Quaternion Valued Adaptive Signal Processing

by
Cyrus Jahanchahi

A Thesis submitted in fulfilment of requirements for the degree of
Doctor of Philosophy of Imperial College London

Communication and Signal Processing Group
Department of Electrical and Electronic Engineering
Imperial College London
2013

Abstract

Recent developments in sensor technology, human centered computing and robotics have brought to light new classes of multidimensional data which are naturally represented as three- or four-dimensional vector-valued processes. Such signals are readily modeled as real vectors in $\mathbb{R}^3$ and $\mathbb{R}^4$, however, it has become apparent that there are advantages in processing such data in division algebras - the quaternion domain. The progress in the statistics of quaternion variable, particularly augmented statistics and widely linear modeling, has opened up a new front of research in vector sensor modeling, however, there are several key problems that need to be addressed in order to exploit the full power of quaternions in statistical signal processing. The principal problem lies in the lack of a mathematical framework, such as the $\mathbb{CR}$-calculus in the complex domain, for the differentiation of non-holomorphic functions. Since most functions (including typical cost functions) in the quaternion domain are non-holomorphic, as defined by the Cauchy-Riemann-Fueter (CRF) condition, this presents a severe obstacle to solving optimisation problems and developing adaptive filtering algorithms in the quaternion domain. To this end, we develop the $\mathbb{HR}$-calculus, an extension of the $\mathbb{CR}$-calculus, allowing the differentiation of non-holomorphic functions. This is followed by the introduction of the I-gradient, enabling for generic extensions of complex valued algorithms to be derived. Using this unified framework we introduce the quaternion least mean square (QLMS), quaternion recursive least squares (QRLS), quaternion affine projection algorithm (QAPA) and quaternion Kalman filter. These estimators are made optimal for the processing of noncircular data, by proposing widely linear extensions of their standard versions. Convergence and steady state properties of these adaptive estimators are analysed and validated experimentally via simulations on both synthetic and real world signals.

Acknowledgment

Firstly, I must thank my supervisor Professor Danilo Mandic for his guidance throughout my PhD and for sharing his enthusiasm on complex valued and quaternion valued signal processing. He has managed to create a friendly research environment where ideas can easily be discussed and explored, and for that I am grateful. This thesis would also not have been possible without the department's EPSRC scholarship, for which I am grateful.

I would also like to thank my research colleagues, particularly Dr. Clive Cheong Took and Dahir Dini, who have collaborated on a number of publications. I have also enjoyed the company of Alireza Ahrabian, Xia Yili, Che Ahmad Bukhari, David Looney, Naveed ur Rehman, Soroush Javidi, Dongpo Xu and Rosa Mara Fernandez-Alcala.

Last, but not least, this thesis would not have been possible without the support of my parents. For that, I will be forever grateful.

Contents

Abstract **2**

Acknowledgment **3**

Contents **4**

List of Figures **9**

List of Tables **12**

Statement of Originality **13**

Copyright Declaration **14**

Abbreviations **18**

Mathematical Notations **20**

Chapter 1. Introduction **22**

1.1 Overview 22

1.1.1 Signal processing in $\mathbb{C}$ 24

1.1.2 Adaptive signal processing in $\mathbb{H}$ 28

1.2 Motivation and aims 28

1.3 Organisation of the thesis 30

Chapter 2. Background Theory **32**

2.1 An overview of quaternions and their application 32

2.1.1 Where and why are quaternions used? 33

2.1.2 Quaternion algebra 35

2.2 Augmented statistics, circularity and widely linear modeling 38

2.2.1 Augmented statistics 38

2.2.2 Circularity 40

2.2.3 Widely linear model 43

2.3 Complex differentiability 46
2.3.1 Analytic functions 46
2.3.2 The $\mathbb{CR}$-calculus 47
2.3.3 The gradient operator 50
2.4 An overview of adaptive signal processing 51
2.4.1 Mathematical models 51
2.4.2 An intuitive description of adaptive filters in $\mathbb{R}$ 57

Chapter 3. The HR-Calculus and Quaternion Gradients **62**
3.1 Introduction 62
3.2 The HR-calculus 65
3.2.1 The $\mathbb{HR}$-derivatives 65
3.2.2 The $\mathbb{HR}^*$-derivatives 67
3.2.3 Comparison with the $\mathbb{CR}$-calculus 69
3.2.4 Examples 72
3.3 Quaternion gradients 79
3.3.1 Unifying the quaternion gradient: The I-gradient 80
3.4 A unifying treatment of quaternion LMS algorithms 82
3.4.1 The quaternion Wiener solution 83
3.4.2 The widely linear wiener solution 84
3.4.3 The quaternion least mean square algorithm 85
3.4.4 The widely linear least mean square algorithm (WLQLMS) 86
3.4.5 A class of generic QLMS algorithms 87
3.5 Equivalence of the various forms of QLMS 88
3.6 Convergence analysis 90
3.6.1 Convergence in the mean of the strictly linear IQLMS algorithm . . 91
3.6.2 Convergence in the mean of HR-QLMS 92
3.6.3 Convergence in the mean of QLMS 94
3.6.4 Mean square error performance of the strictly linear IQLMS for noncircular data 95
3.6.5 Mean square error performance of the widely linear WLIQLMS for noncircular data 97
3.7 Simulations 98
3.8 Summary 101

Chapter 4. Quaternion Recursive Least Squares **103**
4.1 Introduction 103
4.2 Derivation of the widely linear QRLS 103

4.2.1 The least squares solution . . . 104
4.2.2 The widely-linear QRLS algorithm . . . 106
4.2.3 Summary of the widely linear QRLS algorithm . . . 112
4.3 Simulations . . . 113
4.4 Summary . . . 115

Chapter 5. A Class of Quaternion Valued Affine Projection Algorithms 117
5.1 Introduction . . . 117
5.2 A class of quaternion affine projection algorithms . . . 119
5.3 Mean square error analysis of the strictly linear QAPA on noncircular data 121
5.4 Mean square error analysis of the widely linear QAPA on noncircular data 127
5.5 Simulations . . . 129
5.6 Summary . . . 135

Chapter 6. A Class of Quaternion Kalman Filters 138
6.1 Introduction . . . 138
6.2 Reduced widely linear model for three-dimensional signals . . . 140
6.3 Quaternion Kalman filter . . . 143
6.4 Widely linear quaternion Kalman filter . . . 146
6.5 An efficient implementation of the WLQKF . . . 148
6.6 Widely linear quaternion extended Kalman filter . . . 153
6.7 Duality between the quaternion and real valued Kalman filter . . . 156
6.8 Simulations . . . 157
6.8.1 Filtering of an autoregressive process . . . 157
6.8.2 Multistep ahead prediction . . . 159
6.8.3 Bearings only tracking . . . 160
6.9 Summary . . . 162

Chapter 7. Conclusions and Future Work 166
7.1 Conclusions . . . 166
7.2 Further work . . . 168

Bibliography 171

Appendix A. Duality Between Four-Channel Real Valued LMS Filter and the QLMS 187
A.1 The four-channel real valued LMS filter . . . 187
A.2 Duality between the four-channel real LMS and the WLIQLMS . . . 188
A.3 Mathematical complexity . . . 190

Appendix B. QLMS Convergence Proofs 192
B.1 Proof for the equation (3.64) . . . 192
B.2 Expression for the convergence of the QLMS . . . 193
B.3 Convergence comparison of the QLMS, HR-QLMS and IQLMS . . . 194
B.4 Proof for the equation (3.70) . . . 195
B.5 Proof for $E[\mathbf{x}\mathbf{x}^T] = -\frac{1}{2}E[\mathbf{x}\mathbf{x}^H]$ . . . 195
B.6 MSE expression for the IQLMS . . . 196
B.7 Proof for the conservation of weight error energy . . . 197

Appendix C. The Discrepancy Between the Measured MSE of the ARMA Process in (5.39) and Theoretical MSE in (5.36) 199

Appendix D. Proof for the First Order Quaternion Taylor Series 201

Appendix E. Adaptive Convex Combination Approach for the Identification of Improper Quaternion Processes 203
E.1 Introduction . . . 203
E.2 Model order identification . . . 204
E.2.1 Filter weight updates . . . 204
E.2.2 Tap length adaptation . . . 207
E.3 Convergence of the mixing parameter . . . 209
E.4 Steady-state analysis of FT based algorithms . . . 211
E.5 Simulations . . . 217
E.5.1 Optimal tap-length . . . 219
E.5.2 Modelling of quaternion-valued systems . . . 220
E.5.3 Evolution of model parameters in nonstationary systems . . . 221
E.6 Summary . . . 223

Appendix F. An Adaptive Diffusion Quaternion LMS Algorithm for Distributed Networks of 3D and 4D Vector Sensors 225
F.1 Introduction . . . 225
F.2 Derivation of D-IQLMS . . . 226
F.3 The D-WLIQLMS . . . 227
F.4 Stability analysis of D-WLIQLMS . . . 227
F.5 Simulations . . . 229
F.5.1 A circular AR(4) process . . . 231
F.5.2 Noncircular Lorenz attractor . . . 232
F.6 Summary . . . 232

Appendix G. Quaternion Derivatives: GHR Calculus 233

Appendix H. Split Complex Estimation: A Novel Method for Designing Efficient Widely Linear Adaptive Filters 235

List of Figures

2.1 Probability distribution of circular and noncircular complex Gaussian random variable. 41
2.2 Block diagram description of the filtering problem. 52
2.3 Block diagram of an adaptive filter. 57
2.4 Mean square error cost function for a weight vector of size two. 58
2.5 Weight vector trajectory along mean square error surface for input signal $\mathbf{x}(k)$ of differing correlation. 59
2.6 Subspace interpretation of the NLMS and APA, where w_o denotes the optimum weight vector. 61

3.1 Performance comparison of the QLMS, HR-QLMS and IQLMS on the prediction of the circular AR(4) signal, for all the algorithms having the same step size. 100
3.2 Performance comparison of the QLMS, HR-QLMS and IQLMS on the prediction of the circular MA(4) signal, for all the algorithms having the same step size. 100
3.3 Evolution of the adaptive quaternion valued weights of QLMS, HR-QLMS and IQLMS, for a single trial prediction of the MA(4) signal. 101
3.4 Learning curves for the noncircular Lorenz signal, averaged over 100 experiments. 102
3.5 Four-dimensional wind signal. 102
3.6 Learning curves for the strictly linear algorithms on the prediction of the noncircular wind signal, averaged over 20 experiments. 102
3.7 Learning curves for the widely linear algorithms on the prediction of the noncircular wind signal, averaged over 20 experiments. 102

4.1 Performance of WLQLMS and WLQRLS on the prediction of 4D wind field (3D wind speed and air temperature). 114
4.2 Learning curves of WLQLMS and WLQRLS on the prediction of 4D wind field (3D wind speed and air temperature). 115
4.3 Steady state performance of WLQLMS and WLQRLS on the prediction of 4D wind field (3D wind speed and air temperature). 115
4.4 Prediction performance of WLQRLS filter for 5-step ahead prediction of the vertical wind speed component. 116

5.1 Optional caption for list of figures 131
5.2 Simulated MSE of QAPA as a function of the step size μ, for the improper ARMA process in (5.39). 132
5.3 Simulated MSE of WLQAPA as a function of the step size μ, for the improper ARMA process in (5.39). 132
5.4 Learning curves for QAPA and WLQAPA for the circular AR process in (5.37), with $\mu = 1$ and $N = 2$. 133
5.5 Learning curves of QAPA and WLQAPA for the noncircular MA process in (5.38), with $\mu = 1$ and $N = 2$. 134
5.6 Learning curves of WLQAPA for the noncircular MA process in (5.38), for $N = 1$ and $N = 4$. 135
5.7 Learning curves of QAPA and WLQAPA for the noncircular ARMA process in (5.39), with $\mu = 1$ and $N = 2$. 136
5.8 Comparison of the theoretical bound in (5.34) and simulated steady state MSEs for WLQAPA, based on noncircular ARMA signal in (5.39), when $N = 1, 2, 3$ and 4. 137
5.9 Comparison of the theoretical bound in (5.34) and simulated steady state MSE of WLQAPA, on noncircular MA signal in (5.38) when $N = 1, 2, 3$ and 4. 137

6.1 Performance comparison between QKF and WLQKF for the AR(4) process in (6.84), under varying degrees of noncircularity of the state and observation noise. 158

6.2 Performance comparison between QEKF and WLQEKF for the AR(4) process in (6.85), observed through output nonlinearity, for varying degrees of noncircularity of the state and observation noise. 158
6.3 Multistep ahead prediction of real-world four dimensional wind data using QKF and WLQKF. 159
6.4 Geometric view of the wind signal noncircularity via the six scatter plots of quaternion components. 160
6.5 Bearings only tracking using the widely linear quaternion extended Kalman filter. 163

E.1 Hybrid filter structure. 206
E.2 The steady-state MSE for the circular process W_1 and the noncircular process W_2 with respect to the tap-length (model order). 218
E.3 The evolution of the optimal filter length parameter p and the mixing parameter λ for the modelling of the strictly linear system W_1. 219
E.4 The evolution of the optimal filter length parameter p and mixing parameter λ for the modelling of the widely linear system W_2. 220
E.5 The steady-state MSE for the process noncircular W_1 with respect to tap-length. 222
E.6 The evolution of the mixing parameter p for the modelling of the linear system W_1 for the intervals $1 \leq k \leq 3000$, widely linear W_2 for $3001 \leq k \leq 6000$ and noncircular W_1 for $6001 \leq k \leq 9000$. 223
E.7 The evolution of the optimal filter length parameter p for the modelling of the system W_1 for the intervals $1 \leq k \leq 3000$, noncircular W_1 $6001 \leq k \leq 9000$ and W_2 for $3001 \leq k \leq 6000$. 224

F.1 Topology of a distributed network, highlighting the neighbourhood of node i. 226
F.2 Network topology. 230
F.3 Performance comparison of the QLMS, HR-QLMS and IQLMS on a circular AR(4) signal, for all the algorithms having the same step size. 231
F.4 Comparison of MSEs of the D-WLIQLMS and DIQLMS on the noncircular Lorenz signal. 232

List of Tables

2.1 The pesudocovariance matrices written in terms of the real components $\mathbf{q}_r$, $\mathbf{q}_i$, $\mathbf{q}_j$ and $\mathbf{q}_\kappa$. 43

5.1 MSE of QAPA and WLQAPA for different processes. Key: 1. AR(1), 2. MA(3), 3. ARMA, 4. Tai Chi (left), 5. Tai Chi (right), 6. Wind (low), 7. Wind (medium), 8. Wind (high). 136

E.1 Noncircular quadruply white quaternion Gaussian noise used. 221

G.1 Important results of the GHR derivatives 234

Statement of Originality

I hereby declare that to the best of my knowledge, this work contains original contributions to the field of quaternion valued adaptive signal processing. I acknowledge the sources in every instance that ideas pertaining to other people are used. The original contributions arising from this thesis are detailed below:

- The HR-calculus,
- The quaternion gradient and I-gradient,
- The widely linear quaternion affine projection algorithm (WLQAPA) and mean square error analysis,
- The quaternion Wiener filter and widely linear quaternion recursive least squares (WLQRLS),
- The IQLMS, equivalence analysis and convergence analysis,
- The quaternion first order Taylor series, widely linear quaternion Kalman filter and its efficient implementation,

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.

Publications

The following are contributions resulting from this work
Journal articles

[J1] C. Jahanchahi and D.P. Mandic, "A Unifying Approach to Quaternion Valued Adaptive Filtering," IEEE Transactions on Signal Processing, submitted December 2013.

[J2] C. Jahanchahi, S. Kanna and D.P. Mandic, "Split Complex Estimation: A Novel Method for Designing Efficient Widely Linear Filters," Signal Processing, submitted November 2013.

[J3] D. Xu, C. Jahanchahi and D.P. Mandic, "Quaternion Derivatives: GHR Calculus," London Mathematical Society, submitted October 2013.

[J4] Y. Xia, C. Jahanchahi, D.P. Mandic, "Quaternion-Valued Echo State Networks for Nonlinear Adaptive Filtering," IEEE Transactions on Neural Networks and Learning Systems, accepted February 2014.

[J5] B.C. Ujang, C. Jahanchahi, C.C. Took and D.P. Mandic, "Adaptive Convex Combination Approach for the Identification of Improper Quaternion Processes," IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 1, pp. 172-182, 2014.

[J6] C. Jahanchahi, C.C. Took and D.P. Mandic, "A Class of Quaternion Kalman Filters", IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 3,

pp. 533-544.

[J7] C. Jahanchahi, C.C. Took and D.P. Mandic, "A class of Quaternion Valued Affine Projection Algorithms", Signal Processing, vol. 93 , no. 7, pp. 1712-1723, 2013.

[J8] D.H. Dini, C. Jahanchahi and D.P. Mandic, "Kalman Filtering for Widely Linear Complex and Quaternion Valued Bearings Only Tracking," Signal Processing, IET, vol. 6, no. 5, pp. 435-445, 2012.

[J9] D.P. Mandic, C. Jahanchahi and C.C. Took, "A Quaternion Gradient Operator and Its Applications," Signal Processing Letters, IEEE, vol. 18, no. 1, pp. 47-50, 2011.

Conference proceedings

[C1] C. Jahanchahi and D.P. Mandic, "An Adaptive Diffusion Quaternion LMS Algorithm for Distributed Networks of 3D and 4D Vector Sensors ," *in Proceedings of the European Signal Processing Conference (EUSIPCO)*, accepted, September 2013.

[C2] R. M. Fernandez-Alcala, J. Navarro-Moreno, J. C. Ruiz-Molina, C. Jahanchahi and D. H. Dini, "A Quaternion Widely Linear One-stage Prediction Algorithm," *in PIERS Proceedings*, pp.1355-1358, August 2012.

[C3] C. Jahanchahi, C.C. Took and D.P. Mandic, "On Gradient Calculation in Quaternion Adaptive Filtering," *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3773-3776, March 2012.

[C4] C.C. Took, C. Jahanchahi and D.P. Mandic, "A Unifying Framework for the Analysis of Quaternion Valued Adaptive Filters," *in Proceedings of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 1771-1774,

November 2011.

[C5] S. Javidi, C. Cheong Took, C. Jahanchahi, N. Le Bihan and D. P. Mandic, "Blind Extraction of Improper Quaternion Sources," *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3708-3711, May 2011.

[C6] C. Jahanchahi, C.C. Took and D.P. Mandic, "The Widely Linear Quaternion Recursive Least Squares Filter," *in Proceedings of the 2nd International Workshop on Cognitive Information Processing (CIP)*, pp. 87-92, June 2010.

[C7] C. Jahanchahi, C.C. Took and D.P. Mandic, "On HR calculus, Quaternion Valued Stochastic Gradient, and Adaptive Three Dimensional Wind Forecasting," *in Proceedings of The International Joint Conference on Neural Networks (IJCNN)*, pp. 1-5, July 2010.

Abbreviations

AAPA: Augmented Affine Projection Algorithm

ACLMS: Augmented Complex Least Mean Square

APA: Affine Projection Algorithm

AR(n): Autoregressive model of order n

ARMA: Autoregressive Moving Average

BPSK: Binary Phase-Shift Key

CCQLMS: Collaborative Combination of QLMS and WLQLMS

CDMA: Code Division Multiple-Access

CLMS: Complex Least Mean Square

CRF: Cauchy-Riemann Fueter

EMSE: Excess Mean Square Error

FAP: Fast Affine Projection Algorithm

FIR: Finite Impulse Response

FT: Fractional Tap

IQLMS: Involution Quaternion Least Mean Square

LMS: Least Mean Square

LQ: Linear Quadratic

MA(n): Moving Average model of order n

MAI: Multiple Access Interference

MSE: Mean Square Error

NLMS: Normalized Least Mean Square

QAPA: Quaternion Affine Projection Algorithm

QEKF: Quaternion Extended Kalman Filter

QLMS: Quaternion Least Mean Square

QPSK: Quadrature Phase Shift Key

QEKF: Quaternion Extended Kalman Filter

QLMS: Quaternion Least Mean Square

QPSK: Quadrature Phase Shift Key

QRLS: Quaternion Recursive Least Squares

RLS: Recursive Least Squares

TSE: Taylor Series Expansion

WL: Widely Linear

WLCLMS: Widely Linear Complex Least Mean Square

WLIQLMS: Widely Linear Involution Quaternion Least Mean Square

WLQAPA: Widely Linear Quaternion Affine Projection Algorithm

WLQEKF: Widely Linear Quaternion Extended Kalman Filter

WLQLMS: Widely Linear Quaternion Least Mean Square

WLQRLS: Widely Linear Quaternion Recursive Least Squares

WLRLS: Widely Linear Recursive Least Squares

Mathematical Notations

$\mathbb{R}$	Field of real numbers
$\mathbb{C}$	Field of Complex numbers
$(\cdot)^T$	Transpose operator
$(\cdot)^H$	Hermitian operator
$E[\cdot]$	Expected value
$\mathbb{H}$	Field of quaternion numbers
q_r	Vector of real component of $\mathbf{q}$
$q_\imath$, $q_\jmath$, q_κ	Vector of $\imath$, $\jmath$ and κ component of $\mathbf{q}$
$\mathbf{I}$	Identity matrix
$\mathbf{x}$	Bold lower case denotes column vector
x	Lower case denotes scalar
$\mathbf{X}$	Bold upper case denotes matrix
$\in$	Is an element of
$\{\cdot\}$	The set of
Sq	Scalar part of q
Vq	Vector part of q
$\bullet$	Dot product
$\times$	Cross product
q^*	Conjugate of q
$\|\| \cdot \|\|$	Vector or Matrix norm
$q^\imath$, $q^\jmath$, q^κ	$\imath$-, $\jmath$- and κ-involution of q

q^*	Conjugate of q
$\mathbf{R}_{qq}$	Covariance matrix of $\mathbf{q}$
$\mathbf{P}_{qq}$	$\imath$-covariance matrix of $\mathbf{q}$
$\mathbf{S}_{qq}$	$\jmath$-covariance matrix of $\mathbf{q}$
$\mathbf{T}_{qq}$	κ-covariance matrix of $\mathbf{q}$
$\mathbf{R}_{qq}^a$	Augmented covariance matrix of $\mathbf{q}$
$\mathbf{R}_{qq}^r$	Real part of covariance matrix of $\mathbf{q}$
$\mathbf{q}^a$	Augmented vector
∂	Partial derivative operator
∇_w	Gradient operator
$\Re(\cdot)$	Real part
$\Im(\cdot)$	Imaginary part
$(\cdot)_{x,y}$	The x^{th} row and y^{th} column
$(\cdot)_x$	The x^{th} row
$Tr(\cdot)$	Matrix Trace Operator
Λ	Eigenvalue matrix
$[\mathbf{R}_{xx}]_n$	The n-component of $\mathbf{R}_{xx}$ for $n = \{r, \imath, \jmath, \kappa\}$
$e_a(k)$	A priori error
$e_p(k)$	A posteriori error
$\mathbf{x}^c$	Augmented vector of involution terms
σ^2	Noise variance
μ	Step size
$\mathbf{1}$	Column vector of 1 followed by zeros.
$\mathbf{x}_{k\|k-1}$	A priori estimate
$\mathbf{x}_{k\|k}$	A posteriori estimate
$cov(\cdot)$	Covariance matrix of term in brackets
$\mathbb{1}$	Column vector of ones

Chapter 1

Introduction

1.1 Overview

All adaptive signal processing problems have one common feature: the input signals and a desired signal are used to generate an output error signal, which in turn can be used to optimise a set of filter weights. It is the way that the input signal and desired signals are configured that allows adaptive filters to be useful in a wide variety of applications. Depending on these configurations, there are four types of possible adaptive filtering applications:

- System identification: In this group of applications, the aim is to find a linear model that best describes an unknown system. To achieve this, the adaptive filter is configured such that the unknown system is parallel to the adaptive filter, both fed with the same signal.
- Linear prediction: Here, the aim is to estimate the future samples of a signal. In this configuration, the input signal is a delayed version of the desired signal. This model is widely used in speech processing applications.
- Inverse modeling: In this third group of applications, the aim is to obtain the inverse of an unknown noisy system. This is widely used in channel equalization [1], where for example, it is applied in modems to reduce channel distortions resulting from data

transmissions over telephone channels. In this configuration the unknown system and adaptive filter are in series.

- Noise cancellation: In this final class of application, the aim is to subtract an unknown noise source from a primary signal (e.g. in electrocardiography [1, 2]). In this configuration, the desired signal is a signal that is correlated with the noise/interference signal that corrupts the primary signal.

The development of adaptive filters can be traced back to the 1950s. The first appearance of the recursive least squares (RLS) can be attributed to Plackett [3], although many different authors have since derived variants of the RLS. In 1959, using stochastic gradients instead of least squares solutions, Widrow and Hoff [4] developed the LMS (least mean square) while studying an adaptive pattern classification machine called Adaline (adaptive linear) [5]. Due to its simplicity, the LMS has since become the workhorse of the adaptive signal processing community.

In 1960, Rudolf Kalman used probability theory and properties of conditional Gaussian random variables to develop the Kalman filter [6], an adaptive state estimator. One of the first applications of the Kalman filter can be credited to Stanley Schmidt [7], who incorporated the Kalman filter into the Apollo navigation computer. Since then, the Kalman filter has become a vital component of navigation systems, including those of the NASA Space Shuttle and U.S. Navy unclear ballistic missile submarines.

IN 1974, Godard, while working on applications of Kalman filter theory, derived a variant of the RLS algorithm [8] that could be used for Channel Equalization. It was not, however, until 1994 that the exact relationship between the RLS algorithm and Kalman filter was established [9], thus opening the way for the rich literature on Kalman filtering to be exploited for solving adaptive filtering problems. Compared to the LMS, the RLS has the major drawback of being computationally intensive and to overcome this, in the late 1970s, a lot of work was put into developing fast RLS algorithms [10–12].

One important drawback with the LMS lies with the step size (also called learning rate) which is used to control the convergence rate and steady state performance, while

making sure that it is small enough to guarantee convergence in the mean and mean square error. The range of value that the step size takes to ensure convergence is, however, sensitive to the characteristics of the input signal. In practical applications, this means that an overly conservative step size is usually chosen to ensure the stability of the filter. This problem was overcome in 1967 by the development of the NLMS (normalized LMS) [13], which divides the step size by the norm of the input signal, thus ensuring that the filter operates within its stability range without sacrificing any of the convergence speed. The convergence properties of the NLMS were not however confirmed until much later, when it was proven that its convergence is indeed independent of the environment in which it operates [14–16]. In practical applications, a regularization term is usually added to the norm of the signal to ensure that the NLMS does not diverge if the signal power goes to zero [17].

Due to its simplicity and robustness, the NLMS has been among the most popular stochastic gradient based linear adaptive filtering algorithms. However, due to the stochastic nature in which the NLMS performs weight optimization, the convergence can severely deteriorate when the input signals exhibit a degree of correlation. To remedy this, in 1984 Ozeki and Umeda [18] used affine subspace projections to develop the affine projection algorithm (APA), allowing for a faster convergence for coloured inputs. In practical terms, the APA can be seen as structurally between the NLMS and RLS, both in term of performance and computational complexity. Whereas the NLMS updates the weight vectors based only on the current input vector, the APA achieves this by employing the past N input vector, thus making better use of the history of weight update evolutions, and achieving faster convergence than the NLMS. The development of the APA was later followed by an efficient implementation of the algorithm, the fast APA (FAP) [19].

1.1.1 Signal processing in $\mathbb{C}$

Complex valued signals can be broadly split into two groups: those that are complex for convenience of representation and those that are complex by design. An example of signals that are complex by design can be found in applications such as radar, sonar and

communications, where a modulating signal, containing the information to be transmitted, is usually modulated onto a higher frequency carrier wave. The baseband signal can be obtained by moving the narrow band signal to a lower frequency, thus removing the effect of the carrier signal while preserving the message contained within the original signal. It is often the case that the baseband signal is complex valued, i.e.

$$x(k) = |x(k)|e^{\jmath\phi(k)} \tag{1.1}$$

where $|x(k)|$ and ϕ are respectively the magnitude and phase angle on the signal $x(k)$. In Beamforming, arrays of sensors are typically used to distinguish between the spatial properties of a desired signal and noise. Typically, the sensor outputs are in baseband form and weighted using a complex steering vector to obtain the desired signal. [20]. Due to the duality that exists between $\mathbb{R}^2$ and $\mathbb{C}$, any two-dimensional real signal can also be made complex valued. For example wind, velocity measurements can be made complex valued by using the speed and direction measurements to form the real and imaginary components of the complex signal.

In 1975, Widrow extended the LMS to the complex domain [21, 22], thus allowing two-dimensional signals to be processed using the complex LMS (CLMS). The CLMS has since become an important tool in noise cancellation of complex valued baseband signals [23,24] and adaptive filtering of signals in the frequency domain [25–27]. Originally, complex algorithms such as Widrow's complex LMS [21] were derived by simply extending existing algorithms that reside in the real domain to the complex domain, such that the transpose operator $(\cdot)^T$ becomes the hermitian operator $(\cdot)^H$ in the complex domain. A consequence of this is that, the covariance matrix $E[\mathbf{x}\mathbf{x}^T]$ in $\mathbb{R}$ becomes $E[\mathbf{x}\mathbf{x}^H]$ in $\mathbb{C}$. However, an implicit assumption in using $E[\mathbf{x}\mathbf{x}^H]$ to describe the second order statistics of $\mathbf{x}$ is that the distribution of $\mathbf{x}$ is circular. However this assumption is not true for the majority of complex signals and any complex valued algorithm derived using this approach would not be optimal for the generality of complex signals.

Due to the lack of a rigorous mathematical framework for describing complex derivatives, gradients and statistics, the pace of development of complex valued adap-

tive filtering algorithms was initially very slow. The first step in remedying this was taken by Brandwood [28] in 1983. Brandwood introduced the $\mathbb{CR}$-calculus, and while $\mathbb{CR}$-calculus had been developed in 1927 by Wirtinger [29] (translated into English [30] in 1992), and exploited by the German speaking scientific community, it was not widely known in the English speaking scientific community. In addition to introducing the $\mathbb{CR}$-calculus, Brandwood also tackled the problem of minimizing a real function of complex variables by developing the complex gradient (also known as conjugate gradient) and the necessary conditions for a point to be stationary. Both of these results are essential for the developed of complex valued adaptive filtering algorithms.

The breakthrough in complex statistics (known as augmented complex statistics) came in 1993 with the work by Neeser [31] who showed that the covariance matrix only describes the full second order statistics of a subset of complex signals, known as proper complex signals. For the generality of complex signals, he showed that the pseudocovariance matrix is also needed to fully describe the second order statistics of a signal. Only proper (or circular) signals have vanishing pseudocovariance matrix and so have second order statistics fully described by their covariance matrix. This was soon followed in 1994 by Picinbono [32,33] who showed that to take advantage of the full second order statistics of complex signals, one must include the complex conjugate to form the widely linear model, i.e.

$$\mathbf{y} = \mathbf{h}\mathbf{x} + \mathbf{g}\mathbf{x}^*$$

where $\mathbf{y}$, $\mathbf{h}$ and $\mathbf{g}$, and $\mathbf{x}$ are respectively the output of the widely linear model, model coefficients and input signal.

The work carried out on complex gradients, statistics and widely linear models in the 1980s and early 1990s made it possible, for the first time, the development of adaptive filtering algorithms that are suitable for the processing of both circular (proper) and noncircular signals (improper). One of the first applications to take advantage of the widely linear model was in array processing of narrow band signals [34], where widely linear LQ (linear quadratic) filters were developed for optimum detection in non-Gaussian environments. This was later followed by the development of widely linear matched filters

for the detection of signals in the presence of improper complex noise [35] and widely linear structures for multiuser detection of code division multiple-access (CDMA) signals [36]. After showcasing the advantages that widely linear models can have in communication systems, the widely linear complex LMS (WLCLMS) [37], also referred to as the augmented CLMS (ACLMS), was developed for multiple access interference (MAI) suppression and later revisited in [38].

Most recently, using the widely linear model, real valued least squares algorithms have been extended to the complex domain. The widely linear complex RLS (recursive least squares) filter was developed in 2009 [39] and was shown to achieve significantly better performance than the standard RLS when applied to adaptive beamforming of mixed BPSK (binary phase-shift keying) and QPSK (quadrature phase-shift keying) signal transmissions. This was later followed by an efficient implementation of the widely linear complex RLS [40] and the widely linear affine projection algorithm [41] (which computationally and performance wise, lies between the WLCLMS and WLRLS).

For complex valued state space estimation, the widely linear complex Kalman filter was introduced. Unlike the WLCLMS, the Kalman filter considers the statistics of the noise and so can cater for the noncircularity of the noises as well as the widely linearity of the model [42]. To describe non linear systems, the widely linear complex extended and unscented Kalman filters have been developed [43–46].

Recent advances in sensor technology and wireless communication have led to an increasing use of sensor networks in practical applications, ranging from environmental monitoring to target localization. Distributed solutions to estimation problems within sensor networks have been emerging and growing in importance as they require less power for communication between the sensors and are more robust to faults in the network. Examples of distributed solutions include the distributed CLMS [47] and ACLMS [48], distributed RLS [49, 50] and distributed Kalman filter [51, 52].

1.1.2 Adaptive signal processing in $\mathbb{H}$

The development of quaternion valued signal processing algorithms is slowly following in the footsteps of complex valued signal processing. Though quaternions have been extensively used to describe rotations in applications ranging from graphic design to aeronautics (see Section 2.1.1), it is only in the 1990s that quaternion valued signal processing algorithms have been proposed, particularly in the area of digital filter design [53–55] and colour image processing [56–58]. More recently important signal processing tools have been extended to the quaternion domain, including the quaternion fourier transform [59–61], quaternion principal component analysis [62], quaternion MUSIC [63] and augmented statistics [64] [65]. In adaptive signal processing, the quaternion LMS (QLMS) [66] and widely linear QLMS (WLQLMS) [67] have been developed. Quaternions have also been employed in Kalman filtering [68, 69], though the Kalman filter itself is not quaternion valued and merely uses the quaternion representation to describe orientations.

A major stumbling block in the development of quaternion valued algorithms has been the lack of a quaternion gradient. While in the complex domain, the $\mathbb{CR}$-calculus has proved an invaluable tool for differentiating non-analytic function, such as the cost function $J = E[ee^*]$, no such tool exists in the quaternion domain. Furthermore, analyticity in the quaternion domain, as defined by the Fueter derivative [70, 71], is restricted to constants and linear functions [71]. For these reasons, quaternions have found few uses in adaptive signal processing. In the next section we elaborate on how the research done for this thesis aims to facilitate the future development of quaternion algorithm.

1.2 Motivation and aims

The focus of this work is two-fold; firstly, the development of a mathematical framework that facilitates the derivation of quaternion valued adaptive signal processing algorithms, and secondly, the extension of complex valued algorithms to the quaternion domain using augmented quaternion statistics, widely linear modelling and the recently introduced $\mathbb{HR}$-calculus and quaternion gradient. This thesis introduces the following contributions to

quaternion valued signal processing:

- The development of quaternion adaptive signal processing has been hampered by the lack of a mathematical framework for the differentiation of non-holomorphic functions, such as the typical cost function $J = E[ee^2]$. We have thus developed the $\mathbb{HR}$-calculus, which allows the differentiation of non-holomorphic functions and introduces eight derivatives, six of which don't exist in the complex domain. For the first time, the $\mathbb{HR}$-calculus shows that quaternion derivatives cannot be applied to quaternion functions of any dependent variables, but that quaternion function can be re written in the correct form, thus allowing direct differentiation with respect to the quaternion variables. The $\mathbb{HR}$-calculus also shows that variables q and q^*, are not independent, unlike in the complex domain.

- The quaternion gradient is derived and showed to be a generic extension of the complex gradient, also called the conjugate derivative. In addition to the quaternion gradient, we also introduce the I-gradient, allowing generic extensions of complex algorithms to be derived.

- Using the I-gradient we derive a simpler expression for the QLMS (the IQLMS), and show that all existing versions of the QLMS can be written in the same form. Convergence in the mean analysis confirms that all three algorithms have similar convergence rates, though the IQLMS offers a slightly faster convergence rate while requiring only half the mathematical operations.

- The $\mathbb{HR}$-calculus is used to derive the quaternion RLS and quaternion affine projection algorithm, and their widely linear counterpart. Mean square error analysis confirms the superiority of the widely linear algorithms and simulations on synthetic and real world signals demonstrates this.

- We derive the quaternion Kalman filter and widely linear Kalman filter. We show that since all nonlinear functions are non-holomorphic, the first order quaternion Taylor series must, in general, also be widely linear, and hence the extended Kalman filter too. Since Kalman filters are often used to track objects in three dimensional

space, we show that the widely linear model is over parameterized for three dimensional signals and instead propose a more efficient widely linear mode. To further simplify the widely linear Kalman filter, we also develop an efficient implementation of the widely linear Kalman filter that takes advantage of the structure built into the augmented covariance matrix.

1.3 Organisation of the thesis

The thesis is organised as follows. In Chapter 2 we introduce concepts that are key to the development of the work presented in this thesis. We start with an introduction to quaternions and quaternion algebra. Next we introduce the augmented statistics, circularity and widely linear modeling, for both complex and quaternion signals. We follow this with important concepts in complex differentiation, including analyticity, $\mathbb{CR}$-calculus and the gradient operator, that are necessary for the derivation of complex valued adaptive signal processing algorithms. Finally, we end the chapter, by providing a top level overview of adaptive filters, with an emphasis on the least mean square (LMS), affine projection algorithm (APA) and recursive least squares (RLS).

Chapter 3 represents the main body of this thesis, providing a unifying framework for the derivation of quaternion valued adaptive signal processing algorithms. The chapter starts by deriving the $\mathbb{HR}$-calculus and highlighting key differences with the $\mathbb{CR}$-calculus. Simple examples illustrate how the $\mathbb{HR}$-calculus can be used to calculate quaternion derivatives. Next, the quaternion gradient and I-gradient are introduced, and used to derive the quaternion Wiener solution, QLMS and IQLMS, as well as their widely linear counterparts. The performance of the existing QLMS algorithms and IQLMS are compared and convergence in the mean and mean square is performed on the IQLMS. Finally, we end the chapter with simulations on both synthetic and real world data, illustrating the performance of the IQLMS, QLMS and their widely linear counterpart.

In Chapter 4 we derive the widely linear quaternion RLS (QRLS). To bride the gap in computational complexity and performance between the QLMS and QRLS, Chapter

5 develops the quaternion APA (QAPA) and widely linear quaternion APA (WLQAPA), and the mean square error performance is investigated. Chapter 6 presents the quaternion Kalman filter (and its widely linear counterpart) and the widely linear quaternion extended Kalman filter. An efficient implementation of the quaternion widely linear model for three dimensional signals is also proposed. Chapter 7 provides conclusions and suggestions for future work.

Chapter 2

Background Theory

2.1 An overview of quaternions and their application

Quaternions were first introduced in 1843 by Sir William Rowan Hamilton [72]. The development of quaternion numbers resulted from Hamilton's desire to extend complex numbers into a third dimension. In the complex domain, complex numbers can be geometrically represented by points in a two dimensional space (this approach is credited to Caspar Wessel but popularized by Friedrich Gauss) and can be conveniently used to rotate vectors in a plane. With this geometric insight, Hamilton searched for a number system that could be used to describe points in a three- dimensional space and the relationship between two such points. The number system developed, named quaternion numbers, is defined as the quotient of two three-dimensional vectors, i.e.

$$q = \frac{\mathbf{a}}{\mathbf{b}} \tag{2.1}$$

where q is the quaternion operator, and **a** and **b** are three-dimensional vectors that can be of differing length and orientation.

A surprising outcome from the development of quaternions, is that quaternions consists of four numbers (as the name suggests), and not three as one would expect by directly extending the complex number system. This was one of the biggest hurdles faced by Hamilton in the development of quaternions. However, with hindsight, this result

is expected since three numbers are required to describe the axis of rotation in three-dimensional space, while an additional number is required to describe the angle of rotation. The scaling of the vector in three dimensional space can then be achieved by scaling the quaternion. Hamilton defined the quaternion as

$$q = q_r + iq_i + jq_j + kq_\kappa \tag{2.2}$$

where $iq_i + jq_j + kq_\kappa$ became known as the vector part and q_r as the scalar part.

2.1.1 Where and why are quaternions used?

In the late nineteenth-century, an intellectual battle took centre stage of the physics community [73]. On one side was Hamilton and his followers (the most avid one probably being Peter Guttie Tait) who firmly believed that quaternions should be used to describe three- and four-dimensional physical systems. On the other side were those (mainly Oliver Heaviside and Josiah Willard Gibbs) who thought that quaternions were too restrictive and that vector analysis offered a better framework for the analysis of multidimensional systems. Both quaternions and vector notation are largely interchangeable and many topics in physics and geometry (e.g. kinematics and Maxwell's equations) that are now described using vectors, were at the time described entirely using quaternions. Despite the best efforts of both Hamilton and Tait, vector analysis proved to be conceptually and notationally simpler, leading to the widespread adoption of vector analysis, at the expense of quaternions.

Though quaternions never became a standard for three-dimensional vector operations in Hamilton's or Tait's lifetime, they have recently experienced a considerable resurgence [74]. Quaternion's renaissance can be traced back to Ken Shoemake's 1985 paper on orientation and interpolation of 3D objects [75]. Instead of using Euler angles, Shoemake showed that there are important advantages to using quaternions to describe orientations and using the quaternion unit sphere to interpolate between them. Below are some of the reasons that quaternions have been so appealing to people working in computer graphics

- A rotation matrix $\mathbf{A}$ requires nine coefficients to relate two vectors in $\mathbb{R}^3$, however, physically only four parameters are needed (three for the axis of rotation and one for the angle of rotation) [76]. The four elements of a quaternion offer this physical insight - expressing straightforwardly the axis of rotation, angle of rotation, and scaling factor. This compact notation therefore requires less memory in hardware.

- For a rotation matrix $\mathbf{A}$ representing a succession of rotations in the x, y, z directions (using Euler angles), a degree of freedom can be lost if any two axis are aligned, resulting in the gimbal lock phenomenon. This is not a problem in the quaternion domain, where only one rotation around an axis of rotation is required.

- A quaternion rotation q_A is better conditioned than the real rotation matrix $\mathbf{A}$, as q_A is only required to be a unit quaternion whereas $\mathbf{A}$ must satisfy $\mathbf{A}^T\mathbf{A} = \mathbf{I}$ and $det(\mathbf{A}) = 1$. Due to finite precision of computer hardware, when many rotations are concatenated, the rotation matrix may become non-orthogonal and must be periodically re-adjusted. This normalization process is computationally more intensive when performed with rotation matrices than with quaternions [77].

- Using spherical linear interpolation [75], it becomes much easier to produce smooth interpolations of quaternion rotations, resulting in more realistic animations.

The landmark paper by Shoemake has not only encouraged the use of quaternion rotations in graphic design but also other areas where three dimensional rotations need to be manipulated, such as aeronautics [78–82], robotics [83–86] and molecular modeling [87]. So far, we have seen that quaternions have been found useful in applications involving rotations. Recently, quaternions have started to emerge in signal processing and adaptive signal processing where they are used to conveniently process three- and four- dimensional data sources. In Section 2.4 we provide an overview of adaptive signal processing and the role that quaternions have thus far played in three- and four-dimensional adaptive signal processing problems.

