and $N(A)$.

2.2.3 The SVD, generalised inverses and ill-posed regression

The concept of these four subspaces explains all necessary information about the equation $Ax = b$ which forms the backbone of many real world physical problems. The theory of the generalised inverse further provides a rigorous framework for the general solution. It has been shown that the SVD provides a basis for the individual subspaces and, hence, the potential values that can be taken by $x$ for which values of $b$. Furthermore, the SVD then enables the calculation of a generalised inverse given as

$$A^+ = V\Sigma^{-1}U^H,$$

(2.13)

where $\Sigma^{-1}$ is the transpose of the matrix $\Sigma$ whose non-zero diagonal have been reciprocated. There is, therefore, an intuitive link between the expression of the system of linear equations and the calculation of the solutions. This concept will be exploited later through latent variables.

The regression problem (2.1) is essentially the same as the system of linear equations (2.6). In the regression problem the aim is to determine the transform from recorded data, whereas in the system of linear equations the aim to determine what the data can be from the transform. The linear regression solution (2.2) is calculated through finding the generalised inverse of the independent variables $X$. From (2.2), this can now be calculated from the SVD of $X$ as well as the OLS solution (2.5) and they are equivalent for full-rank data. In this case $X^+$ is a Moore-Penrose pseudoinverse [35] which satisfies the four requirements (2.8), (2.9), (2.10) and (2.11) and so it is unique for a given matrix.

The advantage of using the Moore-Penrose pseudoinverse over the OLS solution is that the calculation of the matrix inverse $(X^TX)^{-1}$ is not required. For a sub-rank data matrix $X$ this is intractable. On the other hand, the SVD requires the calculation of the eigenvectors/values of $X^TX$ and $XX^T$ which can be achieved through iterative algorithms (see Section 2.2.4). For real-world regression problems a sub-rank matrix of regressors is a common scenario where data are highly correlated or colinear, a trend in modern signal processing.

Alternative ill-posed regression solutions

Another method used to combat the colinearity problem is ridge regression [38]. This technique adds a scalar to the diagonal of the covariance matrix as $(X^TX + \mu I)$, ensuring it will be full-rank and therefore invertible. This method is effective at producing a stable calculation but allows elements of $X$ which are now known to be noisy/surplus to the process to be used in the model.
2.2.4 The calculation of eigenvectors

The calculation of eigenvectors is central to several signal processing methods, for example, to obtain the SVD. Each matrix can be seen as a transform where it rotates and stretches the initial vector to a new vector. Two special cases of this are transforms that just rotate (i.e. a unitary matrix) or purely stretch a vector. These aspects form the constituent components for many transforms. The rotation is akin to a change of basis while the stretched component indicates the important vector modification directions of the transform. These purely stretching directions are known as the eigenvectors.

An eigenvector of a matrix $A \in \mathbb{R}^{n \times n}$ is given as the solution

$$Av = \lambda v,$$

where $\lambda$ is a constant and $v \in \mathbb{R}^n$ is the eigenvector. The analytical solution is given by the characteristic polynomial $det(\lambda I - A) = 0$ which requires the solution of an $n-$th order polynomial. For higher orders this is analytically intractable and so an iterative solution must be found. This is provided by the power method algorithm [39]. The intuition for this method is demonstrated in Figure 2.2. If a matrix has a dominant eigenvalue $\lambda$ then if the transform defined by the matrix is continually applied to some vector $x$ it will predominantly stretch in the direction of $v$, thus obtaining the eigenvector.

Consider the eigen-decomposition of a matrix

$$A = V\Lambda V^{-1},$$

$$A^k = V\Lambda^k V^{-1},$$

and that

$$x = V\bar{x},$$

that is, a vector, $x$, is created by the operation, $V$, on the vector $\bar{x}$. Then

$$A^kx = A^kV\bar{x} = V\Lambda^k\bar{x},$$
\[ = \sum_{j=1}^{n} v_j \lambda_j x_j = \lambda_1 \left( \sum_{j=1}^{n} v_j \left( \frac{\lambda_j}{\lambda_1} \right)^k \tilde{x}_j \right). \]

As \( k \) is increased \( \left( \frac{\lambda_j}{\lambda_1} \right)^k \to 0 \) for \( j > 1 \) so the first eigenvector will dominate. As such, \( A^k x \) will become parallel to the first eigenvector \( v_1 \). This gives rise to the power method iteration as

\[ x^{k+1} = \frac{Ax^k}{\|Ax^k\|}. \] (2.14)

A problem with this method arises if several eigenvalues are close to each other in magnitude. In this case they will not decay rapidly and so the convergence will be slow. Further developments of this method include the Lanczos algorithm which uses the “history” of the iterations to aid the convergence [39].

### 2.2.5 Statistics and probability distributions

It has been shown that the covariance matrix \( X^T X \) of the regressors is of interest in the calculation of a linear regression solution. In a statistical framework, the covariance matrix in fact provides a way to describe the distribution of the data. In this section a brief overview of statistical distributions is presented.

Consider that the data in the matrix \( X \) is viewed as a collection of \( N \) samples of the \( m \)-variate process, \( x \in \mathbb{R}^{1 \times m} \). In this sense the rows of \( X \) are the separate sources and the columns of \( X \) show how they evolve over time. The \( m \)-variate process, \( x \in \mathbb{R}^{1 \times m} \), is described in terms of its probability density function \(^1 \) (pdf) \( p(x) \) [40]. The integral of the pdf between the limits \( A \) and \( B \) gives the probability of the process \( x \) being between the values \( A \) and \( B \). The Normal (Gaussian) distribution’s pdf for the \( m \)-variate process, \( x \in \mathbb{R}^{1 \times m} \), is described as

\[ x \sim N(\mu, C) = p(x|\mu, C) = \frac{1}{(2\pi)^{-m/2}|C|^{-1/2}} \exp\left( -\frac{1}{2} (x - \mu)^T C^{-1} (x - \mu) \right), \]

where \( p(x|\mu, C) \) denotes the joint pdf of the elements of \( x \) given the mean \( \mu \) and covariance \( C \) [41]. A common task in estimation problems is to determine the parameters which describe a distribution from obtained data, such as the mean and covariance for the Normal distribution. This is inherent in the linear regression problem, which requires the calculation of the empirical covariance matrix as \( C = X^T X \) and cross-covariance matrix \( X^T Y \).

**Remark 3.** The linear regression is optimal and captures the full statistical properties of the processes in the \( X \) and \( Y \) blocks when they are generated from the \( m \) and \( p \)-variate Normal distribution.

### 2.3 Multidimensional Signal Processing

A main thesis goal is to develop the PLS algorithm for application to multidimensional data. To that end, we must examine the array of multidimensional data types and hence

---

\(^1\)For a multivariate process this is a joint pdf.
observe the qualities of the representation. Specifically, the properties of the algebra, computation and statistics must be examined to determine the required form for a linear regression analogue in that domain. Two main types of multidimensional method: (i) the alternative algebras given by complex and quaternion numbers, where the data is still treated with linear algebra concepts and (ii) the multilinear algebra technique provided by tensors which allows new decompositions.

2.3.1 Complex-valued signal processing

Complex numbers encode bivariate data as

\[ x = x_r + jx_i. \]

This offers a natural and physically meaningful representation. For example, a 2D rotation can be performed through a straightforward multiplication. Furthermore, the polar representation can separate the rotational and stretch component of a transform, for example

\[ y = xa = x|a|\exp(j\theta). \]

As a result, complex-valued signal processing offers a useful tool for data analysis [27]. In this section the extension of the previously introduced linear regression to the complex-domain is introduced. The concept of circularity and the duality of processing complex data in the real-domain is explored.

The notion of circularity

Circularity is an important concept for the processing of complex-valued data. It admits an insightful geometric interpretation of a Complex Random Variable (CRV) and quantifies the relationship between the real and imaginary components [42]. This quantification is described by the circularity coefficient, the absolute value of the ratio of pseudocovariance, \( p = E[z^2] \), to covariance, \( c = E[z^*z] \), given by [43]

\[ \rho = \frac{|p|}{c}. \]

For a proper variable \( \rho = 0 \), while for a maximally non-circular variable (i.e. a real number represented as complex-valued) \( \rho = 1 \). Moreover, the circularity quotient (the fraction \( \frac{p}{c} \)) reveals whether the non-circularity arose due to the power imbalance or correlation between the data channels. A CRV \( z = z_r + jz_i \) has pseudocovariance \( p = E[z_r^2 - z_i^2 + 2z_rz_i] \) and covariance \( c = E[z_r^2 + z_i^2] \), and therefore, the real part of their ratio quantifies power imbalance between \( z_r \) and \( z_i \) whereas the imaginary part quantifies their correlation [27].

The circularity coefficient, hence, offers a useful description of the bivariate complex variable and is employed for several applications such as in communications systems [42]. Note that circularity accounts for all statistical moments whereas propriety refers to up to second-order moments.
### Multidimensional Signal Processing

#### Widely linear regression

In the same way as the real-valued linear regression in Section 2.2, the aim of a complex-valued regression is to find an estimate, \( \hat{Y} \), of dependent variables, \( Y \in \mathbb{C}^{N \times p} \), from independent variables \( X \in \mathbb{C}^{N \times m} \) through the minimisation of the mean square error (MSE)

\[
MSE = \text{Tr}\{E[(Y - \hat{Y})^H(Y - \hat{Y})]\}.
\]  

(2.15)

The linear regression problem, \( Y = XB \), is equipped to have enough degrees of freedom to account for the second-order relationships between the real-valued inputs, \( X \), and outputs, \( Y \). When \( X \) and \( Y \) are complex-valued, the general solution must also account for second-order noncircularity (improperness) of the data. This is achieved through a widely linear form \[44\]

\[
\hat{Y} = XH + X^*G,
\]  

(2.16)

where \( H \in \mathbb{C}^{m \times p} \) and \( G \in \mathbb{C}^{m \times p} \). Such an optimal estimator of \( Y \) is, in general, linear in terms of both \( X \) and \( X^* \), in contrast to the strictly linear result in (2.1). Observe that the widely linear regression coefficient matrices, \( H \) and \( G \), cannot be calculated through a generalised inverse of only \( X \). In order to derive the solution, we observe that the residual \((Y - \hat{Y})\) is orthogonal to both \( X \) and \( X^* \), which yields the expectations

\[
E[\hat{Y}^HX] = E[Y^HX], \quad E[\hat{Y}^HX^*] = E[Y^HX^*],
\]  

(2.17)

leading to

\[
H^HC + G^HP = R, \quad H^HP^* + G^HC^* = S^*,
\]  

(2.18)

where \( R = Y^HX \), \( S = Y^TX \), \( C = X^HX \) and \( P = X^TX \) are respectively the empirical cross-covariance, cross-pseudocovariance, covariance and pseudocovariance matrices. A simultaneous solution of (2.18) yields the widely linear regression coefficients given by

\[
H = [C - P^*C^+P]^{-1}[R^H - P^*C^+S^T],
\]

\[
G = [C^* - PC^+P^*]^{-1}[S^T - PC^+R^H],
\]  

(2.19)

The augmented forms, \( \bar{X} = [X, X^*] \) and \( \bar{Y} = [Y, Y^*] \), provide a compact representation for the widely linear regression in (2.16) as

\[
\bar{Y} = \bar{X}B,
\]  

(2.20)

where

\[
B = \begin{bmatrix} H & G^* \\ G & H^* \end{bmatrix}.
\]

The widely linear regression coefficients in \( B \) are consequently obtained as \( B = \bar{X}^+\bar{Y} \). The generalised inverse in this context is given \( \bar{X}^+ = (\bar{X}^H\bar{X})^{-1}\bar{X}^H \) (akin to the linear
least squares solution in (2.4)) which leads to

\[ X^+ = \begin{bmatrix} C & P^* \\ P & C^* \end{bmatrix} X^H. \]

A direct application of block matrix inversion [45] yields a derivation of the widely linear estimation in augmented form which requires the same calculations for the regression coefficient matrices \( G \) and \( H \) as those derived in (2.19). This highlights that the augmented form in (2.20) allows a representation of complex-valued regression in the same form as a standard real-valued linear regression in (2.1).

**Remark 4.** Widely linear regression is solved through a generalised inverse of the augmented complex matrix \( \tilde{X} \), which is a generic extension of standard regression based on a generalised inverse of the matrix \( X \). This is equivalent to the solution in (2.19).

Consider the concept of circularity in Section 2.3.1. The introduced widely linear relationship given by (2.16) is required to provide enough degrees of freedom for estimation with non-circular (improper) data, which is a more general case. However, for rotation invariant (circular) data, only a strictly-linear relationship is required [27] [46].

**Duality between complex-valued processing and processing in \( \mathbb{R}^2 \)**

Complex valued data can be equivalently represented and processed in \( \mathbb{R}^2 \) [27] [47]. To illustrate this duality, consider a matrix, \( X \in \mathbb{C}^{N \times m} \), which can be written in terms of its real and imaginary parts as \( X = (X_R + jX_I) \). The individual parts, \( X_R \in \mathbb{R}^{N \times m} \) and \( X_I \in \mathbb{R}^{N \times m} \), can be represented as

\[ X_{Re} = [2X_R, 2X_I], \tag{2.21} \]

where \( X_{Re} \in \mathbb{R}^{N \times 2m} \). The isomorphism between this representation in \( \mathbb{R}^{N \times 2m} \) and the augmented form of complex-valued data, given by

\[ X = [X, X^*], \]

is described by the transform matrix, \( \Gamma_m \), given by

\[ \Gamma_m = \begin{bmatrix} I_m & -jI_m \\ I_m & jI_m \end{bmatrix}, \tag{2.22} \]

where \( I_m \in \mathbb{R}^{m \times m} \) is the identity matrix of appropriate dimensions. The transform matrix, \( \Gamma_m \), is unitary with a scale factor of 2 so that \( \Gamma_m^H \Gamma_m = 2I_{2m} \) and the mapping in (2.21) then becomes

\[ [2X_R, 2X_I] = [X, X^*] \Gamma_m. \]
2.3. Multidimensional Signal Processing

Table 2.1. Quaternion multiplication

<table>
<thead>
<tr>
<th>×</th>
<th>1</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>i</td>
<td>j</td>
<td>k</td>
</tr>
<tr>
<td>i</td>
<td>i</td>
<td>-1</td>
<td>k</td>
<td>-j</td>
</tr>
<tr>
<td>j</td>
<td>j</td>
<td>-k</td>
<td>-1</td>
<td>i</td>
</tr>
<tr>
<td>k</td>
<td>k</td>
<td>j</td>
<td>-i</td>
<td>-1</td>
</tr>
</tbody>
</table>

2.3.2 Quaternion-valued signal processing

Quaternion algebra and notation

Quaternions, $\mathbb{H}$, are both a skew-field over the real numbers, $\mathbb{R}$, whereby element $g \in \mathbb{H}$ is a four-element number, a four dimensional vector, and an ordered pair of two complex numbers [28] [48]. A quaternion scalar $x \in \mathbb{H}$ is represented as

$$ x = x_r + x_1i + x_jj + x_kk, $$

$$ \text{Re}(x) = x_r, $$

$$ \text{Im}(x) = x_i + x_jj + x_kk, $$

where $i$, $j$ and $k$ are the imaginary units governed by the multiplication scheme in Table 2.1. Observe that $ij = k \neq ji = -k$, and as a result, the multiplication of two quaternion numbers, $x$ and $y$, is in general not commutative, that is

$$ xy \neq yx. $$

The modulus of a quaternion, $x$, is defined as

$$ |x| = \sqrt{x_r^2 + x_i^2 + x_j^2 + x_k^2}. $$

The quaternion conjugate operator negates the imaginary components

$$ x = x_r + x_1i + x_jj + x_kk, $$

$$ x^* = x_r - x_1i - x_jj - x_kk, $$

analogous to the complex conjugate operator. The involution operators are unique to quaternions and is described as $x^\alpha = -\alpha x \alpha$ which represents a rotation by $\pi$ of $x$ around $\alpha$, where $\alpha$ is a pure unit quaternion, that is, a three-dimensional quaternion with a unit norm and has no real component [49]. These involutions have the property $(x^\alpha)^\alpha = x$. Special instances of the quaternion involution around the imaginary axes, where $\alpha \in \{i, j, k\}$, are given as

$$ x = x_r + x_1i + x_jj + x_kk, $$

$$ x^i = x_r + x_1i - x_jj - x_kk, $$

$$ x^j = x_r - x_1i + x_jj - x_kk, $$

$$ x^k = x_r - x_1i - x_jj + x_kk, $$

and are referred to as $i-$, $j-$ and $k-$ involutions. These lead to the $\alpha$-conjugate operations
frequently used in quaternion augmented statistics [28].

\[
x = x_r + x_i i + x_j j + x_k k, \\
x^{i*} = x_r - x_i i + x_j j + x_k k, \\
x^{j*} = x_r + x_i i - x_j j + x_k k, \\
x^{k*} = x_r + x_i i + x_j j - x_k k.
\]

The complete second-order statistics of a real vector-variate random process, \(x \in \mathbb{R}^m\), are captured by its mean, \(E[x]\), and its covariance matrix, \(E[xx^T]\). However, the mean and the direct extension of the covariance matrix, \(E[xx^H]\), are not sufficient for a full second-order description of a quaternion-valued vector-variate random process, \(x \in \mathbb{H}^m\) [28] [48]. Instead, the consideration of the \(i\)–, \(j\)– and \(k\)–complementary covariance matrices \(E[xx^{iH}]\), \(E[xx^{jH}]\) and \(E[xx^{kH}]\) is also required in order to cover the augmented quaternion statistics. This is an extension of augmented statistics for complex-valued data (where the complementary pseudocovariance matrix \(E[xx^T]\) is required in addition to the standard covariance) which is now well understood in the literature [50]. In order to conveniently account for the full second-order statistics, a quaternion-valued vector process can be represented in augmented form as

\[
\mathbf{x} = [x, x^i, x^j, x^k].
\]  

(2.23)

**Quaternion circularity**

Quaternion (non)circularity, analogously to the complex version in Section 2.3.1, indicates whether the distribution of a quaternion random variable is rotation invariant, that is, if all the four components are independent and balanced powers [28]. Circularity involves a full statistical description as it refers to all statistical moments of a distribution. However, in practice, a second-order concept known as properness is commonly used [48]. For a quaternion-valued variable, \(x\), to be proper, all its complementary covariance matrices must vanish, that is, \(E[xx^{i*}]\), \(E[xx^{j*}]\) and \(E[xx^{k*}]\) equal zero. The degree of properness is then given by the three circularity measures [51]

\[
\rho_i = \frac{E[xx^{i*}]}{E[xx^*]}, \quad \rho_j = \frac{E[xx^{j*}]}{E[xx^*]}, \quad \rho_k = \frac{E[xx^{k*}]}{E[xx^*]},
\]  

(2.24)

which are generalisations of the circularity quotient for complex-valued data [43].

**Quaternion processing in \(\mathbb{R}^4\)**

A quaternion data matrix, \(\mathbf{X} \in \mathbb{H}^{N \times m}\) is described as

\[
\mathbf{X} = \mathbf{X}_r + i\mathbf{X}_i + j\mathbf{X}_j + k\mathbf{X}_k,
\]  

(2.25)

where \(\mathbf{X}_\eta \in \mathbb{R}^{N \times m}, \eta = \{r, i, j, k\}\), where \(N\) and \(m\) stand for the number of samples and variables respectively. The quaternion matrix, \(\mathbf{X}\), is then represented in augmented form
as
\[
\mathbf{X} = [\mathbf{X}, \mathbf{X}^i, \mathbf{X}^j, \mathbf{X}^k] \in \mathbb{H}^{N \times 4m},
\]
where \( \mathbf{X}^\alpha = -\alpha \mathbf{X} \alpha \) for \( \alpha = \{i, j, k\} \) is the quaternion involution operator.

An isomorphism between the quaternion domain and a four dimensional real vector space has been established through the transform matrix \( \Gamma_m \) [28] [48]
\[
\Gamma_m = \begin{bmatrix}
  I_m & iI_m & jI_m & kI_m \\
  I_m & -iI_m & -jI_m & -kI_m \\
  -I_m & jI_m & -kI_m & iI_m \\
  -I_m & -jI_m & kI_m & -iI_m
\end{bmatrix} \in \mathbb{H}^{4m \times 4m},
\] (2.26)
which provides a link between the augmented quaternion matrix, \( \mathbf{X} \), and the real-valued matrix \( \mathbf{X}_{\text{Re}} = [\mathbf{X}_r, \mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k] \), given by
\[
\mathbf{X}_{\text{Re}} = \frac{1}{4} \mathbf{X} \Gamma_m^*.
\] (2.27)
This transform is invertible and is, in fact, unitary up to a scale factor of 4, so that the augmented matrix is retrieved as
\[
\mathbf{X} = \mathbf{X}_{\text{Re}} \Gamma_m^T.
\] (2.28)
The transform in (2.27) converts an augmented quaternion variable to a real-valued representation in \( \mathbb{R}^4 \). A general matrix product for a row vector \( \mathbf{x} = [x_r, x_i, x_j, x_k] \in \mathbb{R}^{1 \times 4} \) is given by
\[
[y_r, y_i, y_j, y_k] = [x_r, x_i, x_j, x_k] \mathbf{H},
\]
where the entries of the matrix \( \mathbf{H} \in \mathbb{R}^{4 \times 4} \) can take any real value. The vector \( [x_r, x_i, x_j, x_k] \) is equivalently represented as a quaternion scalar \( x \), however, the quaternion product \( y = xh \) represented in the real-domain is a special case of the matrix equation, \( \mathbf{y} = \mathbf{xH} \), where the matrix, \( \mathbf{H} \), is restricted to the form [52]
\[
\mathbf{H} = \begin{bmatrix}
  h_r & h_i & h_j & h_k \\
  -h_i & h_r & -h_k & h_j \\
  -h_j & h_k & h_r & -h_i \\
  -h_k & -h_j & h_i & h_r
\end{bmatrix},
\] (2.29)
where \([h_r, h_i, h_j, h_k]\) are the real and imaginary components of the quaternion number \( h \).

Remark 5. An augmented quaternion variable can be equivalently processed in the real-domain, with both representations having the same degrees of freedom for processing. The quaternion algebra can be viewed as a constrained matrix product in the real-domain indicating that the structure of a quaternion-valued estimation problem will be distinct from a generic real-valued problem [53].
Quaternion widely linear estimation

In Section 2.3.2 it was shown that to describe the full second-order information of a quaternion-valued process consideration of all the $i$, $j$, $k$-complementary covariance matrices along with the standard covariance matrix is required. This is required when dealing with quaternion-valued estimation, where it is desired to estimate the output $Y \in \mathbb{H}^{N \times p}$ from the input $X \in \mathbb{H}^{N \times m}$. In this case the minimum mean square error (MMSE) estimate of the process in $Y$ is linear in terms of $X$, $X^i$, $X^j$ and $X^k$ as \[ \hat{Y} = XG_1 + X^iG_2 + X^jG_3 + X^kG_4, \] (2.30)

which is referred to as a widely linear quaternion regression [48] and is an extension of complex-valued widely linear estimation [44].

A compact form of the widely linear estimation in (2.30) can be obtained by representing the quaternion variable $X$ in augmented form, $\tilde{X} \in \mathbb{H}^{N \times 4m}$, which yields

\[ \hat{Y} = \tilde{X}B, \] (2.31)

where $B \in \mathbb{H}^{4m \times p}$ is the vector of the regression coefficients. This enables the calculation of the regression coefficients analogously to the real-valued case in (2.2) as

\[ B = \tilde{X}^+Y. \]

Note that if the data are circular (Section 2.24) then only the covariance matrix, $E[xx^*]$, needs to be considered for processing and so the estimation (2.30) becomes strictly linear, $\hat{Y} = XB$.

### 2.3.3 Tensor Notation and Preliminaries

A tensor is a natural generalisation of a two-way matrix to a general M-way case, which yields a multidimensional array [54] [55]. In comparison to the previously introduced (hyper)complex algebra techniques, the power of the tensor technique comes from the extension of linear algebra concepts to a multilinear form with new tensor operations and decompositions. On the other hand, the (hyper)complex methods exist within the standard tools of linear algebra.

A tensor is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$, where $M$ is known as its order which specifies the number of its dimensions, known as modes. Consequently, each element of $\mathcal{X}$ has $M$ indices. For instance, the third-order tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ is composed of scalars $x_{ijk}$. By fixing all but two indices in $\mathcal{X}$, we obtain a matrix substructure that is called a slice, e.g. $\mathcal{X}(\cdot:j:k) = X_{jk}$. Likewise, a vector substructure, or fibre, requires fixing all but one index, e.g. for a third-order tensor is extracted as $\mathcal{X}(i: \cdot : \cdot) = x_{ijk}$. An operation called mode-n unfolding is used to reshape a tensor into a matrix. This is performed by grouping mode-n fibres of a tensor $\mathcal{X}$ as the columns of the unfolded version $\mathcal{X}_{(n)}$. For example, the mode-n
unfoldings of a third order tensor \( \mathbf{X} \in \mathbb{R}^{I \times J \times K} \)

\[
\begin{align*}
\mathbf{X}(1) &= \left[ \mathbf{X}_{(:,1,1)} \quad \mathbf{X}_{(:,1,2)} \quad \cdots \quad \mathbf{X}_{(:,1,K)} \right] \in \mathbb{R}^{I \times J \times K}, \\
\mathbf{X}(2) &= \left[ \mathbf{X}_{(:,1,:)} \quad \mathbf{X}_{(:,2,:)} \quad \cdots \quad \mathbf{X}_{(:,J,:)} \right] \in \mathbb{R}^{K \times I \times K}, \\
\mathbf{X}(3) &= \left[ \mathbf{X}_{(:,1,:)} \quad \mathbf{X}_{(:,1,:)} \quad \cdots \quad \mathbf{X}_{(:,I,:)} \right] \in \mathbb{R}^{K \times J \times I}.
\end{align*}
\]

(2.32)

The three unfolding modes for a third order tensor are shown in Figure 2.3. The mode-1 case essentially stacks the column (vertical) vectors, mode-2 row (horizontal) vectors and mode-3 tube (into the page) vectors into a matrix.

Mode-n unfolding permits the multiplication between a tensor \( \mathbf{X} \) and a matrix \( \mathbf{A} \). This operation is called a \textit{mode-n product} and is defined as follows

\[
\mathbf{Y} = \mathbf{X} \times_n \mathbf{A} \iff \mathbf{Y}_{(n)} = \mathbf{A} \mathbf{X}_{(n)}.
\]

(2.33)

To further illustrate, consider the mode-1 unfolding of a third order tensor \( \mathbf{X} \in \mathbb{R}^{I \times J \times K} \) e.g. the \( n = 1 \) case in Figure 2.3. The mode-n product then multiplies the concatenated column vectors (vertical fibres) \( \mathbf{X}_{(1)} \in \mathbb{R}^{I \times J \times K} \) by a matrix \( \mathbf{A} \in \mathbb{R}^{M \times I} \) to give \( \mathbf{Y}_{(1)} = \mathbf{A} \mathbf{X}_{(1)} \). The resultant \( \mathbf{Y}_{(1)} \) is then folded back into the third order tensor \( \mathbf{Y} \in \mathbb{R}^{M \times J \times K} \).

A tensor tensor type product is given by the \textit{mode-n to mode-q contraction product}. Between tensors \( \mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_M} \) and \( \mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_q \times \cdots \times J_P} \) this is defined as

\[
\mathbf{S} = \langle \mathbf{X}, \mathbf{Y} \rangle_{\{n,q\}},
\]

(2.34)

where \( \mathbf{S} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_M \times J_1 \times \cdots \times J_{q-1} \times J_{q+1} \times \cdots \times J_P} \) has the entries

\[
s(i_1,\ldots,i_{n-1},i_{n+1},\ldots,i_M,j_1,\ldots,j_{q-1},j_{q+1},\ldots,j_P) = \sum_{i_n=1}^{I_n} x(i_1,\ldots,i_{n-1},i_n,i_{n+1},\ldots,i_M) y(j_1,\ldots,j_{q-1},i_n,j_{q+1},\ldots,j_P).
\]

(2.35)

Note that \( I_n = J_q \). For example, the mode-1 to mode-1 contraction product, \( \mathbf{S} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_M \times J_1 \times \cdots \times J_{q-1} \times J_{q+1} \times \cdots \times J_P} \).
\( \mathbb{R}^{I_1 \times I_2 \times I_3 \times J_1 \times J_2 \times J_3} \), of the third order tensors \( \mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) and \( \mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times J_3} \) is given as

\[
\delta(i_2, i_3, j_2, j_3) = \sum_{i_1=1}^{I_1} x(i_1, i_2, i_3) y(i_1, j_2, j_3).
\]

Another kind of unfolding operation is given by the canonical matrix unfolding of a tensor \( \mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_M} \). This is denoted as \( \mathbf{X}^{(n)} \in \mathbb{R}^{I_1 I_2 \cdots I_n+1 I_n+2 \cdots I_M} \), which has entries\(^2\) [54]

\[
X^{(n)}(i_1 i_2 \cdots i_n i_{n+1} \cdots J_M) = X(i_1, i_2, \ldots, i_M).
\]  

(2.36)

The canonical unfolding can be written in terms of the reshape MATLAB function as

\[
\mathbf{X}^{(n)} = \text{reshape}(\mathbf{X}, I_1 I_2 \cdots I_n, I_{n+1} I_{n+2} \cdots I_M).
\]

Note that this is not equivalent to the mode-\(n\) unfolding in (2.32).

**Tucker decomposition**

The Tucker decomposition of an order \( M \) tensor \( \mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M} \) is defined as

\[
\mathbf{X} = \mathbf{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_M \mathbf{A}^{(M)},
\]  

(2.37)

where \( \mathbf{G} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_M} \) is the core tensor and \( \mathbf{A}^{(i)} \in \mathbb{R}^{I_i \times J_i} \) are the factor matrices \([55]\). Figure 2.4 shows a pictorial view of the Tucker decomposition. This gives rise to a new form of rank for tensors known as the multilinear rank. The Tucker decomposition decomposes a tensor into a low multilinear rank form, for the example above this is given as the tuple rank-\((J_1, J_2, \ldots, J_M)\), where each entry is defined as the number of non-zero eigenvalues in each factor matrix. In comparison to the SVD of a matrix in Figure 2.5, the core tensor from the Tucker decomposition, \( \mathbf{G} \), is analogous to the matrix \( \Sigma \) from the SVD whereas the factor matrices are akin to the matrices of singular vectors. The decomposition gives a compression of the original tensor into a smaller core tensor and can be computed by several algorithms, including the higher order SVD (HOSVD) and higher order orthogonal iteration (HOOI). The HOSVD in Algorithm 1 is designed to be a generalisation of the SVD for multilinear algebra and consequently, yields all-orthogonal factor matrices and core tensor. An alternative approach is the HOOI method in Algorithm 2 which uses an iterative scheme to solve the optimisation problem of finding the closest low multilinear-rank tensor approximation of the original tensor.

The mode-\(n\) unfolding for a Tucker decomposition is given in terms of the core tensor and factor matrices as

\[
\mathbf{X}^{(n)} = \mathbf{A}^{(n)} G^{(n)} (\mathbf{A}^{(M)} \otimes \mathbf{A}^{(M-1)} \otimes \cdots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \cdots \otimes \mathbf{A}^{(1)})^T.
\]  

(2.38)

This has a Kronecker product structure which is an important constraint of tensor decompositions.

\(^2\)This uses a little endian ordering where the index \( i_1 i_2 \cdots i_n = i_1 + (i_2 - 1) I_1 + (i_3 - 1) I_1 I_2 \cdots (i_n - 1) I_1 \cdots I_{n-1} [1] \).
2.3. Multidimensional Signal Processing

\[ X \approx U_1 \mathbf{G} U_2 \]

Figure 2.4. Tucker decomposition of a tensor.

\[ X \approx U \Sigma V^T \]

Figure 2.5. SVD of a matrix.

Algorithm 1. The HOSVD algorithm [55]

1: \textbf{Input:} \( X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}, J_1, J_2, \ldots, J_M \)
2: \textbf{for} \( i = 1, \ldots, M \) \textbf{do}
3: \hspace{1em} \( A^{(i)} \) taken as the leading \( J_i \) left singular vectors of \( X^{(i)} \)
4: \textbf{end for}
5: \( \mathbf{G} = X \times_1 A^{(1)T} \times_2 A^{(2)T} \times_3 \cdots \times_M A^{(M)T} \)
6: Store \( \mathbf{G}, A^{(1)}, A^{(2)}, \ldots, A^{(M)} \)

Algorithm 2. The HOOI algorithm [55]

1: \textbf{Input:} \( X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}, J_1, J_2, \ldots, J_M \)
2: Initialise \( A^{(i)} \in \mathbb{R}^{I_i \times J_i} \) for \( i = 1, \ldots, M \) using the HOSVD
3: \textbf{do}
4: \hspace{1em} \textbf{for} \( i = 1, \ldots, M \) \textbf{do}
5: \hspace{2em} \( Y = X \times_1 A^{(1)T} \times_2 A^{(2)T} \times_3 \cdots \times_M A^{(M)T} \)
6: \hspace{2em} \( A^{(i)} \) taken as the leading \( J_i \) singular vectors of \( Y^{(i)} \)
7: \hspace{1em} \textbf{end for}
8: \textbf{while} Until convergence
9: \( \mathbf{G} = X \times_1 A^{(1)T} \times_2 A^{(2)T} \times_3 \cdots \times_M A^{(M)T} \)
10: Store \( \mathbf{G}, A^{(1)}, A^{(2)}, \ldots, A^{(M)} \)
Tensor regression

A tensor regression can be expressed [56]

\[ \mathbf{Y} = f(\mathbf{X}; \mathbf{B}), \] (2.39)

where the input \( \mathbf{X} \in \mathbb{R}^{N \times I_1 \times I_2 \times \cdots \times I_M} \) and output \( \mathbf{Y} \in \mathbb{R}^{N \times J_1 \times J_2 \times \cdots \times J_P} \) are related through a tensor \( \mathbf{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M \times J_1 \times J_2 \times \cdots \times J_P} \) [56]. Note that the tensors \( \mathbf{X} \) and \( \mathbf{Y} \) are considered to be \( N \) samples of the tensor variate processes. To predict a single, scalar, realisation \( y \) from a single tensor observation \( \mathbf{X} \) the function \( f(\cdot; \cdot) \) is given as \( \langle \mathbf{X}, \mathbf{B} \rangle \) which is the inner product given as \( y = \text{vec}^T(\mathbf{X})\text{vec}(\mathbf{B}) \). The general tensor-variate regression (where the output is a tensor) is then expressed succinctly through a mode-1 matrix unfolding of the tensors as

\[ \mathbf{Y}(1) = \mathbf{X}(1)\mathbf{B}_{(M)}. \] (2.40)

In order to calculate \( \mathbf{B}_{(M)} \), the generalised inverse \( \mathbf{X}(1) \) must be found. This gives the calculation

\[ \mathbf{B}_{(M)} = \mathbf{X}^+(1) \mathbf{Y}(1), \] (2.41)

which is akin to two-way multivariate regression in (2.2). The resultant matrices are then folded back into their tensor form.

Remark 6. Consider the real-valued linear regression (2.1) compared to the tensor regression (2.39). The matrix \( \mathbf{X} \in \mathbb{R}^{N \times m} \) in the linear regression problem is viewed as \( N \) observations of a vector variate process \( \mathbf{x} \in \mathbb{R}^{1 \times m} \). Analogously the tensor \( \mathbf{X} \in \mathbb{R}^{N \times I_1 \times I_2 \times \cdots \times I_M} \) is \( N \) observations of a tensor variate process \( \mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M} \), that is, the tensor storing \( N \) samples is a vector of a tensor. In this way, the tensor regression is a general extension of the linear regression where a tensor (of a certain order) is predicted at each realisation rather than just a vector. This storage of the several samples allows an implicit averaging to calculate the regression in both cases.

2.4 Frequency domain analysis

A frequency domain representation, \( \mathbf{X} \), of the time domain signal, \( x \), is given by

\[ \mathbf{X} = \mathcal{F}\{x\}, \]

where \(\mathcal{F}\{\cdot\}\) denotes the Fourier transform. The resulting spectrum, \( \mathbf{X} \), yields the distribution over frequency of a function of time, \( x \). The result is a complex function of frequency where the phase of the signal is encoded as the angle between components and the amplitude is absolute value.

The Fourier transform assumes a continuous function in both the frequency and time domain and is an integral transform. In many applications, however, the recorded data are sampled. The discrete time Fourier transform (DTFT) provides an approach for such cases, assuming that the data are sampled at an even spacing. The discrete Fourier transform goes one step further, where the frequency representation is given as an even
2.4. Frequency domain analysis

sampling of one cycle of the DTFT and so the representation is discrete in both time
and frequency domains. In this thesis, Fourier analysis is typically performed on a vector
containing \( N \) discrete samples, \( x = [x_0, x_1, \ldots, x_{N-1}]^T \). The discrete Fourier transform (DFT) \([57]\) is then given by

\[
X[k] = \sum_{n=0}^{N-1} x[n] \exp \left( -\frac{2\pi i}{N} kn \right).
\]

The inverse transform back to the time domain is then given as

\[
x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \exp \left( \frac{2\pi i}{N} kn \right).
\]

A computationally efficient method to calculate the DFT is given by the fast Fourier
transform (FFT) \([58]\).

The time domain samples are given for \( n = 0, 1, \ldots, N - 1 \) and for frequency as \( k = 0, 1, \ldots, N - 1 \). The time domain signal is usually acquired by sampling a process at
a constant rate denoted as the sampling frequency \( f_s \). The time at which a sample is
acquired is then \( \frac{n}{f_s} \) and the frequency for each entry (or “bin”) is at \( \frac{k}{f_s} \)
with a maximum at the Nyquist rate \( \frac{f_s}{2} \) \([57]\).

The DFT is used to provide an estimate of the spectrum from real world, sampled data.
When the DFT is implemented for such a practical spectral estimation, the time domain
signal is usually multiplied by a window function. This window tapers the beginning and
end of the time series in order to avoid a discontinuity, which occurs because the DFT
essentially considers the time series as a cycle of a periodic function\(^3\). Without an applica-
tion of a specific window function, the result can be considered as from the application
of a rectangular window. This has the effect of convolving the DFT spectrum with a
\( \text{sinc} \) function and so a single frequency tone would leak into neighbouring frequencies.
This is limited by a careful choice of a window function which attenuates the power in
the sidelobes. There is, however, a trade-off between the minimisation of the leakage into
neighbouring frequency bins and the resolution of the mainlobe \([59]\).

The acquired time series signal, \( x \), can also be considered as a statistical realisation
of some process. Therefore, the spectral estimate can be improved by averaging across
different realisations. This is summarised by the Welch method \([60]\). The obtained time
series is split up into separate overlapping sections and the spectral estimation is given
from the average of the DFT computed on each. As a result, the variance of the estimate
is decreased. To summarise, in a spectral estimation there are the following parameters
and trade-offs:

- Window type: the choice of the window dictates the trade-off between the minimisa-
tion of the spectral leakage and the resolution of an individual frequency component.

- Window length: the choice of how long a window is used for estimation is given by
the overlap length and number of realisations to average. More averaging lowers the
estimates’ variance but the frequency resolution is lower.

\(^3\)The DFT is a sampling of one cycle of the discrete time Fourier transform \([57]\).
Remark 7. In this thesis, the Welch method is used for all spectral estimation, averaging over two Hanning windowed time series sections overlapping by 50%, unless stated otherwise.
Part I

Part A
Chapter 3
Latent Variable Regression

Every second of every day, our senses bring in way too much data than we can possibly process in our brains

Peter Diamandis

3.1 Introduction to latent variables

The previous chapter outlined the background of linear regression along with the foundations of component analysis and statistical learning. We will now develop these concepts to introduce latent variable methods for regression, particularly the method of partial least squares, which will then be implemented for multidimensional extensions in this thesis.

Consider again the data stored in a matrix \( X \in \mathbb{R}^{N \times m} \). This is regarded as being \( N \) samples of a vector \( x \) collected into a matrix, as shown in Figure 3.1. The matrix \( X \) is, therefore, a data structure storing observed values of a vector variate process, e.g. a normally distributed vector \( x \sim \mathcal{N}(\mu, C) \). In this statistical framework, the distribution provides the full description of the information stored in the matrix. This is known as the entity and the apportioning of information across the row vector is the entity space.

Storing several vector samples into a matrix, \( X \), as a data structure allows the estimation of the distribution parameters from the observed data through straightforward operations as

\[
\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad \hat{C} = \frac{1}{N} X^T X.
\]

In this case the process or entity is stationary and so there is no time dependent correlation, that is, the samples are independent of their entry location within the matrix. An examination over the samples enables the determination of such qualities and so the estimation of the entity parameters as above. The description of change over this “long” dimension is referred to as the sample space.

Remark 8. A matrix data structure stores several realisations of a vector variate process and can be considered in terms of the “entity” and “sample” space. The entity space describes the possible values of a realisation of the vector variate process, that is, the
3.1. Introduction to latent variables

The matrix $X$ is created from $N$ samples/realisations of an $m$-variate process generated from some distribution $D(\theta)$. The variable, $x$, can be seen as a sliding window through the matrix $X$.

Spatial distribution. In a regression scenario, this is transformed into another entity by the regression coefficients. On the other hand, the sample space describes how the process varies over time and allows estimation of parameters through straightforward operations. In other words, the sample space is the column space, $\mathbb{R}(X)$, and the entity space is the row space, $\mathbb{R}(X^T)$, and so we can study the matrix using the intuitive concepts in linear algebra.

3.1.1 Latent Variables in a matrix

The vector variate process, $x \in \mathbb{R}^{1 \times m}$, with mean $\mu$ and covariance matrix $C$, stored in a data matrix $X \in \mathbb{R}^{N \times m}$, can be redistributed across the rows in an alternative coordinate basis. In fact, the original data matrix $X \in \mathbb{R}^{N \times m}$ is in the coordinate basis given by the identity matrix $I \in \mathbb{R}^{m \times m}$, whereas another coordinate basis, $P \in \mathbb{R}^{m \times r}$, where $r \leq m$, presents the data in terms of another data matrix, $T \in \mathbb{R}^{N \times r}$, with $XP \approx T$. The general problem statement is the matrix factorisation

$$X \approx TP^T. \quad (3.1)$$

The derived entity, $t \in \mathbb{R}^{1 \times r}$ stored in the matrix $T$ presents the information within the stored data matrix $X$ in an alternative space. This is useful as the original entity, $x$, is often obtained only from what can be recorded and may not be the most physically meaningful representation. The matrix factorisation is calculated based on constraints or approximations which offer a more interpretative or parsimonious entity and so de-noise and remove spurious components within the originally recorded data. This introduces the concept of a latent variable, whereby the column variables in the matrix $T$ represent a hidden but physically meaningful component that explains the information in the recorded data matrix $X$.

Figure 3.2 provides a pictorial view of the matrix factorisation that produces a latent variable decomposition of the original data matrix $X$. The matrix, $X$, is a sum of $m$ rank-1 components given by the outer product of a column variable and basis vector. Each variable is a column of $X$ with the coordinates given as a row of the identity matrix $I$. For
the latent variable approximation, the matrix, $X$, is approximated by the sum of rank-1 approximations generated from the columns of the matrix $T$ with basis vectors $P^T$. The consideration of the entity and sample space is important. The new entity is viewed as a latent variable over time, storing several samples of the original data enables computation and separation into the desirable components.

A special case of the matrix factorisation (3.1) is the sub-rank approximation where $r < m$ which is known as dimensionality reduction. Here the original data matrix is assumed to contain components that are redundant to the process being studied, common in real-world problems where the variables are highly-correlated [61]. As such the data matrix $X$ is poorly conditioned or sub-rank and so a latent variable decomposition in terms of a lower rank matrix $T$, offers a more appropriate matrix for further analysis. The number of components selected is chosen as the first $r$ with significant associated singular values compared to the $r+1$th component. In this case it is assumed that all the significant variation of the data within the matrix belongs to the process being studied and what is removed is a redundant contribution. To reject a component with significant variation in the original matrix $X$, it must be identified as extraneous through further analysis. Methods such as PCA therefore identify hidden components in the sense that they were not directly observed, it will not identify a component hidden below noise.

**Remark 9.** A latent variable is a hidden component within recorded data. In the context of linear algebra this is a sum of rank-1 approximations of the original data matrix given $X \approx \tilde{X} = TP^T = \sum_{i=1}^{r} t_i p_i^T$, where $T$ are the new, latent, variables and $P$ is the new coordinate system.
3.1.2 Principal component analysis: covariance matrix diagonalisation

The matrix factorisation (3.1) is often calculated with constraints on the matrices $T$ and $P$. For example, non-negative matrix factorisation has proved extremely popular with clustering problems and offers strong physical meaning with image data [62] [63]. Perhaps the most important latent variable technique is Principal component analysis (PCA) [64] [65] where the matrix $T$ is constrained to be orthogonal ($T^T T$ is a diagonal matrix) and the matrix $P$ is constrained to be unitary ($P^T P = I$). In this way the original components in the data matrix $X$ have been transformed into independent components, $T$, in a second-order sense. As the matrix $P$ is unitary, it represents a basis and so the PCA result can be viewed as a rotation of the original data matrix into a new coordinate system where each component is orthogonal.

The PCA result is the solution to the problem of finding the maximal projection of the data matrix $X$. The aim is to find the unit vector, $p$, that explains the direction of largest variation in $X$. The PCA as an optimisation problem [65] is then given as

$$p = \arg \max_{||p||=1} ||Xp||^2.$$  \hspace{1cm} (3.2)

The result of which is the given by the leading eigenvector of $X^T X$ [65]. Recall that this is the leading right singular vector of the matrix $X$ obtained from the SVD $X = U \Sigma V^T$. As a result, the PCA solution is intrinsically linked to the SVD calculation where

$$T = U \Sigma, \quad P = V.$$ 

Consider again that the matrix $X$, consists of $N$ samples of the normally distributed vector $x \sim N(\mu, C)$. The empirical estimate of the covariance matrix is given $\hat{C} = X^T X$, and explains the variance of each component and their covariance with each other. On the other hand, the empirical estimate of the covariance matrix of the PCA variables is diagonal as the components are orthogonal. This new covariance matrix is given by $\Sigma$ from the SVD result. The PCA result is therefore a covariance diagonalisation operation [65].

3.2 Regression with latent variables

The linear regression problem, introduced in Chapter 2.2, is given as

$$Y = XB.$$ 

It was shown that the regression coefficient matrix $B$ is obtained through the calculation of a generalised inverse for $X$ as $B = X^+ Y$. The OLS solution $B_{OLS} = (X^T X)^{-1} X^T Y$ is ill-posed when the data matrix $X$ is sub-rank and so the Moore-Penrose pseudoinverse was introduced to counteract this [35]. There is, therefore, a clear link between PCA and the linear regression solution. This can be further expanded for latent variables in general.

Each row of the regression coefficient matrix, $B$, indicates how a column variable in
**Chapter 3. Latent Variable Regression**

$X$ is transformed to the dependent variables in $Y$. If the columns of $X$ are in some way correlated then the regression may not be the most parsimonious. As shown in Figure 3.2, latent variable decompositions present the original matrix in a new set of variables and for PCA they are independent. Using the principal components as the independent variables, therefore, yields a regression where no information is shared amongst the independent variables and so the regression coefficients explain how each individual variation is transformed to the domain of the dependent variables $Y$. This is known as principal component regression (PCR) \cite{25} and is equivalent to the Moore-Penrose psuedoinverse. However, a subset of principal components is usually selected to remove those which are purely noise \cite{26} so it is a stronger tool for ill-posed regression scenarios.

Latent variable regression therefore has two goals:

1. To produce a generalised inverse of $X$ which can be used to calculate the regression coefficient matrix as $B = X^+ Y$ in circumstances where the problem may be ill-posed.

2. To enable the regression to be between the fundamental information components within the data and so yield an in-depth data analysis.

The first of these goals is based around the computational tractability and parsimony of the regression. This is most useful for high dimensional datasets where the relationships are obscured and lack intuitive interpretation. Here, creating a parsimonious regression will limit extraneous information that can corrupt the determination of the regression coefficients. On the other hand, the second element is often used to create a tool for inference and so the latent variables may be derived to have significant physical meaning. The two goals are complementary. In linear regression, the process that produces the samples is generally assumed stationary as the estimation is purely about finding how the entity (row) space is transformed between the $X$ and $Y$ domain, where the sample (column) space is used for implicit averaging. Latent variable regression then first determines an appropriate entity space for the regression which yields a generalised inverse. Moreover, using constraints to enforce physical meaning somewhat modifies the column space to better fit to the linear regression model and “de-noise” the measured data. In this scenario prior information of the physical properties of the variables is required. Therefore, the latent variable approach for regression methods offers a significant range of developments based upon searching for an optimal subspace and physical meaning. In this thesis, the focus is placed on methods to determine a latent entity subspace.

### 3.3 The method of partial least squares

To recap, the linear regression problem

$$\hat{Y} = XB, \quad B = X^+ Y,$$

becomes ill-posed for cases where the matrix of independent variables, $X$, becomes sub-rank. Latent variable regression methods have become extremely popular as a method for solving such a problem. The primary function is to discover a signal subspace within the
data that admits a form where the generalised inverse is straightforward to obtain. From this platform further optimisation and inference can be applied to the latent variables to de-noise or enforce some physical meaning. The method of PCR has been introduced which selects the $r$ most significant principal components for the calculation of the pseudoinverse (2.13) and thus regularises the problem. This is equivalent to selecting a sub-rank basis in the input matrix $X$. However, this basis is selected without consideration of whether the components have any impact on the output matrix $Y$. To this end, the method of partial least squares (PLS) has been developed [26] [66] [67]. The PLS latent components are selected so as to explain the joint dynamics (shared latent variables) between $X$ and $Y$, thus yielding a more parsimonious model than PCR.

\subsection{The NIPALS algorithm for PLS}

The PLS decomposition is performed through the factorisations [67]

\begin{align}
X &= TP^T, \quad (3.3) \\
Y &= TC^T, \quad (3.4) \\
Y &= UQ^T, \quad (3.5)
\end{align}

where $T \in \mathbb{R}^{N \times r}$ is the matrix which comprises $r$ latent variables in $X$ with the loadings given by the matrix $P \in \mathbb{R}^{m \times r}$ while $U \in \mathbb{R}^{N \times r}$ is the corresponding matrix of $r$ latent variables in $Y$ with loadings given by the matrix $Q \in \mathbb{R}^{p \times r}$. The matrix $C \in \mathbb{R}^{p \times r}$ describes the “inner-relation” of PLS, that is, the regression between $Y$ and $T$, and indicates the extent to which the latent variables, $T$, are good descriptors of both $X$ and $Y$. In each case, these decompositions represent an approximation of the data matrices from sum of rank-1 components as in Figure 3.2. These relations demonstrate the utility of the PLS solution as both a regression and a component analysis tool, which offers new data analysis opportunities. For example, Abdi [68] shows that plotting the latent variables against one another (through biplots) reveals information about different groups within data.

The multivariate PLS solution is typically calculated by the iterative NIPALS algorithm [67], outlined in Algorithm 3. Each iteration calculates and stores a rank-1 approximation of $X$ and $Y$, and on completion of the algorithm, this results in the relations (3.3)-(3.5). In order to calculate these rank-1 approximations, a score vector $t$ is found as the component in $X$ which has the maximum cross-covariance with $Y$. This is achieved by finding a weight vector, $w$, that yields the maximal projection of the cross-covariance matrix $X^T Y$. In other words, the solution, $w$, arises from the optimisation problem

$$w = \arg \max_{\|w\| = 1} \|w^T X^T Y\|_2^2,$$

and is given as the eigenvector which corresponds to the largest eigenvalue of the matrix $X^T YY^T X$. This matrix is symmetric and positive-semidefinite, hence its eigenvectors are orthogonal and represent a valid basis for the cross-covariance between $X$ and $Y$ [24]. The “score” of this vector is given by

$$t = Xw,$$

and represents a latent variable in $X$, which is optionally normalised to $t^T t = 1$ so that
\[ T^T T = I \]. The NIPALS algorithm next calculates the regressions of \( X \) and \( Y \) to the computed \( t \), resulting in the rank-1 approximations \( \tilde{X} = tp^T \) and \( \tilde{Y} = tc^T \). The score vector in \( Y \) corresponding to \( t \) is \( u = Yc \) and the rank-1 approximation \( \tilde{Y} = uq^T \) is then calculated. Before proceeding to the next iteration, the data matrices \( X \) and \( Y \) are “deflated”, that is, the rank-1 contribution of the current score \( t \) is removed from \( X \) and \( Y \), which makes the components calculated in the following iteration orthogonal to the already extracted components. The vectors \( w, t, p, c, u \) and \( q \) are stored in the columns of their respective matrices and form the PLS decompositions in (3.3)-(3.5). The algorithm is iterated until all significant components are found, with the stopping criterion typically based on a negligible contribution of a new component to the variance in the prediction of \( Y \), found through cross-validation.

**Algorithm 3.** The NIPALS algorithm for PLS

1: Initialise: \( X_1 = X \), \( Y_1 = Y \)
2: for \( i = 1, \ldots, r \) do
3: \( w_i = \text{Eig}_{\text{max}} \{ X_i^T Y_i Y_i^T X_i \} \)
4: \( t_i = X_i w_i \)
5: \( c_i = Y_i^T t_i / t_i^T t_i \)
6: \( u_i = Y_i c_i \)
7: \( p_i = X_i^T t_i / u_i^T u_i \)
8: \( q_i = Y_i^T u_i / u_i^T u_i \)
9: \( X_{i+1} = X_i - t_i p_i^T \), \( Y_{i+1} = Y_i - t_i c_i^T \)
10: Store \( t_i, u_i, p_i, q_i, c_i \) and \( w_i \)
11: end for

In summary, the NIPALS PLS-regression algorithm provides an approximation of \( X \), denoted by \( \tilde{X} \), in the form

\[
\tilde{X} = T(P^T W)W^T,
\] (3.8)

which is based on a linear combination of columns (given by each vector \( w \)) from a repeatedly deflated \( X \), accounted for by the matrix \( P^T W \) which has an upper-triangular form [69]. The generalised inverse of the matrix \( \tilde{X} \) is now straightforward to calculate due to its structure, as it is an orthogonal matrix multiplied by an upper-triangular matrix multiplied by orthogonal matrices, which are all straightforwardly invertible. Hence, the generalised inverse is calculated (through established methods [39]) as

\[
\tilde{X}^+ = W(P^T W)^{-1} T^T
\] (3.9)

to give the PLS solution \( B_{\text{PLS}} = \tilde{X}^+ Y \).

**Remark 10.** The PLS-regression calculates a joint decomposition (in a second-order statistical sense) of variables \( X \) and \( Y \) in order to produce a solution of a regularised generalised inverse problem.
3.3.2 Other methods of PLS

The SIMPLS algorithm

An alternative PLS algorithm for regression was developed by de Jong [70] and is known as the SIMPLS (simple PLS) algorithm. As opposed to the NIPALS algorithm the data matrices $X$ and $Y$ are not explicitly deflated on each iteration. Instead the scores are obtained directly as a linear combination of the original matrix $X$ and the impact is removed from the cross-covariance matrix $S$. The basis vector $r_i$, calculated in each iteration, is determined as the largest eigenvector of $S_i S_i^T$. The score is then obtained as $t_i = X r_i$ and regressed to find the rank-1 approximation as

$$p_i = \frac{X^T t}{t^T t}.$$

The impact of this and all previously obtained scores is removed through subtracting the projection onto the cross-covariance matrix to give the update

$$S_{i+1} = S_i - P (P^T P)^{-1} P^T S_i.$$

In this way the scores are orthogonal to their predecessors. The SIMPLS algorithm is summarised in Algorithm 4. Further algorithmic developments, along with relations to the output matrix $Y$ can be implemented as in NIPALS but are not central to the computation. The regression coefficient matrix for SIMPLS is then determined as

$$B_{SIMPLS} = R T^+ Y.$$

Algorithm 4. The SIMPLS algorithm for PLS

1: Initialise: $S_1 = X^T Y$
2: for $i = 1, \ldots, r$ do
3:   $r_i = \text{Eig}_{\text{max}} \{ S_i S_i^T \}$
4:   $t_i = X r_i$
5:   $p_i = \frac{X^T t}{t^T t}$
6:   Store $t_i$, $r_i$ and $p_i$ into the matrices $T$, $R$ and $P$
7:   $S_{i+1} = S_i - P (P^T P)^{-1} P^T S_i$
8: end for

PLS for cross-covariance analysis

The PLS algorithms introduced so far are intended for producing a regularised regression solution. To this end, they aim to create the most parsimonious representation of the joint information between the $X$ and $Y$ blocks, whereby the latent variable score vectors are calculated iteratively and are orthogonal to each other. This minimises the information shared by each vector. In other data analysis circumstances it may be desirable to directly model the directions of the cross-covariance of the data matrices $X$ and $Y$.

The first such PLS method is known as PLS mode-A which is another form of the NIPALS algorithm and the basis and PLS decompositions (3.3)-(3.5) are calculated in the
same way as already introduced in Section 3.3.1 [24]. The deflation step at the end of each iteration in Algorithm 3 is modified for PLS Mode-A to be

\[ X_{i+1} = X_i - t_i p_i^T, \quad Y_{i+1} = Y_i - u_i q_i^T, \]

which means that a the impact of the score created as a combination of the columns of \( Y \) (the vector \( u_i \)) is removed from the \( Y \) block rather than the score generated from the columns of \( X \). This means that the relationship is “symmetric” and as a consequence the scores are no longer orthogonal to each other [24]. However, the latent variables are produced within their original \( X \) and \( Y \) spaces and their loadings highlight the direction of the cross-covariance and as such models the relationship between data blocks.

Another symmetric PLS algorithm exists termed PLS-SB [24]. This algorithm is calculated from the complete SVD of the initial cross-covariance matrix \( S = X^T Y = W \Sigma V^T \). The scores in the input data block \( X \) are given by \( T = XW \) and the respective scores in the output \( Y \) are given as \( U = YV \).

The PLS is a methodology based upon finding a latent representation which demonstrates the maximal cross-covariance between two data blocks. A symmetric PLS method will favour the examination of the directions of the cross-covariance between two data blocks. On the other hand, the PLS-regression algorithms, NIPALS and SIMPLS, aim to find the most parsimonious subspace in which to calculate a regression and so the regularisation is at the heart of the calculation.

We have introduced the PLS algorithms so far as generic multivariate methods where both \( X \) and \( Y \) have dimensions greater than one. This is often known as PLS2. On the other hand, when the output data block is a univariate vector \( y \), there are several other methods of calculation which are equivalent to the results provided so far and this is known as PLS1 [71].

### 3.3.3 Analysis of the PLS result

The orthogonality of the PLS residuals

A key result for the NIPALS PLS is that the model space is orthogonal to the residuals [72]. This result was derived with a univariate output in mind, however, it does apply for multivariate outputs. The PLS model is given by the decompositions

\[ X = TP^T, \quad Y = TC^T, \]

The residual is the approximation error between the original data matrices and the PLS decomposition. In other words, the PLS model residual is the \((i + 1)\)-th iteration where \( i \) is the number of PLS components so far selected

\[ X_{i+1} = X - T_i P_i^T, \quad Y_{i+1} = Y - T_i C_i^T, \]
3.3. The method of partial least squares

The matrices \( P_i \) and \( C_i \) are given by the least squares regressions

\[
P_i = X^T T_i (T_i^T T_i)^{-1}, \quad C_i = Y^T T_i (T_i^T T_i)^{-1},
\]

which leads to form for the residuals

\[
X_{i+1} = (I - P_{t,i}) X, \quad Y_{i+1} = (I - P_{t,i}) Y,
\] (3.10)

where the matrix \( P_{t,i} = T_i (T_i^T T_i)^{-1} T_i^T \) is a projection matrix onto the space spanned by the \( i \) scores \( T_i \). The NIPALS PLS result, hence, produces a model for the data matrices \( X \) and \( Y \) which spans an orthogonal subspace to that spanned by the residuals, an important property for a regression algorithm.

The convergence of the PLS algorithm for a univariate output

There does not currently exist a proof for the convergence of the NIPALS PLS algorithm for a general case where the inputs \( X \) and outputs \( Y \) are multivariate. For a univariate output, \( y \), it has been shown by Helland [73] that the vectors \( w \) span a space defined by the vectors in a Krylov sequence \( K_i = (s, Ss, \ldots, S^{i-1}s) \) (defined by the Krylov space \( K_i(s, s) = \text{span}(K_i) \)) where \( s = X^T y \), \( S = X^T X \) and \( i \) is the number of components, while the \( t \) vectors span a Krylov sequence, \( K_i \), where \( s = XX^T y \) and \( S = XX^T \) [72] [71].

For a univariate output the cross-covariance matrix on iteration \( i \) is given as \( X_i^T y y^T X_i \) and so the basis vector is given as

\[
w_i = \frac{X_i^T y}{||X_i^T y||},
\]

that is, the eigenvector is proportional to the vector \( X_i^T y \). Note that the score vector \( t_i \) is calculated as \( t_i = X_i w_i \) and so \( t_i \propto X_i^T y \). Using the equation for the PLS residuals (3.10) this leads to the recurrence equation for the PLS scores as

\[
t_{i+1} \propto (I - P_{t,i}) XX^T (I - P_{t,i}) y.
\] (3.11)

**Proposition 1.** The space spanned by the vectors \( K_i = [s, SK_{i-1}] \) (where \( s = XX^T y \), \( S = XX^T \) and \( K_1 = [s] \)) forms a basis for the scores matrix \( T \) obtained from the NIPALS PLS in Algorithm 3.

**Proof.** The aim is to show that the columns of \( T_i \) are produced from a linear combination of the columns of \( K_i \). The proof is obtained by induction in the same way as the proof of Proposition 3.1 in [73]. It has been shown that \( t_1 \propto s = K_1 \) and therefore our hypothesis is true for the base case \( i = 1 \). We then assume that it is true that the columns of the matrix \( T_i \) are a linear combination of the column vectors in \( K_i \). The matrix \( P_{t,i} \) is therefore also a combination of the vectors in \( K_i \). The recursion in (3.11) gives

\[
t_{i+1} \propto s - S P_{t,i} y - P_{t,i} s + P_{t,i} S P_{t,i} y,
\]

Since the matrix \( P_{t,i} \) is a linear combination of the columns of \( K_i \), the vector \( t_{i+1} \) is
then a linear combination of the columns of $K_{i+1}$ which is $[s, SK_i]$ and so if $T_i$ is a linear combination of the columns of $K_i$ then the columns of $T_{i+1}$ are a linear combination of the columns of $K_{i+1}$ which proves Proposition 1.

A similar, equivalent, proof can be constructed to prove that the vectors $w$ span a space defined by the vectors in a Krylov sequence $K_i = (s, Ss, \ldots, S^{i-1}s)$ where $s = X^T y$ and $S = X^T X$ [73]. This result leads to alternative algorithms for which to calculate a PLS1 solution based on the eigenvector decomposition of the Krylov space [71].

### 3.3.4 The core components of a PLS algorithm

The key characteristics of the NIPALS PLS in Algorithm 3 are that it is iterative, eigenvector-centric and based on capturing the joint second-order information. Each iteration consists of four major steps:

1. the cross-covariance structure,
2. the eigendecomposition of the cross-covariance structure,
3. the calculation of the score,
4. the latent variable decomposition and deflation.

The choice in how these steps are computed dictate the qualities of the solution and therefore the aim of the implementation. For example, the latent variable decomposition/deflation step in Algorithm 3 is often changed to $Y_{i+1} = Y_i - u_i q_i^T$ (PLS Mode-A [24]) which means that the relationship between the $X$ and $Y$ blocks is symmetric and therefore more appropriate for correlation analysis. The generalisation of this algorithm for complex-, quaternion- and tensor-valued data then requires the consideration of these steps in the context of the algebra and operations of the respective multidimensional method.

### 3.3.5 Example: PLS decomposition of circulant matrices

The PLS algorithm is a powerful data analysis tool. The ability to determine the joint subspace between two data blocks through identifying the largest cross-covariance components has many applications. A toy example is introduced here to show how latent variable regression using the NIPALS PLS Algorithm 3 can provide physically meaningful results and take advantage of structure within data.

Consider a circulant matrix, $C$, defined as

$$C = \begin{bmatrix}
  c_0 & c_1 & c_2 & \cdots & c_{n-1} \\
  c_{n-1} & c_0 & c_1 & \cdots & c_2 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  c_1 & \cdots & c_{n-1} & c_0 & c_1 \\
  \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad (3.12)$$
where each row in \( C \) is a circular shift of the vector \( c = [c_0, c_1, \ldots, c_{n-1}] \). The circulant matrix, \( C \), can be represented by the factorisation

\[
C = F^H \Lambda F,
\]

where \( \Lambda \in \mathbb{C}^{n \times n} \) is a matrix containing the eigenvalues of \( C \) on the diagonal and \( F \in \mathbb{C}^{n \times n} \) is the DFT matrix

\[
(F)_{qp} = \frac{1}{\sqrt{n}} \exp(2j\pi qp/n), \quad p = 0, 1, \ldots, n - 1 \text{ and } q = 0, 1, \ldots, n - 1.
\]

As a result, the eigenvalues of the matrix \( C \) are also the DFT coefficients for a the signal \( c \) \[74\]. This is intuitive as the eigenvector equation is given as \( Cv = \lambda v \) which, owing to the structure of \( C \), is a set of simultaneous difference equations.

The SVD of \( C \) is then given

\[
C = U|\Lambda|V^H,
\]

where \( V = \text{Re}(F) + \text{Im}(F) \in \mathbb{R}^{n \times n} \) is an addition of the real and imaginary parts of the corresponding DFT basis vector and \( U \in \mathbb{R}^{n \times n} \) is the same matrix with the phase dictated by the corresponding eigenvalues \[75\], that is

\[
U = V(\text{Re}(D\Lambda) - \Pi \text{Im}(D\Lambda)),
\]

where \( D\Lambda \in \mathbb{R}^{n \times n} \) is defined by \( \Lambda = |\Lambda|D\Lambda \) and

\[
\Pi = \begin{bmatrix}
1 & 0 \\
0 & \ddots \\
0 & 1
\end{bmatrix} \in \mathbb{R}^{n \times n}.
\]

This result essentially means that performing PCA on a circulant matrix is equivalent to taking the DFT of the signal used to create the matrix. The latent variables are then in pairs and are the significant Fourier bases of the signal.

On each iteration the NIPALS PLS Algorithm 3 performs the SVD of the cross-covariance matrix \( S = X^TY \). Consider a case where the input \( X = C_1 \) and output \( Y = C_2 \) are circulant matrices. If \( C_1 = C_2 \in \mathbb{R}^{n \times n} \) then the PLS decomposition is the square of the SVD of the circulant matrix and so performs a similar DFT type decomposition. In a general case, the PLS decomposition will therefore produce a joint decomposition where the PLS scores are the common frequency of interest. This can be implemented for a case where two measurements are acquired of the same process containing significant harmonics. To this end, two signals, \( c_1 \) and \( c_2 \), were simulated as

\[
s = \sin(2\pi f_1 t) + \sin(2\pi f_2 t), \quad c_1 = s + n_1, \quad c_2 = s + n_2,
\]

where \( f_1 = 1 \text{ Hz}, \ f_2 = 2 \text{ Hz}, \ n_1, n_2 \sim \mathcal{N}(0, 1) \) and \( t = [0, 1/fs, \ldots, N/fs - 1/fs] \) with \( fs = 100 \text{ Hz} \) and \( N = 1000 \). The vectors \( c_1 \) and \( c_2 \) were folded into circulant matrices.
Figure 3.3. PLS for de-noising sinusoids.

$C_1$ and $C_2$ as in (3.12) and input into the PLS Algorithm 3 as $X = C_1$ and $Y = C_2$ with $r = 4$. The PLS sub-rank decompositions (3.3) and (3.4) are reconstructions of the circulant matrices, $\hat{C}_1 = TP^T$ and $\hat{C}_2 = TC^T$, and are unfolded to give $\hat{c}_1$ and $\hat{c}_2$. Figure 3.3 shows that the estimates $\hat{c}_1$ and $\hat{c}_2$ provide “de-noised” versions of the sinusoidal components.

3.4 Total Least Squares

A latent variable regression method aims to produce a regularisation of the linear regression problem through finding a generalised inverse of the independent variables $X$. Furthermore, the decomposition is often intended to enhance the physical meaning of the components to aid the inference of the model. Whilst the introduced PLS produces a joint decomposition by consideration of the cross-covariance between the $X$ and $Y$ blocks,
it does not directly modify the sample (column) space. As a result, the PLS-regression method also makes the assumption that the linear regression model error is predominantly in the dependent variables $Y$, as stated in Remark 2. The method of total least squares (TLS), however, assumes equal error in both $X$ and $Y$ [76] and instead minimises the square error orthogonal to the prediction and the observed $Y$ described by the optimisation problem

$$
\arg\min_{\Delta X, \Delta Y} ||[\Delta X \Delta Y]||_F \text{ such that } (X + \Delta X) \hat{B}_{TLS} = Y + \Delta Y, \tag{3.13}
$$

where $X$ and $\Delta X \in \mathbb{R}^{N \times p}$, $Y$ and $\Delta Y \in \mathbb{R}^{N \times q}$ and $X$ and $B_{TLS} \in \mathbb{R}^{p \times q}$. The problem can be displayed as

$$
\begin{bmatrix}
X + \Delta X & Y + \Delta Y
\end{bmatrix}
\begin{bmatrix}
\hat{B}_{TLS} \\
I_q
\end{bmatrix} = 0. \tag{3.14}
$$

To find $\hat{B}_{TLS}$ we consider the SVD of the augmented matrix $[X \ Y]$ given by

$$
\begin{bmatrix}
X & Y
\end{bmatrix} = \begin{bmatrix}
U_X & U_Y \\
0 & \Sigma_Y
\end{bmatrix} \begin{bmatrix}
\Sigma_X & 0 \\
0 & \Sigma_Y
\end{bmatrix} \begin{bmatrix}
V_{XX} & V_{XY} \\
V_{YX} & V_{YY}
\end{bmatrix}^T,
$$

setting $\Sigma_Y = 0$ and using only the first $p$ singular values, $\Sigma_X$, simplifies the problem to

$$
\begin{bmatrix}
X + \Delta X & Y + \Delta Y
\end{bmatrix}
\begin{bmatrix}
V_{XY} \\
V_{YY}
\end{bmatrix} = 0,
$$

which gives

$$
\begin{bmatrix}
X + \Delta X & Y + \Delta Y
\end{bmatrix}
\begin{bmatrix}
V_{XY} \\
V_{YY}
\end{bmatrix} = 0.
$$

Post-multiplying by $[V_{YY}^{-1}, V_{YY}^{-1}]^T$ yields an equation of the form (3.14) and so $\hat{B}_{TLS} = V_{XY}V_{YY}^{-1}$. The TLS solution then calculates the perturbations $\Delta X$ and $\Delta Y$ (which act as estimates for the noise in both $X$ and $Y$ blocks) in order to make the regression equation $Y = XB$ exact. Therefore, it takes into account the situation when $X$ and $Y$ both contain noise. Note that this model is equivalent to PCA performed on a matrix of the concatenated independent and dependent variables, $X$ and $Y$.

The perturbations chosen are not unique but are minimal in the Frobenious norm sense. The methodology employed can be expanded to perturbations with different constraints such as non-negativity [63] [62] which have proved popular in image processing problems. While extremely powerful for model understanding, the results of such techniques are not always general and are application specific as prior information is required about the physical structure of the problem.

### 3.5 Optimality of latent variable regression

A novel framework in which to consider the optimality of a latent variable approach to regression is now introduced. This is provided by the concept of forwards and backwards regression. This is used to determine the optimal basis for a component search type latent
variable regression such as PLS. The limits of such a technique are then derived.

### 3.5.1 The Concept of Forwards and Backwards Regression

In order to examine the impact of latent variables on the generalised inverse and hence the regression estimator we introduce the concept of forwards and backwards regression. The forwards regression is given as

\[
Y = XB_f, \quad (3.15)
\]

as defined in (2.1) where \( X \in \mathbb{R}^{N \times m} \), \( Y \in \mathbb{R}^{N \times p} \) and \( B_f = X^+Y \in \mathbb{R}^{m \times p} \). Now consider the backwards regression

\[
X = YB_b, \quad (3.16)
\]

where \( B_b = Y^+X \in \mathbb{R}^{p \times m} \) and the goal is now to predict the inputs \( X \) from the outputs \( Y \). In an ideal situation (there is no noise in either measurements) these equations will be consistent.

**Proposition 2.** If the forwards regression (3.15) and backward regressions (3.16) are consistent, then the backwards regression coefficients matrix \( B_b \) is a generalised inverse of the forwards regression coefficient matrix \( B_f \).

**Proof.** If the equations (3.15) and (3.16) are consistent then

\[
Y = (YB_b)B_f,
\]

\[
XB_f = XB_fB_bB_f,
\]

and

\[
X = (XB_f)B_b,
\]

\[
YB_b = YB_bB_fB_b.
\]

This mirrors the generalised inverse criteria (2.8) and (2.9).

In an ideal regression scenario, therefore, a forwards and backwards estimator of the data matrices \( X \) and \( Y \) would satisfy the criteria for a generalised inverse. However, (3.15) and (3.16) are not generally consistent as they can be corrupted by noise. This view can be used to examine the optimal latent variable choice and the impact of this corruption due to noise on the regression calculation. Note that the TLS solution satisfies this condition as it considers that the two blocks should share the same sample space.

**The optimal basis for latent variable regression**

Given a set of variables, \( X \) and \( Y \), it is of interest to consider what the optimal basis for a latent variable regression. By Proposition 2, the forwards and backwards estimators should satisfy the criteria for a generalised inverse themselves. An optimal basis will produce a solution that is as close as possible to these criteria. This leads to the optimisation
problems
\[
\hat{B}_f, \hat{B}_b = \arg \min_{B_f \neq 0, B_b \neq 0} \| \hat{B}_f \hat{B}_b - \hat{B}_f \|^2_F, \tag{3.17}
\]
\[
\hat{B}_f, \hat{B}_b = \arg \min_{B_f \neq 0, B_b \neq 0} \| B_b \hat{B}_f - \hat{B}_b \|^2_F. \tag{3.18}
\]

Consider the NIPALS in Algorithm 3, where on each iteration a basis vector, \( w \), is found in order to produce a rank-1 approximation of \( X \) and \( Y \). This algorithmic process can be considered as a general blueprint for a latent variable regression algorithm, from which an optimal choice of the basis vector, \( w \), can be determined. The forwards and backwards regression latent variables (\( t_f \) and \( t_b \)) are then given as
\[
t_f = Xw_f, \quad t_b = Yw_b.
\]

The respective forwards and backwards regression coefficient matrices, \( \hat{B}_f \) and \( \hat{B}_b \), for each latent variable are derived by following the PLS decompositions (3.3)-(3.4) for each score
\[
\hat{Y} = t_f c_f^T, \quad \hat{X} = t_b c_b^T
\]
\[
Y = Xw_f c_f^T, \quad X = Yw_b c_b^T
\]
\[
c_f = \frac{\hat{Y}^T t_f}{\hat{t}_f^T \hat{t}_f}, \quad c_b = \frac{\hat{X}^T t_b}{\hat{t}_b^T \hat{t}_b}
\]
\[
\hat{Y} = X \hat{B}_f, \quad \hat{X} = Y \hat{B}_b
\]
\[
\hat{B}_f = w_f c_f^T, \quad \hat{B}_b = w_b c_b^T
\]
\[
\hat{B}_f = \frac{w_f^T X^T Y}{w_f^T X^T X w_f} \quad \hat{B}_b = \frac{w_b^T Y^T X}{w_b^T Y^T Y w_b}.
\]

Substituting the resultant \( \hat{B}_f \) and \( \hat{B}_b \) into (3.17) gives
\[
w_f, w_b = \arg \min_{|w_f| = |w_b| = 1} \rho^2 \left| \frac{w_f^T X^T Y}{w_f^T X^T X w_f} \frac{w_b^T Y^T X}{w_b^T Y^T Y w_b} - \frac{w_f^T X^T Y}{w_f^T X^T X w_f} \right|^2_F. \tag{3.19}
\]

The result (3.19) is zero for \( \rho^2 = 1 \) and hence minimised. To this end, a dual problem of (3.19) is
\[
\rho = \arg \min \| \rho^2 - 1 \|_2^2.
\]

Now consider that
\[
\rho^2 = \frac{w_f^T X^T Y w_b}{\sqrt{w_f^T X^T X w_f} \sqrt{w_b^T Y^T Y w_b}} \frac{w_b^T Y^T X w_f}{\sqrt{w_f^T X^T X w_f} \sqrt{w_b^T Y^T Y w_b}}. \tag{3.20}
\]

It can be seen that \( \rho^2 \) is the square of the correlation coefficient \( \rho \) and has a maximum value of one due to the Cauchy-Schwartz inequality. As a result, finding the basis that maximises the correlation between the two rank-one approximations of \( X \) and \( Y \) is equivalent to finding the basis that produces the forwards and backwards estimators which are the closest to being generalised inverses.
This problem is solved by CCA (Canonical Correlation Analysis) where the basis for \( X, \mathbf{w}_f \), is given by the eigenvectors of \( \frac{X^TYY^TX}{X^TXX^TY} \) and the basis for \( Y, \mathbf{w}_b \), is given by the eigenvectors of \( \frac{Y^TXY}{X^TXY} \). This is intimately related to PLS and they are shown to be part of a family called canonical ridge analysis [24] which is described by the optimisation problem (3.21). This is PLS when \( \lambda_x = \lambda_y = 1 \) and CCA when \( \lambda_x = \lambda_y = 0 \).

\[
\mathbf{w}, s = \arg \max_{|\mathbf{w}| = |s| = 1} \left\| \mathbf{w}^T X^T Y s \right\|^2_2 \quad \frac{(1 - \lambda_x)\mathbf{w}^T X^T X \mathbf{w} + \lambda_x)((1 - \lambda_y)Y^T Y s + \lambda_y)}{((1 - \lambda_x)\mathbf{w}^T X^T X \mathbf{w} + \lambda_x)(((1 - \lambda_y)Y^T Y s + \lambda_y))}.
\]

This result intuitively shows that the best components to select for regression are those which have the highest correlation. The correlation metric normalises for the power of a variable. In a practical implementation this may then bias against the selection of a component which is more significant in the explanation of the joint process, which the PLS criterion would prefer.

De Moor [77] provides an analysis of the SVD result for the case where the matrix \( X \) is made up of a low-rank, signal subspace corrupted by an uncorrelated noise process. The result shows that, under certain conditions, the spatial/entity basis of the signal can be accurately recovered but the sample space of the signal cannot be fully determined. This means that for a general case the correlation between the input and output blocks will not be perfect. As a result, calculated regression coefficients will have an erroneous value but will provide an accurate spatial transform.

### 3.6 Online PLS

It has been shown that a latent variable decomposition enables only the signal within the true entity subspace to be utilised for a regularised calculation of the regression coefficients. The PLS method is shown as a method to do this through a modification of row space. However, the method does not directly attempt to remove any noise perturbation within the sample space which may corrupt the regression coefficients. Moreover, non-stationary entities cannot be taken into account by the PLS algorithm. In addition, block-based methods tend to be computationally prohibitive for large scale problems, where gradient-based and online estimators are becoming standard [78].

Although finding a unified solution to the online PLS algorithm is still an open problem, several efforts in this direction have been made. These include, a recursive solution based on a sliding-window type technique [79], incremental solutions specifically designed for dimensionality reduction [80] or solutions limited to univariate data [81]. None of these methods have exploited ideas from stochastic gradient-based learning which has been shown to routinely outperform other methods in large scale settings [78] [82].

To this end, we propose an online PLS algorithm which is capable of finding a common subspace for multivariate data which are collected in an online fashion. The proposed online PLS algorithm employs well-known ideas, such as recursive computation of empirical cross-covariance matrices and stochastic gradient maximisation of the Rayleigh quotients,
3.6. Online PLS

The NIPALS algorithm outlined in Algorithm 3 requires block data inputs, $X \in \mathbb{R}^{N \times m}$ and $Y \in \mathbb{R}^{N \times p}$, where $N$ refers to the number of observations. However, in many practical scenarios all the observations, $N$, cannot be obtained simultaneously but are observed as streaming values, that is, at every time instant, $n$, only one row of $X$ and $Y$ is observed.

Therefore, the key difference between the NIPALS formulation in Algorithm 3 and the proposed online-NIPALS (OL-PLS) is that in the online setting, at each time instant $n$, we only have access to the $n$-th row of the matrices $X$ and $Y$, denoted by $x_n \in \mathbb{R}^{1 \times m}$ and $y_n \in \mathbb{R}^{1 \times p}$, given as

$$x_n \overset{\text{def}}{=} X(n,1:m), \quad y_n \overset{\text{def}}{=} Y(n,1:p). \quad (3.22)$$

The OL-PLS algorithm is next developed through recursive solutions to the Steps 3-9 in Algorithm 3. To this end, the empirical cross-covariance matrix, $S = X^T X$, is first expressed as a sum of rank-1 matrices $x_n^T y_n$, for $n = 1, \ldots, N$, to give

$$S_{1,N} = \sum_{n=1}^{N} x_n^T y_n, \quad (3.23)$$

where the subscript $N$ in $S_{1,N}$ indicates the number of observations used to compute the sample cross-covariance matrix in (3.23). Then, (3.23) can be computed recursively as

$$S_{1,n} = S_{1,n-1} + x_n^T y_n, \quad (3.24)$$

and the same procedure can also be used for the deflated cross-covariance matrices, to give

$$S_{d,n} = S_{d,n-1} + x_{d,n}^T y_{d,n}, \quad d = 1, \ldots, r, \quad (3.25)$$

where $x_{d,n}$ and $y_{d,n}$ denote the deflated data vectors which constitute the deflated data matrices in Step 9 of Algorithm 3. A forgetting factor, $0 < \lambda < 1$, can be introduced in (3.24) and (3.25) to account for non-stationary data statistics, which yields

$$S_{d,n} = \lambda S_{d,n-1} + (1 - \lambda) x_{d,n}^T y_{d,n}. \quad (3.26)$$

The recursion in (3.26) provides a reliable estimate owing to the inherent averaging of the cross-covariance, which reduces the impact of outliers.

To explain the principle of OL-PLS, recall that the block-based PLS solution uses the empirical cross-covariance matrix, $S_1$, to identify a basis to describe a subspace in $X$ that exhibits co-variation with $Y$. This basis is used to produce the corresponding latent variables, which are the columns of the score matrix $T$, these in turn provide a robust prediction of the output $Y$ from the input $X$. The most important step for an online
PLS algorithm is to produce an estimate of the eigenvectors $w_{d,n}$, $d = 1, 2, \ldots, r$, at each time instant $n$. With an increase in $n$, these estimates should converge to a set of vectors equivalent to the NIPALS solution that describe the same joint subspace. Upon calculating the eigenvector, $w_{d,n}$, the remaining variables can be straightforwardly computed in an online fashion.

The online PLS is now fully equipped to produce estimates of the covariance matrix $S_{d,n}$, which is then used to produce the eigenvectors $w_{d,n}$, at each time instant, $n$, and for every deflation iteration, $d = 1, 2, \ldots, r$. The estimation of $w_{d,n}$ from (3.6), when performed in an online setting, can therefore be considered as a problem of adaptively estimating the largest eigenvectors of the matrix $S_{d,n}S_{d,n}^T$.

### 3.6.2 Adaptive Estimation of the OL-PLS Variables

Consider the Rayleigh quotient, which for a matrix and vector pair $(A, w)$ is defined as [84]

$$R(A, w) \overset{\text{def}}{=} \frac{w^T Aw}{w^T w}. \quad (3.27)$$

The vector $w$ that maximises (3.27) represents the maximum eigenvector of the matrix $A$. Therefore, the estimation of the maximum eigenvector, $w_{d,n}$, of the matrix $S_{d,n}S_{d,n}^T$ in (3.6) can be accomplished using the maximisation of the Rayleigh quotient in (3.27) for the matrix-vector pair $(S_{d,n}S_{d,n}^T, w)$ along with the constraint $\|w\| = 1$. This yields the cost function

$$J_w \overset{\text{def}}{=} \frac{1}{2} \frac{\mathcal{R}(S_{d,n}S_{d,n}^T, w)}{\text{s.t.} \|w\| = 1} = \frac{w^T S_{d,n} S_{d,n}^T w}{2w^T w} + \beta (1 - w^T w), \quad (3.28)$$

where $\beta$ is a Lagrange multiplier. The Rayleigh cost function in (3.28) can be maximised recursively, using gradient ascent methodology [85], in the form

$$w_{d,n+1} = w_{d,n} + \mu \frac{\partial J_w}{\partial w}{\bigg|}_{w = w_{d,n}}, \quad (3.29)$$

where $\mu$ is a step-size which controls the trade-off between convergence speed, stability and steady state error. The gradient $\partial J_w/\partial w$ is calculated as

$$\frac{\partial J_w}{\partial w} = - \left( \frac{\| S_{d,n}^T w \|^2}{\| w \|^2} I - S_{d,n} S_{d,n}^T \right) \frac{w}{\| w \|^2} - \beta w.$$  

The role of the Lagrange optimisation is to keep $\|w\| = 1$ and to ensure the uniqueness of the vector that maximises the Rayleigh quotient. It can be shown that the Lagrange multiplier, $\beta$, is in the form

$$\beta = \frac{w^T S_{d,n} S_{d,n}^T w}{w^T w} (1 - \frac{1}{\| w \|^2}).$$
The OL-PLS algorithm follows the direction of this gradient, estimated from the available variables at the time instant \( n \), to give the OL-PLS update

\[
\mathbf{w}_{d,n+1} = \mathbf{w}_{d,n} - \mu \beta \mathbf{w}_{d,n} + \mu \left( \frac{\| \mathbf{S}_{d,n}^T \mathbf{w}_{d,n} \|^2}{\| \mathbf{w}_{d,n} \|^2} \mathbf{I} - \mathbf{S}_{d,n} \mathbf{S}_{d,n}^T \right) \frac{\mathbf{w}_{d,n}}{\| \mathbf{w}_{d,n} \|^2}. \tag{3.30}
\]

**Remark 11.** The analysis in [85] shows that the gradient ascent step in (3.30) converges to the largest eigenvector of \( \mathbf{S}_d \mathbf{S}_d^T \). This implies that if the estimate of \( \mathbf{S}_{d,n} \) converges to the cross-covariance matrices \( \mathbf{S}_d \) as \( n \to \infty \), then the online estimate, \( \mathbf{w}_{d,n} \) will also converge to the eigenvectors, \( \mathbf{w}_d \) in (3.6).

Upon estimating the eigenvectors, \( \mathbf{w}_{d,n} \), the scores \( t_{d,n} \) from (3.7) are computed as

\[
t_{d,n} = \mathbf{x}_{d,n} \mathbf{w}_{d,n}. \tag{3.31}
\]

The next step is to find the corresponding loading vectors, \( \mathbf{p}_d \) and \( \mathbf{c}_d \), necessary for the deflation. Note that the computation by regression in the block-based NIPALS Algorithm 3 Steps 5 and 7 gives the loading vectors

\[
\mathbf{p}_{d,n} = \frac{\mathbf{x}_{d,n}}{t_{d,n}}, \quad \mathbf{c}_{d,n} = \frac{\mathbf{y}_{d,n}}{t_{d,n}}. \tag{3.32}
\]

However, the deflation of the input and output data using the loadings in (3.32) will yield a null-vector at each time instant. This is obvious from the deflation step, modified for the online case, where

\[
\mathbf{x}_{d+1,n} = \mathbf{x}_{d,n} - t_{d,n} \mathbf{p}_{d,n}^T = \mathbf{x}_{d,n} - t_{d,n} \frac{\mathbf{x}_{d,n}}{t_{d,n}} = 0,
\]

and is a consequence of the fact the available data at each time instant are rank-1. Our solution to this issue is through an online estimation of optimal loading vectors, these are found via gradient descent minimisation of the following cost functions

\[
\mathcal{J}_p = \| \mathbf{x}_{d,n} - t_{d,n} \mathbf{p}^T \|^2, \quad \mathcal{J}_c = \| \mathbf{y}_{d,n} - t_{d,n} \mathbf{c}^T \|^2, \tag{3.33}
\]

for which the gradients are given by

\[
\frac{\partial \mathcal{J}_p}{\partial \mathbf{p}} = - 2 t_{d,n} \left( \mathbf{x}_{d,n}^T - t_{d,n} \mathbf{p} \right), \quad \frac{\partial \mathcal{J}_c}{\partial \mathbf{c}} = - 2 t_{d,n} \left( \mathbf{x}_{d,n}^T - t_{d,n} \mathbf{c} \right).
\]

The online estimates for the loading vectors now take the form

\[
\mathbf{p}_{d,n+1} = \mathbf{p}_{d,n} + \mu t_{d,n} \left( \mathbf{x}_{d,n}^T - t_{d,n} \mathbf{p}_{d,n} \right), \tag{3.34}
\]

\[
\mathbf{c}_{d,n+1} = \mathbf{c}_{d,n} + \mu t_{d,n} \left( \mathbf{x}_{d,n}^T - t_{d,n} \mathbf{c}_{d,n} \right). \tag{3.35}
\]
which are standard least mean square adaptive filters. This solution is equivalent to the block NIPALS algorithm which produces rank-1 least squares estimates of \( \mathbf{X} \) and \( \mathbf{Y} \), denoted by \( \mathbf{X} = \mathbf{tp}^T \) and \( \mathbf{Y} = \mathbf{tc}^T \), for each score on each iteration. Recall that \( \mathbf{tp}^T \) and \( \mathbf{tc}^T \) are then removed from the data blocks \( \mathbf{X} \) and \( \mathbf{Y} \). To this end, the instantaneous score, \( t_{d,n} \), within OL-PLS is used to deflate each \( x_{d,n} \) and \( y_{d,n} \) through the estimates \( \hat{x}_{d,n} = t_{d,n}p_{d,n}^T \) and \( \hat{y}_{d,n} = t_{d,n}c_{d,n}^T \), with the full deflation scheme given by

\[
\begin{align*}
\mathbf{x}_{d+1,n} &= \mathbf{x}_{d,n} - t_{d,n}p_{d,n}^T, \\
\mathbf{y}_{d+1,n} &= \mathbf{y}_{d,n} - t_{d,n}c_{d,n}^T.
\end{align*}
\] (3.36)

In other words, the OL-PLS performs an adaptive estimation of \( p_{d,n} \) and \( c_{d,n} \) in order to converge to the least squares solution.

As a result, the proposed online extension of PLS adaptively estimates \( w_{d,n} \), \( p_{d,n} \) and \( c_{d,n} \) from the inputs \( x_n \) and \( y_n \), which provides the instantaneous estimates of the PLS components \( t_{d,n} \). To this end, the matrices \( S_{d,n} \) are updated recursively based on the deflated inputs \( x_{d,n} \) and \( y_{d,n} \), which are also estimated adaptively as shown in (3.36). The OL-PLS algorithm summarised in Algorithm 5, where the tuning parameters are \( \mu \) and \( \lambda \).

### Algorithm 5. Proposed online PLS (OL-PLS)

1. **Inputs at time instant** \( n \): \( \mathbf{x}_n \in \mathbb{R}^{1 \times m} \) and \( \mathbf{y}_n \in \mathbb{R}^{1 \times p} \)
2. **Initialise**: \( S_{d,1} \leftarrow \mathbf{x}_1^T \mathbf{y}_1 \)
3. **Initialise**: \( p_{d,1}, q_{d,1} \) and \( c_{d,1} \leftarrow 0 \)
4. **Initialise**: \( w_{d,1} \leftarrow \) first column of \( S_{d,1} \)
5. for \( d = 1, \ldots, r \) do
6. \( S_{d,n} = \lambda S_{d,n} + (1 - \lambda)\mathbf{x}_{d,n}^T \mathbf{y}_{d,n} \)
7. \( w_{d,n+1} = w_{d,n} + \mu \frac{\partial J_{S_{d,n}}}{\partial \mathbf{w}_{d,n}^T}, \) using (3.30)
8. \( t_{d,n} = \mathbf{x}_{d,n}w_{d,n} \)
9. \( p_{d,n+1} = p_{d,n} + \mu t_{d,n} \left( \mathbf{x}_{d,n}^T - t_{d,n}p_{d,n} \right) \)
10. \( c_{d,n+1} = c_{d,n} + \mu t_{d,n} \left( \mathbf{x}_{d,n}^T - t_{d,n}c_{d,n} \right) \)
11. \( x_{d+1,n} = \mathbf{x}_{d,n} - t_{d,n}p_{d,n}^T, \quad y_{d+1,n} = \mathbf{y}_d - t_{d,n}c_{d,n}^T \)
12. end for

### 3.6.3 Simulations and Analysis

Simulations on synthetic data were conducted to verify the performance of the proposed OL-PLS algorithm. A particular emphasis was on the accuracy of OL-PLS in an online prediction setting. To this end, we generated \( M = 10 \) realisations of a low-rank input matrix, \( \mathbf{X} \in \mathbb{R}^{N \times 5} \), with rank \( r = 3 \), drawn from a unit variance Gaussian distribution. This input matrix was multiplied by a vector \( \mathbf{b} \in \mathbb{R}^{5 \times 1} \) to produce a univariate output \( \mathbf{y} = \mathbf{Xb} \). The OL-PLS algorithm in Algorithm 5 was then used to estimate the output \( \mathbf{y} \) at each time instant. The performance measure was the empirical mean square error (MSE) computed as

\[
\text{MSE}_n = \frac{1}{M} \sum_{\ell=1}^{M} |y_n^{(\ell)} - \hat{y}_n^{(\ell)}|^2,
\]
where $\hat{y}_n^{(\ell)}$ denotes the $\ell$-th realisation of the OL-PLS output estimate at time $n$. Figure 3.4 (top panel) shows the ensemble average prediction MSE, calculated as a percentage of the total variance in $y$, for $\lambda = 0.9999$ and $\mu = 0.0001$. For successful performance, the OL-PLS solution should be equivalent to that of the NIPALS algorithm, that is, the OL-PLS vectors $w_{d,n}$ should converge to those produced by NIPALS, $w_d$, as $n \to \infty$. To verify this, Figure 3.4 (bottom panel) shows the ensemble average of the MSE of each vector $w_{d,n}$ (as a percentage of its total variance) and the corresponding vector calculated by NIPALS.

### 3.7 Discussion of practical implementations of OLS, PLS and OL-PLS

In this chapter latent variable regression methods have been discussed and introduced for the linear regression model $Y = XB$. It has already been discussed that the model assumes that the dependent variables, $Y$, are predicted as a linear combination of the independent variables $X$. The matrix of coefficients $B$ then describes how much the variation in each column of $X$ corresponds to some variation in each column of $Y$. The linear regression model is considered as the basis for Part A of this thesis, however, its suitability for a certain application depends on whether the data conform to further model assumptions. The main assumptions for a linear regression are:

- The model error is contained in the dependent variables,
- The error variance is constant,
- The errors are independent.

The linear regression model applies for both random variables and deterministic signals as long as the errors are random, uncorrelated and constant. In most real-world applications
Chapter 3. Latent Variable Regression

it is not the case that the acquired data of the independent variables are error free. However, the model typically works well for reasonable SNR and cross-validation would enable a good calibration. Further complexity can be added to this model to cater for a breakdown in these assumptions. For example: weighting, error estimation and through techniques such as the TLS approach was introduced in Section 3.4. In general, it is desireable to use a linear regression approach as the calculation and interpretation is straightforward. For example, kernel based methods attempt to transform a non-linear relationship into a linear problem.

The linear regression model will perform well for any circumstance where the dependent variables are a linear combination of the regressors with added independent Gaussian distributed errors. An example simulation which meets the assumptions can be created by sampling the data from a Gaussian distribution. Common real-world applications for linear regression include economic and biological data.

The PLS method calculates a solution to the linear regression model when the matrix $X$ contains colinear variables. The introduced latent variable decompositions are based on a similar linear model where the approximation of the independent variables $X$ is built from a linear combination of its columns. The approximation for the dependent variables, $Y$ is again the linear regression model and so the PLS uses the same base assumptions. As a result, the added complexity does not aim to remove any time varying noise process but will reject extraneous spatial components. The PLS algorithm is typically applied to problems with a high number of variables and the relationships are not immedatley clear, for example, the chemometric application of predicting a chemical composition from near infrared spectra.

The proposed OL-PLS aims to tackle the problem when the regression may change over the realisations within the data, that is, it exhibits some non-linearity/non-stationarity. This model assumes that the prediction, however, is locally linear. As a result, this algorithm will work well for applications where the underlying (possibly latent) contributions to the dependent variables are changing over time but are still predicted from a linear combination of the variables in the regressors. For example, this can be the case for financial applications.

3.8 Conclusion

This chapter has introduced the concept of latent variables from a data matrix analysis point of view. They are viewed as a change of basis to a representation which separates the information within the original process. As such, the goal is to enable the description of the original entity. For example, the method of PCA is shown to decompose the original matrix into orthogonal components. If the entity is stationary then only the spatial distribution needs to be determined, through the row space.

Consider that a linear regression of the form $Y = XB$ maps the entity space of the matrix $X$ to the entity space of the matrix $Y$. In this way a new entry to the matrix $X$ can be used to predict the respective value in $Y$. Producing a latent variable approximation of the matrix of independent variables $X$ yields an appropriate entity space for the
regression. When the PCA basis is applied for a regression, the regression coefficients for each component describes separate, uncorrelated second-order information. As such, the regression operates on the underlying processes and the structure allows for a regularisation. This is further developed by the method of PLS which finds a latent basis describing the maximal covariance between the $X$ and $Y$ blocks. The PLS algorithm is calculated based on finding successive, orthogonal rank-1 approximations of the data inputs, until the joint variation is sufficiently described. The representation of the data produced is therefore highly parsimonious. In doing so, a generalised inverse of $X$ is found to render the linear regression problem well-posed for ill-conditioned cases. The analysis of the PLS-regression solution is demonstrated and the method was applied for the denoising of sinusoids by exploiting the orthogonal decomposition properties of circulant matrices.