2.1.2 Quaternion algebra

Quaternions are a skew field over $\mathbb{R}^4$ defined as

$$\{q_r, q_\imath, q_\jmath, q_\kappa\} \in \mathbb{R}^4 \rightarrow q_r + \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa \in \mathbb{H} \tag{2.3}$$

The unit axis vectors $\imath$, $\jmath$ and κ are also the imaginary units, and obey the following rules (the top row describes a cyclic relationship)

$$\imath\jmath = \kappa \quad \jmath\kappa = \imath \quad \kappa\imath = \jmath$$

$$\imath^2 = \jmath^2 = \kappa^2 = \imath\jmath\kappa = -1$$

Note that quaternion multiplication is not commutative, that is, $\imath\jmath \neq \jmath\imath = -\kappa$.

A quaternion variable $q \in \mathbb{H}$ can be more conveniently expressed as [88]

$$q = Sq + Vq$$

where $Sq = q_r$ denotes the scalar part of q and $Vq = \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa$ the vector part. The product of quaternions $q_1, q_2 \in \mathbb{H}$ is given by

$$\begin{aligned} q_1 q_2 &= (Sq_1 + Vq_1)(Sq_2 + Vq_2) \\ &= Sq_1 Sq_2 - Vq_1 \bullet Vq_2 + Sq_2 Vq_1 + Sq_1 Vq_2 + Vq_1 \times Vq_2 \end{aligned}$$

where the symbol '$\bullet$' denotes the dot-product and '$\times$' the cross-product. Note the close relationship that exists between the products in vector analysis and quaternion product. It is due to the presence of the cross product that the quaternion product is not commutative. It is, however, associative and distributive. The quaternion conjugate, denoted by q^*, is given by

$$q^* = Sq - Vq = q_r - \imath q_\imath - \jmath q_\jmath - \kappa q_\kappa \tag{2.4}$$

and the norm $\| q \|$ is defined as

$$\| q \| = \sqrt{qq^*} = \sqrt{q_r^2 + q_\imath^2 + q_\jmath^2 + q_\kappa^2}$$

where $\| q_1 q_2 \| = \| q_1 \| \| q_2 \|$ and $\| \frac{q_1}{q_2} \| = \frac{\|q_1\|}{\|q_2\|}$. Similarly to the complex domain, the

inverse of a quaternion q can be written as

$$q^{-1} = \frac{q^*}{\| q \|}$$

where $(qp)^{-1} = p^{-1}q^{-1}$ and $qq^{-1} = q^{-1}q = 1$. It is important to note that due to the non-commutativity of the quaternion product, we have that $q_1 q_2^{-1} \neq q_2^{-1} q_1$

The three-dimensional vector part Vq is also called a *pure quaternion*, while the inclusion of the real part Sq gives a *full quaternion*. The algebraic structure of quaternions enables a unified processing of both three- and four-dimensional vector processes.

Quaternion involutions

Quaternion involutions are similarity relations, or self-inverse mappings, defined as [89]

$$\begin{aligned} q^{\imath} &= -\imath q \imath = q_r + \imath q_{\imath} - \jmath q_{\jmath} - \kappa q_{\kappa} \\ q^{\jmath} &= -\jmath q \jmath = q_r - \imath q_{\imath} + \jmath q_{\jmath} - \kappa q_{\kappa} \\ q^{\kappa} &= -\kappa q \kappa = q_r - \imath q_{\imath} - \jmath q_{\jmath} + \kappa q_{\kappa} \end{aligned} \tag{2.5}$$

and have the following properties (for $inv_3 \neq inv_2 \neq inv_1$)

$$P1: \quad (q^{inv})^{inv} = q \text{ for } inv \in \{\imath, \jmath, \kappa\} \tag{2.6}$$

$$P2: \quad (q_1 q_2)^{inv} = q_1^{inv} q_2^{inv} \tag{2.7}$$

$$P3: \quad (q_1 + q_2)^{inv} = q_1^{inv} + q_2^{inv} \tag{2.8}$$

$$P4: \quad (q^{inv_1})^{inv_2} = (q^{inv_2})^{inv_1} = q^{inv_3} \tag{2.9}$$

Unlike the complex domain, the quaternion conjugate is not an involution but an anti-involution, since $(q_1 q_2)^* = q_2^* q_1^*$. Involutions can be seen as a counterpart of the complex conjugate, as they allow for the components of a quaternion variable q to be expressed in

terms of the actual variable q and its 'partial conjugates' $q^{\imath}$, $q^{\jmath}$, q^{κ}, that is[1]

$$\begin{aligned} q_r &= \frac{1}{4}[q + q^{\imath} + q^{\jmath} + q^{\kappa}] \qquad q_{\imath} = \frac{1}{4\imath}[q + q^{\imath} - q^{\jmath} - q^{\kappa}] \\ q_{\jmath} &= \frac{1}{4\jmath}[q - q^{\imath} + q^{\jmath} - q^{\kappa}] \qquad q_{\kappa} = \frac{1}{4\kappa}[q - q^{\imath} - q^{\jmath} + q^{\kappa}] \end{aligned} \tag{2.10}$$

This representation will be instrumental in the derivation of both the $\mathbb{HR}$-calculus and the quaternion *widely linear* model, and underpins the development of quaternion valued adaptive filters for noncircular signals.

Quaternion rotations

To describe a rotation in three dimensional space, a quaternion q can be written as

$$q = cos(\theta/2) + \imath u_x sin(\theta/2) + \jmath u_y sin(\theta/2) + \kappa u_z sin(\theta/2) \tag{2.11}$$

where $\mathbf{u} = [u_x, u_y, u_z]$ is a unit vector defining the axis of rotation and θ is the angle of rotation. To rotate a three-dimensional vector $\mathbf{p} = [p_{\imath}, p_{\jmath}, p_{\kappa}]$ by the unit quaternion q, we apply the following transformation

$$\mathcal{R}(p) = qpq^* \tag{2.12}$$

where $p = 0 + \imath p_{\imath} + \jmath p_{\jmath} + \kappa p_{\kappa}$. Observe that the quaternion p is both pre- and post-multiplied to make $\mathcal{R}(p)$ a pure quaternion, i.e. a three-dimensional vector.

The quaternion rotation and quaternion involution are closely linked together since setting the quaternion q to perform a rotation of 180 degrees about the i-axis, that is,

$$q = cos(\pi/2) + 1\imath sin(\pi/2) + 0\jmath sin(\pi/2) + 0\kappa sin(\pi/2) = \imath \tag{2.13}$$

yields the rotation

$$\mathcal{R}(p) = qpq^* = \imath p\imath^* = -\imath p\imath \tag{2.14}$$

[1]Compare with the complex domain where the real and imaginary parts of the complex numbers $z = x + \imath y$ are expressed by $x = \frac{1}{2}(z + z^*)$ and $y = \frac{1}{2\imath}(z - z^*)$.

which is equal to i-involution. A similar result can be obtained for the other two involutions.

2.2 Augmented statistics, circularity and widely linear modeling

In this section we introduce three concepts that are intimately related and which form the backbone of this thesis. First, we introduce each concept in the complex domain version, allowing the reader to become familiar with the complex domain results before proceeding to the quaternion domain.

2.2.1 Augmented statistics

Complex augmented statistics

In the real domain, the second order statistics of a real random variable $\mathbf{x} \in \mathbb{R}^N$ is given by the covariance matrix $E[\mathbf{x}\mathbf{x}^T] \in \mathbb{R}^{N \times N}$. In the complex domain, however, it was shown in 1993 by Neeser et al. [31] and subsequently by Picinbono [90, 91] that the covariance matrix does not provide the full second order statistics of a complex random variable $\mathbf{z} = \mathbf{z}_r + \imath\mathbf{z}_\imath \in \mathbb{C}^N$. To introduce the augmented statistics we start by noting that the complex random variable $\mathbf{z}$ can be written as a real vector $\mathbf{x} = [\mathbf{z}_r, \mathbf{z}_\imath]^T \in \mathbb{C}^{2N}$ containing the exact same information as the complex vector $\mathbf{z}$. Using the duality that exists between the complex and real domain [45, 92, 93], we can rewrite the real vector $\mathbf{x}$ as

$$\mathbf{z}^a = \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^* \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I} & \imath\mathbf{I} \\ \mathbf{I} & -\imath\mathbf{I} \end{bmatrix}}_{\mathbf{J} \in \mathbb{C}^{2N \times 2N}} \underbrace{\begin{bmatrix} \mathbf{z}_r \\ \mathbf{z}_\imath \end{bmatrix}}_{\mathbf{x}} \tag{2.15}$$

where $\mathbf{z}^a$ is the augmented complex vector and $\mathbf{J}$ is a transformation matrix that maps a point from the real domain to complex domain. The mapping $\mathbf{J}$ is invertible, with $\mathbf{J}^{-1} = \frac{1}{2}\mathbf{J}^H$. It follows from this linear mapping that both $\mathbf{z}$ and $\mathbf{z}^*$ are needed to describe

$\mathbf{z}_r$ and $\mathbf{z}_\imath$. This becomes more clear for the scalar case where we have

$$z_r = \frac{1}{2}(z + z^*) \qquad\qquad z_\imath = \frac{1}{2\imath}(z - z^*) \tag{2.16}$$

We can now obtain the augmented covariance matrix $\mathbf{R}^a_{zz}$ as [45, 93]

$$\mathbf{R}^a_{zz} = E\left[\begin{bmatrix}\mathbf{z}\\ \mathbf{z}^*\end{bmatrix}\begin{bmatrix}\mathbf{z}, \mathbf{z}^*\end{bmatrix}^H\right] = \begin{bmatrix}\mathbf{R}_{zz} & \mathbf{P}_{zz}\\ \mathbf{P}^*_{zz} & \mathbf{R}^*_{zz}\end{bmatrix} \tag{2.17}$$

where $\mathbf{R}_{zz} = E[\mathbf{z}\mathbf{z}^H]$ and $\mathbf{P}_{zz} = E[\mathbf{z}\mathbf{z}^T]$ are respectively the covariance and pseudo-covariance matrices.

Quaternion augmented statistics

Following from the complex domain, the quaternion augmented statistics is developed by first establishing a relationship between the four real components $\mathbf{q}_r$, $\mathbf{q}_\imath$, $\mathbf{q}_\jmath$ and $\mathbf{q}_\kappa$ of a quaternion random variable $\mathbf{q} = \mathbf{q}_r + \imath\mathbf{q}_\imath + \jmath\mathbf{q}_\jmath + \kappa\mathbf{q}_\kappa \in \mathbb{H}^N$ and the variable $\mathbf{q}$. As shown in [64], by forming an augmented vector $\mathbf{q}^a = [\mathbf{q}^T, \mathbf{q}^{\imath T}, \mathbf{q}^{\jmath T}, \mathbf{q}^{\kappa T}] \in \mathbb{H}^{4N}$ the following linear mapping can be obtained between $\mathbb{R}^4$ and $\mathbb{H}$

$$\underbrace{\begin{bmatrix}\mathbf{q}\\ \mathbf{q}^\imath\\ \mathbf{q}^\jmath\\ \mathbf{q}^\kappa\end{bmatrix}}_{\mathbf{q}^a \in \mathbb{H}^{4N}} = \underbrace{\begin{bmatrix}\mathbf{I} & \imath\mathbf{I} & \jmath\mathbf{I} & \kappa\mathbf{I}\\ \mathbf{I} & \imath\mathbf{I} & -\jmath\mathbf{I} & -\kappa\mathbf{I}\\ \mathbf{I} & -\imath\mathbf{I} & \jmath\mathbf{I} & -\kappa\mathbf{I}\\ \mathbf{I} & -\imath\mathbf{I} & -\jmath\mathbf{I} & \kappa\mathbf{I}\end{bmatrix}}_{\mathbf{J} \in \mathbb{H}^{4N\times 4N}} \underbrace{\begin{bmatrix}\mathbf{q}_r\\ \mathbf{q}_\imath\\ \mathbf{q}_\jmath\\ \mathbf{q}_\kappa\end{bmatrix}}_{\mathbf{q}^r \in \mathbb{R}^{4N}} \tag{2.18}$$

where $\mathbf{I} \in \mathbb{R}^{N\times N}$ is the identity matrix. The mapping $\mathbf{J}$ is invertible with

$$\mathbf{J}^{-1} = \frac{1}{4}\mathbf{J}^H$$

It follows from this duality between $\mathbf{q}^a$ and $\mathbf{q}^r$, that to capture the full second order statistics of a quaternion random vector, one must use the augmented vector $\mathbf{q}^a$ to obtain

the augmented covariance matrix [64]

$$\mathbf{R}^a_{qq} = E[\mathbf{q}^a \mathbf{q}^{aH}] = \begin{bmatrix} \mathbf{R}_{qq} & \mathbf{P}_{qq} & \mathbf{S}_{qq} & \mathbf{T}_{qq} \\ \mathbf{P}^{\imath}_{qq} & \mathbf{R}^{\imath}_{qq} & \mathbf{T}^{\imath}_{qq} & \mathbf{S}^{\imath}_{qq} \\ \mathbf{S}^{\jmath}_{qq} & \mathbf{T}^{\jmath}_{qq} & \mathbf{R}^{\jmath}_{qq} & \mathbf{P}^{\jmath}_{qq} \\ \mathbf{T}^{\kappa}_{qq} & \mathbf{S}^{\kappa}_{qq} & \mathbf{P}^{\kappa}_{qq} & \mathbf{R}^{\kappa}_{qq} \end{bmatrix} \tag{2.19}$$

where $\mathbf{R}_{qq} = E[\mathbf{x}\mathbf{x}^H]$ is the covariance matrix and $\mathbf{P}_{qq} = E[\mathbf{x}\mathbf{x}^{\imath H}]$, $\mathbf{S}_{qq} = E[\mathbf{x}\mathbf{x}^{\jmath H}]$ and $\mathbf{T}_{qq} = E[\mathbf{x}\mathbf{x}^{\kappa H}]$ are respectively the $\imath$-covariance, $\jmath$-covariance and κ-covariance matrices. The latter three covariance matrices are collectively known as the pseudocovariance matrices and are respectively $\imath$-Hermitian, $\jmath$-Hermitian and κ-Hermitian, that is

$$\begin{aligned} \mathbf{P}_{qq} &= \mathbf{P}^{\imath H}_{qq} \\ \mathbf{S}_{qq} &= \mathbf{S}^{\jmath H}_{qq} \\ \mathbf{T}_{qq} &= \mathbf{T}^{\kappa H}_{qq} \end{aligned}$$

It can be shown that the augmented covariance matrix is related to the real covariance matrix $\mathbf{R}^r_{qq} = E[\mathbf{q}^r \mathbf{q}^{rT}]$ by the mapping $\mathbf{J}$ [64]

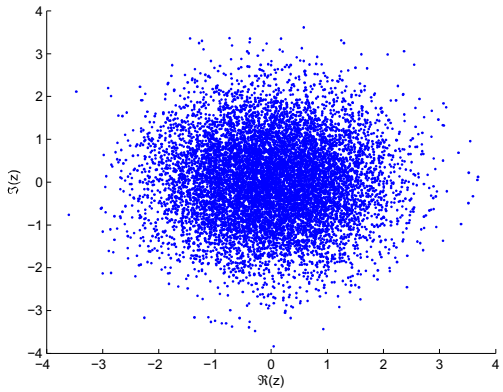
$$\mathbf{R}^r_{qq} = \frac{1}{16}\mathbf{J}^H \mathbf{R}^a_{qq} \mathbf{J}$$

This confirms that the full second order statistics of a quaternion random variable is indeed given by the augmented covariance matrix.

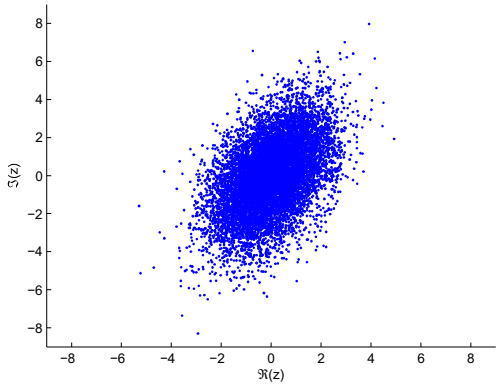
2.2.2 Circularity

Complex circularity

A random variable $\mathbf{z}$ is said to be second order circular if it has a rotation invariant probability distribution (see Figure 2.1). In other words, a complex random variable $\mathbf{z}$ is second order circular if $\mathbf{z}$ and $\mathbf{z}^\theta = \mathbf{z}e^{\jmath\theta}$, have, for any θ, the same probability distribution. The circularity of a random variable is closely related to its pseudocovariance. To see this,



(a) Circular distribution



(b) Noncircular distribution

Figure 2.1: Probability distribution of circular and noncircular complex Gaussian random variable.

take the pseudocovariance of $\mathbf{z}e^{j\theta}$

$$\mathbf{P}_{z^\theta z^\theta} = E[\mathbf{z}e^{\jmath\theta}\mathbf{z}^T e^{\jmath\theta}] = E[\mathbf{z}\mathbf{z}^T]e^{\jmath 2\theta} = \mathbf{P}_{zz}e^{\jmath 2\theta} \tag{2.20}$$

It follows that to satisfy $\mathbf{P}_{z^\theta z^\theta} = \mathbf{P}_{zz}$ we must have $\mathbf{P}_{zz} = \mathbf{0}$ [90]. A commonly misunderstood point in the literature is that although all circular complex random variables have $\mathbf{P}_{zz} = \mathbf{0}$, the converse is not true, i.e. not all random variables with $\mathbf{P}_{zz} = \mathbf{0}$ are circular [94]. A random variable that satisfies $\mathbf{P}_{zz} = \mathbf{0}$ (and which therefore has rotation invariant second order moments) is called proper. Circularity is a stricter condition requiring the probability distribution to be circular, and hence all moments.

On the other hand, as shown below the covariance matrix is always rotation invari-

ant.

$$\mathbf{R}_{z^{\theta}z^{\theta}} = E[\mathbf{z}e^{\jmath\theta}\mathbf{z}^{H}e^{-\jmath\theta}] = E[\mathbf{z}\mathbf{z}^{H}] = \mathbf{R}_{zz} \tag{2.21}$$

By rewriting the pesudocovariance matrix as

$$\mathbf{P}_{zz} = E[(\mathbf{z}_r + j\mathbf{z}_i)((\mathbf{z}_r + j\mathbf{z}_i)^T)] = \underbrace{E[\mathbf{z}_r\mathbf{z}_r^T]}_{\mathbf{R}_{rr}} - \underbrace{E[\mathbf{z}_i\mathbf{z}_i^T]}_{\mathbf{R}_{ii}} + \jmath\underbrace{E[\mathbf{z}_r\mathbf{z}_i^T]}_{\mathbf{R}_{ri}} + \jmath\underbrace{E[\mathbf{z}_i\mathbf{z}_r^T]}_{\mathbf{R}_{ir}} \tag{2.22}$$

we observe that for $\mathbf{P}_{zz}$ to vanish the following conditions must be satisfied [31]

$$\mathbf{R}_{rr} = \mathbf{R}_{ii} \qquad \text{and} \qquad \mathbf{R}_{ri} = -\mathbf{R}_{ri}^T \tag{2.23}$$

This is equivalent to saying that the real and imaginary part of the random variable $\mathbf{z}$ have identical autocovariance matrices and their crosscovariance matrix is skew symmetric. To be skew symmetric, $\mathbf{R}_{ri}$ must have a diagonal of zeros, implying that the real and imaginary part of each element of $\mathbf{z}$ must be uncorrelated. This however does not imply that the real part of one element of $\mathbf{z}$ must be uncorrelated to the imaginary part of another element of $\mathbf{z}$. For a scalar random variable z, the condition above simplify into

$$E[z_r^2] = E[z_i^2] \qquad \text{and} \qquad E[z_r z_i] = 0 \tag{2.24}$$

i.e. z_r and z_i must have the same power and be uncorrelated.

Quaternion circularity

In analogy with the complex domain, a quaternion random variable $\mathbf{q}$ is circular if it has a probability distribution that is rotation invariant with respect to the six possible pairs of axes $\{1,\imath\}$,$\{1,\jmath\}$,$\{1,\kappa\}$, $\{\imath,\jmath\}$,$\{\imath,\kappa\}$,$\{\jmath,\kappa\}$. Since the $\imath$-, $\jmath$-, and κ-covariance matrices are not rotation invariant, circularity implies that the pseudocovariance matrices must vanish. The converse however is not true, i.e. a quaternion random vector with pseudocovariance matrices equal to zero is not necessarily circular. If all pseudocovariance matrices are equal to zero, the signal is called Q-proper [64,65], and the full second order statistics are captured by the covariance matrix.

	$\mathbf{P}_{qq}$	$\mathbf{S}_{qq}$	$\mathbf{T}_{qq}$
$\Re\{\cdot\}$	$E[\mathbf{q}_r\mathbf{q}_r^T] + E[\mathbf{q}_\imath\mathbf{q}_\imath^T]$ $-E[\mathbf{q}_\jmath\mathbf{q}_\jmath^T] - E[\mathbf{q}_\kappa\mathbf{q}_\kappa^T]$	$E[\mathbf{q}_r\mathbf{q}_r^T] - E[\mathbf{q}_\imath\mathbf{q}_\imath^T]$ $+E[\mathbf{q}_\jmath\mathbf{q}_\jmath^T] - E[\mathbf{q}_\kappa\mathbf{q}_\kappa^T]$	$E[\mathbf{q}_r\mathbf{q}_r^T] - E[\mathbf{q}_\imath\mathbf{q}_\imath^T]$ $-E[\mathbf{q}_\jmath\mathbf{q}_\jmath^T] + E[\mathbf{q}_\kappa\mathbf{q}_\kappa^T]$
$\Im_\imath\{\cdot\}$	$E[\mathbf{q}_\imath\mathbf{q}_r^T] - E[\mathbf{q}_r\mathbf{q}_\imath^T]$ $+E[\mathbf{q}_\jmath\mathbf{q}_\kappa^T] - E[\mathbf{q}_\kappa\mathbf{q}_\jmath^T]$	$E[\mathbf{q}_\imath\mathbf{q}_r^T] + E[\mathbf{q}_r\mathbf{q}_\imath^T]$ $+E[\mathbf{q}_\kappa\mathbf{q}_\jmath^T] + E[\mathbf{q}_\jmath\mathbf{q}_\kappa^T]$	$E[\mathbf{q}_\imath\mathbf{q}_r^T] + E[\mathbf{q}_r\mathbf{q}_\imath^T]$ $-E[\mathbf{q}_\jmath\mathbf{q}_\kappa^T] - E[\mathbf{q}_\kappa\mathbf{q}_\jmath^T]$
$\Im_\jmath\{\cdot\}$	$E[\mathbf{q}_r\mathbf{q}_\jmath^T] + E[\mathbf{q}_\jmath\mathbf{q}_r^T]$ $-E[\mathbf{q}_\kappa\mathbf{q}_\imath^T] - E[\mathbf{q}_\imath\mathbf{q}_\kappa^T]$	$E[\mathbf{q}_\jmath\mathbf{q}_r^T] - E[\mathbf{q}_r\mathbf{q}_\jmath^T]$ $-E[\mathbf{q}_\kappa\mathbf{q}_\imath^T] + E[\mathbf{q}_\imath\mathbf{q}_\kappa^T]$	$E[\mathbf{q}_r\mathbf{q}_\jmath^T] + E[\mathbf{q}_\jmath\mathbf{q}_r^T]$ $+E[\mathbf{q}_\kappa\mathbf{q}_\imath^T] + E[\mathbf{q}_\imath\mathbf{q}_\kappa^T]$
$\Im_\kappa\{\cdot\}$	$E[\mathbf{q}_\kappa\mathbf{q}_r^T] + E[\mathbf{q}_r\mathbf{q}_\kappa^T]$ $+E[\mathbf{q}_\imath\mathbf{q}_\jmath^T] + E[\mathbf{q}_\jmath\mathbf{q}_\imath^T]$	$E[\mathbf{q}_\kappa\mathbf{q}_r^T] + E[\mathbf{q}_r\mathbf{q}_\kappa^T]$ $-E[\mathbf{q}_\imath\mathbf{q}_\jmath^T] - E[\mathbf{q}_\jmath\mathbf{q}_\imath^T]$	$E[\mathbf{q}_\kappa\mathbf{q}_r^T] - E[\mathbf{q}_r\mathbf{q}_\kappa^T]$ $+E[\mathbf{q}_\imath\mathbf{q}_\jmath^T] - E[\mathbf{q}_\jmath\mathbf{q}_\imath^T]$

Table 2.1: The pesudocovariance matrices written in terms of the real components $\mathbf{q}_r$, $\mathbf{q}_\imath$, $\mathbf{q}_\jmath$ and $\mathbf{q}_\kappa$.

It order to understand the mathematical structure that is needed for a quaternion random variable to be circular, Table 2.1 breaks down the pseudocovariance matrices into their real components $\mathbf{q}_r$, $\mathbf{q}_\imath$, $\mathbf{q}_\jmath$ and $\mathbf{q}_\kappa$. It now follows that for all three pseudocovariances to vanish, the following conditions must be satisfied [65]

$$\begin{aligned}
E[\mathbf{q}_r\mathbf{q}_r^T] = E[\mathbf{q}_\imath\mathbf{q}_\imath^T] = E[\mathbf{q}_\jmath\mathbf{q}_\jmath^T] = E[\mathbf{q}_\kappa\mathbf{q}_\kappa^T] \\
E[\mathbf{q}_\imath\mathbf{q}_r^T] = -E[\mathbf{q}_r\mathbf{q}_\imath^T] = -E[\mathbf{q}_\jmath\mathbf{q}_\kappa^T] = E[\mathbf{q}_\kappa\mathbf{q}_\jmath^T] \\
E[\mathbf{q}_\jmath\mathbf{q}_r^T] = -E[\mathbf{q}_r\mathbf{q}_\jmath^T] = E[\mathbf{q}_\imath\mathbf{q}_\kappa^T] = -E[\mathbf{q}_\kappa\mathbf{q}_\imath^T] \\
E[\mathbf{q}_\kappa\mathbf{q}_r^T] = -E[\mathbf{q}_r\mathbf{q}_\kappa^T] = -E[\mathbf{q}_\imath\mathbf{q}_\jmath^T] = E[\mathbf{q}_\jmath\mathbf{q}_\imath^T]
\end{aligned} \tag{2.25}$$

Observe that these conditions are more restrictive than those in the complex domain (see equation 2.23), where in the complex domain the covariance matrices must be equal, and the cross-covariance matrices must be skew symmetric.

2.2.3 Widely linear model

Complex widely linear model

Consider a real valued mean square error (MSE) estimator given by [95]

$$\hat{y} = E[y \mid \mathbf{x}]$$

where $\hat{y}$ is the estimated process and x the observed variable. For jointly Gaussian x and y, the optimal solution is a linear estimator, given by

$$\hat{y} = \mathbf{h}^T\mathbf{x} \tag{2.26}$$

where $\mathbf{h}$ and $\mathbf{x}$ are respectively the coefficient and regressor vector. For the standard complex domain MSE estimator it is also assumed that $\hat{y} = E[y\,|\,\mathbf{x}]$, leading to the strictly linear model

$$\hat{y} = \mathbf{h}^T\mathbf{x} \tag{2.27}$$

However,

$$\hat{y}_r = E[y_r\,|\,\mathbf{x}_r, \mathbf{x}_\imath] \qquad\qquad \hat{y}_\imath = E[y_\imath\,|\,\mathbf{x}_r, \mathbf{x}_\imath]$$

and since $x_r = \frac{x+x^*}{2}$ and $x_\imath = \frac{x-x^*}{2\imath}$, the complex *widely linear* model is given by [91]

$$\hat{y} = E[y\,|\,\mathbf{x}, \mathbf{x}^*] \Rightarrow y = \mathbf{h}^T\mathbf{x} + \mathbf{g}^T\mathbf{x}^*$$

that is, it comprises both the strictly linear part $\mathbf{h}^T\mathbf{x}$ and the conjugate part $\mathbf{g}^T\mathbf{x}^*$, where $\mathbf{g}$ is a coefficient vector.

To establish the conditions under which the widely linear model is superior to the strictly linear model, we start with expressions for the MSE (mean square error) δe_l^2 of the strictly linear model [32]

$$\delta e_l^2 = E[|y|^2] - \mathbf{r}^H\mathbf{R}_{xx}^{-1} \tag{2.28}$$

where $\mathbf{r} = E[y^*\mathbf{x}]$ and $\mathbf{R}_{xx} = E[\mathbf{x}\mathbf{x}^H]$, and the MSE δe_{wl}^2 of the widely linear model

$$\delta e_{wl}^2 = E[|y|^2] - (\mathbf{u}^H\mathbf{r} + \mathbf{v}^H\mathbf{s}^*) \tag{2.29}$$

where $\mathbf{s} = E[y\mathbf{x}]$, and

$$\begin{aligned}\mathbf{u} &= [\mathbf{R}_{xx} - \mathbf{P}_{xx}\mathbf{R}_{xx}^{-1*}\mathbf{P}_{xx}^*][\mathbf{r} - \mathbf{P}_{xx}\mathbf{R}_{xx}^{-1*}\mathbf{s}^*]\\ \mathbf{v} &= [\mathbf{R}_{xx}^* - \mathbf{P}_{xx}^*\mathbf{R}_{xx}^{-1}\mathbf{P}_{xx}][\mathbf{s}^* - \mathbf{P}_{xx}^*\mathbf{R}_{xx}^{-1}\mathbf{r}_{xx}]\end{aligned}$$

Subtracting δe_{wl} in (2.29) from δe_l in (2.28), Picinbono et al. [32] showed that the MSE difference between the widely linear model and strictly linear model can be written

as

$$\begin{aligned}\delta e &= \delta e_l - \delta e_{wl} \\ &= [\mathbf{s}^* - \mathbf{P}^*_{xx}\mathbf{R}^{-1}_{xx}\mathbf{r}]^H[\mathbf{R}^*_{xx} - \mathbf{P}^*_{xx}\mathbf{R}^{-1}_{xx}\mathbf{P}_{xx}]^{-1}[\mathbf{s}^* - \mathbf{P}^*_{xx}\mathbf{R}^{-1}_{xx}\mathbf{r}] \qquad (2.30)\end{aligned}$$

Since the matrix $[\mathbf{R}^*_{xx} - \mathbf{P}^*_{xx}\mathbf{R}^{-1}_{xx}\mathbf{P}_{xx}]$ is positive definite, the expression above is always greater or equal to 0 and hence the widely linear model always performs at least as well as the strictly linear model (as measured by the MSE). The MSE of the strictly linear model is only equal to that of the widely linear model when $\mathbf{s}^* - \mathbf{P}^*_{xx}\mathbf{R}^{-1}_{xx}\mathbf{r} = \mathbf{0}$,i.e. when $\mathbf{P}_{xx} = 0$ and $\mathbf{s} = 0$. This condition is known as jointly circular.

Quaternion widely linear model

In analogy with the complex domain, the MSE estimator $\hat{y}$ of an observed quaternion variable $\mathbf{q}$ can be written as [64]

$$\hat{y} = E[y_r \mid \mathbf{x}_r, \mathbf{x}_\imath, \mathbf{x}_\jmath, \mathbf{x}_\kappa] + \imath E[y_\imath \mid \mathbf{x}_r, \mathbf{x}_\imath, \mathbf{x}_\jmath, \mathbf{x}_\kappa] + \jmath E[y_\jmath \mid \mathbf{x}_r, \mathbf{x}_\imath, \mathbf{x}_\jmath, \mathbf{x}_\kappa] + \kappa E[y_\kappa \mid \mathbf{x}_r, \mathbf{x}_\imath, \mathbf{x}_\jmath, \mathbf{x}_\kappa]$$

Using the involutions in (2.10), we can substitute for the real variables $q_r, \ldots, q_\kappa$ to obtain

$$\hat{y} = E[y_r \mid \mathbf{x}, \mathbf{x}^\imath, \mathbf{x}^\jmath, \mathbf{x}^\kappa] + \imath E[y^\imath \mid \mathbf{x}, \mathbf{x}^\imath, \mathbf{x}^\jmath, \mathbf{x}^\kappa] + \jmath E[y^\jmath \mid \mathbf{x}, \mathbf{x}^\imath, \mathbf{x}^\jmath, \mathbf{x}^\kappa] + \kappa E[y^\kappa \mid \mathbf{x}, \mathbf{x}^\imath, \mathbf{x}^\jmath, \mathbf{x}^\kappa]$$

The quaternion widely linear model then becomes

$$\hat{y} = \mathbf{w}^{aH}\mathbf{x}^a = \mathbf{u}^H\mathbf{x} + \mathbf{v}^H\mathbf{x}^\imath + \mathbf{g}^H\mathbf{x}^\jmath + \mathbf{h}^H\mathbf{x}^\kappa$$

where the augmented weight vector $\mathbf{w}^a = [\mathbf{u}^T, \mathbf{v}^T, \mathbf{g}^T, \mathbf{h}^T]^T$ and the augmented input vector $\mathbf{x}^a = [\mathbf{x}^T, \mathbf{x}^{\imath T}, \mathbf{x}^{\jmath T}, \mathbf{x}^{\kappa T}]^T$.

2.3 Complex differentiability

Adaptive filtering algorithm typically involves the minimization of a cost function such as $J = e(k)e^*(k) = e_r^2(k) + e_i^2(k)$, where $e(k)$ is a measure of the output error. The cost function J is not however analytic based on the Cauchy-Riemann equations [45] and so cannot be differentiated using standard complex domain differentiation tools. In this section we present the tools that are needed to calculate the gradient of cost functions such as $J = e(k)e^*(k)$. For most of this section, quaternion extensions of the results presented are a subject of the research carried out for this thesis and hence are not presented here, but in later chapters.

2.3.1 Analytic functions

A complex function $f(z) = u(x, y) + iv(x, y)$ with derivative

$$\frac{\partial f}{\partial z} = \lim_{z \to z_0} \frac{f(z) - f(z_0)}{z - z_0} \tag{2.31}$$

at a point z_0, is said to be differentiable if the limit is independent of the direction taken by z to approach z_0. A complex function f is said to be holomorphic at a point z if it is differentiable at all points in some neighbourhood $\varepsilon > 0$ of z_0 [96]. A function f that is holomorphic on the whole complex plane is said to be entire, while points that are not holomorphic are called singular points. In the complex domain, unlike the real domain, any holomorphic function may be locally expanded in a power series, and is therefore infinitely differentiable, i.e. analytic [97]. It follows from this that in the complex domain the terms holomorphic and analytic describe the same functions and so they can be used interchangeably, though for consistency, in this thesis we will be using the term holomorphic.

Cauchy-Riemann equations

A function $f(z) = u(x, y) + \imath v(x, y)$ is differentiable at a point if the partial derivatives of $u(x, y)$ and $v(x, y)$ exist, and the following Cauchy-Riemann equations hold

$$\frac{\partial u(x,y)}{\partial x} = \frac{\partial v(x,y)}{\partial y} \qquad \text{and} \qquad \frac{\partial u(x,y)}{\partial y} = -\frac{\partial x(x,y)}{\partial x} \tag{2.32}$$

In optimisation problems, one often encounters complex functions which are not holomorphic. Take for example the cost function $J = z(k)z^*(k) = x^2(k) + y^2(k)$, where

$$\frac{\partial u(x,y)}{\partial x} = 2x \ , \ \frac{\partial u(x,y)}{\partial y} = 2y \ , \ \frac{\partial v(x,y)}{\partial y} = 0 \ , \ \frac{\partial v(x,y)}{\partial y} = 0 \tag{2.33}$$

In this case, the Cauchy-Riemann equation is only satisfied at the point (0,0), and hence the cost function $J = z(k)z^*(k) = x^2(k) + y^2(k)$ is nowhere holomorphic. As seen from this simple example, the Cauchy-Riemann conditions impose a very strict structure on complex functions. However, if $f(z) : \mathbb{C} \to \mathbb{C}$ is viewed as a bivariate function $g(x, y) : \mathbb{R}^2 \to \mathbb{R}^2$, then the mere existence of the partial derivatives of $u(x, y)$ and $v(x, y)$ is a necessary and sufficient for a stationary point to exist [98]. For example, the function $J = z(k)z^*(k) = x^2(k) + y^2(k)$ may not be holomorphic in z but it is holomorphic in x and y. Using this duality that exists between the $\mathbb{R}^2$ and $\mathbb{C}$ vector space, we next show how the $\mathbb{CR}$-calculus provides an alternative way to differentiating complex functions.

2.3.2 The $\mathbb{CR}$-calculus

The $\mathbb{CR}$-calculus [28, 30, 98], (also known as Wirtinger calculus, after W. Wirtinger) provides a framework for differentiating nonanalytic functions by treating the function $f(z, z^*) : \mathbb{C}^2 \to \mathbb{C}^2$ as equivalent to the function $g(x, y) : \mathbb{R}^2 \to \mathbb{R}^2$ and finding an expression such that the real partial derivatives are a function $\Phi(\cdot)$ of the complex derivatives,

i.e.

$$\frac{\partial g(x,y)}{\partial x} = \Phi\left(\frac{\partial f(z,z^*)}{\partial z}, \frac{\partial f(z,z^*)}{\partial z^*}\right)$$
$$\frac{\partial g(x,y)}{\partial y} = \Phi\left(\frac{\partial f(z,z^*)}{\partial z}, \frac{\partial f(z,z^*)}{\partial z^*}\right)$$

We now solve the two expressions above by following the approach taken in [28, 45].

Let $f(z) = f(z,z^*) = u(x,y) + \jmath v(x,y) : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ be a complex function of complex variables z and z^*, and $g(x,y) : \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ be a complex function of real variables x and y, such that $f(z,z^*) = g(x,y)$. Starting with the total differential of $g(x,y)$ as

$$dg(x,y) = \frac{\partial g(x,y)}{\partial x}dx + \frac{\partial g(x,y)}{\partial y}dy \tag{2.34}$$

our aim is to get the total differential into a form that is comparable to the total differential of $f(z,z^*)$. Substituting for $g(x,y)$ we have

$$dg(x,y) = \frac{\partial u(x,y)}{\partial x}dx + \imath\frac{\partial v(x,y)}{\partial x}dx + \frac{\partial u(x,y)}{\partial y}dy + \imath\frac{\partial v(x,y)}{\partial y}dy$$

and making the variable swap

$$dx = \frac{1}{2}(dz + dz^*) \qquad\qquad dy = \frac{1}{2\imath}(dz - dz^*)$$

we obtain

$$\begin{aligned} dg(x,y) &= \frac{1}{2}\left(\frac{\partial u(x,y)}{\partial x} + \frac{\partial v(x,y)}{\partial y} + \imath\left[\frac{\partial v(x,y)}{\partial x} - \frac{\partial u(x,y)}{\partial y}\right]\right)dz \\ &\quad + \frac{1}{2}\left(\frac{\partial u(x,y)}{\partial x} - \frac{\partial v(x,y)}{\partial y} + \imath\left[\frac{\partial v(x,y)}{\partial x} + \frac{\partial u(x,y)}{\partial y}\right]\right)dz^* \end{aligned}$$

Simplifying $dg(x,y)$ above yields

$$dg(x,y) = \frac{1}{2}\left(\frac{\partial g(x,y)}{\partial x} - \imath\frac{\partial g(x,y)}{\partial y}\right)dz + \frac{1}{2}\left(\frac{\partial g(x,y)}{\partial x} + \jmath\frac{\partial g(x,y)}{\partial y}\right)dz^*$$

If we now compare the differential $dg(x,y)$ above with the differential of the complex

function $f(z, z^*)$

$$df(z, z^*) = \frac{\partial f(z)}{\partial z} dz + \frac{\partial f(z)}{\partial z^*} dz^*$$

we observe that the following must be true

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - \jmath \frac{\partial f}{\partial y}\right) \tag{2.35}$$

$$\frac{\partial f}{\partial z^*} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + \jmath \frac{\partial f}{\partial y}\right) \tag{2.36}$$

where $\frac{\partial f(z)}{\partial z}$ is called the $\mathbb{R}$-derivative and $\frac{\partial f(z)}{\partial z^*}$ the $\mathbb{R}^*$-derivative. In the definitions above, z and z^* are treated as independent variables and so the partial derivatives $\frac{\partial f}{\partial z}$ and $\frac{\partial f}{\partial z^*}$ can be used in the same way as any partial derivative of a real multi-variate function. To appreciate this, notice how z^* can be regarded as a constant when differentiating with respect to z, and how z can be regarded as a constant when differentiating with respect to z^*, i.e.

$$\frac{\partial z^*}{\partial z} = \frac{\partial z}{\partial z^*} = 0 \tag{2.37}$$

Note however that this is only a formalism since z cannot be varied while keeping z^* constant.

For analytic functions, the $\mathbb{R}$- and $\mathbb{R}^*$- derivatives are intimately related to the Cauchy-Riemann derivatives. To see this we rewrite the $\mathbb{R}$- and $\mathbb{R}^*$- derivatives by substituting for $f(z) = u(x, y) + \jmath v(x, y)$

$$\frac{\partial f(z)}{\partial z} = \frac{1}{2}\left(\frac{\partial u(x,y)}{\partial x} + \jmath\frac{\partial v(x,y)}{\partial x} - \jmath\frac{\partial u(x,y)}{\partial y} + \frac{\partial v(x,y)}{\partial y}\right) \tag{2.38}$$

$$\frac{\partial f(z)}{\partial z^*} = \frac{1}{2}\left(\frac{\partial u(x,y)}{\partial x} + \jmath\frac{\partial v(x,y)}{\partial x} + \jmath\frac{\partial u(x,y)}{\partial y} - \frac{\partial v(x,y)}{\partial y}\right) \tag{2.39}$$

Observe now that if the Cauchy-Riemann conditions are satisfied, i.e. $\frac{\partial u(x,y)}{\partial x} = \frac{\partial v(x,y)}{\partial y}$ and $\frac{\partial v(x,y)}{\partial x} = -\frac{\partial u(x,y)}{\partial y}$, the following is true

- The $\mathbb{R}^*$-derivative $\frac{\partial f(z)}{\partial z^*} = 0$, i.e. a function $f(z)$ is holomorphic if and only if it is not a function of the complex conjugate variable z^*.

- the $\mathbb{R}$-derivative $\frac{\partial f(z)}{\partial z}$ is equal to the standard $\mathbb{C}$-derivative, though this in general will not be the case for nonanalytic functions.

In using the $\mathbb{R}$- and $\mathbb{R}^*$-derivatives for optimization problems, the following properties are important [98]

$$\frac{\partial f^*}{\partial z^*} = \left(\frac{\partial f^*}{\partial z}\right)^* \tag{2.40}$$

$$\frac{\partial f^*}{\partial z} = \left(\frac{\partial f^*}{\partial z^*}\right)^* \tag{2.41}$$

$$df = \frac{\partial f}{\partial z}dz + \frac{\partial f}{\partial z^*}dz^* \tag{2.42}$$

$$\frac{\partial f(g)}{\partial z} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial z} + \frac{\partial f}{\partial g^*}\frac{\partial g^*}{\partial z} \tag{2.43}$$

$$\frac{\partial f(g)}{\partial z^*} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial z^*} + \frac{\partial f}{\partial g^*}\frac{\partial g^*}{\partial z^*} \tag{2.44}$$

2.3.3 The gradient operator

In this section we present two important results that hold for real functions of complex variables.

Stationary points

Let $f(z, z^*) : \mathbb{C} \times \mathbb{C} \Rightarrow \mathbb{R}$ and $g(x, y) : \mathbb{R} \times \mathbb{R} \Rightarrow \mathbb{R}$ such that $f(z, z^*) = g(x, y)$. For the function $g(x, y)$, a scalar field, it is well known that the stationary points must satisfy [28]

$$\frac{\partial g(x,y)}{\partial x} = 0 \qquad \text{and} \qquad \frac{\partial g(x,y)}{\partial y} = 0$$

Substituting these two equations into the $\mathbb{R}$- and $\mathbb{R}^*$-derivatives in (2.36), we observe that for stationary points we have $\frac{\partial f(z)}{\partial z} = 0$ and $\frac{\partial f(z)}{\partial z^*} = 0$. We also note that $\frac{\partial f(z)}{\partial z} = 0$ and $\frac{\partial f(z)}{\partial z^*} = 0$, if and only if $\frac{\partial g(x,y)}{\partial x} = 0$ and $\frac{\partial g(x,y)}{\partial y} = 0$. It follows from this that a necessary and sufficient condition for f(z) to have a stationary poind is that

$$\frac{\partial f(z)}{\partial z} = 0 \qquad \text{or} \qquad \frac{\partial f(z)}{\partial z^*} = 0$$

The gradient operator

The gradient operator is commonly defined as the derivative that points in the direction of the maximum rate of increase of a scalar field. To obtain the gradient operator for a function $f(\mathbf{z}, \mathbf{z}^*) : \mathbb{C}^N \times \mathbb{C}^N \Rightarrow \mathbb{R}$, we start with the first-order Taylor series expansion of $f(z, z^*)$

$$\delta f(\mathbf{z}, \mathbf{z}^*) = \sum_{k=1}^{N} \left(\frac{\partial f(\mathbf{z}, \mathbf{z}^*)}{\partial z_k} \partial z_k + \frac{\partial f(\mathbf{z}, \mathbf{z}^*)}{\partial z_k^*} \partial z_k^* \right)$$

For a real function of complex variables $\left(\frac{\partial f(\mathbf{z},\mathbf{z}^*)}{\partial \mathbf{z}}\right)^* = \frac{\partial f^*(\mathbf{z},\mathbf{z}^*)}{\partial \mathbf{z}^*} = \frac{\partial f(\mathbf{z},\mathbf{z}^*)}{\partial \mathbf{z}^*}$ and the first order Taylor series can be simplified into

$$\delta f(\mathbf{z}, \mathbf{z}^*) = 2\Re[(\nabla_{\mathbf{z}} f(\mathbf{z}, \mathbf{z}^*))^T \delta \mathbf{z}] = 2\Re[(\nabla_{\mathbf{z}^*} f(\mathbf{z}, \mathbf{z}^*))^H \delta \mathbf{z}]$$

where $\nabla_{\mathbf{z}^*} f(\mathbf{z}, \mathbf{z}^*) = \frac{\partial f(\mathbf{z},\mathbf{z}^*)}{\partial \mathbf{z}^*}$. From Schwarz's inequality [99]

$$\mathbf{a}^H \mathbf{b} \leq \| \mathbf{a} \| \| \mathbf{b} \|$$

where the equality holds only when $b = ka$, for a real or complex scalar constant k. It follows from this that $\delta f(\mathbf{z}, \mathbf{z}^*)$ in (2.3.3) is maximised when $\nabla_{\mathbf{z}^*} f(\mathbf{z}, \mathbf{z}^*))$ is in the direction $\delta \mathbf{z}$, and hence the gradient operator is given by the $\mathbb{R}^*$-derivative $\nabla_{\mathbf{z}^*} f(\mathbf{z}, \mathbf{z}^*) = \frac{\partial f(\mathbf{z},\mathbf{z}^*)}{\partial \mathbf{z}^*}$. Note that the gradient operator is also known as the conjugate gradient.

2.4 An overview of adaptive signal processing

In this section by providing a high level overview of adaptive filtering algorithms and how the filters are related to each other.

2.4.1 Mathematical models

The adaptive filtering algorithms used or developed in this thesis are all linear discrete time filters. Here, linearity refers to the filter being linear in the parameters rather than in the variables. It is important to appreciate that all the adaptive filters presented in

this thesis are based on the Wiener filter, and differ only in their mathematical complexity and optimality. Due to its importance in all adaptive filters that we present, we will first introduce the Wiener filter.

The Wiener filter and least squares solution

Consider the filtering problem involved in generating a signal $y(k)$ which provides an estimate of a desired signal $d(k)$ (also called teaching signal), given input samples $x(k), x(k-1), \ldots, x(k-l)$ of an input signal $\mathbf{x}$. In addition to this, the filtering model is restricted to being linear in the filtering parameters and discrete in time, that is

$$y(k) = w_0 x(k) + w_1 x(k-1) + \cdots + w_l x(k-l) = \mathbf{w}^H \mathbf{x}(k) \tag{2.45}$$

where the coefficient vector $\mathbf{w} \in \mathbb{C}^{1\times l} = [w_0, \ldots, w_l]^T$, the input signal vector $\mathbf{x}(k) \in \mathbb{C}^{1\times l} = [x(k), \ldots, x(k-l)]^T$ and l is the filter order. The aim is to design a filter that minimizes the error $e(k) = d(k) - y(k)$. Figure 2.2 summarises the filtering problem.

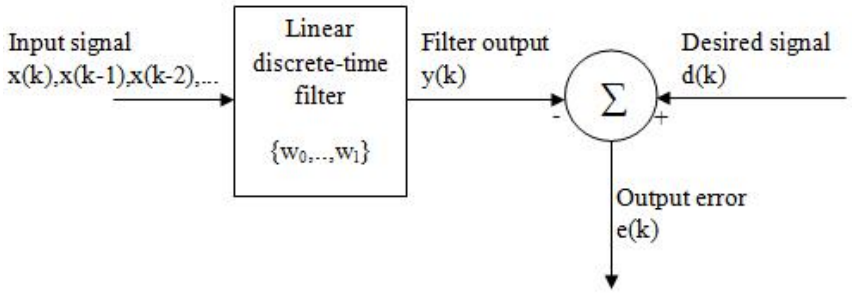


Figure 2.2: Block diagram description of the filtering problem.

It is worth noting that no assumptions have been made about the underlying relationship between $d(k)$ and $x(k)$ or the statistics of either signal. A solution to this filtering problem can be obtained by minimizing the mean square error (MSE)

$$J(k) = E[e^*(k)e(k)]$$

where

$$e(k) = d(k) - y(k) = d(k) - \mathbf{w}^H \mathbf{x}(k) \tag{2.46}$$

Setting the conjugate gradient of the cost function $J(k)$ to 0 we obtain

$$E[e(k)\mathbf{x}(k)] = 0 \tag{2.47}$$

This equation, referred to as the principle of orthogonality, is an important result and states that the mean square error is minimized when the error $e(k)$ is orthogonal to the input signal $\mathbf{x}(k)$. Substituting for $e(k)$ in (2.47) and solving for $\mathbf{w}$ we obtain [20] [100]

$$\mathbf{w} = \mathbf{R}_{xx}^{-1} \mathbf{r}_{dx} \tag{2.48}$$

where we assume $d(k)$ and $\mathbf{x}(k)$ to be stationary, such that the covariance matrix $\mathbf{R}_{xx} = E[\mathbf{x}(k)\mathbf{x}(k)^H]$ and the crosscorrelation $\mathbf{r}_{dx} = E[\mathbf{d}^*(k)\mathbf{x}(k)]$. Equation (2.47) is known as the Wiener filter solution. In practice, $\mathbf{R}_{xx} = E[\mathbf{x}\mathbf{x}^H]$ and $\mathbf{r}_{dx} = E[\mathbf{d}^*\mathbf{x}]$ are not known, however, assuming that the signals are ergodic[2] and weak sense tationary[3], the statistics can be estimated as

$$\mathbf{R}_{xx} \approx \sum_{n=l+1}^{k} \mathbf{x}(n)\mathbf{x}(n)^H \tag{2.49}$$

$$\mathbf{r}_{dx} \approx \sum_{n=l+1}^{k} d^*(n)\mathbf{x}(n)^H \tag{2.50}$$

If the approximations above are made, we have the least squares solution [95] (also called ordinary least squares [101]), which can be independently derived by using the cost function

$$J(k) = \sum_{n=l+1}^{k} e^*(n)e(n)$$

Rewriting the linear model in (2.45) more compactly as

$$\mathbf{y} = \mathbf{H}\mathbf{w}$$

[2] An ergodic process has ensemble statistics that are equal to its time statistics.
[3] A signal is weak sense stationary (WSS) if its first and second moments are independent of time.

where $\mathbf{y} \in \mathbb{C}^{N\times 1}$ is the filter output vector and $\mathbf{H} \in \mathbb{C}^{N\times l}$ is a matrix of the input signal $\mathbf{x}(k)$, we can write the lease squares solution as

$$\mathbf{w} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H\mathbf{x} \tag{2.51}$$

Note that the least squares solution above is the same as the Wiener solution in (2.48) when used with the approximations in (2.50). Having introduced the Wiener solution and least squares solution, we now look at the optimality of the filters. To do this, we must make assumptions about the underlying signal generating model.

It can be shown that, if the underlying signal generating model satisfies[4]

$$E[\mathbf{d}] = \mathbf{H}\mathbf{w} \tag{2.52}$$

and $\hat{\mathbf{w}}$ is constrained to be linear and unbiased, i.e.

$$E[\hat{\mathbf{w}}] = \mathbf{A}E[\mathbf{d}] = \mathbf{w}$$

where $\mathbf{A}$ is a coefficient matrix, then, the best[5] linear unbiased estimator (BLUE) is given by [95]

$$\widehat{\mathbf{w}}_{BLUE} = (\mathbf{H}^H\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{C}^{-1}\mathbf{d} \tag{2.53}$$

where the covariance matrix $\mathbf{C} = E[\mathbf{d}\mathbf{d}^H] \in \mathbb{C}^{N\times N}$ is also the covariance matrix of the noise since $\mathbf{d} = \mathbf{H}\mathbf{w} + \mathbf{n}$. It is worth noting that the BLUE is not dependent on the statistics of the noise, i.e. no assumptions have been made about the probability density function of the noise. By comparing (2.53) to (2.51), we notice that the least squares solution is the best linear unbiased estimator if $\mathbf{C} = \sigma^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. In other words, all noise samples must be uncorrelated and have constant variance.

It can be shown that if instead of assuming that the underlying model takes the

[4] Notice that the assumption $E[\mathbf{d}] = \mathbf{H}\mathbf{w}$ is equivalent to saying that the underlying model is of the form $\mathbf{d} = \mathbf{H}\mathbf{w} + \mathbf{n}$ with the noise variable $\mathbf{n}$ having zero mean.

[5] "Best" in this context refers to the estimator being the linear estimator with the minimum variance.

form of (2.52), we assume it takes the form

$$\mathbf{d} = \mathbf{H}\mathbf{w} + \mathbf{n} \tag{2.54}$$

where the noise vector $\mathbf{n} \sim \mathcal{N}(0, \mathbf{C})$ is Gaussian, then the minimum variance unbiased (MVU) estimator is given by

$$\widehat{\mathbf{w}}_{MVU} = (\mathbf{H}^H\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^H\mathbf{C}^{-1}\mathbf{d} \tag{2.55}$$

This estimator is not only the MVU estimator, but it is also efficient, in the sense that it attains the Cramer-Row lower bound (CRLB). Comparing the MVU in (2.55) to the BLUE in (2.53), we observe that both are equal to each other and it follows from this that if the underlying model is linear as in (2.54) and the noise is Gaussian, then the BLUE is not only the best estimator (in the sense that it has the minimum variance) but it is also efficient.

Adaptive filters

The Wiener solution and least squares solutions are both block based solutions, requiring all data to be available before parameters can be estimated and for the signals to be jointly stationary (in the case of the Wiener solution). In practical applications, however, data is often available online, requiring parameters to be estimated in real time. Furthermore, signal statistics are often nonstationary, requiring parameters to be constantly recalculated and covariance and cross-covariance matrices to be estimated online. These limitations can be partially overcome by splitting the nonstationary data into smaller blocks of data such that the data within each block becomes stationary. However, this approach assumes piecewise stationarity which in most circumstances is unlikely to offer an accurate description of the underlying model. Furthermore, even if piecewise stationarity can be assumed, the statistics of the data might be changing too fast to allow the for data blocks to be long enough to obtain accurate estimates of the parameters. For these reasons, block based solutions are often not desirable, and instead adaptive solutions are developed, where the

assumption of stationarity is removed.

In order to motivate the approach that is commonly used to derive adaptive filtering algorithms, let's remind ourselves what it is that the Wiener filter attempts to achieve. For a desired signal $d(k)$, the Wiener filter attempts to find the filter weight vector $\mathbf{w}$ such that the mean square error of the output error

$$e(k) = d(k) - \mathbf{w}^H \mathbf{x}(k)$$

is minimized. For jointly stationary $d(k)$ and $\mathbf{x}(k)$ the optimum weight vector is given by

$$\mathbf{w} = \mathbf{R}_{xx}^{-1} \mathbf{r}_{dx}$$

If, however, we assume that $d(k)$ and $\mathbf{x}(k)$ are not stationary, the Wiener solution becomes [100]

$$\mathbf{w}(k) = \mathbf{R}_{xx}^{-1}(k) \mathbf{r}_{dx}(k)$$

where the time index indicates that the optimum weight vector $\mathbf{w}(k)$ varies with time. In adaptive signal processing the aim is then to solve for $\mathbf{w}(k)$ such that

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}(k) \tag{2.56}$$

where the correction term $\Delta\mathbf{w}(k)$ is a function of the output error $e(k)$ (see Figure 2.3 for a block diagram of an adaptive filter).

There are a number of advantages to solving for $\mathbf{w}(k)$ recursively

- First, if the filter order l is large, solving for the Wiener-Hopf equations may not be practical.
- Secondly, if the covariance matrix $\mathbf{R}_{xx}$ is ill conditioned, its inverse will be sensitive to finite precision.
- Thirdly, the covariance matrix $\mathbf{R}_{xx}^{-1}(k)$ and cross-covariance matrix $\mathbf{r}_{dx}(k)$ are not known, and must be estimated. A recursive solution would automatically estimate

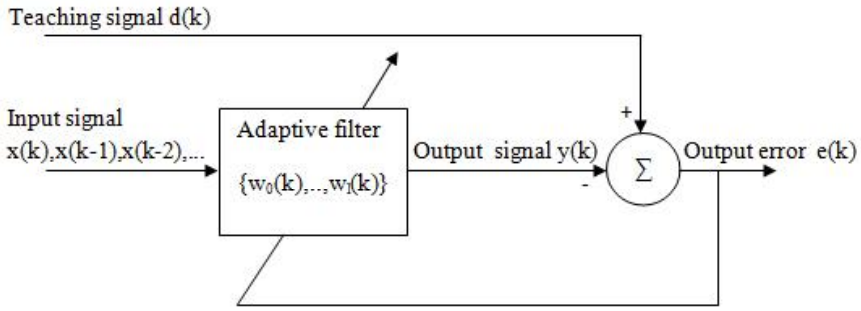


Figure 2.3: Block diagram of an adaptive filter.

these quantities as and when the data is received.

For the adaptive filter to be useful, it must satisfy the following rules

- If the signals are stationary, the optimum weight vector $\mathbf{w}(k)$ should converge to the Wiener solution, i.e.

$$\lim_{k\to\infty} \mathbf{w}(k) = \mathbf{R}_{xx}^{-1}\mathbf{r}_{dx}$$

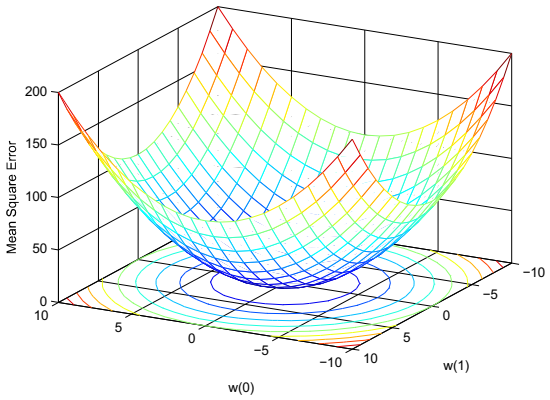
- If the signals are nonstationary, the optimum weight vector $\mathbf{w}(k)$ should be able to track the changing dynamics of the system.

There are two distinctive approaches to deriving adaptive filtering algorithms; the stochastic gradient approach (based on the Wiener Hopf equations) and the least squares estimation approach. Both of these approaches will be used in later chapters to derive quaternion valued adaptive filtering algorithms. We next give an intuitive approach to how the LMS, Affine projection algorithm and RLS work. For ease of representation we will focus on the real domain but the same concepts apply to the complex and quaternion domain.

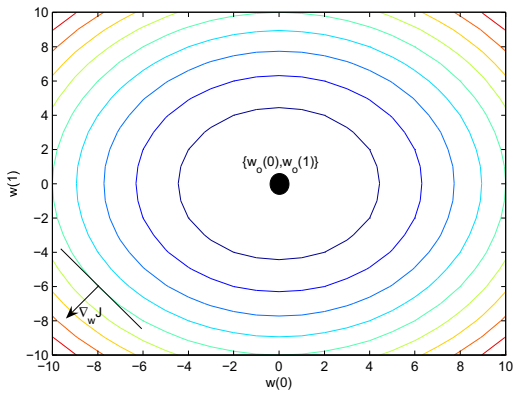
2.4.2 An intuitive description of adaptive filters in $\mathbb{R}$

In Section 2.4.1 we saw that the Wiener filter attempts to find the weight vector $\mathbf{w}(k)$ that minimizes the cost function $J(k) = E[e(k)e^*(k)]$, where $e(k) = d(k) - \mathbf{w}^T\mathbf{x}(k)$. For

a weight vector with only two weights, the problem is illustrated in Figure 2.4 , with the mean square error surface shown in Figure 2.4(a) and the contour diagram shown in Figure 2.4(b). The LMS attempts to find the optimal vector by using the gradient



(a) Error surface.



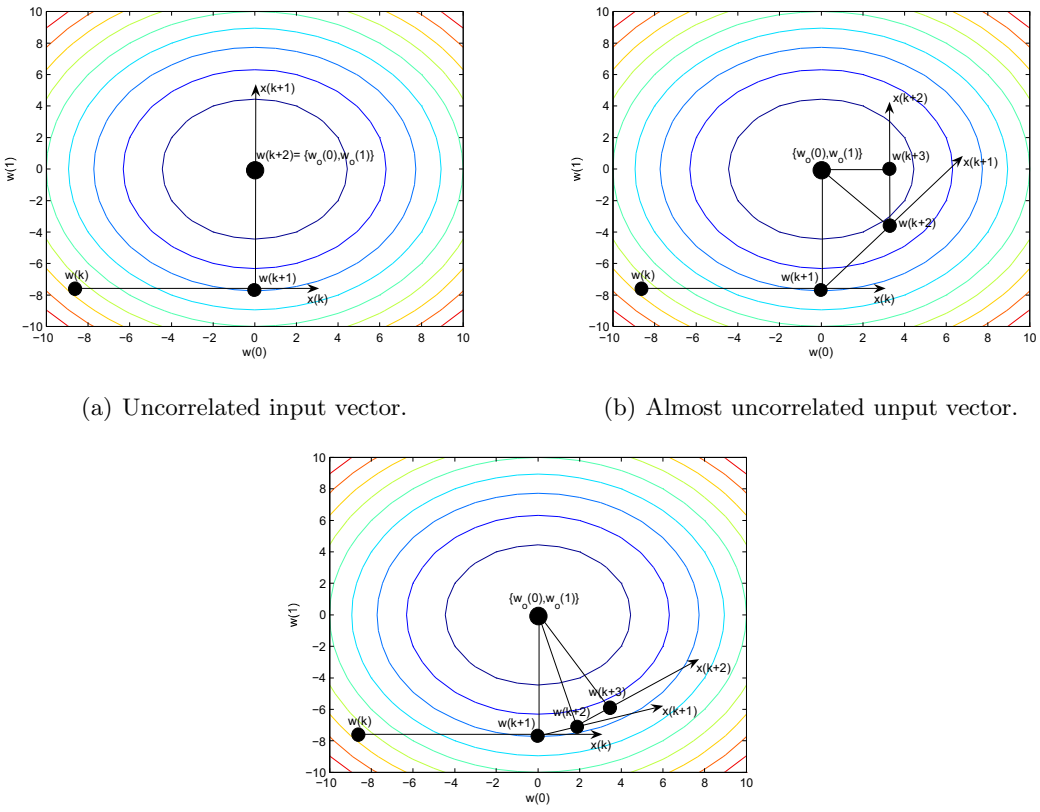
(b) Contours of constant error showing the gradient vector which is perpendicular to the line that is tangent to the contour.

Figure 2.4: Mean square error cost function for a weight vector of size two.

operator $\nabla_w J(k)$ (see Figure 2.4(b)), which points in the direction of maximum ascent. The negative of the gradient operator therefore points in the direction of maximum descent and is the direction that a ball would follow if left free to move on the error surface in Figure 2.4(a). Therefore, for the LMS, the weight update in (2.56) become

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \mu\nabla_w J(k) \\ &= \mathbf{w}(k) + \mu e(k)\mathbf{x}(k) \end{aligned} \tag{2.57}$$

where μ is the step size. Observe from the equation above that the weight $\mathbf{w}(k)$ moves along the error surface in the direction defined by $\mathbf{x}(k)$. In Figure 2.5 we illustrate three different scenarios for the weight update, where in each scenario the step size is set to $\frac{1}{||x||^2}$ (for this particular step size, the filter is called the normalized LMS). From Figure



(a) Uncorrelated input vector.

(b) Almost uncorrelated unput vector.

(c) Highly correlated input vector.

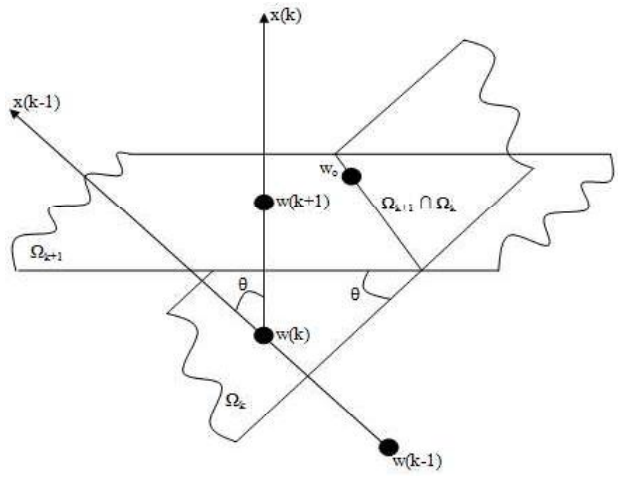
Figure 2.5: Weight vector trajectory along mean square error surface for input signal $\mathbf{x}(k)$ of differing correlation.

2.5(a) we observe that if the input vectors are orthogonal, then the weight vector reaches the optimum value in only two steps and that the weight error vector $\mathbf{w}(k+1) - \mathbf{w}_o$ is always orthogonal to $\mathbf{x}(k)$ (and hence also $\delta w(k+1) = w(k+1) - w(k)$). More generally, if the input vectors are orthogonal, then an n-dimensional weight vector requires n steps to converge to the optimum value. Figures 2.5(b) and 2.5(c) illustrate how the weights converge more slowly the more correlated the input vector, since the weights are limited to moving in the same direction as the input vector.

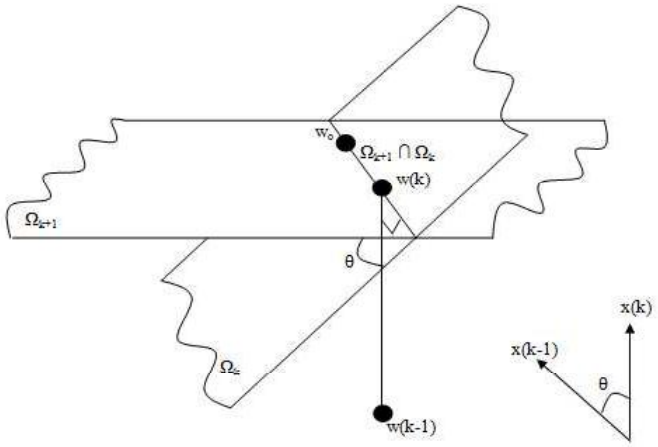
To speed up the convergence rate when input vectors are correlated, the affine pro-

jection algorithm (APA) provides an alternative weight trajectory along the error surface. To illuminate the difference between the NLMS and APA, consider the following interpretation of the NLMS: The NLMS minimizes the change in weight vector from one iteration to the next (i.e. $\delta\mathbf{w}(k+1)$) while ensuring that the constraint $\mathbf{w}^T(k+1)\mathbf{x}(k) = d(k)$ is satisfied. Solving this constrained optimization using the method of Lagrange multipliers yields the NLMS. Figure 2.6(a) offers a geometric perspective to this optimization problem. The plane Ω_k describes the set of all weights $\mathbf{w}(k)$ that act on the input $\mathbf{x}(k-1)$ to produce the output $y(k)$, and similarly for Ω_{k+1}. The angle θ is the angle between the two planes Ω_k and Ω_{k+1}, and is equal to the angle between the input vectors $\mathbf{x}(k)$ and $\mathbf{x}(k-1)$. Of importance, note that the weight change $\delta\mathbf{w}(k+1) = \mathbf{w}(k+1) - \mathbf{w}(k)$ is perpendicular to the plane Ω_{k+1} and parallel to the input vector $\mathbf{x}(k)$.

The APA can be derived by reformulating the constrained optimization problem as follows: The APA minimizes the change in weight vector from one iteration to the next (i.e. $\delta\mathbf{w}(k+1)$) while ensuring that the constraint $\mathbf{w}^T(k+1)\mathbf{x}(k-n) = d(k-n)$ is satisfied for $n = 0, 1, \ldots, N-1$ is satisfied. Figure 2.6(b) offers a geometric perspective to this optimization problem. Compared to the NLMS subspace diagram in Figure 2.6(a), observe from Figure 2.6(b) that the weight change $\delta\mathbf{w}(k) = \mathbf{w}(k) - \mathbf{w}(k-1)$ is no longer perpendicular to the plane Ω_k, but to the intersection of the planes Ω_{k+1} and Ω_k. It follows from this that the convergence rate of the APA is higher than that of the NLMS as the weight vector $\mathbf{w}(k)$ is no longer limited to moving in the same direction as the input vector $\mathbf{x}(k-1)$. The speed of convergence of the APA increases as the number of constraints N, and hence number of intersecting planes N increases. As N increases to include all intersecting planes available, the convergence rate of the APA approaches that of the recursive least squares (RLS).



(a) Normalized LMS



(b) Affine projection algorithm

Figure 2.6: Subspace interpretation of the NLMS and APA, where w_o denotes the optimum weight vector.

Chapter 3

The HR-Calculus and Quaternion Gradients

3.1 Introduction

Recent developments in sensor technology, human centered computing, and robotics [63, 102, 103] have made possible the recording of new classes of multidimensional signals which are naturally represented as two- and three-dimensional vector-valued processes. Such signals are readily modeled as real vectors (in $\mathbb{R}^2$ and $\mathbb{R}^3$), however, there are big advantages in processing such data in division algebras - the complex and quaternion domain. The benefits of complex domain modeling are well understood [45], while for three- and four-dimensional data quaternions (a skew field over $\mathbb{R}^4$), form a noncommutative division algebra and provide a unifying treatment of two-, three- and four-dimensional processes.

Additional benefits arising from the use of such a 3D and 4D division algebra include very accurate rotation and orientation modeling. This has made quaternions a standard in computer graphics [75], where they have replaced real rotation matrices, owing to their ability to provide a compact and accurate representation of sequences of rotations, together with a natural account of the spherical angle [74]. This has brought new opportunities in a variety of applications, such as color imaging [104], body-sensor networks [105], and

aeronautics [69].

Although the ordering relationships (e.g. '$\geq$') are not defined for quaternions (and hence the notion of probability density function), there exists a 'static' isomorphism between $\mathbb{R}^3$ and $\mathbb{R}^4$ and the field of quaternions $\mathbb{H}$ (pure and full), which can be used to introduce the notions of probability distribution and gradient. Such tools are a prerequisite for the design of learning algorithms in the quaternion domain, which offers theoretical and practical advantages when dealing with the large (and rotational) dynamics of vector signals. Examples are quite recent and include the quaternion least mean square (QLMS) [66] and developments in 'augmented' quaternion statistics [64, 65, 106]. These have been followed by *widely linear* adaptive filtering algorithms, including the widely linear QLMS (WLQLMS) [67] and widely linear blind source separation algorithms for improper quaternion sources [107–110].

Standard, or strictly linear, statistical models rest upon the covariance matrix $E[\mathbf{q}\mathbf{q}^H]$, where $\mathbf{q} \in \mathbb{H}^{N\times 1}$ is a regressor vector, and are second order optimal only for circular signals, for which the probability distributions are rotation invariant. In terms of second order statistics, such signals have equal powers in all the four quaternion components and are termed 'proper', however, real world processes are typically 'improper' (with different component powers). To fully describe their second order properties, three complementary covariance matrices must also be employed [64, 65].

A rigorous treatment of the gradient is a key prerequisite to the derivation of stochastic gradient optimization procedures, and the inconsistencies and lack of coherence in their definition has so far prevented a more widespread use of quaternions. Current approaches typically employ quaternion gradient calculation based on the so called 'pseudogradient' [70], however, this gradient, also known as the Cauchy-Riemann-Fueter (CRF) derivative, is very stringent, and poses a number of restrictions in the derivation of the adaptive filtering algorithms. For instance, due to the ambiguity related to the noncommutativity of quaternions, the CRF derivative can be calculated with the imaginary unit vectors $\imath$, $\jmath$ and κ on either the left or right hand side of the gradient components, introducing uncertainty in choosing a correct form. Another problem with the CRF derivative

is its uniqueness, as exemplified on the linear function $J = w$, whereby we obtain $\frac{\partial J}{\partial w^*} = 0$ when differentiating directly with respect to w^* and $\frac{\partial J}{\partial w^*} = -\frac{1}{2}$ using the CRF derivative. These inconsistencies have led to numerous different expressions for structurally similar algorithms, and have slowed down the development and applications of quaternion valued statistical signal processing.

The recent progress in the statistics of quaternion variable (augmented quaternion statistics [64,65]) has facilitated the development of advanced algorithms for vector sensor modeling, however, this has also highlighted several key remaining issues that need to be addressed in order to exploit the full power of quaternions in statistical signal processing applications. To that end, this chapter addresses the following problems:

- The $\mathbb{HR}$-calculus and quaternion gradient are derived, thus providing a framework for the direct differentiation of quaternion functions and enabling the differentiation of nonholomorphic cost functions.

- Unifying different forms of QLMS. Due to the noncommutativity of the quaternion product numerous expressions for the quaternion LMS exist. Our analysis shows that the different forms are equivalent in the way they use the available information.

- Generic extensions of LMS. The existing quaternion LMS algorithms are not a generic extension of the real and complex LMS (CLMS). By introducing a new gradient operator (the I-gradient) we provide a generic expression for the quaternion LMS (the IQLMS).

- Unifying convergence analysis. An in depth analysis for the convergence in the mean of the strictly linear QLMS is still lacking. To that end, we provide a convergence analysis for the two existing variants of the QLMS and for the novel IQLMS. The analysis shows that the IQLMS offers the same performance while requiring only half the mathematical operations.

- A rigorous MSE analysis. No analysis exists for the convergence in the mean square of the widely linear QLMS algorithms. We provide a rigorous and unifying mean

square convergence analysis which quantifies the advantage offered by the widely linear algorithms for noncircular signals.

In this way, we provide a unifying set of tools and an analysis platform for future developments in the field, which has many theoretical benefits but has found only a few applications due to the above uncertainties. The generic form of algorithms enabled by the I-gradient makes it possible to extend the vast resources from the real and complex adaptive filtering into the field of quaternions.

3.2 The HR-calculus

In adaptive filtering, the aim of gradient based optimization is to minimise a measure of error power, typically a scalar function of quaternion variables. This has been a main stumbling block in the development of learning algorithms, as the Cauchy-Riemann-Fueter (CRF) conditions

$$\frac{\partial J}{\partial \mathbf{w}^*} = \frac{1}{4}\left(\frac{\partial J}{\partial \mathbf{w}_r} + \imath\frac{\partial J}{\partial \mathbf{w}_\imath} + \jmath\frac{\partial J}{\partial \mathbf{w}_\jmath} + \kappa\frac{\partial J}{\partial \mathbf{w}_\kappa}\right) = \mathbf{0} \tag{3.1}$$

do not admit the calculation of such gradients since they are not defined for real functions. To that end, following on the corresponding result in the complex domain, where the $\mathbb{CR}$-calculus [45,98] is used to calculate the gradient of real functions of complex variables, we have recently introduced the $\mathbb{HR}$-calculus [111]. We now derive the $\mathbb{HR}$-calculus, which consists of two sets of derivatives, the $\mathbb{HR}$-calculus and $\mathbb{HR}^*$-derivatives.

3.2.1 The $\mathbb{HR}$-derivatives

In the complex domain, the $\mathbb{CR}$-calculus is derived by establishing a relationship between a complex function $f(z) : \mathbb{C} \to \mathbb{C}$ and an equivalent dualvariate real function $f(x, y) : \mathbb{R}^2 \to \mathbb{C}$. Similarly to the $\mathbb{CR}$-calculus [45], in order to establish the duality between the derivatives of a quaternion valued function $f(q) : \mathbb{H} \to \mathbb{H}$ and the derivatives of the corresponding quadrivariate real function $g(q_r, q_\imath, q_\jmath, q_\kappa) : \mathbb{R}^4 \to \mathbb{H}$, we start from (since

the fields $\mathbb{H}$ and $\mathbb{R}^4$ are isomorphic) the quaternion function

$$\begin{aligned} f(q) &= f(q, q^{\imath}, q^{\jmath}, q^{\kappa}) \\ &= f_0(q_r, q_{\imath}, q_{\jmath}, q_{\kappa}) + f_1(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})\imath + f_2(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})\jmath + f_3(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})\kappa \\ &= g(q_r, q_{\imath}, q_{\jmath}, q_{\kappa}) \end{aligned} \tag{3.2}$$

The total differential of the function $g(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})$ can be expressed as

$$dg(q_r, q_{\imath}, q_{\jmath}, q_{\kappa}) = \frac{\partial g(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_r} dq_r + \frac{\partial g(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\imath}} dq_{\imath} + \frac{\partial g(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\jmath}} dq_{\jmath} + \frac{\partial g(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\kappa}} dq_{\kappa} \tag{3.3}$$

Due to the isomorphism that exists between the spaces $\mathbb{H}$ and $\mathbb{R}^4$, we ca substitute (3.2) into (3.3) to obtain

$$\begin{aligned} &dg(q_r, q_{\imath}, q_{\jmath}, q_{\kappa}) = \\ &\frac{\partial f_0(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_r} dq_r + i\frac{\partial f_1(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_r} dq_r + j\frac{\partial f_2(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_r} dq_r + k\frac{\partial f_3(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_r} dq_r \\ &+\frac{\partial f_0(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\imath}} dq_{\imath} + \imath\frac{\partial f_1(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\imath}} dq_{\imath} + \jmath\frac{\partial f_2(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\imath}} dq_{\imath} + \kappa\frac{\partial f_3(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\imath}} dq_{\imath} \\ &+\frac{\partial f_0(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\jmath}} dq_{\jmath} + \imath\frac{\partial f_1(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\jmath}} dq_{\jmath} + \jmath\frac{\partial f_2(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\jmath}} dq_{\jmath} + \kappa\frac{\partial f_3(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\jmath}} dq_{\jmath} \\ &+\frac{\partial f_0(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\kappa}} dq_{\kappa} + \imath\frac{\partial f_1(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\kappa}} dq_{\kappa} + \jmath\frac{\partial f_2(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\kappa}} dq_{\kappa} + \kappa\frac{\partial f_3(q_r, q_{\imath}, q_{\jmath}, q_{\kappa})}{\partial q_{\kappa}} dq_{\kappa} \end{aligned} \tag{3.4}$$

Using the identities in (2.10) we can write dq_r, $dq_{\imath}$, $dq_{\jmath}$ and dq_{κ} in terms of involutions[1]

$$\begin{aligned} dq_r &= \frac{1}{4}[dq + dq^{\imath} + dq^{\jmath} + dq^{\kappa}] \qquad & dq_{\imath} &= \frac{-\imath}{4}[dq + dq^{\imath} - dq^{\jmath} - dq^{\kappa}] \\ dq_{\jmath} &= \frac{-\jmath}{4}[dq - dq^{\imath} + dq^{\jmath} - dq^{\kappa}] \qquad & dq_{\kappa} &= \frac{-\kappa}{4}[dq - dq^{\imath} - dq^{\jmath} + dq^{\kappa}] \end{aligned} \tag{3.5}$$

Equation (3.4) can therefore be written in the following form

$$dg(q_r, q_{\imath}, q_{\jmath}, q_{\kappa}) = Adq + Bdq^i + Cdq^j + Ddq^k \tag{3.6}$$

[1]Note that we can also use the relations $dq_r = \frac{1}{2}(q + q^*)$, $dq_{\imath} = \frac{1}{2}(q - q^{\jmath *})$, $dq_{\imath} = \frac{1}{2}(q - q^{\jmath *})$, $dq_{\kappa} = \frac{1}{2}(q - q^{\kappa *})$, however, we lose two degrees of freedom, rendering this approach inappropriate for analysing the duality with the gradients of general quadrivariate functions.

By substituting the identities in (3.5) into the total differential $dg(q_r, q_\imath, q_\jmath, q_\kappa)$ in (3.4),we can solve for A, B, C and D

$$\begin{aligned}
A &= \frac{1}{4}\left(\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_r} - \imath\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\imath} - \jmath\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\jmath} - \kappa\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\kappa}\right)\\
B &= \frac{1}{4}\left(\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_r} - \imath\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\imath} + \jmath\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\jmath} + \kappa\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\kappa}\right)\\
C &= \frac{1}{4}\left(\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_r} + \imath\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\imath} - \jmath\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\jmath} + \kappa\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\kappa}\right)\\
D &= \frac{1}{4}\left(\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_r} + \imath\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\imath} + \jmath\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\jmath} - \kappa\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\kappa}\right)
\end{aligned} \tag{3.7}$$

The total differential of the function $f(q, q^\imath, q^\jmath, q^\kappa)$ can also be written as

$$\begin{aligned}
df(q) &= df(q,q^\imath,q^\jmath,q^\kappa) = \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q}dq + \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q^\imath}dq^\imath + \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q^\jmath}dq^\jmath\\
&\quad + \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q^\kappa}dq^\kappa
\end{aligned} \tag{3.8}$$

Comparing $df(q)$ in (3.8) to $dg(q_r, q_\imath, q_\jmath, q_\kappa)$ in (3.6), we can obtain the following partial derivatives, which are denoted as $\mathbb{HR}$-derivatives

$$\begin{bmatrix} \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q} \\ \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q^\imath} \\ \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q^\jmath} \\ \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q^\kappa} \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & -\imath & -\jmath & -\kappa \\ 1 & -\imath & \jmath & \kappa \\ 1 & \imath & -\jmath & \kappa \\ 1 & \imath & \jmath & -\kappa \end{bmatrix}\begin{bmatrix} \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_r} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_i} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\jmath} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\kappa} \end{bmatrix} \tag{3.9}$$

3.2.2 The $\mathbb{HR}^*$-derivatives

In deriving the $\mathbb{HR}$-derivatives we used the substitution

$$\begin{aligned}
dq_r &= \frac{1}{4}[dq + dq^\imath + dq^\jmath + dq^\kappa] \qquad dq_\imath = \frac{1}{4\imath}[dq + dq^\imath - dq^\jmath - dq^\kappa]\\
dq_\jmath &= \frac{1}{4\jmath}[dq - dq^\imath + dq^\jmath - dq^\kappa] \qquad dq_\kappa = \frac{1}{4\kappa}[dq - dq^\imath - dq^\jmath + dq^\kappa]
\end{aligned} \tag{3.10}$$

to obtain a set of derivatives that are a function of the q, q^i ,q^j and q^k. Alternatively, using the conjugate involutions, we could have used the following substitution

$$\begin{aligned} dq_r &= \frac{1}{4}[dq^* + dq^{\imath *} + dq^{\jmath *} + dq^{\kappa *}] \qquad dq_\imath = \frac{1}{4\imath}[-dq^* - dq^{\imath *} + dq^{\jmath *} + dq^{\kappa *}] \\ dq_\jmath &= \frac{1}{4\jmath}[-dq^* + dq^{\imath *} - dq^{\jmath *} + dq^{\kappa *}] \qquad dq_\kappa = \frac{}{4\kappa}[-dq^* + dq^{\imath *} + dq^{\jmath *} - dq^{\kappa *}] \end{aligned} \tag{3.11}$$

By substituting (3.11) above into the total differential $dg(q_r, q_\imath, q_\jmath, q_\kappa)$ in (3.4), we can write $dg(q_r, q_\imath, q_\jmath, q_\kappa)$ as

$$dg(q_r, q_\imath, q_\jmath, q_\kappa) = \overline{A}dq^* + \overline{B}dq^{\imath *} + \overline{C}dq^{\jmath *} + \overline{D}dq^{\kappa *} \tag{3.12}$$

where,

$$\begin{aligned} \overline{A} &= \frac{1}{4}\left(\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_r} + \imath\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\imath} + \jmath\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\jmath} + \kappa\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\kappa}\right) \\ \overline{B} &= \frac{1}{4}\left(\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_r} + \imath\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\imath} - \jmath\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\jmath} - \kappa\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\kappa}\right) \\ \overline{C} &= \frac{1}{4}\left(\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_r} - \imath\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\imath} + \jmath\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\jmath} - \kappa\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\kappa}\right) \\ \overline{D} &= \frac{1}{4}\left(\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_r} - \imath\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\imath} - \jmath\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\jmath} + \kappa\frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\kappa}\right) \end{aligned} \tag{3.13}$$

The differential of the function $f(q^*, q^{i*}, q^{j*}, q^{k*})$ can be written as

$$\begin{aligned} df(q) = df(q^*, q^{\imath *}, q^{\jmath *}, q^{\kappa *}) = \frac{\partial f(q^*, q^{\imath *}, q^{\jmath *}, q^{\kappa *})}{\partial q^*}dq^* + \frac{\partial f(q^*, q^{\imath *}, q^{\jmath *}, q^{k^*})}{\partial q^{\imath *}}dq^{\imath *} \\ + \frac{\partial f(q^*, q^{\imath *}, q^{\jmath *}, q^{\kappa *})}{\partial q^{\jmath *}}dq^{\jmath *} + \frac{\partial f(q^*, q^{\imath *}, q^{\jmath *}, q^{k^*})}{\partial q^{\kappa *}}dq^{\kappa *} \end{aligned} \tag{3.14}$$

and comparing (3.14) to (3.12) we obtain the $\mathbb{HR}^*$-derivatives

$$\begin{bmatrix} \frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^*} \\ \frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^{\imath *}} \\ \frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^{\jmath *}} \\ \frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^{\kappa *}} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & \imath & \jmath & \kappa \\ 1 & \imath & -\jmath & -\kappa \\ 1 & -\imath & \jmath & -\kappa \\ 1 & -\imath & -\jmath & \kappa \end{bmatrix} \begin{bmatrix} \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_r} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\imath} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\jmath} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\kappa} \end{bmatrix} \tag{3.15}$$

3.2.3 Comparison with the $\mathbb{CR}$-calculus

From the $\mathbb{HR}$-derivatives in (3.9) and $\mathbb{HR}^*$-derivatives in (3.15) we observe that the derivative $\frac{\partial f(q,q^\imath,q^\jmath,q^k)}{\partial q}$ and $\frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^*}$ are generic extensions of the derivative $\frac{\partial f(z,z^*)}{\partial z}$ and $\frac{\partial f(z,z^*)}{\partial z^*}$ in the complex domain, with two additional terms that account for the two additional imaginary unit vectors $\jmath$ and κ, scaled by a quarter instead of a half. Due to its quadrivariate nature, $\mathbb{HR}$-calculus also results in six other derivatives not present in $\mathbb{CR}$-calculus. There are, however, three subtle but important differences with $\mathbb{CR}$-calculus, as discussed below.