Although the variation between $X$ and $Y$ is considered in PLS (unlike PCR where only the $X$ block is examined), the latent decomposition is produced as a linear combination of the original independent variables $X$. The linear regression assumption that the model error is contained within the measured dependent variables $Y$ (Remark 2) still holds. To this end, the method of TLS was introduced which finds the minimum perturbation in $X$ and $Y$ to produce a consistent relation $Y = XB$. In this case, the sample space is directly modified and alternative constraints on the perturbation offer a way to enforce physical meaning and denoising. However, this result is not general and such an approach is application dependent. This is summarised by the concept of forwards and backwards regression which states that the prediction of the output block from the input should be consistent with the reverse prediction of the input block from the output. This assumption leads to the proof that an optimal basis for regression is that which selects components that represent the maximal correlation between the two data blocks. This is satisfied by the perturbed TLS solution whereas the PLS method, which uses the associated (and more practical) cross-covariance measure, will obtain the closest solution within the original least-squares type regression framework. An algorithm for an Online PLS is then proposed. The extension for use with time-varying, streaming data is aimed at taking into account entities with non-stationary properties.

This thesis will next focus on the development of the NIPALS algorithm for a PLS-regression for use when the data have a multidimensional structure.
Chapter 4

Widely Linear Complex Partial Least Squares

I always like to look on the optimistic side of life, but I am realistic enough to know that life is a complex matter

Walt Disney

4.1 Introduction

The introduced latent variable approach to regression is useful for the regularisation of an ill-posed problem and for decomposing the information into its constituent parts. As such, it is readily applicable for modern applications involving high-dimensional datasets. Furthermore, the multidimensional methods introduced in Section 2.3 are also highly successful at exploiting the rich structure in such datasets. A goal of this thesis is to develop the latent variable regression concepts for use with multidimensional techniques. The combined algorithm creates a latent variable which is no longer just a single vector component but a multivariate source, constrained by the multidimensional algebra. These can then be applied to a variety of regression problems for which the representation is meaningful and allows the regularisation of a regression for such data. The focus in this chapter is specifically placed on complex-valued signal processing which, in recent years, has led to advances in analysis of wind profiles [50], power systems [31] [86], acoustics [87] and communications [29] [30].

We propose the WL-CPLS algorithm as a development of the real-valued PLS algorithm for a widely-linear complex-valued regression. The main technical contributions are

- An NIPALS based algorithm is developed taking into account full complex-valued augmented statistics,

- A method to calculate the widely linear regression coefficients akin to the real-domain PLS algorithm is derived,
• The properties of the WL-CPLS model residuals are determined and the algorithm convergence is proved for a univariate output,

• The WL-CPLS is verified on practical applications of complex-valued covariance matrix diagonalisation and for smart grid frequency estimation.

The aspects of complex-valued statistical estimation are, hence, combined with the core-components of PLS. As a result, this chapter provides a summary of the application of complex-valued latent variable regression.

4.1.1 What should a complex-valued partial least squares be?

The fundamental aim of the PLS algorithm is to provide a regularised linear regression estimator, the extension of which to the complex-domain is given by the widely linear model of the form $Y = XH + X^*G$, as described in Section 2.3.1. To this end, the PLS algorithm for complex-valued data should produce a widely linear regression estimator. The solution is created through a latent variable decomposition of $X$ and based on full joint second-order complex statistics and so must consider the augmented relationships introduced in Section 2.3.1. The PLS-regression solution is calculated through the generalised inverse of $\tilde{X}$ (3.9) which is, by structural design, straightforward to calculate. The so-obtained latent variable decomposition for the complex-valued algorithm will, therefore, also admit a straightforward calculation of the generalised inverse of the augmented data matrix $\tilde{X}$, based on its structure. In doing so, the widely linear regression solution will be regularised. In general, the requirements for the widely linear complex PLS (WL-CPLS) are therefore to:

1. Create a joint latent variable decomposition of the data matrices, $X$ and $Y$, that describes the complete complex second-order statistics;

2. Account for a widely linear relationship between the $X$ and $Y$ data blocks;

3. Obtain approximations $\tilde{X}$ and $\tilde{Y}$ that admit a straightforward and tractable computation of a widely linear regression $Y = \tilde{X}H + \tilde{X}^*G$.

A complex-valued PLS algorithm has already been proposed by Schreier [88]. This method, however, is aimed at dataset cross-covariance analysis and is a generalisation of the PLS-SB approach described in Section 3.3.2. Here, it is shown that this version of the PLS methodology is aimed at finding the individual component directions between datasets and as such the components are not necessarily orthogonal. The result is then not the most parsimonious decomposition of the latent variables.

4.2 A widely linear complex-valued PLS algorithm

To extend NIPALS for a general complex-valued input $X \in \mathbb{C}^{N \times m}$ and output $Y \in \mathbb{C}^{N \times p}$, the components in $X$ must be found that represent the maximal joint second-order information with $Y$. This is achieved in the form of vectors, $w$, which give the “score”
Chapter 4. Widely Linear Complex Partial Least Squares

vectors \( t = \mathbf{X}w \). The augmented representation is then given by

\[
[t, t^*] = [\mathbf{X}, \mathbf{X}^*] \begin{bmatrix} w & 0 \\ 0 & w^* \end{bmatrix}.
\]

This structure is required as the vectors \( \mathbf{w} \) must form an orthogonal basis for the inputs \( \mathbf{X} \). To find the vector \( \mathbf{w} \) that meets this criterion and produces the required structure the isomorphism between \( \mathbb{R}^2 \) and \( \mathbb{C} \), given in (2.21), is employed to transform \( \mathbf{X} \) and \( \mathbf{Y} \) as

\[
\mathbf{X}_{\text{Re}} = \mathbf{X} \Gamma_m, \quad \mathbf{Y}_{\text{Re}} = \mathbf{Y} \Gamma_n,
\]

where \( \Gamma_m \) and \( \Gamma_n \) are defined in (2.22). The matrices \( \mathbf{X}_R \in \mathbb{R}^{N \times m} \), \( \mathbf{Y}_R \in \mathbb{R}^{N \times p} \), \( \mathbf{X}_I \in \mathbb{R}^{N \times m} \) and \( \mathbf{Y}_I \in \mathbb{R}^{N \times p} \) denote the real and imaginary parts of \( \mathbf{X} \) and \( \mathbf{Y} \) respectively. The cross-covariance criterion for the real-valued PLS in (3.6) is then extended to the complex domain as

\[
\mathbf{w}_{\text{Re}} = \underset{||\mathbf{w}_{\text{Re}}||=1}{\arg \max} |\mathbf{w}_{\text{Re}}^T \mathbf{X}_{\text{Re}}^T \mathbf{Y}_{\text{Re}}||^2_2. \tag{4.1}
\]

The solution, \( \mathbf{w}_{\text{Re}} \in \mathbb{R}^{2m} \), is found as the eigenvector of \( \mathbf{X}_{\text{Re}}^T \mathbf{Y}_{\text{Re}} \mathbf{Y}_{\text{Re}}^T \mathbf{X}_{\text{Re}} \) which corresponds to its largest eigenvalue. The resulting vector, \( \mathbf{w}_{\text{Re}} \), is of the form \([\mathbf{w}_R^T, -\mathbf{w}_I^T]^T\) and is transformed back to the complex domain through

\[
[\mathbf{w}^T, \mathbf{w}^H]^T = \Gamma_m \mathbf{w}_{\text{Re}}. \tag{4.2}
\]

Notice that the computation of the vector, \( \mathbf{w} \), in the real domain before converting it back to the complex domain does not affect any benefit gained by the complex representation.

We have seen that the augmented form of a complex-valued variable makes it possible to capture the full second-order statistics for complex-valued data. However, in this case, an SVD of the augmented cross-covariance matrix \( \mathbf{X}^H \mathbf{Y} \), does not provide the orthogonal basis for the \( \mathbf{X} \) block in the structure required for the WL-CPLS algorithm, and the result would not be usable for the NIPALS extension. On the other hand, the proposed calculation of \( \mathbf{w} \) does fulfil this structural requirement. The problem of an appropriate structure for a complex-valued PLS is also addressed in that proposed by Schreier [88], where the computation is also performed in the real-domain before transforming back to the complex-domain. This process, therefore, is implicitly complex-valued and is not equivalent to an application of the NIPALS algorithm on complex data transformed to the real domain.

The so-obtained complex-valued vector \( \mathbf{w} \in \mathbb{C}^{m \times 1} \) caters for full joint second-order statistics between the input \( \mathbf{X} \) and the output \( \mathbf{Y} \). The corresponding component vector, \( \mathbf{t} \), is then calculated as

\[
\mathbf{t} = \mathbf{X} \mathbf{w}.
\]

**Remark 12.** The latent variable, \( \mathbf{t} \), is obtained through a consideration of both cross-covariance and cross-pseudocovariance between the input \( \mathbf{X} \) and output \( \mathbf{Y} \).

The joint approximations of the input \( \mathbf{X} \) and the output \( \mathbf{Y} \) are produced by regressing the so-obtained \( \mathbf{t} \) onto \( \mathbf{X} \) and \( \mathbf{Y} \). For complex-valued data, these regressions now become...
4.2. A widely linear complex-valued PLS algorithm

widely linear in terms of \( t \) and \( t^* \), which results in the approximations \( X = tp_1^H + t^*p_2^H \) and \( Y = tc_1^H + t^*c_2^H \), calculated in an augmented form

\[
\begin{align*}
p &= (t^+ X)^H, & c &= (t^+ Y)^H, \\
(4.3)
\end{align*}
\]

where \( p = [p_1, p_2] \), \( t = [t, t^*] \) and \( c = [c_1, c_2] \).

The latent variables in \( Y \), denoted by the vectors \( u \) that correspond to the latent variables \( t \) in \( X \), are obtained as

\[
u = Yc_1 + Y^*c_2.
\]

Similarly, the regression of \( u \) to \( Y \) is an approximation \( Y = uq_1^H + u^*q_2^H \) which can be represented in an augmented form as

\[
\bar{q} = (u^+ Y)^H
\]

where \( \bar{q} = [q_1, q_2] \) and \( u = [u, u^*] \). The above steps describe the WL-CPLS decomposition for the latent variable \( t \) in each PLS iteration. Before the next latent variable can be calculated, the impact of the currently extracted component, \( t_n \), must be removed from the data matrices \( X \) and \( Y \). This is achieved by deflating (subtracting) the respective approximations (4.6) and (4.7) from \( X \) and \( Y \) to give

\[
\begin{align*}
X_{i+1} &= X_i - t_i p_{1,i}^H - t_i^* p_{2,i}^H, & Y_{i+1} &= Y_i - t_i c_{1,i}^H - t_i^* c_{2,i}^H, \\
(4.5)
\end{align*}
\]

or in an augmented form

\[
\begin{align*}
X_{i+1} &= X_i - t_i p_i^H, & Y_{i+1} &= Y_i - t_i c_i^H, \\
\end{align*}
\]

where the subscript \( i \) indicates the iteration number with \( X_1 = X \) and \( Y_1 = Y \). In the next iteration, the matrices \( X_{i+1} \) and \( Y_{i+1} \) are used in place of \( X_i \) and \( Y_i \), which ensures that the new extracted score (latent variable) will be orthogonal to the component extracted in the previous iterations. As such, the information expressed by each extracted component is accounted for separately, that is

\[
\begin{align*}
X &= TP_1^H + T^* P_2^H, & Y &= TC_1^H + T^* C_2^H, \\
(4.6) & & (4.7)
\end{align*}
\]

which gives the WL-CPLS decomposition corresponding to (3.5), in the form

\[
Y = UQ_1^H + U^* Q_2^H
\]

(4.8)

The complete WL-CPLS algorithm is outlined in Algorithm 6 and is iterated until \( r \) latent components have been found. The required number of components is determined based on a stopping criterion, described in Section 4.4.2.

---

1 This is a rank-2 approximation unless a strictly linear regression of the component \( t \) to \( X \) and \( Y \) is sufficient.
Algorithm 6. The NIPALS algorithm for widely linear complex PLS (WL-CPLS)

1: Initialise: $X_1 = [X, X^*], Y_1 = [Y, Y^*]$
2: for $i = 1, \ldots, r$ do
3: $X_{i,Re} = X_i \Gamma_m$, and $Y_{i,Re} = Y_i \Gamma_n$
4: $w_{i,Re} = \text{Eig}_{\text{max}} \{X_{i,Re}^T Y_{i,Re} X_{i,Re}\}^T$
5: $t_i = X_i w_i$, $t_i^* = [t_i, t_i^*]^T$
6: $c_i^T = (t_i^* Y_i)^H$
7: $p_i = (t_i^* X_i)^H$
8: $u_i = Y_i c_{i,1} + Y_i^* c_{2,i}$, $u_i^* = [u_i, u_i^*]^T$
9: $q_i = (u_i^* Y_i)^H$
10: $X_{i+1} = X_i - t_i p_i^H$, $Y_{i+1} = Y_i - t_i c_i^H$
11: Store $t_i$, $u_i$, $p_{1,i}$, $p_{2,i}$, $q_{1,i}$, $q_{2,i}$, $c_{1,i}$, $c_{2,i}$ and $w_i$
12: end for

Figure 4.1. Geometric view of WL-CPLS regression.

Remark 13. For a regularised regression application it is desirable to identify a latent subspace within the regressors, $X$, that is used to predict the dependent variables, $Y$. The matrix $W$ within the proposed WL-CPLS in Algorithm 6 provides such a basis, which is then used to sequentially calculate the score vectors in the matrix $T$. Owing to the optimisation process in (4.1), this subspace is chosen so as to contain the full joint complex second-order information. The widely linear regression of the scores, $T$, to the data matrices $X$ and $Y$ described in (4.6) and (4.7) then give the WL-CPLS approximations of the identified subspace and account for the required degrees of freedom needed in complex-valued estimation. As such, the WL-CPLS algorithm is a generic extension of the real-valued NIPALS algorithm described in Section 3.3.1 for PLS-regression.

Remark 14. The complex-valued WL-CPLS takes into account the widely linear relationship between the $X$ and $Y$ components.

Figure 4.1 shows a geometric interpretation of WL-CPLS. In contrast to real-valued
4.2. A widely linear complex-valued PLS algorithm

PLS, the scores $T$ in WL-CPLS define a regression subspace in both $X$ and $X^*$. This subspace is then used to provide a widely linear regression to predict dependent variables $Y$.

Remark 15. An important aspect of the WL-CPLS algorithm is that the scores $T$ are fully uncorrelated in terms of augmented complex-valued second-order statistics, that is, the score covariance and pseudocovariance matrices, $T^HT$ and $TT^T$, are diagonal. For normalised score vectors, $t$, their covariance matrix becomes $T^HT = I$. This, in turn, means that the diagonal elements of the score pseudocovariance matrix, $TT^T$, readily give their circularity quotient (as defined in Section 2.3.1), thus further highlighting the physical insight of WL-CPLS. This would not be available using the original NIPALS Algorithm 3 on complex data cast into the real domain.

4.2.1 Alternative WL-CPLS algorithms

The WL-CPLS described in Algorithm 6 can serve as a platform for a class of WL-CPLS results. The choice of the vector, $w$, from (4.2) aims to reflect the maximum augmented cross-second-order statistics (cross-covariance and cross-pseudocovariance) between the input $X$ and the output $Y$. However, another choice can be made through a different criterion. For example, we can choose $w$ to reflect only the maximum cross-covariance between the input $X$ and the output $Y$. This is achieved by the following optimisation

$$w = \arg \max_{||w||=1} ||w^HX^HY||^2_2.$$

The solution, $w$, is the largest eigenvector of the matrix, $X^HYY^HX$. This alternative WL-CPLS is summarised in Algorithm 7. It is important to note, however, that this alternative vector, $w$, is not capable of detecting a latent variable, $t$, that represents only cross-pseudocovariance between $X$ and $Y$.

Algorithm 7. The WL-CPLS NIPALS algorithm with an alternative cross-covariance criterion

1: Initialise: $X_1 = [X, X^*]$, $Y_1 = [Y, Y^*]$
2: for $i = 1, \ldots, r$ do
3: \hspace{1em} $w_i = \text{Eigmax} \{X_i^HY_iY_i^HX_i\}$
4: \hspace{1em} $t_i = X_iw_i$, $t_i^* = [t_i, t_i^*]$ \hspace{1em} $c_i = (t_i^* Y_i)^H$
5: \hspace{1em} $\overline{p}_i = (t_i^* X_i)^H$
6: \hspace{1em} $u_i = Y_i c_{1,i} + Y_i^* c_{2,i}$, $u_i = [u_i, u_i^*]$ \hspace{1em} $q_i = (u_i^* Y_i)^H$
7: \hspace{1em} $\overline{X}_{i+1} = \overline{X}_i - t_i \overline{p}_i^H$, $\overline{Y}_{i+1} = \overline{Y}_i - t_i c_i^H$
8: \hspace{1em} Store $t_i, u_i, p_{1,i}, p_{2,i}, q_{1,i}, q_{2,i}, c_{1,i}, c_{2,i}$ and $w_i$\hspace{1em} end for
Table 4.1. Comparison of the real-valued NIPALS and WL-CPLS algorithms

<table>
<thead>
<tr>
<th>Algorithm Step</th>
<th>real-valued PLS</th>
<th>WL-CPLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-covariance structure</td>
<td>$S_i = X_i^T Y_i$</td>
<td>$S_{i,Re} = X_{i,Re}^T Y_{i,Re}$</td>
</tr>
<tr>
<td></td>
<td>The cross-covariance matrix</td>
<td>Transformed to the real-domain</td>
</tr>
<tr>
<td>Eigen-decomposition</td>
<td>$w_i = \text{Eig}_{\text{max}} {S_i S_i^T}$</td>
<td>$w_{i,Re} = \text{Eig}<em>{\text{max}} {S</em>{i,Re} S_{i,Re}^T}$</td>
</tr>
<tr>
<td></td>
<td>Left-singular vector of $S_i$</td>
<td>Left singular vectors of $S_{i,Re}$</td>
</tr>
<tr>
<td>Score calculation</td>
<td>$t_i = X_i w_i$</td>
<td>$t_i = X_i w_i$</td>
</tr>
<tr>
<td></td>
<td>The projection of $X_i$</td>
<td>The projection of $X_i$</td>
</tr>
<tr>
<td>LV decomposition/Deflation</td>
<td>$X_{i+1} = X_i - t_i p_i^T$</td>
<td>$X_{i+1} = X_i - t_i p_i^H$</td>
</tr>
<tr>
<td></td>
<td>$Y_{i+1} = Y_i - t_i c_i^T$</td>
<td>$Y_{i+1} = Y_i - t_i c_i^H$</td>
</tr>
</tbody>
</table>

4.3 Analysis of the WL-CPLS result

The WL-CPLS result is shown to cater for full complex-valued second-order statistics and yields a widely-linear relationship between the obtained latent variables and the original data blocks. As such, this has extended the PLS algorithm for use with a multidimensional algebra, and it is possible to obtain bivariate latent variables. The result is now analysed to examine its properties.

The focus of our analysis of the WL-CPLS algorithm is on the calculation of the widely linear regression coefficients, in (2.16), based on the structure of the WL-CPLS decomposition of $X$ and $Y$. This also highlights the ability of the WL-CPLS algorithms to produce a regularised widely linear regression solution. Next, the application of the WL-CPLS algorithm as a complex covariance matrix diagonalisation transform is demonstrated, including a special case where the result is equivalent to an existing technique, the strong uncorrelating transform (SUT) [89]. The orthogonality between the latent variables, $t$, and the model residuals is then examined. Finally, the convergence of the WL-CPLS algorithm is proven for a univariate output, $y$, through a recurrence relation for the latent variables, $t$.

4.3.1 The WL-CPLS with respect to the PLS core-components

In this section, the proposed WL-CPLS solution is compared to the core-components of a PLS algorithm described in Section 3.3.4 as: (i) the cross-covariance structure, (ii) its eigendecomposition, (iii) the score vectors calculation and (iv) the latent variable decomposition deflation on each iteration. Table 4.1 compares the choice of each component for the proposed WL-CPLS to the original NIPALS Algorithm 3.
4.3.2 The augmented matrix generalised inverse: WL-CPLS for a regularised widely linear estimation

It has already been shown that the PLS method calculates a joint latent variable decomposition of both the input $X$ and output $Y$. In real-valued PLS, this property is used to calculate a regularised regression solution in scenarios where the input matrix $X$ is sub-rank through the generalised inverse of the independent variables in (3.9). Next it is shown that the WL-CPLS algorithm allows a straightforward calculation of the coefficient matrices $H$ and $G$ in the widely linear regression model

$$\hat{Y} = \tilde{X}H + \tilde{X}^*G,$$

where $\tilde{X}$ is the WL-CPLS approximation of the input $X$ given

$$\tilde{X} = T(P_1^H W)^H + T^*(P_2^H W)^H. \quad (4.9)$$

This is a rigorous generalisation of the real-valued NIPALS algorithm, where the matrix $W$ is unitary and the matrices $P_1^H W$ and $P_2^H W$ are upper triangular [69]. Upon substituting (4.9) into (2.19), the WL-CPLS solutions for $H$ and $G$ are calculated as

$$H_{WL-CPLS} = [\tilde{C} - \tilde{P}^* \tilde{C}^* + \tilde{P}]^+ [\tilde{R}^H - \tilde{P}^* \tilde{C}^* + \tilde{S}]^T,$$

$$G_{WL-CPLS} = [\tilde{C}^* - \tilde{P}^* \tilde{C} + \tilde{P}^*]^+ [\tilde{S}^T - \tilde{P}^* \tilde{C} + \tilde{R}^H]^T, \quad (4.10)$$

where $\tilde{R} = \tilde{Y}^H \tilde{X}$, $\tilde{S} = \tilde{Y}^T \tilde{X}$, $\tilde{C} = \tilde{X}^H \tilde{X}$, $\tilde{P} = \tilde{X}^T \tilde{X}$ and $\tilde{Y}$ is given by the decomposition in (4.7). Therefore, the computation of $H$ and $G$ requires calculation of the generalised inverses\(^2\) of $\tilde{C}$ and $\tilde{C} - \tilde{P}^* \tilde{C}^* + \tilde{P}$, which requires special attention.

**Lemma 1.** (Miller [90]) Consider a matrix $M = N + O$, where $N \in \mathbb{C}^{n \times n}$ is invertible and $O \in \mathbb{C}^{n \times n}$ can be split into rank-1 matrices, $O = E_1 + E_2 + \cdots + E_r$, with $r$ denoting the rank of $O$. Then $M$ can be inverted iteratively as

$$M_{k+1}^{-1} = M_k^{-1} + \frac{M_k^{-1} E_k M_k^{-1}}{1 + \text{Tr}(M_k^{-1} E_k)}, \quad (4.11)$$

where $M_1 = N$. The iteration is terminated at $k = r$.

**Proposition 3.** The inverses of the matrices $\tilde{C}$ and $\tilde{C} - \tilde{P}^* \tilde{C}^* + \tilde{P}$ can be calculated through the exploitation of their special structure and with the aid of Lemma 1.

**Proof.** The proof consists of two parts, the calculation of $\tilde{C}^+$ and the calculation of $(\tilde{C} - \tilde{P}^* \tilde{C}^* + \tilde{P})^+$.

In order to calculate the generalised inverse of $\tilde{C}$, the WL-CPLS approximation (4.9) can be used to give

$$\tilde{C} = W(F_1 + F_2 + F_3 + F_4) W^H, \quad (4.12)$$

\(^2\)Note that if the matrices in (2.19) are generally invertible then $\tilde{C}^*$ and $\tilde{C}^* - \tilde{P} \tilde{C}^* + \tilde{P}^*$ are also generally invertible.
with $F_1, F_2, F_3$ and $F_4$ defined as the $LDU$ decompositions

\[
\begin{align*}
F_1 &= (P_1^H W)^H (P_1^H W) = L_1 L_1^H \\
F_2 &= (P_2^H W)^H D^* (P_2^H W) = L_2 D^* L_2^H \\
F_3 &= (P_3^H W)^H D (P_1^H W) = L_2 D L_1^H \\
F_4 &= (P_3^H W)^H (P_3^H W) = L_2 L_2^H,
\end{align*}
\]  

(4.13)

where $D = T^T T$ is diagonal, and the implicit assumption of normalised scores, $T^H T = I$, was employed. Since $W$ is unitary, a generalised inverse for $C$ is then given by

\[
\tilde{C}^+ = W(F_1 + F_2 + F_3 + F_4)^{-1} W^H, 
\]

(4.14)

which boils down to the problem of inverting the matrix $(F_1 + F_2 + F_3 + F_4))$. Although this matrix is non-singular, it is possible to calculate its inverse using only its structure. This is equivalent to the real-domain NIPALS algorithm, where the necessary generalised inverses for a regularised regression are calculated solely due to the decompositions’ structure [69].

From the $LDU$ decompositions of $F_n$, $n = 1, 2, 3, 4$ given in (4.13), we further factorise $(F_1 + F_2 + F_3 + F_4)$ as

\[
F_1 + F_2 + F_3 + F_4 = L_1 (L_1^H + D^* L_2^H) + L_2 (L_2^H + D L_1^H) \\
= L_1 U_A + L_2 U_B \\
= A + B.
\]

Upon substituting $N = A$ and $O = B$ into Lemma 1, this form permits the calculation of the inverse whereby the $LU$ structure of $A$ guarantees that the generalised inverse of $\tilde{C}$ is readily obtained.

Next, consider the inversion of $\tilde{C} - \tilde{P}^* \tilde{C}^+ \tilde{P}$. Similarly to $\tilde{C}$, the matrix $\tilde{P}$ can be factorised using the WL-CPLS approximation in (4.9), to yield

\[
\tilde{P} = W^* (K_1 + K_2 + K_3 + K_4) W^H, 
\]

(4.15)

where

\[
\begin{align*}
K_1 &= (P_1^H W)^T D (P_1^H W) = L_1^* D L_1^H \\
K_2 &= (P_2^H W)^T (P_2^H W) = L_2^H L_2^H \\
K_3 &= (P_3^H W)^T (P_1^H W) = L_2^H L_1^H \\
K_4 &= (P_3^H W)^T D^* (P_2^H W) = L_2^H D^* L_2^H.
\end{align*}
\]  

(4.16)

Denote $K_1 + K_2 + K_3 + K_4 = K$ and $F_1 + F_2 + F_3 + F_4 = F$, we can now obtain $C = WFW^H$ and $P = W^* KW^H$. The generalised inverse of $\tilde{C} - \tilde{P}^* \tilde{C}^+ \tilde{P}$ can therefore be obtained as

\[
(\tilde{C} - \tilde{P}^* \tilde{C}^+ \tilde{P})^+ = W(F - K^* F^{-1} K)^{-1} W^H.
\]

To calculate the inverse of $(F - K^* F^{-1} K)^{-1}$ we can employ again Lemma 1 with $N = F$ and $O = -K^* F^{-1} K$. The inverse of $F$ is retained from the calculation of the generalised inverse of $\tilde{C}$ which satisfies the requirements of Lemma 1. The so obtained generalised inverses of $\tilde{C}$ and $\tilde{C} - \tilde{P}^* \tilde{C}^+ \tilde{P}$ conclude the proof of Proposition 3. \hfill \Box
The steps in the calculation of the WL-CPLS regression coefficient matrices, $H_{WL-CPLS}$ and $G_{WL-CPLS}$, are summarised in Algorithm 8.

**Algorithm 8. Calculation of $H_{WL-CPLS}$ and $G_{WL-CPLS}$**

1. Use the WL-CPLS in Algorithm 6 to obtain: $T$, $P_1$, $P_2$ and $W$
2. Calculate WL-CPLS approximation $\tilde{X} = T(P_1^H W)W^H + T^*(P_2^H W)W^H$
3. Obtain $C = \tilde{X}^H \tilde{X}$, $P = \tilde{X}^T \tilde{X}$, $R = \tilde{Y}^H \tilde{X}$ and $S = \tilde{Y}^T \tilde{X}$
4. Obtain $L_1 = (P_1^H W)^H$, $L_2 = (P_2^H W)^H$ and $D = T^T T$
5. Obtain $U_A = (L_1^H + D^* L_1^H)$ and $U_B = (L_2^H + DL_2^H)$
6. Define $F = L_1 U_A + L_2 U_B$
7. Calculate $F^{-1}$ using Lemma 1 with $N = L_1 U_A$ and $O = L_2 U_B$
8. Calculate $\tilde{C}^+ = WF^{-1} W^H$
9. Define $K = L_1 DL_1^H + L_1^H L_1 + L_2^H L_2 + L_2 D^* L_2^H$
10. Obtain $(F - K^* F^{-1} K)^{-1}$ using Lemma 1 with $N = F$ and $O = -K^* F^{-1} K$
11. Calculate $[\tilde{C} - P^* C^* + P]^+ = W^H (F - K^* F^{-1} K)^{-1} W$
12. Calculate $H_{WL-CPLS} = [\tilde{C} - P^* C^* + P]^+ [R^H - P^* C^* S^T]$
13. Calculate $G_{WL-CPLS} = ([\tilde{C} - P^* C^* + P]^+)^* [S^T - \tilde{P} C^* R^H]$

**Remark 16.** The widely linear complex PLS algorithm in Algorithm 6 provides an approximation of the matrix $X$, denoted by $\tilde{X}$, which admits a regularised widely linear regression $Y = \tilde{X} H_{WL-CPLS} + \tilde{X}^* G_{WL-CPLS}$. The regression coefficients are obtained from Algorithm 8, owing to the inherent structure of the WL-CPLS decomposition.

### 4.3.3 WL-CPLS as a covariance matrix diagonalisation transform

Diagonalisation of covariance matrices is an essential data analysis tool, and for real-valued data this is accomplished through PCA/SVD. It provides an unsupervised learning implementation of latent variables, where they are obtained from a single block. In the case of PCA this is based around the determination of orthogonal components. For similar applications in the complex domain [91], both the covariance and pseudocovariance matrices must be simultaneously diagonalised. This can be achieved through the strong uncorrelating transform (SUT) [89], given by

$$X_{SUT} = X \Theta,$$

where $\Theta \in \mathbb{C}^{m \times m}$ is a transform, obtained through Algorithm 9, which maps the measured variables, $X \in \mathbb{C}^{N \times m}$, to the uncorrelated SUT variables, $X_{SUT} \in \mathbb{C}^{N \times m}$. The transformed covariance matrix $C_{X_{SUT}} = X_{SUT}^H X_{SUT}$ is an identity matrix, and the pseudocovariance matrix $P_{X_{SUT}} = X_{SUT}^T X_{SUT}$ is diagonal and with real-valued entries which represent the circularity quotient (defined in Section 2.3.1) of each component (column vector) in $X_{SUT}$.

The WL-CPLS result can be viewed, similarly to the SUT, as an uncorrelating transform by considering the augmented form of the decomposition of $X$, given by

$$T = X \left( \begin{array}{c} (P_1^H W)W^H \\ ((P_2^H W)W^H) \\ (P_2^H W)W^H \\ ((P_1^H W)W^H) \\ \end{array} \right) ^+,$$

(4.18)
Algorithm 9. The SUT algorithm

1: **Initialise:** The data matrix $X$ is provided
2: Calculate empirical covariance matrix $C = XX^H$
3: Apply the SVD to give $C = U\Delta U^H$
4: Obtain new variables $\hat{X} = XU\Delta^{-1/2}$
5: Calculate empirical pseudocovariance matrix $P = \hat{X}^T\hat{X}$
6: Apply the Takagi factorisation to give $P = V\Lambda V^T$
7: Obtain SUT variables $\tilde{X}_{SUT} = XU\Delta^{-1/2}V^* = X\Theta$

The matrix $T^HT$ is block diagonal, and hence, the transformed covariance matrix, $T^HT$, and the pseudocovariance matrix, $T^TT$, are both diagonal. Therefore, the matrix

$$
\begin{pmatrix}
(P_1^HT)^{\dagger} (P_1^HT)^{WH} (P_1^HT)^{WH}^* \\
(P_2^HT)^{\dagger} (P_2^HT)^{WH} (P_2^HT)^{WH}^*
\end{pmatrix},
$$

obtained by the WL-CPLS solution from regressing $X$ to itself can be considered as an uncorrelating transform in the same way as $\Theta$.

To validate the performance of the so-obtained WL-CPLS uncorrelating transform for dimensionality reduction, consider a case where a data matrix

$$
X = Z + N,
$$

contains a low-rank “signal” subspace component, denoted by the matrix $Z$, and a full-rank “noise” subspace component, denoted by the matrix $N$. This is a common real-world scenario and if the signal accounted for the majority of the total variance in $X$ then a practical uncorrelating transform would clearly identify the low-rank subspace. To this end, we generated $N = 1000$ samples of 20 independent, identically distributed (i.i.d.), non-circular, Gaussian sources which were mixed to give a matrix $Z \in \mathbb{C}^{1000 \times 50}$. A noise source, $N \in \mathbb{C}^{1000 \times 50}$, was added to the matrix, $Z$, to give the data matrix $X = Z + N$, in (4.19), drawn from a circular, Gaussian i.i.d. distribution with an SNR=26dB, where the SNR is defined as

$$
SNR = 10 \log_{10} \frac{\text{Tr}\{E[Z^HZ]\}}{\text{Tr}\{E[N^HN]\}},
$$

(4.20)

The SUT and WL-CPLS transforms were then performed. Figure 4.2 shows the percentage of the total variance in $X$ that is explained by the approximation obtained from each individual score described by the metric

$$
VE = 100 \frac{\text{Tr}\{\tilde{A}^H\tilde{A}\}}{\text{Tr}\{A^HA\}}.
$$

(4.21)

This metric is evaluated for the WL-CPLS result where $A = \hat{X} = t_ip_i^H$, where $i$ is the iteration number for the respective WL-CPLS score. For the SUT result the metric is calculated for $A = x_{SUT,i}\Theta^*_i$, where $x_{SUT,i}$ is the $i$-th component (column of $X_{SUT}$) and $\Theta^*_i$ is the $ith$ row of the pseudoinverse of the SUT transform $\Theta$. The metric $VE$ was calculated for both the WL-CPLS/SUT results obtained from both the data matrix $A = X$ and for the noiseless case $A = Z$ from above in (4.19).
4.3. Analysis of the WL-CPLS result

Figure 4.2. Proportion of the variance (in %) explained by each component in the SUT and WL-CPLS transform, $VE$ in (4.21).

Figure 4.2 indicates that the uncorrelating transform produced by the WL-CPLS is more powerful for subspace identification than the SUT. Observe that the WL-CPLS concentrates the energy in the signal into as few components as possible, owing to the fact it selects a component in each iteration which explains the maximum variance in $X$ (due to the cross-covariance optimisation problem in (4.1)). On the other hand for the SUT, owing to the inherent whitening, each element accounts for a similar amount of variance. The SUT does find the low-rank subspace with when the input is just the low-rank signal matrix $Z$. For even a small amount of noise, however, the signal power is spread amongst all components, which makes it difficult for the SUT to identify a subspace. On the other hand, the WL-CPLS performs similarly in both cases. This application shows that the WL-CPLS result produces complex-valued latent variables which are representative of a physically meaningful process. Note that the computational requirements of the SUT are only two SVDs whereas the WL-CPLS transform requires one SVD per latent component extracted.

A special case of duality with the SUT

A special case where the application of the WL-CPLS algorithm as a transform and the SUT yield identical results exists for a unitary input, $X$, that has an empirical covariance matrix $C = X^HX = I$. For such data, both methods result in the transform matrix $V^*$ obtained from the Takagi factorisation of the matrix $P = X^TX = VAV^T$. In other words, for “white” data, the SUT transformed variables, $X_{SUT}$ in (4.17), are equal to the scores matrix, $T$, produced by the WL-CPLS algorithm.

To prove this equivalence, observe first that the whitening transform in the SUT calculation (outlined in Steps 3 and 4 of Algorithm 9) is no longer required, and so the Takagi factorisation on step 6 can be calculated directly from the empirical pseudocovariance matrix $P$. On the other hand, for the WL-CPLS result, the input and output deflations (Step 11 of Algorithm 6) are symmetric and are achieved by a strictly linear regression. This means that $p_1 = c_1 = w$ and $p_2 = c_2 = 0$. The WL-CPLS solution can now be calculated in a closed form through the SVD of $S_{Re}S_{Re}^T$ (where $S_{Re} = X_{Re}^TY_{Re}$) as
Chapter 4. Widely Linear Complex Partial Least Squares

the first \( m \) (where \( m \) is the number of columns of \( X \)) singular vectors are cast into the complex domain according to (4.2). To demonstrate the equivalence to the Takagi factorisation of the empirical pseudocovariance matrix, \( P \), consider the link between the SVDs of \( X^\text{Re}X_{\text{Re}} = U_{\text{Re}}\Sigma_{\text{Re}}U_{\text{Re}}^\text{T} \) and \( X^H X = U_C\Sigma_C U_C^H \). It is well known that \( \Sigma_{\text{Re}} = 2\Sigma_C \) [92] and hence, the transformed singular vectors \( \Gamma_m U_{\text{Re}} \) produce the same diagonalisation of the matrix \( X^H X \) as the SVD, up to a factor of 2.

**Lemma 2.** (Horn [93]) Let \( A \in \mathbb{C}^{n \times n} \) be a square matrix. There exists a unitary matrix, \( X \), a diagonal matrix, \( \Lambda \), with non-negative entries, and a matrix \( Y \) with orthonormal rows such that \( A = X\Lambda Y \). The columns of matrix \( X \) are the eigenvectors of \( AA^H \) and the diagonal entries of the matrix \( \Lambda \) are the square root of the corresponding eigenvalues of \( AA^H \). If \( AA^H \) has distinct eigenvalues, then \( X \) is determined up to a right diagonal factor \( D = \text{diag}(e^{i\theta_1}, \ldots, e^{i\theta_n}) \) with all \( \theta \in \mathbb{R} \) and \( |d_{ij}| = |e^{i\theta}| = 1 \); that is, if \( A = X_1\Lambda Y_1 = X_2\Lambda Y_2 \) then \( X_2 = X_1D \).

Lemma 2 (which is a property of the SVD) states that if a square matrix \( A \) can be represented as \( A = X_1\Lambda Y_1 = X_2\Lambda Y_2 \) (where \( X_1, X_2, Y_1 \) and \( Y_2 \) are unitary matrices) then \( X_2 = X_1D \) where \( D = \text{diag}(e^{i\theta_1}, \ldots, e^{i\theta_n}) \), \( \theta \in \mathbb{R} \) and \( |d_{ij}| = |e^{i\theta}| = 1 \). Using Lemma 2 we can now derive the relationship

\[
\Gamma_m U_{\text{Re}} = \sqrt{2}U_C D. \tag{4.22}
\]

Therefore, the latent vectors obtained by the WL-CPLS are identical to those obtained from the SVD of \( X^\text{H}X \), each rotated to be of the form

\[
\sqrt{2}U_C D = \begin{pmatrix}
U_{S_1} & U_{S_2} \\
U_{S_1}^* & U_{S_2}^*
\end{pmatrix}, \tag{4.23}
\]

where the matrix \( [U_{S_1}^T, U_{S_2}^H]^T \) represents the matrix of the first \( m \) (of \( 2m \)) eigenvectors of \( S_{\text{Re}}S_{\text{Re}}^T \) transformed to the complex domain (as in (4.2)) and the matrix \( [U_{S_2}^T, U_{S_2}^H]^T \) represents the remaining \( m \) eigenvectors of \( S_{\text{Re}}S_{\text{Re}}^T \). The matrix \( U_{S_1} \) is then identical to the matrix \( W \) for the WL-CPLS result. In order for the matrix \( \sqrt{2}U_C D \) to diagonalise \( X^H X \) in this special case, the sum \( U_{S_1}^H P^* U_{S_1} + U_{S_1}^T P U_{S_1} + U_{S_2}^H U_{S_1} + U_{S_1}^T U_{S_1} \) must be diagonal. To show this, consider the transform of the original variables \( \tilde{X} \) (in augmented form) to the new variables \( \tilde{X} \) by means of the matrix \( \Gamma_m U_{\text{Re}} \) which, through (4.23), gives

\[
\tilde{X} = X \begin{pmatrix}
U_{S_1} & U_{S_2} \\
U_{S_1}^* & U_{S_2}^*
\end{pmatrix}.
\]

This is precisely the result produced by the WL-CPLS transform. The covariance matrix
of the transformed variables is then given by

\[ \hat{X}^H \hat{X} = \left( \begin{array}{ll} U_{S1}^H & U_{S1}^T \\ U_{S2}^H & U_{S2}^T \end{array} \right) \left( \begin{array}{cc} I & P^* \\ P & I \end{array} \right) \left( \begin{array}{cc} U_{S1} & U_{S2} \\ U_{S1}^* & U_{S2}^* \end{array} \right), \]

(4.24)

\[ = \left( \begin{array}{cc} U_{S1}^H U_{S1} + U_{S1}^T P U_{S1} + U_{S1}^H P^* U_{S1}^* + U_{S1}^T U_{S1}^* \\ U_{S2}^H U_{S1} + U_{S2}^T P U_{S1} + U_{S2}^H P^* U_{S1}^* + U_{S2}^T U_{S1}^* \\ U_{S1}^H U_{S2} + U_{S1}^T P U_{S2} + U_{S1}^H P^* U_{S2}^* + U_{S1}^T U_{S2}^* \\ U_{S2}^H U_{S2} + U_{S2}^T P U_{S2} + U_{S2}^H P^* U_{S2}^* + U_{S2}^T U_{S2}^* \end{array} \right). \]

This matrix is known to be diagonal, which gives rise to the condition that \( U_{S1}^H P^* U_{S1} + U_{S1}^T P U_{S1} + U_{S1}^H U_{S1} + U_{S1}^T U_{S1}^* \) must be diagonal. Through inspection, this is satisfied when \( U_{S1} = V^* \). Therefore, the WL-CPLS result for \( X \) serving as both the input and output is equivalent to that of the SUT for data for which the empirical covariance matrix is given by \( C = I \).

### 4.3.4 The convergence of the WL-CPLS algorithm

In Section 3.3.3 it was shown that for a real-valued NIPALS PLS solution the residuals to the PLS decomposition models (3.3) and (3.4) are orthogonal to the space used for the regression. Moreover, a convergence proof was introduced for when the output is a univariate vector, \( y \). In such a case, the space spanned by the scores on the \( i \)-th iteration, \( T_i, \) is given by the space spanned by the vectors \( K_i = [s, SK_{i-1}] \) where \( s = XX^T y, S = XX^T \) and \( K_1 = [s] \). Here it is shown that these aspects can be equivalently proven for the WL-CPLS decompositions (4.6) and (4.7).

The orthogonality of the model residuals

Consider the WL-CPLS approximation with \( i \) components of \( X \) in (4.6). We can write the residual of \( X \) as

\[ X_{i+1} = X - T_i \bar{P}_i^H, \]

(4.25)

where \( \bar{P}_i = [P_{1,i}, P_{2,i}] \). Note that \( X_{i+1} \) is equivalent to the deflation step in (4.5) at the \( i \)-th iteration. The matrix \( \bar{P}_i \) is obtained (from (4.6)) as

\[ \bar{P}_i = X^H T_i (T_i^H T_i)^{-1}, \]

and the residuals from (4.25) can be expressed as

\[ X_{i+1} = (I - P_{i,i}) X, \]

(4.26)

where \( P_{i,i} = T_i (T_i^H T_i)^{-1} T_i^H \) is a projection onto the space spanned by \( T_i \). A similar relationship can be derived for the output \( Y \) using the WL-CPLS model (4.7), where the model residuals are given as \( Y_{i+1} = Y - T_i \bar{C}_i^H \) with \( \bar{C}_i = Y^H T_i (T_i^H T_i)^{-1} \). As above, this leads to a relationship of the form

\[ Y_{i+1} = (I - P_{i,i}) Y. \]

(4.27)
From (4.27) and (4.26), the residuals for both multivariate $X_i$ and $Y_i$ must be orthogonal to the subspace defined by the augmented latent variables $T_i$ as $P_{t,i}$ is a projection matrix, verified by the computation of $P_{t,i}^H P_{t,i} = P_{t,i}$.

### Proof of WL-CPLS convergence to a Krylov space

Consider the univariate output $y$. In this case we can derive a recurrence relation for the augmented scores $t = [t_i, t_i^*]$ obtained in the alternative form of WL-CPLS described in Algorithm 7, where the vector $w_i$ in each iteration is obtained as the largest eigenvector of the matrix $X_i^H y, y_i^H X_i$. This eigenvector is given by [73]

$$w_i = \frac{X_i^H y}{||X_i^H y||},$$

and is proportional to the vector $X_i^H y$. The score vector is then given by $t_i = X_i w_i$ which is proportional to $X_i X_i^H y$. Utilising the relation obtained in (4.26) we can now write

$$t_{i+1} \propto (I - P_{t,i}) XX^H (I - P_{t,i}) y.$$  

This can be written in the form of a recurrence relationship of the augmented scores vector

$$t_{i+1} \propto [(I - P_{t,i}) XX^H (I - P_{t,i}) y, (I - P_{t,i})^* X^* X^T (I - P_{t,i})^* y^*].$$  \hspace{1cm} (4.28)

**Proposition 4.** The space defined by the vectors $K_i = [s, SK_{i-1}, SK_{i-1}^*]$ and its conjugate $K_i^*$, where $s = X X^H y$, $S = XX^H$ and $K_1 = [s]$ forms a basis for the augmented scores matrix, $T_i = [T_i, T_i^*]$, obtained from the WL-CPLS in Algorithm 7.

*Proof.* The proof follows the same structure as for Proposition 1, except we must show that the columns of $T_i$ are formed from a linear combination of the columns of $[K_i, K_i^*]$. The first score is given as $t_1 \propto [s, s^*] = [K_1, K_1^*]$ and so the base case $i = 1$ is correct. We induce that the columns of the matrix $T_i$ are a linear combination of the column vectors of $K_i$ and $K_i^*$ concatenated, which means that the matrix $P_{t,i}$ is also a linear combination of the vectors in $K_i$ and $K_i^*$. The recursion in (4.28) represents the composite of the vector

$$t_{i+1} \propto s - SP_{t,i} y - P_{t,i} s + P_{t,i} SP_{t,i} y,$$

and the vector

$$t_{i+1}^* \propto s^* - S^* P_{t,i}^* y^* - P_{t,i}^* s^* + P_{t,i}^* S^* P_{t,i}^* y^*.$$  

As the matrix $P_{t,i}$ is a linear combination of the columns of $[K_i, K_i^*]$, the vectors $t_{i+1}$ and $t_{i+1}^*$ are then also a linear combination of the columns of $[K_{i+1}, K_{i+1}^*]$ which is the matrix $[s, s^*, SK_i, S^* K_i, SK_i^*, S^* K_i^*]$. As a result, if $T_i$ is a linear combination of the columns of $[K_i, K_i^*]$ then the columns of $T_i$ are a linear combination of the columns of $[K_{i+1}, K_{i+1}^*]$ which proves Proposition 4. \hfill \square
4.4 Simulation results

The performance of the proposed WL-CPLS algorithm is verified in a variety of scenarios. The performance metric used was the prediction mean square error (MSE), defined as

\[ MSE = E[\|Y - \hat{Y}\|^2], \]

where \( Y \) are the original “correct” output variables and \( \hat{Y} \) are their predictions from the WL-CPLS model, \( \hat{Y} = \hat{X}H + X^*G \), calculated from the approximation \( \hat{X} = TH_1H_2W + T^*(P_2W)W^H \). For rigour, we examined the key factors that affect the performance of the WL-CPLS algorithm: the number of WL-CPLS components selected and the noise level in \( X \) and \( Y \).