The argument of the function

In the $\mathbb{CR}$-calculus, the function $f(\cdot)$ in the $\mathbb{R}$-derivative $\frac{\partial f(z,z^*)}{\partial z}$ and $\mathbb{R}^*$-derivative $\frac{\partial f(z,z^*)}{\partial z^*}$ takes the same argument, that is, z and z^*. This is not the case in $\mathbb{HR}$-calculus, where the function $f(\cdot)$ in the $\mathbb{HR}$-derivatives has the argument $\{q, q^\imath, q^\jmath, q^\kappa\}$, whereas the function $f(\cdot)$ in the $\mathbb{HR}^*$-derivatives has the argument $\{q^*, q^{\imath *}, q^{\jmath *}, q^{\kappa *}\}$. An important consequence of this structure is that to differentiate quaternion functions, the argument of the function must be in the correct form. However, quaternion functions are often written in terms of both q and q^* and are thus not written in the correct form. To overcome this, the substitutions

$$q^* = \frac{1}{2}(q^\imath + q^\jmath + q^\kappa - q) \qquad \text{and} \qquad q = \frac{1}{2}(q^{\imath *} + q^{\jmath *} + q^{\kappa *} - q^*) \tag{3.16}$$

can be used to get the argument of the function into the form $\{q, q^\imath, q^\jmath, q^\kappa\}$ or $\{q^*, q^{\imath *}, q^{\jmath *}, q^{\kappa *}\}$.

Independence of the variables q and q^*

In the $\mathbb{CR}$-calculus, the variables z and z^* are treated as independent since $\frac{\partial z}{\partial z^*} = \frac{\partial z^*}{\partial z} = 0$. This is not, however, the case in the $\mathbb{HR}$-calculus, where we have $\frac{\partial q}{\partial q^*} = \frac{\partial q^*}{\partial q} = -\frac{1}{2}$, and therefore q and q^* cannot be treated as independent. Intuitively, this is not surprising since we can see from (3.16) that q can be written in terms of q^*. Instead, in the quaternion domain, variables within the groups $\{q, q^\imath, q^\jmath, q^\kappa\}$ and $\{q^*, q^{\imath *}, q^{\jmath *}, q^{\kappa *}\}$ are independent to each other.

Placement of the imaginary unit vectors $\imath$, $\jmath$ and κ in the $\mathbb{HR}$- and $\mathbb{HR}^*$-derivatives

In equations (3.9) and (3.15), the imaginary unit vectors $\imath$, $\jmath$ and κ are placed on the left of the partial derivatives $\frac{\partial f(\cdot)}{\partial q_r}$, $\frac{\partial f(\cdot)}{\partial q_\imath}$, $\frac{\partial f(\cdot)}{\partial q_\jmath}$ and $\frac{\partial f(\cdot)}{\partial q_\kappa}$. However, as shown below, equations (3.9) and (3.15) can also be written as row vectors, with the imaginary unit vectors on the right of the partial derivatives.

$$\begin{bmatrix} \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q} \\ \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q^\imath} \\ \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q^\jmath} \\ \frac{\partial f(q,q^\imath,q^\jmath,q^\kappa)}{\partial q^\kappa} \end{bmatrix}^T = \frac{1}{4} \begin{bmatrix} \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_r} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\imath} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\jmath} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\kappa} \end{bmatrix}^T \begin{bmatrix} 1 & 1 & 1 & 1 \\ -\imath & -\imath & \imath & \imath \\ -\jmath & \jmath & -\jmath & \jmath \\ -\kappa & \kappa & \kappa & -\kappa \end{bmatrix} \tag{3.17}$$

$$\begin{bmatrix} \frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^*} \\ \frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^{\imath *}} \\ \frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^{\jmath *}} \\ \frac{\partial f(q^*,q^{\imath *},q^{\jmath *},q^{\kappa *})}{\partial q^{\kappa *}} \end{bmatrix}^T = \frac{1}{4} \begin{bmatrix} \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_r} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\imath} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\jmath} \\ \frac{\partial f(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_\kappa} \end{bmatrix}^T \begin{bmatrix} 1 & 1 & 1 & 1 \\ \imath & \imath & -\imath & -\imath \\ \jmath & -\jmath & \jmath & -\jmath \\ \kappa & -\kappa & -\kappa & -\kappa \end{bmatrix} \tag{3.18}$$

To differentiate between both arrangements, we will refer to (3.9) and (3.15) as the

left $\mathbb{HR}$- and $\mathbb{HR}^*$- derivatives (making up the left $\mathbb{HR}$-calculus), and (3.17) and (3.18) as the right $\mathbb{HR}$- and right $\mathbb{HR}^*$- derivatives (making up the right $\mathbb{HR}$-calculus) . It is important to note that due to the non commutativity of quaternion products, the left $\mathbb{HR}$-calculus and right $\mathbb{HR}$-calculus are not equal to each other. Only by examining the derivation of the $\mathbb{HR}$-calculus can we ascertain the validity of both forms.

In deriving the $\mathbb{HR}$- and $\mathbb{HR}^*$-derivatives, we started with the total differential (see equation (3.3))

$$\begin{aligned} dg(q_r, q_\imath, q_\jmath, q_\kappa) &= \frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_r} dq_r + \frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\imath} dq_\imath + \frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\jmath} dq_\jmath \\ &\quad + \frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\kappa} dq_\kappa \end{aligned} \tag{3.19}$$

Note that the position of the differentials dq_r, $dq_\imath$, $dq_\jmath$ and dq_κ is not important as all terms are real and therefore we could also have started with

$$\begin{aligned} dg(q_r, q_\imath, q_\jmath, q_\kappa) &= dq_r \frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_r} + dq_\imath \frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\imath} + dq_\jmath \frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\jmath} \\ &\quad + dq_\kappa \frac{\partial g(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q_\kappa} \end{aligned} \tag{3.20}$$

However, the two differentials above are no longer equal to each other when we use (3.5) to substitute for dq_r, $dq_\imath$, $dq_\jmath$ and dq_κ above. The introduction of the unit vectors $\imath$, $\jmath$ and κ in conjunction with the non commutativity of quaternion products means that $\frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_b} \frac{idq}{4} \neq \frac{idq}{4} \frac{\partial g(q_r,q_\imath,q_\jmath,q_\kappa)}{\partial q_b}$. Since both (3.19) and (3.20) are correct and identical, it follows that both the left $\mathbb{HR}$-calculus and right $\mathbb{HR}$-calculus must be valid solutions. However, since the derivative of a function at a point can only take one value, it follows that there is an uncertainty as to which derivative should be used. In the next section, simple examples show that for some function the left $\mathbb{HR}$-calculus provides the correct derivative while for others it is the right $\mathbb{HR}$-calculus.

As a result of this uncertainty, it is therefore preferable to differentiate a function directly with respect to q, $q^\imath$, $q^\jmath$, or q^κ and use the substitution $q^* = \frac{1}{2}(q^\imath + q^\jmath + q^\kappa - q)$ and $q = \frac{1}{2}(q^{\imath *} + q^{\jmath *} + q^{\kappa *} - q^*)$ where necessary, rather than differentiate with respect to q_r, $q_\imath$, $q_\jmath$ and q_κ. In doing so, we thus circumvent the uncertainty in the placement of the

unit vectors.

3.2.4 Examples

Example 1. $f(q) = q$

Using the left $\mathbb{HR}$-derivatives in (3.9) we obtain the four derivatives for $f(q) = q = q_r + q_\imath \imath + q_\jmath \jmath + q_\kappa \kappa$, as

$$\begin{aligned}
\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q} &= \frac{1}{4}[1 - \imath\imath - \jmath\jmath - \kappa\kappa] = 1 \\
\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q^\imath} &= \frac{1}{4}[1 - \imath\imath + \jmath\jmath + \kappa\kappa] = 0 \\
\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q^\jmath} &= \frac{1}{4}[1 + \imath\imath - \jmath\jmath + \kappa\kappa] = 0 \\
\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q^\kappa} &= \frac{1}{4}[1 + \imath\imath + \jmath\jmath - \kappa\kappa] = 0
\end{aligned}$$

The four results above are what we expect to obtain if $f(q) = q$ is directly differentiated with respect to the variables q, $q^\imath$, $q^\jmath$ and q^κ.

Example 2. $f(q^*) = q^*$

Using the left $\mathbb{HR}$-derivatives in (3.9) we obtain the four derivatives for $f(q^*) = q^* = q_r - q_b \imath - q_c \jmath - q_d \kappa$, as

$$\begin{aligned}
\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q} = \frac{1}{4}[1 + \imath\imath + \jmath\jmath + \kappa\kappa] &= -\frac{1}{2} \\
\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q^\imath} = \frac{1}{4}[1 + \imath\imath - \jmath\jmath - \kappa\kappa] &= \frac{1}{2} \\
\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q^\jmath} = \frac{1}{4}[1 - \imath\imath + \jmath\jmath - \kappa\kappa] &= \frac{1}{2} \\
\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q^\kappa} = \frac{1}{4}[1 - \imath\imath - \jmath\jmath + \kappa\kappa] &= \frac{1}{2}
\end{aligned}$$

At first this result appears erroneous since the function $f(q)$ is not a function of any of the variables it is being differentiated with. However, remembering that the $\mathbb{HR}$-derivatives

are defined for functions of q, $q^\imath$, $q^\jmath$ and q^κ, and not q^*, we must first substitute for q^* in $f(q^*)$ to give $f(q^*) = \frac{1}{2}(q^\imath + q^\jmath + q^\kappa - q)$. Performing direct differentiation of $f(q^*)$ in this new form will give the partial derivatives calculated above. It is important to realise that a consequence of the result above is that, unlike the complex domain, q and q^* cannot be treated as independent variables.

Example 3. $f(q) = q^\imath$

Using the left $\mathbb{HR}$-derivatives in (3.9) we obtain the four derivatives for $f(q) = q^\imath = q_r + q_\imath\imath - q_\jmath\jmath - q_\kappa\kappa$, as

$$\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q} = \frac{1}{4}[1 - \imath + \jmath\jmath + \kappa\kappa] = 0$$

$$\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q^\imath} = \frac{1}{4}[1 - \imath - \jmath\jmath - \kappa\kappa] = 1$$

$$\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q^\jmath} = \frac{1}{4}[1 + \imath + \jmath\jmath - \kappa\kappa] = 0$$

$$\frac{\partial f(q_r, q_\imath, q_\jmath, q_\kappa)}{\partial q^\kappa} = \frac{1}{4}[1 + \imath - \jmath\jmath + \kappa\kappa] = 0$$

The four results obtained above is exactly what we expect if $f(q) = q^\imath$ is directly differentiated with respect to the variables q, $q^\imath$, $q^\jmath$, q^κ. A similar result is obtained for the function $f(q) = q^\jmath$ and $f(q) = q^\kappa$.

Example 4. $f(w) = wx$ **where** w **and** x **are both quaternion variables**

Using the left $\mathbb{HR}$-derivatives in (3.9) to calculate the derivative $\frac{\partial f(w)}{\partial w}$ for the function

$$f(w) = wx = [w_r x_r - w_\imath x_\imath - w_\jmath x_\jmath - w_\kappa x_\kappa] + [w_r x_\imath + w_\imath x_r + w_\jmath x_\kappa - w_\kappa x_\jmath]\imath$$
$$+[w_r x_\jmath + w_\jmath x_r + w_\kappa x_\imath - w_\imath x_\kappa]\jmath + [w_r x_\kappa + w_\kappa x_r + w_\imath x_\jmath - w_\jmath x_\imath]\kappa$$

we obtain

$$\begin{aligned}\frac{\partial f(w)}{\partial w} &= \frac{1}{4}\big([x_r + x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - \imath[-x_\imath + x_r\imath + x_\jmath\kappa - x_\kappa\jmath] - \jmath[x_r\jmath - x_\jmath + x_\kappa\imath - x_\imath\kappa] \\ &\quad -\kappa[x_r\kappa + x_\imath\jmath - x_\kappa - x_\jmath\imath]\big) \\ &= \frac{1}{4}\big([x_r + x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - [-x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] - [-x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] \\ &\quad -[-x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa]\big) \\ &= \frac{1}{4}(x + x + x + x) = x\end{aligned}$$

However, if we use the right $\mathbb{HR}$-derivatives in (3.17) to calculate the derivative $\frac{\partial f(w)}{\partial w}$, we have

$$\begin{aligned}\frac{\partial f(w)}{\partial w} &= \frac{1}{4}\big([x_r + x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - [-x_\imath + x_r\imath + x_\jmath\kappa - x_\kappa\jmath]\imath - [x_r\jmath - x_\jmath + x_\kappa\imath - x_\imath\kappa]\jmath \\ &\quad -[x_r\kappa + x_\imath\jmath - x_\kappa - x_\jmath\imath]\kappa\big) \\ &= \frac{1}{4}\big([x_r + x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - [-x_r - x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - [-x_r + x_\imath\imath - x_\jmath\jmath + x_\kappa\kappa] \\ &\quad -[-x_r + x_\imath\imath + x_\jmath\jmath - x_\kappa\kappa]\big) \\ &= x_r\end{aligned}$$

Observe from this example that the left $\mathbb{HR}$-derivatives and right $\mathbb{HR}$-derivatives don't always give the same answer. In this case, it is the left $\mathbb{HR}$-derivative that provides an answer that is consistent with complex domain results.

Example 5. $f(w) = xw$ **where** w **and** x **are both quaternion variables**

Using the left $\mathbb{HR}$-derivatives in (3.9) to calculate the derivative $\frac{\partial f(w)}{\partial w}$ for the function

$$\begin{aligned}f(w) = xw = [x_r w_r - x_\imath w_\imath - x_\jmath w_\jmath - x_\kappa w_\kappa] + [x_r w_\imath + x_\imath w_r + x_\jmath w_\kappa - x_\kappa w_\jmath]\imath \\ +[x_r w_\jmath + x_\jmath w_r + x_\kappa w_\imath - x_\imath w_\kappa]\jmath + [x_r w_\kappa + x_\kappa w_r + x_\imath w_\jmath - x_\jmath w_\imath]\kappa\end{aligned}$$

we obtain

$$\begin{aligned}\frac{\partial f(w)}{\partial w} &= \frac{1}{4}\big([x_r + x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - \imath[-x_\imath + x_r\imath + x_\kappa\jmath - x_\jmath\kappa] - \jmath[x_r\jmath - x_\jmath - x_\kappa\imath + x_\imath\kappa] \\ &\quad -\kappa[x_r\kappa - x_\imath\jmath - x_\kappa + x_\jmath\imath]\big) \\ &= \frac{1}{4}\big([x_r + x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - [-x_r - x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - [-x_r + x_\imath\imath - x_\jmath\jmath + x_\kappa\kappa] \\ &\quad -[-x_r + x_\imath\imath + x_\jmath\jmath - x_\kappa\kappa]\big) \\ &= x_r\end{aligned}$$

However, if we use the right $\mathbb{HR}$-derivatives in (3.17) to calculate the derivative $\frac{\partial f(w)}{\partial w}$, we have

$$\begin{aligned}\frac{\partial f(w)}{\partial w} &= \frac{1}{4}\big([x_r + x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - [-x_\imath + x_r\imath + x_\kappa\jmath - x_\jmath\kappa]\imath - [x_r\jmath - x_\jmath - x_\kappa\imath + x_\imath\kappa]\jmath \\ &\quad -[x_r\kappa - x_\imath\jmath - x_\kappa + x_\jmath\imath]\kappa\big) \\ &= \frac{1}{4}\big([x_r + x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - [-x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] - [-x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] \\ &\quad -[-x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa]\big) \\ &= \frac{1}{4}(x + x + x + x) = x\end{aligned}$$

Observe the right $\mathbb{HR}$-derivatives of the function $f(w) = xw$ is equal to the left $\mathbb{HR}$-derivatives of the function $f(w) = wx$ (see Example 4). This is an interesting, if not surprising result since the function $f(w) = xw$ and $f(w) = wx$ are not equal to each other due to the noncommutativity of the quaternion product. After Example 6, we will elaborate on this observation.

Example 6. $f(w) = x^*w^*$ **where** w **and** x **are both quaternion variables**

Using the right $\mathbb{HR}$-derivatives in (3.9) to calculate the derivative $\frac{\partial f(w)}{\partial w}$ for the function

$$\begin{aligned}f(w) = x^*w^* = [w_r x_r - w_\imath x_\imath - w_\jmath x_\jmath - w_\kappa x_\kappa] + [-w_r x_\imath - w_\imath x_r - w_\jmath x_\kappa + w_\kappa x_\jmath]\imath \\ +[-w_r x_\jmath - w_\jmath x_r - w_\kappa x_\imath + w_\imath x_\kappa]\jmath + [-w_r x_\kappa - w_\kappa x_r - w_\imath x_\jmath + w_\jmath x_\imath]\kappa\end{aligned}$$

we obtain

$$\begin{aligned}\frac{\partial f(w)}{\partial w} &= \frac{1}{4}\big([x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] - [-x_\imath - x_r\imath - x_\jmath\kappa + x_\kappa\jmath]\imath - [-x_r\jmath - x_\jmath - x_\kappa\imath + x_\imath k]\jmath \\ &\quad -[-x_r k - x_\imath\jmath - x_\kappa + x_\jmath\imath]\kappa\big) \\ &= \frac{1}{4}\big([x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] - [x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] - [x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] \\ &\quad - [x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa]\big) \\ &= \frac{1}{4}(x^* - x^* - x^* - x^*) = -\frac{x^*}{2}\end{aligned}$$

The result above can be obtained by directly differentiation $f(w^*) = x^*w^*$ with respect to w if we first substitute w^* with $\frac{1}{2}(w^i + w^j + w^k - w)$. The function then becomes $f(w) = \frac{1}{2}x^*(w^i + w^j + w^k - w)$, giving the result above when differentiated directly with respect to w.

If instead we use the left $\mathbb{HR}$-derivatives in (3.17) to calculate the derivative $\frac{\partial f(w^*)}{\partial w}$, we obtain

$$\begin{aligned}\frac{\partial f(w)}{\partial w} &= \frac{1}{4}\big([x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] - \imath[-x_\imath - x_r\imath - x_\jmath\kappa + x_\kappa\jmath] - \jmath[-x_r\jmath - x_\jmath - x_\kappa\imath + x_\imath k] \\ &\quad -\kappa[-x_r k - x_\imath\jmath - x_\kappa + x_\jmath\imath]\big) \\ &= \frac{1}{4}\big([x_r - x_\imath\imath - x_\jmath\jmath - x_\kappa\kappa] - [x_r - x_\imath\imath + x_\jmath\jmath + x_\kappa\kappa] - [x_r + x_\imath\imath - x_\jmath\jmath + x_\kappa\kappa] \\ &\quad - [x_r + x_\imath\imath + x_\jmath\jmath - x_\kappa\kappa]\big) \\ &= \frac{1}{4}(x^* - x^{\imath*} - x^{\jmath*} - x^{\kappa*}) = -\frac{x}{2}\end{aligned}$$

We notice that this is not the same answer as that obtained with the right $\mathbb{HR}$-derivatives or by performing direct differentiation after substituting for w^*.

Evaluating derivatives numerically

We have seen from Examples 4, 5 and 6 that the left $\mathbb{HR}$-calculus and right $\mathbb{HR}$-calculus do not always give the same answer. We also noticed that one of the two derivatives can be obtained straightforwardly by directly differentiating the function with respect to the quaternion variable (after writing the argument of the function in the correct form), while

the other derivative cannot be explained in an intuitive way. We therefore suggested that the latter derivative is likely to be incorrect. We now provide numerical evidence for this observation, i.e. that only the derivative which is consistent with direct differentiation is correct.

Visualizing the gradient of a quaternion function (by estimating the gradient graphically) is not possible as it would require an eight dimensional plot. We can, however, confirm the gradient of a linear function numerically by taking two points on a line and comparing the slope of the line with the gradient obtained by $\mathbb{HR}$-calculus. We take the function $y_{left} = wx$ from Example 4. Using the left $\mathbb{HR}$-calculus we showed that $\frac{\partial y_{left}}{\partial w} = x$, while using the right $\mathbb{HR}$-calculus we showed that $\frac{\partial y_{left}}{\partial w} = x_r$. For the function in Example 5, i.e. $y_{right} = xw$, we saw that using the left $\mathbb{HR}$-calculus we have $\frac{\partial y_{right}}{\partial w} = x_r$, while using the right $\mathbb{HR}$-calculus we have $\frac{\partial y_{right}}{\partial w} = x$. This is a striking result since y_{left} and y_{right} are two different functions due to the non-commutativity of quaternion numbers, but both have the same gradient according to $\mathbb{HR}$-calculus. We now proceed to verifying this result numerically.

Evaluating the gradient of $y = wx$ numerically

The function $y = wx$ is the equation of a straight line and so the gradient can be obtained by taking two points on a line. Taking the constant x to be $x = 1 - 2\imath + 3\jmath - \kappa$, and two points w_1 and w_2 to be $w_1 = 1 - 2\imath + \jmath - 4\kappa$ and $w_2 = 2 + 2\jmath - 3\kappa$, we can obtain the following

$$y_1 = w_1 x = -10 + 7\imath + 10\jmath - 9\kappa$$

$$y_2 = w_2 x = -7 + 3\imath + 14\jmath - \kappa$$

$$dy = y_2 - y_1 = 3 - 4\imath + 4\jmath + 8\kappa$$

$$dw = w_2 - w_1 = 1 + 2\imath + \jmath + \kappa$$

The gradient can be obtained by dividing dy by dw. However, it is preferable to keep dy and dw in this form since the division of a quaternion can be ambiguous due to the

noncommutativity of the quaternion product.

According to left $\mathbb{HR}$-derivative $\frac{\partial f(x)}{\partial w} = x$. This can be written as $\partial y = x\partial w$. Because of the uncertainty of the reciprocal $\frac{1}{\partial w}$, this could instead be written as $\partial y = (\partial w)x$. By using the value for dw and x we obtain

$$dy = x(dw) = 3 + 4\imath + 4\jmath - 8\kappa$$

or

$$dy = (dw)x = 3 - 4\imath + 4\jmath + 8\kappa$$

We observe that out of these two expressions, the second one gives the same value for dy (and is therefore the correct expanded form of $\frac{\partial y}{\partial w} = x$) as that obtained numerically and so the gradient according to the left $\mathbb{HR}$-derivative is indeed correct. If instead we use the right $\mathbb{HR}$-derivative $\frac{\partial(x)}{\partial w} = x_r$ we have

$$\partial y = x_r\partial w = \partial w(x_r) = 3 + 6\imath + 3\jmath + 3\kappa$$

This simple numerical analysis confirms that the right $\mathbb{HR}$-derivative of the function $y = wx$ is not correct and it is the left $\mathbb{HR}$-derivative that must be used.

Evaluating the gradient of $y = xw$ numerically

Taking the same values previously used for w_1,w_2 and x, we can obtain the following

$$y_1 = xw_1 = -10 - 15\imath - 2\jmath - \kappa$$

$$y_2 = xw_2 = -7 - 11\imath + 2\jmath - 3\kappa$$

$$dy = y_2 - y_1 = 3 + 4\imath + 4\jmath - 8\kappa$$

$$dw = w_2 - w_1 = 1 + 2\imath + \jmath + \kappa$$

Using the right $\mathbb{HR}$-derivative in Example 5, we calculated $\frac{\partial y}{\partial w} = x$. This can be written as $\partial y = x\partial w$. Because of the uncertainty of the reciprocal $\frac{1}{\partial w}$, this can also be written as

$\partial y = \partial w x$. By using the value for dw and x we can write the differential dy as

$$dy = x(dw) = 3 + 4\imath + 4\jmath - 8\kappa$$

$$dy = (dw)x = 3 - 4\imath + 4\jmath + 8\kappa$$

We observe that out of these two expressions, the first one gives the same value for dy (and is therefore the correct expanded form of $\frac{\partial y}{\partial w} = x$) as that obtained numerically and so the right $\mathbb{HR}$-derivative gives the correct value. By repeating the procedure above with the left $\mathbb{HR}$-derivative, we can show that the left $\mathbb{HR}$-derivative does not give the correct value.

Although the numerical analysis carried out confirms that both functions have the same derivative ($\frac{\partial y}{\partial w} = x$), the difference between both functions is reflected by the fact that for the function $y_{left} = wx$, the differential can be written as $\partial y_{left} = (\partial w)x$ whereas for the function $y_{right} = xw$, the differential can be written as $\partial y_{right} = x\partial w$.

3.3 Quaternion gradients

Consider the first-order multivariate Taylor series expansion (TSE) of a function $f(q, q^\imath, q^\jmath, q^\kappa) : \mathbb{H}^4 \to \mathbb{R}$, given by

$$df = \frac{\partial f}{\partial q}dq + \frac{\partial f}{\partial q}dq^\imath + \frac{\partial f}{\partial q}dq^\jmath + \frac{\partial f}{\partial q}dq^\kappa \tag{3.21}$$

For a real function of quaternion variables (e.g. the typical cost function $f(q) = ee^* = |e|^2$), we have

$$\frac{\partial f^\eta(q)}{\partial q} = \left(\frac{\partial f(q)}{\partial q^\eta}\right)^\eta \tag{3.22}$$

for $\eta = \{\imath, \jmath, \kappa\}$. Using the involution property P2 in (2.9), it follows that

$$df = \frac{\partial f}{\partial q}dq + \left(\frac{\partial f}{\partial q^\imath}dq\right)^\imath + \left(\frac{\partial f}{\partial q^\jmath}dq\right)^\jmath + \left(\frac{\partial f}{\partial q^\kappa}dq\right)^\kappa \tag{3.23}$$

Upon applying the identity (3.16) and the conjugate property $(q_1 q_2)^* = q_2^* q_1^*$, this yields

$$df = \frac{\partial f}{\partial q} dq + 2dq^* \frac{\partial f}{\partial q^*} + \frac{\partial f}{\partial q} dq = 2\left(\frac{\partial f}{\partial q} dq + dq^* \frac{\partial f}{\partial q^*}\right)$$

finally arriving at[2]

$$df = 4\Re\left(\frac{\partial f}{\partial q} dq\right)$$

where the symbol $\Re(\cdot)$ denotes the real (scalar) part of a quaternion. This result can be extended to the quaternion vector variables, where $\mathbf{q} = [q_1, q_2, ..., q_N]^T$ and $\nabla_q f = \frac{\partial f}{\partial \mathbf{q}} = \left[\frac{\partial f}{\partial q_1}, \frac{\partial f}{\partial q_2}, ..., \frac{\partial f}{\partial q_N}\right]^T$, so that

$$df = 2\left(\frac{\partial f}{\partial \mathbf{q}}\right)^T d\mathbf{q} + 2(d\mathbf{q}^*)^T \frac{\partial f}{\partial \mathbf{q}^*} = 4\Re\left\{\left(\frac{\partial f}{\partial \mathbf{q}}\right)^T d\mathbf{q}\right\}$$

Upon substituting $\frac{\partial f}{\partial q^*} = \left(\frac{\partial f}{\partial q}\right)^*$ to give $df = 4\Re\left\{\left(\frac{\partial f}{\partial \mathbf{q}^*}\right)^H d\mathbf{q}\right\}$ and applying the Schwarz inequality[3] we have

$$df = 4\Re|\langle \frac{\partial f}{\partial \mathbf{q}^*}, d\mathbf{q}\rangle| \rightarrow df \leq 4\|\frac{\partial f}{\partial \mathbf{q}^*}\|\|d\mathbf{q}\| \tag{3.24}$$

Remark#1: The maximum change of the differential df occurs when $d\mathbf{q}$ is in the direction of $\frac{\partial f}{\partial \mathbf{q}^*}$, thus making the conjugate gradient $\nabla_{q^*} f$ a natural choice of gradient in the optimization of real valued functions of quaternion variables.

3.3.1 Unifying the quaternion gradient: The I-gradient

As shown in the previous section, the gradient based on the $\mathbb{HR}^*$-derivatives is rigorous and has provided a quantum step forward in the derivation of stochastic gradient algorithms in the quaternion domain. However, despite its mathematical correctness it does not necessarily equip the quaternion domain with algorithms that are generic extensions of its real- and complex- valued counterparts, the LMS or CLMS (as seen from [66]). It is therefore necessary to introduce a novel gradient definition that not only unifies the QLMS algo-

[2]Using the relationship between a quaternion and its conjugate, that is, $q_r = \frac{1}{2}(q + q^*)$.

[3]The inner product of two vectors $\mathbf{a}$ and $\mathbf{b}$ satisfies $\langle \mathbf{a}, \mathbf{b}\rangle \leq \|\mathbf{a}\|\|\mathbf{b}\|$, where the equality holds when $\mathbf{a}$ is collinear with $\mathbf{b}$, that is, $\mathbf{a} = \lambda\mathbf{b}$.

rithms but also simplifies into CLMS and LMS for lower dimensional data. To that end, consider the involution based representation in (2.5), where $q = \frac{1}{2}\left(q^{\imath^*} + q^{\jmath^*} + q^{\kappa^*} - q^*\right)$ and $q^* = \frac{1}{2}\left(q^{\imath} + q^{\jmath} + q^{\kappa} - q\right)$. In this way, we can write the partial derivative $\frac{\partial f}{\partial q}$ and its conjugate $\left(\frac{\partial f}{\partial q}\right)^*$ as

$$\frac{\partial f}{\partial q} = \frac{1}{2}\left(\left(\frac{\partial f}{\partial q}\right)^{\imath^*} + \left(\frac{\partial f}{\partial q}\right)^{\jmath^*} + \left(\frac{\partial f}{\partial q}\right)^{\kappa^*} - \left(\frac{\partial f}{\partial q}\right)^*\right) \tag{3.25}$$

$$\left(\frac{\partial f}{\partial q}\right)^* = \frac{1}{2}\left(\left(\frac{\partial f}{\partial q}\right)^{\imath} + \left(\frac{\partial f}{\partial q}\right)^{\jmath} + \left(\frac{\partial f}{\partial q}\right)^{\kappa} - \frac{\partial f}{\partial q}\right) \tag{3.26}$$

For a real valued function $f(q)$, using the identity in (3.22) and the fact that for a real function of quaternion variables $\frac{\partial f}{\partial q} = \left(\frac{\partial f(q)}{\partial q^*}\right)^*$, we arrive at

$$\frac{\partial f}{\partial q} = \frac{1}{2}\left(\frac{\partial f}{\partial q^{\imath^*}} + \frac{\partial f}{\partial q^{\jmath^*}} + \frac{\partial f}{\partial q^{\kappa^*}} - \frac{\partial f}{\partial q^*}\right) \tag{3.27}$$

$$\frac{\partial f}{\partial q^*} = \frac{1}{2}\left(\frac{\partial f}{\partial q^{\imath}} + \frac{\partial f}{\partial q^{\jmath}} + \frac{\partial f}{\partial q^{\kappa}} - \frac{\partial f}{\partial q}\right) \tag{3.28}$$

Observe from (3.24), that it is the conjugate gradient

$$\frac{\partial f}{\partial q^*} = \frac{1}{2}\left(\frac{\partial f}{\partial q^{\imath}} + \frac{\partial f}{\partial q^{\jmath}} + \frac{\partial f}{\partial q^{\kappa}} - \frac{1}{2}\left(\frac{\partial f}{\partial q^{\imath *}} + \frac{\partial f}{\partial q^{\jmath *}} + \frac{\partial f}{\partial q^{\kappa *}} - \frac{\partial f}{\partial q^*}\right)\right)$$

that provides the maximum rate of change (steepest direction). The conjugate gradient can also be written as

$$\frac{3}{4}\frac{\partial f}{\partial q*} = \frac{1}{2}\left(\frac{1}{2}\left(\frac{\partial f}{\partial q^{\imath}} + \frac{\partial f}{\partial q^{\jmath}} + \frac{\partial f}{\partial q^{\kappa}}\right) + \frac{1}{2}\left(\frac{\partial f}{\partial q^{\imath}} - \frac{\partial f}{\partial q^{\imath *}}\right) + \frac{1}{2}\left(\frac{\partial f}{\partial q^{\jmath}} - \frac{\partial f}{\partial q^{\jmath *}}\right) + \frac{1}{2}\left(\frac{\partial f}{\partial q^{\kappa}} - \frac{\partial f}{\partial q^{\kappa *}}\right)\right)$$

and after substituting $\frac{1}{2}(q - q^*) = \Im(q)$, where $\Im$ stands for the imaginary (vector) part of q, and using $\frac{\partial f(q)}{\partial q^*} = \left(\frac{\partial f(q)}{\partial q}\right)^*$, we have

$$\frac{\partial f}{\partial q^*} = \frac{1}{3}\left(\left(\frac{\partial f}{\partial q^{\imath}} + \frac{\partial f}{\partial q^{\jmath}} + \frac{\partial f}{\partial q^{\kappa}}\right) + 2\left(\Im\left(\frac{\partial f}{\partial q^{\imath}}\right) + \Im\left(\frac{\partial f}{\partial q^{\jmath}}\right) + \Im\left(\frac{\partial f}{\partial q^{\kappa}}\right)\right)\right) \tag{3.29}$$

Notice that now both the real and imaginary parts are a function of only $\frac{\partial f}{\partial q^\imath}$, $\frac{\partial f}{\partial q^\jmath}$ and $\frac{\partial f}{\partial q^\kappa}$, that is

$$\Re\left[\frac{\partial f}{\partial q^*}\right] + \Im\left[\frac{\partial f}{\partial q^*}\right] = \frac{1}{3}\Re\left[\frac{\partial f}{\partial q^\imath} + \frac{\partial f}{\partial q^\jmath} + \frac{\partial f}{\partial q^\kappa}\right] + \Im\left[\frac{\partial f}{\partial q^\imath} + \frac{\partial f}{\partial q^\jmath} + \frac{\partial f}{\partial q^\kappa}\right] \tag{3.30}$$

and that the direction of the vector part of the gradient $\frac{\partial f}{\partial q^*}$ is equivalent to that within the $\mathbb{HR}$-calculus, that is, $\frac{\partial f}{\partial q^\imath} + \frac{\partial f}{\partial q^\jmath} + \frac{\partial f}{\partial q^\kappa}$. In this way, we introduce a novel quaternion gradient, termed the involution- or I-gradient, given by

$$\nabla_q J = \frac{\partial J}{\partial q^\imath} + \frac{\partial J}{\partial q^\jmath} + \frac{\partial J}{\partial q^\kappa} \tag{3.31}$$

Using (3.28) and the $\mathbb{HR}$- and $\mathbb{HR}^*$-derivatives in (3.9) and (3.15), the I-gradient can be written in terms of the conjugate gradient as

$$\nabla_q J = \frac{\partial J}{\partial q^*} + \frac{1}{2}\frac{\partial J}{\partial q_r} \tag{3.32}$$

Notice that the I-gradient is equivalent to the $\mathbb{HR}$-gradient for 3D processes (pure quaternions), a typical case in most real world applications.

Remark#2: For 4D processes (full quaternion), the I-gradient includes an additional term $\frac{1}{2}\frac{\partial J}{\partial q_r}$ as compared to the $\mathbb{HR}$-gradient, thus exhibiting increased steepness in the direction of the real component, and providing a faster convergence of the so derived learning algorithms. However, as the sign of the real gradient component remains unchanged, both gradients converge to the same optimum solution.

3.4 A unifying treatment of quaternion LMS algorithms

A major stumbling block in the development of quaternion-valued adaptive filtering algorithms has been the issue of quaternion gradient, which in its standard form introduces uncertainties in the position of the imaginary units and is thus not unique. We shall now illustrate that by using the quaternion conjugate gradient defined by $\nabla_{w^*} J$ (as given by the $\mathbb{HR}$-calculus), a real function of quaternion variables can be differentiated directly, without

resorting to tedious component-wise partial gradients or introducing any uncertainty in the placement of the unit vectors.

This also gives us the opportunity to unify a number of existing adaptive filtering algorithms in $\mathbb{H}$, and to provide a rigorous platform for future developments. In addition, we show how the I-gradient $\nabla_w J = \nabla_{w^\imath} J + \nabla_{w^\jmath} J + \nabla_{w^\kappa} J$, derived in Section 3.3.1 can be used to introduce generic quaternion valued extensions of the existing real and complex adaptive filters, both strictly linear and widely linear ones. To be consistent with the original derivation of the QLMS in [66], the filter output assumes the form $y(k) = \mathbf{w}^T(k)\mathbf{x}(k)$, however, any other inner product of $\mathbf{w}(k)$ and $\mathbf{x}(k)$, e.g. $\mathbf{x}^T(k)\mathbf{w}(k)$, can be equally used to derive the algorithms.

3.4.1 The quaternion Wiener solution

The Wiener filter aims to find the optimal coefficient (weight) vector that minimizes the mean square error, or equivalently to reach the minimum of the cost function, given by

$$J(k) = E[e(k)e^*(k)] = E[|e(k)|^2] \tag{3.33}$$

where $e(k)$ is the output error of an adaptive filter

$$e(k) = d(k) - \mathbf{w}^T\mathbf{x}(k)$$

and $\mathbf{w} \in \mathbb{H}^{N\times 1}$ and $d(k) \in \mathbb{H}$ are respectively the filter weight vector and teaching signal. To obtain the optimal weight vector $\mathbf{w}_o$, we need to calculate the gradient (using the conjugate gradient in (3.15)) in the form

$$\nabla_{\mathbf{w}^*} J = \frac{\partial J(k)}{\partial \mathbf{w}^*} = E\left[e(k)\frac{\partial e^*(k)}{\partial \mathbf{w}^*} + \frac{\partial e(k)}{\partial \mathbf{w}^*}e^*(k)\right] \tag{3.34}$$

Normally, this is not straightforward to achieve using the pseudogradient, however, by virtue of the $\mathbb{HR}$-calculus in (3.34), we directly obtain

$$\frac{\partial J(k)}{\partial \mathbf{w}^*} = E\left[\frac{1}{2}\mathbf{x}(k)e^*(k) - e(k)\mathbf{x}^*(k)\right]$$

Substitute for $e(k)$ and set to zero to give

$$E\left[\frac{1}{2}\mathbf{x}(k)\big(d^*(k)-\mathbf{x}^H(k)\mathbf{w}^*\big)-\big(d(k)-\mathbf{w}^T\mathbf{x}(k)\big)\mathbf{x}^*(k)\right]=\mathbf{0}$$

Upon rearranging the terms above we have

$$E\big[\mathbf{w}^T\mathbf{x}(k)\mathbf{x}^*(k)-\frac{1}{2}\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}^*\big]=E\big[d(k)\mathbf{x}^*(k)-\frac{1}{2}\mathbf{x}(k)d^*(k)\big]$$

or in a more compact form

$$\big(\mathbf{w}^T\mathbf{R}_{xx}\big)^T-\frac{1}{2}\mathbf{R}_{xx}\mathbf{w}^*=\mathbf{r}_{dx}-\frac{1}{2}\mathbf{r}_{xd} \tag{3.35}$$

Noting that $\big(\mathbf{w}^T\mathbf{R}_{xx}\big)^H=\mathbf{R}_{xx}\mathbf{w}^*$ and $(\mathbf{r}_{dx})^*=\mathbf{r}_{xd}$, we arrive at the strictly linear *quaternion Wiener filter* in the form

$$\mathbf{w}_o^T=\mathbf{r}_{dx}^T\mathbf{R}_{xx}^{-1} \tag{3.36}$$

Notice the subtle difference from the real- and complex-valued Wiener solution - the quaternion solution is given in terms of the transpose of $\mathbf{w}$ as the rigid noncommutativity of $\mathbf{r}_{dx}^T\mathbf{R}_{xx}^{-1}$ means that $\mathbf{R}_{xx}^{-1}\mathbf{r}_{dx}$ would represent a different solution.

3.4.2 The widely linear wiener solution

When the system model is widely linear as in (2.2.3), the augmented filter weight takes the form $\mathbf{w}^a=[\mathbf{u}^T,\mathbf{v}^T,\mathbf{g}^T,\mathbf{h}^T]^T$ and the augmented input vector $\mathbf{x}^a=[\mathbf{x}^T,\mathbf{x}^{\imath T},\mathbf{x}^{\jmath T},\mathbf{x}^{\kappa T}]$. Following on (3.36), the widely linear Wiener solution then becomes

$$\mathbf{w}_o^{aT}=\mathbf{r}_{dx}^{aT}\left(\mathbf{R}_{xx}^a\right)^{-1} \tag{3.37}$$

where $\mathbf{R}_{xx}^a$ is the augmented covariance matrix, described in (2.19) and $\mathbf{r}_{dx}^{aT}=E[d\mathbf{x}^H,d\mathbf{x}^{\imath H},d\mathbf{x}^{\jmath H},d\mathbf{x}^{\kappa H}]^T$. This model represents an optimal second order solution for both proper and improper signals, whereas the standard, strictly linear, Wiener filter in (3.36) is optimal only for proper signals, for which the pseudocovariances matrices $\mathbf{P}_{xx}$, $\mathbf{S}_{xx}$ and $\mathbf{T}_{xx}$ in (2.19) vanish.