4.4.1 Prediction MSE for a varying SNR in \( Y \)

<table>
<thead>
<tr>
<th>Table 1. Generation of synthetic test data matrices ( X ) and ( Y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Initialise inputs: ( N, r, m, p, \sigma_X ) and ( \sigma_Y )</td>
</tr>
<tr>
<td>2: Define ( s_n \in \mathbb{C}^{N \times 1} ) as vector of ( N ) samples generated such that ( \text{Re}{s_n} = N(0, 1) ) and ( \text{Im}{s_n} = N(0, 1) )</td>
</tr>
<tr>
<td>3: Define ( 0_n ) as a vector of ( N ) zeros</td>
</tr>
<tr>
<td>4: Generate matrix ( M \in \mathbb{R}^{m \times m} ) such that ( M^TM = I )</td>
</tr>
<tr>
<td>5: Create matrix ( X = [s_1, s_2, \ldots, s_r, 0_1, 0_2, \ldots, 0_{m-r}]M )</td>
</tr>
<tr>
<td>6: Generate matrices ( H \in \mathbb{C}^{m \times p} ) and ( G \in \mathbb{C}^{m \times p} ) such that ( H^HH = I ) and ( G^HG = I )</td>
</tr>
<tr>
<td>7: Define ( n_n \in \mathbb{C}^{N \times 1} ) as vector of ( N ) samples generated as ( \text{Re}{n_n} = N(0, \sigma^2_Y) ) and ( \text{Im}{n_n} = N(0, \sigma^2_Y) )</td>
</tr>
<tr>
<td>8: Create matrix ( N_Y = [n_1, \ldots, n_p] )</td>
</tr>
<tr>
<td>9: Create matrix ( Y = XH + X^*G + N_Y )</td>
</tr>
<tr>
<td>10: Define ( n_n \in \mathbb{C}^{N \times 1} ) as vector of ( N ) samples generated as ( \text{Re}{n_n} = N(0, \sigma^2_X) ) and ( \text{Im}{n_n} = N(0, \sigma^2_X) )</td>
</tr>
<tr>
<td>11: Create matrix ( N_X = [n_1, \ldots, n_m] )</td>
</tr>
<tr>
<td>12: Create matrix ( X = X + N_X )</td>
</tr>
</tbody>
</table>

The performance of the WL-CPLS estimator of the output, \( Y \), was assessed for a varying SNR in \( Y \). This is a classical linear regression scenario which assumes that the input variables, \( X \), are accurate, whereas the output variables, \( Y \), are corrupted by noise. The data were generated as described in Table 1 with \( N = 1000, r = 20, m = p = 100, \sigma_X = 0 \) while \( \sigma_Y \) was varied to give a range of SNRs defined as in (4.20) where \( Z = Y \) and \( N = N_Y \).

The WL-CPLS estimator was obtained for 20 components from training data and the average prediction MSE was then calculated for an ensemble of 100 realisations. The results are shown in Figure 4.3 and the prediction MSE was calculated as a percentage of the total variance in \( Y \). Observe that even for negative SNRs the WL-CPLS produced a regularised regression estimator with less than a 10% error.
Figure 4.3. Prediction MSE for $\hat{Y}$ (as a percentage of total variance) against the SNR in $Y$.

Figure 4.4. Performance, in terms of MSE, of the prediction of $Y$ (as a percentage of total variance) for a varying number of WL-CPLS components, in both training and test data.

### 4.4.2 Prediction MSE for a varying number of WL-CPLS components

The WL-CPLS solution was next examined over a varying number of components, in order to reflect the rank of the approximation for $\tilde{X}$. In this set of simulations, the average prediction MSE was calculated over an ensemble of 100 trials, for a varying SNR and a number of components in WL-CPLS. The data was generated as described in Table 1, with $N = 1000$, $r = 40$, $m = p = 100$ and no noise added ($\sigma_X = \sigma_Y = 0$). The MSE (as a percentage of the total variance) between the training data and the corresponding WL-CPLS estimate is shown by the solid red line in Figure 4.4. Observe a characteristic “elbow” when more than 40 components are used to calculate the WL-CPLS solution, beyond this point there was no significant information added by further components and the WL-CPLS model “over-fits”. This is demonstrated by the ensemble average prediction MSE for test data using the WL-CPLS estimator and shown by the dashed blue line in Figure 4.4. The MSE was lowest for 40 WL-CPLS components, which confirms that the number of WL-CPLS latent variables selected should be the number of independent components used to create the joint process.

**Stopping criteria for WL-CPLS algorithm**

Notice that the WL-CPLS formulations in Algorithm 6 and Algorithm 7 are iterated until $r$ components are found. The value of $r$ is often not known a priori but can be determined based on a stopping criterion. For example, the WL-CPLS solution is computed for a range of component numbers, $r$, then the prediction MSE for both training and test data
is determined, and the variable \( r \) is selected as the index of the last component after which adding a new component to the data no longer significantly improves the MSE. This is indicated by the “elbow” in Figure 4.4.

4.4.3 Prediction MSE for a varying SNR in \( X \) and \( Y \)

We next assessed the performance of the WL-CPLS in the presence of noise in both \( X \) and \( Y \). The data were generated as shown in Table 1 with \( N = 1000, r = 40, m = p = 100 \), while \( \sigma_X \) and \( \sigma_Y \) were varied so as to give a range of SNRs, defined as in (4.20), where \( Z \) is either \( X \) or \( Y \) and \( N \) is \( N_X \) or \( N_Y \) respectively.

**Remark 17.** The ordinary least squares (OLS) regression model assumes that the input variables, \( X \), are accurate and only the output variables, \( Y \), may contain error, however, this is not generally the case in real-world scenarios. The PLS aims to improve the estimate by using only the relevant subspace shared between the variables \( X \) and \( Y \), so as to eliminate spurious correlations from erroneous variables from the regression calculation.

Figure 4.5 shows the average prediction MSE (as a percentage of the total variance) of an ensemble of 100 realisations for the WL-CPLS solution obtained from training data (generated as before) with varying noise levels for \( X \) and \( Y \) (the noise level for \( X \) was defined similarly to the noise in \( Y \) above) where the training data \( X \) had 40 independent components mixed over \( m = 100 \) variables and transformed to 100 variables in \( Y \). For comparison, we used the Moore-Penrose pseudoinverse [35] to provide the inversions required in Section 4.3.2, and Figure 4.6 shows the average prediction MSE for the same ensemble as in Figure 4.5. Observe that the MSE of WL-CPLS was lower, especially for higher noise levels in input data \( X \).

![Figure 4.5. Prediction MSE against the noise level in \( Y \) (as a percentage of total variance), with noise also present in \( X \).](image-url)
Chapter 4. Widely Linear Complex Partial Least Squares

4.5 Distributed frequency estimation in power grids

The motivation behind the integration of multidimensional methods, such as complex-valued signal processing, was that their algebra allows certain relationships to be conveniently represented. To examine this for the WL-CPLS algorithm consider the problem complex-valued of frequency estimation in multi-node systems. This is important in modern smart grids [94] [95] [96], where any imbalance is indicated by a noncircular behaviour of a voltage phasor.

4.5.1 Frequency estimation: Problem specification

Consider a network of $M$ voltage sensors where each node has access to sampled three-phase voltage measurements, at the discrete time instant $k$, given by [94]

$$
\mathbf{s}_k = \begin{bmatrix}
v_{a,k} \\
v_{b,k} \\
v_{c,k}
\end{bmatrix} = \begin{bmatrix}
V_a \cos(\omega k + \phi_a) \\
V_b \cos(\omega k + \phi_b - \frac{2\pi}{3}) \\
V_c \cos(\omega k + \phi_c + \frac{2\pi}{3})
\end{bmatrix}.
$$

The amplitudes of the phase voltages $v_{a,k}$, $v_{b,k}$, $v_{c,k}$, are $V_a, V_b, V_c$, while the phase values are denoted by $\phi_a, \phi_b, \phi_c$ and the angular frequency is $\omega = 2\pi f T$, with $f$ the fundamental power system frequency. Observe that both the frequency $\omega$ and phasors (amplitude and phases) are assumed to be identical over a local area.

The three-phase representation of $\mathbf{s}_k$ in (4.30) is over-parameterised and is routinely represented as a compact “two-phase” Clarke voltage, $v_{a,k}$ and $v_{b,k}$ via the Clarke transform, given by [86]

$$
\begin{bmatrix}
v_{a,k} \\
v_{b,k}
\end{bmatrix} \overset{\text{def}}{=} \sqrt{\frac{2}{3}} \begin{bmatrix}
1 & -\frac{1}{2} & -\frac{1}{2} \\
\sqrt{3} & 0 & -\sqrt{3}
\end{bmatrix} \begin{bmatrix}
v_{a,k} \\
v_{b,k} \\
v_{c,k}
\end{bmatrix}.
$$

Moreover, the Clarke transform enables, $v_{a,k}$ and $v_{b,k}$, to be conveniently represented

Figure 4.6. Prediction obtained from the Moore-Penrose pseudoinverse against the noise level in $Y$ (as a percentage of total variance), with noise also present in $X$. 
4.5. Distributed frequency estimation in power grids

jointly as a complex-valued scalar,

\[ s_k \overset{\text{def}}{=} v_{\alpha,k} + jv_{\beta,k}. \] (4.32)

The complex \(\alpha\beta\) voltage in (4.32) therefore admits a widely linear auto-regressive (WLAR) representation given by \([31][86]\)

\[ s_k = h^* s_{k-1} + g^* s_{k-1}^*, \] (4.33)

where the WLAR coefficients \(h\) and \(g\) contain the information of the system frequency, \(\omega_i\), and level of imbalance in the system, that is, the degree of improperness. The system frequency is then calculated as \([86]\)

\[ e^{j\omega} = \text{Re}\{h\} + j\sqrt{\text{Im}^2\{h\} - |g|^2} \] (4.34)

\[ \Rightarrow \omega = \text{angle}\left\{ \text{Re}\{h\} + j\sqrt{\text{Im}^2\{h\} - |g|^2} \right\}. \] (4.35)

Note that if the system is in a balanced condition, only a single parameter, \(h\), is required to estimate the system frequency (a strictly linear system).

An important task in electricity grids is to estimate the system frequency, \(\omega\), given noisy observations of the Clarke voltage \(s_k\) in (4.32), which can be expressed

\[ z_{i,k} = s_k + \eta_{i,k}, \] (4.36)

where \(\eta_{i,k}\) is a zero-mean complex-valued white Gaussian noise process, with variance \(\sigma_\eta^2 = E|\eta_{i,k}|^2\).

4.5.2 Balanced multiple node case

The noisy voltage measurements in (4.36) at each node are given by

\[ z_{i,k} \overset{\text{def}}{=} \begin{bmatrix} z_{i,k}, & z_{i,k+1}, & \cdots, & z_{i,k+N-1} \end{bmatrix}^T. \]

To construct a classical strictly linear least squares problem, while exploiting all the measurements in the network, we fold a collection of voltage measurements at each node into a single column vector

\[ z_{-1} = \begin{bmatrix} z_{1,k-1}, & \cdots, & z_{M,k-1} \end{bmatrix}^T, \quad z = \begin{bmatrix} z_{1,k}, & \cdots, & z_{M,k} \end{bmatrix}^T, \] (4.37)

where the subscript “\(-1\)” indicates that the sample at a given time index is delayed by one time instant compared to the vector \(z\). This gives the formulation of the strictly linear least squares solution in the form

\[ \hat{h} = (z_{-1}^H z_{-1})^{-1} z_{-1}^H z, \] (4.38)
from which the system frequency is calculated as

\[ \hat{\omega}_{SL} = \text{angle}\{\hat{h}\}. \]

### 4.5.3 Unbalanced multiple node case

Three-phase systems under unbalanced conditions require a widely linear solution [86], given by

\[ z_{i,k} = h_i z_{i,k-1} + g_i z_{i,k-1}^*, \]  

(4.39)

where the regression can be represented through the augmented matrix of the vector of delayed system voltages, \( z_{-1} \), in (4.37), defined as

\[ z_{-1} = \begin{bmatrix} z_{-1}, & z_{-1}^* \end{bmatrix}, \]

such that the widely linear model of the voltage in (4.39) assumes an augmented form

\[ z = z_{-1} h^a, \]  

(4.40)

where \( h^a = [h, g]^H \). The WL-LS solution is then given by

\[ \hat{h}^a = \begin{bmatrix} h, & g \end{bmatrix} = Z^+ y, \]

where the frequency can be obtained from (4.35).

---

### 4.5.4 Exploiting redundancy with a WL-CPLS solution

The formulations proposed so far rearrange the data from each node into a single composite vector and, hence, destroy any spatial information. Alternatively, consider the following matrix of regression variables

\[
\begin{bmatrix}
z_{1,k-1}, & z_{2,k-1}, & \cdots, & z_{M,k-1} \\
z_{1,k}, & z_{2,k}, & \cdots, & z_{M,k} 
\end{bmatrix}, \quad
\begin{bmatrix}
z_{-1}, & z_{-1}^* 
\end{bmatrix},
\]

The problem now assumes a multivariate WL-LS form

\[ Z = \begin{bmatrix} Z_{-1}, & Z_{-1}^* \end{bmatrix} \]

Note that \( Z_{-1} \) is a rank-1 matrix (as the system voltage signal at each node should be the same) corrupted by noise, a natural scenario for the WL-CPLS algorithm as the widely linear OLS solution is computationally intractable. The augmented regression coefficient
4.5. Distributed frequency estimation in power grids

Unbalanced 3-Phase Power System
Frequency Estimation

```
<table>
<thead>
<tr>
<th>Node Sensor SNR (dB)</th>
<th>Estimator MSE (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>-</td>
</tr>
</tbody>
</table>
```

Figure 4.7. Comparison of MSE for an unbalanced 3-phase power grid frequency estimator, for a range of SNR.

matrix can now be expressed as

$$
B^a = \begin{bmatrix}
  h_1 & \cdots & \cdots \\
  \vdots & h_2 & \vdots \\
  \vdots & \vdots & \ddots \\
  \vdots & \vdots & \ddots & \cdots & h_M \\
  g_1 & \cdots & \cdots \\
  \vdots & g_2 & \vdots \\
  \vdots & \vdots & \ddots \\
  \vdots & \vdots & \ddots & \cdots & g_M \\
\end{bmatrix}
$$

From (4.33), the elements $h_1, h_2, \ldots, h_M$ and $g_1, g_2, \ldots, g_M$ are the coefficients of the WLAR model, for each node, while the matrix $B^a$ can be estimated using the WL-CPLS algorithm with a single component. This admits a new estimator of the grid frequency in the form

$$
\omega = \frac{1}{M} \sum_{i=1}^{M} \angle \left\{ \Re \{h_i\} + j \sqrt{\Im \{h_i\}^2 - |g_i|^2} \right\}.
$$

4.5.5 Simulation comparison of estimators

The performance of the three system frequency estimators was evaluated for a power grid in an unbalanced condition. Synthetic data were generated for a system with 100 nodes over a range of SNRs. The three phase voltages, $v_a$, $v_b$ and $v_c$, were generated as $f = 50Hz$ sinusoids sampled at $f_s = 5kHz$ for $t = 0.3s$ and with a $\frac{2\pi}{3}$ phase difference. The system imbalance was Type B sag which was caused by a resistive fault on $v_a$ causing it to drop to half the amplitude of the other phases, and the fault was propagated over $M = 100$ nodes corrupted with i.i.d. Gaussian noise to give the corresponding SNR. An ensemble MSE of the three system frequency estimators was calculated over 100 realisations over a range of SNR, and the results are given in Figure 4.7. Observe that the strictly linear estimator did not have enough degrees of freedom to provide an accurate result. The performance of the widely linear least squares estimator rapidly degraded for low SNRs, exhibiting a “blow up” at SNR=12dB. This can be attributed to the fact that autoregressive mod-
elling of sinusoids (strictly or widely linear) produces biased estimates in the presence of measurement noise [97]. The estimator derived from the WL-CPLS algorithm with only one component, however, provided an accurate estimate even for low SNRs. This demonstrates the ability of the WL-CPLS to find a common subspace between noisy data blocks, a common application in dimensionality reduction [98]. Moreover, the WL-CPLS allows for spatial information to be maintained and, since all the quantities (e.g. phasors, widely linear auto-regressive model) involved are complex-valued, the computation is performed in a physically meaningful manner\(^3\).

### 4.6 Data properties and the WL-CPLS performance

Section 3.7 explains the assumptions for the application of the strictly linear regression model \(Y = XB\). These assumptions also hold for the widely linear regression with the exception that the dependent variables \(Y\) are now a linear combination of the dependent variables \(X\) and their conjugate \(X^*\). Note that both the data and the model error are now complex-valued and the assumptions are in terms of complex augmented statistics. This means that the errors are uncorrelated and constant in terms of covariance and pseudocovariance.

It has been shown how the proposed WL-CPLS algorithm provides a PLS solution for a widely linear regression problem. In the context of the discussion in Section 3.7, which stated that the PLS latent variable decompositions also use (and are optimal under) the linear regression assumptions, the WL-CPLS latent variable decompositions are based on the widely linear regression assumptions and will remove spatial noise. As such, the WL-CPLS algorithm works well for situations where the data are conveniently represented by complex numbers and when there is high redundancy. For example, wave like data collected from an array.

The simulated data in Section 4.4 are generated from complex-valued Gaussian distributions. Note that impropriety in the error or variables does not break the model assumptions. This choice was made so that the WL-CPLS algorithm’s features could be exhibited independently of any property of the data. The smart-grid frequency estimation application in Section 4.5 uses a simulator to generate the signals. This mimics the real-world sinusoidal voltage signal but assumes that the errors are Gaussian. For a situation where this wasn’t the case then the performance would degrade. The strictly and widely linear estimator the WL-CPLS is compared to would similarly degrade, however, as the WL-CPLS solution exploits redundancy in a large network so the assumption is milder.

### 4.7 Conclusion

A widely linear complex partial least squares regression algorithm (WL-CPLS) has been derived as a generalisation of the original NIPALS algorithm, described in Section 3.3.1, for noncircular complex data. It has been shown that the proposed WL-CPLS provides

\(^3\)For example, the variable \(g_i\) in the WL complex solution indicates whether the system is balanced or not.
4.7. Conclusion

A latent variable decomposition of a data matrix, $X$, which, in turn, admits a tractable computation of the generalised inverses of the required matrices in order to calculate a widely linear regression. Moreover, the proposed algorithm has been shown to extend the cross-covariance criterion of real-valued PLS to select suitable components based on the maximum joint second-order information between the complex-valued input and output blocks. In this way, both significant data components are prioritised and the subspace containing the full relevant information is identified.

The derived WL-CPLS is a thorough generalisation to the complex-domain, taking into account the necessary complex-domain analogues of each of the PLS core components. This is further examined by showing that the properties of the WL-CPLS model residuals and convergence are equivalent to that of the real-domain algorithm and, as such, the result provides a parsimonious latent variable decomposition for regression with complex-valued data. This is further elucidated through synthetic simulation results.

The aim was to derive an algorithm which harnesses the utility of the multidimensional, complex number algebra. This was highlighted by the application of the WL-CPLS for complex covariance matrix diagonalisation. Here, it was shown that it provides a more efficient technique for subspace identification that the SUT. The benefits from the inherent structure of the complex-valued representation within the WL-CPLS framework have also been exemplified through real-world multi-node frequency estimation in unbalanced power grids. As a result, the proposed WL-CPLS has provided a step towards the thesis’ goal to produce multidimensional latent variable regression algorithms.
Chapter 5

Quaternion and Tensor Partial Least Squares

5.1 Introduction

Building on the previous chapter, the NIPALS algorithm is now developed for use with quaternion- and tensor-valued data. The quaternion version is derived in the same way as the WL-CPLS was for complex-valued data, through the quaternion analogues. The proposed algorithm provides a quaternion PLS solution (WL-QPLS) for the widely linear estimation scenario outlined in Section 2.3.2. As a result, the scores are 4D quaternion latent variables, and are orthogonal in the quaternion algebra. The implementation of the algorithm as an quaternion covariance matrix diagonalisation tool is then presented.

The PLS algorithm has already been extended for tensor-based data including the HO-PLS [99] using a restricted Tucker representation and the N-way PLS [72] which uses a canonical polyadic decomposition (CPD) format. This Chapter re-examines the NIPALS algorithm in terms of tensor-based operations and the PLS core components outlined in Section 3.3.4 to derive two new tensor PLS algorithms, the generalised HOPLS (GHO-PLS) and the higher-order NIPALS (HONIPALS). These provide a general, unrestricted multilinear rank Tucker solution utilising the full range of available tensor operations.

The focus of the Chapter then turns to the comparison of these multidiimensional PLS algorithms. It has been shown that complex and quaternion-valued data can be processed in the real domain owing to the isomorphisms in Section 2.3. This is extended through a further transform between quaternions and tensors. The derived transform is then utilised to show that each of the three proposed algorithms process quadrivariate data equivalently but in alternative ways. The main difference between the calculations is the choice of the
cross-covariance data structure and its eigen-type decomposition. It is shown that although
the HONIPALS and WL-QPLS calculate and decompose the cross-covariance matrix in
the same way, the transformation back to the quaternion domain and the corresponding
processing yield different overall results. Moreover, it is shown that the GHOPLS and
HONIPALS cross-covariance structures share the same elements but their decompositions
are in general different. As a result, each algorithm provides an alternative solution. This
is further investigated through simulation and is demonstrated by the application for image
classification of coffee plantations from satellite data.

5.2 The Quaternion NIPALS algorithm

In this section an extension of the real-valued NIPALS Algorithm 3 is introduced. Previ-
ous work on PLS for quaternion-valued data can be found in Via et al. [48] who shows
that the PLS result should have a widely linear form. This PLS method is aimed at data
block cross-correlation analysis and is a generalisation of PLS-SB described in Section
3.3.2, in the same way that Schreier did for complex-valued data [88]. As a result, the
obtained decomposition is useful for an analysis of the directions of cross-covariance be-
tween the input and output blocks, but the components may not be orthogonal and it is
not the most parsimonious for regression. The proposed extension of the NIPALS PLS
algorithm for quaternion-valued data, instead, produces a regularised PLS solution for
the quaternion widely linear regression problem in (2.31) based on an orthogonal latent
variable decomposition.

The proposed quaternion widely linear PLS (WL-QPLS) algorithm first calculates a
vector component \( t \in \mathbb{H}^N \) from the input matrix \( X \in \mathbb{H}^{N \times m} \) which represents the maximal
cross-covariance with the output matrix \( Y \in \mathbb{H}^{N \times n} \). The vector component \( t \) is given as

\[
t = Xw,
\]

In order to find the vector, \( w \), the augmented versions of the input and output matrices
are transformed to the real domain as \( X_{Re} \in \mathbb{R}^{N \times 4m} \) and \( Y_{Re} \in \mathbb{R}^{N \times 4n} \) through the
isomorphism introduced in (2.26). The real-valued vector, \( w_{Re} \in \mathbb{R}^{4m} \), is found from the
solution to the optimisation problem

\[
w_{Re} = \arg \max_{||w_{Re}|| = 1} ||w_{Re}^T X_{Re}^T Y_{Re}||^2.
\]

(5.1)

which yields the leading eigenvector of \( X_{Re}^T Y_{Re} Y_{Re}^T X_{Re} \). This is then cast back into the
quaternion-domain through the relation

\[
[w^T, w^iT, w^jT, w^kT] = \Gamma_m w_{Re}.
\]

(5.2)

In this way, the component chosen at each iteration of the WL-QPLS algorithm, \( t \), rep-
resents maximal joint second-order information between the \( X \) and \( Y \) blocks. The com-
putation of \( w_{Re} \) in the real domain does not affect the quaternion representation [28] [48]
and is not equivalent to a real-valued NIPALS implementation, as was the case with the
WL-CPLS.

Following the calculation of the vector, $t$, its relation to the $X$ and $Y$ blocks is found. This relationship is described by a quaternion widely linear regression given in augmented form as

\[ X = t\overline{p}^H, \quad Y = t\overline{c}^H, \]  

where the vectors $\overline{p} = [p_1, p_2, p_3, p_4]$ and $\overline{c} = [c_1, c_2, c_3, c_4]$ represent the widely linear regression coefficients and are calculated as

\[ \overline{p} = (t^+ X)^H, \quad \overline{c} = (t^+ Y)^H. \]  

To calculate the next PLS component, the matrices $X$ and $Y$ must be “deflated”, that is, the impact of the current score $t$ must be removed. The deflation is performed as

\[ X_{i+1} = X_i - t_i\overline{p}_i^H, \quad Y_{i+1} = Y_i - t_i\overline{c}_i^H, \]  

where the subscript $i$ represents the current iteration number. The matrices $X_{i+1}$ and $Y_{i+1}$ are used in place of $X_i$ and $Y_i$ on the next iteration. As such, the latent variables are mutually orthogonal which leads to a parsimonious representation of the joint process in the input and output blocks.

The full WL-QPLS procedure is summarised in Algorithm 11. After every iteration the vectors $t$ are stored into the columns of the matrices $T$, $\overline{p}$ and $\overline{c}$ are stored as $\overline{P} = [P_1, P_2, P_3, P_4]$ and $\overline{C} = [C_1, C_2, C_3, C_4]$ which leads to the widely linear relationships

\[ \tilde{X} = TP^H, \quad \tilde{Y} = TC^H. \]  

Therefore, the input $X$ and output $Y$ are both represented in terms of the new latent variables $T$. This provides a natural representation of the joint process and represents a useful decomposition to calculate the regression coefficients, $B$, for the widely linear regression

\[ \tilde{Y} = \tilde{X}B. \]  

This is achieved through

\[ B = \tilde{X}^+\tilde{Y}. \]  

Calculating the regression coefficients based on the WL-QPLS model means that only shared information is included in the calculation.

In addition to the relations (5.6), an equivalent version of the real-valued PLS relationship (3.5) can be found, whereby the score within the data block $Y$ is calculated as

\[ u = Yc_1 + Y^i c_2 + Y^j c_3 + Y^k c_4, \]

with the respective loadings, from the augmented scores $u$, given by

\[ q = (u^+ Y)^H. \]
5.2. The Quaternion NIPALS algorithm

Algorithm 11. The NIPALS algorithm for widely linear quaternion PLS (WL-QPLS)

1: Initialise: $X_1 = [X, X^i, X^j, X^k]$, $Y_1 = [Y, Y^i, Y^j, Y^k]$
2: for $i = 1, \ldots, r$ do
3: $X_{i,Re} = X_i \Gamma_m$, and $Y_{i,Re} = Y_i \Gamma_n$
4: $w_{i,Re} = \text{Eig}_{\text{max}}\{X_{i,Re}^T Y_{i,Re} Y_{i,Re}^T X_{i,Re}\}$
5: $[w_i^T, w_i^T, w_i^T, w_i^T]^T = \Gamma_m w_{i,Re}$
6: $t_i = X_i w_i$, $c_i = [t_i, t_i, t_i, t_i]$  
7: $\overline{c}_i = (t_i^T Y_i)^H$
8: $p_i = (t_i^T X_i)^H$
9: $X_{i+1} = X_i - t_i p_i^H$, $Y_{i+1} = Y_i - t_i c_i^H$
10: Store $t_i$, $p_{1,i}$, $c_i$ and $w_i$
11: end for

Alternative cross-covariance criterion

The choice of the basis vector $w$ from (5.2) can have alternative forms, as in some situations it may be desired to maximise purely the cross-covariance (rather than considering the full second-order statistics) whereby the optimisation problem is given by

$$w = \text{arg max}_{||w||=1} ||w^H X^H Y||^2_2.$$  

The vectors $w$ are then calculated as the left leading singular vector of the matrix $X^H Y$.

This and other choices allow latent variables to be selected in a variety of bases. The full alternative WL-QPLS algorithm is given in Algorithm 12.

Algorithm 12. The WL-QPLS algorithm with an alternative cross-covariance criterion

1: Initialise: $X_1 = [X, X^i, X^j, X^k]$, $Y_1 = [Y, Y^i, Y^j, Y^k]$
2: for $i = 1, \ldots, r$ do
3: $w_i = \text{Eig}_{\text{max}}\{X_i^H Y_i Y_i^H X_i\}$
4: $t_i = X_i w_i$, $c_i = [t_i, t_i, t_i, t_i]$  
5: $\overline{c}_i = (t_i^T Y_i)^H$
6: $p_i = (t_i^T X_i)^H$
7: $X_{i+1} = X_i - t_i p_i^H$, $Y_{i+1} = Y_i - t_i c_i^H$
8: Store $t_i$, $p_{1,i}$, $c_i$ and $w_i$
9: end for

5.2.1 Quaternion covariance matrix diagonalisation

The WL-QPLS algorithm can me implemented as a tool for the diagonalisation of the quaternion empirical covariance matrices in Section 2.3.2, given by $X^H X$, $X^i H X$, $X^j H X$ and $X^k H X$. This result is analogous to the WL-CPLS result for complex-valued data in Section 4.3.3. Covariance matrix diagonalisation applications are ubiquitous in real-valued signal processing, as discussed in Section 3.1, where the goal is to provide a decomposition $X = TP^T$ where $T \in \mathbb{R}^{n \times r}$ are the $r$ “principal components” and $P \in \mathbb{R}^{m \times r}$ are the $r$ leading eigenvectors from the covariance matrix $X^T X$.

In Section 2.3.2 it was shown that to fully cater for quaternion-valued second-order
statistics, the four covariance matrices $E[X^H X], E[X_i^H X], E[X_j^H X]$ and $E[X_k^H X]$ must be considered. To that end, quaternion covariance matrix diagonalisation requires all four matrices to be simultaneously diagonalised as well as the traditional covariance matrix. An extension of PCA for quaternion-valued data, QPCA, has been developed in [100] [101]. The result is obtained from the quaternion SVD of the quaternion covariance matrix,

$$X^H X = U \Sigma U^H$$

as a direct extension of the real-valued PCA. However, in this way the full augmented statistics are not considered and so the components are not fully uncorrelated. Xiang et. al. [102] show that an extension of the complex-valued SUT for quaternions, the QUT, can only diagonalise the quaternion covariance matrix and one complementary covariance matrix at a time. Furthermore, the quaternion approximate uncorrelating transform (QAUT) can approximately diagonalise the covariance matrix along with the three complementary covariance matrices.

Owing to the ability of the WL-QPLS to calculate orthogonal latent-variables, $T$, it consequently produces a decomposition of the input variables, $X$, which has diagonalised all the quaternion covariance matrices. In fact, the WL-QPLS algorithm can be implemented with both the input and output being the same matrix, $X$, with the aim of producing a diagonalising transform. This diagonalisation is given by

$$T = X \Theta^+,$$

where the transform matrix $\Theta$ is given by

$$\Theta = \begin{pmatrix}
(P_1^H W) W^H & (P_2^H W) W^H & (P_3^H W) W^H & (P_4^H W) W^H \\
(P_1^H W) W^H & (P_2^H W) W^H & (P_3^H W) W^H & (P_4^H W) W^H \\
(P_1^H W) W^H & (P_2^H W) W^H & (P_3^H W) W^H & (P_4^H W) W^H \\
(P_1^H W) W^H & (P_2^H W) W^H & (P_3^H W) W^H & (P_4^H W) W^H \\
\end{pmatrix}.$$

**Remark 18.** The WL-QPLS in Algorithm 11 with the input and output blocks given by the quaternion-valued matrix $X$ yields a score matrix $T$ which is fully uncorrelated in terms of quaternion-valued second-order statistics. The augmented covariance matrix, $T^H T$, of the WL-QPLS scores is now block diagonal which means that all four covariance matrices are diagonalised.

**Remark 19.** If, on each iteration of the WL-QPLS Algorithm 11, the score vector $t$ is normalised, then the complementary covariance matrices of the score matrix, $T$, will yield their circularity quotients in (2.24).

### 5.2.2 The convergence of the WL-QPLS algorithm

The proposed WL-QPLS algorithm is a generalisation of the real-valued NIPALS algorithm described in Section 3.3.1 for use with noncircular quaternion-valued data, in the same way the WL-CPLS is a generalisation for noncircular complex-valued data. To analyse
their similarities we now examine the properties of the model residuals and convergence in comparison to the real-valued PLS results in Section 3.3.3 and the WL-CPLS in Section 4.3.4.

5.2. The Quaternion NIPALS algorithm

5.2.3 The orthogonality of the WL-QPLS model residuals

The WL-QPLS model yields the same orthogonality property as the real-valued NIPALS case and the WL-CPLS. The residuals for the model obtained on the \( l \)-th iteration are given by

\[
X_{l+1} = X - T_l \overline{P}_l^H,
\]

from the relationship (5.5), where \( \overline{P}_l = [P_{1,l}, P_{2,l}, P_{3,l}, P_{4,l}] \) and is calculated as

\[
\overline{P}_l = ((T_l^H T_l)^{-1} T_l^H X)^H,
\]

Therefore, the residuals can be expressed in the same way as in (4.26). Using the projection matrix \( P_{t,l} = T_l (T_l^H T_l)^{-1} T_l^H \) (which is a projection onto the space spanned by \( T_l \)) then produces the form

\[
X_{l+1} = (I - P_{t,l}) X,
\]

so that the score obtained on the \((l+1)\)-th iteration is orthogonal to any from the previous \( l \) iterations. The same result is also obtained for the output \( Y \) block as

\[
Y_{l+1} = (I - P_{t,l}) Y.
\]

Proof of WL-QPLS convergence to a Krylov space

To goal is to show that the WL-QPLS result for a univariate output, \( y \), converges to a Krylov space, as was the case for the real-valued and complex-valued counterparts. To this end, consider the WL-QPLS using the alternative cross-covariance criterion in Algorithm 12 for the case of a univariate output, \( y \). The leading eigenvector of the matrix \( X_l^H y_l y_l^H X_l \) (the calculation required to obtain the alternative cross-covariance basis vector) is given as \( w_l = \frac{X_l^H y_l y_l^H X_l}{||X_l^H y_l||} \) and so \( w_l \propto X_l^H y \) as in [73]. Coupled with the orthogonality properties of the residuals outlined earlier, a recurrence relation for the WL-QPLS scores is then given as

\[
t_{l+1} \propto (I - P_{t,l}) XX^H (I - P_{t,l}) y,
\]

and is akin to the WL-CPLS result.

**Proposition 5.** The space defined by the vectors in the matrix \( K_l = [s, SK_{l-1}, SK_{l-1}^i, SK_{l-1}^{j,k}] \) concatenated with its \( \alpha = i, j, k \) conjugates \( K_l^{\alpha} \), where \( s = XX^H y \), \( S = XX^H \) and \( K_1 = [s] \), forms a basis for the augmented scores matrix, \( T_l = [T_l, T_l^i, T_l^j, T_l^{k}] \), produced from the WL-QPLS in Algorithm 12.

**Proof.** The proof follows the same induction structure as that for Proposition 1 and 4 where, for the WL-QPLS version, so that the first score is now given as \( t_1 \propto [s, s^i, s^j, s^k] \) = \([K_1, K_1^i, K_1^j, K_1^k]\). Inducing that the columns of \( T_l \) are a linear combination of the columns
Chapter 5. Quaternion and Tensor Partial Least Squares

5.3 Multilinear PLS-regression algorithms

Algorithms for a PLS solution have been derived for (hyper)complex-valued data. These algorithms take into account the constraints presented by their algebra to produce a latent variable decomposition for a regression equipped with the required degrees of freedom.

Tensors offer another type of multidimensional data structure. Whereas the complex and quaternion methods are underpinned by a number system governed by a division algebra, the tensor-domain is equipped with a multilinear algebra, introduced in Section 2.3.3.

In this section the main existing tensor-based PLS algorithm, known as the higher order PLS (HOPLS), is first reviewed. The generalised HOPLS (GHOPLS) and higher order NIPALS (HONIPALS) algorithms are then proposed as an extension of the two-way NIPALS algorithm to counteract the restrictions of the HOPLS result and utilise the range of available tensor processing methods.

5.3.1 The HOPLS algorithm

Zhao et al. [99] introduced a tensor-based PLS algorithm, HOPLS, which provides a tensor regression between the input data tensor $X \in \mathbb{R}^{N \times I_1 \times I_2 \times \cdots \times I_M}$ and the output data tensor $Y \in \mathbb{R}^{N \times J_1 \times J_2 \times \cdots \times J_P}$, introduced in Section 2.3.3. The HOPLS algorithm decomposes $X$ and $Y$ into a sum of $r$ rank-(1, $K_1$, $K_2$, $\ldots$, $K_M$) and rank-(1, $L_1$, $L_2$, $\ldots$, $L_P$) components respectively. This yields the decompositions

\[
X = \sum_{i=1}^{r} \mathcal{G}_i \times_1 t_i \times_2 P^{(1)}_i \times_3 P^{(2)}_i \times_4 \cdots \times_m P^{(m)}_i, \\
Y = \sum_{i=1}^{r} \mathcal{D}_i \times_1 s_i \times_2 Q^{(1)}_i \times_3 Q^{(2)}_i \times_4 \cdots \times_p Q^{(p)}_i,
\]

(5.9)

where $\mathcal{G} \in \mathbb{R}^{1 \times K_1 \times K_2 \times \cdots \times K_M}$ and $\mathcal{D} \in \mathbb{R}^{1 \times L_1 \times L_2 \times \cdots \times L_P}$ are the “core” tensors of the Tucker and $P^{(n)} \in \mathbb{R}^{K_n \times I_n}$ $n = 1, 2, \ldots, M$, $Q^{(n)} \in \mathbb{R}^{I_n \times J_n}$ $n = 1, 2, \ldots, P$ are factor matrices.

The full HOPLS process is outlined in Algorithm 13. The first step in each iteration is to compute the contraction product (2.34) between $X$ and $Y$ along the first mode. The

in $[K_l, K^l_1, K^l_2, K^l_3]$ (and as such $P_{l,l}$) the recurrence relation yields

\[
t_{l+1}^{1} \propto s - S P_{l,l} y - P_{l,l} s + P_{l,l} S P_{l,l} y,
\]

\[
t_{l+1}^{j} \propto s^{j} - S^{j} P_{l,l} y^{j} - P_{l,l} s^{j} + P_{l,l} S^{j} P_{l,l} y^{j},
\]

\[
t_{l+1}^{k} \propto s^{k} - S^{k} P_{l,l} y^{k} - P_{l,l} s^{k} + P_{l,l} S^{k} P_{l,l} y^{k}.
\]

The columns of $T_{l+1}$ are then a linear combination of the columns in $[K_{l+1}, K^l_1, K^l_2, K^l_3]$. Therefore, the induced hypothesis holds and it is true for the base case where $l = 1$ and so we have proven Proposition 5.
5.3. Multilinear PLS-regression algorithms

orthogonal rank-($K_1, K_2, \ldots, K_M, L_1, L_2, \ldots, L_P$) Tucker decomposition of the resultant tensor, $S$, is then obtained through the HOOI Algorithm 2 [55]. The HOPLS score vector $t$ is found as the first singular vector of the the unfolded matrix $(X_i \times_2 P_i^{(1)T} \times_3 P_i^{(2)T} \ldots \times_{M+1} P_i^{(M)T})$, where $P^{(n)}$ for $n = 1, 2, \ldots, M$ are the factor matrices produced from the prior Tucker decomposition of $S$. The respective core tensors, $G$ and $D$, for the HOPLS decomposition of $X$ and $Y$ can then be solved for.

**Algorithm 13.** The HOPLS algorithm [99]

1: **Input:** $X$, $Y$, $r$, $K_n$ for $n = 1, 2, \ldots, M$ and $L_n$ for $n = 1, 2, \ldots, P$
2: **for** $i = 1, \ldots, r$ **do**
3: \hspace{1em} $S_i = \langle X_i, Y_i \rangle_{(1,1)}$
4: \hspace{1em} Perform rank-($K_1, K_2, \ldots, K_M, L_1, L_2, \ldots, L_P$) HOOI decomposition to give $S = G_i^C \times_1 P_i^{(1)T} \times_2 P_i^{(2)T} \times_3 \ldots \times_M P_i^{(M)T} \times_{M+1} Q_i^{(1)T} \times_{M+2} Q_i^{(2)T} \times_3 \ldots \times_{M+P} Q_i^{(P)T}$
5: \hspace{1em} $t_i$ is first leading left singular vector of $(X_i \times_2 P_i^{(1)T} \times_3 P_i^{(2)T} \ldots \times_{M+1} P_i^{(M)T})$
6: \hspace{1em} $G_i = X_i \times_1 t_i^T \times_2 P_i^{(1)T} \times_3 P_i^{(2)T} \times_4 \ldots \times_{M+1} P_i^{(M)T}$
7: \hspace{1em} $D_i = Y_i \times_1 t_i^T \times_2 Q_i^{(1)T} \times_3 Q_i^{(2)T} \times_4 \ldots \times_{P+1} Q_i^{(P)T}$
8: \hspace{1em} $X_{i+1} = X_i - G_i \times_1 t_i \times_2 P_i^{(1)} \times_3 P_i^{(2)} \times_4 \ldots \times_{M+1} P_i^{(M)}$
9: \hspace{1em} $Y_{i+1} = Y_i - D_i \times_1 t_i \times_2 Q_i^{(1)} \times_3 Q_i^{(2)} \times_4 \ldots \times_{P+1} Q_i^{(P)}$
10: **end for**

In comparison to the two-way NIPALS PLS, the score at each iteration is also a vector component regardless of the order of the original data. Although this kind of decomposition can be extremely useful [103], a generalisation of the NIPALS algorithm for tensor component regardless of the order of the original data. Although this kind of decomposition would allow a more flexible decomposition.

To calculate the tensor regression coefficients for prediction using the HOPLS model, a mode-1 unfolding of the latent variable representation of $X$ is first performed

$$X_{(1)} = T_{(1)} V,$$

(5.10)

where $T_{(1)}$ is a matrix of $r$ columns which contains the $r$ score vectors obtained by the HOPLS algorithm and $V$ is a matrix of $r$ rows with each row, $v_r$, given by

$$v_r = G_{(1)r}(P^{(M)} \otimes P^{(M-1)} \otimes \ldots \otimes P^{(1)})^T.$$

A mode-1 unfolding of $Y$ is performed as [99]

$$Y_{(1)} = T_{(1)} Q^*,$$

(5.11)

where the matrix $Q^*$ has $r$ rows, $q^*_r$, given by

$$q^*_r = D_{(1)r}(Q^{(P)} \otimes Q^{(P-1)} \otimes \ldots \otimes Q^{(1)})^T.$$

Now the prediction of the mode-1 unfolded dependent variables, $Y_{(1)}$, can be performed as

$$Y_{(1)} = X_{(1)} V^* Q^*,$$

(5.12)
Chapter 5. Quaternion and Tensor Partial Least Squares

Table 5.1. Comparison of two-way PLS and HOPLS algorithms

<table>
<thead>
<tr>
<th>Algorithm Step</th>
<th>two-way PLS</th>
<th>HOPLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-covariance structure</td>
<td>( S_i = X_i^T Y_i )</td>
<td>( S_i = { \langle X_i, Y_i \rangle }_{i=1} )</td>
</tr>
<tr>
<td></td>
<td>The cross-covariance matrix</td>
<td>Tensor-based contraction product</td>
</tr>
<tr>
<td>Eigen-decomposition</td>
<td>( w_i = \text{ Eig}_{\max} { S_i S_i^T } )</td>
<td>HOOI decomposition of ( S_i )</td>
</tr>
<tr>
<td>Score calculation</td>
<td>( t_i = X_i w_i )</td>
<td>( t_i ) is first leading left singular</td>
</tr>
<tr>
<td></td>
<td>The projection of ( X_i )</td>
<td>vector of the projection of the obtained</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tucker factor matrices onto the unfolded</td>
</tr>
<tr>
<td>LV decomposition/Deflation</td>
<td>( X_{i+1} = X_i - t_i p_i^T )</td>
<td>( X_{i+1} = X_i - G_{i} \times_1 t_i \times_2 P_{1}^{(1)} \times_3 P_{i}^{(2)} \times_4 \cdots \times_{M+1} P_{M}^{(M)} )</td>
</tr>
<tr>
<td></td>
<td>( Y_{i+1} = Y_i - t_i c_i^T )</td>
<td>( Y_{i+1} = Y_i - D_{i} \times_1 t_i \times_2 Q_{i}^{(1)} \times_3 Q_{i}^{(2)} \times_4 \cdots \times_{P+1} Q_{P}^{(P)} )</td>
</tr>
</tbody>
</table>

where each column \( v_r^+ \) of \( V^+ \) is calculated as

\[
v_r^+ = (P^{(M)} \otimes P^{(M-1)} \otimes \cdots \otimes P^{(1)}) G^{(1)} r.
\]

The HOPLS with respect to the PLS core components

The core components for a NIPALS PLS-regression algorithm have been outlined in Section 3.3.4 and a comparison of the HOPLS and two-way PLS algorithms is given in Table 5.1.

5.3.2 A NIPALS algorithm for tensor-variate PLS regression

A general tensor-variate NIPALS style PLS-regression algorithm for the input data tensor, \( X \in \mathbb{R}^{N \times I_1 \times I_2 \times \cdots \times I_M} \), and the output data tensor, \( Y \in \mathbb{R}^{N \times J_1 \times J_2 \times \cdots \times J_P} \), iteratively calculates a desired low-rank approximation of the original data tensors. The mode-1 factor matrix of the decomposition of a data tensor constitutes the tensor-variate scores which are found based upon a cross-covariance criterion. Note that the tensor-variate samples are assumed to be stacked along the first mode. In every iteration, the impact of the obtained tensor score is then removed from the data tensors through deflation. The Tucker decomposition of the outputs \( Y \) is also performed in terms of the scores produced from the inputs \( X \), in order to create a decomposition for a parsimonious regression. The proposed algorithms are derived as generic tensor extensions of the four core components of a NIPALS PLS-regression algorithm outlined in Section 3.3.4.

The first step of each iteration is to calculate a cross-covariance structure. One such tensor-based method (employed by the HOPLS) is to calculate the contraction product in (2.34) along the first mode of the input tensor, \( X \), and the output tensor, \( Y \), to give the
structure

$$\mathcal{S} = \langle \mathbf{X}, \mathbf{Y} \rangle_{\{1,1\}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M \times J_1 \times J_2 \times \cdots \times J_P}. \tag{5.13}$$

Alternatively, the data tensors can be unfolded along the first mode, as is done in the HOSVD algorithm [55], which yields the matrix cross-covariance structure

$$\mathbf{S} = \mathbf{X}^T_\{1\} \mathbf{Y}_\{1\} \in \mathbb{R}^{I_1 I_2 \cdots I_M \times J_1 \times J_2 \times \cdots \times J_P}. \tag{5.14}$$

Note that the mode-wise variations are encoded explicitly in the modes of the contraction product, $\mathbf{S}$. However, for the matrix, $\mathbf{S}$, obtained from the unfolded data tensors, they implicitly integrated.

The next step of the algorithm is to perform an eigen-type decomposition of the cross-covariance data structures in order to determine a spatial basis which describes the maximal cross-covariance between the inputs and the outputs. The HOPLS algorithm already employs a contraction product based cross-covariance structure. The eigen-type decomposition is then given by the HOOI algorithm which finds orthogonal factor tensors that produce a rank-$\{K_1, K_2, \ldots, K_m, L_1, L_2, \ldots, L_p\}$ Tucker approximation

$$\mathbf{S} = \mathbf{G}^{C \times 1} \mathbf{P}^{(1)} \times_2 \mathbf{P}^{(2)} \times_3 \cdots \times_1 \mathbf{P}^{(M)} \times_{M+1} \mathbf{Q}^{(1)} \times_{M+2} \mathbf{Q}^{(2)} \times_{M+3} \cdots \times_{M+p} \mathbf{Q}^{(P)} \tag{5.15}.$$  

The HOOI Algorithm 2 gives an alternating least squares (ALS) type solution to the optimisation problem of finding the best Tucker approximation of a tensor. Note that other methods can be used to calculate such a decomposition including the HOSVD. For some input data structures (e.g. with a third order tensor and a vector as an output) the contraction product in (5.13) produces a matrix. In such cases, the decomposition is instead provided by the SVD the the resultant matrix.

Now consider an alternative case where the unfolded cross-covariance structure, $\mathbf{S}_{\{1\}}$, is implemented, so that the eigen-problem is determined as

$$\mathbf{w} = \arg \max_{||\mathbf{w}||_2 = 1} ||\mathbf{w}^T \mathbf{X}_{\{1\}} \mathbf{Y}_{\{1\}}||_2^2. \tag{5.16}$$

The result is obtained from the leading eigenvector of the matrix $\mathbf{SS}^T$, that is, from the SVD

$$\mathbf{S} = \mathbf{W} \mathbf{\Sigma} \mathbf{V}^T. \tag{5.17}$$

In both cases, the scores in a tensor PLS can be represented first as a matrix, $\mathbf{T}_{\{1\}} \in \mathbb{R}^{N \times K}$ where $K$ is the desired number of first mode fibres. For a general PLS, the factor matrix can then be folded into any structure required as the natural structure for a score in a given implementation. To find the scores factor matrix, the eigendecomposition result is used to find a projection of the inputs $\mathbf{X}$. This can be achieved from a matrix unfolding as

$$\mathbf{T}_{\{1\}} = \mathbf{X}_{\{1\}} \mathbf{W}_{\{1\}} \in \mathbb{R}^{N \times K}. \tag{5.18}$$

When the unfolded cross-covariance data structure, $\mathbf{S}$ in (5.14), is used, the matrix $\mathbf{W}_{\{1\}} \in \mathbb{R}^{I_1 I_2 \cdots I_m \times K}$ is given by the first $K$ left singular vectors through its SVD, that is, the first
$K$ columns of $W$ from (5.17). On the other hand, to compute the scores from the cross-covariance structure $S$ in (5.13), the matrix, $W_{(1)}$, takes the form

$$W_{(1)} = (P^{(M)}T \otimes P^{(M-1)}T \otimes \ldots \otimes P^{(1)}T)^T,$$

(5.19)

which is the Kronecker product unfolded form of the factor matrices obtained from the Tucker decomposition in (5.15) where $K = K_1 K_2 \ldots K_M$. Alternatively, the score tensor can be computed using tensor mode-$n$ products as

$$T = X \times_2 P^{(1)}T \times_3 P^{(2)}T \times_4 \ldots \times_{M+1} P^{(M)}T.$$

(5.20)

**Remark 20.** The structure of the score tensor $T$ is free to be chosen and depends on the form suggested by the data. When the scores are calculated from the Tucker decomposition of the contraction product, $S$ in (5.15), the rank of the decomposition gives the dimensions of the score tensor $T \in \mathbb{R}^{N \times K_1 \times K_2 \times \ldots \times K_M}$. On the other hand, when the scores are calculated from the SVD in (5.17), the choice of folding is not intrinsic to the calculation.