3.4.3 The quaternion least mean square algorithm

We now illustrate the advantages of the quaternion gradients introduced in this work, by providing an elegant derivation of the strictly linear QLMS, for which the original derivation [66] used cumbersome component-wise pseudogradients. Starting from the cost function in (3.33), the stochastic gradient weight update is given by (μ is the real-valued step size)

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \nabla_{\mathbf{w}^*} J(k) \tag{3.38}$$

and noting that

$$e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k) \qquad\qquad e^*(k) = d^*(k) - \mathbf{x}^H(k)\mathbf{w}^*(k)$$

the gradient of the cost function becomes

$$\nabla_{\mathbf{w}^*} J(k) = \frac{1}{2}\left(e(k)\frac{\partial e^*(k)}{\partial \mathbf{w}^*(k)} + \frac{\partial e(k)}{\partial \mathbf{w}^*(k)} e^*(k) \right) \tag{3.39}$$

Using the $\mathbb{HR}$-gradient $\nabla_{\mathbf{w}^*}$, we can differentiate $e(k)$ and $e^*(k)$ directly, without resorting to the partial derivatives with respect to $\mathbf{w}_r$, $\mathbf{w}_\imath$, $\mathbf{w}_\jmath$ and $\mathbf{w}_\kappa$, thus giving

$$\frac{\partial e^*(k)}{\partial \mathbf{w}^*} = -\mathbf{x}^*(k) \tag{3.40}$$

On the other hand, to calculate $\frac{\partial e(k)}{\partial \mathbf{w}^*(k)}$, we must first write $\mathbf{w}$ in terms of the conjugate involutions $\mathbf{w}^*, \mathbf{w}^{\imath *}, \mathbf{w}^{\jmath *}, \mathbf{w}^{\kappa *}$, so that the argument takes the form required by (3.15), given by

$$e(k) = d(k) - \frac{1}{2}\left(\mathbf{w}^{\imath *}(k) + \mathbf{w}^{\jmath *}(k) + \mathbf{w}^{\kappa *}(k) - \mathbf{w}^*(k)\right)^T \mathbf{x}(k) \tag{3.41}$$

Using the $\mathbb{HR}$-calculus in (3.15) to perform the direct differentiation of $e(k)$ gives

$$\frac{\partial e(k)}{\partial \mathbf{w}^*} = \frac{1}{2}\mathbf{x}(k) \tag{3.42}$$

and combining the terms above yields the update, referred to as the HR-QLMS, in the form

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\left(\frac{1}{2}e(k)\mathbf{x}^*(k) - \frac{1}{4}\mathbf{x}(k)e^*(k)\right) \tag{3.43}$$

A comparison with the original QLMS algorithm in [66] whose form is

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\big(2e(k)\mathbf{x}^*(k) - \mathbf{x}^*(k)e^*(k)\big) \tag{3.44}$$

leads to the following observations.

Remark#3: The standard QLMS weight update term is four times larger than that within the HR-QLMS, the difference arising due to the component-wise gradient calculation in the standard QLMS. However, this difference can be absorbed into the learning rate μ.

Remark#4: The second term in the original QLMS weight update uses $\mathbf{x}^*(k)$ instead of $\mathbf{x}(k)$; this difference arises due to the interpretation of the gradient $\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}^*}$ in (3.15). In the standard QLMS in [66] it is assumed that the unit vectors $\imath$, $\jmath$ and κ are always on the right hand side of the real component-wise derivatives $\frac{\partial f}{\mathbf{q}_r}$, $\frac{\partial f}{\mathbf{q}_\imath}$, $\frac{\partial f}{\mathbf{q}_\jmath}$ and $\frac{\partial f}{\mathbf{q}_\kappa}$. In the QLMS derivation based on the $\mathbb{HR}$-gradient, having the imaginary units before the derivatives $\frac{\partial f}{\mathbf{q}_r}$, $\frac{\partial f}{\mathbf{q}_\imath}$, $\frac{\partial f}{\mathbf{q}_\jmath}$ and $\frac{\partial f}{\mathbf{q}_\kappa}$ for the gradient $\frac{\partial e}{\partial \mathbf{w}^*}$ and after the partial derivatives for the gradient $\frac{\partial e^*}{\partial \mathbf{w}^*}$ results in the HR-QLMS weight update in (3.43).

Section 3.5 provides an in-depth comparative analysis, and shows that these two expressions effectively provide the same solution.

3.4.4 The widely linear least mean square algorithm (WLQLMS)

Using the widely linear model in Section 2.2.3 and following on the analysis of the QLMS class of algorithms, we can now rederive the WLQLMS [67] in an efficient way, and establish an elegant and rigorous framework for the derivation of widely linear quaternion valued algorithms that account for the improperness of quaternion data. We assume that the output of the filter takes the form $y(k) = \mathbf{w}^{aT}(k)\mathbf{x}^a(k)$ where the augmented weight vector $\mathbf{w}^a(k) = [\mathbf{u}^T(k), \mathbf{v}^T(k), \mathbf{g}^T(k), \mathbf{h}^T(k)]^T$ and the augmented input vector $\mathbf{x}^a(k) = [\mathbf{x}^T(k), \mathbf{x}^{\imath T}(k), \mathbf{x}^{\jmath T}(k), \mathbf{x}^{\kappa T}(k)]^T$. The weight updates are calculated from (3.38), based on

the gradient

$$\nabla_{\mathbf{w}} J(k) = \frac{1}{2}\left(e(k)\frac{\partial e^*(k)}{\partial \mathbf{w}^{a*}(k)} + \frac{\partial e(k)}{\partial \mathbf{w}^{a*}(k)} e^*(k)\right) \tag{3.45}$$

By virtue of the $\mathbb{HR}$-calculus, the partial derivatives $\left[\frac{\partial e^*(k)}{\partial \mathbf{u}^*(k)}, \frac{\partial e^*(k)}{\partial \mathbf{v}^*(k)}, \frac{\partial e^*(k)}{\partial \mathbf{g}^*(k)}, \frac{\partial e^*(k)}{\partial \mathbf{h}^*(k)}\right]^T$ within the gradient $\frac{\partial e^*(k)}{\partial \mathbf{w}^{a*}(k)}$ can be calculated by direct differentiation, giving

$$\frac{\partial e^*(k)}{\partial \mathbf{u}^*(k)} = -\mathbf{x}^*(k) \quad \frac{\partial e^*(k)}{\partial \mathbf{v}^*(k)} = -\mathbf{x}^{\imath^*}(k) \quad \frac{\partial e^*(k)}{\partial \mathbf{g}^*(k)} = -\mathbf{x}^{\jmath^*}(k) \quad \frac{\partial e^*(k)}{\partial \mathbf{h}^*(k)} = -\mathbf{x}^{\kappa^*}(k) \tag{3.46}$$

To obtain the partial derivatives $\frac{\partial e(k)}{\partial \mathbf{w}^{a*}(k)} = \left[\frac{\partial e(k)}{\partial \mathbf{u}^*(k)}, \frac{\partial e(k)}{\partial \mathbf{v}^*(k)}, \frac{\partial e(k)}{\partial \mathbf{g}^*(k)}, \frac{\partial e(k)}{\partial \mathbf{h}^*(k)}\right]$, substitute $\mathbf{w}^a = \frac{1}{2}\left((\mathbf{w}^a)^{\imath *} + (\mathbf{w}^a)^{\jmath *} + (\mathbf{w}^a)^{\kappa *} - \mathbf{w}^{a*}\right)$, to yield

$$\frac{\partial e(k)}{\partial \mathbf{u}^*(k)} = \frac{1}{2}\mathbf{x}(k) \quad \frac{\partial e(k)}{\partial \mathbf{v}^*(k)} = \frac{1}{2}\mathbf{x}^{\imath}(k) \quad \frac{\partial e(k)}{\partial \mathbf{g}^*(k)} = \frac{1}{2}\mathbf{x}^{\jmath}(k) \quad \frac{\partial e(k)}{\partial \mathbf{h}^*(k)} = \frac{1}{2}\mathbf{x}^{\kappa}(k) \tag{3.47}$$

finally arriving at the weight update for the WLQLMS algorithm in the form

$$\mathbf{w}^a(k+1) = \mathbf{w}^a(k) + \mu\left(\frac{1}{2}e(k)\mathbf{x}^{a*}(k) - \frac{1}{4}\mathbf{x}^a(k)e^*(k)\right) \tag{3.48}$$

This expression for the WLQLMS differs from the existing WLQLMS algorithm introduced in [67], the differences arising for the same reason as those for the QLMS.

3.4.5 A class of generic QLMS algorithms

Upon applying the I-gradient derived in Section 3.3.1 to the cost function in (3.33), we obtain

$$\begin{aligned}\nabla_{\mathbf{w}} J(k) &= \frac{\partial J}{\partial w^{\imath}} + \frac{\partial J}{\partial w^{\jmath}} + \frac{\partial J}{\partial w^{\kappa}} \\ &= \frac{1}{2}\left(e(k)\frac{\partial e^*(k)}{\partial \mathbf{w}^{\imath}(k)} + \frac{\partial e(k)}{\partial \mathbf{w}^{\imath}(k)}e^*(k) + e(k)\frac{\partial e^*(k)}{\partial \mathbf{w}^{\jmath}(k)} + \frac{\partial e(k)}{\partial \mathbf{w}^{\jmath}(k)}e^*(k)\right. \\ &+ \left. e(k)\frac{\partial e^*(k)}{\partial \mathbf{w}^{\kappa}(k)} + \frac{\partial e(k)}{\partial \mathbf{w}^{\kappa}(k)}e^*(k)\right)\end{aligned} \tag{3.49}$$

Using the $\mathbb{HR}$-calculus, $\frac{\partial e(k)}{\partial \mathbf{w}^{\imath}} = \frac{\partial e(k)}{\partial \mathbf{w}^{\jmath}} = \frac{\partial e(k)}{\partial \mathbf{w}^{\kappa}} = \mathbf{0}$, and similarly to the derivation of QLMS, we can write

$$e^*(k) = d^*(k) - \frac{1}{2}\big(\mathbf{w}^{\imath}(k) + \mathbf{w}^{\jmath}(k) + \mathbf{w}^{\kappa}(k) - \mathbf{w}(k)\big)^T \mathbf{x}(k)$$

By direct differentiation, $\frac{\partial e^*(k)}{\partial \mathbf{w}^{\imath}} = \frac{\partial e^*(k)}{\partial \mathbf{w}^{\jmath}} = \frac{\partial e^*(k)}{\partial \mathbf{w}^{\kappa}} = -\frac{1}{2}\mathbf{x}(k)$ yielding the weight update

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{3}{4}\mu e(k)\mathbf{x}^*(k) \tag{3.50}$$

This algorithm is termed the IQLMS (the I-gradient based QLMS).

Remark#5: The IQLMS has the same generic form as the LMS and CLMS [21], as the term $\frac{3}{4}$ can be absorbed into the learning rate.

Remark#6: The IQLMS requires half the computations of the QLMS.

In the same spirit, from (3.48) in order to derive the WLIQLMS, a generic extension of the complex widely linear ACLMS [38] algorithm, we employ the I-gradient to obtain

$$\mathbf{w}^a(k+1) = \mathbf{w}^a(k) + \frac{3}{4}\mu e(k)\mathbf{x}^{a*}(k) \tag{3.51}$$

where $\mathbf{x}^a$ is the augmented input vector. It is easy to show, following the approach in [112] (see Appendix A) that the WLIQLMS is structurally identical to the four-channel real LMS, whereas the WLQLMS is not.

Remark#7: Note that the Wiener filter is already in its generic form and hence the corresponding algorithm derived using the I-gradient would be the same.

3.5 Equivalence of the various forms of QLMS

Starting from the expression for the original QLMS given by [66]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\left(\frac{1}{2}e(k)\mathbf{x}^*(k) - \frac{1}{4}\mathbf{x}^*(k)e^*(k)\right) \tag{3.52}$$

together with the expression for the proposed HR-QLMS

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\left(\frac{1}{2}e(k)\mathbf{x}^*(k) - \frac{1}{4}\mathbf{x}(k)e^*(k)\right) \tag{3.53}$$

and the IQLMS algorithm

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{3}{4}\mu e(k)\mathbf{x}^*(k) \tag{3.54}$$

we can now rewrite the QLMS algorithms (3.52)-(3.54) in a form that allows for a convenient analytical comparison.

Based on (3.52), from $\mathbf{x}^*(k) = \mathbf{x}(k) - 2\Im[\mathbf{x}(k)]$ and by splitting the term $\frac{1}{2}e(k)\mathbf{x}^*(k)$ we have

$$\begin{aligned}
\mathbf{w}(k+1) &= \mathbf{w}(k) + \frac{1}{4}\mu e(k)\mathbf{x}^*(k) + \frac{1}{4}\mu e(k)\mathbf{x}^*(k) - \frac{1}{4}\mu\mathbf{x}(k)e^*(k) + \frac{1}{2}\mu\Im[\mathbf{x}(k)]e^*(k) \\
&= \mathbf{w}(k) + \mu\frac{1}{2}\Im[e(k)\mathbf{x}^*(k)] + \frac{1}{4}\mu e(k)\mathbf{x}^*(k) + \frac{1}{2}\mu\Im[\mathbf{x}(k)]e^*(k) \\
&= \mathbf{w}(k) + \mu\frac{1}{2}\Im[e(k)\mathbf{x}^*(k)] + \frac{1}{4}\mu e(k)\mathbf{x}^*(k) - \frac{1}{2}\mu\big[e(k)\Im[\mathbf{x}(k)]\big]^*
\end{aligned}$$

The real part of the weight update can therefore be evaluated as

$$\Re[\mathbf{w}(k+1)] = \Re[\mathbf{w}(k)] + \frac{1}{4}\mu\Re[e(k)\mathbf{x}^*(k)] - \frac{1}{2}\mu\Re\big[e(k)\Im[\mathbf{x}(k)]\big]$$

which, upon using $\mathbf{x}(k) = \Re[\mathbf{x}(k)] + \Im[\mathbf{x}(k)]$, yields

$$\Re[\mathbf{w}(k+1)] = \Re[\mathbf{w}(k)] + \frac{1}{4}\mu\Re\big[e(k)\Re[\mathbf{x}(k)]\big] - \frac{3}{4}\mu\Re\big[e(k)\Im[\mathbf{x}(k)]\big] \tag{3.55}$$

Similarly, for the imaginary part of the QLMS weight update in (3.52), we have

$$\Im[\mathbf{w}(k+1)] = \Im[\mathbf{w}(k)] + \frac{3}{4}\mu\Im[e(k)\mathbf{x}^*(k)] + \frac{1}{2}\mu\Im\big[e(k)\Im[\mathbf{x}(k)]\big]$$

$$\Im[\mathbf{w}(k+1)] = \Im[\mathbf{w}(k)] + \frac{3}{4}\mu\Im\big[e(k)\Re[\mathbf{x}(k)]\big] - \frac{1}{4}\mu\big[e(k)\Im[\mathbf{x}(k)]\big] \tag{3.56}$$

Repeating the above process for the weight updates in (3.53) and (3.54), it can be shown that all the various forms of the updates are based on a combination of the terms,

$+[e(k)\Re[\mathbf{x}(k)]]$ and $-[e(k)\Im[\mathbf{x}(k)]]$, or more explicitly

$$\Re[\mathbf{w}(k+1)] = \Re[\mathbf{w}(k)] + a\ \mu\Re[e(k)\Re[\mathbf{x}(k)]] - b\ \mu\Re[e(k)\Im[\mathbf{x}(k)]] \tag{3.57}$$

$$\Im[\mathbf{w}(k+1)] = \Im[\mathbf{w}(k)] + c\ \mu\Im[e(k)\Re[\mathbf{x}(k)]] - d\ \mu\Im[e(k)\Im[\mathbf{x}(k)]] \tag{3.58}$$

Therefore, the weight updates for QLMS, HR-QLMS and IQLMS differ only in the weighting factors a, b, c and d, that is

- For the QLMS derived in [66], the weighting coefficients are $a = d = \frac{1}{4}$ and $b = c = \frac{3}{4}$.
- For the QLMS in (3.53), the weighting coefficients are $a = b = \frac{1}{4}$ and $c = d = \frac{3}{4}$.
- For the IQLMS in (3.54), the weighting coefficients are $a = b = c = d = \frac{3}{4}$.

Remark#8: The three QLMS algorithms in (3.52)-(3.54) represent the same solution, as the weight updates differ only in the different weightings applied to the four components of the weight vector. This also suggests that the higher complexity of QLMS as opposed to IQLMS does not automatically result in a better performance.

Remark#9: The different weightings in the weight update terms mean that although all the algorithms will arrive at the same Wiener solution, the three versions of the QLMS will exhibit differences in their convergence and steady state performances. For instance, we would expect the IQLMS to have a faster initial convergence than the QLMS, due to the larger weighting of the weight vector components.

Indeed, the analysis in Section 3.6 and simulations in Section 3.7 show that all the three algorithms have similar performances, yet the IQLMS requires only half the operations.

3.6 Convergence analysis

We now provide a unified platform for the convergence analysis of the class of QLMS algorithms. For rigour, this is achieved starting from the stability conditions through to

the mean square error performance for noncircular data of both strictly linear and widely linear filters. The mean square error analysis is conducted for the IQLMS and also provides a bound on the performance of the QLMS and HR-QLMS.

3.6.1 Convergence in the mean of the strictly linear IQLMS algorithm

To simplify the derivation, the output is re-written in the form $\mathbf{w}^H\mathbf{x}$; this does not change any of its properties, but makes the analysis more mathematically tractable. Without loss of generality, assume first that the teaching signal for the IQLMS is given by (output of the strictly linear model)

$$d(k) = \mathbf{w}_o^H\mathbf{x}(k) + n(k)$$

where $\mathbf{w}_o$ is the optimal filter weight vector and $n(k)$ quaternion quadruply white zero mean Gaussian noise[4]. The output error can therefore be written as

$$e(k) = \mathbf{w}_o^H\mathbf{x}(k) + n(k) - \mathbf{w}^H(k)\mathbf{x}(k) \tag{3.59}$$

Substitute $e(k)$ into the IQLMS weight update in (3.54) to obtain

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{3}{4}\mu\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}_o(k) + \frac{3}{4}\mu\mathbf{x}(k)n^*(k) - \frac{3}{4}\mu\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}(k)$$

and subtract $\mathbf{w}_o$ from both sides to give the weight error vector $\mathbf{r}(k) = \mathbf{w}(k) - \mathbf{w}_o$ in the form

$$\mathbf{r}(k+1) = \mathbf{r}(k) - \frac{3}{4}\mu\mathbf{x}(k)\mathbf{x}^H(k)\left(\mathbf{w}(k) - \mathbf{w}_o(k)\right) + \frac{3}{4}\mu\mathbf{x}(k)n^*(k)$$

For convergence in the mean, take the statistical expectation of both sides to yield

$$\begin{aligned} E[\mathbf{r}(k+1)] &= \left(\mathbf{I} - \frac{3}{4}\mu E[\mathbf{x}(k)\mathbf{x}^H(k)]\right) E[\mathbf{r}(k)] + \frac{3}{4}E[\mathbf{x}(k)n^*(k)] \\ &= (\mathbf{I} - \frac{3}{4}\mu\mathbf{R}_{xx})E[\mathbf{r}(k)] \end{aligned} \tag{3.60}$$

The recursion for the weight error vector $\mathbf{r}(k)$ converges for $||\mathbf{I} - \frac{3}{4}\mu\mathbf{R}_{xx}|| < 1$, where the norm is defined as the spectral norm (corresponding to the largest absolute eigenvalue).

[4]Each component of quaternion noise is drawn from a zero mean independent Gaussian distribution with equal variance.

Making use of the unitary transform $\mathbf{R}_{xx} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$, where $\mathbf{Q}$ is the matrix of eigenvectors and $\mathbf{\Lambda}$ is the eigenvalue matrix, and rotating $\mathbf{r}(k)$ to obtain $\mathbf{r}'(k) = \mathbf{Q}^H E[\mathbf{v}(k)]$ yields

$$\mathbf{r}'(k+1) = (\mathbf{I} - \frac{3}{4}\mu\mathbf{\Lambda})\mathbf{r}'(k)$$

Since $\mathbf{I} - \frac{3}{4}\mu\mathbf{\Lambda}$ is diagonal, the 'modes of convergence' can be expressed as

$$r'_n(k+1) = \left(1 - \frac{3}{4}\mu\lambda_n(\mathbf{R}_{xx})\right) r'_n(k) \quad r'_n \in \mathbf{r},\ n = 1, ..., N \tag{3.61}$$

where $\lambda_n(\mathbf{R}_{xx})$ denotes the nth eigenvalue of $\mathbf{R}_{xx}$. Therefore, in order for every $r_n(k) \in \mathbf{r}(k)$ to converge

$$|1 - \frac{3}{4}\mu\lambda_{max}(\mathbf{R}_{xx})| < 1$$

$$0 < \mu < \frac{8}{3\lambda_{max}(\mathbf{R}_{xx})} \tag{3.62}$$

where the symbol $\lambda_{max}(\mathbf{R}_{xx})$ denotes the maximum eigenvalue of $\mathbf{R}_{xx}$. Using the relationship between the trace of a matrix and its eigenvalues, that is, $Tr(\mathbf{R}_{xx}) = \sum_{i=1}^{N} \lambda_i$, the upper bound on the learning rate can be conveniently approximated by

$$0 < \mu < \frac{8}{3Tr(\mathbf{R}_{xx})}$$

where $Tr(\cdot)$ denotes matrix trace operation.

3.6.2 Convergence in the mean of HR-QLMS

We shall first rewrite the weight update of the QLMS as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\left(\frac{1}{2}e(k)\mathbf{x}^*(k) - \frac{1}{4}\mathbf{x}(k)e^*(k)\right) \tag{3.63}$$

By substituting for $e(k)$ into (3.63) and for the weight error vector $\mathbf{r}(k) = \mathbf{w}(k) - \mathbf{w}_o(k)$, it can be shown (see Appendix B.1) that the weight error vector takes the form

$$\begin{aligned}\mathbf{r}^T(k+1) &= \mathbf{r}^T(k) - \frac{1}{2}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right) + \frac{1}{4}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right)^* + \frac{1}{2}\mu n(k)\mathbf{x}^H(k) \\ &\quad -\frac{1}{4}\mu\mathbf{x}^T(k)n^*(k)\end{aligned} \tag{3.64}$$

It is not possible to obtain a recursive expression for $\mathbf{r}(k)$, as for the IQLMS, due to the presence of the conjugate terms $\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right)^*$. Instead, by splitting $\mathbf{r}(k)$ into its real and imaginary components (see Appendix B.2) and taking the statistical expectation, the following recursive expression can be obtained

$$\boldsymbol{\omega}^T(k+1) = \boldsymbol{\omega}^T(k)[\mathbf{I} - \frac{3}{4}\mu\mathbf{R}_b] \tag{3.65}$$

where $\boldsymbol{\omega}(k) = [E[\mathbf{r}_r^T(k)],\imath E[\mathbf{r}_\imath^T(k)],\jmath E[\mathbf{r}_\jmath^T(k)],\kappa E[\mathbf{r}_\kappa^T(k)]]^T$ and

$$\mathbf{R}_b = \begin{pmatrix} \frac{1}{3}[\mathbf{R}_{xx}]_r & \imath[\mathbf{R}_{xx}]_\imath & \jmath[\mathbf{R}_{xx}]_\jmath & \kappa[\mathbf{R}_{xx}]_\kappa \\ \imath\frac{1}{3}[\mathbf{R}_{xx}]_\imath & [\mathbf{R}_{xx}]_r & \kappa[\mathbf{R}_{xx}]_\kappa & \jmath[\mathbf{R}_{xx}]_\jmath \\ \jmath\frac{1}{3}[\mathbf{R}_{xx}]_\jmath & \kappa[\mathbf{R}_{xx}]_\kappa & [\mathbf{R}_{xx}]_r & \imath[\mathbf{R}_{xx}]_\imath \\ \kappa\frac{1}{3}[\mathbf{R}_{xx}]_\kappa & \jmath[\mathbf{R}_{xx}]_\jmath & \imath[\mathbf{R}_{xx}]_\imath & [\mathbf{R}_{xx}]_r \end{pmatrix}$$

and the notation $[\mathbf{R}_{xx}]_\eta$ indicates the η part of $\mathbf{R}_{xx}$ for $\eta = \{r, \imath, \jmath, \kappa\}$. Similarly to the IQLMS, the condition for convergence in the mean becomes

$$|\mathbf{I} - \frac{3}{4}\mu\mathbf{R}_b| < 1 \tag{3.66}$$

while the bound on the step size in terms of the eigenvalues of $\mathbf{R}_b$, becomes

$$0 < \mu < \frac{8}{3\lambda_{max}(\mathbf{R}_b)} \tag{3.67}$$

To compare the performance of the IQLMS and QLMS, the eigenvalues of the matrix $\mathbf{R}_b$ must be related to the eigenvalues of the matrix $\mathbf{R}_{xx}$. As shown in Appendix B.3, this gives

$$\lambda_{max}(\mathbf{R}_b) \approx \lambda_{max}(\mathbf{R}_{xx}) \tag{3.68}$$

$$\lambda_{min}(\mathbf{R}_b) \approx \frac{1}{3}\lambda_{min}(\mathbf{R}_{xx}) \tag{3.69}$$

3.6.3 Convergence in the mean of QLMS

Starting from the weight update of the QLMS in (3.52) it can be shown (see Appendix B.4) that the weight error vector takes the form

$$\begin{aligned}\mathbf{r}^T(k+1) &= \mathbf{r}^T(k) - \frac{1}{2}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right) + \frac{1}{4}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\right)^* + \frac{1}{2}\mu n(k)\mathbf{x}^H(k) \\ &\quad -\frac{1}{4}\mu\mathbf{x}^H(k)n^*(k) \end{aligned} \tag{3.70}$$

Since QLMS is strictly linear, it is second order optimal only for proper signals. For a second order circular $\mathbf{x}(k)$ we have $\mathbf{x}(k)\mathbf{x}^T(k) = -\frac{1}{2}\mathbf{x}(k)\mathbf{x}^H(k)$ (see Appendix B.5) allowing us to arrive at

$$\begin{aligned}\mathbf{r}^T(k+1) &= \mathbf{r}^T(k) - \frac{1}{2}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right) - \frac{1}{8}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right)^* + \frac{1}{2}\mu n(k)\mathbf{x}^H(k) \\ &\quad -\frac{1}{4}\mu\mathbf{x}^H(k)n^*(k)\end{aligned}$$

Similarly to the analysis of HR-QLMS, after splitting $\mathbf{r}(k)$ into its real and imaginary components (as shown in Appendix B.2) and taking the statistical expectation, we have

$$\boldsymbol{\omega}^T(k+1) = \boldsymbol{\omega}^T(k)[\mathbf{I} - \frac{3}{4}\mu\mathbf{R}_c] \tag{3.71}$$

where

$$\mathbf{R}_c = \begin{pmatrix} \frac{5}{6}[\mathbf{R}_{xx}]_r & \imath\frac{1}{2}[\mathbf{R}_x]_\imath & \jmath\frac{1}{2}[\mathbf{R}_{xx}]_\jmath & \kappa\frac{1}{2}[\mathbf{R}_{xx}]_\kappa \\ \frac{5}{6}\imath[\mathbf{R}_{xx}]_\imath & \frac{1}{2}[\mathbf{R}_{xx}]_r & \kappa\frac{1}{2}[\mathbf{R}_{xx}]_\kappa & \jmath\frac{1}{2}[\mathbf{R}_{xx}]_\jmath \\ \frac{5}{6}\jmath[\mathbf{R}_{xx}]_\jmath & \kappa\frac{1}{2}[\mathbf{R}_{xx}]_\kappa & \frac{1}{2}[\mathbf{R}_{xx}]_r & \imath\frac{1}{2}[\mathbf{R}_{xx}]_\imath \\ \frac{5}{6}\kappa[\mathbf{R}_{xx}]_\kappa & \jmath\frac{1}{2}[\mathbf{R}_{xx}]_\jmath & \imath\frac{1}{2}[\mathbf{R}_{xx}]_\imath & \frac{1}{2}[\mathbf{R}_{xx}]_r \end{pmatrix}$$

Therefore, the condition for convergence in the mean becomes

$$|\mathbf{I} - \frac{3}{4}\mu\mathbf{R}_c| < 1 \tag{3.72}$$

and the bound on the stepsize

$$0 < \mu < \frac{8}{3\lambda_{max}(\mathbf{R}_c)} \tag{3.73}$$

Repeating the analysis from Appendix B.3, we can show that the eigenvalues of the matrices $\mathbf{R}_c$ and $\mathbf{R}_{xx}$ are related by

$$\lambda_{max}(\mathbf{R}_c) \approx \frac{5}{6}\lambda_{max}(\mathbf{R}_{xx}) \tag{3.74}$$

$$\lambda_{min}(\mathbf{R}_c) \approx \frac{1}{2}\lambda_{min}(\mathbf{R}_{xx}) \tag{3.75}$$

Remark#10: From the expressions for the maximum eigenvalue of the correlation matrices (3.68) and (3.74), all the three algorithms have very similar stability properties, with the QLMS offering a slightly wider stability range.

Remark#11: From the analysis of the corresponding minimum eigenvalues of the correlation matrices (3.69) and (3.75), the IQLMS is governed by the smallest eigenvalue spread $\left(\frac{\lambda_{max}(\mathbf{R}_{xx})}{\lambda_{min}(\mathbf{R}_{xx})}\right)$ and thus exhibits the fastest convergence rate, followed by QLMS and HRQLMS.

We have illustrated that a structural equivalence exists among the three forms of QLMS algorithms. The analysis in the mean shows that in addition to requiring fewer computational operations, the IQLMS also offers a faster convergence rate. In view of the advantages offered by the IQLMS and the fact that IQLMS keeps the generic form of LMS and CLMS and inherits their mathematical tractability, we provide the MSE analysis for the IQLMS as it is most likely to be used in practical applications.

3.6.4 Mean square error performance of the strictly linear IQLMS for noncircular data

For generality, our analysis is performed for second order noncircular inputs, and naturally simplifies into the corresponding analysis for circular signals. The analysis evaluates the mean square error (MSE), defined as

$$MSE = \lim_{k\to\infty} E[|e(k)|^2] \tag{3.76}$$

for $e(k) = d(k) - \mathbf{w}^H(k)\mathbf{x}(k)$ where $d(k)$ is an improper teaching signal. This is achieved using the approach proposed in [113] [114], based on the energy conservation of filter

weights at each iteration. To this end, we introduce the a priori error $e_a(k)$ and a posteriori error $e_p(k)$ as

$$e_a(k) = \mathbf{r}^H(k)\mathbf{x}(k) \qquad\qquad e_p(k) = \mathbf{r}^H(k+1)\mathbf{x}(k)$$

where $\mathbf{r}(k) = \mathbf{w}(k) - \mathbf{w}_o$ is the weight error vector. The error $e(k)$ can now be written as (see Appendix B.6)

$$e(k) = e_a(k) + \mathbf{w}^{cH}\mathbf{x}^a(k) + n(k) \tag{3.77}$$

and the MSE as

$$MSE = EMSE + \lim_{k\to\infty} E\left[\|\mathbf{w}^{cH}\mathbf{x}^c(k)\|^2\right] + \sigma^2 \tag{3.78}$$

where σ^2 is the noise variance and EMSE is the excess mean square error, given by

$$EMSE = \lim_{k\to\infty} E[\|e_a(k)\|^2] \tag{3.79}$$

while $\mathbf{x}^c(k) = [\mathbf{x}^{\imath T}(k), \mathbf{x}^{\jmath T}(k), \mathbf{x}^{\kappa T}(k)]^T$ and $\mathbf{w}^c = [\mathbf{v}_o^T, \mathbf{g}_o^T, \mathbf{h}_o^T]^T$. The expression for the conservation of energy of the weights (see Appendix B.7) is given by

$$\|\mathbf{r}(k+1)\|^2 + \frac{\|e_a(k)\|^2}{\|\mathbf{x}(k)\|^2} = \|\mathbf{r}(k)\|^2 + \frac{\|e_p(k)\|^2}{\|\mathbf{x}(k)\|^2}$$

Upon taking the expectation of both sides, for the limit $k \to \infty$, we have

$$E\left[\frac{\|e_a(k)\|^2}{\|\mathbf{x}(k)\|^2}\right] = E\left[\frac{\|e_p(k)\|^2}{\|\mathbf{x}(k)\|^2}\right]$$

and upon substituting for $e_p(k) = e_a(k) - \mu e(k)\|\mathbf{x}(k)\|^2$ we arrive at

$$\mu^2 E\left[\|\mathbf{x}(k)\|^2\|e(k)\|^2\right] = \mu E\left[e_a(k)e^*(k)\right] + \mu E\left[e(k)e_a^*(k)\right] \tag{3.80}$$

Using the result in (B.7) and assumption A.1 in Appendix B.6

$$2\mu E[\|e_a(k)\|^2] = \mu^2 E\left[\|\mathbf{x}(k)\|^2\|e_a(k)\|^2\right] + \mu^2 Tr(\mathbf{R}_{xx})\sigma^2 + \mu^2 E\left[\|\mathbf{x}(k)\|^2\|\mathbf{w}^{cH}\mathbf{x}(k)\|^2\right] \tag{3.81}$$

where $E\left[\|\mathbf{x}\|^2\right] = Tr(\mathbf{R}_{xx})$. We can now obtain an expression for $E[\|e_a(k)\|^2]$ under two different conditions:

- For a small step size μ the term $E\left[\|\mathbf{x}(k)\|^2\|e_a(k)\|^2\right]$ in (3.81) becomes negligible and the expression for the EMSE can be written as

$$E[\|e_a(k)\|^2] = \frac{\mu}{2}Tr(\mathbf{R}_{xx})\sigma^2 + \mu^2 E\left[\|\mathbf{x}(k)\|^2\|\mathbf{w}^{cH}\mathbf{x}(k)\|^2\right] \tag{3.82}$$

- For large values of μ where we cannot neglect the term $E\left[\|\mathbf{x}(k)\|^2\|e_a(k)\|^2\right]$, we make the usual independence assumption that $\|\mathbf{x}\|^2$ is statistically independent to $\|e_a(k)\|^2$, giving

$$E[\|e_a(k)\|^2] = \frac{\mu Tr(\mathbf{R}_{xx})\sigma^2 + \mu^2 E\left[\|\mathbf{x}\|^2\|\mathbf{w}^{cH}\mathbf{x}(k)\|^2\right]}{2 - \mu Tr(\mathbf{R}_{xx})} \tag{3.83}$$

3.6.5 Mean square error performance of the widely linear WLIQLMS for noncircular data

The analysis for the MSE performance of the WLIQLMS is similar to that carried out for the IQLMS, the difference arising from the relationship between $e(k)$ and $e_a(k)$, given by

$$e(k) = e_a(k) + n(k) \tag{3.84}$$

that is, the term $\mathbf{w}^{cH}\mathbf{x}^c$ vanishes, and

$$MSE = EMSE + \sigma^2 \tag{3.85}$$

Remark#12: Comparing the MSE of WLIQLMS in (3.85) to the MSE expression for the IQLMS in (3.78), we observe that the term $\|\mathbf{w}^{cH}\mathbf{x}^c(k)\|^2$ is always greater or equal to zero, and therefore subject on the EMSE for the IQLMS being smaller or equal than the EMSE for the WLIQLMS, the MSE of the WLIQLMS will always be smaller or equal than that of the IQLMS.

The expressions for the EMSE of the WLIQLMS are thus given by:

- For a small step size

$$E[\|e_a(k)\|^2] = \frac{\mu}{2} Tr(\mathbf{R}_{xx}^a)\sigma^2 = 2\mu Tr(\mathbf{R}_{xx})\sigma^2 \tag{3.86}$$

- For large values of the stepsize

$$E[\|e_a(k)\|^2] = \frac{\mu Tr(\mathbf{R}_{xx}^a)\sigma^2}{2 - \mu Tr(\mathbf{R}_{xx}^a)} = \frac{2\mu Tr(\mathbf{R}_{xx})\sigma^2}{1 - 2\mu Tr(\mathbf{R}_{xx})} \tag{3.87}$$

where we have used the fact that $Tr(\mathbf{R}_{xx}^a) = 4Tr(\mathbf{R}_{xx})$.

Remark#13: Comparing (3.86)-(3.87) with the EMSE expressions in (3.82)-(3.83), we observe that for circular signals, where $\mathbf{w}^c = \mathbf{0}$, the EMSE of the WLIQLMS is four times greater than that of the IQLMS, though both tend to 0 as the step size μ approaches 0. It follows from this that if the underlying model is know to be strictly, then the IQLMS should be used. From Remark 12, it then follows that the MSE of IQLMS is always smaller than that of WLIQLMS if the signal is circular.

For noncircular signals, we observe that while the term containing the noise variance is four times larger for the WLIQLMS in (3.86)-(3.87) than the IQLMS in (3.82)-(3.83), we must also consider the effect of the second term $\|\mathbf{w}^{cH}\mathbf{x}^c(k)\|^2$ in the EMSE of the IQLMS. For all noncircular signals, other than those that are marginally noncircular, the latter term will be the dominant factor and hence the EMSE of the IQLMS will be larger than that of the WLIQLMS for noncircular signals. From Remark 12, it then follows that the MSE of IQLMS is always greater than that of WLIQLMS if the signal is noncircular.

Remark#14: For two-dimensional signals the analysis for the MSE of IQLMS and WLIQLMS simplifies respectively into that of the complex LMS and the widely linear CLMS (CLMS and ACLMS).

3.7 Simulations

In order to experimentally validate the analysis, we next evaluated the QLMS, HR-QLMS and IQLMS and their widely linear extensions on both circular and non-circular

quaternion-valued data.

Figure 3.1 and Figure 3.2 show the performances of the HR-QLMS and IQLMS in the prediction setting on the circular AR(4) signal given by

$$y(k) = 1.79y(k-1) - 1.85y(k-2) + 1.27y(k-3) - 0.41y(k-4) + n(k) \tag{3.88}$$

and the circular MA(4) signal

$$y(k) = ax(k) + bx(k-1) + cx(k-2) + dx(k-3) + ex(k-4) + n(k) \tag{3.89}$$

where $a, \ldots, e$ are quaternion valued weights and $n(k)$ quadruply white circular Gaussian noise. For both benchmark models the driving noise $n(k)$ had a variance of 0.1. For the AR(4) signal, all the filters had a step size of $\mu = 0.08$ whereas for the MA(4) signal, all the filters had a step size of $\mu = 0.04$. The experiments were repeated independently 100 times and the performances were averaged to obtain the learning curves. Observe from Figure 3.1 that for the circular AR(4) signal, IQLMS and QLMS offered marginally faster convergence than HR-QLMS, conforming with the analysis. In the steady state (for $k > 10000$), all the three QLMS algorithms exhibited the same performance. Figure 3.2 shows the performance of the MA(4) signal, illustrating the theoretical findings that all the three algorithms offer similar steady state performances, with the IQLMS exhibiting the fastest convergence. This also verifies the benefits of the I-gradient over the $\mathbb{HR}$ gradient.

Figure 3.3 shows the evolution of the real, $\imath-$, $\jmath-$, and κ-component of one of the filter weight vectors for the MA(4) signal. Verifying (3.57) and (3.58), the IQLMS offered faster convergence for all the four data components. In addition, supporting the analysis, the HR-QLMS converged faster than the QLMS in all the three imaginary channels (with the QLMS converging faster in the real channel).

Figure 3.4 compares the prediction performances of IQLMS, HR-QLMS and QLMS for the noncircular 3D Lorenz signal. A step size of $\mu = 2 \times 10^{-4}$ was used and the learning curves were averaged over 100 independent trials. Observe that the IQLMS achieved a slightly lower steady state error compared to QLMS and HR-QLMS, conforming with the

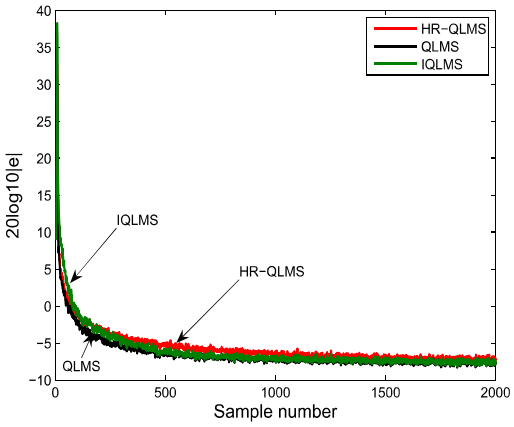


Figure 3.1: Performance comparison of the QLMS, HR-QLMS and IQLMS on the prediction of the circular AR(4) signal, for all the algorithms having the same step size.

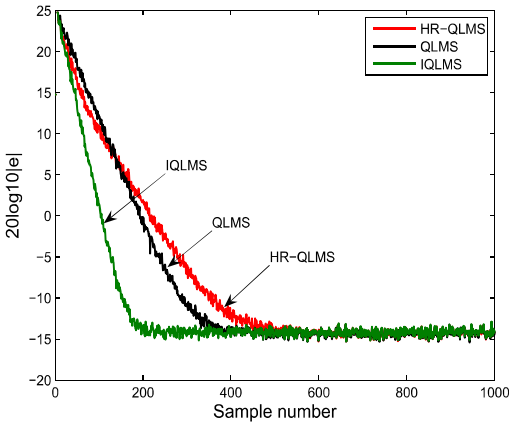


Figure 3.2: Performance comparison of the QLMS, HR-QLMS and IQLMS on the prediction of the circular MA(4) signal, for all the algorithms having the same step size.

analysis in Section 3.6.

Short term wind forecasting plays an important role in renewable energy, pollution modeling and aviation safety [115] [116] [117] [118] [119]. In this scenario, the QLMS, HR-QLMS and IQLMS were used to perform 10-step ahead prediction of a four dimensional wind signal[5]. The wind signal consists of three channels representing the wind speed measurement in the north-south, east-west and vertical direction and a fourth channel for the air temperature (see Figure 3.5), and had a noncircularity degree[6] of $r_s = 0.49$. A step size of 2×10^{-2} was used while the filters had a order of $N = 4$. Figure 3.6 shows the learning curves of the filters considered, averaged over 20 trials. Observe that all three filters achieved the same steady state performance and that the IQLMS converged considerably faster. The performances of the three widely linear filters are shown in Figure 3.7, using the same step size and filter order as for the strictly linear filters. Observe that all three filters achieved the same steady state performance while the WLIQLMS exhibited the fastest convergence rate. Also note that, as expected from the noncircular nature of the wind signal, the widely linear filters achieved a better steady state performance than

[5]The wind data were recorded by Prof. K. Aihara and his team at the University of Tokyo, in an urban environment. The wind was initially sampled at 50 Hz, but resampled at 5 Hz for simulation purposes.

[6]The noncircularity measure $r_s = \frac{E[xx^{i*}]+E[xx^{j*}]+E[xx^{\kappa*}]}{3E[xx^*]}$ where $r_s \in [0, 1]$ and the value $r_s = 0$ indicates a circular source while $r_s = 1$ indicates a highly noncircular source.

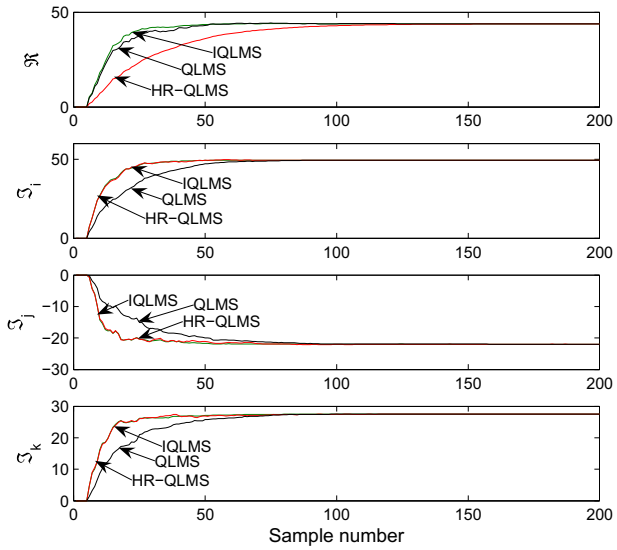


Figure 3.3: Evolution of the adaptive quaternion valued weights of QLMS, HR-QLMS and IQLMS, for a single trial prediction of the MA(4) signal.

the corresponding strictly linear filters.

3.8 Summary

In this chapter we have helped resolve ambiguities present in current quaternion valued adaptive filtering approaches and provided a rigorous platform for the analysis and future developments in this field. We have shown that most inconsistencies in the treatment of such algorithms are due to the noncommutativity of the quaternion product and the limitations of current quaternion gradients. The analysis has introduced two new gradient definitions and has, for generality, been conducted for second order noncircular (improper) signals. The mean and mean square analysis have been performed in a general setting and have been shown to naturally simplify into the corresponding analysis for special cases (e.g. circular/strictly linear). The proposed I-gradient has been shown to facilitate generic forms of quaternion least mean square (QLMS) algorithms, providing natural extensions of LMS, CLMS and ACLMS. The rigorous treatment of quaternion gradients, noncommutativity of products, together with convergence analysis and treatment of noncircularity, has provided a unifying platform for quaternion valued adaptive filtering.

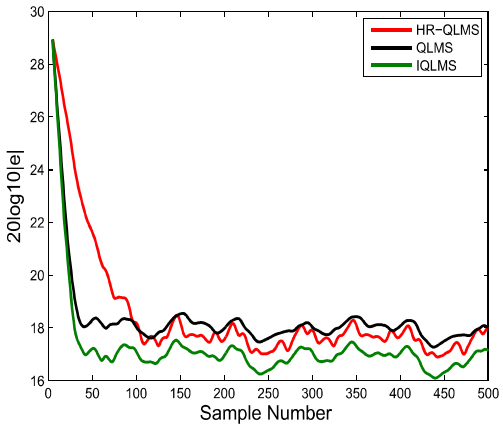


Figure 3.4: Learning curves for the noncircular Lorenz signal, averaged over 100 experiments.

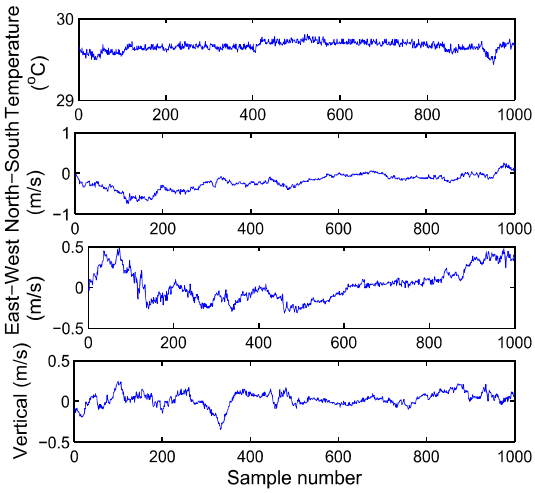


Figure 3.5: Four-dimensional wind signal.

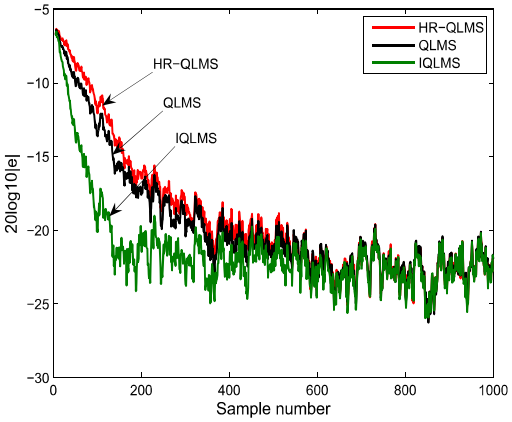


Figure 3.6: Learning curves for the strictly linear algorithms on the prediction of the noncircular wind signal, averaged over 20 experiments.

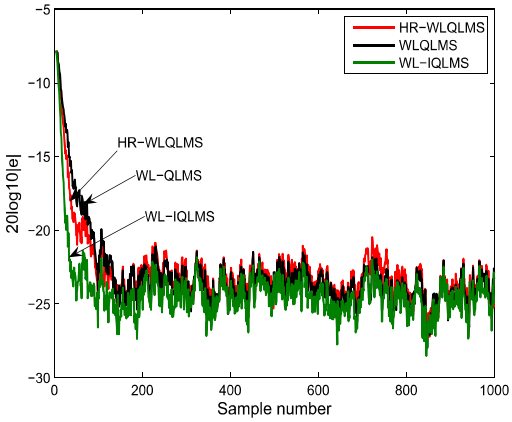


Figure 3.7: Learning curves for the widely linear algorithms on the prediction of the noncircular wind signal, averaged over 20 experiments.

Chapter 4

Quaternion Recursive Least Squares

4.1 Introduction

The recursive least squares (RLS) algorithm is a standard in many real world applications requiring accurate adaptive filters. Compared to the least mean square (LMS) algorithm, it generally exhibits improved convergence and better steady state properties, at the cost of greater computational requirement. It also has a desirable property that the rate of convergence is invariant to the condition number of the correlation matrix $\mathbf{R}_{xx}$ of the input signal [20].

4.2 Derivation of the widely linear QRLS

Within the RLS class of algorithms, the aim is to minimize the following objective function:

$$J(n) = \sum_{k=0}^{n} \lambda^{n-k} \mid e(k) \mid^2 = \sum_{k=0}^{n} \lambda^{n-k} e(k) e^*(k) \tag{4.1}$$

where the forgetting factor $\lambda \in (0, 1]$ and the output error

$$e(k) = d(k) - \mathbf{w}_n^{aT} \mathbf{x}^a(k) = d(k) - \sum_{r=0}^{p} w_n^a(r) x^a(k-r) \tag{4.2}$$

where p is the filter order, and $\mathbf{x}^a(k)$ and $\mathbf{w}_n^a$ denote respectively the augmented input vector $[\mathbf{x}^T(k), \mathbf{x}^{\imath T}(k), \mathbf{x}^{\jmath T}(k), \mathbf{x}^{\kappa T}(k)]$ at time k and the augmented weight vector $[\mathbf{w}_n^T, \mathbf{w}_n^{\imath T}, \mathbf{w}_n^{\jmath T}, \mathbf{w}_n^{\kappa T}]$ at time n.

4.2.1 The least squares solution

Our aim is to find a recursive solution that minimizes the objective function in (4.1). To this end, following $\mathbb{HR}$-calculus in (3.15), we set the partial derivative with respect to $w_n^{a*}(l)$ for $l = 0$ to p, to zero, to give

$$\begin{aligned}\frac{\partial J(n)}{\partial w_n^{a*}(l)} &= \sum_{k=0}^{n} \lambda^{n-k} \frac{\partial e(k) e(k)^*}{\partial w_n^{a*}(l)} \\ &= \sum_{k=0}^{n} \lambda^{n-k} \left(e(k) \frac{\partial e^*(k)}{\partial w_n^{a*}(l)} + \frac{\partial e(k)}{\partial w_n^{a*}(l)} e^*(k) \right)\end{aligned}$$

which yields,

$$\frac{\partial J(n)}{\partial w_n^{a*}(l)} = \sum_{k=0}^{n} \lambda^{n-k} \frac{1}{2} x^a(k-l) e^*(k) - \sum_{k=0}^{n} \lambda^{n-k} e(k) x^{a*}(k-l)$$

Substitute for $e(k)$ in (4.2) and set the gradient to zero to obtain[1]:

$$\begin{aligned}\frac{\partial J(n)}{\partial w_n^{a*}(l)} = \frac{1}{2} \sum_{k=0}^{n} \lambda^{n-k} x^a(k-l) \left(d^*(k) - \sum_{r=0}^{p} x^{a*}(k-r) w_n^{a*}(r) \right) \\ - \sum_{k=0}^{n} \lambda^{n-k} \left(d(k) - \sum_{r=0}^{p} w_n^a(r) x^a(k-r) \right) x^{a*}(k-l) = 0\end{aligned}$$

The terms above can be re-arranged as

$$\begin{aligned}\frac{\partial J(n)}{\partial w_n^{a*}(l)} &= \sum_{k=0}^{n} \sum_{r=0}^{p} \lambda^{n-k} w_n^a(r) x^a(k-r) x^{a*}(k-l) \\ &+ \frac{1}{2} \sum_{k=0}^{n} \lambda^{n-k} x^a(k-l) d^*(k) - \sum_{k=0}^{n} \lambda^{n-k} d(k) x^{a*}(k-l) \\ &- \frac{1}{2} \sum_{k=0}^{n} \sum_{r=0}^{p} \lambda^{n-k} x^a(k-l) x^{a*}(k-r) w_n^{a*}(r) = 0\end{aligned}$$

[1]The quaternion conjugate property $(x_1 x_2)^* = x_2^* x_1^*$ was used.

In a more compact form, we have

$$\mathbf{w}_n^{aT}\mathbf{R}_{xx}^a(n) - \left(\frac{1}{2}\mathbf{R}_{xx}^a\mathbf{w}_n^{a*}\right)^T = \mathbf{r}_{dx}^{aT}(n) - \frac{1}{2}\mathbf{r}_{xd}^{aT}(n) \tag{4.3}$$

where

$$\begin{aligned}\mathbf{R}_{xx}^a(n) &= \sum_{k=0}^{n}\lambda^{n-k}\mathbf{x}^a(k)\mathbf{x}^{aH}(k)\\ \mathbf{r}_{xd}^a(n) &= \sum_{k=0}^{n}\lambda^{n-k}\mathbf{x}^a(k)d^*(k)\\ \mathbf{r}_{dx}^a(n) &= \sum_{k=0}^{n}\lambda^{n-k}d(k)\mathbf{x}^{a*}(k)\end{aligned} \tag{4.4}$$

Observe that $(\mathbf{w}_n^{aT}\mathbf{R}_{xx}^a(n))^* = \left((\mathbf{w}_n^{aT}\mathbf{R}_{xx}^a(n))^H\right)^T = \left(\mathbf{R}_{xx}^{aH}(n)\mathbf{w}_n^{a*}\right)^T = (\mathbf{R}_{xx}^a(n)\mathbf{w}_n^{a*})^T$ and $(\mathbf{r}_{dx}^{aT}(n))^* = \mathbf{r}_{xd}^{aT}(n)$. It therefore follows from (4.3) that

$$\mathbf{w}_n^{aT}\mathbf{R}_{xx}^a(n) = \mathbf{r}_{dx}^{aT}(n)$$

Post-multiplying both sides by $(\mathbf{R}_{xx}^a(n))^{-1}$ we finally obtain

$$\mathbf{w}_n^{aT} = \mathbf{r}_{dx}^{aT}(n)(\mathbf{R}_{xx}^a(n))^{-1} \tag{4.5}$$

The Wiener filter solution in the quaternion domain can be obtained from (4.5), as the recursive estimate of the correlation matrix $\mathbf{R}_{xx}^a(n)$ converges to a true correlation matrix $\mathbf{R}_{xx}^a$ for $n \to \infty$. The same applies to the estimated cross-correlation $\mathbf{r}_{dx}^a$, giving

$$\begin{aligned}(\mathbf{R}_{xx}^a(n))^{-1} &= n^{-1}(\mathbf{R}_{xx}^a)^{-1}\\ \mathbf{r}_{dx}^{aT}(n) &= n\mathbf{r}_{dx}^{aT}\end{aligned}$$

and the Wiener filter solution

$$\mathbf{w}^{aT} = \mathbf{r}_{dx}^{aT}(\mathbf{R}_{xx}^a)^{-1} \tag{4.6}$$

The difference from the complex domain Wiener solution arises due to the non-

commutativity of quaternion products, as the position of $\mathbf{r}_{dx}^{a}$ relative to $(\mathbf{R}_{xx}^{a}(n))^{-1}$ is important and hence the term $\mathbf{w}^{aT}$.

We shall next derive the WLQRLS algorithm, a recursive solution to the least squares solution in (4.5). Our approach follows the general idea of the derivation of the complex widely-linear RLS algorithm[2] in [39], and extends this approach by accounting for the specific properties of quaternion algebra.

4.2.2 The widely-linear QRLS algorithm

The term $\mathbf{x}^{a}(k)\mathbf{x}(k)^{aH}$ within the covariance matrix $\mathbf{R}_{xx}^{a}(n)$ in (4.4) can be expressed as

$$\mathbf{R}_{xx}^{a}(n) = \sum_{k=1}^{n} \lambda^{n-k} \begin{bmatrix} \mathbf{x}(k)\mathbf{x}^{H}(k) & \mathbf{x}(k)\mathbf{x}^{\imath H}(k) & \mathbf{x}(k)\mathbf{x}^{\jmath H}(k) & \mathbf{x}(k)\mathbf{x}^{\kappa H}(k) \\ \mathbf{x}^{\imath}(k)\mathbf{x}^{H}(k) & \mathbf{x}^{\imath}(k)\mathbf{x}^{\imath H}(k) & \mathbf{x}^{\imath}(k)\mathbf{x}^{\jmath H}(k) & \mathbf{x}^{\imath}(k)\mathbf{x}^{\kappa H}(k) \\ \mathbf{x}^{\jmath}(k)\mathbf{x}^{H}(k) & \mathbf{x}^{\jmath}(k)\mathbf{x}^{\imath H}(k) & \mathbf{x}^{\jmath}(k)\mathbf{x}^{\jmath H}(k) & \mathbf{x}^{\jmath}(k)\mathbf{x}^{\kappa H}(k) \\ \mathbf{x}^{\kappa}(k)\mathbf{x}^{H}(k) & \mathbf{x}^{\kappa}(k)\mathbf{x}^{\imath H}(k) & \mathbf{x}^{\kappa}(k)\mathbf{x}^{\jmath H}(k) & \mathbf{x}^{\kappa}(k)\mathbf{x}^{\kappa H}(k) \end{bmatrix} \tag{4.7}$$

yielding a simpler form

$$\mathbf{R}_{xx}^{a}(n) = \sum_{k=1}^{n} \lambda^{n-k} \mathbf{x}^{a}(k)\mathbf{x}(k)^{aH} = \begin{bmatrix} \mathbf{R}_{xx}(n) & \mathbf{P}_{xx}^{\imath}(n) & \mathbf{S}_{xx}^{\jmath}(n) & \mathbf{T}_{xx}^{\kappa}(n) \\ \mathbf{P}_{xx}(n) & \mathbf{R}_{xx}^{\imath}(n) & \mathbf{T}_{xx}^{\jmath}(n) & \mathbf{S}_{xx}^{\kappa}(n) \\ \mathbf{S}_{xx}(n) & \mathbf{T}_{xx}^{\imath}(n) & \mathbf{R}_{xx}^{\jmath}(n) & \mathbf{P}_{xx}^{\kappa}(n) \\ \mathbf{T}_{xx}(n) & \mathbf{S}_{xx}^{\imath}(n) & \mathbf{P}_{xx}^{\jmath}(n) & \mathbf{R}_{xx}^{\kappa}(n) \end{bmatrix} \tag{4.8}$$

where

$$\begin{aligned} \mathbf{R}_{xx}(n) &= \sum_{k=1}^{n} \lambda^{n-k} \mathbf{x}(k)\mathbf{x}^{H}(k) \\ \mathbf{P}_{xx}(n) &= \sum_{k=1}^{n} \lambda^{n-k} \mathbf{x}^{\imath}(k)\mathbf{x}^{H}(k) \\ \mathbf{S}_{xx}(n) &= \sum_{k=1}^{n} \lambda^{n-k} \mathbf{x}^{\jmath}(k)\mathbf{x}^{H}(k) \\ \mathbf{T}_{xx}(n) &= \sum_{k=1}^{n} \lambda^{n-k} \mathbf{x}^{\kappa}(k)\mathbf{x}^{H}(k) \end{aligned} \tag{4.9}$$

[2]The complex WLRLS algorithm, proposed by C. Douglas, takes advantage of a special structure inherent to the augmented covariance matrix to simplify the calculation [39].

To demonstrate the matrix $(\mathbf{R}^a_{xx})^{-1}$ has the same structure as the matrix $\mathbf{R}_{xx}$, observe that when $(\mathbf{R}^a_{xx})^{-1}$ takes the form in (4.8) then $\mathbf{R}^a_{xx}(\mathbf{R}^a_{xx})^{-1} = \mathbf{I}$ is a valid solution, that is

$$\begin{aligned}
\mathbf{R}^a_{xx}(n)(\mathbf{R}^a_{xx}(n))^{-1} &= \begin{bmatrix} \mathbf{R}_{xx}(n) & \mathbf{P}^{\imath}_{xx}(n) & \mathbf{S}^{\jmath}_{xx}(n) & \mathbf{T}^{\kappa}_{xx}(n) \\ \mathbf{P}_{xx}(n) & \mathbf{R}^{\imath}_{xx}(n) & \mathbf{T}^{\jmath}_{xx}(n) & \mathbf{S}^{\kappa}_{xx}(n) \\ \mathbf{S}_{xx}(n) & \mathbf{T}^{\imath}_{xx}(n) & \mathbf{R}^{\jmath}_{xx}(n) & \mathbf{P}^{\kappa}_{xx}(n) \\ \mathbf{T}_{xx}(n) & \mathbf{S}^{\imath}_{xx}(n) & \mathbf{P}^{\jmath}_{xx}(n) & \mathbf{R}^{\kappa}_{xx}(n) \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B}^{\imath} & \mathbf{C}^{\jmath} & \mathbf{D}^{\kappa} \\ \mathbf{B} & \mathbf{A}^{\imath} & \mathbf{D}^{\jmath} & \mathbf{C}^{\kappa} \\ \mathbf{C} & \mathbf{D}^{\imath} & \mathbf{A}^{\jmath} & \mathbf{B}^{\kappa} \\ \mathbf{D} & \mathbf{C}^{\imath} & \mathbf{B}^{\jmath} & \mathbf{A}^{\kappa} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}
\end{aligned}$$

Out of the 16 equations (one for each entry of the identity matrix),

$$\begin{aligned}
\mathbf{R}_{xx}(n)\mathbf{A} + \mathbf{P}^{\imath}_{xx}(n)\mathbf{B} + \mathbf{S}^{\jmath}_{xx}(n)\mathbf{C} + \mathbf{T}^{\kappa}_{xx}(n)\mathbf{D} &= \mathbf{I} \\
\mathbf{P}_{xx}(n)\mathbf{A} + \mathbf{R}^{\imath}_{xx}(n)\mathbf{B} + \mathbf{T}^{\jmath}_{xx}(n)\mathbf{C} + \mathbf{S}^{\kappa}_{xx}(n)\mathbf{D} &= \mathbf{0} \\
\mathbf{S}_{xx}(n)\mathbf{A} + \mathbf{T}^{\imath}_{xx}(n)\mathbf{B} + \mathbf{R}^{\jmath}_{xx}(n)\mathbf{C} + \mathbf{P}^{\kappa}_{xx}(n)\mathbf{D} &= \mathbf{0} \\
\mathbf{T}_{xx}(n)\mathbf{A} + \mathbf{S}^{\imath}_{xx}(n)\mathbf{B} + \mathbf{P}^{\jmath}_{xx}(n)\mathbf{C} + \mathbf{R}^{\kappa}_{xx}(n)\mathbf{D} &= \mathbf{0}
\end{aligned}$$

only four are linearly independent, as all others are involutions of these equations. This set of linear equations has four unknowns $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$. As there is only one solution for $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$, hence the structure assumed for $(\mathbf{R}^a_{xx})^{-1}$ must be valid. A direct consequence of this structure is that the matrix $(\mathbf{R}^a_{xx})^{-1}$ can be fully described by only a quarter of its entries.

The first step towards the WLQRLS algorithm is to find a recursive solution for $(\mathbf{R}^a_{xx})^{-1}$ and $\mathbf{r}^a_{dx}$. Using Woodbury's identity, the matrix inverse $(\mathbf{R}^a_{xx})^{-1}$ can be written

as:

$$
\begin{aligned}
(\mathbf{R}^a_{xx}(n))^{-1} &= \lambda^{-1}(\mathbf{R}^a_{xx}(n-1))^{-1} \\
&\quad -\frac{\lambda^{-2}(\mathbf{R}^a_{xx}(n-1))^{-1}\mathbf{x}^a(n)\mathbf{x}^{aH}(n)(\mathbf{R}^a_{xx}(n-1))^{-1}}{1+\lambda^{-1}\mathbf{x}^{aH}(n)(\mathbf{R}^a_{xx}(n-1))^{-1}\mathbf{x}^a(n)}
\end{aligned}
\tag{4.10}
$$

A recursive solution for $\mathbf{A}(n)$, $\mathbf{B}(n)$, $\mathbf{C}(n)$ and $\mathbf{D}(n)$ can now be obtained by expanding the expression for $\mathbf{R}^{-1}_{xx}(n)$ into its components.

$$
\begin{bmatrix}
\mathbf{A}(n)\,\mathbf{B}^i(n)\,\mathbf{C}^j(n)\,\mathbf{D}^k(n)\\
\mathbf{B}(n)\,\mathbf{A}^i(n)\,\mathbf{D}^j(n)\,\mathbf{C}^k(n)\\
\mathbf{C}(n)\,\mathbf{D}^i(n)\,\mathbf{A}^j(n)\,\mathbf{B}^k(n)\\
\mathbf{D}(n)\,\mathbf{C}^i(n)\,\mathbf{B}^j(n)\,\mathbf{A}^k(n)
\end{bmatrix}
=
\lambda^{-1}
\begin{bmatrix}
\mathbf{A}(n-1)\,\mathbf{B}^i(n-1)\,\mathbf{C}^j(n-1)\,\mathbf{D}^k(n-1)\\
\mathbf{B}(n-1)\,\mathbf{A}^i(n-1)\,\mathbf{D}^j(n-1)\,\mathbf{C}^k(n-1)\\
\mathbf{C}(n-1)\,\mathbf{D}^i(n-1)\,\mathbf{A}^j(n-1)\,\mathbf{B}^k(n-1)\\
\mathbf{D}(n-1)\,\mathbf{C}^i(n-1)\,\mathbf{B}^j(n-1)\,\mathbf{A}^k(n-1)
\end{bmatrix}
$$

$$
-\lambda^{-2}\frac{
\begin{bmatrix}
\mathbf{A}(n-1)\,\mathbf{B}^i(n-1)\,\mathbf{C}^j(n-1)\,\mathbf{D}^k(n-1)\\
\mathbf{B}(n-1)\,\mathbf{A}^i(n-1)\,\mathbf{D}^j(n-1)\,\mathbf{C}^k(n-1)\\
\mathbf{C}(n-1)\,\mathbf{D}^i(n-1)\,\mathbf{A}^j(n-1)\,\mathbf{B}^k(n-1)\\
\mathbf{D}(n-1)\,\mathbf{C}^i(n-1)\,\mathbf{B}^j(n-1)\,\mathbf{A}^k(n-1)
\end{bmatrix}
\begin{bmatrix}\mathbf{x}(n)\\ \mathbf{x}^i(n)\\ \mathbf{x}^j(n)\\ \mathbf{x}^k(n)\end{bmatrix}^H
\begin{bmatrix}
\mathbf{A}(n-1)\,\mathbf{B}^i(n-1)\,\mathbf{C}^j(n-1)\,\mathbf{D}^k(n-1)\\
\mathbf{B}(n-1)\,\mathbf{A}^i(n-1)\,\mathbf{D}^j(n-1)\,\mathbf{C}^k(n-1)\\
\mathbf{C}(n-1)\,\mathbf{D}^i(n-1)\,\mathbf{A}^j(n-1)\,\mathbf{B}^k(n-1)\\
\mathbf{D}(n-1)\,\mathbf{C}^i(n-1)\,\mathbf{B}^j(n-1)\,\mathbf{A}^k(n-1)
\end{bmatrix}
\begin{bmatrix}\mathbf{x}(n)\\ \mathbf{x}^i(n)\\ \mathbf{x}^j(n)\\ \mathbf{x}^k(n)\end{bmatrix}
}{
1+\lambda^{-1}
\begin{bmatrix}\mathbf{x}(n)\\ \mathbf{x}^i(n)\\ \mathbf{x}^j(n)\\ \mathbf{x}^k(n)\end{bmatrix}^H
\begin{bmatrix}
\mathbf{A}(n-1)\,\mathbf{B}^i(n-1)\,\mathbf{C}^j(n-1)\,\mathbf{D}^k(n-1)\\
\mathbf{B}(n-1)\,\mathbf{A}^i(n-1)\,\mathbf{D}^j(n-1)\,\mathbf{C}^k(n-1)\\
\mathbf{C}(n-1)\,\mathbf{D}^i(n-1)\,\mathbf{A}^j(n-1)\,\mathbf{B}^k(n-1)\\
\mathbf{D}(n-1)\,\mathbf{C}^i(n-1)\,\mathbf{B}^j(n-1)\,\mathbf{A}^k(n-1)
\end{bmatrix}
\begin{bmatrix}\mathbf{x}(n)\\ \mathbf{x}^i(n)\\ \mathbf{x}^j(n)\\ \mathbf{x}^k(n)\end{bmatrix}
}
\tag{4.11}
$$

Note from (4.11) that $\mathbf{A}(n)$, $\mathbf{B}(n)$, $\mathbf{C}(n)$ and $\mathbf{D}(n)$ can be written as

$$\begin{aligned}
\mathbf{A}(n) &= \lambda^{-1}\mathbf{A}(n-1) - \lambda^{-2}\frac{\mathbf{p}\mathbf{p}^H}{c} \\
\mathbf{B}(n) &= \lambda^{-1}\mathbf{B}(n-1) - \lambda^{-2}\frac{\mathbf{p}^{\imath}\mathbf{p}^H}{c} \\
\mathbf{C}(n) &= \lambda^{-1}\mathbf{C}(n-1) - \lambda^{-2}\frac{\mathbf{p}^{\jmath}\mathbf{p}^H}{c} \\
\mathbf{D}(n) &= \lambda^{-1}\mathbf{D}(n-1) - \lambda^{-2}\frac{\mathbf{p}^{\kappa}\mathbf{p}^H}{c}
\end{aligned} \tag{4.12}$$

where $\mathbf{p}$ and c are defined as

$$\begin{aligned}
\mathbf{p} &= [\mathbf{A}(n-1)\mathbf{x}(n) + \mathbf{B}^{\imath}(n-1)\mathbf{x}^{\imath}(n) + \mathbf{C}^{\jmath}(n-1)\mathbf{x}^{\jmath}(n) \\
&\quad + \mathbf{D}^{\kappa}(n-1)\mathbf{x}^{\kappa}(n)] \\
c &= 1 + \lambda^{-1} 4\Re\left(\mathbf{x}^H(n)\mathbf{p}\right)
\end{aligned}$$

where symbol $\Re(\cdot)$ denotes the real part of a quaternion. The cross-correlation $\mathbf{r}_{dx}^a$ can be recursively estimated by

$$\mathbf{r}_{dx}^a(n) = \lambda \mathbf{r}_{dx}^a(n-1) + d(n)\mathbf{x}^{a*}(n) \tag{4.13}$$

The augmented weight update of the WLQRLS algorithm, $\mathbf{w}_n^T = \mathbf{r}_{dx}^{aT}(\mathbf{R}_{xx}^a)^{-1}$ can be expanded as:

$$\begin{bmatrix} \mathbf{u}_n \\ \mathbf{v}_n \\ \mathbf{g}_n \\ \mathbf{h}_n \end{bmatrix}^T = \begin{bmatrix} \mathbf{r}_{dx_1}(n) \\ \mathbf{r}_{dx_2}(n) \\ \mathbf{r}_{dx_3}(n) \\ \mathbf{r}_{dx_4}(n) \end{bmatrix}^T \begin{bmatrix} \mathbf{A}(n) & \mathbf{B}^{\imath}(n) & \mathbf{C}^{\jmath}(n) & \mathbf{D}^{\kappa}(n) \\ \mathbf{B}(n) & \mathbf{A}^{\imath}(n) & \mathbf{D}^{\jmath}(n) & \mathbf{C}^{\kappa}(n) \\ \mathbf{C}(n) & \mathbf{D}^{\imath}(n) & \mathbf{A}^{\jmath}(n) & \mathbf{B}^{\kappa}(n) \\ \mathbf{D}(n) & \mathbf{C}^{\imath}(n) & \mathbf{B}^{\jmath}(n) & \mathbf{A}^{\kappa}(n) \end{bmatrix}$$

We can now write expressions for $\mathbf{u}_n$, $\mathbf{v}_n$, $\mathbf{g}_n$ and $\mathbf{h}_n$ in terms of $\mathbf{A}(n)$, $\mathbf{B}(n)$, $\mathbf{C}(n)$ and

$$
\begin{aligned}
\mathbf{u}_n^T &= \mathbf{r}_{dx_1}^T(n-1)\mathbf{A}(n-1) + \mathbf{r}_{dx_2}^T(n-1)\mathbf{B}(n-1) + \mathbf{r}_{dx_3}^T(n-1)\mathbf{C}(n-1) + \mathbf{r}_{dx_4}^T(n-1)\mathbf{A}(n-1) \\
&+ \lambda^{-1}d(n)\mathbf{x}^H(n)\mathbf{A}(n-1) + \lambda^{-1}d(n)\mathbf{x}^{\imath H}(n)\mathbf{B}(n-1) + \lambda^{-1}d(n)\mathbf{x}^{\jmath H}(n)\mathbf{C}(n-1) + \lambda^{-1}d(n)\mathbf{x}^{\kappa H}(n)\mathbf{D}(n-1) \\
&- \lambda^{-1}\mathbf{r}_{dx_1}^T(n-1)\frac{\mathbf{p}\mathbf{p}^H}{c} - \lambda^{-1}\mathbf{r}_{dx_2}^T(n-1)\frac{\mathbf{p}\mathbf{p}^{\imath H}}{c} - \lambda^{-1}\mathbf{r}_{dx_3}^T(n-1)\frac{\mathbf{p}\mathbf{p}^{\jmath H}}{c} - \lambda^{-1}\mathbf{r}_{dx_4}^T(n-1)\frac{\mathbf{p}\mathbf{p}^{\kappa H}}{c} \\
&- \lambda^{-2}d(n)\mathbf{x}^H(n)\frac{\mathbf{p}\mathbf{p}^H}{c} - \lambda^{-2}d(n)\mathbf{x}^{\imath H}(n)\frac{\mathbf{p}^{\imath}\mathbf{p}^H}{c} - \lambda^{-2}d(n)\mathbf{x}^{\jmath H}(n)\frac{\mathbf{p}^{\jmath}\mathbf{p}^H}{c} - \lambda^{-2}d(n)\mathbf{x}^{\kappa H}(n)\frac{\mathbf{p}^{\kappa}\mathbf{p}^H}{c}
\end{aligned} \tag{4.15}
$$

$\mathbf{D}(n)$

$$
\begin{aligned}
\mathbf{u}_n^T &= \mathbf{r}_{dx_1}^T(n)\mathbf{A}(n) + \mathbf{r}_{dx_2}^T(n)\mathbf{B}(n) + \mathbf{r}_{dx_3}^T(n)\mathbf{C}(n) + \mathbf{r}_{dx_4}^T(n)\mathbf{D}(n) \\
\mathbf{v}_n^T &= \mathbf{r}_{dx_1}^T(n)\mathbf{B}^{\imath}(n) + \mathbf{r}_{dx_2}^T(n)\mathbf{A}^{\imath}(n) + \mathbf{r}_{dx_3}^T(n)\mathbf{D}^{\imath}(n) + \mathbf{r}_{dx_4}^T(n)\mathbf{C}^{\imath}(n) \\
\mathbf{g}_n^T &= \mathbf{r}_{dx_1}^T(n)\mathbf{C}^{\jmath}(n) + \mathbf{r}_{dx_2}^T(n)\mathbf{D}^{\jmath}(n) + \mathbf{r}_{dx_3}^T(n)\mathbf{A}^{\jmath}(n) + \mathbf{r}_{dx_4}^T(n)\mathbf{B}^{\jmath}(n) \\
\mathbf{h}_n^T &= \mathbf{r}_{dx_1}^T(n)\mathbf{D}^{\kappa}(n) + \mathbf{r}_{dx_2}^T(n)\mathbf{C}^{\kappa}(n) + \mathbf{r}_{dx_3}^T(n)\mathbf{B}^{\kappa}(n) + \mathbf{r}_{dx_4}^T(n)\mathbf{A}^{\kappa}(n)
\end{aligned} \tag{4.14}
$$

By substituting (4.12) and (4.13) into (4.14), we can obtain a recursive time update for the weight vectors $\mathbf{u}_n$, $\mathbf{v}_n$, $\mathbf{g}_n$ and $\mathbf{h}_n$. For illustration, the update of the weight component $\mathbf{u}_n$ is given below. The update for the other three weight vectors are obtained analogously.

$$
\begin{aligned}
\mathbf{u}_n^T = &\left(\lambda\mathbf{r}_{dx_1}^T(n-1) + d(n)\mathbf{x}^H(n)\right)\left(\lambda^{-1}\mathbf{A}(n-1) - \lambda^{-2}\frac{\mathbf{p}\mathbf{p}^H}{c}\right) \\
&+ \left(\lambda\mathbf{r}_{dx_2}^T(n-1) + d(n)\mathbf{x}^{\imath H}(n)\right)\left(\lambda^{-1}\mathbf{B}(n-1) - \lambda^{-2}\frac{\mathbf{p}^{\imath}\mathbf{p}^H}{c}\right) \\
&+ \left(\lambda\mathbf{r}_{dx_3}^T(n-1) + d(n)\mathbf{x}^{\jmath H}(n)\right)\left(\lambda^{-1}\mathbf{C}(n-1) - \lambda^{-2}\frac{\mathbf{p}^{\jmath}\mathbf{p}^H}{c}\right) \\
&+ \left(\lambda\mathbf{r}_{dx_4}^T(n-1) + d(n)\mathbf{x}^{\kappa H}(n)\right)\left(\lambda^{-1}\mathbf{D}(n-1) - \lambda^{-2}\frac{\mathbf{p}^{\kappa}\mathbf{p}^H}{c}\right)
\end{aligned}
$$

Expanding $\mathbf{u}_n^T$ we can split the terms above into four groups, as shown in (4.15). It can be shown that the first row equals $\mathbf{u}_{n-1}^T$ and the second row equals $\lambda^{-1}d(n)\mathbf{p}^H(n)$. The third row can be written as

$$
\lambda^{-1}\left(\mathbf{u}_{n-1}^T\mathbf{x}(n) + \mathbf{v}_{n-1}^T\mathbf{x}^{\imath}(n) + \mathbf{g}_{n-1}^T\mathbf{x}^{\jmath}(n) + \mathbf{h}_{n-1}^T\mathbf{x}^{\kappa}(n)\right)\frac{\mathbf{p}^H}{c}
$$

whereas the last row can be written as

$$\lambda^{-2}d(n)\left(4\Re(\mathbf{x}^H\mathbf{p})\right)\frac{\mathbf{p}^H}{c}$$

Thus giving,

$$\begin{aligned}\mathbf{u}_n^T &= \mathbf{u}_{n-1}^T + \lambda^{-1}d(n)\mathbf{p}^H - \lambda^{-2}d(n)(4\Re(\mathbf{x}^H(n)\mathbf{p})\frac{\mathbf{p}^H}{c} \\ &\quad -\lambda^{-1}\left(\mathbf{u}_{n-1}^T\mathbf{x}(n) + \mathbf{v}_{n-1}^T\mathbf{x}^\imath(n) + \mathbf{g}_{n-1}^T\mathbf{x}^\jmath(n) + \mathbf{h}_{n-1}^T\mathbf{x}^\kappa(n)\right)\frac{\mathbf{p}^H}{c} \\ &= \mathbf{u}_{n-1}^T + \lambda^{-1}d(n)\mathbf{p}^H - \lambda^{-2}d(n)(4\Re(\mathbf{x}^H(n)\mathbf{p})\frac{\mathbf{p}^H}{c} - \lambda^{-1}\left(\mathbf{w}_{n-1}^{aT}\mathbf{x}^a(n)\right)\frac{\mathbf{p}^H}{c}\end{aligned}$$

Using $\lambda^{-1}4\Re(\mathbf{x}^H(n)\mathbf{p}) = c - 1$, we finally obtain

$$\mathbf{u}_n^T = \mathbf{u}_{n-1}^T + \lambda^{-1}\left(d(n) - \left(\mathbf{w}_{n-1}^{aT}\mathbf{x}^a(n)\right)\right)\frac{\mathbf{p}^H}{c}$$

The update for $\mathbf{v}_n^T$, $\mathbf{g}_n^T$ and $\mathbf{h}_n^T$ follows a similar form, summarized in (4.22).

4.2.3 Summary of the widely linear QRLS algorithm

The WLQRLS algorithm is summarized below

$$\mathbf{x}^a(n) = [\mathbf{x}^T(n), \mathbf{x}^{\imath T}(n), \mathbf{x}^{\jmath T}(n), \mathbf{x}^{\kappa T}(n)]^T \tag{4.16}$$

$$\mathbf{w}_n^a = [\mathbf{u}_n^T, \mathbf{v}_n^T, \mathbf{g}_n^T, \mathbf{h}_n^T]^T \tag{4.17}$$

$$e(n) = d(n) - \mathbf{w}^{aT}(n-1)\mathbf{x}^a \tag{4.18}$$

$$c = 1 + 4\Re\left(\mathbf{x}^H(n)\mathbf{p}\right) \tag{4.19}$$

$$\mathbf{p} = [\mathbf{A}(n-1)\mathbf{x}(n) + \mathbf{B}^\imath(n-1)\mathbf{x}^\imath(n) + \mathbf{C}^\jmath(n-1)\mathbf{x}^\jmath(n) + \mathbf{D}^\kappa(n-1)\mathbf{x}^\kappa(n)] \tag{4.20}$$

$$\begin{aligned}
\mathbf{A}(n) &= \lambda^{-1}\mathbf{A}(n-1) - \lambda^{-2}\frac{\mathbf{p}\mathbf{p}^H}{c} \\
\mathbf{B}(n) &= \lambda^{-1}\mathbf{B}(n-1) - \lambda^{-2}\frac{\mathbf{p}^{\imath}\mathbf{p}^H}{c} \\
\mathbf{C}(n) &= \lambda^{-1}\mathbf{C}(n-1) - \lambda^{-2}\frac{\mathbf{p}^{\jmath}\mathbf{p}^H}{c} \\
\mathbf{D}(n) &= \lambda^{-1}\mathbf{D}(n-1) - \lambda^{-2}\frac{\mathbf{p}^{\kappa}\mathbf{p}^H}{c}
\end{aligned} \tag{4.21}$$

The updates of the augmented weight vector $\mathbf{w}_n^a = [\mathbf{u}_n^T\ \mathbf{v}_n^T\ \mathbf{g}_n^T\ \mathbf{h}_n^T]^T$ of the WLQRLS are given by

$$\begin{aligned}
\mathbf{u}_n &= \mathbf{u}_{n-1} + \lambda^{-1}\frac{e(n)\mathbf{p}^H}{c} \\
\mathbf{v}_n &= \mathbf{v}_{n-1} + \lambda^{-1}\frac{e(n)\mathbf{p}^{\imath H}}{c} \\
\mathbf{g}_n &= \mathbf{g}_{n-1} + \lambda^{-1}\frac{e(n)\mathbf{p}^{\jmath H}}{c} \\
\mathbf{h}_n &= \mathbf{h}_{n-1} + \lambda^{-1}\frac{e(n)\mathbf{p}^{\kappa H}}{c}
\end{aligned} \tag{4.22}$$

4.3 Simulations

The effectiveness of the WLQRLS is demonstrated by comparing its performance to that of WLQLMS algorithm in forecasting wind[3]. For the simulations, four signals (temperature reading and wind speed in the North, East and vertical directions[4]) were combined into one quaternion valued signal.

For comparison purposes, the step size of the widely linear QLMS algorithm [67] was chosen to give the fastest convergence for the signal at hand, whereas for the widely linear QRLS, a forgetting factor $\lambda = 0.95$ was employed. The quantitative performance

[3]The quaternion toolbox was used to extend the capabilities of Matlab to the quaternion domain [120].

[4]The wind data were recorded by Prof. K. Aihara and his team at the University of Tokyo, in an urban environment, sampled at 5Hz. We considered the three wind speed components as a pure quaternion, with the air temperature being the real part.

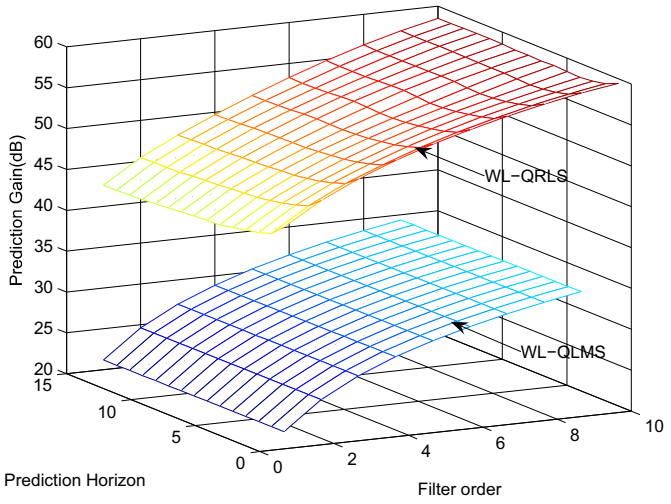


Figure 4.1: Performance of WLQLMS and WLQRLS on the prediction of 4D wind field (3D wind speed and air temperature).

measure was the prediction gain, given by

$$R_p = 10\log\frac{\sigma_x^2}{\sigma_e^2} \quad [dB] \tag{4.23}$$

where σ_x^2 and σ_e^2 denote respectively the input signal power and error power.

Figure 4.1 compares the prediction gain of WLQLMS and WLQRLS for the prediction of a wind signal over a range of prediction horizons. Over all filter orders and prediction horizons, the performance of the WLQRLS was approximately 20dB higher than that of the WLQLMS. Figure 4.2 shows the error curves for both the algorithms, illustrating the faster convergence rate of WLQRLS. Figure 4.3 compares the prediction gain when both the filters are at the steady state, illustrating the enhanced performance of the WLQRLS. Finally, Figure 4.4 visualizes the tracking performance of the WLQRLS on the vertical wind speed component of the quaternion wind signal.

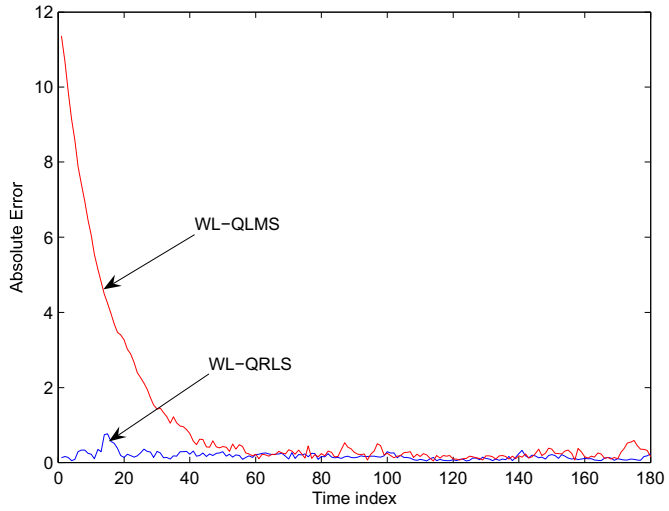


Figure 4.2: Learning curves of WLQLMS and WLQRLS on the prediction of 4D wind field (3D wind speed and air temperature).