The score tensor, $T$, is now regressed to the data tensors, $X$ and $Y$, through the parameters $G$ and $D$ respectively, to give the relations

$$X = f(T; G), \quad Y = f(T; D),$$

(5.21)

where the parameters $G$ and $D$ are calculated

$$G = X \times_1 T_{(1)}^+, \quad D = Y \times_1 T_{(1)}^+.\quad (5.22)$$

Prior to the next iteration, the impact of the score tensor is removed through the deflation process

$$X_{i+1} = X_i - G_i \times_1 T_{(1),i}, \quad Y_{i+1} = Y_i - D_i \times_1 T_{(1),i},$$

(5.23)

where $i$ denotes the current iteration and the tensors $X_{i+1}$ and $Y_{i+1}$ are used in place of $X_i$ and $Y_i$ in the next iteration. The sum of these tensor regressions provides the tensor version of the low-rank PLS approximations, akin to the two-way solution.

Notice that the factor matrices for the Tucker decomposition of the latent variable regression tensors $G$ and $D$ are not computed inherently as in HOPLS, where they are obtained from the decomposition of $S$, and are not required for computation. Here, they are instead obtained by computing a Tucker decomposition (i.e. the HOSVD) for each regression tensor, $G$ and $D$ obtained in (5.22), to give the factor matrices$^1$ $P^{(n)}_i \in \mathbb{R}^{K_n \times I_n}$ $n = 1, 2, \ldots, M$ and $C^{(n)}_i \in \mathbb{R}^{K_n \times J_n}$ $n = 1, 2, \ldots, P$ respectively. This leads to the expressions

$$X = G \times_1 T_{(1)} \times_2 P^{(1)} \times_3 P^{(2)} \times_4 \ldots \times_{M+1} P^{(M)},$$

$$Y = D \times_1 T_{(1)} \times_2 C^{(1)} \times_3 C^{(2)} \times_4 \ldots \times_{P+1} C^{(P)},$$

(5.24)

where $G$ and $D$ are now the core tensors within the Tucker decompositions and not the

$^1$Note that these matrices are not the same as in (5.15)
A further decomposition of the output data tensor $\mathbf{Y}$ that is analogous to the two-way PLS expression in (3.5) is next obtained. In this case the impact of the unfolded latent variables $\mathbf{T}(1)$ in the data tensor $\mathbf{Y}$ is found as

$$\mathbf{C} = \mathbf{T}(1) \mathbf{Y}(1), \quad \mathbf{U}(1) = \mathbf{Y}(1) \mathbf{C}^\top.$$

These derived latent variables are related to $\mathbf{Y}$ through the regression tensor $\mathbf{E}$, calculated as

$$\mathbf{E} = \mathbf{Y} \times_1 \mathbf{U}(1)^\top.$$

As is the case for the latent variable regression tensors $\mathbf{G}$ and $\mathbf{D}$, the tensor $\mathbf{E}$ is decomposed using the HOSVD to give the factor matrices $\mathbf{Q}^{(n)}_i \in \mathbb{R}^{K_n \times J_n}$, $n = 1, 2, \ldots, P$ leading to the relation

$$\mathbf{Y} = \mathbf{E} \times_1 \mathbf{U}(1) \times_2 \mathbf{Q}^{(1)}_1 \times_3 \mathbf{Q}^{(2)}_2 \times_4 \cdots \times_P \mathbf{Q}^{(P)}_P.$$

(5.25)

Two separate algorithms are now introduced: (i) The Higher Order NIPALS (HONIPALS) which stems from the unfolded cross-covariance structure in (5.14) and its SVD, as summarised in Algorithm 14, (ii) The Generalised HOPLS (GHOPLS) which is based on the contraction product in (5.13) and its Tucker decomposition, as summarised in Algorithm 15. Once the score tensor $\mathbf{T}$ has been obtained, the steps to produce the latent variable representations in (5.24) and the subsequent deflation in (5.23) are the same for both algorithms regardless of the choice of the eigen-decomposition/cross-covariance structure. Although the GHOPLS and HOPLS algorithms share the same cross-covariance and eigen-decomposition steps, they differ as the score in GHOPLS is not generally produced as a single vector as the case with HOPLS, but can be selected as desired. The deflation steps also differ as, for the HOPLS solution, the factor matrices in (5.9) are calculated directly from the HOOI decomposition of the contraction product whereas, for the GHOPLS solution, the deflation is given by a tensor regression. The number of iterations, $r$, is chosen as the number of latent variable components required to well approximate the data tensors $\mathbf{X}$ and $\mathbf{Y}$, and this number is usually determined from cross-validation.

**Algorithm 14. The HONIPALS Algorithm**

1: Input: $\mathbf{X}$, $\mathbf{Y}$, $r$, $K$
2: for $i = 1, \ldots, r$ do
3: \hspace{1em} $\mathbf{S}_{(1)i} = \mathbf{X}_{(1)i}^T \mathbf{Y}_{(1)i}$
4: \hspace{1em} $\mathbf{W}_{(1)i} = $ first $K$ left singular vectors of $\mathbf{S}_{(1)i}$
5: \hspace{1em} $\mathbf{T}_{(1)i} = \mathbf{X}_{(1)i} \mathbf{W}_{(1)i}$
6: \hspace{1em} $\mathbf{G}_i = \mathbf{X}_i \times_1 \mathbf{T}_{(1)i}$
7: \hspace{1em} Perform HOSVD on $\mathbf{G}_i$ to find factor matrices $\mathbf{P}^{(n)}_i \in \mathbb{R}^{K_n \times J_n}, n = 1, 2, \ldots, M$
8: \hspace{1em} $\mathbf{D}_i = \mathbf{Y}_i \times_1 \mathbf{T}_{(1)i}$
9: \hspace{1em} Perform HOSVD on $\mathbf{D}_i$ to find factor matrices $\mathbf{C}^{(n)}_i \in \mathbb{R}^{K_n \times J_n}, n = 1, 2, \ldots, P$
10: \hspace{1em} $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{G}_i \times_1 \mathbf{T}_{(1)i}$
11: \hspace{1em} $\mathbf{Y}_{i+1} = \mathbf{Y}_i - \mathbf{D}_i \times_1 \mathbf{T}_{(1)i}$
12: \hspace{1em} Store $\mathbf{G}_i$, $\mathbf{D}_i$, $\mathbf{T}_i$ and $\mathbf{W}_i$
13: end for
Algorithm 15. The GHOPLS Algorithm

1: Input: \( \mathbf{X}, \mathbf{Y}, r, K_1, K_2, \ldots, K_M, L_1, L_2, \ldots, L_P \)
2: for \( i = 1, \ldots, r \) do
3: \( \mathbf{S}_i = \langle \mathbf{X}_i, \mathbf{Y}_i \rangle_{(1,1)} \)
4: Perform rank-\((K_1, K_2, \ldots, K_M, L_1, L_2, \ldots, L_P)\) HOOI decomposition to give \( \mathbf{S}_i = \mathbf{G}_i \times_1 \mathbf{T}_i, \mathbf{D}_i \)
5: \( \mathbf{G}_i = \mathbf{X}_i \times_1 \mathbf{T}_i, \mathbf{D}_i \)
6: \( \mathbf{X}_{i+1} = \mathbf{X}_i - \mathbf{G}_i \times_1 \mathbf{T}_i \)
7: \( \mathbf{Y}_{i+1} = \mathbf{Y}_i - \mathbf{D}_i \times_1 \mathbf{T}_i \)
8: Store \( \mathbf{G}_i, \mathbf{D}_i, \mathbf{T}_i \) and \( \mathbf{W}_i \)
9: end for

Remark 21. The solution obtained from the HONIPALS Algorithm 14 degenerates to the two-way NIPALS Algorithm 3 when the input and output data tensors \( \mathbf{X} \) and \( \mathbf{Y} \) are given by the matrices \( \mathbf{X} \in \mathbb{R}^{N \times M} \) and \( \mathbf{Y} \in \mathbb{R}^{N \times P} \) and \( K = 1 \).

Remark 22. A symmetric Mode-A type PLS decomposition is also possible for both the HONIPALS and GHOPLS algorithms. This is achieved by replacing the approximation of \( \mathbf{Y} \) in terms of the scores, \( \mathbf{T} \), used in the deflation step in (5.23) and instead using the scores derived from the output \( \mathbf{U} \) in (5.25).

Remark 23. The relations (5.21) are expressed in a general form of tensor regression and therefore are direct extensions of the two-way PLS relations (3.3) and (3.4). If desired, they can be expressed as a mode-1 product of the unfolded matrix \( \mathbf{T}_{(1)} \) and the respective core tensors. Furthermore, by Remark 22 both algorithms can be used for different types of PLS, similarly to the two-way algorithm. Therefore, the GHOPLS and HONIPALS are both generalisations of the original 2-way NIPALS algorithm harnessing the available tensor-based operations.

Calculation of the regularised tensor regression

The \( r \) approximations obtained on each iteration in (5.24) are summed up to yield the final PLS approximations for the HONIPALS and GHOPLS algorithms in the form

\[
\tilde{\mathbf{X}} \approx \sum_{i=1}^{r} \mathbf{G}_i \times_1 \mathbf{T}_i, \quad \tilde{\mathbf{Y}} \approx \sum_{i=1}^{r} \mathbf{D}_i \times_1 \mathbf{T}_i, \quad (5.26)
\]

where \( \mathbf{G} \) and \( \mathbf{D} \) result from the tensor regression in (5.22). This latent variable decomposition can be used to produce a regularised tensor regression solution to predict \( \tilde{\mathbf{Y}} \) from \( \tilde{\mathbf{X}} \) through a regression coefficient tensor \( \mathbf{B} \). To this end, their mode-1 unfolded versions are given as

\[
\tilde{\mathbf{X}}(1) = [T_{(1),1}, T_{(1),2}, \ldots, T_{(1),r}]^T \mathbf{G}(1), \mathbf{G}(1), \mathbf{G}(1), \ldots, \mathbf{G}(1), r]^T,
\]

\[
\tilde{\mathbf{Y}}(1) = [T_{(1),1}, T_{(1),2}, \ldots, T_{(1),r}]^T \mathbf{D}(1), \mathbf{D}(1), \mathbf{D}(1), \ldots, \mathbf{D}(1), r]^T.
\]
where $T_{(1),i}$ denotes the unfolded version of the $i$-th score tensor $T$. Based on this, we can now obtain the prediction of $\tilde{Y}_{(1)}$ from $\tilde{X}_{(1)}$ as

$$\tilde{Y}_{(1)} = \tilde{X}_{(1)} [G_{(1),1}, G_{(1),2}, \ldots, G_{(1),r}]^T [D_{(1),1}, D_{(1),2}, \ldots, D_{(1),r}]^T,$$

which yields the unfolded regression coefficient tensor, $B$, in the form

$$B_{(M)} = [G_{(1),1}, G_{(1),2}, \ldots, G_{(1),r}]^T [D_{(1),1}, D_{(1),2}, \ldots, D_{(1),r}]^T.$$

This is analogous to the two-way NIPALS result in (3.9) and the HOPLS regression coefficients in (5.12).

### 5.4 Comparison of the quaternion and Tensor PLS algorithms

A thorough summary of the proposed multidimensional quaternion and tensor PLS algorithms is now provided. This analysis highlights the equivalence of processing 4D data types and the differences between the proposed solutions.

#### 5.4.1 Core components of the multidimensional NIPALS PLS algorithms

It has been shown that the three algorithms WL-QPLS, GHOPLS and HONIPALS are generalisations of the two-way NIPALS algorithm for performing PLS-regression on higher order multidimensional data. The choices of the core components, outlined in Section 3.3.4 with respect to the NIPALS PLS, are summarised in Table 5.2.
### Table 5.2. Comparison of the multidimensional PLS algorithms

<table>
<thead>
<tr>
<th>Algorithm Step</th>
<th>WL-QPLS</th>
<th>GHOPALS</th>
<th>HONIPALS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-covariance structure</td>
<td>$S_{i,Re} = X_i^T Y_i,Re$ Transformed to the real-domain</td>
<td>$S_i = \langle X_i, Y_i \rangle_{{1,1}}$ Tensor-based contraction product</td>
<td>$S_i = X_{(1),i}^T Y_{(1),i}$ Unfolded to matrices</td>
</tr>
<tr>
<td>Eigen-decomposition</td>
<td>$w_i,Re = \text{Eig}<em>{\text{max}}{S</em>{i,Re}S_{i,Re}^T}$ Left singular vectors of $S_{i,Re}$</td>
<td>$\text{HOOI/HOSVD decomposition of } S_i$</td>
<td>$W_{(1),i} = \text{Eig}_{K,\text{max}}{S_i S_i^T}$ First K left singular vectors of $S_i$</td>
</tr>
<tr>
<td>Score calculation</td>
<td>$t_i = X_i w_i$ The projection of $X_i$</td>
<td>$T_{(1)} = X_{(1)} (P^{(M)} \otimes P^{(M-1)} \otimes \cdots \otimes P^{(1)})^T$ The projection of unfolded $X_i$</td>
<td>$T_{(1),i} = X_{(1),i} W_i$ The projection of unfolded $X_i$</td>
</tr>
<tr>
<td>Deflation</td>
<td>$X_{i+1} = X_i - t_i P_i^H$ $Y_{i+1} = Y_i - t_i c_i^H$</td>
<td>$X_{i+1} = X_i - G_i \times_1 T_{(1),i}$ $Y_{i+1} = Y_i - D_i \times_1 T_{(1),i}$</td>
<td>$X_{i+1} = X_i - G_i \times_1 T_{(1),i}$ $Y_{i+1} = Y_i - D_i \times_1 T_{(1),i}$</td>
</tr>
</tbody>
</table>
5.4. The isomorphism between quaternions and tensors

Consider a matrix of quaternion variables, \( \mathbf{X} \), which can also be represented as a tensor, \( \mathbf{X} \in \mathbb{R}^{N \times 4 \times m} \), where each mode-2 slice is constructed as

\[
\mathbf{X}_{(:,1,:)} = \mathbf{X}_r, \quad \mathbf{X}_{(:,2,:)} = \mathbf{X}_i,
\]
\[
\mathbf{X}_{(:,3,:)} = \mathbf{X}_j, \quad \mathbf{X}_{(:,4,:)} = \mathbf{X}_k.
\]

Hence, the frontal slices, \( \mathbf{X}_{(:,m,:)} \in \mathbb{R}^{N \times 4 \times m} \), are matrices that contain the four components of a quaternion vector. This can be regarded as an extension of the isomorphism in (2.26), \( \Gamma_m \), which yields a quaternion to tensor transform. To this end, we shall re-write (2.27), \( \mathbf{X}_{\Re} = \frac{1}{4} \mathbf{X} \Gamma^*_m \), as

\[
\mathbf{X}_{(1)} = \mathbf{X}_{\Re} \Phi,
\]

where \( \mathbf{X}_{(1)} \in \mathbb{R}^{N \times 4m} \) indicates the mode-1 unfolding of the tensor representation of the augmented quaternion-valued matrix \( \mathbf{X} \). The matrix \( \Phi \) is a permutation matrix which re-orders the columns of the matrix, \( \mathbf{X}_{\Re} \), to group each variable’s four quaternion vector components, \( \mathbf{x}_r, \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \), into consecutive columns, that is

\[
\mathbf{X}_{(1)} = \mathbf{X}_{\Re} \Phi.
\]

Through the right multiplication of (5.30) by an identity matrix \( \mathbf{I}_{(1)} \in \mathbb{R}^{4m \times 4m} \), which is also considered to be in a mode-1 unfolded state we obtain the relation

\[
\mathbf{X}_{(1)} = (\mathbf{X} \Gamma^* \Phi) \mathbf{I}_{(1)} \iff \mathbf{X} = \mathcal{I} \times_1 (X \Gamma^* \Phi),
\]

where \( \mathcal{I} \in \mathbb{R}^{4m \times 4m} \) is constructed from the \( 4m \times 4m \) dimensional identity matrix folded such that each mode-3 slice has a \( 4 \times 4 \) identity matrix shifted down by 4 from the previous slice. The full quaternion to tensor transform is then given by

\[
\mathbf{X} = \mathcal{A} \times_1 \mathbf{X},
\]

where

\[
\mathcal{A} = \mathcal{I} \times_1 \Gamma^* \Phi \in \mathbb{H}^{4m \times 4m},
\]

is the complete transform.

The mapping back into the quaternion domain is then performed as

\[
\mathbf{X} = \mathbf{X}_{(1)} \mathbf{A}_{(1)}^H.
\]

5.4.3 WL-QPLS as a tensor-variate PLS

The link between a quaternion-valued and tensor-based PLS solution is described through the quaternion to tensor transform in (5.34). Consider the case where it is desired to calculate a PLS-regression between the quaternion-valued input data matrix, \( \mathbf{X} \in \mathbb{H}^{N \times m} \), and the quaternion-valued output data matrix, \( \mathbf{Y} \in \mathbb{H}^{N \times p} \). Using the transform \( \mathcal{A} \) constructed
in Section 5.4.2 we can express the matrices, \( \mathbf{X} \) and \( \mathbf{Y} \), as third order tensors

\[
\mathbf{X} = \mathbf{A}_m \times_1 \mathbf{X}, \quad \mathbf{Y} = \mathbf{A}_p \times_1 \mathbf{Y},
\]

(5.36)

where \( \mathbf{X} \in \mathbb{R}^{N \times 4 \times m} \) and \( \mathbf{Y} \in \mathbb{R}^{N \times 4 \times p} \). These data tensors can be input into the tensor-valued HOPLS or the proposed GHOPLS and HONIPALS algorithms to compute an alternative PLS solution.

In each iteration the WL-QPLS, in Algorithm 11, computes a widely linear regression between a vector score, \( \mathbf{t}_i \), and the input and output blocks \( \mathbf{X}_i \) and \( \mathbf{Y}_i \). The equivalent regression (with the same degrees of freedom) in the real-domain requires the quaternion vector, \( \mathbf{t} \), to be represented as an \( N \times 4 \) matrix, \( \mathbf{t}_{Re} = [t_r, t_i, t_j, t_k] \), produced by the transform (2.27). Therefore, the score of an equivalent tensor PLS solution would also be an \( N \times 4 \) matrix. This score can be computed by the HONIPALS and GHOPLS in Algorithm 14 (with \( M = 4 \)) and Algorithm 15 (with \( K_1 = L_1 = 4 \) and \( K_2 = L_2 = 1 \)) respectively, however, the HOPLS in Algorithm 13 cannot produce an alternative solution as it only calculates a vector score component. Therefore, for the same data represented as quaternion or tensor-variate, which are related through the transform (5.36), the three algorithms, WL-QPLS, GHOPLS and HONIPALS each have enough degrees of freedom to provide a comparable regression result.

**Simulation: the quaternion WL-QPLS and tensor variate PLS**

A simulation on synthetic data is now presented to further examine the WL-QPLS, HONIPALS and GHOPLS algorithms. An improper quaternion-valued data matrix, \( \mathbf{X} \in \mathbb{H}^{N \times m} \) (\( N = 1000 \) and \( m = 10 \)), was generated from a mixture of \( r = 5 \) independent improper sources. Each quaternion axis \( q_r, q_i, q_j, q_k \) was an i.i.d. source, sampled from the distribution as \( q_n \sim \mathcal{N}(0, 1) \) for \( n = r, i, j, k \). The data matrix \( \mathbf{Y} \in \mathbb{H}^{N \times p} \) was then calculated as \( \mathbf{Y} = \mathbf{X} \mathbf{B} \) where \( \mathbf{X} \in \mathbb{H}^{N \times 4m} \) is in the augmented form and \( \mathbf{B} \in \mathbb{H}^{4m \times p} \) are the quaternion-valued regression coefficients, where \( p = 10 \). A noise matrix \( \mathbf{N}_Y \in \mathbb{H}^{N \times p} \) was then created where each quaternion axis was sampled from the distribution \( q_n \sim \mathcal{N}(0, \sigma^2) \) for \( n = r, i, j, k \) and \( \sigma^2 \) was varied to give a range of SNRs calculated as

\[
SNR = 10 \log_{10} \frac{\text{Tr}\{E[\mathbf{Y}^H \mathbf{Y}]\}}{\text{Tr}\{E[\mathbf{N}_Y^H \mathbf{N}_Y]\}}.
\]

The matrices \( \mathbf{X} \) and \( \mathbf{Y} \) were transformed to tensors using the quaternion to tensor transform in (5.34) to give the data tensors \( \mathbf{X} \in \mathbb{R}^{N \times 4 \times m} \) and \( \mathbf{Y} \in \mathbb{R}^{N \times 4 \times p} \). The widely linear and tensor regression estimates were calculated using the WL-QPLS, GHOPLS and HONIPALS algorithms for the same data and for a range of components. The tensor GHOPLS and HONIPALS, outlined respectively in Algorithm 15 and Algorithm 14, were implemented to calculate rank-(\( N, 4, 1 \)) scores. The results were transformed back into the quaternion domain to calculate the error as

\[
MSE = \frac{|\mathbf{Y} - \hat{\mathbf{Y}}|^2}{\text{Tr}\{E[\mathbf{Y}^H \mathbf{Y}]\}},
\]

(5.37)
where $\hat{Y}$ are the estimated dependent variables. The results are shown in Figure 5.1 for three different SNRs and include the results for training data and the ensemble average of the estimate on test data, for an ensemble with 50 members. The right hand plots compare the MSE (on the training data) for each algorithm for the range of SNR. For more than $r = 5$ PLS components, the error for the test data degraded for all the algorithms as they were over-fitting. For a general application, the structure of the data within this representation must be considered in order to indicate which is the most appropriate algorithm to use.

![Figure 5.1](image)

Figure 5.1. The MSEs of widely linear and tensor regression estimates calculated for a range of SNRs and different numbers of PLS components, as a ratio of the power in the dependent variables.

5.4.4 Comparison of the WL-QPLS and HONIPALS cross-covariance structures

The quaternion-valued data matrices $X \in \mathbb{H}^{N \times m}$ and $Y \in \mathbb{H}^{N \times p}$ can be transformed, through (5.36), to the data tensors, $X \in \mathbb{R}^{N \times 4 \times m}$ and $Y \in \mathbb{R}^{N \times 4 \times m}$, which, as stated in Section 5.4.3, allow alternative PLS solutions using the WL-QPLS, HONIPALS and GHOPLS algorithms. The data inputs are related as

$$X_{(1)} = X_{Re} \Phi_X, \quad Y_{(1)} = Y_{Re} \Phi_Y,$$

where $X_{(1)}$ and $Y_{(1)}$ are the mode-1 unfolded data tensors, $X_{Re}$ and $Y_{Re}$ denote the transformed real-valued data matrices through (2.27) and $\Phi_X$ and $\Phi_Y$ are the permutation matrix of the required dimensions needed for the quaternion to tensor transform (5.34).
Observe that the WL-QPLS algorithm in (5.2) calculates the cross-covariance structure in the real-domain, given by the matrix \( X_{Re}^T Y_{Re} \), before transforming back to the quaternion-domain. On the other hand, the HONIPALS algorithm creates the cross-covariance structure from the mode-1 unfolded data tensors, as \( S = X_{(1)}^T Y_{(1)} \). The cross-covariance structures are therefore related as

\[
S_{(1)} = \Phi_X^T S_{Re} \Phi_Y,
\]

while the corresponding SVDs are related as

\[
U \Sigma V^T = \Phi_X^T U_{Re} \Sigma V_{Re}^T \Phi_Y.
\]

The leading four left singular vectors are chosen for the basis of the HONIPALS score vectors \( T_{(1)} \) in (5.18) when used to provide an alternative to the quaternion-valued solution. This basis for the HONIPALS algorithm is therefore the same as that calculated in the real-domain for the WL-QPLS, as they are related by

\[
T_{(1)} = X_{Re} \Phi_X (\Phi_X^T U_{Re})[: , 1:4].
\]

However, to calculate the quaternion-valued score vector, \( t \), only the first left singular vector, \( U_{Re}[:,1] \), is transformed back into the quaternion domain, as \( w \) in (5.2). Moreover, the quaternion algebra is used to compute the score, \( t = Xw \), and a multiplication in the quaternion algebra computed in the real-domain requires the constrained matrix multiplication (2.29).

**Remark 24.** The cross-covariance structures for the proposed HONIPALS and WL-QPLS algorithms are linked through a permutation matrix which means that they share the same eigendecomposition computation. The HONIPALS uses the four leading singular vectors to calculate the scores from the unfolded data matrix \( X_{(1)} \). On the other hand, the WL-QPLS uses only the first singular vector of the real-valued cross-covariance structure which is folded back to the quaternion-domain prior to the calculation of the score vector. This operation performed equivalently in the real-domain requires a constrained matrix multiplication. As a result, the score and the subspace identified in every iteration of the WL-QPLS and HONIPALS algorithms are not the same.

### 5.4.5 Comparison of HONIPALS and GHOPLS cross-covariance structures

The difference between the HONIPALS and GHOPLS algorithms is the cross-covariance structure implemented and its subsequent deflation. In this section, we will compare and contrast these core-component choices to elucidate the nature of each method.

Consider the tensors \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_M} \) and \( \mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_n \times \cdots \times J_P} \) with \( I_1 = J_1 \). The mode-1 contraction product used by the GHOPLS algorithm is given as \( S = \langle \mathcal{X}, \mathcal{Y} \rangle_{(1,1)} \in \mathbb{R}^{I_2 \times I_3 \times \cdots \times I_M \times J_2 \times \cdots \times J_P} \). On the other hand, The HONIPALS uses the product of the mode-1 unfolded data tensors as \( S = X_{(1)}^T Y_{(1)} \in \mathbb{R}^{I_2 I_3 \cdots I_M \times J_2 J_3 \cdots J_P} \). These share
the same elements as
\[
\sum_{i_n=j_n=1}^{I_1} x_{i_1,i_2,\ldots,i_M} y_{j_1,j_2,\ldots,j_P} = \mathbf{X}[; i_2 j_3 \ldots i_M]^\top \mathbf{Y}(1)[; j_2 j_3 \ldots j_P].
\]
from the definition of the contraction product (2.34). The link between them is provided through the canonical matrix unfolding (2.36), whereby the mode-(M - 1) canonical unfolding of the tensor \(\mathbf{S}\) yields the matrix \(\mathbf{S}_{(M-1)} \in \mathbb{R}^{I_2 J_3 \cdots J_M \times J_2 J_3 \cdots J_P}\) which is the same as the matrix \(\mathbf{S} = \mathbf{X}(1)^\top \mathbf{Y}(1)\).

The two cross-covariance structures contain the same elements and, intuitively, share the same information. However, the Tucker decomposition of the contraction product form will yield a different result to the SVD of the unfolded cross-covariance matrix, that is, the matrices, \(\mathbf{W}(1)\), derived from each decomposition select a different unfolded score matrix as \(\mathbf{T}(1) = \mathbf{X}(1)^\top \mathbf{W}(1)\).

**Proposition 6.** Consider the tensors \(\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_M}\) and \(\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_n \times \cdots \times J_P}\) with \(I_1 = J_1\). The mode-(M - 1) canonical unfolding of the Tucker decomposition of the tensor \(\mathbf{S} = \langle \mathbf{X}, \mathbf{Y} \rangle_{(1,1)}\) (obtained from the mode-1 contraction product) does not yield the SVD of the matrix \(\mathbf{S}_{(M-1)}\), equivalently obtained as \(\mathbf{S} = \mathbf{X}(1)^\top \mathbf{Y}(1) = \mathbf{W} \mathbf{\Sigma} \mathbf{V}^\top\).

**Proof.** The tensor given by \(\mathbf{S} = \langle \mathbf{X}, \mathbf{Y} \rangle_{(1,1)}\) admits a Tucker decomposition
\[
\mathbf{S} = \mathbf{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \cdots \times_{M+P-2} \mathbf{A}^{(M+P-2)},
\]
where \(\mathbf{G} \in \mathbb{R}^{K_1 \times K_2 \times \cdots \times K_{M+P-2}}\) and \(\mathbf{A}^{(i)} \in \mathbb{R}^{K_i \times K_i}\) with \(K_i = I_{i+1}\) for \(i < M\) and \(K_i = J_{i+1}\) for \(i \geq M\). The Tucker decomposition of the tensor \(\mathbf{S}\) is computed through the HOOI or HOSVD algorithms so that the factor matrices \(\mathbf{A}^{(i)}\), and hence their Kronecker products, are orthogonal. The mode-(M - 1) canonical unfolding is then given as [54]
\[
\mathbf{S}_{(M-1)} = (\mathbf{A}^{(M-1)} \otimes \mathbf{A}^{(M-2)} \otimes \cdots \otimes \mathbf{A}^{(1)}) \mathbf{G}_{(M-1)} (\mathbf{A}^{(M+P-2)} \otimes \cdots \otimes \mathbf{A}^{(M)})^\top.
\]
This yields the left and right multiplication of the unfolded core tensor \(\mathbf{G}_{(M-1)}\) by two unitary matrices. However, the matrix \(\mathbf{G}_{(M-1)}\) is in general not diagonal which means that the result is not generally equivalent to the SVD of the matrix \(\mathbf{S}_{(M-1)} = \mathbf{X}(1)^\top \mathbf{Y}(1) = \mathbf{W} \mathbf{\Sigma} \mathbf{V}^\top\), where the matrix \(\mathbf{\Sigma}\) is diagonal.

**Remark 25.** The cross-covariance structures for the HONIPALS, \(\mathbf{S}_{(1)} = \mathbf{X}(1)^\top \mathbf{Y}(1)\), and GHOPLS, \(\mathbf{S} = \langle \mathbf{X}, \mathbf{Y} \rangle_{(1,1)}\), algorithms have the same entries, linked by a canonical mode-1 unfolding, and thus, they share the same information. However, according to Proposition 6, the Tucker decomposition in (5.15) employed within the GHOPLS algorithm is in general not equivalent to the SVD in (5.17) used for HONIPALS. As a result, the unfolded score matrices in both algorithms are obtained as \(\mathbf{T}(1) = \mathbf{X}(1)^\top \mathbf{W}(1)\), where \(\mathbf{W}(1)\), given by the leading left singular vectors of \(\mathbf{S}\) or the mode-1 unfolded Tucker factor matrices as in (5.19), will not be the same.
5.5 Application of multidimensional PLS algorithms for image classification

The works of Risojević and Babić [104] [105] and Le Bihan and Sangwine [100] amongst others demonstrate the advantages of using quaternion-valued signal processing for image data. Encoding the RGB values as a pure quaternion

\[ Q(x, y) = iQ_r(x, y) + jQ_g(x, y) + kQ_b(x, y), \]

means that the original geometric information in images is preserved. This resolves the problem experienced with standard approaches which use vectorised images, whereby the relations between pixel colours are lost. Furthermore, it has been shown that quaternion-valued filters maintain the physical meaning in image representations [101] [106]. The tensor representation can be similarly used to provide a geometry preserving representation.

Risojević and Babić [104] have developed a quaternion-valued image classification method. The first step is to convert an image into overlapping \( p \times p \) pixel patches. These patches are then filtered with a set of quaternion-valued filters learned through QPCA [100]. The \( N \) patches are vectorised and stacked into a matrix \( X \in \mathbb{H}^{N \times p^2} \). The covariance matrix \( X^H X \) is then diagonalised through the QPCA filter \( U \), obtained as \( X^H X = U \Sigma U^H \). Here, we show that the WL-QPLS, GHOPLS and HONIPALS algorithms can be used in place of the QPCA to obtain a quadrivariate filter. The image classification process in [104] is summarised as:

1. Convert the image into an array of patches,
2. Filter each patch,
3. Obtain a sparse representation through dictionary learning,
4. Convert the image into a real-valued feature vector,
5. Classify using an SVM.

The original method was applied to the Brazilian Coffee Scenes dataset [107], obtained using a satellite to take images in the near infrared (NIR), red and green spectrum. Each image is categorised into either an image of a coffee plantation (over 80% of pixels containing coffee plants) or a non-coffee area. Figure 5.2 shows a series of examples for each class. It is noted that this dataset exhibits high intraclass variation [107]. Using the colour information within the processing enables new possibilities to find effective discriminative features for such data. The image is encoded into a quaternion representation as

\[ Q(x, y) = iQ_{NIR}(x, y) + jQ_r(x, y) + kQ_g(x, y). \]

In our experiment, these images were converted into a set of, 5 \( \times \) 5 pixel patches (overlapping by 4 pixels) vectorised as \( s \in \mathbb{H}^{1 \times 25} \) which were then filtered using the PLS basis obtained from the WL-QPLS, GHOPLS or HONIPALS algorithms. A random selection
5.5. Application of multidimensional PLS algorithms for image classification

Table 5.3. Coffee image classification

<table>
<thead>
<tr>
<th>Filter</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>QPCA</td>
<td>88.6%</td>
</tr>
<tr>
<td><strong>WL-QPLS</strong></td>
<td><strong>89.7%</strong></td>
</tr>
<tr>
<td>HONIPALS</td>
<td>88.5%</td>
</tr>
<tr>
<td>GHOPLS</td>
<td>88.8%</td>
</tr>
</tbody>
</table>

of 10000 patches was taken from the dataset and used to calculate the filters through an implementation of the algorithm where the input and output were the same data structure, as required for the covariance matrix diagonalisation application in Section 5.2.1. For the WL-QPLS implementation the filter was the matrix $\mathbf{W}$ of the concatenated score projections in (5.2). For the tensor-based algorithms, the quaternion-valued patch was transformed to a tensor representation through (5.34) and the filter was obtained from the GHOPLS and HONIPALS basis in (5.18). In each case the number of components selected was $r = 15$. The filtered patches were then represented in the quaternion domain as

$$\mathbf{s} = \mathbf{s}^{(0)} + \mathbf{s}^{(1)}_i + \mathbf{s}^{(2)}_j + \mathbf{s}^{(3)}_k.$$  

The dictionary learning step employed in [104] was neglected in order to provide a direct comparison between the algorithms considered. The real-valued feature vector, $\mathbf{f}^{(l)} = [f_q^{(l)}, f_{q+r}^{(l)}, f_{q+2r}^{(l)}]$, $q = 1, 2, \ldots, 15$ was then obtained from

$$f_q^{(l)} = |\mathbf{s}_q^{(l)}|,$$

$$f_{q+r}^{(l)} = \max(0, \mathbf{s}_q^{(l)} - \theta),$$

$$f_{q+2r}^{(l)} = \max(0, -\mathbf{s}_q^{(l)} - \theta),$$

where $\theta = 1 \times 10^{-10}$, $l = 0, 1, 2, 3$, for each quaternion axis and $r$ is the length of the vectorised patch. These are then pooled for each patch as

$$\mathbf{F}^{(l)} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{f}_i^{(l)},$$

where $N$ was the number of patches within an image. Each element was power-law transformed as $\mathbf{F} = \mathbf{F}^\alpha$, where $\alpha = 0.5$ and then normalised through

$$\overline{\mathbf{F}}^{(l)} = \frac{\mathbf{F}^{(l)}}{||\mathbf{F}^{(l)}||_2},$$

before being concatenated into one feature vector $\mathbf{F} = [\overline{\mathbf{F}}^{(0)}, \overline{\mathbf{F}}^{(1)}, \overline{\mathbf{F}}^{(2)}, \overline{\mathbf{F}}^{(3)}]$ which was fed into an SVM implemented in Matlab. The 5-fold cross validated results are presented in Table 5.3, demonstrating the validity of the proposed quaternion and tensor-valued representations.
Chapter 5. Quaternion and Tensor Partial Least Squares

Coffee Non-coffee

Figure 5.2. A selection of the satellite images taken of Brazilian Coffee scenes dataset and classified as either containing a coffee plantation or a non-coffee zone. The intraclass variance is known to be high for this dataset as the images cover different regions and so terrains and crop types. The multidimensional filters used in this application offer an alternative approach to segment such information.

5.6 Data properties and the MD-PLS algorithms performance

In Section 3.7 the assumptions on the data for the application of the linear regression model $Y = XB$ were outlined and their impact on the PLS result was discussed. Moreover, in Section 4.6 it was shown how these assumptions and qualities are extended for the WL-CPLS algorithm. In the same way they are extended here for the WL-QPLS, GHQPLS and HONIPALS class of multidimensional PLS algorithms.

Quaternion widely linear regression imposes the quaternion extensions of the classical linear regression assumptions on the data. As such, the errors must be independent and constant in terms of quaternion full second-order statistics. The dependent variables are now also a linear combination of the independent variables and their $i$-, $j$- and $k$-th conjugate. On the other hand, the tensor regression is based on the linear regression assumptions applied to the unfolded representation in (2.40). Note that in each case the prediction must be linear in the multidimensional framework (formalised through widely linear and tensor regression) and so the data must conform to this and their properties/structures cannot change over the realisations.

The synthetic comparison in Section 5.4.3 aims to show the performance of the algorithms without the data properties impacting the result. The extra complexity in these multidimensional PLS algorithms is aimed at exploiting the redundancy in data while preserving geometry in its structure. The quaternion representation is frequently used to represent rotations and coordinates, whereas the tensor representation has found uses in time series from tensorisation/folding techniques. Appropriate applications of the WL-QPLS, HONIPALS and GHQPLS algorithms will be based on data for which the multidimensional relationship exhibits a certain property and has high redundancy. For example, the image classification problem in Section 5.5 highlights a successful real-world application of the proposed algorithms. It is shown that the colour information within the image can be encoded as part of the data structure and that the data dimensionality can be reduced using these techniques. This added feature of the processing enables good classification performance of a difficult dataset and so an appropriate application would aim to make
use of such flexibility.

5.7 Conclusion

This Chapter has introduced an algorithm for quaternion-valued PLS, termed WL-QPLS, and two algorithms for tensor-based PLS, termed HONIPALS and GHOPLS. The WL-QPLS, proposed in Section 5.2, produces a regularised widely linear regression for noncircular quaternion-valued data. It therefore takes into account the full second-order augmented statistics required, as described in Section 2.3.2. Quaternion-valued data can be equivalently processed as a third order tensor, shown through a derived transform in Section 5.34. This has been shown to provide a useful tool to examine the links and differences between processing the data as a quaternion or a tensor. The tensor-based algorithms, HONIPALS and GHOPLS derived in Section 5.3, utilise the full range of tensor operations. This is based on the choice of cross-covariance data-structure and its eigen-type decomposition. The HONIPALS algorithm uses an unfolding methodology similar to that employed by the HOSVD, whereas the GHOPLS keeps a tensor representation throughout so the multiway relations are explicit, similar to the HOPLS algorithm. A comparison of these methods shows that they provide alternative computations of the same information, differing mainly in the decompositions structure. Both, however, provide a general PLS solution which admits any desired multilinear-rank Tucker decomposition of the data. Furthermore, the contrast between the WL-QPLS and tensor-based solutions is examined. This shows that the calculation of the cross-covariance decomposition is the same for the HONIPALS and WL-QPLS but the transform step (5.2) back to the quaternion-domain yields an alternative calculation of the rest of the iteration and so a different solution. Finally, an application of the algorithms for structured component identification and as a pre-processing step in image classification from 3D NIR satellite images has demonstrated that multidimensional methods allow the preservation of the data geometry during processing.

This thesis has provided a study of the PLS method for multidimensional data structures. In doing so algorithms for complex-, quaternion- and tensor-valued regularised regression have been established. The links between the data structures have highlighted the equivalent processing power of the methods (for up to 4D data) while showing the differences in each algorithms computation, corroborated by synthetic simulations. The methods have been applied to practical applications in order to show how each method can provide physically meaningful representation in its computation. As such, the concept of latent variables has been examined, in a rigorous component analysis framework for regression algorithms, to extend current techniques for a more general data-structure considerations. The PLS score is specifically the latent variable obtained and has been examined in the same sense as they were introduced in Chapter 3, whereby the latent variables were the components within an entity subspace of a data matrix, with consideration of higher order data structures. The developments presented in this thesis, therefore, allow the exploitation of the natural structure of data, following the modern philosophy of data applications that the data should dictate the processing.
Part II

Part B
Chapter 6

Self-Noise determination of a microseismometer for planetary science

6.1 Introduction

The main science goals of the InSight mission are to resolve the internal structure (the core, mantle and crust) of Mars [2] [3]. To do so, the SEIS experiment has defined a set of sensitivity requirements for its sensors that would allow the necessary seismology to be performed. The sensitivity is defined, in the first instance, as the self-noise of the sensor. To this end, this Chapter will address the self-noise determination of the microseismometer implemented as the short-period sensor of this mission, currently en-route to Mars. The goal is to describe, analyse and develop the processes behind the self-noise determination in order to provide a rigorous understanding of what can be established from experimental data. To summarise, the chapter has three main themes:

1. The self-noise estimation methods, their analysis and the impact of practical setups,
2. Algorithmic developments,
3. The definitive self-noise determination of the NASA InSight microseismometer.

The techniques and methodologies studied in this thesis are motivated by the modern trend of data-centric applications, due to the recent developments in both sensor and computing technology. Seismology as a discipline has benefited greatly from such developments. For example, numerical optimisation is frequently employed for solving the complex inverse problems posed by the field [6]. Furthermore, the advent of closed-loop, feedback seismometers [108] has significantly improved instrument performance. This, in
Chapter 6. Self-Noise determination of a microseismometer for planetary science

The concept of latent variables can therefore be useful for the understanding of several key aspects of the self-noise determination process.

Remark 26. A seismometer records the ground motion. The signal will also contain components from environmental sources and the intrinsic sensor noise itself. Each component is hidden within the signal and therefore, the self-noise of the seismometer can be considered as a latent variable.

The instrument self-noise is a latent variable within the recorded signal (Remark 26). The PCA/PLS methods decompose a matrix of data based on mathematical properties such as orthogonality. Here, the self-noise is first estimated based on the property of coherence with another seismometer [109]. In addition, signal injections can be made from aseismic sources which require a physical model and further measurements to isolate them [7]. This chapter, though, focuses purely on identifying the microseismometer self-noise component.

To enable this extraction of the self-noise latent variable, a summary of the instrument will first be given followed by the background and analysis of the numerical methods used to estimate a seismometer’s self-noise from experimental data. Several algorithmic developments for the self-noise process are then provided. Finally, the sensor is characterised over a wide bandwidth, in terms of transfer-function and self-noise, based on several datasets from flight and qualification units.

6.2 The microseismometer outputs and response

The microseismometer is a MEMS device manufactured from single-crystal silicon [22] [23]. The displacement of the proof-mass die is sensed through measuring the change in capacitance between fixed transducer electrodes and electrodes on the die itself. For further technical details of the sensor see the SEIS paper [2].

Figure 6.1 shows a system diagram of the microseismometer. The sensor operates in a feedback loop where the main and integral coils are driven to oppose the force on the sensor mass, according to the force balance principle [22] [108]. The corresponding signals which drive these coils, velocity (VEL) and mass position (POS), are recorded as the outputs of the sensor. The velocity channel provides the main science data channel for the mission and its transfer-function is modelled as a two pole-zero pair high-pass filter with a corner frequency at \( f_{SP} \approx 1/35 \) Hz. The choice of this transfer-function for the velocity output is made so that it can be digitised over a smaller range and the analogue to digital converter (ADC) will require fewer bits. This means that a high gain can be used to give a good resolution over the main bandwidth of interest. On the other hand, the mass position channel is directly proportional to acceleration at low frequencies. The transfer-function for acceleration is well modelled by a two pole-zero pair low pass filter with a corner frequency at \( f_{SP} \approx 1/35 \) Hz. Figure 6.2 shows a comparison of the respective transfer-functions of the velocity and mass position channels for acceleration. Prior to analysis, such as self-noise determination, these transfer-functions must be removed.
6.2. The microseismometer outputs and response

Figure 6.1. The schematic shows (in cross-section) the MEMS sensor, connectivity to the functional blocks of the electronics and the packaging of the sensor in its enclosure. $RT_{enc}$ and $RT_{die}$ indicate the locations of the resistance thermometers.

Figure 6.2. The top two plots show the frequency response of the velocity channel ($V/m/s$) and mass position channel ($V/m/s^2$). The bottom plot shows both the responses to acceleration ($V/m/s^2$).
6.2.1 Transfer-function correction

The frequency response of a seismometer’s outputs and the corresponding correction is of interest in order to obtain a good scientific interpretation, especially for long period signals [110]. The general problem is described as

\[ a(t) = h(t) * b(t), \]
\[ A(s) = H(s)B(s), \]

where \(*\) indicates convolution, the seismic signal is denoted as lowercase \(b(t)\) (Fourier transform \(B(s)\), where \(s\) denotes frequency), the output from the seismometer is given as \(a(t)\) (Fourier transform \(A(s)\)) and \(h(t)\) (Fourier transform \(H(s)\)) is the transfer-function of the seismometer. The goal is to remove the transfer-function, \(h(t)\), from the measured signal, \(a(t)\), to make available the seismic signal, \(B(t)\), for analysis. As a result, the transfer-function \(h(t)\) or \(H(s)\) must be known. For the microseismometer, as described in Section 6.2, the velocity channel transfer-function is modelled in the frequency-domain as the two pole-zero pair high-pass filter

\[ H(s) = \frac{|H|(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)}, \]

where \(|H|\) is the gain, \(z_1, z_2 = 0\) are the system zeros and the system poles are given by

\[ p_1 = f(-D - i(1 - D^2)^{\frac{1}{2}}), \quad p_2 = f(-D + i(1 - D^2)^{\frac{1}{2}}), \]

where \(f\) is the corner-frequency of the transfer-function and \(D\) is the damping constant.

The impact of the transfer-function is removed from recorded data through a division by the estimated response in the frequency-domain, that is, the Fourier transform of the velocity channel, \(A(s) = H(s)B(s)\), is divided by an estimated transfer-function of the form (6.2), denoted as \(\tilde{H}(s)\), to give \(\hat{B}(s) = \frac{H(s)}{\tilde{H}(s)}B(s)\). The ratio of the transfer-function to the estimated transfer-function is given by \(R(s)\). The algorithm for transfer-function correction is discussed in detail in Section 6.7.2 and Chapter 7.

6.3 Seismometer Self-Noise Determination

The self-noise determination of a seismic sensor is achieved through coincidence testing [21] [111] [109]. This is when a “test” seismometer is collocated with a “reference” seismometer to record the same ambient signal. The time series are then compared in order to produce an analysis of the test seismometer. This topology is described in Figure 6.3 where the two instruments are assumed to have recorded the same ambient signal, \(Z\), with their independent noise processes, \(N_1\) and \(N_2\), added to the signal and filtered through their respective transfer functions, \(H_X\) and \(H_Y\). The goal is then to correct the transfer-functions, remove the \(Z\) component and isolate the noise processes \(N_1\) and \(N_2\).

The noise of seismometer is typically presented in the frequency domain as a power spectral density (PSD) which provides a straightforward interpretation of the bandwidth.
in which the instrument is capable of examining. From the topology in Figure 6.3, the PSDs of each observable signal are given as

\[ P_{XX} = |H_X|^2(P_{ZZ} + P_{N1N1} + P_{ZN1} + P_{N1Z}), \]  
\[ P_{YY} = |H_Y|^2(P_{ZZ} + P_{N2N2} + P_{ZN2} + P_{N2Z}), \]

Moreover, the cross-PSD is given as

\[ P_{XY} = H_X H_Y^* (P_{ZZ} + P_{N1Z} + P_{ZN2} + P_{N1N2}), \]

Theoretically the cross-terms \( P_{N1Z}, P_{ZN2}, P_{N1N2} \) vanish as there is no statistical dependence between \( Z, N1 \) and \( N2 \). However, in a practical estimation there is some numerical residual from spurious correlations owing to finite data lengths. This is shown in Figure 6.4 where two signals \( X \) and \( Y \) were simulated as \( N \) i.i.d. samples from the distribution \( \mathcal{N}(0, 1) \). The CPSD \( P_{XY} \) and PSD \( P_{XX} \) was calculated for \( N = 10^2, 5 \times 10^2, 10^3, \ldots, 5 \times 10^5 \) and the ensemble (of 100 realisations) mean value of each spectrum is shown in Figure 6.4 along with the calculated \( P_{XY} \) and \( P_{XX} \) for \( N = 5 \times 10^5 \). This shows that, although decreasing with data length, the cross-terms spurious correlations cannot be neglected for the analysis involving the PSDs (6.4), (6.5) and (6.6).

The transfer-functions of the seismometers \( X \) and \( Y \) in Figure 6.3 are typically corrected prior to analysis. This is achieved by a division in frequency with an estimate of the transfer functions \( \tilde{H}_X \) and \( \tilde{H}_Y \). The ratio of the actual transfer function and the estimate is denoted \( R_X \) and \( R_Y \).

### 6.3.1 Seismometer testing literature review

As introduced in the prequel, Holcomb [109] [112] outlined the main concepts of seismometer noise testing. The experiment is based around the analysis of simultaneously acquired data from two colocated instruments. In the original method, it is assumed that the inco-
Chapter 6. Self-Noise determination of a microseismometer for planetary science

Figure 6.4. Simulation to show the residual correlation in a CPSD between two uncorrelated sources. Calculated with the Welch method averaging over two Hanning windows with 50% overlap, where the data are assumed to be sampled at $f_s = 100$ Hz.

Inherent part of the two recorded signals is the seismometer’s self-noise. This method was built on by Sleeman et. al. [113] who derived a coherence-based test for three colocated seismometers. The main advantage of this method is that it does not require a priori transfer-function correction, avoiding a source of error.

A summary review of the coherence-based methods is provided by Ringler et. al. [21] [111]. The review first compares the two and three-sensor methods and then describes the issues which lead to error injection from a practical setup/estimation. Such injections should be considered especially when comparing results across different tests. Guidelines have been presented for self-noise computation/processing (frequency estimation, time period selction and experimental setup) to provide consistency in such comparisons [21] [114]. It is also noted that time domain aspects (such as the operating range) should be considered [114].

The literature highlights that there are several key sources of excess terms in the self-noise estimate. The accuracy of the alignment between the seismometer axes is shown to be a major contributor to excess terms in the self-noise estimate and methods have been developed to overcome this [115] [116]. On top of this several experimental contributions to the incoherent signal are considered such as timing errors, the location/setup, environmental response and other cultural noise. To this end, it is recommended that a good self-noise determination should be performed on quiet periods of data from seismometers placed on a stable slab with minimal temperature variation [21].

The setup of a seismic station requires careful commissioning and calibration [20]. This entails the consideration of the instrument response to environmental sources such as temperature, which a setup should attempt to limit. Furthermore, the transfer-function must be determined. This can be achieved with known test waveforms [20] [117] or from a measured transfer-function between two sensors from the ambient background [118].
6.3. Seismometer Self-Noise Determination

These elements are expanded throughout this chapter to produce a process for the self-noise analysis of the microseismometer. This involves an analysis of the core methods, a discussion of the external causes of incoherent contribution and several algorithmic developments to counteract certain existing problems. The aim is to provide a rigorous analysis and development to minimise error injections and understand external contributions to allow a definitive self-noise determination.

6.3.2 Coherence noise testing

The workhorse approach for seismometer noise analysis is known as the coherence method and was developed by Holcomb [109] [112]. This technique assumes that the coherent component between the $X$ and $Y$ seismometers will be the ambient $Z$ and so the remainder will be the test seismometer’s self-noise, as per a linear time invariant system. In this format it is assumed the test seismometer has much higher noise than the reference. The coherence noise estimate is then given as

$$\hat{P}_{N1N1} = P_{XX}(1 - \gamma^2), \quad (6.7)$$

where the spectral coherence is calculated as

$$\gamma^2 = \frac{|P_{XY}|^2}{P_{XX}P_{YY}}. \quad (6.8)$$

The coherence method for calculating the self-noise of a test seismometer, $X$, in the topology shown in Figure 6.3 is summarised in Algorithm 16. An analysis of the method in [109] shows that the result can be represented as a hyperbolic function of the SNR between the ambient, $Z$, and each seismometer’s noise. This means that when the test and reference seismometer have the same noise, it can be evenly apportioned between them leading to an accurate determination. This analysis, however, assumes that the spectral estimation is accurate and the cross-terms do not play a part. Holcomb [109] also introduces a “direct method” of determining the self-noise as

$$\hat{P}_{N1N1} = P_{XX} - P_{XY}.$$

This method determines the self-noise based on the same assumptions as the coherence method.

A three-sensor test has also been developed by Sleeman et al. [113]. The topology of the recorded signal (where $m = 1, 2, 3$ for each sensor) for this three-sensor method is assumed to be

$$Y_m = H_m * Z + N_m.$$

Note that the added noise of each seismometer, $N_m$, is not assumed to be modified by the transfer-function in contrast to the two-sensor approach of Holcomb. The noise estimate of the test seismometer 1 is then given as

$$P_{N1N1} = P_{Y1Y1} - P_{Y2Y1} \frac{P_{Y1Y3}}{P_{Y2Y3}}.$$

In this case the transfer-function does not have to be corrected on all original signals and only on the final estimate, avoiding a source of error.

A synthetic empirical comparison of the direct and three-sensor method by Ringler et al. [21] shows that the Sleeman method is superior for very high SNR (ambient recorded signal to sensor noise). Note that a comparison to the coherence method (6.7) is not given. In this thesis we focus on the analysis of the two-sensor coherence method (6.7).

**Algorithm 16.** The two-sensor self-noise determination coherence method

1: Calculate the PSDs \( P_{XX} \) from (6.4)
2: Calculate the Spectral Coherence \( \gamma^2 \) from (6.8)
3: Estimate the self-noise of seismometer \( X \) as \( \hat{P}_{N1N1} = P_{XX}(1 - \gamma^2) \)

**Analysis of the coherence testing method**

The expression for the spectral coherence, \( \gamma^2 \), is expanded using (6.4), (6.5) and (6.6) to give the expression

\[
\gamma^2 = \frac{|R_X|^2 |R_Y|^2 (P_{ZZ} + P_{Z_1Z} + P_{N1Z} P_{N1N2})}{|R_X|^2 (P_{ZZ} + P_{N1N1} + P_{Z_{N1}} + P_{N1Z}) |R_Y|^2 (P_{ZZ} + P_{N2N2} + P_{Z_{N2}} + P_{N2Z})}.
\]

Notice that the transfer-function correction will have no impact on the coherence. Substituting this into the estimate of the seismometer self-noise (6.7) produces a form

\[
\hat{P}_{N1N1} = |R_X|^2 P_{N1N1} + |R_X|^2 \frac{P_{N2N2} (P_{ZZ} + P_{N1Z} + P_{Z_{N1}})}{(P_{ZZ} + P_{N2N2} + P_{Z_{N2}} + P_{N2Z})} + |R_X|^2 \frac{P_{N1N2} (P_{ZZ} + P_{Z_{N2}} + P_{Z_{N1}} + P_{N2N1})}{(P_{ZZ} + P_{N2N2} + P_{Z_{N2}} + P_{N2Z})} - |R_X|^2 \frac{P_{N1N2} (P_{ZZ} + P_{N1Z} + P_{Z_{N2}})}{(P_{ZZ} + P_{N2N2} + P_{Z_{N2}} + P_{N2Z})} - |R_X|^2 \frac{P_{N2N1} (P_{ZZ} + P_{N1Z} + P_{Z_{N2}})}{(P_{ZZ} + P_{N2N2} + P_{Z_{N2}} + P_{N2Z})} - |R_X|^2 \frac{(P_{N2Z} + P_{Z_{N1}} P_{N1Z})}{(P_{ZZ} + P_{N2N2} + P_{Z_{N2}} + P_{N2Z})}.
\]

This is of the form of a linear equation \( y = mx + c \) where the “true” value is scaled and a constant is added.

The coherence method estimated \( \hat{P}_{N1N1} \) cannot be equal to the true value as the \( c \) term will never vanish except for under the ideal case assumptions that the reference seismometer’s noise is zero, \( N2 = 0 \), and that the cross-terms do completely vanish.

**6.3.3 Delta noise testing**

The coherence method in (6.7) only produces a PSD of the self-noise estimate. For further analysis of the properties of the self-noise it is interesting to obtain an estimate of the time series of the self-noise. To this end, we propose the delta noise method for the determination of the self-noise of a seismometer. The delta noise signal is given by the
6.3. Seismometer Self-Noise Determination

Subtraction of the reference seismometer time series, \( Y \), from the test seismometer time series, \( X \), as

\[
\Delta N_X = X - Y.
\] (6.10)

The PSD of the delta noise signal, \( \Delta N_X \), then produces a self-noise estimate

\[
\hat{P}_{N1N1} = \frac{|\mathcal{F}\{X - Y\}|^2}{\mathcal{F}\{X\}^* \mathcal{F}\{X\} + \mathcal{F}\{Y\}^* \mathcal{F}\{Y\} - (\mathcal{F}\{X\}^* \mathcal{F}\{Y\}) + \mathcal{F}\{Y\}^* \mathcal{F}\{X\}).
\] (6.11)

Substituting (6.4), (6.5) and (6.6) in leads to the estimate of the form

\[
\hat{P}_{N1N1} = |R_X|^2 P_{N1N1}
+ P_{ZZ}(|R_X|^2 + |R_Y|^2 - 2 \cos(\angle R_X - \angle R_Y)|R_X||R_Y|)
+ P_{N2N2}|R_Y|^2
+ P_{Z1N1}(|R_X|^2 - R_X R_Y^*)
+ P_{N1Z}(|R_X|^2 - R_X^* R_Y)
+ P_{Z2N1}(|R_Y|^2 - R_X R_Y^*)
+ P_{N2Z}(|R_Y|^2 - R_X^* R_Y)
- P_{N1N2}(R_X^* R_Y) - P_{N2N1}(R_X R_Y^*),
\] (6.12)

where \( \angle \) denotes the angle between the real and imaginary components. The error for this method is dependent on the accuracy and precision of the transfer-function correction and the noise of the reference seismometer. For example, in the case when the transfer-function correction of both seismometers is exact, \( R_X = R_Y = 1 \), then the delta noise result is given as

\[
\Delta N_X = N1 - N2.
\]

Therefore, if the reference seismometer’s noise is much lower than the test seismometer’s and the transfer-function is well known, the noise estimation can be very accurate. Moreover, the magnitude of the error due to the transfer function correction is impacted by the strength of the ambient signal.

**Remark 27.** Both the delta noise and coherence noise estimates are of the form \( y = mx + c \) where \( y \) is the estimate of the true noise \( x \).

6.3.4 Bias and Variance of the Noise Estimators

The quality of an estimator, \( \hat{\theta} \), of some parameter, \( \theta \), is described in terms of its bias and variance. The bias indicates how far the expected value of the estimator is from the true value and is derived as

\[
\text{Bias}(\hat{\theta}) = E[\hat{\theta}] - \theta.
\]

On the other hand, the variance indicates how much the estimator is expected to deviate from its average value. This is derived

\[
\text{Var}(\hat{\theta}) = E[(\hat{\theta} - E[\hat{\theta}])^2].
\]

Remark 27 states that the noise estimators are of the form \( \hat{x} = mx + c \) where \( x = P_{N1N1} \)
is the noise parameter that is being estimated. The bias of an estimator of this form is given as

\[ Bias(\hat{x}) = E[\hat{x}] - x \]

\[ = E[mx + c] - x \]

\[ = mE[x] + c - x \]

\[ = (m - 1)x + c. \]

Note that the bias is dependent on the \( c \) term of each method and on the ratio of the transfer function correction, \( R_X \). The variance is then given by

\[ Var(\hat{x}) = E[(\hat{x} - E[\hat{x}])^2] \]

\[ = E[(mx + c - E[mx + c])^2] \]

\[ = m^2E[(x - E[x])^2]. \]

### 6.4 Simulation Analysis of Noise Testing Methods

In this section the analysis of the coherence and delta noise methods is applied to simulations. The goal is to quantify and demonstrate how common issues in the test setup and analysis would contribute to the estimation error of each method. The main variables in (6.12) and (6.9) are the transfer-function correction ratios \( R_X \) and \( R_Y \) as these are dictated by estimated parameters. To this end, a simulation is provided for a case where the correction is perfect, and where each estimated parameter is incorrect. This is then summarised for a range of errors and the derived expressions (6.12) and (6.9) are developed into a form to predict an expected bias for a certain test setup.

#### 6.4.1 The effects of transfer-function correction

For both self-noise determination methods, the noise estimate is scaled by the squared magnitude of the transfer-function correction ratio \( |R_X|^2 \). Moreover, the bias term for the delta noise estimate (6.12) is dependent on the difference between the correction ratios of the test and reference seismometer. The coherence method (6.9), though, only depends on the absolute value of the correction ratio of the test seismometer \( R_X \) and does not depend on the correction of the reference seismometer \( R_Y \).

The performance limit of the coherence and delta noise estimation

Consider the case that \( R_X = R_Y = 1 \), that is, the transfer-function has been perfectly corrected. The delta noise estimate is limited by the \( P_{N2N2} \) term. On the other hand, the coherence noise is limited by cross-terms between the \( N1 \) and \( Z \) on top of terms including the reference seismometer noise \( N2 \).

This was verified through a simulation of the seismometer self-noise testing topology in Figure 6.3 where each component \( Z, N1 \) and \( N2 \) was generated as a white Gaussian process. The transfer-function for the test and reference seismometer was of the form (6.2).
Table 6.1. Noise method simulation parameters

<table>
<thead>
<tr>
<th>Component</th>
<th>Simulation 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seismic signal $Z$</td>
<td>$Z \sim \mathcal{N}(0, 10)$</td>
</tr>
<tr>
<td>Test seismometer noise $N_1$</td>
<td>$N_1 \sim \mathcal{N}(0, 0.1)$</td>
</tr>
<tr>
<td>Reference seismometer noise $N_2$</td>
<td>$N_2 \sim \mathcal{N}(0, 0.001)$</td>
</tr>
<tr>
<td>Test seismometer TF $H_X$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$f_X = 1/35 \text{Hz}$, $\bar{f_X} = 1/35 \text{Hz}$</td>
</tr>
<tr>
<td></td>
<td>$D_X = 1.1, \bar{D_X} = 1.1$</td>
</tr>
<tr>
<td>Reference seismometer TF $H_Y$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$f_Y = 1/34 \text{Hz}$, $\bar{f_Y} = 1/34 \text{Hz}$</td>
</tr>
<tr>
<td></td>
<td>$D_Y = 1, \bar{D_Y} = 1$</td>
</tr>
</tbody>
</table>

where $z_1 = z_2 = 0$ and the poles were derived from (6.3). The simulation parameters are given in Table 6.1 and 100000 samples of each process were generated. Figure 6.5 shows the noise estimates for both delta noise and coherence method along with the $c$ terms. In this case the delta noise method had superior performance.

![Noise Estimates](image1)

![Noise Estimate Errors](image2)

Figure 6.5. Simulation 1: Perfect correction. PSDs calculated using the Welch method averaging over two Hanning windows with sampling rate defined as $f_s = 1 \text{Hz}$.

**Gain estimation error**

The gain, $|H|$, of the transfer-function in (6.2) purely affects the amplitude of the output. As such, error in the estimation of $|H|$ means that the corrected ratio $R$ is a real number.
and so $\angle R = 0$. Figure 6.6 shows the result for two simulations with parameters as in Table 6.1 except $|\tilde{H}_X| = 900$ in the first and $|\tilde{H}_X| = 1100$ in the second, and so the estimated gain is incorrect. The result shows that the error level is homogeneous across the bandwidth meaning that error in the gain parameter leads to a self-noise estimation error which is constant across frequency and does not cause phase errors.

![Noise Estimates and Noise Estimate Errors](image)

**Figure 6.6. Simulation 2: Incorrect estimated gain correction.** PSDs calculated using the Welch method averaging over two Hanning windows with sampling rate defined as $f_s = 1$ Hz.

### Corner-frequency and damping constant estimation error

The corner frequency, $f$, and damping constant, $D$, set the values of the poles in (6.3). An error in the value of these estimated parameters used to invert the transfer-function will cause an error in both the magnitude and angle of the correction ratio, $R$. Figure 6.7 shows the results for two simulations with the parameters as in Table 6.1 except the estimated corner frequency for the test seismometer, $\tilde{f}_X$, was given as $\tilde{f}_X = 1/33$ Hz or $\tilde{f}_X = 1/37$ Hz in each simulation. The coherence noise estimate was close to the actual value of the noise PSD whereas the delta noise estimate below $f_X$ had a significant error. The delta noise bias was shaped around the corner-frequency, where the error had a high constant value below $f_X$ and returned to the level of the noise PSD $P_{N1N1}$.

Figure 6.8 shows the results for two simulations with the parameters as in Table 6.1 except the estimated damping constant for the test seismometer, $\tilde{D}_X$, was altered to be $\tilde{D}_X = 1$ then $\tilde{D}_X = 1.2$. The coherence noise estimate was again close to the actual value of the noise PSD as was the case for the corner frequency. On the other hand, the delta noise estimate bias has a “hump” around the corner-frequency but returns to the level of the noise PSD, $P_{N1N1}$.

**Remark 28.** An error in the estimated gain of the transfer-function causes a constant error across the bandwidth of the self-noise estimation. On the other hand, a change in the corner-frequency or damping constant cause a shaped response as they modify the poles.
6.4. Simulation Analysis of Noise Testing Methods

Figure 6.7. Simulation 3: Incorrect estimated corner frequency. PSDs calculated using the Welch method averaging over two Hanning windows with sampling rate defined as $f_s = 1$ Hz.

Figure 6.8. Simulation 4: Incorrect estimated Damping. PSDs calculated using the Welch method averaging over two Hanning windows with sampling rate defined as $f_s = 1$ Hz.

6.4.2 Error analysis: estimate performance and prediction

To examine the full range of behaviour of the coherence and delta noise estimation methods a model was simulated with the parameters as in Table 6.1. The estimated test seismometer gain was varied so that the transfer-function correction ratio was between $|R_X| = 0.2$ and $|R_X| = 2$. Figure 6.9 shows that the delta noise bias was a positive concave symmetric function with a minimum at $|R| = 1$. On the other hand, the coherence noise estimate bias begins negative for $|R_X| = 0.2$ and increases according to the scaling by $|R_X|^2$ in the estimator (6.9).

In summary, the coherence method is significantly more robust to errors in the transfer-function correction than the delta noise estimation. However, in the case where the transfer-function correction ratio is perfect, $R_X = 1$, the delta noise estimate is more accurate and has the added benefit that it can produce a time series estimate of the noise.

We have examined the affects of a practical estimation on the coherence and delta noise
determination methods. The goal now is to use this to predict what level of accuracy of
the estimation can be obtained for a certain scenario. This is described in terms of the
transfer-function correction accuracy and SNR, the ratio of the ambient signal, \(Z\), to the
test seismometer noise, \(N_1\). This can be achieved by examining the bias functions in
Section 6.3.4 given as

\[
\text{Bias}(\hat{P}_{N1N1}) = (m - 1)P_{N1N1} + c.
\]

Dividing by the noise PSD yields

\[
\frac{\text{Bias}(\hat{P}_{N1N1})}{P_{N1N1}} = (m - 1) + c,
\]

where \(\frac{\text{Bias}(\hat{P}_{N1N1})}{P_{N1N1}}\) can be viewed as the error as a ratio of the noise power. Substituting the
values for the coherence noise expression (6.9) and assuming that the reference seismometer
is perfectly corrected \((R_Y = 1)\) gives the expression

\[
\frac{\text{Bias}(\hat{P}_{N1N1})}{P_{N1N1}} = \left( |R_X|^2 - 1 \right) \frac{P_{ZZ}}{P_{N1N1}} + \frac{P_{N2N2}}{P_{N1N1}} \frac{P_{ZZ} + P_{N2N2}}{P_{N2N2} + P_{N1N1}} + \frac{c'}{P_{N1N1}} R_Y \text{TF magnitude error}
\]

\[
\frac{\text{Bias}(\hat{P}_{N1N1})}{P_{N1N1}} = \left( |R_X|^2 - 1 \right) \frac{P_{ZZ}}{P_{N1N1}} + \frac{P_{N2N2}}{P_{N1N1}} \left( |R_X|^2 + 1 - 2 \cos(\angle R_X) |R_X| \right) \text{SNR scaled by TF error}
\]

\[
\frac{\text{Bias}(\hat{P}_{N1N1})}{P_{N1N1}} = \left( |R_X|^2 - 1 \right) \frac{P_{ZZ}}{P_{N1N1}} + \frac{P_{N2N2}}{P_{N1N1}} + \frac{c'}{P_{N1N1}} \frac{P_{ZZ} + P_{N2N2}}{P_{N2N2} + P_{N1N1}} + \frac{c'}{P_{N1N1}} R_Y \text{Remaining cross terms.}
\]

On the other hand, the delta noise expression (6.12) yields

\[
\frac{\text{Bias}(\hat{P}_{N1N1})}{P_{N1N1}} = \left( |R_X|^2 - 1 \right) \frac{P_{ZZ}}{P_{N1N1}} + \frac{P_{N2N2}}{P_{N1N1}} \text{TF magnitude error}
\]

\[
\frac{\text{Bias}(\hat{P}_{N1N1})}{P_{N1N1}} = \left( |R_X|^2 - 1 \right) \frac{P_{ZZ}}{P_{N1N1}} + \frac{P_{N2N2}}{P_{N1N1}} \left( |R_X|^2 + 1 - 2 \cos(\angle R_X) |R_X| \right) \text{SNR scaled by TF error}
\]

\[
\frac{\text{Bias}(\hat{P}_{N1N1})}{P_{N1N1}} = \left( |R_X|^2 - 1 \right) \frac{P_{ZZ}}{P_{N1N1}} + \frac{P_{N2N2}}{P_{N1N1}} + \frac{c'}{P_{N1N1}} R_Y \text{Ratio of seismometer noise}
\]

\[
\frac{\text{Bias}(\hat{P}_{N1N1})}{P_{N1N1}} = \left( |R_X|^2 - 1 \right) \frac{P_{ZZ}}{P_{N1N1}} + \frac{P_{N2N2}}{P_{N1N1}} + \frac{c'}{P_{N1N1}} \frac{P_{ZZ} + P_{N2N2}}{P_{N2N2} + P_{N1N1}} + \frac{c'}{P_{N1N1}} \text{Remaining cross terms.}
\]

(6.15)

The cross-terms cannot be directly estimated and so the are summarised by the \(c'\) term.

Figure 6.9. Mean noise estimation bias with gain change.
For the coherence noise estimator these are

$$c'_{\text{Coh}} = \frac{|R_X|^2}{P_{N1N1}} \frac{P_{N2N2}(P_{N1Z} + P_{Z1N})}{P_{N1Z}(P_{ZZ} + P_{N2N2} + P_{Z2Z} + P_{N22})} + \frac{|R_X|^2}{P_{N1N1}} \frac{(P_{Z1N} P_{N22} + P_{N1Z} P_{N22})}{P_{N22}(P_{ZZ} + P_{N2N2} + P_{Z2Z} + P_{N22})} - \frac{|R_X|^2}{P_{N1N1}} \frac{P_{N1Z}(P_{ZZ} + P_{N2N2} + P_{Z2Z} + P_{N22})}{P_{N22}(P_{ZZ} + P_{N2N2} + P_{Z2Z} + P_{N22})}$$

The delta noise estimators cross-terms are given as

$$c'_{\text{Delta}} = \frac{P_{N1Z}}{P_{N1N1}} \left( |R_X|^2 - R_X R_Y^* \right) + \frac{P_{N1Z}}{P_{N1N1}} \left( |R_X|^2 - R_X R_Y^* \right) + \frac{P_{N2Z}}{P_{N1N1}} \left( |R_Y|^2 - R_X R_Y^* \right) + \frac{P_{N2Z}}{P_{N1N1}} \left( |R_Y|^2 - R_X R_Y^* \right) - \frac{P_{N2N2}(R_X^* R_Y) - P_{N2N1}(R_X R_Y^*)}{P_{N1N1}}$$

and is therefore affected by the transfer-function correction. The $c'_{\text{Coh}}$ terms are effectively “squared” and so the contribution of $c'_{\text{Coh}}$ is relatively small compared to the delta noise $c'_{\text{Delta}}$.

Nevertheless, the equations (6.15) and (6.16) can be used to examine the accuracy of the self-noise determination for a given site’s transfer-function correction and SNR, neglecting the impact of the cross-terms. Figure 6.10 shows the $\frac{\text{Bias}(P_{N1N1})}{P_{N1N1}}$ as function of SNR for $R_X = 0.99, 1, 1.01$ and for $P_{N2N2} = 0.01, 1$. This was achieved with a fixed $P_{N1N1} = 1$, $P_{N2N2} = 0.01$ or $P_{N2N2} = 1$ and $P_{ZZ}$ varied to give the reported SNR. When $R_X = 1$, the coherence method tended to the ratio $\frac{P_{N2N2}}{P_{N1N1}}$ as the SNR increased and towards zero when the SNR was low. The change in $R_X$ caused an offset bias (owing to the $|R_X|^2 - 1$ term) around these points. The delta noise methods tended towards $\frac{P_{N2N2}}{P_{N1N1}}$ for low SNR and was constant for $R_X = 1$. When $R_X \neq 1$, the error is offset from the ratio $\frac{P_{N2N2}}{P_{N1N1}}$ (owing to the $|R_X|^2 - 1$ term) at low SNR and rapidly increased for high SNR. Note that this error analysis is framed by the assumption that the test seismometer’s noise is much lower than the reference. For the case where $\frac{P_{N2N2}}{P_{N1N1}} = 1$ this is not the case and the estimate would be apportioned between the two, as per Holcomb’s previous analysis [109] described in Section 6.3.2.

**Remark 29.** The equations (6.15) and (6.16) provide a method to examine the coherence and delta noise method in terms of the expected accuracy that might be obtained from an estimation scenario. Excess test seismometer noise determination error is introduced from an error in the transfer-function correction. For the delta noise method this causes the accuracy to rapidly degrade at high SNRs. For a low SNR the error is bounded by the reference seismometer. The coherence method is robust for all SNRs and is bounded by the reference seismometer noise.

**Remark 30.** For a self-noise determination, the coherence method should be used to provide the definitive result. On the other hand, the delta noise method provides a time series and is useful for an in depth analysis, complementing the coherence method.
6.5 Estimation of the microseismometer noise: Theoretical perspectives

So far, the analysis has considered the error at a point in frequency for “white” ambient and noise signals. To apply the results for a practical estimation, we must consider the variation across the bandwidth. When combined with the bounds derived in Section 6.4.2, this yields an understanding of the bandwidths in which an estimation is reliable/robust in the context of the limitations of the experimental setup.

Here, the frequency dependence of the estimation based on the ambient background is first addressed. A description of the theoretical noise model of the microseismometer is then provided. This yields a method to obtain a prospective SNR and, hence, accuracy of the self-noise determination for a given experiment.

6.5.1 Experimental setup and background signal considerations

We first consider how the physical domain (displacement, velocity etc.) in which the computation is performed affects the self-noise determination process, the flow of which is shown in Figure 6.11. The inputs are considered to be the noise of the test seismometer, reference sensor and the ambient signal. The transfer-function block refers to the application and subsequent inversion of the seismometer response before the self-noise determination. The simulations in Section 6.4 consider these to be defined in the same domain as the transfer-function. For the microseismometer, the self-noise determination is derived from the velocity channel and so the two pole-zero pair high-pass filter model in (6.2) is defined in the velocity-domain, whereas the self-noise is typically described in the acceleration domain. As a result, the defined input signals are effectively integrated before the application of the transfer-function and differentiated after its inversion, within the transfer-function block. This must be considered when interpreting the results of a self-noise determination with respect to error in the transfer-function correction.

The ambient seismic signal, $Z$, is also spectrally shaped. This is described by Peterson
6.5. Estimation of the microseismometer noise: Theoretical perspectives

[119] who derives a “New Low Noise Model” (NLNM) and “New High Noise Model” (NHNM) which describe the range of the geophysical ambient signal. The main feature within the seismic bandwidth is the microseismic peak which can be 0.01 – 0.25 Hz. This is shown in the top pane of Figure 6.15.

Excess cultural signal contributions

The base assumption of the introduced noise testing methods is that the self-noise is the incoherent component between the test and reference seismometers. As a result, any element that does not couple linearly or is orthogonal will inject an excess contribution to the self-noise estimate. In Section 6.3.1 several such contributions were mentioned and are termed as cultural noise. Note that these are, in fact, latent variable contributions. This excess noise is typically significant at both high (> 2 Hz) and low (< 0.01 Hz) frequencies, and has both anthropogenic and environmental sources [21] [120] [121] [122].

To provide context for how each cultural component source will cause an excess self-noise term we separate the latent components on their coupling to the seismometer. Three signal contribution mechanisms are defined:

1. A parallel component in the ground dynamics,
2. An orthogonal or non-linear dynamic coupling,
3. Non-dynamic coupling to the microseismometer.

Each case is considered with respect to the reference seismometer, that is, non-linear dynamic refers to a non-linear coupling between the ground accelerations sensed by the microseismometer and reference sensors. This is illustrated in Figure 6.12.

The parallel component in the ground dynamics is coherent between the microseismometer and reference sensors and is usually seismic in origin. However, we consider that there can be aseismic contributions to this signal such as pressure [8]. Aseismic dynamic contributions cannot be mitigated for a given site as they create a ground acceleration. Instead a measurement of the source and a model of the coupling to ground acceleration is required to remove it [8]. Note that these do not cause an issue for the self-noise estimation as long as their relationship between the reference and microseismometer is linear.

Non-dynamic contributions correspond to the response of the sensor to a direct incident environmental source, such as temperature [123], magnetic [20] or acoustic [118]. These sources can either be mitigated from some physical attenuation or removed during post-processing. For the latter, a model of the coupling is required. In the following Chapter 7, such a method for removing the thermal component from the microseismometer is proposed. The reference seismometer will not necessarily respond in the same way as the test seismometer so these contributions are incoherent [20].
Orthogonal components between the reference and microseismometer are a legitimate measurement and not a noise term. However, an orthogonal component will be injected into a self-noise estimate as it is incoherent to the test sensor [115]. This effect is analysed in Section 6.7.1. Furthermore, an acceleration can be caused by a rotational component. If the microseismometer is rotating with respect to the reference instrument at a rate $\dot{\theta}$ then this will impart an angular acceleration
\[
a_{\text{angular}} = r \ddot{\theta},
\]
and a centripetal acceleration
\[
a_{\text{centripetal}} = r (\dot{\theta})^2,
\]
where $r$ is the distance to the reference seismometer. The rotational/tilt acceleration is often separated from the linear component in seismic studies, for example, in Rayleigh wave seismology [124] or to remove the impact of wind [125]. In terms of the self-noise estimation, this component will be an excess contribution and is limited by shortening the distance, $r$, between the seismometers.

Non-linear components can be injected either through a local heterogeneity or mounting resonance [21] [118]. These are difficult to reject with post-processing and so are best limited using a stable slab [21].

The suggestions of Ringler et. al. [21] are chosen to limit such injections. The use of a common, stable slab will improve coupling and the selection of a quiet period would limit the injection from external non-dynamic or non-linear components. This is on top of isolation and mitigation from environmental sources. These latent contributions are shown in Figure 6.13. Mimoun et. al. [7] provides a noise model/map for the InSight deployment, taking into account such contributions. In this chapter, the derived noise method’s analysis is applied and further methods are developed for the limitation and identification of incoherent contributions.
6.5. Estimation of the microseismometer noise: Theoretical perspectives

Figure 6.13. Seismic noise testing and incoherent source contributions.

6.5.2 Noise Model of the microseismometer

The noise of the microseismometer is also shaped over the frequency domain according to
the its physical mechanics and acquisition electronics. There are three main contributions:

1. the the analogue to digital converter (ADC),
2. the thermodynamic limits of the suspension,
3. the pre-amplifier, electronics noise.

The ADC noise is produced from the quantisation floor shaped by the sensor’s transfer-
function, whereas the thermodynamic suspension noise is a broadband source [126]. The
electronics contribution is the pre-amplifier noise shaped by the response of the suspension
to a displacement [23]. Figure 6.14 shows the rPSD for each of these aspects. The total
self-noise is given by the sum of each component. This sum is modelled by a combination
of power-laws as (6.17) which splits the contributions into a random walk, $f^{-1}$, a variable
power law, white noise, $f^0$, and a high frequency $f^2$ component. This model incorporates
the aspects of the model in Figure 6.14 and can also account for environmental components,
limiting the estimation for a given site. The overall self-noise is designed to be under the
requirements of the InSight mission, described by the green line in Figure 6.14, so that
the mission can meet its fundamental science goals [3].

\[
\text{rPSD}_{\text{noise}}(f) = \left( \left( \frac{A_{RW}}{f} \right)^2 + \left( \frac{A_{PL}}{f} \right)^{\text{expPL}} + A_{W}^2 + \left( \frac{f}{A_{HF}} \right)^4 \right)^{\frac{1}{2}}. \tag{6.17}
\]

6.5.3 Synthetic simulation of the noise model

The microseismometer noise model can be used to produce a simulated, synthetic time
series. Creating a synthetic time series from a power spectrum can be achieved by per-
forming the inverse FFT with an assigned random phase. The resulting time series is
termed a surrogate of the original [127] [128]. A surrogate generated from the microseis-
ometer noise model is then calculated by first evaluating (6.17) for the argument $f$ being
the values of the one-sided, positive, frequency vector, given as

$$s_{\text{one-sided}} = [0, f_s/N, 2f_s/N, \ldots, f_s/2], \quad (6.18)$$

where $f_s$ is the sampling rate and $N$ is the number of data points for the desired time series. This yields the noise model spectrum. In order to select the surrogate’s phase, each element of the spectrum $rPSD(f)$ is then multiplied by a complex-valued scalar, given as

$$\theta_{\text{Rand}} = \exp(-j2\pi p), \quad (6.19)$$

where $p \sim \mathcal{N}(0, 1)$ is a sample of a Gaussian distribution with zero mean and unit variance. This yields $rPSD(f_i)_{\text{Rand}} = rPSD(f_i)\theta_{\text{Rand},i}$ for every element of $f_i, i = 1, 2, \ldots, N/2$ in $s_{\text{one-sided}}$. The so-obtained spectrum is then converted into a double-sided spectrum of length $N$ as

$$rPSD_{\text{noise-model-surrogate}} = [rPSD(s_{\text{one-side}}), \text{Rev}(rPSD(s_{\text{one-side}}))], \quad (6.20)$$

where $\text{Rev}(\cdot)$ denotes the reversed one-sided spectrum as

$$\text{Rev}(rPSD(s_{\text{one-side}})) = [rPSD(f_s/2 - f_s/N), rPSD(f_s/2 - 2f_s/N), \ldots, f_s/N], \quad (6.21)$$

when $N$ is even, and

$$\text{Rev}(rPSD(s_{\text{one-side}})) = [rPSD(f_s/2), rPSD(f_s/2 - f_s/N), \ldots, f_s/N],$$

when $N$ is odd. As such, the $rPSD_{\text{noise-model-surrogate}}$ is conjugate symmetric so its IFFT is real. The resultant time series is normalised by $\frac{N}{\sqrt{2N/f_s}}$. The process is summarised in Algorithm 17.
6.5. Estimation of the microseismometer noise: Theoretical perspectives

**Algorithm 17.** Synthetic microseismometer noise time series generation

Set the noise model parameters $A_{RW}$, $A_{PL}$, $exp_{PL}$, $A_{FL}$, $A_W$ and $A_{HF}$ and the time series length $N$.

Calculate half-spectrum noise model $rPSD_{\mu S-noise}(s_{one-sided})$ from (6.17) and (6.18). Multiply each element of $rPSD_{\mu S-noise}(s_{one-sided})$ by a randomised phase calculated in (6.19).

Create double-sided noise model, $rPSD_{noise-model-surrogate}$, in (6.20).

Calculate the IFFT of $rPSD_{noise-model-surrogate}$ to obtain the time series $N_{noise-model}$ and normalise by $\frac{N}{\sqrt{2N/f_s}}$.

6.5.4 Practical self-noise determination: the accuracy limits for a given site

In this Section, the theoretical spectrum of the ambient signal and the microseismometer noise has been introduced. The theoretical SNR is determined as the ratio of the NLNM or NHNM and the derived noise model and, hence, the equations (6.15) and (6.16) predict the expected potential error, $\frac{Bias_{N1N1}}{P_{N1N1}}$, for a self-noise determination. This is displayed in Figure 6.15 which shows the NLNM and NHNM along with the microseismometer noise model and the predicted $\frac{Bias_{N1N1}}{P_{N1N1}}$ for each SNR (where $P_{ZZ}$ is given by the NLNM/NLHM) and $R_X = 0.99, 1, 1.01$ with $P_{N2N2} = 10^{-20}$. In the case of the coherence method, the result is predominantly shaped by $\frac{1}{P_{N1N1}}$ and there is no apparent difference between the estimation for the high or low noise model. The delta noise method is significantly affected by the high SNR in the bandwidth of the microseismic peak, which is at the noise floor of the sensor, and at higher frequencies in the NHNM model. Note that this simulation does not take into account the cultural noise as this is dictated by the individual test.

6.5.5 Expected results and interpretation of self-noise determination results

The results of this chapter are intended to determine the best self-noise estimate of the microseismometer from experimental data. These results are presented in the form of a "noise curve" figure. It has been shown that this estimate depends on the method used, transfer-function parameters and environmental conditions/injections. In this Section, the expected appearance of this curve are discussed qualititively to aid the interpretation with respect to the analysis.

The noise curves are presented as a rootPSD (sometimes described as an amplitude spectral density) of the acceleration. The rtPSD is in terms of m/s$^2$/rtHz to indicate the amplitude of the acceleration signal in each bandwidth. As such, the curve shows the sensitivity floor of the sensor to ground motion acceleration although seismic signals are often also presented or in the velocity and displacement domain. In general logarithmic axes are used here. This is a common choice for seismic data owing to the typical bandwidths and features of interest. For cases where the features are more tonal then linear axes are sometimes used. Other works may display the data in terms of power (decibels per 1 (m/s$^2$))$^2$/Hz) instead of amplitude or the period (in seconds) instead of frequency.
The noise curve can be divided into low, mid and high frequency sections. In terms of the theoretical noise model of the microseismometer (in Figure 6.14) the high-frequency bandwidth begins above the sensor resonance at 6 Hz where the ADC noise contribution becomes dominant. The low-frequency portion begins below $1 \times 10^{-1}$ Hz where the noise rises from the preamplifier. In the middle range in between these frequencies, the noise model is mostly flat with a slight dip at the 6 Hz sensor resonance.

The NLNM and NHNM introduced in Section 6.5.4 shows the spectrum belonging to the ambient geophysical signal [119]. The key feature is the microseismic peak (in the frequency range $5 \times 10^{-1}$ Hz – 1 Hz) which is observed with local variation everywhere in the world. This feature is within the middle part of the microseismometer’s noise curve and thus represents the highest signal to noise ratio section. On top of the geophysically generated background signal, several cultural components were discussed in Section 6.5.1. Anthropogenic signals are typically introduced in the high frequency range whereas environmental injections are typically in the lower frequency bandwidth.

Figure 6.12 shows the mode in which these signals couple to the seismometers. The presented analysis of the noise testing methods shows that given accurate transfer-function corrections, parallel signals between a test and reference seismometer coupling through ground-dynamics are removed to give a noise curve estimate. Should this not be the case...
then the ambient signal will be injected into the noise curve as an orthogonal component. 
For example, the microseismic peak is common to both sensors and so its appearance 
in the noise curve indicates a misalignment or transfer-function parameter error. There 
is little geophysical background signal in the low frequency zone (until tidal frequencies 
$< 5 \times 10^{-5}$ Hz [110]) and the environmental injections in this region are typically from 
non-dynamic coupling. These components are not removed through the noise estimation 
methods and will be directly present in the noise curve. Therefore, the obtained low 
frequency performance is test setup dependent. The high frequency injections typical in 
anthropogenic environments are also difficult to reject as they can excite resonances on 
top of being orthogonal or non-dynamic. Finally, transient signals (i.e. glitches or events) 
present in the test data are difficult to remove from processing and so inject power into 
the noise curve.

6.6 Noise campaigns and data acquisition

The goal of the experimental data is two fold:

1. Determine the performance limits of the microseismometer instrument across a full 
bandwidth of operation,

2. Determine the self-noise and transfer-function parameters of the InSight flight model 
sensors.

The analysis in Section 6.5 shows that the self-noise determination can be difficult at 
certain bandwidths owing to cultural and environmental components. To this end, noise 
tests were performed at

1. The CNES Assembly and Integration Test (AIT) facility in Toulouse,

2. The Black Forest Observatory (BFO),

3. The Oxford University Atmospheric Physics laboratory.

The CNES laboratory was available for the testing of the flight model (FM) sensor units. 
Human activity was minimised to avoid non-stationary features, however, the cultural 
optice at this site is a factor at high frequencies. The tests performed here are termed 
SP1, SP2 and SP3 CNES. The SP1 sensor is the vertical and the SP2 and SP3 sensors are 
the two horizontal units for the InSight mission. At Oxford it was possible to run a long 
period test on a vertical qualification model (QM) to obtain a self-noise estimate for the 
sensor at low frequencies, in a test termed QM Vert 2017. The BFO vault is a well-known 
seismically quiet environment [129]. A horizontal QM sensor was tested here to obtain a 
good self-noise determination at high frequencies, in a test termed QM BFO. The datasets 
are summarised in Table 6.2.

The InSight probe was launched on 5th May 2018 and is currently in the cruise phase, 
en-route to Mars. In the cruise phase the system is in free-fall and so the two horizontal 
sensors can function out of saturation. There is no ambient signal in this environment and 
the sensors will only observe the spacecraft on top of its self-noise. Two observations are
Table 6.2. Noise testing datasets used

<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
<th>Reference</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>QM Vert 2017</td>
<td>Oxford</td>
<td>1×Guralp CMG-3T</td>
<td>49 hours</td>
</tr>
<tr>
<td>QM BFO</td>
<td>BFO</td>
<td>2×STS2</td>
<td>5 hours</td>
</tr>
<tr>
<td>SP1 CNES</td>
<td>CNES</td>
<td>1×STS2, 1×Trillium Compact</td>
<td>8 hours</td>
</tr>
<tr>
<td>SP2 CNES</td>
<td>CNES</td>
<td>2×STS2</td>
<td>8 hours</td>
</tr>
<tr>
<td>SP3 CNES</td>
<td>CNES</td>
<td>2×STS2</td>
<td>8 hours</td>
</tr>
<tr>
<td>SP2 Space event</td>
<td>Deep space</td>
<td>none</td>
<td>15 minutes</td>
</tr>
<tr>
<td>SP3 Space event</td>
<td>Deep space</td>
<td>none</td>
<td>15 minutes</td>
</tr>
<tr>
<td>SP2 Space long period</td>
<td>Deep space</td>
<td>none</td>
<td>28 hours</td>
</tr>
<tr>
<td>SP3 Space long period</td>
<td>Deep space</td>
<td>none</td>
<td>28 hours</td>
</tr>
</tbody>
</table>

Data acquired at Oxford

For the dataset acquired at Oxford, the QM vertical microseismometer was placed in a test chamber and tilted so the gravity vector along the sensor axis was equivalent to Martian gravity. The velocity channel was recorded at $f_s = 100$ Hz by a 24-bit Centaur data logger. A Guralp three-axis seismometer was recorded alongside the microseismometer with the same data logger for use as a reference.

Data acquired at CNES and BFO

The data acquired at CNES and BFO were recorded using the full flight system. The data logger, known as the “Ebox”, recorded at a sample rate of $f_s = 100$ Hz on a 24-bit ADC. Moreover, an STS2 or Trillium Compact seismometer was recorded on a 26-bit Centaur data logger for use as the reference seismometer. At CNES the tests were performed in a cleanroom facility with two STS2 seismometers for the SP2 and SP3 datasets but only one STS2 instruments and a Trillium Compact for the SP1 dataset. At BFO the test was performed in an underground vault along with two STS2 instruments.

Data acquired from InSight cruise

The data acquired from the cruise stage is also recorded on the Ebox data logger. The SP2/3 Space long period dataset is recorded at $f_s = 2$ Hz whereas the SP2/3 Space event high-rate dataset is recorded at $f_s = 100$ Hz. The calibration data is also recorded at $f_s = 100$ Hz.
6.7 The Self-Noise analysis of the microseismometer: algorithmic developments

In this section the required algorithmic developments for the self-noise determination of the microseismometer are provided. The coherence and proposed delta noise methods for a self-noise estimation have been outlined in Section 6.3 and are calculated through (6.7) and (6.11). The analysis of both methods has shown that the coherence method is more robust and so is used to provide the definitive noise estimate rPSD. This is obtained from an estimation of $P_{XX}$, $P_{YY}$ and $P_{XY}$ using the Welch periodogram method [60]. Although the coherence method is robust to high SNR, it is prudent to follow the advice of Ringler et. al. and evaluate the self-noise over a quiet period of the data containing no obvious features such as a spike. This minimises any injection of the incoherent sources discussed in Section 6.5.1. In this section, the algorithmic developments to perform the required preprocessing and calibration are introduced. Moreover, a technique is developed to obtain the most appropriate self-noise curve of a sensor for a given dataset.

6.7.1 Geometric Alignment

The microseismometer is designed to operate under Martian gravity which is $3.71 \text{ ms}^{-2}$. The vertical sensor will then be saturated when it is operated vertically under the Earth’s gravity of $9.81 \text{ ms}^{-2}$. As such, it must be tilted until the incident gravity component is equivalent to Mars. To perform a noise determination of the sensor it must be compared to an equivalent, geometrically aligned component of the reference. To this end, a method to calculate the alignment angles is proposed.

The synthetic reference signal aligned with the microseismometer is created as

$$Y_U = (\sin(\theta_{NE})Y_E + \cos(\theta_{NE})Y_N) \cos(\theta_Z) + \sin(\theta_Z)Y_Z,$$

(6.22)

where $Y_N$, $Y_E$ and $Y_Z$ are the three north, east and vertical axes of the reference seismometer. The goal is to find the angles $\theta_{NE}$ and $\theta_Z$ that produce the component $Y_U$ which has the maximum coherence with the microseismometer output $X$. This is achieved through a coarse to fine grid search (hill climbing) optimisation. First, a quiet period (meaning it contains no obvious spikes or glitches) of the recorded data is selected for the time series $X$, $Y_N$, $Y_E$ and $Y_Z$. The next step is to select the desired bandwidth (the region between the frequencies $f_{bottom}$ and $f_{top}$) in which to maximise the coherence. This is chosen to be in the main seismic band where cultural noise and instrument noise are not significant, generally $0.1 - 6 \text{ Hz}$. These are input to Algorithm 18 with an initialisation of the angles $\theta_{Z,\text{init}}$ and $\theta_{NE,\text{init}}$ and the range and step-size of the angle change, $\theta_{\text{range}}$ and $\Delta \theta$. For each step of the algorithm the candidate $Y_U$ is evaluated and its coherence with $X$ calculated. The output is values of $\theta_{Z,i}$ and $\theta_{NE,j}$ which yields the maximum value of $E[\gamma^2]$ (calculated in (6.8)) within the frequency range $f_{bottom}$ and $f_{top}$. These values are again input into Algorithm 18 with a smaller range and step-size until the desired precision has been achieved.
Algorithm 18. Geometric alignment of seismometer axes

1: **Initialise:** input time series $X$, $Y_N$, $Y_E$ and $Y_Z$.
2: **Initialise:** grid search parameters $\theta_{Z,\text{init}}$, $\theta_{NE,\text{init}}$, $\theta_{\text{range}}$, $\Delta \theta$ $f_{\text{bottom}}$ and $f_{\text{top}}$.
3: for $i = 1, \ldots, r$ do
4: $\theta_{Z,i} = \theta_{Z,\text{init}} - \theta_{\text{range}} + i\Delta \theta$.
5: for $j = 1, \ldots, r$ do
6: $\theta_{NE,j} = \theta_{NE,\text{init}} - \theta_{\text{range}} + j\Delta \theta$.
7: Obtain $Y_{U,i}$ from (6.22) with $\theta_{Z,i}$ and $\theta_{NE,j}$.
8: Evaluate coherence, $\gamma^2$ between $Y_{U,i}$ and $X$ from (6.8).
9: Calculate $E[\gamma^2]$ for the range $f_{\text{bottom}} \leq f \leq f_{\text{top}}$.
10: end for
11: end for
12: **Output:** the value of $\theta_{Z,i}, \theta_{NE,j}$ which has the maximum value of $E[\gamma^2]$.

Alignment coherence bandwidth selection

The result of the coherence alignment Algorithm 18 depends on the coherence of the data within the bandwidth range $f_{\text{bottom}} \leq f \leq f_{\text{top}}$. The coherence is affected either by the lack of an ambient signal or the presence of an excess noise, as described in Section 6.5. Figures 6.16 and 6.17 show the alignment angle error computed by Algorithm 18 for a range of bandwidths for both BFO and SP3 datasets compared to that in the range $f_{\text{bottom}} = 0.1$ Hz and $f_{\text{top}} = 6$ Hz. It is shown the error fluctuates significantly at lower frequencies as there is no ambient component so the choice is effectively random. At higher frequencies the error degrades to the order of several degrees. In general it is best to use the largest bandwidth possible so $f_{\text{bottom}} = 0.1$ Hz and $f_{\text{top}} = 6$ Hz provides a compromise between this and the injection of incoherent contributions. The angle error for high frequency bandwidths for the SP3 dataset is higher than for BFO as it contains significant cultural noise. The result for the BFO dataset begins to degrade at high frequencies as the the signal power decreases, at as low frequencies.

Impact of alignment error

A misalignment causes an erroneous estimation of the self-noise for both vertical and horizontal sensors [21] [115]. We now present an analysis of the effect of alignment errors on the coherence and delta noise methods.