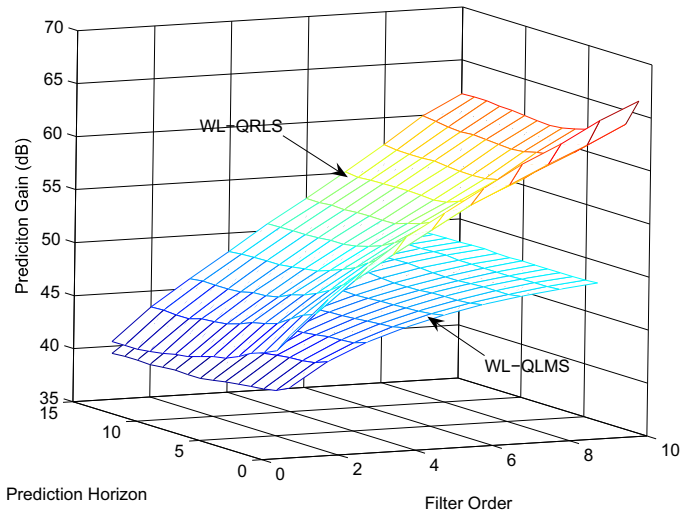


Figure 4.3: Steady state performance of WLQLMS and WLQRLS on the prediction of 4D wind field (3D wind speed and air temperature).

4.4 Summary

We have introduced the widely linear QRLS (WLQRLS) algorithm for the processing of circular and noncircular signals in the quaternion domain. This has been achieved by using the recently introduced quaternion widely linear model and $\mathbb{HR}$-calculus, and following the

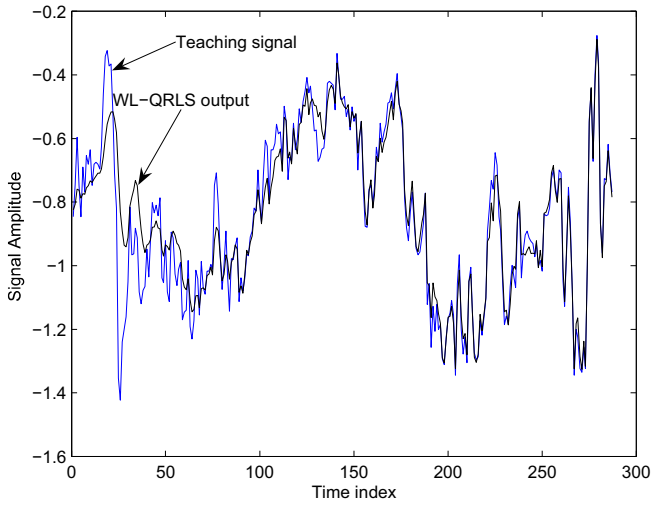


Figure 4.4: Prediction performance of WLQRLS filter for 5-step ahead prediction of the vertical wind speed component.

RLS algorithm's steps. The superiority of the WLQRLS has been demonstrated on the forecasting of a 4D wind field (3D wind as a pure quaternion and air temperature as a real part). The WLQRLS algorithm has been found to have an increased convergence and superior steady state performance, as compared to other available widely linear adaptive algorithms in $\mathbb{H}$, the WLQLMS.

Chapter 5

A Class of Quaternion Valued Affine Projection Algorithms

5.1 Introduction

The quaternion least mean square (QLMS) [66] was recently introduced for adaptive filtering of quaternion valued data, however, it also highlighted the need for faster converging practical algorithms. The stochastic gradient based normalized QLMS can solve this issue only partially whereas the fast converging quaternion recursive least squares (QRLS) [121] is computationally demanding. To that end, our aim is to introduce a fast converging and computationally scalable affine projection scheme for adaptive filtering of quaternion valued data.

Ozeki and Umeda [18] employed affine subspace projections to develop the affine projection algorithm (APA) for real valued finite impulse response (FIR) adaptive filters, thus ensuring fast convergence for coloured inputs. Structurally, the APA spans the range between the normalized least mean square (NLMS) and recursive least squares (RLS), both in terms of performance and computational requirements. In practical terms, whereas the NLMS updates the weight vector based only on the current input vector, the APA employs N past input vectors, thus making use of the history of weight update evolutions, and achieving faster convergence than the NLMS. For large N, the performance of APA

compares with RLS, which updates the weight vectors based on the cumulative error powers calculated from all the past input vectors, therefore exhibiting a high computational complexity. These algorithms were developed for a single channel (univariate) scenario, and were only recently extended to the widely linear complex scenarios [41], thus catering for the generality of complex signals, both circular and noncircular.

The development of real time adaptive filtering algorithms in the quaternion domain has long been hampered by the lack of a rigorous definition of quaternion gradient. This is not surprising, as the gradient calculation is critical in division algebras (complex, quaternion), where standard differentiability conditions (e.g. Cauchy Riemann) do not allow for the gradient of real function of complex and quaternion variables to be calculated; yet typical cost functions are real valued error power. For instance, it is only the $\mathbb{CR}$-calculus [28] [98], that made it possible for the real APA to be extended to the complex domain [41], and for the augmented complex APA (AAPA) to be derived for the processing of noncircular complex signals. Similarly, until the recent development of the $\mathbb{HR}$-calculus [122] and the augmented quaternion statistics [64] [65] [67], stochastic gradient algorithms in the quaternion domain lacked a formal treatment of the gradient, while statistics were only suited for the processing of second order circular signals[1].

Unlike the standard, strictly linear, quaternion valued algorithms where the covariance matrix $E[\mathbf{q}\mathbf{q}^H]$ is assumed to be sufficient to model second order quaternion statistics, widely linear quaternion statistics use the additional pseudocovariance matrices $E[\mathbf{q}\mathbf{q}^{\imath H}]$, $E[\mathbf{q}\mathbf{q}^{\jmath H}]$ and $E[\mathbf{q}\mathbf{q}^{\kappa H}]$ to fully describe second order information in $\mathbb{H}$. This offers more degrees of freedom in the modeling and the possibility to perform optimal second order filtering of noncircular (improper) signals, for which the powers in the components of the quaternion signal are different, a typical case in real world scenarios. Building upon those advantages, the augmented statistics have recently been used to derive the widely linear quaternion recursive least squares (WLQRLS) [121] and the widely linear quaternion least mean square (WLQLMS) [67].

In this chapter, we introduce a general class of quaternion APA algorithms com-

[1]Second order circular signals (proper) have rotation invariant distributions and equal powers in all the signal components.

prising both the strictly linear (QAPA) and widely linear (WLQAPA) cases, in order to provide a rigorous framework for fast converging adaptive filtering of the generality of quaternion valued signals. Based on the principle of energy conservation [123], expressions for the mean square error (MSE) of both the QAPA and WLQAPA are also derived. Simulations show that the WLQAPA offers lower MSE when processing noncircular data, and is thus second order optimal for noncircular signals. These advantages are illustrated in the context of renewable energy (wind modeling) and human centered computing (inertial bodysensors).

5.2 A class of quaternion affine projection algorithms

Consider the real APA which can be seen as an approximation to the steepest descent weight update [20], given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\big(E[\mathbf{x}(k)d(k)] - E[\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{w}(k)\big) \tag{5.1}$$

where $\mathbf{w}$ is the filter coefficient vector, μ the stepsize, $\mathbf{x}$ the input vector and d the teaching signal. The weight update is a function of the second order statistics of the input vector $\mathbf{x}(k)$ as exemplified by the term $\mathbf{x}\mathbf{x}^T$. To deal with both circular and noncircular signals the quaternion valued APA must therefore employ the augmented statistics based on the augmented covariance matrix.

To provide a unifying approach to the derivation of both the QAPA and WLQAPA, we start by deriving an expression for the WLQAPA and show that for circular signals it simplifies into the QAPA. The aim of WLQAPA class of algorithms is to minimize adaptively the squared Euclidean norm of the change in the augmented weight vector $\mathbf{w}^a \in \mathbb{H}^{4M\times 1}$, that is

$$\begin{aligned} &\text{minimize} && ||\delta\mathbf{w}^a(k+1)||^2 = ||\mathbf{w}^a(k+1) - \mathbf{w}^a(k)||^2 \\ &\text{subject to} && d(k-n) = \mathbf{w}^{aH}(k+1)\mathbf{x}^a(k-n) \end{aligned} \tag{5.2}$$

for $n = 0, \ldots, N-1$, where the augmented weight vector $\mathbf{w}^a = [\mathbf{u}^T, \mathbf{v}^T, \mathbf{g}^T, \mathbf{h}^T]^T \in \mathbb{H}^{4M\times 1}$,

the augmented input vector $\mathbf{x}^a = [\mathbf{x}^T, \mathbf{x}^{iT}, \mathbf{x}^{jT}, \mathbf{x}^{kT}] \in \mathbb{H}^{4M\times 1}$ and $||\cdot||^2$ is the Euclidean norm.

Using the Lagrange multipliers to solve the constrained optimization problem, the cost function to be minimized can be written as

$$\begin{aligned} J(k) &= ||\mathbf{u}(k+1)-\mathbf{u}(k)||^2 + ||\mathbf{v}(k+1)-\mathbf{v}(k)||^2 + ||\mathbf{g}(k+1)-\mathbf{g}(k)||^2 \\ &+ ||\mathbf{h}(k+1)-\mathbf{h}(k)||^2 + \Re\big[\big(\mathbf{d}^T(k) - \mathbf{u}^H(k+1)\mathbf{A}(k) - \mathbf{v}^H(k+1)\mathbf{A}^{\imath}(k) \\ &- \mathbf{g}^H(k+1)\mathbf{A}^{\jmath}(k) - \mathbf{h}^H(k+1)\mathbf{A}^{\kappa}(k)\big)\boldsymbol{\lambda}^*\big] \end{aligned}$$

where the symbol $\Re[\cdot]$ denotes the real part of a quaternion variable and

$$\begin{aligned} \mathbf{A}(k) &= [\mathbf{x}(k), \mathbf{x}(k-1), \ldots, \mathbf{x}(k-N+1)] \\ \mathbf{d}(k) &= [d(k), d(k-1), \ldots, d(-N+1)]^T \\ \boldsymbol{\lambda} &= [\lambda_0, \lambda_1, \ldots, \lambda_{k-N+1}]^T \end{aligned}$$

are past values in the filter memory. Standard quaternion differentiation does not allow for the calculation of the gradient of $J(k)$, however, using the $\mathbb{HR}^*$-gradient in (3.15), the gradient of $J(k)$ with respect to the augmented weight vector $\mathbf{w}^{a*}(k+1)$ can be obtained by employing the vector derivatives

$$\begin{aligned} \frac{\partial J(k)}{\partial \mathbf{u}^*(k+1)} &= \mathbf{u}(k+1) - \mathbf{u}(k) - \frac{1}{2}(\mathbf{u}(k+1)-\mathbf{u}(k))^* + \mathbf{A}(k)\boldsymbol{\lambda}^* - \frac{1}{2}(\mathbf{A}(k)\boldsymbol{\lambda}^*)^* \\ \frac{\partial J(k)}{\partial \mathbf{v}^*(k+1)} &= \mathbf{v}(k+1) - \mathbf{v}(k) - \frac{1}{2}(\mathbf{v}(k+1)-\mathbf{v}(k))^* + \mathbf{A}^{\imath}(k)\boldsymbol{\lambda}^* - \frac{1}{2}(\mathbf{A}^{\imath}(k)\boldsymbol{\lambda}^*)^* \\ \frac{\partial J(k)}{\partial \mathbf{g}^*(k+1)} &= \mathbf{g}(k+1) - \mathbf{g}(k) - \frac{1}{2}(\mathbf{g}(k+1)-\mathbf{g}(k))^* + \mathbf{A}^{\jmath}(k)\boldsymbol{\lambda}^* - \frac{1}{2}(\mathbf{A}^{\jmath}(k)\boldsymbol{\lambda}^*)^* \\ \frac{\partial J(k)}{\partial \mathbf{h}^*(k+1)} &= \mathbf{h}(k+1) - \mathbf{h}(k) - \frac{1}{2}(\mathbf{h}(k+1)-\mathbf{h}(k))^* + \mathbf{A}^{\kappa}(k)\boldsymbol{\lambda}^* - \frac{1}{2}(\mathbf{A}^{\kappa}(k)\boldsymbol{\lambda}^*)^* \end{aligned}$$

where $\frac{\partial J(k)}{\partial \mathbf{w}^*} = [\frac{\partial J(k)}{\partial w_0^*}, \frac{\partial J(k)}{\partial w_1^*}, ..., \frac{\partial J(k)}{\partial w_M^*}]^T$ for $\mathbf{w} = \{\mathbf{u}, \mathbf{v}, \mathbf{g}, \mathbf{h}\}$. Setting the above derivatives to $\mathbf{0}$ and solving for $\boldsymbol{\lambda}$ gives the weight updates

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{A}^a(k)\big(\mathbf{A}^{aH}(k)\mathbf{A}^a(k)\big)^{-1}\mathbf{e}^*(k) \tag{5.3}$$

where

$$\begin{aligned}\mathbf{A}^a &= [\mathbf{A}^T, \mathbf{A}^{\imath T}, \mathbf{A}^{\jmath T}, \mathbf{A}^{\kappa T}]^T \\ \mathbf{e}(k) &= \mathbf{d}(k) - \left(\mathbf{w}^{aH}(k+1)\mathbf{A}^a(k)\right)^T\end{aligned}$$

To prevent the normalization matrix $\mathbf{A}^{aH}(k)\mathbf{A}^a(k)$ from becoming singular, a small regularization term $\varepsilon\mathbf{I} \in \mathbb{H}^{4N\times 4N}$ is typically added, where $\mathbf{I}$ is the identity matrix. A real step size μ can also be incorporated to improve steady state performance, giving the final weight update of the widely linear quaternion valued affine projection algorithm (WLQAPA) in the form

$$\mathbf{w}^a(k+1) = \mathbf{w}^a(k) + \mu\mathbf{A}^a(k)\left(\mathbf{A}^{aH}(k)\mathbf{A}^a(k) + \epsilon\mathbf{I}\right)^{-1}\mathbf{e}^*(k) \tag{5.4}$$

where ε takes a small value, and for stability $\mu < 2$ (shown later).

When the input vector is strictly linear, then $\mathbf{x}^a = \mathbf{x}$, and the WLQAPA simplifies into the strictly linear QAPA, for which the weight update takes the form

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\mathbf{A}(k)\left(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon\mathbf{I}\right)^{-1}\mathbf{e}^*(k) \tag{5.5}$$

where the error term $\mathbf{e}(k) = \mathbf{d}(k) - \left(\mathbf{w}^H(k+1)\mathbf{A}(k)\right)^T$ and the regularization term $\varepsilon\mathbf{I} \in \mathbb{H}^{N\times N}$.

5.3 Mean square error analysis of the strictly linear QAPA on noncircular data

To investigate the MSE performance, we next evaluate the MSE of the strictly linear QAPA, given by

$$MSE = \lim_{k\to\infty} E[|e(k)|^2] \tag{5.6}$$

where

$$e(k) = d(k) - \mathbf{w}^H(k)\mathbf{x}(k) \tag{5.7}$$

For generality, the teaching signal $d(k)$ is assumed to be coming from a widely linear system, taking the form

$$d(k) = \mathbf{w}_o^H \mathbf{x}(k) + \mathbf{v}_o^H \mathbf{x}^\imath(k) + \mathbf{g}_o^H \mathbf{x}^\jmath(k) + \mathbf{h}_o^H \mathbf{x}^\kappa(k) + v(k) \tag{5.8}$$

where $v(k)$ is zero mean quadruply white Gaussian noise.

To evaluate the mean square error we substitute for $\tilde{\mathbf{w}}(k) = \mathbf{w}_o - \mathbf{w}(k)$ into the weight update (5.5) to yield

$$\tilde{\mathbf{w}}(k+1) = \tilde{\mathbf{w}}(k) - \mu \mathbf{A}(k)\big(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon \mathbf{I}\big)^{-1}\mathbf{e}^*(k)$$

Upon applying the Hermitian transpose operator to both sides

$$\tilde{\mathbf{w}}^H(k+1) = \tilde{\mathbf{w}}^H(k) - \mu \mathbf{e}^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon \mathbf{I}\big)^{-1}\mathbf{A}^H(k) \tag{5.9}$$

and after post-multiplying by $\mathbf{A}(k)$ we obtain

$$\tilde{\mathbf{w}}^H(k+1)\mathbf{A}(k) = \tilde{\mathbf{w}}^H(k)\mathbf{A}(k) - \mu \mathbf{e}^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon \mathbf{I}\big)^{-1}\mathbf{A}^H(k)\mathbf{A}(k) \tag{5.10}$$

The a priori error, $\mathbf{e}_a(k)$, and a posteriori error $\mathbf{e}_p(k)$ can now be defined as

$$\mathbf{e}_a^T(k) = \tilde{\mathbf{w}}^H(k)\mathbf{A}(k) \qquad\qquad \mathbf{e}_p^T(k) = \tilde{\mathbf{w}}^H(k+1)\mathbf{A}(k)$$

which in combination with (5.10) gives

$$\mu \mathbf{e}^T(k)(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon \mathbf{I})^{-1} = \big(\mathbf{e}_a^T(k) - \mathbf{e}_p^T(k)\big)\big(\mathbf{A}^H(k)\mathbf{A}(k)\big)^{-1}$$

allowing us to rewrite (5.9) in the form

$$\tilde{\mathbf{w}}^H(k+1) = \tilde{\mathbf{w}}^H(k) - (\mathbf{e}_a^T(k) - \mathbf{e}_p^T(k))(\mathbf{A}^H(k)\mathbf{A}(k))^{-1}\mathbf{A}^H(k)$$

and to re-arrange the above terms and evaluate the energy ($\|\cdot\|^2$), to give

$$\|\mathbf{w}(k+1)\|^2 + \mathbf{e}_a^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k)\big)^{-1}\mathbf{e}_a^*(k) = \|\mathbf{w}(k)\|^2 + \mathbf{e}_p^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k)\big)^{-1}\mathbf{e}_p^*(k)$$

This expression is known as the energy conservation relationship [123]. Since we are interested in the mean square error at the steady state, that is, in the limit as $k \to \infty$, upon application of the statistical expectation operator we have

$$E\big[\mathbf{e}_a^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k)\big)^{-1}\mathbf{e}_a^*(k)\big] = E\big[\mathbf{e}_p^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k)\big)^{-1}\mathbf{e}_p^*(k)\big]$$

Substitute for $\mathbf{e}_p^T(k) = \mathbf{e}_a^T(k) - \mu\mathbf{e}^T(k)(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon\mathbf{I})^{-1}\mathbf{A}^H(k)\mathbf{A}(k)$ to arrive at

$$\begin{aligned}
E\big[\mathbf{e}_a^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k)\big)^{-1}\mathbf{e}_a^*(k)\big] &= E\big[\mathbf{e}_a^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k)\big)^{-1}\mathbf{e}_a^*(k)\big] \\
- \ \mu E\big[\mathbf{e}_a^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon\mathbf{I}\big)^{-1}\mathbf{e}^*(k)\big] \\
- \ \mu E\big[\mathbf{e}^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon\mathbf{I}\big)^{-1}\mathbf{e}_a^*(k)\big] \\
+ \ \mu^2 E\big[\mathbf{e}^T(k)\big(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon\mathbf{I}\big)^{-1}\mathbf{A}^H(k)\mathbf{A}(k)\big(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon\mathbf{I}\big)^{-1}\mathbf{e}^*(k)\big]
\end{aligned}$$

For this to be satisfied, it must hold that

$$\mu E\big[\mathbf{e}^T(k)\mathbf{D}(k)\mathbf{e}^*(k)\big] = E\big[\mathbf{e}_a^T(k)\mathbf{C}(k)\mathbf{e}^*(k)\big] + E\big[\mathbf{e}^T(k)\mathbf{C}(k)\mathbf{e}_a^*(k)\big] \tag{5.11}$$

where

$$\begin{aligned}
\mathbf{C}(k) &= (\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon\mathbf{I})^{-1} \\
\mathbf{D}(k) &= (\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon\mathbf{I})^{-1}\mathbf{A}^H(k)\mathbf{A}(k)(\mathbf{A}^H(k)\mathbf{A}(k) + \epsilon\mathbf{I})^{-1}
\end{aligned}$$

From (5.7), it can be shown that the MSE and the a priori error $\mathbf{e}_a(k)$ are related by

$$\mathbf{e}^T(k) = \mathbf{e}_a^T(k) + \mathbf{w}^{cH}(k)\mathbf{A}^c(k) + \mathbf{v}^T(k) \tag{5.12}$$

where

$$\mathbf{A}^c(k) = [\mathbf{A}^{\imath T}(k), \mathbf{A}^{\jmath T}(k), \mathbf{A}^{\kappa T}(k)]^T \tag{5.13}$$

$$\mathbf{w}^c(k) = [\mathbf{v}_o^T(k), \mathbf{g}_o^T(k), \mathbf{h}_o^T(k)]^T \tag{5.14}$$

From the definition of the error vector $\mathbf{e}(k) = [e(k), e(k-1), \ldots, e(k-N+1)]^T$ and using the symbol $(\cdot)_1$ to define the first element of the term in hand, the mean square error can now be expressed as

$$MSE = \lim_{k\to\infty} E\|e(k)\| = \lim_{k\to\infty} E\|e_a(k) + v(k) + \left(\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\right)_1\|^2 \tag{5.15}$$

Using the usual 'independence assumptions' and assuming that $\lim_{k\to\infty} E[e_a(k)] = E[v(k)] = 0$, we arrive at

$$MSE = \lim_{k\to\infty} E[\|e_a(k)\|^2] + \|(\mathbf{w}^{cH}(k)\mathbf{A}^c(k))_1\|^2 + E\|v(k)\| \tag{5.16}$$

$$MSE = EMSE + \|\left(\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\right)_1\|^2 + \sigma_v^2 \tag{5.17}$$

where the acronym EMSE is the excess mean square error, that is, the deviation from the optimal theoretical MSE, defined as

$$EMSE = \lim_{k\to\infty} E[\|e_a(k)\|^2] \tag{5.18}$$

To find an expression for $\lim_{k\to\infty} E[\|e_a(k)\|^2]$ using (5.11), we proceed by simplifying the three components in (5.11). Using (5.12) to substitute for $\mathbf{e}(k)$ in $\mu E[\mathbf{e}^T(k)\mathbf{D}(k)\mathbf{e}^*(k)]$ and taking the limit as $k \to \infty$ ($E[\mathbf{e}_a(k)] = 0$]) we have[2]

$$\begin{aligned}\mu E[\mathbf{e}^T(k)\mathbf{D}(k)\mathbf{e}^*(k)] &= \mu E[\mathbf{e}_a^T(k)\mathbf{D}(k)\mathbf{e}_a^*(k)] + \mu E[\mathbf{v}^T(k)\mathbf{D}(k)\mathbf{v}^*(k)] \\ &\quad + \mu E[\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\mathbf{D}(k)\mathbf{A}^{cH}(k)\mathbf{w}^c(k)]\end{aligned}$$

[2]We make the usual 'independence' assumptions that $\mathbf{e}_a(k)$ and $\mathbf{v}(k)$ is statistically independent from $\mathbf{A}(k)$.

Repeating this procedure for the other two terms in (5.11) gives

$$\begin{aligned} E[\mathbf{e}_a^T(k)\mathbf{C}(k)\mathbf{e}^*(k)] &= E[\mathbf{e}_a^T(k)\mathbf{C}(k)\mathbf{e}_a^*(k)] \\ E[\mathbf{e}^T(k)\mathbf{C}(k)\mathbf{e}_a^*(k)] &= E[\mathbf{e}_a^T(k)\mathbf{C}(k)\mathbf{e}_a^*(k)] \end{aligned}$$

allowing us to re-write (5.11) as

$$\begin{aligned} \mu E[\mathbf{e}_a^T(k)\mathbf{D}(k)\mathbf{e}_a^*(k)] &+ \mu E[\mathbf{v}^T(k)\mathbf{D}(k)\mathbf{v}^*(k)] \\ &+ \mu E[\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\mathbf{D}(k)\mathbf{A}^{cH}(k)\mathbf{w}^c] = 2E[\mathbf{e}_a^T(k)\mathbf{C}(k)\mathbf{e}_a^*(k)(k)] \end{aligned} \tag{5.19}$$

This form cannot be used to obtain an expression for EMSE, and to this end, we shall now rewrite each of the three terms in (5.19) in a more convenient form, starting with $E[\mathbf{e}_a(k)\mathbf{D}(k)\mathbf{e}_a^H(k)]$.

The matrix $\mathbf{D}(k)$ is Hermitian and hence positive definite, and therefore the above term is a positive scalar and we therefore have[3]

$$\begin{aligned} \mu E[\mathbf{e}_a^T(k)\mathbf{D}(k)\mathbf{e}_a^*(k)] &= \mu\Re\left[Tr(E\left[\mathbf{e}_a^*(k)\mathbf{e}_a^T(k)\mathbf{D}(k)\right])\right] \\ &= \mu\Re\left[Tr(E[\mathbf{e}_a^*(k)\mathbf{e}_a^T(k)]E[\mathbf{D}(k)])\right] \end{aligned}$$

It can be easily shown that at high signal to noise ratio (SNR), at steady state we have $E[\mathbf{e}_a^*(k)\mathbf{e}_a(k)] = E|e_a(k)|^2\mathbf{S}$, where for a step size μ close to unity, $\mathbf{S} \approx [\mathbf{1}\mathbf{1}^T]$ where $\mathbf{1} = [1, 0, \ldots, 0]^T$. Thus, for a step size approaching zero $\mathbf{S} \approx \mathbf{I}$ and using the property $Tr(c\mathbf{A}) = cTr(\mathbf{A})$ we have

$$\mu E[\mathbf{e}_a^T(k)\mathbf{D}(k)\mathbf{e}_a^*(k)] = \mu E|e_a(k)|^2\Re\left[Tr(\mathbf{S}E[\mathbf{D}(k)])\right]$$

Following the same approach for the other two terms in (5.19) gives

$$\begin{aligned} E[\mathbf{e}_a^T(k)\mathbf{C}(k)(\mathbf{e}_a^*(k)] &= E|e_a(k)|^2\Re\left[Tr(\mathbf{S}E[\mathbf{C}(k)])\right] \\ \mu E[\mathbf{v}^T(k)\mathbf{D}(k)\mathbf{v}^*(k)] &= \mu\sigma_v^2\Re\left[Tr(E[\mathbf{D}(k)])\right] \end{aligned}$$

[3]The real part operator $\Re[\cdot]$ arises because although quaternion multiplication is noncommutative, the real component of a quaternion product is commutative for cyclic permutations of the product.

allowing us to re-write (5.19) as

$$E|e_a(k)|^2\big(2\Re[Tr(\mathbf{S}E[\mathbf{C}(k)])] - \mu\Re[Tr(\mathbf{S}E[\mathbf{D}(k)])]\big) \;=\; \mu\sigma_v^2\Re\big[Tr(E[\mathbf{D}(k)])\big] + \mu E\big[\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\mathbf{D}(k)\mathbf{A}^{cH}(k)\mathbf{w}^c(k)\big]$$

to give the expression for EMSE in the final form

$$EMSE = \frac{\sigma_v^2\mu\Re[Tr(E[\mathbf{D}(k)])] + \mu E[\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\mathbf{D}(k)\mathbf{A}^{cH}(k)\mathbf{w}^c(k)]}{2\Re[Tr(\mathbf{S}E[\mathbf{C}(k)])] - \mu\Re[Tr(\mathbf{S}E[\mathbf{D}(k)])]} \tag{5.20}$$

Finally, for the theoretical MSE we have

$$\begin{aligned} MSE &= \frac{\sigma_v^2\mu\Re[Tr(E[\mathbf{D}(k)])] + \mu E[\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\mathbf{D}(k)\mathbf{A}^{cH}(k)\mathbf{w}^c(k)]}{2\Re[Tr(\mathbf{S}E[\mathbf{C}(k)])] - \mu\Re[Tr(\mathbf{S}E[\mathbf{D}(k)])]} \\ &+ |(\mathbf{w}^{cH}(k)\mathbf{A}^c(k))_1|^2 + \sigma_v^2 \end{aligned} \tag{5.21}$$

Remark#15: For a small value of the regularization parameter ϵ, we can assume $\mathbf{C}(k) = \mathbf{D}(k)$ and the EMSE simplifies into

$$EMSE = \frac{\sigma_v^2\mu\Re[Tr(E[\mathbf{C}(k)])] + \mu E[\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\mathbf{C}(k)\mathbf{A}^{cH}(k)\mathbf{w}^c(k)]}{(2-\mu)\Re[Tr(\mathbf{S}E[\mathbf{C}(k)])]} \tag{5.22}$$

Observe that due to the term in the denominator the EMSE diverges for $\mu \approx 2$, imposing a bound on the stepsize $0 < \mu < 2$. The analysis of APA class of methods is routinely performed separately for the small and big stepsize cases. The small stepsize cases is addressed in Remark 16 whereas the big stepsize case is addressed in Remark 17.

Remark#16: (small stepsize analysis) For a small value of the regularization parameter ϵ and a small step size μ, we can assume $\mathbf{C}(k) = \mathbf{D}(k)$ and $\mathbf{S} = \mathbf{I}$, allowing us to simplify the EMSE as

$$EMSE = \frac{\sigma_v^2\mu}{(2-\mu)} + \frac{\mu E[\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\mathbf{C}(k)\mathbf{A}^{cH}(k)\mathbf{w}^c(k)]}{(2-\mu)\Re[Tr(E[\mathbf{C}(k)])]} \tag{5.23}$$

Remark#17: (large stepsize analysis) For a small value of the regularization parameter ϵ and a large step size $\mu \approx 1$, we can assume $\mathbf{C}(k) = \mathbf{D}(k)$ and $\mathbf{S} = \mathbf{1}\mathbf{1}^T$, allowing us to

simplify the EMSE into

$$EMSE = \frac{\sigma_v^2 \mu \Re[Tr(E[\mathbf{C}(k)])] + \mu E[\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\mathbf{C}(k)\mathbf{A}^{cH}(k)\mathbf{w}^c(k)]}{(2-\mu)\Re[(E[\mathbf{C}(k)])_{1,1}]} \tag{5.24}$$

where $(\cdot)_{x,y}$ is the element in the x^{th} row and y^{th} column of the bracketed term.

Remark#18: (MSE for circular signals) The term $E[\mathbf{w}^{cH}(k)\mathbf{A}^c(k)\mathbf{C}(k)\mathbf{A}^{cH}(k)\mathbf{w}^c(k)]$ in (5.23) and (5.24) vanishes, giving the EMSE of strictly linear QAPA for proper signals. For improper data, however, this term can have a high value, illustrating the inadequacy of the strictly linear QAPA for the processing of second order noncircular signals.

5.4 Mean square error analysis of the widely linear QAPA on noncircular data

Following the same approach as for the QAPA, we can re-write the weight updates of the widely linear QAPA based on the augmented weight error vector $\tilde{\mathbf{w}}^a(k) = \mathbf{w}_o^a - \mathbf{w}^a(k)$, to give

$$\tilde{\mathbf{w}}^a(k+1) = \tilde{\mathbf{w}}^a(k) - \mathbf{A}^a(k)\big(\mathbf{A}^{aH}(k)\mathbf{A}^a(k) + \delta\mathbf{I}\big)^{-1}\mathbf{e}^*(k)$$

In this way, the expression for the conservation of energy of the weight update takes the form

$$\|\mathbf{w}^a(k+1)\|^2 + \mathbf{e}_a^T(k)\big(\mathbf{A}^{aH}(k)\mathbf{A}^a(k)\big)^{-1}\mathbf{e}_a^*(k) = \|\mathbf{w}^a(k)\|^2 + \mathbf{e}_p^T(k)\big(\mathbf{A}^{aH}(k)\mathbf{A}^a(k)\big)^{-1}\mathbf{e}_p^*(k)$$

where the a posteriori error $\mathbf{e}_p(k)$ and a priori error $\mathbf{e}_a(k)$ are defined as

$$\mathbf{e}_p^T(k) = \tilde{\mathbf{w}}^{aH}(k+1)\mathbf{A}^a(k) \tag{5.25}$$

$$\mathbf{e}_a^T(k) = \tilde{\mathbf{w}}^{aH}(k)\mathbf{A}^a(k) \tag{5.26}$$

and

$$\mathbf{e}(k) = \mathbf{e}_a(k) + \mathbf{v}(k) \tag{5.27}$$

Comparing with the error in (5.12) for QAPA, we observe that the expression for WLQAPA does not comprise the term, $\mathbf{w}^{cH}\mathbf{A}^c$, and MSE takes the form

$$MSE = \lim_{k\to\infty} E[\|e_a(k)\|^2] + E\|v(k)\| \tag{5.28}$$

$$MSE = EMSE + \sigma_v^2 \tag{5.29}$$

Following on the analysis of EMSE for the strictly linear QAPA in (5.18)-(5.20), we obtain the EMSE for WLQAPA in the form

$$EMSE = E|e_a(k)|^2 = \frac{\sigma_v^2\mu\Re[Tr(E[\mathbf{D}(k)])]}{2\Re[Tr(\mathbf{S}E[\mathbf{C}(k)])] - \mu\Re[Tr(\mathbf{S}E[\mathbf{D}(k)])]}$$

where

$$\mathbf{C}(k) = \left(\mathbf{A}^{aH}(k)\mathbf{A}^a(k) + \epsilon\mathbf{I}\right)^{-1} \tag{5.30}$$

$$\mathbf{D}(k) = \left(\mathbf{A}^{aH}(k)\mathbf{A}^a(k) + \epsilon\mathbf{I}\right)^{-1}\mathbf{A}^{aH}(k)\mathbf{A}^a(k)\left(\mathbf{A}^{aH}(k)\mathbf{A}^a(k) + \epsilon\mathbf{I}\right)^{-1} \tag{5.31}$$

Remark#19: For a small value of the regularization parameter ϵ, we can assume that $\mathbf{C}(k) = \mathbf{D}(k)$, allowing us to simplify the EMSE into

$$EMSE = E|e_a(k)|^2 = \frac{\sigma_v^2\mu\Re[Tr(E[\mathbf{C}(k)])]}{(2-\mu)\Re[Tr(\mathbf{S}E[\mathbf{C}(k)])]} \tag{5.32}$$

Similarly to the QAPA, the bounds $0 < \mu < 2$ are imposed on the step size μ. Below we look at the cases with a small and big stepsize.

Remark#20: (small stepsize analysis) For a small value of the regularization parameter ϵ and a small step size μ, we can assume $\mathbf{C}(k) = \mathbf{D}(k)$ and $\mathbf{S} = \mathbf{I}$, allowing us to simplify the EMSE into

$$EMSE = E|e_a(k)|^2 = \frac{\sigma_v^2\mu}{(2-\mu)} \tag{5.33}$$

Note that this expression conforms to that obtained in the complex domain in [123].

Remark#21: (large stepsize analysis) For a small value of the regularization parameter ϵ and a step size $\mu \approx 1$, we can assume $\mathbf{C}(k) = \mathbf{D}(k)$ and $\mathbf{S} = \mathbf{1}\mathbf{1}^T$, allowing us to simplify the EMSE into

$$EMSE = E|e_a(k)|^2 = \frac{\sigma_v^2 \mu \Re[Tr(E[\mathbf{C}(k)])]}{(2-\mu)(E[\mathbf{C}(k)])_{1,1}} \tag{5.34}$$

When the cross-diagonal terms of $E[\mathbf{C}(k)]$ are small compared to the diagonal terms (i.e. the regression vectors are orthogonal or close to being orthogonal), we can further assume that

$$\frac{1}{(E[\mathbf{C}(k)])_{1,1}} \approx Tr(\mathbf{R}_{xx}^a) \qquad \Re[Tr(E[\mathbf{C}(k)])] \approx E\left[\frac{N}{||\mathbf{x}^a(k)||^2}\right] \tag{5.35}$$

where $\mathbf{R}_{xx}^a = E[\mathbf{x}^a(k)\mathbf{x}^{aH}(k)]$, allowing us to simplify the expression for EMSE as

$$EMSE = \frac{\sigma_v^2 \mu}{(2-\mu)} Tr(\mathbf{R}_{xx}^a) E\left[\frac{N}{||\mathbf{x}^a(k)||^2}\right] \tag{5.36}$$

Notice that, similarly to the complex domain [123], the EMSE is proportional to the number of constraints N.

Remark#22: Compared to the EMSE expression in (5.24), the EMSE of WLQAPA in (5.34) does not depend on the noncircularity of the signal, and is second order optimal for the generality of quaternion signals, both proper and improper. WLQAPA therefore achieves the same MSE as the strictly linear QAPA for proper signals and greatly enhanced performance for improper signals. The QAPA offers a lower computational cost and faster convergence for circular signals due to the fewer weights that are to be adjusted but is inadequate for general noncircular signals.

5.5 Simulations

The quaternion affine projection algorithm and its widely linear extension are next validated by comprehensive simulations over a number of scenarios. All the simulations were conducted in the prediction setting (1 step ahead), with filter length $M = 10$ and the

regularization parameter $\epsilon = 0.01$. The test signals employed in the simulations were:

- The circular AR(1) process driven by both quadruply white circular Gaussian noise given by

$$y(k) = 0.9y(k-1) + n(k) \tag{5.37}$$

- The noncircular MA(3) process driven by quadruply white noncircular Gaussian noise

$$y(k) = ax(k-1) + bx^{\imath}(k-2) + cx^{\kappa}(k-3) + n(k) \tag{5.38}$$

  where $\{a, b, c\}$ are quaternion valued coefficients.

- The widely linear autoregressive moving average process driven by quadruply white circular Gaussian noise, giving the noncircular signal

$$y(k) = y(k-1) + 2n(k) + 0.5n^{\imath}(k) + n(k-1) + 0.9n^{\jmath}(k-1) \tag{5.39}$$

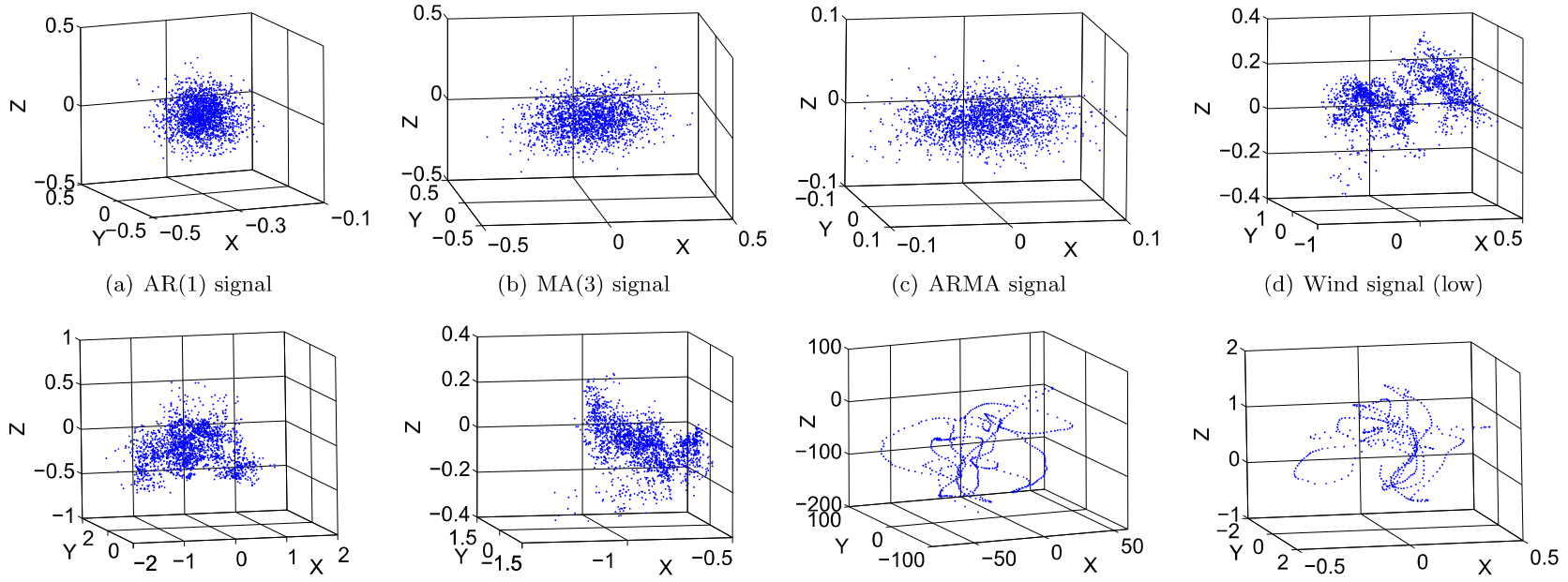
- The 4D noncircular wind signal[4] with three different dynamic regions, identified as low, medium and high dynamics, based on the changes in the wind intensity.

- The 3D noncircular body motion signal. Two gyroscopes were placed on the left hand and right hand of an athlete performing Tai Chi movements, recording two three-dimensional signals.[5]

Figure 5.1 illustrates a geometric notion of noncircularity, by showing the scatter plots of the quaternion-valued signals considered. Observe that only the AR(1) signal had a rotation invariant distribution (circular) and that all the real world signals (wind, body motion) were noncircular.

[4]The wind speed measurement in the North, East and vertical direction formed the imaginary part of the full quaternion while the temperature was incorporated in the real part to form a full quaternion. The dataset was recorded using the WindMaster, a 3D Gill Instruments ultrasonic anemometer, which was resampled at 5 Hz for simulation purposes.

[5]The motion data was recorded using the XSense MTx 3DOF Orientation Tracker.



(a) AR(1) signal (b) MA(3) signal (c) ARMA signal (d) Wind signal (low)

(e) Wind signal (medium) (f) Wind signal (high) (g) Tai Chi body motion (left) (h) Tai Chi body motion (right)

Figure 5.1: Geometric view of circularity. To visualise the circularity on a three dimensional plot, only the i-, j- and k-components of the quaternion signals are plotted within a scatter diagram. Observe that only the AR(1) signal in (a) is circular Gaussian. The MA(3) and ARMA signals are both noncircular Gaussian signals. The real world wind signal and Tai Chi body motion signal both exhibit noncircular distributions.

In the first experiment we addressed by simulations the theoretical stability bound on the learning rate μ for the class of APA algorithms derived in Section 5.3 and 5.4. Figures 5.2-5.3 show that both QAPA and WLQAPA have a stability bound of $\mu < 2$ for all values of the constraint N, conforming with the corresponding bound in the complex domain [113] and Remark 15 and 19. Figure 5.4 shows the learning curves of

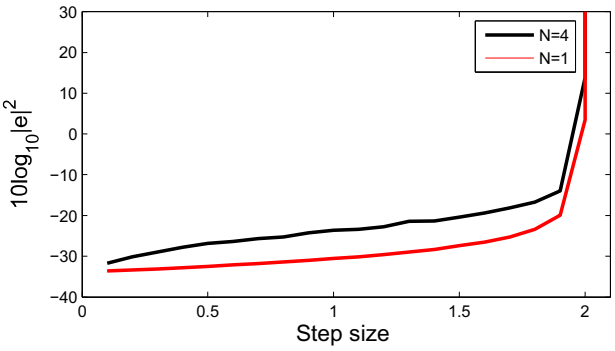


Figure 5.2: Simulated MSE of QAPA as a function of the step size μ, for the improper ARMA process in (5.39).