The noise testing topology in Figure 6.3 shows that the recorded signals from the test and reference seismometers consist of a common ambient $Z$ component and the individual noise processes $N1$ and $N2$. The aligned reference seismometer axis, $Y$, is synthesised from three orthogonal axes (as in (6.22)) and the ambient signal in this direction denoted as $Z(\theta_{NE},\theta_Z)$. Note that the reference noise process, $N2$, is also synthesised along this axis but can be considered isotropic. To illustrate the effect of an alignment error on the self-noise estimate, consider the 2D case where the aligned ambient component is

$$Z(\theta) = (\sin(\theta)Z_1 + \cos(\theta)Z_2),$$

where $Z_1$ and $Z_2$ are the orthogonal (reference) axes within the plane. In this case the
6.7. The Self-Noise analysis of the microseismometer: algorithmic developments

The self-noise analysis of the microseismometer: algorithmic developments is a crucial aspect of ensuring accurate and reliable seismic data. This section focuses on the development and implementation of algorithms designed to analyze self-noise, which is the internal noise generated by the microseismometer itself. The analysis involves understanding the coherence bandwidth and the coherence of the recorded signals.

The coherence bandwidth is defined as the range of frequencies over which the coherence between two signals remains significant. In this context, the coherence bandwidth is critical for understanding the spectral characteristics of the self-noise.

The coherence formula is given by

\[
\text{Coherence} = \frac{\text{Cross-Spectrum}}{\text{Spectrum of one signal} \times \text{Spectrum of the other signal}}
\]

This formula helps in quantifying the similarity between two signals as a function of frequency. The coherence value ranges from 0 to 1, with 1 indicating perfect correlation.

In the context of self-noise analysis, the coherence is calculated between the self-noise and the ambient noise. The results are then used to estimate the noise level and the alignment error.

The alignment error, \( \Delta \theta \), is introduced to study the impact on the noise determination methods. This error is added to the orientation angles of the seismometers, \( \theta_{NE} \) and \( \theta_Z \), which are then used to calculate the coherence and delta noise.

Ambient signal seen by the test and aligned reference seismometers is given as \( Z(\theta) \) and so \( X = (Z(\theta) + N1) \) and \( Y = (Z(\theta) + N2) \) assuming the transfer-function is accurately removed from each sensor. If the reference seismometer angle contains an error \( \Delta \theta \) then the recorded time series becomes \( Y = Z(\theta + \Delta \theta) + N2 \). The signal \( Z(\theta + \Delta \theta) \) can be expressed as

\[
Z(\theta + \Delta \theta) = Z(\theta) \cos(\Delta \theta) + Z(\theta + 90) \sin(\Delta \theta),
\]

where \( Z(\theta + 90) \) indicates the orthogonal \( Z \) component in the plane. As a result, the aligned component \( Z(\theta) \) is attenuated and an extraneous orthogonal component is included in the noise determination methods. This component is incoherent with that recorded by the test seismometer and will inject into a noise estimate from both the coherence and delta noise methods. As described in Section 6.5.5, this injection will appear as the ambient background signal.

To understand the impact of this error on the noise determination methods, a simulation with an alignment error, \( \Delta \theta \), was performed on the BFO (horizontal), SP3 (horizontal) and SP1 (vertical) datasets. The coherence (6.7) and delta noise (6.11) was calculated between the microseismometer and the synthesised reference axis (6.22) where either the horizontal (NE) or vertical (Z) alignment angles, \( \theta_{NE} \) and \( \theta_Z \), were modified by a range of errors given by \( \Delta \theta_{NE} \) or \( \Delta \theta_Z \). Figures 6.18, 6.20 and 6.22 show the noise determination for each dataset where the alignment error, \( \Delta \theta_{NE} \), was in the horizontal plane. Figures 6.19, 6.21 and 6.23 show the noise determination for the same data but the alignment

Figure 6.16. The alignment angle determination as function of input bandwidth \( f_{bottom} \leq f \leq f_{top} \) for Algorithm 18 for BFO dataset.
error, $\Delta \theta_Z$, was in the vertical plane. The alignment error injects a component into the noise estimate similar to the orthogonal axis that has been added. It can be seen that the error is injected in the highest SNR bandwidths, notably at the microseismic peak. As a result, the error depends on the power in the orthogonal axis. Furthermore, the presence of such a peak in the self-noise estimate indicates that the alignment is incorrect.

Remark 31. Figures 6.18, 6.19, 6.20, 6.21, 6.22 and 6.23 demonstrate the spurious injection into the noise curve for a horizontal and vertical misalignment for different environments and horizontal/vertical seismometers. It is demonstrated that this injection is an orthogonal component between the two seismometers in the context of Figure 6.12. As discussed in Section 6.5.5, this would add background signal into the noise curve. In this case the quiet BFO environment shows the introduction of the microseismic peak. The high noise CNES environment also injects a high frequency component.

Remark 32. An alignment error is modelled by a rotation in the plane of the error. This shows that the noise determination contains an orthogonal component and attenuates the aligned component of the ambient signal. As a result, the noise determination contains a spurious component dependent on the SNR.

6.7.2 Transfer-Function Correction

The analysis of the coherence and delta noise methods has shown that the accuracy of the transfer-function correction impacts the self-noise estimation. In Section 6.2.1, it was
shown that the microseismometer’s velocity response is modelled as the linear system
\[ a(t) = h(t) * b(t), \]
where \( a(t) \) is the recorded velocity output, \( b(t) \) is the seismic signal and \( h(t) \) is the instrument transfer-function. To obtain the seismic component from recorded data, the output, \( a(t) \), must have the transfer-function, \( h(t) \), removed. This problem is stated by (6.1), and is solved through a division in frequency by the estimated response, \( \tilde{H}(s) \), of the form \( H(s) = \frac{|H|}{(s-z_1)(s-z_2)} \) in (6.2). Such an operation is available in commonly used seismology toolboxes such as the seismic analysis code (SAC) [130] [131]. To achieve this, the obtained signal, denoted as the vector \( a \), is first transformed to the frequency-domain through an FFT to give the vector \( A \). This yields the spectrum of the velocity response, \( A(s) = H(s)B(s) \), at the discrete frequency bins given by
\[ s = [0, f_s/N, 2f_s/N, \ldots, f_s/2, f_s/N, \ldots, f_s], \]
where \( f_s \) is the sample rate and \( N \) is the number of samples recorded. This is a two-sided spectrum between 0 and 2\( \pi \) radians. The estimated frequency response, \( \tilde{H}(s) \), in (6.2) is evaluated at each frequency bin of the transformed time-series for the determined inputs \( |H|, z_1, z_2, p_1 \) and \( p_2 \). The one-sided frequency vector
\[ s_{\text{one-side}} = [0, f_s/N, 2f_s/N, \ldots, f_s/2], \]
is used to first obtain \( \tilde{H}(s_{\text{one-side}}) \). The two-sided transfer-function vector is then given as
\[ \tilde{H} = [\tilde{H}(s_{\text{one-side}}), \text{Rev}(\tilde{H}(s_{\text{one-side}}))^*], \]
where the operation \( \text{Rev}(\cdot) \) is defined in (6.21). This ensures that the vector is conjugate symmetric. The frequency-domain velocity response is then corrected as \( \tilde{B} = \frac{A}{\tilde{H}} \) before
Chapter 6. Self-Noise determination of a microseismometer for planetary science

Figure 6.19. Microseismometer self-noise estimation on the QM BFO dataset for a range of alignment errors, $\Delta \theta$, in the vertical plane. The noise determination was calculated using the Welch periodogram averaged over eight Hanning windows with 50% overlap.

the IFFT is performed to yield the corrected time-series vector $\hat{b}$. This is summarised by Algorithm 19.

Algorithm 19. Frequency division transfer-function inversion

1: Initialise: Input signal vector $a \in \mathbb{R}^N$, $f_s$, $p_1, p_2, z_1$ and $z_2$
2: Calculate FFT of $a$ to give $A$
3: Create vector $s_{one-side} = [0, f_s/N, 2f_s/N, \ldots, f_s/2]$
4: Calculate $\tilde{H}(s_{one-side})$
5: Create the vector $\tilde{H} = [\tilde{H}(s_{one-side}), \text{Rev}(\tilde{H}(s_{one-side}))^*]$
6: Perform frequency division to obtain $\hat{B} = \frac{A}{\tilde{H}}$
7: Calculate IFFT of $\hat{B}$ to give the output vector $\hat{b}$

The result for velocity signals is tapered at low frequencies. This is because the theoretical transfer-function tends to zero at low frequencies causing the inversion to be numerically ill-posed, that is, the transfer-function, $H(s)$, at $s = 0$ is equal to zero and so the zeroth frequency of the velocity output would need to be divided by zero. The zeroth frequency is regularised by adding a small value for $H(0)$ and so the output will contain spurious results and the lowest frequencies are not accurately recovered. In Chapter 7 we propose a solution based on sensor fusion.

Transfer-function parameter estimation

The parameters $|H|, z_1, z_2, p_1$ and $p_2$ of the transfer-function $H(s) = \frac{|H|(s-z_1)(s-z_2)}{(s-p_1)(s-p_2)}$ are required to be estimated from acquired data. The zeros are at zero, that is, $z_1 = z_2 = 0$, whereas the poles, $p_1$ and $p_2$, are determined from (6.3) by the damping constant, $D$ and corner-frequency, $f$. This are derived from a calibration.

An experimental calibration [20] [118] is performed by comparing data from the microseismometer’s velocity channel and a reference seismometer (whose response is known
6.7. The Self-Noise analysis of the microseismometer: algorithmic developments

and already corrected through Algorithm 19) and fitting the transfer-function equation (6.2) to the result. The transfer-function estimate between the recorded microseismometer velocity response, \( X \), and the corrected reference seismometer output, \( Y \), is given by

\[
\hat{H}(s) = \frac{P_{XY}(s)}{P_{YY}(s)}.
\] (6.23)

The parameters \(|H|, D \text{ and } f\) are then chosen to produce the estimate, \(\hat{H}(s)\), which best fits the obtained \(\hat{H}(s)\). This can alternatively be seen as providing the parameters that produce an estimated transfer-function, \(\hat{H}(s)\), between the corrected test and reference seismometer velocity response equal to one for all frequencies. This result is shown for the SP3 CNES and QM BFO datasets in Figures 6.24 and 6.25.

The accuracy of the calibration calculated in this way is dependent on the estimation of \(P_{XY}\) and \(P_{YY}\) across the entire bandwidth. As discussed in Section 6.5.1, the seismic background noise is not constant for all frequencies and at lower frequencies (< \(1 \times 10^{-1}\) Hz) the signal power can be below the noise of the sensor. Therefore, the input-output relationship of the sensor transfer-function is not captured by the estimate (6.23) as both \(X\) and \(Y\) signals do not contain a shared signal. This effect is seen in the BFO dataset in Figure 6.25. Furthermore, if there is a high incoherent cultural noise contribution, then the estimation will also be erroneous, as seen at the higher frequencies in Figure 6.24 which show a resonance.

Coil calibration of the transfer-function

In the prequel it was shown that for bandwidths where the coherence between the reference and microseismometer outputs is low, the transfer-function cannot be straightforwardly determined from an ambient experiment. To this end, relative driven methods are employed [20]. The microseismometer is equipped with a calibration coil (the third coil in
Figure 6.21. Microseismometer self-noise estimation on the SP3 CNES dataset for a range of alignment errors, $\Delta \theta$, in the vertical plane. The noise determination was calculated using the Welch periodogram averaged over eight Hanning windows with 50% overlap.

Figure 6.1) which is driven by a known waveform to impart an electromagnetic force onto the proof-mass. The velocity response can then be compared to the input waveform in the same way as the calibration from an ambient signal.

Such a calibration was performed during cruise, in the deep space environment. The calibration waveform was a 200 s duration logarithmically swept sinusoid. The input signal and resultant response are shown in Figure 6.26 for the SP3 sensor. The transfer-function is then estimated through (6.23) where the input $Y$ is the calibration waveform and $X$ the uncorrected microseismometer output and is shown in Figures 6.27 and 6.28 for the SP2 and SP3 sensors. This is shown to match the nominal transfer-function until the coherence becomes low at 0.1 s, owing to the 200 s duration.

**Delta Noise Optimisation**

The coherence method is used to produce the definitive noise estimate from a dataset as it is the most robust. On the other hand, the analysis in Section 6.3 shows that the delta noise estimate is highly susceptible to the accuracy and precision of the transfer-function correction. A corollary of this is that when the coherence and delta noise estimates are close in value, the transfer-function correction has been extremely accurate. As a result, the optimisation of the delta noise to match the coherence noise yields an effective method to obtain a transfer-function estimate.

**Remark 33.** The delta noise estimate can be optimised through selection of the transfer-function parameters to best match the coherence noise. Owing to the intrinsic sensitivity of the delta noise determination, a close match indicates that the obtained transfer-function parameters are highly accurate and precise.

This method is more effective at obtaining the transfer-function parameters than through the fit-based approach to the transfer-function estimate (6.23). As discussed in Section
6.7. The Self-Noise analysis of the microseismometer: algorithmic developments

6.7.2, the transfer-function estimate is poor for bandwidths with little coherence. This is often the case for frequencies below the microseismic peak < 0.1 Hz which is near the cut-off frequency of the microseismometer. Therefore, the transfer-function shape is not well constrained from most datasets. On the other hand, the delta noise method is most difficult to optimise in the presence of a large ambient signal, such as the microseismic peak. The optimisation requires that both the gain and the phase of the corrected microseismometer signal must accurately match and in doing so all the transfer-function parameters are well determined. The method is, however, limited by the build up of contributions of incoherent external sources from cultural noise, described in Section 6.5.1. This means that it is particularly difficult to optimise at high frequencies > 5 Hz. At the frequency around the microseismic peak, the SNR between the ambient and sensor noise is usually above 100. To achieve a 1% error in the determination from the delta noise optimisation, the transfer-function must be correct at the level of 100 ppm which indicates the accuracy and precision than can be obtained through this method.

Nominal transfer-function parameters

The nominal transfer-functions of the flight models are given in Table 6.3 [2], determined from the coil and ambient calibrations. The ambient transfer-function method is useful as it can be obtained for any test data, e.g. on Mars in relation to the VBB sensor. It is, however, limited by the available power of the ambient signal within the bandwidth. The calibration from driving the coil provides a clean signal across the bandwidth to obtain the sensor response. This is instead limited by the duration of the calibration profile and does not provide an absolute determination to ground acceleration. The delta noise optimisation approach, however, provides an extremely accurate method from which to determine the transfer-function from ambient test data. Owing to the method’s sensitivity to gain and angle, all transfer-function parameters can be estimated in the coherent microseismic
Figure 6.23. Microseismometer self-noise estimation on the SP1 CNES dataset for a range of alignment errors, $\Delta \theta$, in the vertical plane. The noise determination was calculated using the Welch periodogram averaged over eight Hanning windows with 50% overlap.

Table 6.3. Flight model nominal transfer-functions

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Gain</th>
<th>Damping</th>
<th>Corner-frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP1</td>
<td>27600</td>
<td>1.1</td>
<td>1/35.2</td>
</tr>
<tr>
<td>SP2</td>
<td>28100</td>
<td>1.1</td>
<td>1/34.9</td>
</tr>
<tr>
<td>SP3</td>
<td>24500</td>
<td>1.09</td>
<td>1/35.3</td>
</tr>
</tbody>
</table>

bandwidth. This does require some prior knowledge of the transfer-function form as this is not given by the delta noise optimisation.

Absolute calibration methods have been further explored for the InSight mission. The SEIS sensors are mounted on top of a levelling (LVL) system which can actuate its legs to create a signal for the absolute calibration at low frequencies for the VBB sensor [2] [132]. In addition, the datasets acquired during cruise phase observed the deadband attitude control thrusters firing. This imparted a velocity step onto the sensors, however, the step is not ideal and so a step-function calibration could not be accurately extracted.

6.7.3 Spectrogram and ranked noise estimation

The spectrogram is a useful tool to examine how the frequency content of a signal changes over time. It is calculated using the short time Fourier transform (STFT). The original signal is divided into windows of length $T_{\text{interval}}$, which may overlap if desired. The DFT is computed on each window separately and stored. The spectrogram is created by plotting the consecutive DFTs over time. As a result, it is possible to see tones and chirps clearly along with local features. There is a trade-off between the frequency and time resolution as using a small time window means only a few frequency bins are calculated and a long time window will not be able to resolve short-time events. In this thesis the DFT is calculated using the Welch periodogram [60].

A spectrogram of the sensor self-noise can also be computed. This is calculated in
6.7. The Self-Noise analysis of the microseismometer: algorithmic developments

Figure 6.24. Transfer-function between corrected reference and corrected microseismometer signals for SP3 CNES dataset.

Figure 6.25. Transfer-function between corrected reference and corrected microseismometer signals for QM BFO dataset.
Chapter 6. Self-Noise determination of a microseismometer for planetary science

Figure 6.26. Microseismometer calibration waveform and corrected response in velocity.

Figure 6.27. The transfer-function estimate to the calibration waveform for SP2.
the same way as the standard self-noise determination but on each individual STFT time window. As a result, time-frequency features can be identified that belong to the microseismometer noise. Moreover, quiet periods in the observation can straightforwardly be identified which are desired for self-noise determination.

In general, a sensor’s measured output signal power will not be below the instrument’s intrinsic noise floor. As a result, it is appropriate to search for the lowest power self-noise estimate obtained from an observation to use for a definitive noise curve. This can be achieved using the self-noise STFT. The spectrogram windows are ranked in terms of average power within a certain bandwidth, usually between $0.1-1$ Hz which is the key seismic bandwidth. The autoregressive average of the $n$ lowest noise power rPSDs, denoted as $b[i]$ ranked lowest to highest for $i = 1, 2, \ldots, n$, is then calculated as

$$a[i + 1] = \frac{(ia[i] + b[i])}{(i + 1)},$$

where $a[0] = 0$. Examining $a[i]$ for $i = 1, 2, \ldots, n$ gives an indication of the observation’s statistics and reduces the estimation’s variance. This is akin to the averaging in the Welch periodogram method except instead of consecutive windows, the ordered quietest windows are used.

Remark 34. Calculating the self-noise estimate on sliding time windows within a dataset (i.e. the STFT) and searching for the lowest power spectra obtained is an effective technique to produce a self-noise determination of an instrument.
6.8 The self-noise analysis of the microseismometer: experimental results

The goal here is to validate the noise model of the microseismometer (in Section 6.5.2) from experimental data and so understand the performance limits of the FM sensors on the InSight mission. This is achieved using the algorithms introduced in Section 6.7. In Section 6.5, it was shown that the noise determination is more difficult in some bandwidths due to the frequency dependence of the SNR and, significantly, cultural noise to the seismometer noise. As a result, test data acquired from several different observatories on different sensor units are used to explore the full bandwidth of the sensor’s operation, described in Section 6.6. The best terrestrial noise achieved from the FM sensors is given by the CNES tests. The BFO dataset was recorded at a quiet site with minimal cultural noise and so the high frequency capability of the instrument can be examined before the test at Oxford is used to show the low frequency noise floor of the sensor. Finally, data acquired from the horizontal FM sensors during the cruise phase of the InSight mission are analysed which gives the performance in the extremely quiet, deep space environment.

In Section 6.3 the coherence and delta noise method were introduced as techniques to determine the self-noise of a test seismometer. The established coherence estimate is more robust and so provides the baseline self-noise determination. On the other hand, the delta noise is used to produce an optimised transfer-function between the microseismometer and the reference, as described in Section 6.7.2, and so the self-noise can be well determined. Prior to this, the alignment Algorithm 18 and transfer-function correction Algorithm 19 are applied to the test data. The best sensor self-noise is presented from the ranked-noise, averaging estimate described in Section 6.7.3. The process to obtain a self-noise estimate of the microseismometer in a two-sensor noise test is summarised in Algorithm 20.

Algorithm 20. The process to obtain a self-noise estimate of the microseismometer

1: Pre-processing 1: Create a synthetic reference axis aligned to the microseismometer using Algorithm 18.
2: Pre-processing 2: Remove the nominal transfer-function function from the microseismometer and reference seismometer using Algorithm 19.
3: Self-noise determination: Employ the coherence and delta noise methods to obtain a self noise estimate of the microseismometer.
4: Refine and optimise: Perform the delta noise optimisation to refine the transfer-function parameters for that experiment and compute the spectrogram and ranked-noise of the microseismometer to identify the quietest period for the final self-noise determination.

6.8.1 Site noise floor

In Section 6.5.4 it was shown that the equations (6.15) and (6.16) can be used to predict the limits of a self-noise determination for the site SNR based upon the NLNM/NHNM. These equations require the inputs:

1. The ambient signal $Z$. 
2. The microseismometer noise $N_1$,

3. The reference seismometer noise $N_2$,

4. The transfer-function correction ratio $R_X$.

Consider the test topology shown in Figure 6.29, where there are two reference seismometers. In this topology it is possible to derive an estimate for the above components and so predict the self-noise determination. This allows the examination of the limitations at a certain test site. The microseismometer noise is given from the noise model (6.17). The ambient signal is approximated by the first reference sensor and so the reference noise, $N_2$, can be provided by a coherence noise estimation between it and the second reference seismometer. Note that this reference noise will also contain some of the cultural injections described in Section 6.5.1. The transfer-function correction ratio can be estimated by (6.23) or from a candidate error size.

Figures 6.30 and 6.31 show the expected bias as a ratio in the self-noise determination for the SP3 CNES and BFO datasets respectively. Each shows the predicted error for the delta noise and coherence noise for the estimated transfer-function (shown in Figures 6.24 and 6.25) and for a candidate correction ratio of $R_X = 1.1$. For the SP3 CNES dataset, the predicted error for the coherence method with the estimated $R_X$ is close to the ratio $\frac{P_{N_2N_2}}{P_{N_1N_1}}$ across the entire bandwidth of this experiment although it has a higher variance at low frequency. At high frequencies the predicted error is high indicating the presence of incoherent cultural noise. This indicates that the reference noise provides the limits for the site. For the QM BFO estimate, however, the predicted error of the coherence method with the estimated $R_X$ deviates at high frequencies. This is because the ambient signal is lower than the microseismometer noise model and so there is a loss of coherence and the transfer-function estimation is not accurate, as is also the case at low frequencies. This is confirmed from the predicted error for $R_X = 1.1$ as this is low at high frequencies, whereas for the SP3 CNES dataset this was also high.

Remark 35. The analysis of the noise testing methods in Section 6.4 and 6.5.1 can be used to predict the expected bias for a self-noise determination at a specific experiment with the topology in Figure 6.29. It has been shown that the tests performed at CNES will not be able to provide a good self-noise determination at high frequency owing to cultural
Chapter 6. Self-Noise determination of a microseismometer for planetary science

Figure 6.30. Prediction of the bias of self-noise determination in SP3 CNES dataset.

Figure 6.31. Prediction of the bias of self-noise determination in QM BFO dataset.

noise but are adequate in the microseismic bandwidth. On the other hand, the BFO site does not have such a component and can be used to achieve a good self-noise determination at high frequencies in order to prove the microseismometer noise model.

6.8.2 Self-noise determination of the flight model units

The noise testing process in Algorithm 20 was applied to the data acquired from the flight model units during testing at CNES. The self-noise determination for a quiet period of the data set is shown in Figure 6.32 for the vertical SP1 sensor, Figure 6.35 for the horizontal SP2 sensor and Figure 6.38 horizontal SP3 sensor. In each of these the noise estimate rises at around 2 Hz, above the requirements for the mission, for both the microseismometer and the STS2 reference. This indicates that the CNES site has a high incoherent cultural noise contribution.

The spectrogram for each of the three tests for SP1, SP2 and SP3 are shown in Figures 6.33, 6.36 and 6.39 respectively. For the SP1 dataset, there were no significant features. However, for the horizontal, SP2 and SP3 test there were periodic spikes visible in the time series and spectrogram. These were due to the goniometer mount shifting abruptly. Such features are avoided for the computation of the sensor self-noise. The ranked-noise for each sensor is shown in Figures 6.34, 6.37 and 6.39.

6.8.3 High frequency noise: QM BFO

The data acquired from the black forest observatory contained a very low ambient signal, containing a minimal cultural contribution and only the microseism. This is seen in Figure 6.41 which contains an rPSD of the reference and microseismometer signals. Compared to the CNES tests (such as Figure 6.32) the high frequencies are much lower, in fact, above
6.8. The self-noise analysis of the microseismometer: experimental results

![Noise determination for SP1](image)

Figure 6.32. SP1 CNES dataset: The microseismometer rPSD, reference rPSD, reference noise along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%.

20 Hz the microseismometer noise dominates the ambient signal. Therefore, the obtained self-noise estimate validates the microseismometer noise model at high frequencies. The spectrogram and ranked noise is shown in Figures 6.42 and 6.43. This confirms that the site is extremely quiet compared to the CNES tests.

6.8.4 Low frequency noise: QM Vert 2017 Oxford

The QM Vert 2017 dataset is taken over 49 hours and so allowed the noise to be examined out to $1 \times 10^{-5}$ Hz, at a tidal range. The self-noise determination is shown in Figure 6.44. Note that the delta noise optimisation has not been as successful at removing the microseismic peak from the noise estimate compared to the FM datasets. This is because the optimisation was performed over a much longer time period and the transfer-function parameters can be affected by environmental sources such as temperature. This highlights a general problem for self-noise determination at low frequencies as the environment must be kept stable for a long period.

6.8.5 Deep space observatory

The two horizontal microseismometer units were operated during the cruise phase of the InSight mission. As such, data was acquired from a deep space observatory, the quietest location in the Solar System. The ambient signal is therefore negligible and so the sensor should record only its inherent self-noise and spacecraft/lander excitation. The mission nominally downlinks “continuous” data at a sample rate of $f_s = 2$ Hz and, following a request, “event” data at a rate up to $f_s = 100$ Hz. The two space datasets described in Section 6.6 (SP2/3 Space event and long period) provide this high and low rate data. As a result, the microseismometer performance across the entire bandwidth can be validated without the need for much further processing.
Figure 6.33. SP1 CNES dataset: Spectrograms of the microseismometer signal, reference noise and microseismometer noise. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometer, reference and the delta noise is low-pass filtered with a corner frequency at 1 Hz.
6.8. The self-noise analysis of the microseismometer: experimental results

Figure 6.34. SP1 CNES dataset: The ranked-noise estimate (6.24) for \( i = 1, 2, \ldots, 32 \) of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth 0.01 – 1 Hz. Calculated from the 800s period STFT (400s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%.

Figure 6.35. SP2 CNES dataset: The microseismometer rPSD, reference rPSD, the reference noise along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%.
Figure 6.36. SP2 CNES dataset: Spectrograms of the microseismometer signal, reference noise and microseismometer noise. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometer, reference and the delta noise is low-pass filtered with a corner frequency at 1 Hz.
6.8. The self-noise analysis of the microseismometer: experimental results

Figure 6.37. SP2 CNES dataset: The ranked-noise estimate (6.24) for $i = 1, 2, \ldots, 32$ of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth $0.01 - 1$ Hz. Calculated from the 800s period STFT (400s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%.

Figure 6.38. SP3 CNES dataset: The microseismometer rPSD, reference rPSD, the reference noise along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%.
Chapter 6. Self-Noise determination of a microseismometer for planetary science

Figure 6.39. SP3 CNES dataset: Spectrograms of the microseismometer signal, reference noise and microseismometer noise. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometer, reference and the delta noise is low-pass filtered with a corner frequency at 1 Hz.
6.8. The self-noise analysis of the microseismometer: experimental results

Figure 6.40. SP3 CNES dataset: The ranked-noise estimate (6.24) for $i = 1, 2, \ldots, 32$ of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth $0.01 - 1$ Hz. Calculated from the 800s period STFT (400s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%.

Figure 6.41. QM BFO dataset: The microseismometer rPSD, reference rPSD, the reference noise along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%.
Figure 6.42. QM BFO dataset: Spectrograms of the microseismometer signal, reference noise and microseismometer noise. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometer, reference and the delta noise is low-pass filtered with a corner frequency at 1 Hz.
6.8. The self-noise analysis of the microseismometer: experimental results

Figure 6.43. QM BFO dataset: The ranked-noise estimate (6.24) for $i = 1, 2, \ldots, 32$ of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth $0.01 - 1$ Hz. Calculated from the 800s period STFT (400s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%.

Figure 6.44. QM Vert 2017 dataset: The microseismometer rPSD, reference rPSD along with the coherence and delta self-noise determinations for the microseismometer over a quiet period. All calculated using the Welch method averaging over two Hanning windows overlapping by 50%.
Figure 6.45. SP2/3 Space event dataset: Spectrograms of the microseismometer signals. Calculated for 100s period STFT at 10s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometers is low-pass filtered with a corner frequency at 1 Hz.
6.8. The self-noise analysis of the microseismometer: experimental results

Figure 6.46. SP2/3 Space event dataset: The ranked-noise estimate (6.24) for \( i = 1, 2, \ldots, 32 \) of the sorted microseismometer and reference coherence noise rPSDs according to the lowest average value in the bandwidth 0.01 – 1 Hz. Calculated from the 200s period STFT (20s overlap) using the Welch method averaging over two Hanning windows overlapping by 50%.

Figure 6.47. SP2/3 Space long period dataset: self-noise rPSD, calculated using the Welch method averaging over two Hanning windows overlapping by 50%, of each microseismometer signal.

Figure 6.46 shows the ranked acceleration rPSD of the outputs from the FM sensors for the SP2/3 Space event dataset. This result is similar to the BFO result, confirming the lack of ambient signal. Figure 6.45 shows the spectrogram and time series of the SP2/3 Space event dataset. This confirms that the ambient signal is much quieter compared to the terrestrial experiments. There are some excess terms present, above the noise model, which appear as constant modes in the Spectrogram. Notice that the noise on the SP2 sensor rises at low frequency earlier than the SP3 sensor, that is, the low frequency power law component in (6.17) is larger for this sensor.

The spectrogram and time series of the SP2/3 Space long period dataset is shown in Figure 6.48. This shows that there are several impulses throughout the acquisition. These are thruster impulses from the deadband attitude control of the spacecraft. Such transient events corrupt a self-noise determination as the spike injects a frequency contribution across the entire bandwidth. Moreover, a coil calibration is present at the beginning of the data. To this end, the first hour of the data was removed and a period of 20s before and 60s after each thruster spike was removed with the samples replaced by a linear interpolation between the two points. This dataset contains a significant low frequency trend. This was removed through a fitted function as it is not due to the inherent sensor noise but
Chapter 6. Self-Noise determination of a microseismometer for planetary science

Figure 6.48. SP2/3 Space long period dataset: Spectrograms of the microseismometer signals. Calculated for 200s period STFT at 100s intervals using the Welch method averaging over two Hanning windows overlapping by 50%. The time series of the microseismometers is low-pass filtered with a corner frequency at 0.5 Hz.
from a start up transient, temperature variation and the fact that the transfer-function correction Algorithm 19 is ill-posed for low frequency. The resultant noise estimate rPSD of the pre-processed data is then shown in Figure 6.47.

6.9 Conclusions: Summary of microseismometer self-noise analysis

The industry standard two sensor method for a self-noise determination of a seismometer is the coherence method introduced by Holcomb [109] described in Section 6.3.2. A further useful tool, the delta noise method, was proposed in Section 6.3.3 and models the difference between a test and reference seismometer outputs as the self-noise. The analysis of both techniques in Section 6.4 showed that the coherence method is more robust and should be used for a definitive self-noise determination, whereas the delta noise is susceptible to errors in the transfer-function. This was confirmed through the expressions (6.15) and (6.16) which indicate that the SNR of the experiment does not greatly affect the coherence method while the delta noise method’s performance degenerates significantly. The generation of a self-noise time series, however, is highly useful for further sensor analysis.

In Section 6.5, the implications of a practical experiment were examined. The theoretical SNR of a test site was determined through consideration of the microseismometer noise model and the NLNM/NHNM to give an expected error of the two self-noise determination methods. This SNR was input into the derived self-noise estimator bias equations (6.15) and (6.16) to predict the likely error. Furthermore, the impact of cultural noise was considered. It was shown that such sources can couple to the sensor either dynamically through ground motion, as a non-linear/orthogonal component or directly to the seismometer via another physical mode. These sources can be injected to the self-noise estimate. This was confirmed through an analysis of the test sites in Section 6.8.1. The self-noise estimator bias equations (6.15) and (6.16) were used with the SNR and reference noise predicted from two datasets with low and high cultural noise to show how this limits the self-noise determination.

Remark 36. The coherence method is robust to high SNR. However, high cultural noise causes an incoherent injection. The lack of coupling at these frequencies is contributed to by sensed non-ground acceleration signals, the physical resonance of platforms and mounts and small changes in the transfer-function parameters due to the environment.

Several algorithmic developments were introduced in Section 6.7 for effective pre-processing, calibration and post-processing. An algorithm to calculate the geometric alignment between the test and reference signals was introduced. The affect of an angle error was shown to introduce an orthogonal component to the self-noise calculation and so inject an error dependent on the ambient conditions. The frequency-division Algorithm 19 was introduced to correct the instrument transfer-function prior to the self-noise determination. The gain, damping constant and corner-frequency of the model (6.2) for the microseismometer must be estimated to input into the algorithm. It was shown that
these parameters can be calculated from the transfer-function estimate (6.23) to reference seismometer in ambient background or from a driven coil calibration. The ambient test is limited to regions where there is a strong coherent signal and so is difficult at low and high frequencies. On the other hand, the coil calibration is limited by the length of the signal and is not absolute. The delta noise optimisation, however, provides a robust method to obtain accurate and precise transfer-function parameters as it utilises the inherent susceptibility of the delta noise method to these parameters. The self-noise estimate is then robustly determined using the ranked-noise method. This calculates an autoregressive average of the sorted self-noise rPSDs obtained from the windows in the STFT and so provides a Welch type averaging without injecting noise from transient features. These developments are summarised in Algorithm 20 which outlines the process and refinement to obtain an accurate self-noise determination.

Section 6.8 then provides the self-noise determination of the InSight flight and qualification units to display the full range of the sensor’s operation. Figure 6.49 shows a summary of the noise testing performed on Earth. It is desired to establish both each FM unit’s performance and the potential limits of the instrument. The results from the BFO dataset show performance to match the instrument noise model at high frequencies. On the other hand, the FM units are shown to match the quiet, low high frequency noise, BFO result until 2 Hz where the self-noise rises above the InSight requirements. This signifies the injection of cultural signals into the self-noise determination. Although the coherence estimate does not degrade at high SNR, significant cultural noise contributions do not couple or incoherently couples between the seismometers. Furthermore, these signals may not be recorded ground accelerations but could couple from other mechanisms, e.g. acoustics [118]. On top of this coupling there is also a resonance peak associated with the system mounting. Moreover, the QM Vert 2017 dataset provides an improved self-noise determination at low frequencies as the environment was stable for a long period. Therefore, the tests on QM units in different scenarios provides validation of the instrument performance across the full bandwidth. These aspects of self-noise determination at high and low frequency mean that, combined with the results of the tests performed directly on the FM units, the performance of the InSight missions microseismometers is validated on Earth prior to the mission launch.

Remark 37. The self-noise determination at low frequencies for a seismometer is difficult due to the presence of environmental noise components. These affect each sensor differently and so are incoherent. Moreover, the test must be environmentally stable over a long period, avoiding transient features.

Figures 6.50 and 6.51 show a summary of the results from the deep space observatory for the SP2 and SP3 sensors compared to their respective CNES FM test. At low frequency the result is much better than obtained for any sensor on Earth. This is because the terrestrial experiment is subject to a more variable environmental contribution (e.g. temperature and pressure) which may be incoherent between two sensors. The higher frequencies are similar to those seen on the QM sensor in the BFO test. Notice that the sensor noise floor is slightly worse than for the CNES tests. This could be to do with the affect of radiation on
6.9. Conclusions: Summary of microseismometer self-noise analysis

Figure 6.49. A summary of the microseismometer self-noise determinations on terrestrial datasets.

Figure 6.50. A summary of the self-noise determination from the SP2 FM sensor in cruise.

Figure 6.51. A summary of the self-noise determination from the SP3 FM sensor in cruise.
the electronics, sensor degradation during the launch or from an unknown environmental contribution. This component is injected as the low frequency power law in the noise model. Compared to the terrestrial model, the power law exponent is smaller in the deep space environment. The sensors performance is similar aside from the low frequency power law component which is lower for SP3 compared to SP2, owing to manufacturing differences. The green line represents the performance requirements for the InSight mission which are satisfied for both horizontal sensors. It is also expected that the vertical SP1 sensor will also meet its requirements for the full bandwidth based on the terrestrial tests.

In this chapter, the problem of self-noise determination of a seismometer has been addressed. Analysis of the coherence method and proposed delta noise method on top of algorithmic developments for alignment, transfer-function parameter estimation and the ranked-noise estimator have led to a process to achieve robust results and understand the limits for a given site dataset. As a result, it has been possible to validate the performance of the NASA InSight mission’s microseismometers. These results are currently being applied for an application of micrometeoroid and space debris impact detection.

**Remark 38.** The self-noise determinations from the experimental data have validated the microseismometer’s noise model and, hence, shown that the performance requirements for the InSight mission are satisfied.

### 6.9.1 Future work - noise testing development

The analysis of the self-noise determination methods and its implications for a practical estimation (outlined in Sections 6.4 and 6.5.4) can be directly applied to a specific experiment. This was shown in Section 6.8.1 in order to understand the limitations of a particular test setup due to incoherent sources. A key target for future research is then to develop a method for separating the excess, latent, incoherent noise sources (introduced in Section 6.5.1) from the sensor self-noise and so obtain an improved self-noise determination. To this end, consider again that the equations (6.9) and (6.12) provide a method to predict the coherence and delta noise estimates from the input signals:

1. The ambient source $Z$,
2. The test seismometer noise $N_1$,
3. The reference seismometer noise $N_2$.

The ambient source for certain site can be modelled as the output of the reference seismometer. Moreover, the noise processes $N_1$ and $N_2$ can be modelled by surrogates derived from the seismometer noise models. In this way, the microseismometer noise floor is viewed as a prior for the estimation. The comparison of the self-noise determination obtained through both coherence and delta noise methods to their predictions from the priors yields a method to (i) show if it is likely that the test seismometer’s self-noise is that given by the model, (ii) optimise the transfer-function parameters and (iii) identify and model individual noise processes.

One such method to achieve this would be through a probabilistic approach using Gaussian Processes [41]. The sensor noise models are then statistical priors and can be
used to solve Bayesian estimation problems, which would ascertain the “likely” noise model for the microseismometer from even a high ambient noise environment. Obtaining a GP model for the seismometer self-noise and excess noise contributions then offers an in-depth site analysis, which can also be used for science activities. This can be conjoined with the physical perspective of each latent variable term, outlined in Section 6.5.1. In each case a derivation of the physical model is required to decorrelate the process from the signal. Combining this with the GP approach could provide a greater range of modelling ability.
Chapter 7

Full-Band Signal Extraction: Temperature Decorrelation

An experiment is a question which science poses to Nature, and a measurement is the recording of Nature’s answer.

Max Planck

7.1 Introduction

The microseismometer’s performance has been demonstrated to meet the self-noise requirements, through the processing of specific experiments. However, it was highlighted that latent cultural contributions to the ambient signal can inject an excess term into the estimate. Moreover, for a science application, these incoherent terms may obscure the seismic component that is desired for study. In Section 6.5.1 the different physical mechanisms for which these contributions are coupled to the microseismometer was described. Although the anthropogenic sources will not be present on Mars, the difficulties posed by the environment will be much harsher. In order to remove aseismic dynamic contributions a physical model of the coupling to ground acceleration and a measurement of the input source is required e.g. pressure [8]. On the other hand, non-dynamic contributions are sensor specific and can either be physically attenuated or removed through evaluating a model of the microseismometer response for a measured signal. This chapter aims to develop the necessary model and required processing to cater for the latent variable of temperature for the microseismometer.

Temperature is known to be a major cause of drift in seismometers [20], however it is more prevalent for the InSight mission. The range of deployment capabilities on Mars are limited due to the constrained power and mass. The sensor will be deployed directly onto the surface of Mars without the vault typical for terrestrial applications. Power constraints and the large Mars ambient diurnal temperature range prevent active thermal control (e.g. [133] [134]). To this end, a wind and thermal shield (WTS) will be placed on top of the
seismometer to attenuate the wind, pressure and thermal contribution to the signal [7] [9] [8]. Further attenuation is provided by the packaging of the microseismometer. Within the sensor, a silicon/solder bilayer has been incorporated into the suspension to counteract the proportional change in acceleration due to temperature [123]. The aim of the techniques presented in this paper is to provide a “virtual vault” to mitigate the limitations of these passive methods. We have developed a two-step general approach that helps solve for an accurate instrument response characterisation followed by the decorrelation of a temperature dependent component from the microseismometer outputs using a physics-based model.

The first step is to determine the acceleration response of the microseismometer, shown for the the mass position and velocity channels in Figure 6.2. This is the problem of transfer-function inversion, that is, converting the electronic velocity and mass position outputs of the instrument to physical units. This is a common task for seismic instruments and such functionality is implemented in the commonly used Seismic Analysis Code (SAC) [130]. This is performed by the frequency division algorithm in Section 6.7.2. This is known to be numerically ill-posed, that is, the velocity transfer-function, $H(s)$, at $s = 0$ is equal to zero and so the zeroth frequency of the velocity output would need to be divided by zero [110] [135]. Therefore, the zeroth frequency is regularised by adding a small value for $H(0)$ and so the output will contain spurious results as the lowest frequencies are not accurately recovered. Moreover, the mass position transfer-function tends to zero at high frequency. To counteract this, seismologists will usually taper the result for each channel to obtain a corrected response in the bandwidth they are interested in [131]. However, to characterise the response of the microseismometer it is desirable to obtain the instrument’s acceleration over the full available bandwidth. To this end, we propose a sensor fusion based method to combine the mass position and velocity outputs using a Kalman Filter [136]. This enables the complementary properties of the two outputs to be synthesised, producing an acceleration signal response for the microseismometer across the entire signal bandwidth. The Kalman Filter is often used in such sensor fusion applications [137] [138] but, to our knowledge, such an approach has not been used before for the outputs of a seismometer.

Next, a thermal model coupling the measured temperature signals to the acceleration response of the microseismometer is developed using the corrected signal. We incorporate the model into the Kalman Filter approach to give an improved estimation of the outputs of the sensor. This, in turn, leads to an algorithm for temperature decorrelation from the microseismometer’s output. The result yields the purely seismic component measured by the microseismometer, which can then be used for the required science goals of the mission [3]. Our developed approach is physics-based, on the low-level response of the sensor across the full bandwidth.

This full-band decorrelation approach involves all the available information and is therefore limited by the physics model rather than the priors or linearisation used in other approaches [136]. The resulting output is valid over the full bandwidth, without processing artefacts, which is of particular value where signals of interest are not always known before deployment. For example, icy worlds like Europa present a scientific challenge where the
distribution of information across the frequency spectrum is not well constrained [139].
Through applying this method to experimental data, we validate our approach and are able to quantify:

1. the contribution of our passive bilayer thermoelastic compensation,
2. the noise floor of the MEMS microseismometer out to tidal frequencies,
3. the sensor performance down to $-65^\circ C$.

The motivation for this chapter is given from the previous discussion of the transfer-function correction in Sections 6.2 and 6.2.1 and the incoherent sources in Section 6.5.1. It has been shown that the method in Algorithm 19 is ill-posed for the microseismometer velocity and mass position transfer-functions but the bandwidths in which each is prevalent is complementary. In this chapter, the remaining theory for the thermal model and sensor fusion method are first introduced. A method to obtain an accurate, transfer-function corrected, acceleration signal is then presented. This technique is further developed in order to determine the temperature response of the sensor. We then provide experimental results for the proposed transfer-function correction and show the performance increase available from the temperature decorrelation. The significance of the temperature decorrelation method is demonstrated through the observation of the Earth tide and the validation of the thermal compensation included in the microseismometer. We then provide an analysis of a “cold test” to prove the sensor performance in dynamic, inhospitable environments.

### 7.2 The microseismometer thermal model

It has been shown that a temperature input to the microseismometer produces a proportional acceleration response owing to the change in the Young’s modulus of the silicon cantilever [123]. Moreover, the sensor can tilt relative to its enclosure due to the thermal response of the mounting.

The microseismometer is mounted inside an enclosure as shown in Figure 6.1. This housing is designed to isolate the sensor from external temperatures, that is, an externally applied temperature is not coupled directly to the sensor die. This is described by the heat equation expressed through the electrical analogy for heat transfer

$$T_{\text{die}} = T_{\text{enc}} - RC \frac{dT_{\text{die}}}{dt},$$

(7.1)

where $T_{\text{die}}$ is the temperature of the sensor die, $T_{\text{enc}}$ is the temperature at the enclosure and $RC$ is the “time constant” of the microseismometer mounting.

The resultant temperature difference between the circuit nodes causes a heterogeneous expansion in the materials, thus tilting the sensor. As a result, the added tilt acceleration is modelled as being proportional to the difference between $T_{\text{die}}$ and $T_{\text{enc}}$. The thermal response model of the sensor is then

$$a_{\mu ST} = \alpha_{\text{die}} T_{\text{die}} + \alpha_{\Delta T},$$

(7.2)
where $a_{\mu ST}$ is the acceleration response of the microseismometer due to temperature, $\alpha_{\text{die}}$ is the directly proportional sensitivity to the die temperature $T_{\text{die}}$, $\alpha_\Delta$ is the sensitivity to the mounting temperature difference $\Delta T = T_{\text{die}} - T_{\text{enc}}$.

### 7.3 The Kalman Filter

Many physical systems are described by the state-space equation

$$x_k = Fx_{k-1} + Bu_k + w_k,$$

(7.3)

where $x_k$ is the state vector containing the value of interest for the system at time instant $k$, the vector $u_k$ are the inputs to the system related to the state vector through the matrix $B$, $w_k$ is an additive noise term and the matrix $F$ describes the forward evolution of the state vector from the previous time instant $k-1$. There is sometimes uncertainty in the accuracy of the forward model either through missing physics or noisy inputs. This is accounted for by the noise vector $w_k$ which has the covariance matrix $Q$. Such a system can also be measured through the equation

$$z_k = Hx_k + v_k,$$

(7.4)

where $z_k$ is the vector of measurements and the matrix $H$ defines the relation between the measurements and the state vector, $x_k$. The vector $v_k$ is an additive noise term which accounts for uncertainty in the measurement equation and has covariance matrix $R$. Owing to the uncertainty in each equation, the state vector $x_k$ from the forward model (7.3) may differ to that derived from the measurement equation (7.4). The covariance matrix of the error between the state vector and its estimate is denoted as $P_{k|k-1} = E[(\hat{x}_k - x_k)|k-1]$ for the a priori prediction from the previous time step $\hat{x}_{k|k-1}$ and $P_{k|k} = E[(\hat{x}_k - \hat{x}_k|k)(\hat{x}_k - \hat{x}_k|k)^T]$ for the a posteriori estimate $\hat{x}_k|k$.

The Kalman filter, summarised in Algorithm 21, solves this problem to produce an optimal estimate of the state of a system described by two such equations. This is achieved through the Kalman gain, $K$, which mixes the predictions derived from the measurements and the forward model. The mixture depends on the level of uncertainty of each model, described through the noise covariance matrices $Q$ and $R$. As a result, the Kalman gain calculates how much to “trust” either the measurement or forward model. The combination is then given as the optimal prediction of the state vector $x_k$.

**Algorithm 21. The Kalman filter**

1. **Initialise:** $Q = E[ww^T]$ and $R = E[vv^T]$
2. **State prediction:** $x_{k|k-1} = Fx_{k-1|k-1} + Bu_k$
3. **Error covariance prediction:** $P_{k|k-1} = FP_{k-1|k-1}F^T + Q$
4. **Kalman gain:** $K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1}$
5. **Measurement prediction:** $x_{k|k} = x_{k|k-1} + K_k(z_k - Hx_{k|k-1})$
6. **Error covariance:** $P_{k|k} = P_{k|k-1} + K_kHP_{k|k-1}$
Kalman filter derivation

The estimated state is calculated at each time instant, $k$, through a prediction stage

$$x_{k|k-1} = Fx_{k-1|k-1} + Bu_k,$$

as in (7.3), where the subscript $k|k-1$ indicates the value at time $k$ predicted from time $k-1$. This is then combined with the estimate given by the measurements as

$$x_{k|k} = x_{k|k-1} + K_k(z_k - Hx_{k|k-1}),$$

(7.5)

where the Kalman gain $K_k$ is calculated to give the appropriate balance.

To derive the Kalman gain we first consider the covariance matrix of the state error which is given as

$$P_{k|k} = E[(x_k - x_{k|k})(x_k - x_{k|k})^T],$$

(7.6)

where $x_k$ is the true state vector at time $k$ and $x_{k|k}$ is the estimated value at time $k$. Substituting in (7.5) and (7.4), the error covariance matrix is expanded as

$$P_{k|k} = (I - K_kH)P_{k|k-1}(I - K_kH)^T + K_kRK_k^T.$$

(7.7)

In order to produce an optimal estimate of the state vector the Kalman gain matrix, $K_k$, is calculated so as to minimise the mean square error, that is, minimise the trace of the error covariance matrix. The trace is given

$$\text{Tr}[P_{k|k}] = \text{Tr}[P_{k|k-1}] - 2\text{Tr}[K_kHP_{k|k-1}] + \text{Tr}[K_k(HP_{k|k-1}H^T + R)K_k^T].$$

To minimise the trace it is differentiated with respect to the Kalman gain matrix and set equal to zero.

$$\frac{d\text{Tr}[P_{k|k}]}{dK_k} = -2(HP_{k|k-1})^T + 2K_k(HP_{k|k-1}H^T + R) = 0.$$

Solving this equation we arrive at the solution for the Kalman gain

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1}.$$

(7.8)

Note that the error state covariance matrix can be equivalently represented as

$$P_{k|k} = P_{k|k-1} + K_kHP_{k|k-1}.$$

Therefore, based upon the covariance matrices $Q$ and $R$ of the respective noise vectors $w$ and $v$, the Kalman gain calculates how much to “trust” the measurement or forward model. The combination is then given as the optimal prediction of the state vector $x_k$.

Further detailed derivations can be found in [136] [140].
The Indirect Kalman Filter

The Kalman filter has been introduced here as a method to estimate the state values of a system based on both a physical model and measurement data. The filter can be implemented in an “indirect” form, where the state vector estimates a hidden or error state [141].

In many applications it is known that a particular sensor has an inherent instability or drift and so its data are not suitable as a state value over the full frequency range. Consequently, this indirect Kalman filter topology is frequently used for obtaining high quality measurements from inertial sensors [137] [142].

7.4 Full-Band Sensor Fusion Algorithms

In this section, two “full-band” algorithms are proposed. This term refers to the fact that they aim to provide the required treatment to an instrument response across the entire bandwidth of operation. The first method is for the task transfer-function correction to counteract the described ill-posedness of the frequency division Algorithm 19. This is then extended to include the thermal model derived in Section 7.2.

7.4.1 Sensor fusion for transfer-function correction

In order to obtain an acceleration signal from the microseismometer for all frequencies we synthesise the acceleration response derived from the transfer-function corrected velocity channel and the mass position output, the responses of which are shown in Figure 6.2. This is achieved by implementing a Kalman filter in an indirect topology where the error state is the error in the microseismometer acceleration signal derived from the velocity channel. In this case we view the acceleration signal derived from the velocity output as a stable measurement containing an erroneous low frequency component, whereas the mass position output provides a good low frequency measurement of the acceleration. To this end, we construct the state variable

\[
\mathbf{x}_k = \begin{bmatrix} a_{\mu S,k} \\ a_{\text{err},k} \end{bmatrix}, \quad (7.9)
\]

where \( k \) is the time instant, \( a_{\mu S,k} \) denotes the acceleration response of the microseismometer and \( a_{\text{err},k} \) is the artefact introduced as a result of the ill-posed transfer-function correction of the velocity channel. The state equation (7.3) is then given

\[
\mathbf{F} = \begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad (7.10)
\]

where the input is

\[
\mathbf{u}_k = a_{\text{VEL},k}, \quad (7.11)
\]
with $a_{VEL}$ denoting the acceleration signal derived from the velocity channel. The measurement equation (7.4) is then given

$$z_k = a_{POS,k} \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

(7.12)

where $a_{POS}$ is the acceleration derived from the mass position.

At each time instant, $k$, the prediction of $a_{\mu S,k}$ from the forward model (Step 2 in Algorithm 21) is given as $a_{VEL,k} - a_{err,k-1}$. This result (along with the estimate of $a_{err,k}$) is adjusted by the error to the “measurement”, which is the acceleration derived from the mass position, $a_{POS}$, in Step 5 of Algorithm 21. The level of the adjustment is dictated by the Kalman gain, $K_k$, which is calculated using the noise covariance matrices for the forward model, $Q$, and the measurement, $R$. In this indirect topology the respective noise power assigns the relative level of “trust” to each equation. As such, the covariance matrix choice dictates the mixture of $a_{VEL}$ and $a_{POS}$ in producing the estimate of the state variable $x_k = [a_{\mu S,k}, a_{err,k}]^T$, demonstrated in Section 7.5.2. Therefore, the computation of the state vector is recursive and this implementation operates as a filter.

**Remark 39. The Kalman filter for transfer-function correction (K-TF).** An accurate acceleration response from the microseismometer is obtained by applying the Kalman Filter in Algorithm 21 with chosen noise covariance matrices, $Q$ and $R$, to the system described in (7.9), (7.10), (7.11) and (7.12). This is the Kalman filter for transfer-function correction (K-TF).

### 7.4.2 Full-band temperature decorrelation

The thermal model for the microseismometer acceleration response, (7.2), can be seen as a measurement of the acceleration itself. In fact, the thermal response dominates the signal at low frequencies. This can therefore be added to the Kalman filter model outlined in Section 7.4.1 to provide an improved estimate of the acceleration along with the mass position. The Kalman filter topology now becomes

$$
\begin{bmatrix}
a_{\mu S,k} \\
 a_{err,k}
\end{bmatrix} = \begin{bmatrix} 0 & -1 \\
 0 & 1 \end{bmatrix} \begin{bmatrix} a_{\mu S,k-1} \\
 a_{err,k-1} \end{bmatrix} + \begin{bmatrix} 1 \\
 0 \end{bmatrix} a_{VEL,k} + w_k,
$$

(7.13)

$$
\begin{bmatrix}
T_{die,k} \\
\Delta T_k \\
 a_{POS,k}
\end{bmatrix} = \begin{bmatrix} \frac{a_{die}}{2} & \frac{a_{die}}{2} & \frac{1}{2} \\
 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} a_{\mu S,k} \\
 a_{err,k} \end{bmatrix} + v_k,
$$

(7.14)

where $(\cdot)^+$ denotes the generalised inverse. The measurements now include the die temperature and temperature difference as well as the mass position. The coefficients are given from an initial regression and are scaled by a factor of two to average the information included in both measurements.

**Remark 40. The Kalman filter for acceleration fusion from mass position and temperature (K-TF+T).** The thermal model for the microseismometer acceleration in (7.2) can be incorporated as a measurement into the system outlined by (7.13) and (7.14).
Applying the Kalman filter in Algorithm 21 to this system yields an estimation of the acceleration response, $a_{\mu S}$, from the sensor.

This evolution of the K-TF can be used to provide both a refined estimation of the thermal model parameters in (7.2) and a decorrelation of the thermally induced acceleration from the sensor output. To this end, we introduce Algorithm 22. The estimate of the microseismometer acceleration response, $a_{\mu S}$, is given as the output of the final iteration of the K-TF+T method in Step 4. The parameters for the thermal model (7.2) are calculated through a regression to the delta noise (6.10) to give an estimate of the temperature dependent microseismometer acceleration, $a_{\mu ST}$. The temperature decorrelated acceleration response is then given as $a_{\mu S} - a_{\mu ST}$. This novel procedure allows us to better determine the performance of the microseismometer in relation to its thermal response (including determining the effect of the thermal compensation [123]) and then show the sensor’s intrinsic noise at low frequencies.

**Algorithm 22. Temperature Decorrelation Process**

1. **Correct the sensor transfer-function**: create synthesised mass position and velocity output acceleration signal using the K-TF algorithm.
2. **Calculate the initial thermal model values**: perform a linear regression between the temperature values and the delta noise (6.10) of the K-TF obtained $a_{\mu S}$.
3. **Apply the Kalman filter**: apply the K-TF+T topology using the obtained initial values.
4. **Iterate the Kalman filter**: recalculate the thermal model parameters using the output obtained in the previous step and apply the K-TF+T topology with the updated parameters and iterate until convergence.
5. **Remove the temperature impact from acceleration**: subtract the fitted thermal model from the K-TF+T output obtained on the last iteration to give the corrected acceleration signal.

### 7.4.3 Implementation of the K-TF and K-TF+T methods

In this section we provide MATLAB code and implementation notes for the proposed K-TF and K-TF+T methods introduced in Section 7.4. Each method implements the Kalman filter in Algorithm 21 which requires: (i) a row vector of $N$ samples for the measurements, $z$, and the forward model inputs $u$, (ii) the forward model and measurement equation matrices $F$, $B$ and $H$ and (iii) the noise covariance matrices $Q$ and $R$.

The input to the forward model, $u \in \mathbb{R}^{1 \times N}$, is given by the acceleration response derived from the velocity channel, $a_{VEL}$, for both the K-TF and K-TF+T methods. To obtain the $N$ data samples, the logged data are first scaled by a gain factor and the transfer-function is provisionally corrected using the frequency division method described in Section 6.2.1 which yields the velocity response to be differentiated. For the K-TF method the measurement samples $z \in \mathbb{R}^{1 \times N}$ are given by $a_{POS}$, which is obtained from a scaling of the raw mass position outputs by a gain factor. The frequency response does not need to be corrected as it is flat in the required bandwidth. The K-TF+T method also requires the calibrated temperature signals $T_{die}$ and $\Delta T$ which are concatenated to give the measurement samples data matrix $z = [T_{die}, \Delta T, a_{POS}]^T \in \mathbb{R}^{3 \times N}$. 

In the K-TF method, the matrices $F \in \mathbb{R}^{2 \times 2}$, $B \in \mathbb{R}^{2 \times 1}$, and $H \in \mathbb{R}^{1 \times 2}$ are given in (7.10) and (7.12) whereas for the K-TF+T method, the forward model ($F$ and $B$) is unchanged but the measurement model matrix $H \in \mathbb{R}^{3 \times 2}$ is given in (7.14). The noise covariance matrices $Q = \sigma_{FM}^2 I$ and $R = \sigma_M^2 I$ are modelled as identity matrices scaled by the noise variances, $\sigma_{FM}^2$ and $\sigma_M^2$ which are determined depending on the desired mixture of the two predictions.

Listing 7.1 provides code to provide the inputs for the Kalman filter for the K-TF method. Listing 7.2 shows the updated measurement equation required for the Kalman filter inputs in the K-TF+T method. The output state vector, $x$, consists of the estimate of the microseismometer acceleration response, $a_{\mu S}$, and the correction error, $a_{err}$. In Listing 7.2, the code is given for the least-squares regression to recalculate the thermal model parameters from the K-TF+T output. This code can be used to implement Algorithm 22.

%The derived accelerations are the input and measurements of the Kalman filter
u = aVEL;
z = aPOS;
%The matrices for the Kalman filter in the K-TF configuration
F = [0 -1; 0 1]
H = [1 0]
B = [1; 0]
%Set the noise covariance matrices
Q = eye(2)*sig2_FM
R = eye(1)*sig2_M
[x, K, P] = Kalman_Filter( u, z, H, F, B, Q, R)


%Change the measurement equation for K-TF+T
z = [T_die; Delta_T; a_POS];
H = pinv([alpha_die/2 alpha_Delta/2 1/2; 0 0 0])
R = eye(3)*sig2_M
[x, K, P] = Kalman_Filter( u, z, H, F, B, Q, R)
%Calculate the thermal model parameter updates from the output acceleration
X = [T_die', Delta_T', zeros(N,1)]
alpha_reg = inv(X'*X)*X'*(x(1,:)'-a_REF')
alpha_die = alpha_reg(1)
alpha_Delta = alpha_reg(2)

Listing 7.2. Code to update the measurement equation for the K-TF+T method and calculate the thermal model coefficients.
7.5 Experimental Results

7.5.1 Experimental setup

Experimental data has been obtained from testing in two locations: Oxford University and CNES in Toulouse. The data obtained from Oxford are from a qualification model (QM) vertical sensor and the data obtained from CNES are of vertical and horizontal flight model (FM) sensors that have been flown on the InSight mission. In total five datasets were used to demonstrate the proposed K-TF and K-TF+T algorithms:

- two ambient temperature noise tests performed in a seismically quiet laboratory at Oxford which we refer to as QM Vert 2017 and QM Vert 2016;
- two temperature sensitivity tests in the CNES cleanroom which we refer to as FM Vert and FM Horiz;
- a low temperature test at Oxford in a specially designed “cold test” facility which we refer to as QM Vert Cold.

Table 7.1 summarises the collected data. In all cases the data were decimated or interpolated from their original acquisition rate to a sampling frequency of 10 Hz. Furthermore, all the signals were zero-meaned prior to analysis but no additional detrending was performed.

The equipment and nature of the different test facilities is described in the following. The tests and Oxford and CNES use the same equipment as described in Section 6.6.

Low temperature noise test setup

The cold test of the QM vertical unit was performed in a specialist rig designed to determine the microseismometer’s behaviour down to typical Mars temperatures [143]. The inside of the rig was cooled by a liquid nitrogen system which was then allowed to passively warm. Inside the rig, the vertical microseismometer was installed and recorded in the same way as the ambient temperature tests. A Guralp reference seismometer was placed outside the rig but on a coupled platform to the microseismometer mount in order to record the ambient seismic signal and enable a coherence test (6.7).

Temperature sensitivity test setup

The sensor mounting was attached to a gonionmeter which tilted the vertical microseismometer to simulate Martian gravity. No tilting was required for the testing of the horizontal microseismometer. No reference seismometer was available to allow for a self-noise determination. The sensor assembly was externally heated from $\sim$ 20°C to $\sim$ 30°C and allowed to cool passively.

Temperature data acquisition

The $T_{enc}$ and $T_{die}$ temperature signals were measured respectively by a platinum resistance thermometer (PRT) and a gold resistance thermometer, integrated in the sensor
die itself, and recorded on a 24-bit data logger. The resistance values were converted to temperature and low-pass filtered with an IIR filter designed to remove the contribution of the measurement noise. In each case the filter was designed to have 15dB attenuation between the stop-band frequency (given in Table 7.1) and the pass-band frequency, which was $1.1 \times 10^{-1}$ Hz below. The filters had a pass-band ripple of 1dB and an attenuation of 40dB/decade above the stop-band frequency. This stop-band frequency was determined through inspection of the temperature PSDs in Figure 7.1.
### Table 7.1. Datasets used

<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
<th>Length</th>
<th>Temperature rate (Hz)</th>
<th>Filter Stop-Band frequency (Hz)</th>
<th>Temperature variance</th>
<th>Notes</th>
</tr>
</thead>
</table>
| QM   | Vert     | Oxford   | 49 hours              | 1/30                            | $9 \times 10^{-5}$   | $var(T_{die}) = 1.4 \times 10^{-2}$  
$var(\Delta T) = 9.9 \times 10^{-6}$  
$var(T_{die}) = 2.1 \times 10^{-3}$  
$var(\Delta T) = 1.6E - 6$  
$var(T_{die}) = 462$  
$var(\Delta T) = 9.8E - 2$  
$var(T_{die}) = 11.5$  
$var(\Delta T) = 0.34$  |
|      |          |          |                       |                                 |                      | No solder-silicon bilayer thermal compensation, ambient temperature noise test |
|      |          |          |                       |                                 |                      | No solder-silicon bilayer thermal compensation, ambient temperature noise test |
| QM   | Vert     | Oxford   | 112 hours             | 1/60                            | $2 \times 10^{-5}$   | $var(T_{die}) = 1.4 \times 10^{-2}$  
$var(\Delta T) = 9.9 \times 10^{-6}$  
$var(T_{die}) = 2.1 \times 10^{-3}$  
$var(\Delta T) = 1.6E - 6$  
$var(T_{die}) = 462$  
$var(\Delta T) = 9.8E - 2$  
$var(T_{die}) = 11.5$  
$var(\Delta T) = 0.34$  |
|      |          |          |                       |                                 |                      | No solder-silicon bilayer thermal compensation, ambient temperature noise test |
| QM   | Vert     | Oxford   | 49 hours              | 1/60                            | $2 \times 10^{-4}$   | $var(T_{die}) = 462$  
$var(\Delta T) = 9.8E - 2$  
$var(T_{die}) = 11.5$  
$var(\Delta T) = 0.34$  |
| Cold |          | Thermal  |                       |                                 |                      | Performed in a cold testing rig, no solder-silicon bilayer thermal compensation, low temperature noise test |
|      |          | Vacuum   |                       |                                 |                      | FM signal chain, thermal sensitivity test |
| FM   | Vert     | CNES     | 6.4 hours             | 1/5                             | $7 \times 10^{-3}$   | $var(T_{die}) = 5.3$  
$var(\Delta T) = 9.8E - 2$  
$var(T_{die}) = 11.5$  
$var(\Delta T) = 0.34$  |
|      |          |          |                       |                                 |                      | FM signal chain, thermal sensitivity test |
| FM   | Horiz    | CNES     | 4.4 hours             | 1/5                             | $7 \times 10^{-3}$   | $var(T_{die}) = 11.5$  
$var(\Delta T) = 0.34$  |
|      |          |          |                       |                                 |                      | FM signal chain, thermal sensitivity test |
7.5.2 Mass position and velocity output fusion

The K-TF method for transfer-function inversion in Remark 39 was applied to the QM Vert 2017 dataset. The velocity channel output time series was corrected using the standard ill-posed frequency division Algorithm 19 and converted to an acceleration signal, $a_{VEL}$ in (7.11). The mass position output was purely scaled by a gain factor to convert it to an acceleration signal, $a_{POS}$ in (7.12), as the response to acceleration is flat in the desired bandwidth for the sensor fusion. The time series and root PSD (rPSD) of $a_{VEL}$, $a_{POS}$ and the sensor fusion result of the K-TF method, $a_{µS}$, are shown in Figure 7.2. In the time domain, the synthesised acceleration response, $a_{µS}$, is similar to the temperature profile (in the fourth pane of Figure 7.3) when compared to that derived from the velocity channel, $a_{VEL}$. Therefore, the sensor fusion for transfer-function correction has produced a physically meaningful acceleration response over the entire bandwidth. This result can then be used to obtain a corrected time series in any desired bandwidth, without any processing artefacts, for further analysis.

7.5.3 Validation of the microseismometer thermal compensation and temperature decorrelation

The validation of Algorithm 22 will in turn validate the SP performance at low frequencies, the effectiveness of the thermal compensation developed in [123], the microseismometer performance at extremely low temperatures and the ability of the sensor to observe tidal
Figure 7.3. QM Vert 2017. First pane: Self-noise determination from the $a_{VEL}$ and temperature decorrelated, $a_{µS} - a_{µST}$, signals. Second pane: Improvement ratio between self-noise before and after temperature decorrelation. Third Pane: acceleration time series for the reference, $a_{REF}$, and decorrelated microseismometer acceleration with a comparison of the K-TF+T synthesised acceleration and thermal model fit inset. Fourth pane: $T_{Die}$ and $T_{Enc}$ signals, Fifth pane: $∆T$ signal.
signals. The obtained parameters for the thermal model (7.2) are given in Table 7.2, which reports the value of the coefficients with a standard error calculated as the diagonal elements of $S_{\text{error}} = \left( (X^TX)^{-1}(y - Xb)^2 \right)^{1/2}$ where $X = [T_{\text{die}}, \Delta T, 1]$ is the data matrix of the thermal model components with a bias component, $y = a_{\mu ST}$ is the temperature dependent microseismometer acceleration, and $b = [a_{\text{die}}, a_{\Delta}, b]^T$ is the vector of regression coefficients. The $R^2$ statistic is also reported as KF1 and KF2 respectively for the fit to the K-TF produced $a_{\mu S}$ (Step 2 of Algorithm 22) and for the final iteration of the K-TF+T produced $a_{\mu S}$ (Step 4 of Algorithm 22). The derived self-noise is with respect to the Guralp reference seismometer acceleration, $a_{\text{REF}}$.

The process in Algorithm 22 was applied to the QM Vert 2017 dataset and the results are shown in Figure 7.3. The obtained linear sensitivity parameter is $a_{\text{die}} = -2.50 \pm 0.12 \times 10^{-4}$ ms$^{-2}$/K which is near the theoretical value of $-2.2 \times 10^{-4}$ ms$^{-2}$/K (or -60 ppm/K) for an uncompensated sensor [123]. This dataset has a slowly varying temperature, this means the $\Delta T$ signal has a low variance (Table 7.1) and is less significant to the fit than the linear parameter $a_{\text{die}}$. As a result, the $a_{\Delta}$ component is poorly determined, as seen by the coefficient’s standard error.

Algorithm 22 was also applied to the FM units, of which the vertical microseismometer has the thermal compensation technology. In these tests no reference seismometer was available. Therefore, a purely seismic component cannot be removed from the microseismometer acceleration to calculate the delta noise (6.10), and so the temperature dependent acceleration, $a_{\mu ST}$, was approximated by the sensor output. This is a reasonable assumption as the low frequency ambient seismic signal is much less than the microseismometer’s output when subjected to large temperature variation. As such, the acceleration rPSD provides a reasonable estimation of the microseismometer self-noise. The results for the FM Vert dataset are shown in Figure 7.4. In this case, the microseismometer acceleration is more correlated to the $\Delta T$ signal than the $T_{\text{die}}$ component, conversely to the QM Vert 2017 case. This is seen as the thermal model parameter $a_{\Delta} = -3.80 \pm 0.21 \times 10^{-4}$ ms$^{-2}$/K is well determined in the standard error compared to the parameter $a_{\text{die}} = -0.84 \pm 2.78 \times 10^{-6}$ ms$^{-2}$/K. Therefore, the $\Delta T$ component is now more significant to the thermal model of the microseismometer’s acceleration response, owing to the reduced linear sensitivity. The $a_{\text{die}}$ parameter determined for the FM vertical unit is at least under two orders of magnitude lower than the uncompensated QM sensor, thus validating the performance of the thermal compensation [123]. Moreover, our approach was highly effective at decorrelating the thermal component from the signal with a reduction of up to 35 times in the low frequency seismic amplitude. The rPSD of the temperature decorrelated signal, $a_{\mu S} - a_{\mu ST}$ in Figure 7.5, has a higher power than the $a_{\text{VEL}}$ signal around $2 \times 10^{-3}$ Hz. This is because the mass position is less finely quantised in the FM electronics compared to the velocity channel, introducing an error in this bandwidth. This could be mitigated in the future by a more targeted approach to the attenuation of signal-correlated quantisation errors through, for example, interpolation.

On the other hand, the horizontal FM unit does not have thermal compensation as the proof mass will not be subject to an appreciable component of gravity. Only change in the tilt, $\Delta \theta$, of the sensor will produce a signal due to the change in the gravity
Figure 7.4. FM Vert. First pane: rPSD of the $a_{VEL}$, and temperature decorrelated, $a_{\mu S} - a_{\mu ST}$, signals. Second pane: Improvement ratio between rPSDs before and after temperature decorrelation. Third Pane: time series for the decorrelated microseismometer acceleration compared to the K-TF+T synthesised acceleration and thermal model fit with the K-TF+T synthesised acceleration and mass position inset. Fourth pane: $T_{die}$ and $T_{enc}$ signals. Fifth pane: $\Delta T$ signal.
Figure 7.5. FM Horiz. First pane: rPSD of the $a_{VEL}$, and temperature decorrelated, $a_{µS} - a_{µST}$, signals. Second pane: Improvement ratio between rPSDs before and after temperature decorrelation. Third Pane: time series for the decorrelated microseismometer acceleration compared to the K-TF+T synthesised acceleration and thermal model fit with the K-TF+T synthesised acceleration and mass position inset. Fourth pane: $T_{die}$ and $T_{enc}$ signals. Fifth pane: $\Delta T$ signal.
component, giving some linear temperature sensitivity. The results for the FM Horiz dataset are shown in Figure 7.5 with the thermal model parameters determined as $\alpha_{\text{die}} = -3.72 \pm 0.15 \times 10^{-5} \text{ms}^{-2}/\text{K}$ and $\alpha_\Delta = 1.248 \pm 0.086 \times 10^{-4} \text{ms}^{-2}/\text{K}$. The thermal signal has again been effectively decorrelated from the data with a reduction of 50 times in the low frequency seismic amplitude.

### Table 7.2. Thermal Model Parameters and Fit Quality

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Thermal Model Parameter (ms$^{-2}$/K)</th>
<th>Fit $R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha_{\text{die}}$</td>
<td>$\alpha_\Delta$</td>
</tr>
<tr>
<td>QM Vert 2017</td>
<td>-2.5E-4</td>
<td>2.2E-4</td>
</tr>
<tr>
<td></td>
<td>$\pm$ 0.12E-4</td>
<td>$\pm$ 4.1E-4</td>
</tr>
<tr>
<td>QM Vert 2016</td>
<td>-2.629E-3</td>
<td>1.75E-2</td>
</tr>
<tr>
<td></td>
<td>$\pm$ 0.034E-3</td>
<td>$\pm$ 0.12E-2</td>
</tr>
<tr>
<td>QM Vert Cold</td>
<td>-1.080E-4</td>
<td>-6.1E-4</td>
</tr>
<tr>
<td></td>
<td>$\pm$ 0.016E-4</td>
<td>$\pm$ 6.9E-4</td>
</tr>
<tr>
<td>FM Vert</td>
<td>-0.84E-6</td>
<td>-3.80E-4</td>
</tr>
<tr>
<td></td>
<td>$\pm$ 2.78E-6</td>
<td>$\pm$ 0.21E-4</td>
</tr>
<tr>
<td>FM Horiz</td>
<td>-3.72E-5</td>
<td>1.248E-4</td>
</tr>
<tr>
<td></td>
<td>$\pm$ 0.15E-5</td>
<td>$\pm$ 0.086E-4</td>
</tr>
</tbody>
</table>

**Tidal signals on the microseismometer**

The QM Vert 2016 dataset is taken over 112 hours. As a result, it is possible to observe tidal frequencies in the recording [110] [133]. However, a temperature dependent signal in the data would obscure this tidal signal. Figure 7.6 shows the results for the temperature decorrelation method applied to the QM Vert 2016 dataset. The decorrelated time series contains a significant component also seen by the reference seismometer at the M2 tidal frequency, which is not observable in the uncorrected time series in the inset figure. Moreover, the self-noise estimate of the sensor now continues to follow the power law seen up to $1 \times 10^{-5} \text{Hz}$ with an improvement ratio of up to 28. This demonstrates the ability of Algorithm 22 to recover physical signals accurately and so provide a seismometer output for a wide range of science applications.

The overall temperature variation in this experiment is the lowest of the datasets, as seen by the variance of $T_{\text{die}}$ and $\Delta T$ (in Table 7.1). Due to this low variation, the temperature of each part of the system was coupled to $T_{\text{die}}$. Therefore, $\alpha_\Delta$ does not contribute significantly to the acceleration response and so is poorly determined. Furthermore, the obtained $\alpha_{\text{die}}$ parameter is not representative of just the sensor. Specifically, the feedback electronics board used in this test was known to have a high temperature sensitivity, subsequently corrected for the QM Vert 2017 and FM tests.
Figure 7.6. QM Vert 2016. First pane: Self-noise determination from the $a_{VEL}$ and temperature decorrelated, $a_{\mu S} - a_{\mu ST}$, signals. Second pane: Improvement ratio between self-noise before and after temperature decorrelation. Third Pane: acceleration time series for the reference, $a_{REF}$, and decorrelated microseismometer acceleration with a comparison of the K-TF+T synthesised acceleration and thermal model fit inset. Fourth pane: $T_{die}$ and $T_{enc}$ signals. Fifth pane: $\Delta T$ signal.
Cold testing of the microseismometer

The results for the low temperature test are shown in Figure 7.7 with the thermal model parameters determined as $\alpha_{\text{die}} = -1.080 \pm 0.016 \times 10^{-4} \text{ms}^{-2}/\text{K}$ and $\alpha_{\Delta} = -6.1 \pm 6.9 \times 10^{-4} \text{ms}^{-2}/\text{K}$. Again in this test the $T_{\text{die}}$ component is much more significant to the acceleration response than the $\Delta T$ signal and so $\alpha_{\Delta}$ is poorly determined. Notice that the linear sensitivity parameter, $\alpha_{\text{die}}$ is $1.42 \times 10^{-4} \text{ms}^{2}/\text{K}$ greater than the $\alpha_{\text{die}}$ value obtained from the QM Vert 2017 test. This corresponds to the tilting in the cold test rig mount caused by the large temperature difference between the chamber and the ambient. As the ambient is relatively constant, the temperature difference is proportional to $T_{\text{die}}$ and so the $\alpha_{\text{die}}$ coefficient also accounts for its effect. The tilt from the test rig is equivalent to $1.49 \times 10^{-5} \text{rad}/\text{K}$, a low value indicative of the careful thermoelastic design of the cold test rig.

7.6 Conclusion

This chapter has focused on the thermal response of the microseismometer which constitutes a non-dynamic latent injection into the output signal. To obtain this, a sensor fusion based method for transfer-function correction using a Kalman filter was first introduced. The novel approach utilises the complementary properties of a seismometer’s mass position and velocity outputs to counteract the ill-posedness of current conventional methods.

Furthermore, a thermal model of the microseismometer’s acceleration has been introduced. This model was integrated into the transfer-function correction method and implemented for thermal decorrelation of the microseismometer’s acceleration signal. This is a latent variable regression application where the entity is defined by a model rather than determined from mathematical properties, as done by the PLS used in Part A. Owing to this physics model-based approach, the method is limited by two aspects: the fidelity/SNR of the acquired data and the range for which the coupling applies for the physics-based model. For example, as shown in Figure 7.5, the quantisation of the FM mass position can cause an erroneous fusion for certain bandwidths.

Experimental results confirmed that this technique successfully removes the thermal component from the data for bandwidths $< 1 \times 10^{-2} \text{Hz}$. For the temperature sensitivity tests and the cold test the acceleration signal has been recovered to the level of the sensor noise/second-order effects with a thermal attenuation of 25-50. The ambient temperature tests with a reference show that the signal can be recovered to reveal tidal signals with the estimated sensor self-noise following a single power law. The performance of the passive thermal compensation technology incorporated in the sensor is also validated. In this and other applications, the demonstrated ability to implement a virtual vault will contribute to the maximum retrieval of seismic information from the sensors.
Figure 7.7. QM Vert Cold. First pane: Self-noise determination from the $a_{VEL}$ and temperature decorrelated, $a_{µS} - a_{µST}$, signals. Second pane: Improvement ratio between self-noise before and after temperature decorrelation. Third Pane: acceleration time series for the K-TF+T synthesised acceleration, thermal model fit and temperature decorrelated singals with the reference, $a_{REF}$, and the decorrelated microseismometer acceleration inset. Fourth pane: $T_{Die}$ and $T_{Enc}$ signals. Fifth pane: $∆T$ signal.
Chapter 8

Conclusion

So it goes...

Kurt Vonnegut

This thesis is concerned with the twin goals:

1. to equip latent variable regression methods with the ability to process multidimensional data,

2. To provide a performance analysis, using the latent variable framework, of the MEMS microseismometer for the InSight mission to Mars.

Both goals are framed by the central theme of latent variables. The first goal considers this concept in a classical signal processing sense for the component analysis decomposition of a data matrix. This decomposition is implemented for a regularised regression via the PLS algorithm. The development of such techniques for multidimensional data types allow a single latent vector variable to become a multivariate component and so enables new representations of data. The concept of latent variables is then implemented in a practical application for the self-noise determination of the NASA InSight microseismometer. Here, the latent variables are the seismic and non-seismic contributions to the signal acquired from the sensor and their separation is studied.

8.1 Summary of results

8.1.1 Part A: LV regression with multidimensional methods

The latent variable approach for matrix decomposition and regression is rigorously introduced in Chapter 3. It is shown that the original data, a collection of entity samples in a matrix, is transformed to a new basis to produce the latent entity. The goal is then to find a basis of the independent variables that is more representative for the regression of the dependent variables, and so both regularise the ill-posed regression and delineate the information within the matrix. To that end, the method of PLS is introduced. The PLS philosophy is to decompose the data blocks based on maximising the cross-covariance between them, a physically meaningful cost function. Several types of PLS are introduced
to show that it can be used in several modalities, but the NIPALS algorithm for a PLS-regression solution is used as the focus in this thesis.

The NIPALS algorithm is an iterative approach which calculates successive rank-1 approximations based upon finding the component which represents the maximal cross-covariance direction. This raises the question, what is the optimal rank-1 basis choice for a regression? To answer this question the concept of forwards-backwards regression was introduced. This builds on the generalised inverse theory to show that the regression coefficients for the forwards prediction of the output block should be the generalised inverse of the regression coefficients from the backwards prediction of the input block from the output block. It is then proven that the optimal basis choice which produces the regression coefficients closest to meeting the forwards-backwards criterion is that which maximises the correlation between the input and output block. This basis is generated by the method of CCA and is intimately linked to PLS.

This basis approach edits the spatial information used for regression but does not attempt to correct noise processes over time. This can be achieved through constrained optimisation techniques. It is of interest to consider PLS in this framework, where the entity may have time-varying, non-stationary properties. To this end, an online algorithm for PLS is proposed, OL-PLS, which adapts the basis over the time samples.

Chapters 4 and 5 are concerned with adapting the PLS-regression NIPALS algorithm for use with multidimensional techniques. The latent variable decomposition from the NIPALS algorithm produces orthogonal, single vector, components and so is highly restrictive. The introduced multidimensional techniques of complex numbers, quaternion numbers and tensors instead consider a single variable to be a multivariate data structure. Their respective algebras and decompositions then can encode more physically meaningful relationships in straightforward numerical operations. Each are related, as shown through the derivation of a quaternion to tensor transform in Chapter 5.

The generalisation of the NIPALS algorithm for complex-valued data is proposed as the WL-CPLS in Chapter 4. The extension takes into account full complex-valued second-order statistics in each step to first find the PLS basis and then perform the required widely linear regressions. It is shown that the final approximation produced from the WL-CPLS result yields the required, regularised, generalised inverse for the widely linear latent variable regression. The ability for this to produce a physically meaningful result is highlighted through its application to smart grid frequency estimation. The quaternion-valued algorithm, WL-QPLS, is derived in the same way in Chapter 5. The result calculates a similar regularised quaternion widely linear regression and an orthogonal decomposition of the quaternion-valued independent variables. This is significant for quaternion-valued data as the algorithm can be applied for the diagonalisation of all four complementary covariance matrices, as analogously shown for the WL-CPLS result.

Several tensor-based algorithms already exist, notably the HOPLS algorithm. However, through an in-depth consideration of the core components of the NIPALS algorithm, two new tensor PLS algorithms are introduced. The proposed GHOPLS and HONIPALS are direct generalisations of the two-way NIPALS algorithm using the alternative tensor-based methods for each step and, are thus, algorithms which both exploit the tensor operations
and allow a general Tucker decomposition, as opposed to the HOPLS algorithm. Note that the HONIPALS result degenerates to that from the two-way NIPALS solution. The main divergence of the tensor algorithms is based around how the cross-covariance structure is calculated and decomposed. It is shown that although they represent the same information, the decompositions are fundamentally different. Moreover, it is shown that even though the cross-covariance structure is the same for the WL-QPLS and the HONIPALS algorithms, the subsequent transformation to the quaternion domain and its algebra means that they will have different results. These multidimensional PLS algorithms are applied for an image classification problem where they provide a pre-processing filter, which shows that treating the colour information as a single variable yields a performance advantage.

8.1.2 Part B: The performance analysis of the NASA InSight microseismometer

In Part A latent variables are used in the sense of a component analysis on a data matrix for linear regression applications. This concept is applicable in an extended scope for the performance analysis of the NASA InSight microseismometer. The sensor records data containing several components including: seismic, the self-noise, anthropogenic and environmental. For scientific goals, it is desired to know the seismic ground acceleration. In Chapter 6 the self-noise latent contribution to the recorded signal is determined in order to understand the fundamental limit of the sensor. Chapter 7 is then concerned with the separation of the thermal component within the recorded signal.

In Chapter 6 the self-noise determination of the microseismometer is presented. This is achieved from comparing experimental data from two, colocated seismometers. The coherence method is typically implemented for this application. A thorough analysis of this approach enabled the derivation of a set of equations used to predict the algorithm's result from candidate input components. This provides an insight into the qualities of the coherence method, showing that it is robust to errors in the transfer-function correction and the ambient environment, that is, high SNR between the ambient signal and instrument noise. A further self-noise estimate is proposed; the delta noise method. This is derived as the difference between the test and reference seismometer. A similar set of predictive equations is derived which shows that the delta noise is highly susceptible to transfer-function errors, however, it also provides a time series of the seismometer noise. Although the analysis shows that the coherence method is not adversely affected by the ambient signal SNR, a quiet site is still desired. The nature of the injection of excess cultural terms into the seismometer's output is discussed. These terms can produce an incoherent signal between the two seismometers and so increase the self-noise estimate. Several algorithmic developments are then presented, in particular: (i) a geometric alignment algorithm, (ii) the delta noise optimisation for transfer-function estimation and (iii) the ranked-noise estimator. The geometric alignment is analysed to examine the error that can be introduced. The delta noise optimisation exploits the instability of the delta noise self-noise estimate to counteract the difficulties shown in estimating the transfer-function parameters from the traditional methods. The ranked-noise estimator (based on the coherence method) then provides a definitive self-noise determination, discarding
high noise sections of the time series and averaging over the quiet periods based on the
notion that the instrument cannot measure something below its floor. These techniques
are then applied to data acquired from the microseismometer. This was done for the FM
sensors and for QM sensors at other locations where the conditions meant that cultural
and environmental components were absent or stable so their contribution was negligible.
This enables the performance limits for the instrument itself at high and low frequency to
be determined. The FM units show comparable performance within the range not dom-
ninated by these excess contributions which confirms that they meet the requirements for
the InSight mission. Test data was also obtained from the horizontal FM sensors during
the cruise phase of the mission. This deep space observatory was a quieter environment
than any on Earth and it can be assumed that there is no constant ambient signal. As
a result, the raw data provides a self-noise estimate. This corroborates the results found
from the terrestrial tests.

Since the self-noise component of the signal recorded by the microseismometer has been
identified, it is of interest to understand the contribution of environmental latent variables.
In Chapter 7, the thermal contribution to the microseismometer output is considered. In
order to do so, the ill-posed problem of transfer-function correction must be addressed. The
microseismometer’s velocity response is used as the primary channel for science data as it
has the highest fidelity and its transfer-function is a high-pass filter. The industry standard
method of transfer-function correction is by a frequency division deconvolution, outlined in
Chapter 6, which is then ill-posed at low frequency. On the other hand, the mass position
output, used for the low frequency control of the feedback system, has a low-pass filter
output and so is ill-posed at high frequency. A sensor fusion of the two, complementary,
outputs through a Kalman filter yields a regularisation of the correction and as such, a
transfer-function corrected acceleration signal across the full bandwidth. The thermal
model of the the coupling of temperature to the microseismometer acceleration response
is also provided. This was integrated into the sensor fusion and so an algorithm for
temperature decorrelation is proposed. This is applied to test data that determines the
sensitivity of the FM units to be below the requirements, shows that the passive thermal
compensation has worked and obtain the sensor performance at tidal frequencies and in
extreme cold environments.

8.2 Discussion of method approaches

The work presented in this thesis utilises the concept of latent variables for regression
applications. This approach is first developed by using mathematical properties to determine
individual latent components within signal processing algorithms and is later employed
as a framework to separate specific physical components. In each of these approaches the
models/methods are developed by adding some complexity/attribute. In this section the
effects of this additional complexity is discussed. The aim is to show how their results
may change for certain types of application/data and, thus, where the developments in
this thesis can be used.

The developments in Part A are based on the classical linear regression model, \( Y = XB \),
which relies on the assumptions outlined in Section 3.7. At a fundamental level, this model
describes by how much the $Y$ block will vary depending on a change in the $X$ block. This
is limited to a linear relationship, that is, the dependent variables must be able to be
produced from a linear combination of the independent variables. As a result, there can
be no variation in the process statistics (including the errors) over time/realisation. Any
application whose data conform to these qualities can successfully implement this model.
Should the data not adhere to these assumptions strictly, the solution will provide the
best linear model.

This linear regression model is utilised by the PLS method, which forms the backbone
for the multidimensional latent variable regression algorithms developed in Part A. This
means that the PLS solution is based on the same assumptions of the linear regression
approach. The complexity provided by this method is to overcome the situation where
the solution cannot be calculated. To do so, the PLS separates latent variables based
on orthogonality, which dictates that each component must be statistically independent.
Orthogonality is optimal for the separation of Gaussian sources. However, for more func-
tional, curve-shaped data (such as polynomials) the orthogonality property does not always
separate the data into the desired constituent components. For example, consider that
the inner product of two vectors containing a linear and quadratic function is not zero.
In such circumstances, though, the solution would still identify the subspace containing
such components and provide a regression solution. In short, the PLS approach employs
complexity primarily for regularisation. If the data are Gaussian then the solution is opti-
mal and the latent variables are suitably separated. The PLS will still work effectively for
other types of data but the latent variable decomposition may not split the variables into
the desired contributions, in which case further work is required. The PLS method will,
therefore, yield the most benefit for applications where there are many variables and the
data are generally statistical in nature. The OL-PLS algorithm in Chapter 3 is developed
to combat the stationarity requirement. The additional complexity to do this is provided
by the adaptive calculation and the algorithm instead assumes local stationarity.

In Chapters 4 and 5 the PLS solution is broadened for use with the multidimensional
data types: complex-valued numbers, quaternion-valued numbers and tensors. In the same
way as the original PLS algorithm, the multidimensional extensions provide a regularisa-
tion for the calculation of their respective multidimensional linear regression analogues.
The model here explains how much of a variation in the multidimensional data entity
in the regressors can affect the dependent data structure. Sections 4.6 and 5.6 discuss
how these multidimensional extensions follow similar assumptions to the original linear
regression and have the same requirements (within their respective frameworks) on the
data. These MD-PLS algorithms also separate the latent variables dependent on the ex-
tended version of the orthogonality property, that is, statistical independence in terms of
complex, quaternion or tensor algebra. Unlike the real-domain case the individual com-
ponents within the multidimensional data type do not then have to be independent from
each other. As a result, the added complexity in the MD-PLS models provides a structure
for a PLS type regression (and the separability of the latent variables) with alternative
algebras. The assumptions of the original linear regression model still hold within the new
In Part B the latent variable regression approach is used for an application specific purpose. As such, the focus was no longer on the creation of a general algorithm but to tailor the technique to provide a result which was in part expected from prior knowledge. This problem of determining the seismic signal from instrument and environmental components is formulated in Chapter 6. The aim here is to add complexity in order to delineate these specific factors, in contrast to Part A where the separation is performed for a general array of applications. The main task of Chapter 6 is to determine the self-noise of the microseismometer instrument. This is achieved through the coherence and delta noise methods. The results are given in the frequency domain and are based on the assumption of spectral coherence between a jointly recorded ambient signal and that the topology consists of linear time invariant systems. Note that this uses a mathematical property to discern the self-noise from data. This makes the method straightforward to implement but it is shown that the obtained self-noise estimate can contain spurious contributions. The analysis of these injections and their sources allows them to be identified and minimised, implementing complexity through prior knowledge to tailor the result for seismometer noise determination.

In Chapter 7, the presented work is aimed at removing the temperature response of the microseismometer output, a specific instance of an environmental latent contribution outlined in Chapter 6. To do so, a thermal model for the microseismometer response is developed. The result is cast into the linear regression framework where the output is dependent on the pre-derived components. This is advantageous as the contributions can be calibrated from a straightforward application to acquired data. To implement such an approach, however, the physical mechanisms for each component must be understood. In this instance, the added complexity is applied through pre-determination of the latent factors for a straightforward, easy to interpret model. On top of the thermal model the K-TF algorithm for transfer-function inversion is proposed. This method combines two signals with complementary properties. The development of the K-TF+T algorithm incorporates further information into the synthesis model. In these algorithms an alternative approach to linear regression is required, given by the Kalman filter. The Kalman filter is based on a state space model where the state variables are desired to be estimated. The state space model incorporates prior information through a forward model and the Kalman filter combines this prediction with measured data through a recursive calculation. The Kalman filter employs complexity to best cater for a situation with both measurements and prior physics understanding by combining the two. However, it will suffer from uncertainty if either are missing information/erroneous. It is also not possible to implement without reasonable prior knowledge of the system. In this thesis the standard Kalman filter is implemented but further extensions exist for non-linear problems.

In general, added complexity improves the power and robustness of the estimate but the technique is then less flexible or straightforward to use, requiring refinement and prior information. The methods used in this thesis and their potential application types are summarised in Table 8.1.
### Part A

<table>
<thead>
<tr>
<th>Method/Algorithm</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>OLS</td>
<td>To determine linear variations between data. Examples: finance, economics and biology.</td>
</tr>
<tr>
<td>PLS</td>
<td>Linear regression with many, not easily interpretable and highly correlated variables. Examples: chemometrics and sensor array data.</td>
</tr>
<tr>
<td>WL-CPLS</td>
<td>For PLS applications represented by complex-valued relationships. Examples: wave data and smart-grids.</td>
</tr>
<tr>
<td>WL-QPLS</td>
<td>For PLS applications represented by quaternion-valued relationships. Examples: images, orientation estimation and smart-grids.</td>
</tr>
<tr>
<td>HONIPALS/GHOPLS</td>
<td>For PLS applications represented by tensor datastructures. Examples: batch data, neurological signals and tensor-folded time series.</td>
</tr>
</tbody>
</table>

### Part B

<table>
<thead>
<tr>
<th>Method/Algorithm</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coherence and delta noise</td>
<td>Calculates an instrument’s self-noise based on the incoherence assumption. Prior knowledge used for better interpretation of the result.</td>
</tr>
<tr>
<td>$\mu$S thermal model</td>
<td>A tailored linear model for the microseismometer temperature response.</td>
</tr>
<tr>
<td>K-TF/K-TF+T</td>
<td>The K-TF algorithm can be used for fusion of complementary sensor data and the K-TF+T shows how to integrate physical components to this approach.</td>
</tr>
</tbody>
</table>
8.3 Future Work

The main theme for future work is that of probabilistic and statistical model based methods. Probabilistic methods have already been developed for component analysis (probabilistic PCA [144]) and enjoyed success. The probabilistic model caters for noise/extraneous components in the acquired data and so the resultant generative model can better describe a latent entity. Work is in progress to extend this for tensors in terms of a probabilistic Tucker decomposition. Moreover, in Chapter 6, it was shown that the self-noise determination problem can be treated as a statistical problem where the noise model can be used as a prior. In this way it may be possible to separate the excess, incoherent injections and so obtain an improved self-noise determination from noisy environments. In Chapter 7 the temperature latent variable was treated through a physical model. Physical models for other contributions (e.g. the rotation induced acceleration) are being developed. The combination of statistical models and more deterministic models allows the behaviour of systems to be well characterised and therefore separate information on a more physical basis.

This leads to the field of Gaussian processes [41] which use a Bayesian approach to determine a generative model from data. This model is based on a covariance function, typically informed by observed structure within the data. Further analysis then allows the underlying physics to be determined where necessary. This can be applied to produce an improved generative noise model for the microseismometer and as such improve the robustness of the delta noise estimate. These proposed developments are motivated by future development of the microseismometer, for example, it has been suggested as the baseline for the seismic instrument in NASA’s Europa Lander Study by its Science Definition Team [145] [146] [147].

8.4 Summary and closing remarks

The latent variable methods used in Part A of this thesis are based around the matrix decomposition approach outlined in Chapter 3. Here, it was shown that the change of basis, entity space, provides a regression on purely the significant components. The approach of identifying the entity through the row space is employed by the PLS algorithm, utilised for multidimensional data structures throughout Part A. This result is general, but is limited for applications where the entity is non-stationary and degrades in the presence of noise. The method of TLS does take noise into account, as a perturbation in the regression model, but the result is only general if the noise follows such a mathematical constraint. Furthermore, the proposed OL-PLS allows non-stationary entities to be considered in this framework.

Chapter 7 can be viewed as a latent variable regression application. The sensor fusion approach to obtain an accurate acceleration response was, in contrast, a form of modification of the sample space so that the entity produced was more physically meaningful. Moreover, the application of the thermal model provided the independent variables from a physics basis. This demonstrates the spectrum of available techniques where standard
signal processing algorithms (such as PLS) use mathematical properties to separate the information and determine an entity, whereas application specific approaches integrate a physical model for the entity. The former are, in fact, often used initially to inform the latter. Furthermore, consider the injection of the signal-correlated quantisation error highlighted as a shortcoming of the full-band approach in Section 7.5.3. This is not taken into account by the entity model and so is an example of how sample space errors can affect the result and so any constraints applied (in methods like TLS) should be general for the data. In the case of extremely high dimensional data, however, a physics model may not make sense and so the techniques in Part A are used purely to make a problem tractable.

In prediction applications, a general and consistent least squares type solution is usually desired, as highlighted by the forwards and backwards regression concept in Chapter 3. This is because the functional form indicates how much the output varies dependent on the input. If the sample space is required to be corrected for an application, then the modification should be repeatable (as is the case for the full-band algorithms) which allows the problem to be reformed into the standard regression model.

In this thesis we have therefore explored the process for implementing a latent variable regression on a variety of application types. We have specifically created a bespoke solution for the NASA InSight microseismometer which will enable us to determine the most accurate seismic signal possible on Mars.
Bibliography