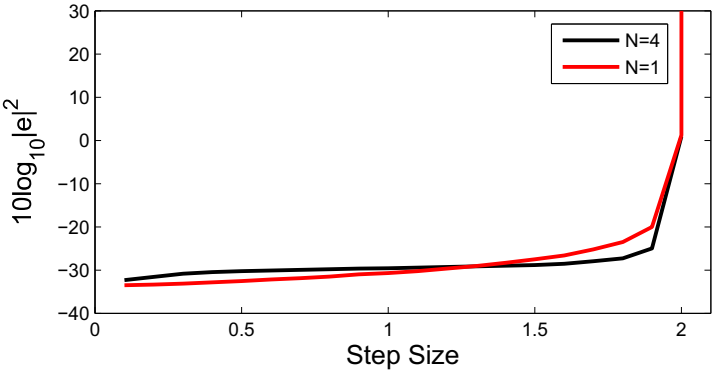


Figure 5.3: Simulated MSE of WLQAPA as a function of the step size μ, for the improper ARMA process in (5.39).

QAPA and WLQAPA for the circular AR(1) process. Observe that, for the second order circular (proper) process both the strictly linear QAPA and the widely linear WLQAPA algorithms offered the same steady state performance and that QAPA exhibited faster initial convergence due to having fewer coefficients to adjust, conforming with Remark 22.

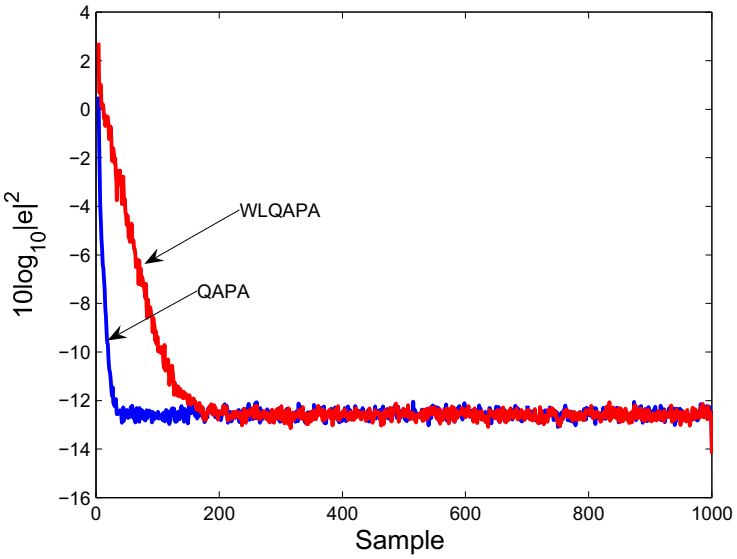


Figure 5.4: Learning curves for QAPA and WLQAPA for the circular AR process in (5.37), with $\mu = 1$ and $N = 2$.

Figure 5.5 repeats the previous experiment for the noncircular MA process; the WLQAPA was able to track the underlying dynamics of the signal in a second order optimal manner and therefore offered better steady state performance compared to the QAPA, validating the analysis in Section 5.4. In Figure 5.6 we show how the performance of the WLQAPA on the noncircular MA process varies as the number of constraints N is increased, when the driving noise in (5.38) was coloured. This is inline with the analysis in (5.36) (see Remark 21). The steady state performance decreased as the number of constraints N increased from $N = 1$ (where the filter is equivalent to the normalized QLMS) to $N = 4$. Also, as is the case with all APA algorithms [18] the convergence rate was faster as the number of constraints increased; this is expected as the driving noise to the model (5.38) was coloured. Figure 5.7 shows the learning curves for the noncircular ARMA process, with WLQAPA being able to better track the underlying dynamics of the signal, therefore offering enhanced steady state performance.

In order to verify the theoretical result in (5.36) for the steady state performance of WLQAPA, we next performed one step ahead prediction of the improper ARMA signal in (5.39) and improper MA(3) process in (5.38) and compared the measured MSEs to the

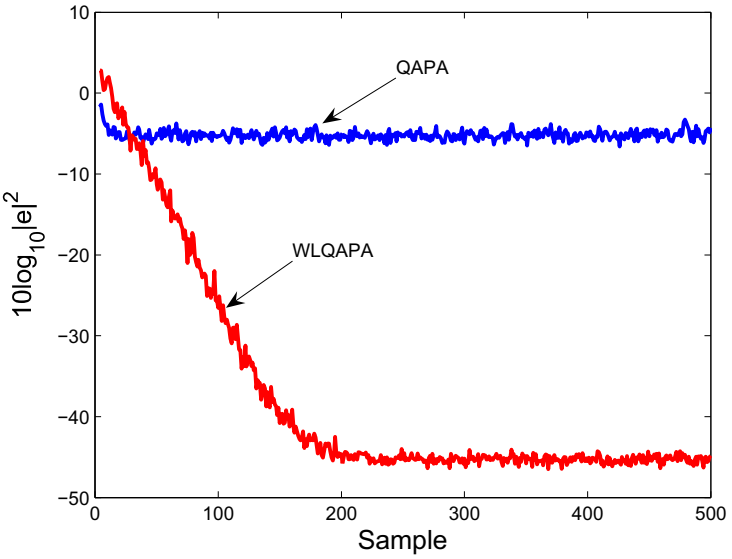


Figure 5.5: Learning curves of QAPA and WLQAPA for the noncircular MA process in (5.38), with $\mu = 1$ and $N = 2$.

theoretical MSEs in (5.36). Figure 5.8 compares the theoretical MSE derived in (5.36) to the simulated MSE for a step size in the range $\mu \in [0.05 - 1]$. This allows us to make two observations. First, the theoretical MSE closely matches the simulated MSE when the number of constraints $N = 1$. Second, for large stepsizes ($\mu > 0.5$), as the number of constraints increases, the theoretical bound in (5.36) is no longer an accurate estimate of MSE. We can explain the discrepancy that exists for $N > 1$ by the assumptions made in deriving the theoretical MSE expression in (5.36) as elaborated in the Appendix C.

In Figure 5.9 we compare the theoretical MSE to the simulated MSE for the MA signal in (5.38). Observe that in this case the theoretical MSE closely matches the simulated MSE for every value of the constraint N. Note also that as the step size approaches 0 the performance difference between the number of constraints $N = 1$ and $N = 4$ decreases. This is expected from equation (5.33) which shows that for a small step size the MSE is independent of the number of constraints.

Table 5.1 illustrates the average MSE of the proposed algorithms in all the scenarios considered. Columns 1-3 show the results for the synthetic data. For the proper AR(1) model the performances of QAPA and WLQAPA were nearly identical, illustrating the

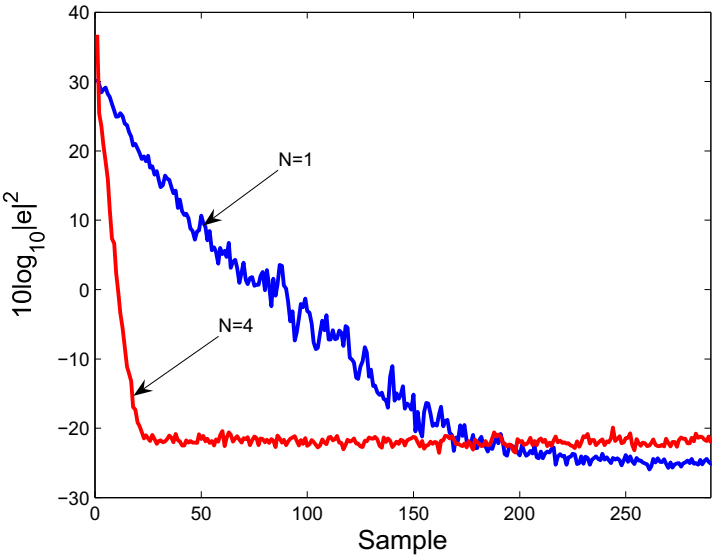


Figure 5.6: Learning curves of WLQAPA for the noncircular MA process in (5.38), for $N = 1$ and $N = 4$.

ability of WLQAPA to deal with both proper and improper data. The MA(3) model had by design a very high degree of noncircularity (improper noise driving a widely linear model) and consequently the strictly linear QAPA was inadequate, the fact also confirmed in Figure 5.5. The behaviour for the improper ARMA process was along the same lines. Columns 4-8 show the MSE for both the QAPA and WLQAPA for the one step ahead prediction of the real world 4D wind signal and two 3D Tai Chi body motion signals, for $M = 10$, $N = 4$ and $\mu = 1$. For the three wind regimes (low, medium, high) of different dynamics, due to the noncircular nature of wind (see Figure 5.1), the WLQAPA offered better steady state performance for all the three wind regimes considered. As shown in Figure 5.1 the Tai Chi body motion signal for both the left hand and right hand was also highly noncircular, consequently, the WLQAPA offered better tracking performance than the QAPA.

5.6 Summary

A class of quaternion affine projection algorithms (QAPA) have been introduced in order to provide a unified platform for fast and accurate adaptive filtering of both second order

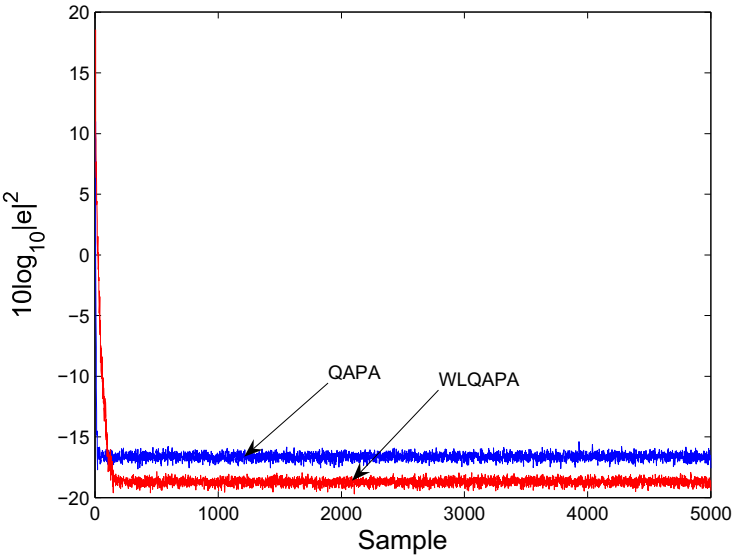


Figure 5.7: Learning curves of QAPA and WLQAPA for the noncircular ARMA process in (5.39), with $\mu = 1$ and $N = 2$.

Table 5.1: MSE of QAPA and WLQAPA for different processes. Key: 1. AR(1), 2. MA(3), 3. ARMA, 4. Tai Chi (left), 5. Tai Chi (right), 6. Wind (low), 7. Wind (medium), 8. Wind (high).

MSE (dB)	1	2	3	4	5	6	7	8
QAPA	-12.48	-4.19	-16.11	-20.38	-20.86	-13.72	-13.90	-11.13
WLQAPA	-12.49	-45.86	-17.39	-22.31	-23.26	-15.24	-15.31	-12.49

circular (proper) and noncircular (improper) real world signals. This has been achieved using the recent advances in the quaternion statistics (augmented quaternion statistics) and the emergence of the $\mathbb{HR}$-calculus for gradient calculation. Expressions for the MSE of the QAPA and WLQAPA have been obtained for the general case of improper signals, highlighting the advantage offered by the widely linear WLQAPA over the strictly linear QAPA in real world scenarios. Simulations on both circular and noncircular signals support the analysis.

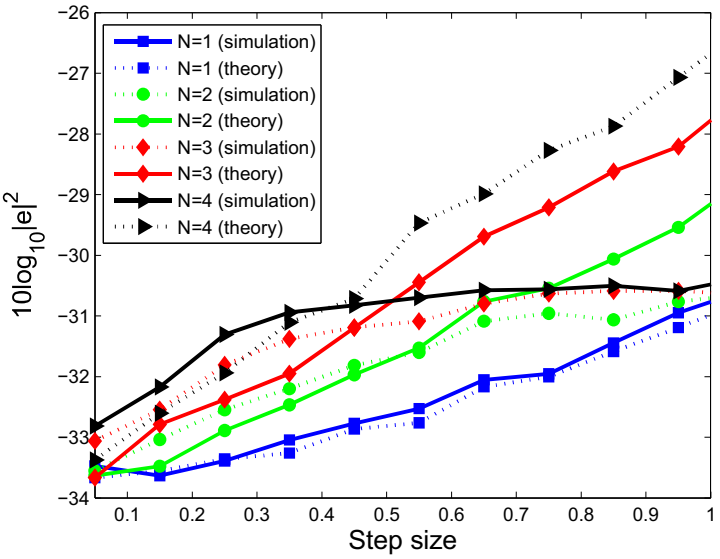


Figure 5.8: Comparison of the theoretical bound in (5.34) and simulated steady state MSEs for WLQAPA, based on noncircular ARMA signal in (5.39), when $N = 1, 2, 3$ and 4.

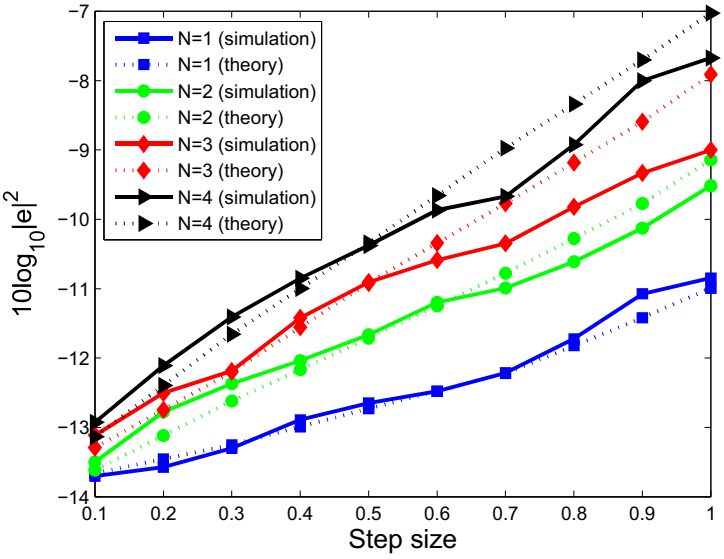


Figure 5.9: Comparison of the theoretical bound in (5.34) and simulated steady state MSE of WLQAPA, on noncircular MA signal in (5.38) when $N = 1, 2, 3$ and 4.

Chapter 6

A Class of Quaternion Kalman Filters

6.1 Introduction

The Kalman filter is a standard tool in sequential state space estimation, and has found application in a number of areas, including space navigation [124], training of neural networks [125] [126] [127] [46], and sensor fusion [128] [129]. The extended Kalman filter enhances the scope of Kalman filtering to nonlinear state and observation models. It has proven invaluable in, for instance, nonlinear attitude estimation [130], and has also recently been combined with the quaternion domain representation to offer an effective way for solving three-dimensional attitude estimation problems [69] [131].

The benefits of quaternions are conveniently illustrated in spacecraft orientation tracking, an important application of Kalman filters, where the aim is to find the mapping between the coordinate system on a reference-frame $\mathbf{r} \in \mathbb{R}^3$ and a local frame $\mathbf{b} \in \mathbb{R}^3$ on the spacecraft body-frame, such that $\mathbf{b} = \mathbf{A}\mathbf{r}$ where $\mathbf{r}$ is a reference-frame vector, $\mathbf{b}$ is the body-frame vector and $\mathbf{A}$ is the attitude matrix (or rotation matrix). Quaternions provide a convenient closed form solution (Davenport's q-method [132]) to this mapping, which is generally referred to as Wahba's problem [133]. This is largely due to a more compact notation for rotations compared to that obtained with a 3×3 matrix, resulting in

fewer constraints and a more mathematically tractable functional expression. Numerous authors have used this quaternion representation to track orientation in three-dimensional spaces [134] [69] [131], and for the training of quaternion valued neural networks for time series prediction [135].

Existing quaternion Kalman filters are, however, not intrinsically quaternion valued as they use quaternions in the form of a quadrivariate real ordered vector, that is $[q_r, q_\imath, q_\jmath, q_\kappa] \in \mathbb{R}^4$ rather than using their natural representation $q = q_r + \imath q_\imath + \jmath q_\jmath + \kappa q_\kappa \in \mathbb{H}$. It would therefore be analytically more appropriate and practically more physically meaningful for the algorithms to be derived directly in the quaternion domain, rather than having to switch between quaternions and the reals. This would also allow for a more intuitive approach to the concept of circularity and widely linear modeling, key notions for the optimal processing of the generality of quaternion valued signals. Such a fully quaternion Kalman filter would not only be suited for orientation tracking but also for the processing of general three- and four-dimensional processes, and as an enabling tool in multidimensional learning systems.

One major obstacle to deriving a full quaternion Kalman filter is the inability of standard second order statistics to capture full information in quaternion correlation matrices. It has recently been shown that to fully utilize the second order statistical properties of quaternion valued signals, three pseudocovariance matrices must be used in addition to the standard covariance matrix $E[\mathbf{x}\mathbf{x}^H]$ [65] [64]; this is achieved by employing the widely linear model. Conventional quaternion signal processing algorithms are strictly linear, and are therefore only suitable for the processing of second order circular (proper) signals[1]. For orientation problems, a strictly linear model is usually sufficient because this transformation is described by a standard quaternion number. However, most real world signals are noncircular and should therefore be processed using a widely linear model in order to capture complete second order statistics available. Furthermore, as shown in this chapter, the widely linear model is a prerequisite for the development of extended Kalman filter.

[1]Second order circular signals have rotation invariant probability distributions.

In this chapter we develop a class of linear and widely linear Kalman filters, that operate directly in the quaternion domain, allowing for a unified treatment of both circular and noncircular data. For the widely linear Kalman filter an efficient implementation is also proposed, reducing the computations by approximately three quarter. For non-linear state space models, the quaternion extended Kalman filter is introduced and is shown to be inherently widely linear. This allows us to bridge the gap in the literature on three- and four- dimensional Kalman filters and provide an algorithmic support for learning systems that operate on vector sensor data.

The organization of the chapter is as follows; In Section 6.2 we introduce an efficient implementation of the widely linear model for three-dimensional signals. In Section 6.3 the quaternion Kalman filter is derived. Section 6.4 introduces the widely linear quaternion Kalman filter while in Section 6.5 we provide its efficient implementation. To cater for nonlinear state models, Section 6.6 presents the quaternion extended Kalman filter. Section 6.7 demonstrates the isomorophism that exists between the quaternion Kalman filter and the real Kalman filter in $\mathbb{R}^4$. In Section 6.8, simulations on both benchmark and real world signals illustrate the tracking capability of the algorithms derived.

6.2 Reduced widely linear model for three-dimensional signals

For many Kalman filtering applications, the aim is to track a three-dimensional quantity, as is the case in aeronautics. The widely linear model in (2.2.3) is the only rigorous model for general noncircular three-dimensional signals, and in order to reduce its computational complexity, we show that it is overparameterized and provide a simplified solution. To arrive at the more parsimonious form, we start from the linear model $\hat{y} = hx \in \mathbb{H}$ and,

using the duality between quaternions and real numbers, it can be rewriten in $\mathbb{R}^4$ as

$$\begin{bmatrix} \hat{y}_r \\ \hat{y}_\imath \\ \hat{y}_\jmath \\ \hat{y}_\kappa \end{bmatrix} = \mathbf{H}\mathbf{x} = \begin{bmatrix} u_r & u_\imath & u_\jmath & u_\kappa \\ -u_\imath & u_r & -u_\kappa & u_\jmath \\ -u_\jmath & u_\kappa & u_r & -u_\imath \\ -u_\kappa & -u_\jmath & u_\imath & u_r \end{bmatrix} \begin{bmatrix} x_r \\ x_\imath \\ x_\jmath \\ x_\kappa \end{bmatrix} \in \mathbb{R}^4$$

For the strictly linear model above, observe that the matrix $\mathbf{H}$ has only 4 out of the possible 16 degrees of freedom needed to describe the transformation from $\mathbf{x}$ to $\widehat{\mathbf{y}}$. If the signal $\mathbf{x}$ is three-dimensional (i.e. a pure quaternion), then the first column of $\mathbf{H}$ vanishes and matrix $\mathbf{H}$ reduces to a 4×3 matrix. Hence, to fully describe the transformation between the three dimensional vector $\mathbf{x}$ and the 4 dimensional vector $\mathbf{y}$, 12 degrees of freedom are needed. We next show that for three-dimensional signals, only three of the four terms that constitute the widely linear model are needed in order to obtain the required 12 degrees of freedom. Indeed, consider the reduced widely linear model for the three dimensional data, given by

$$\hat{y} = \mathbf{u}\mathbf{x} + \mathbf{v}\mathbf{x}^\imath + \mathbf{g}\mathbf{x}^\jmath \qquad \in \mathbb{H} \tag{6.1}$$

and rewrite it in $\mathbb{R}^4$ as

$$\begin{bmatrix} \hat{y}_r \\ \hat{y}_\imath \\ \hat{y}_\jmath \\ \hat{y}_\kappa \end{bmatrix} = \begin{bmatrix} u_r & u_\imath & u_\jmath & u_\kappa \\ -u_\imath & u_r & -u_\kappa & u_\jmath \\ -u_\jmath & u_\kappa & u_r & -u_\imath \\ -u_\kappa & -u_\jmath & u_\imath & u_r \end{bmatrix} \begin{bmatrix} x_r \\ x_\imath \\ x_\jmath \\ x_\kappa \end{bmatrix} + \begin{bmatrix} v_r & v_\imath & v_\jmath & v_\kappa \\ -v_\imath & v_r & -v_\kappa & v_\jmath \\ -v_\jmath & v_\kappa & u_r & -v_\imath \\ -v_\kappa & -v_\jmath & u_\imath & v_r \end{bmatrix} \begin{bmatrix} x_r \\ x_\imath \\ -x_\jmath \\ -x_\kappa \end{bmatrix}$$
$$+ \begin{bmatrix} g_r & g_\imath & g_\jmath & g_\kappa \\ -g_\imath & g_r & -g_\kappa & g_\jmath \\ -g_\jmath & g_\kappa & g_r & -g_\imath \\ -g_\kappa & -g_\jmath & g_\imath & g_r \end{bmatrix} \begin{bmatrix} x_r \\ -x_\imath \\ x_\jmath \\ -x_\kappa \end{bmatrix}$$

Upon absorbing the negative signs in the vector $\mathbf{x}$ into the coefficient matrices, we can rewrite the above as

$$\begin{bmatrix} \hat{y}_r \\ \hat{y}_\imath \\ \hat{y}_\jmath \\ \hat{y}_\kappa \end{bmatrix} = \mathbf{Hx} = \begin{bmatrix} u_r + v_r + g_r & u_\imath + v_\imath - g_\imath & u_\jmath - v_\jmath + g_\jmath & u_\kappa - v_\kappa - g_\kappa \\ -u_\imath - v_\imath - g_\imath & u_r + v_r - g_r & -u_\kappa + v_\kappa - g_\kappa & u_\jmath - v_\jmath - g_\jmath \\ -u_\jmath - v_\jmath - g_\jmath & u_\kappa + v_\kappa - g_\kappa & u_r - v_r + g_r & -u_\imath + v_\imath + g_\imath \\ -u_\kappa - v_\kappa - g_\kappa & -u_\jmath - v_\jmath + g_\jmath & u_\imath - v_\imath - g_\imath & u_r - v_r - g_r \end{bmatrix} \begin{bmatrix} x_r \\ x_\imath \\ x_\jmath \\ x_\kappa \end{bmatrix}$$

For $\mathbf{x}$ a pure quaternion, the first column of $\mathbf{H}$ vanishes while the remaining 12 terms in $\mathbf{H}$ are linearly independent, and are described by

$$\begin{aligned} u_r + v_r - g_r &= \mathbf{H}_{22} \qquad u_\imath + v_\imath - g_\imath = \mathbf{H}_{12} \\ u_r - v_r + g_r &= \mathbf{H}_{33} \qquad -u_\imath + v_\imath + g_\imath = \mathbf{H}_{34} \\ u_r - v_r - g_r &= \mathbf{H}_{44} \qquad u_\imath - v_\imath - g_\imath = \mathbf{H}_{43} \end{aligned}$$

and

$$\begin{aligned} -u_\jmath - v_\jmath + g_\jmath &= \mathbf{H}_{42} \qquad u_\kappa + v_\kappa - g_\kappa = \mathbf{H}_{32} \\ u_\jmath - v_\jmath + g_\jmath &= \mathbf{H}_{13} \qquad -u_\kappa + v_\kappa - g_\kappa = \mathbf{H}_{23} \\ u_\jmath - v_\jmath - g_\jmath &= \mathbf{H}_{24} \qquad u_\kappa - v_\kappa - g_\kappa = \mathbf{H}_{14} \end{aligned}$$

Observe that now the 12 equations above are linearly independent (this can be confirmed by solving for $\mathbf{Hx} = \mathbf{0}$) and therefore the reduced widely linear model is sufficient to model three dimensional data (pure quaternions).

6.3 Quaternion Kalman filter

Following on the derivation of the Kalman filter in the complex domain [20], we assume that the quaternion state $\mathbf{x}_k \in \mathbb{H}^{N\times 1}$ evolves according to the following model

$$\mathbf{x}_k = \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k \tag{6.2}$$

where $\mathbf{A}_k \in \mathbb{H}^{N\times N}$ is the state transition matrix, $\mathbf{B}_k \in \mathbb{H}^{N\times N}$ is the control-input matrix for the control input $\mathbf{u}_k \in \mathbb{H}^{N\times 1}$ and $\mathbf{w}_k \in \mathbb{H}^{N\times 1}$ is the state noise.

The state $\mathbf{x}_k$ cannot be observed directly but we can measure the quantity $\mathbf{z}_k \in \mathbb{H}^{M\times 1}$ that relates to the state $\mathbf{x}_k$ through

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{6.3}$$

where $\mathbf{H} \in \mathbb{H}^{M\times N}$ is the observation matrix and $\mathbf{v}_k \in \mathbb{H}^{M\times 1}$ is the measurement noise. Both the model noise and measurement noise are zero mean and Gaussian, with properties

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{ww,k}) \tag{6.4}$$

$$\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{vv,k}) \tag{6.5}$$

Since in the real world noise normally comprises of a number of independent sources, by the central limit theorem this is a reasonable assumption.

The a priori state estimate $\hat{\mathbf{x}}_{k|k-1} \in \mathbb{H}^{N\times 1}$ (the estimate of the state $\mathbf{x}_k$ before obtaining the new measurement) can be obtained from the state model

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k \tag{6.6}$$

where $\hat{\mathbf{x}}_{k-1|k-1}$ is the previous state estimate.

Using the measurement $\mathbf{z}_k$, the estimate $\hat{\mathbf{x}}_{k|k-1}$ can be improved to obtain the a

posteriori state estimate $\hat{\mathbf{x}}_{k|k} \in \mathbb{H}^{N\times 1}$, given by

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\big(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}\big) \tag{6.7}$$

where $\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}$ is the innovation term (the error between the estimate and the measurement). The aim is to find the Kalman gain $\mathbf{K}_k$ that minimizes the mean square error $E[\mathbf{e}_{k|k}^H\mathbf{e}_{k|k}]$, where the error is given by

$$\mathbf{e}_{k|k} = \mathbf{x}_k - \hat{\mathbf{x}}_{k|k} = \mathbf{x}_k - \big[\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1})\big]$$

Minimizing the error $E[\mathbf{e}_{k|k}^H\mathbf{e}_{k|k}]$ is equivalent to minimizing the trace of the error covariance $\mathbf{P}_k = cov(\mathbf{e}_{k|k}) = E[\mathbf{e}_{k|k}\mathbf{e}_{k|k}^H]$. This allows us to write the mean square error as

$$E[\mathbf{e}_{k|k}^H\mathbf{e}_{k|k}] = Tr[cov(\mathbf{x}_k - (\hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1})))]$$

Substituting for $\mathbf{z}_k$, we have

$$\begin{aligned} E[\mathbf{e}_{k|k}^H\mathbf{e}_{k|k}] &= Tr\big[cov\big(\mathbf{x}_k - \big(\hat{\mathbf{x}}_{k|k-1} \\ &\quad + \mathbf{K}_k(\mathbf{H}\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1})\big)\big)\big] \\ &= Tr\big[cov\big[(\mathbf{I} - \mathbf{K}_k\mathbf{H})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) - \mathbf{K}_k\mathbf{v}_k\big]\big] \end{aligned}$$

and using the property $cov(\mathbf{A}\mathbf{x}) = E[(\mathbf{A}\mathbf{x})(\mathbf{A}\mathbf{x})^H] = \mathbf{A}E[\mathbf{x}\mathbf{x}^H]\mathbf{A}^H$ (the Hermitian is commutative in the quaternion domain), we arrive at

$$\begin{aligned} E[\mathbf{e}_{k|k}^H\mathbf{e}_{k|k}] =& Tr\big[(\mathbf{I} - \mathbf{K}_k\mathbf{H})cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})(\mathbf{I} - \mathbf{K}_k\mathbf{H})^H \\ &+\mathbf{K}_k cov(\mathbf{v}_k)\mathbf{K}_k^H\big] \end{aligned}$$

Denoting the a priori error covariance matrix $cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})$ by $\mathbf{P}_{k|k-1}$, we can now

write

$$\begin{aligned} E[\mathbf{e}_{k|k}^H \mathbf{e}_{k|k}] &= Tr[(\mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k|k-1})(\mathbf{I} - \mathbf{K}_k\mathbf{H})^H \\ &\quad +\mathbf{K}_k\mathbf{R}_{vv,k}\mathbf{K}_k^H]] \\ &= Tr[\mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k|k-1} \\ &\quad -\mathbf{P}_{k|k-1}\mathbf{H}^H\mathbf{K}_k^H + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^H] \end{aligned} \tag{6.8}$$

where $\mathbf{S}_k$ is independent of $\mathbf{K}_k$ and is given by $\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^H + \mathbf{R}_{vv,k}$. Upon calculating the derivatives (found using $\mathbb{HR}$-calculus) with respect to $\mathbf{K}_k$ and setting to zero, we obtain

$$\begin{aligned} \frac{\partial E[\mathbf{e}_{k|k}^H \mathbf{e}_{k|k}]}{\partial \mathbf{K}_k} = \frac{\partial Tr[\mathbf{K_k}\mathbf{H}\mathbf{P}_{k|k-1}]}{\partial \mathbf{K}_k} - \frac{\partial Tr[\mathbf{P}_{k|k-1}\mathbf{H}^H\mathbf{K}_k^H]}{\partial \mathbf{K}_k} \\ +\frac{\partial \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^H}{\partial \mathbf{K}_k} = \mathbf{0} \end{aligned} \tag{6.9}$$

where the partial derivatives can be written as (observe that these derivatives are different to what would be obtained in the complex domain)

$$(\mathbf{S}_k\mathbf{K}_k^H)^T - \frac{1}{2}(\mathbf{K}_k\mathbf{S}_k) = (\mathbf{H}\mathbf{P}_{k|k-1})^T - \frac{1}{2}\mathbf{P}_{k|k-1}\mathbf{H}^H \tag{6.10}$$

The following decomposition is important for the derivation

$$\begin{aligned} ((\mathbf{S}_k\mathbf{K}_k^H)^T)^* &= (\mathbf{S}_k\mathbf{K}_k^H)^H = \mathbf{K}_k\mathbf{S}_k^H = \mathbf{K}_k\mathbf{S}_k \\ ((\mathbf{H}\mathbf{P}_{k|k-1})^T)^* &= (\mathbf{H}\mathbf{P}_{k|k-1})^H = \mathbf{P}_{k|k-1}^H\mathbf{H}^H = \mathbf{P}_{k|k-1}\mathbf{H}^H \end{aligned}$$

allowing us to obtain a solution to (6.10) in the form $(\mathbf{S}_k\mathbf{K}_k^H)^T = (\mathbf{H}\mathbf{P}_{k|k-1})^T$ (since this also implies $-\frac{1}{2}\mathbf{K}_k\mathbf{S}_k = -\frac{1}{2}\mathbf{P}_{k|k-1}\mathbf{H}^H$), finally giving

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^H\mathbf{S}_k^{-1} \tag{6.11}$$

Therefore, the a priori error covariance matrix $\mathbf{P}_{k|k-1}$ can be solved recursively as

$$\begin{aligned}\mathbf{P}_{k|k-1} &= cov[\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}] \\ &= \mathbf{A}E[\mathbf{e}_{k|k-1}\mathbf{e}_{k|k-1}^H]\mathbf{A}^H + 2\mathbf{A}E[\mathbf{e}_{k|k-1}\mathbf{w}_k] \\ &\quad + E[\mathbf{w}_k\mathbf{w}_k^H] \end{aligned} \tag{6.12}$$

The assumption made that the noise is white means that $\mathbf{e}_{k|k-1}$ is uncorrelated with $\mathbf{w}(k)$ and so we can simplify the above expression as

$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^H + \mathbf{Q}_{ww,k} \tag{6.13}$$

In order to obtain a recursive expression for $\mathbf{P}_{k|k}$ (used in obtaining $\mathbf{P}_{k|k-1}$), observe that from (6.8) the matrix $\mathbf{P}_k$ can be written as

$$\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1}\mathbf{H}^H\mathbf{K}_k^H + \mathbf{K}_k\mathbf{S}_k\mathbf{K}_k^H \tag{6.14}$$

Substituting for $\mathbf{K}_k$ in the above, the last two terms on the right cancel out, giving

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k\mathbf{H}\mathbf{P}_{k|k-1} \tag{6.15}$$

This completes the derivation of the strictly linear quaternion Kalman filter, which, unlike the existing approaches, is derived directly in the quaternion domain, and is given in Algorithm 1.

6.4 Widely linear quaternion Kalman filter

The Kalman filter derived in the previous section is optimal for strictly linear models or for circular data. To account for the noncircularity associated with real world data, the Kalman filter must propagate the augmented state estimates and augmented covariance matrix. Following on the widely linear model in (2.2.3) and the state update equation in (6.2), to incorporate the widely linear model we can rewrite the state and observation equations as

Algorithm 1: The Quaternion Kalman Filter (QKF)
Model output

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k \tag{6.16}$$
$$\mathbf{P}_{k|k-1} = \mathbf{A}_k\mathbf{P}_{k-1|k-1}\mathbf{A}_k^H + \mathbf{Q}_{ww,k} \tag{6.17}$$

Measurement output

$$\mathbf{S}_k = \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^H + \mathbf{R}_{vv,k} \tag{6.18}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^H\mathbf{S}_k^{-1} \tag{6.19}$$
$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}] \tag{6.20}$$
$$\mathbf{P}_{k|k} = [\mathbf{I} - \mathbf{K}_k\mathbf{H}]\mathbf{P}_{k|k-1} \tag{6.21}$$

$$\begin{aligned}\mathbf{x}_k^a &= \mathbf{A}_k^a\mathbf{x}_{k-1}^a + \mathbf{B}_k^a\mathbf{u}_k^a + \mathbf{w}_k^a \\ \mathbf{z}_k^a &= \mathbf{H}^a\mathbf{x}_{k-1}^a + \mathbf{v}_k^a\end{aligned} \tag{6.22}$$

where the augmented state vector $\mathbf{x}_k^a = [\mathbf{x}_k^T, \mathbf{x}_k^{\imath T}, \mathbf{x}_k^{\jmath T}, \mathbf{x}_k^{\kappa T}]^T$, the augmented state noise vector $\mathbf{w}_k^a = [\mathbf{w}_k^T, \mathbf{w}_k^{\imath T}, \mathbf{w}_k^{\jmath T}, \mathbf{w}_k^{\kappa T}]^T$, the augmented observation noise vector $\mathbf{v}_k^a = [\mathbf{v}_k^T, \mathbf{v}_k^{\imath T}, \mathbf{v}_k^{\jmath T}, \mathbf{v}_k^{\kappa T}]^T$,while the augmented state transition matrix and observation matrix are respectively given by

$$\begin{aligned}\mathbf{A}_k^a &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 \\ \mathbf{A}_2^{\imath} & \mathbf{A}_1^{\imath} & \mathbf{A}_4^{\imath} & \mathbf{A}_3^{\imath} \\ \mathbf{A}_3 & \mathbf{A}_4^{\jmath} & \mathbf{A}_1^{\jmath} & \mathbf{A}_2^{\jmath} \\ \mathbf{A}_4^{\kappa} & \mathbf{A}_3^{\kappa} & \mathbf{A}_2^{\kappa} & \mathbf{A}_4^{\kappa} \end{bmatrix} \\ \mathbf{H}^a &= \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{H}_4 \\ \mathbf{H}_2^{\imath} & \mathbf{H}_1^{\imath} & \mathbf{H}_4^{\imath} & \mathbf{H}_3^{\imath} \\ \mathbf{H}_3^{\jmath} & \mathbf{H}_4^{\jmath} & \mathbf{H}_1^{\jmath} & \mathbf{H}_2^{\jmath} \\ \mathbf{H}_4^{\kappa} & \mathbf{H}_3^{\kappa} & \mathbf{H}_2^{\kappa} & \mathbf{H}_1^{\kappa} \end{bmatrix}\end{aligned} \tag{6.23}$$

Note that the augmented block matrix structures in $\mathbf{H}^a$ and $\mathbf{A}_k^a$ (the subscript k is dropped in the partitioned matrix for ease of representation) are necessary for the model

in (6.22) to hold. Using the widely linear model described above, the widely linear quaternion Kalman filter algorithm becomes a generic extension of the quaternion Kalman filter described in (6.16) to (6.21), and provides a unified treatment of both the second order circular and noncircular data, with the subscript a emphasizing the fact that all vectors and matrices are augmented. The widely linear quaternion Kalman filter (WLQKF) is summarized in Algorithm 2.

Observe from (6.22) that the widely linear quaternion Kalman filter is not only able to model the noncircularity of the state model and observation model via the widely linear $\mathbf{A}_k^a$ and $\mathbf{H}^a$, but also the noncircularity of the state and observation noise via the augmented covariance matrices $\mathbf{Q}_{ww,k}^a$ and $\mathbf{R}_{vv,k}^a$.

Remark#23: Even for strictly linear state or observation model, the widely linear Kalman filter can still offer better performance than its strictly linear counterpart if the noise is noncircular. This is in contrast to the widely linear quaternion recursive least squares [121] which can only cater for the noncircularity of the model and not explicitly for the noise noncircularity.

Algorithm 2: The Widely Linear Quaternion Kalman Filter (WLQKF)
Model output

$$\hat{\mathbf{x}}_{k|k-1}^a = \mathbf{A}_k^a \hat{\mathbf{x}}_{k-1|k-1}^a + \mathbf{B}_k^a \mathbf{u}_k^a \tag{6.24}$$
$$\mathbf{P}_{k|k-1}^a = \mathbf{A}_k^a \mathbf{P}_{k-1|k-1}^a \mathbf{A}_k^{aH} + \mathbf{Q}_{ww,k}^a \tag{6.25}$$

Measurement output

$$\mathbf{S}_k^a = \mathbf{H}^a \mathbf{P}_{k|k-1}^a \mathbf{H}^{aH} + \mathbf{R}_{vv,k}^a \tag{6.26}$$
$$\mathbf{K}_k^a = \mathbf{P}_{k|k-1}^a \mathbf{H}^{aH} \mathbf{S}_k^{a^{-1}} \tag{6.27}$$
$$\hat{\mathbf{x}}_{k|k}^a = \hat{\mathbf{x}}_{k|k-1}^a + \mathbf{K}_k^a [\mathbf{z}_k^a - \mathbf{H}^a \hat{\mathbf{x}}_{k|k-1}^a] \tag{6.28}$$
$$\mathbf{P}_k^a = [\mathbf{I} - \mathbf{K}_k^a \mathbf{H}^a] \mathbf{P}_{k|k-1}^a \tag{6.29}$$

6.5 An efficient implementation of the WLQKF

Observe that matrices $\mathbf{A}_k^a$ and $\mathbf{H}_k^a$ in (6.23) are overparameterized since only one quarter of the entries are needed to fully describe each matrix. We shall next make use of this

property to simplify each of the six equations describing the quaternion widely linear Kalman filter.

Equation (6.24)

$$\hat{\mathbf{x}}^a_{k|k-1} = \mathbf{A}^a_k \hat{\mathbf{x}}^a_{k-1|k-1} + \mathbf{B}^a_k \mathbf{u}^a_k \tag{6.30}$$

Upon partitioning each matrix and array into its components we have

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \hat{\mathbf{x}}^\imath_{k|k-1} \\ \hat{\mathbf{x}}^\jmath_{k|k-1} \\ \hat{\mathbf{x}}^\kappa_{k|k-1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1k} & \mathbf{A}_{2k} & \mathbf{A}_{3k} & \mathbf{A}_{4k} \\ \mathbf{A}^\imath_{2k} & \mathbf{A}^\imath_{1k} & \mathbf{A}^\imath_{4k} & \mathbf{A}^\imath_{3k} \\ \mathbf{A}^\jmath_{3k} & \mathbf{A}^\jmath_{4k} & \mathbf{A}^\jmath_{1k} & \mathbf{A}^\jmath_{2k} \\ \mathbf{A}^\kappa_{4k} & \mathbf{A}^\kappa_{3k} & \mathbf{A}^\kappa_{2k} & \mathbf{A}^\kappa_{1k} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_{k-1|k-1} \\ \hat{\mathbf{x}}^\imath_{k-1|k-1} \\ \hat{\mathbf{x}}^\jmath_{k-1|k-1} \\ \hat{\mathbf{x}}^\kappa_{k-1|k-1} \end{bmatrix} \tag{6.31}$$

Observe that there is only one linearly independent row above, and so Equation (6.24) can be simplified into

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{A}_{1k}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{A}_{2k}\hat{\mathbf{x}}^\imath_{k-1|k-1} + \mathbf{A}_{3k}\hat{\mathbf{x}}^\jmath_{k-1|k-1} \\ &\quad + \mathbf{A}_{4k}\hat{\mathbf{x}}^\kappa_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k \\ &= \mathbf{A}^{\alpha T}_k \hat{\mathbf{x}}^a_{k-1|k-1} + \mathbf{B}_k\mathbf{u}_k \end{aligned} \tag{6.32}$$

where $\mathbf{A}^{\alpha T}_k = [\mathbf{A}_{1k}, \mathbf{A}_{2k}, \mathbf{A}_{3k}, \mathbf{A}_{4k}]$.

Remark#24: The advantage of using this representation is twofold. The output has only one quarter of the terms and so less memory is needed in the implementation (this is also the case for matrix $\mathbf{A}^\alpha_k$ as compared to $\mathbf{A}^a_k$). Secondly, this form is four times less computationally intensive.

Equation (6.25)

$$\mathbf{P}^a_{k|k-1} = \mathbf{A}^a_k \mathbf{P}^a_{k-1|k-1} \mathbf{A}^{aH}_k + \mathbf{Q}^a_{ww,k} \tag{6.33}$$

To simplify the above, we start from the definition of $\mathbf{P}^a_{k|k-1}$, that is

$$\mathbf{P}^a_{k|k-1} = E[\mathbf{e}^a_{k|k-1}\mathbf{e}^{aH}_{k|k-1}] = E\left[\begin{bmatrix}\mathbf{e}_{k|k-1}\\ \mathbf{e}^{\imath}_{k|k-1}\\ \mathbf{e}^{\jmath}_{k|k-1}\\ \mathbf{e}^{\kappa}_{k|k-1}\end{bmatrix}\begin{bmatrix}\mathbf{e}_{k|k-1}\\ \mathbf{e}^{\imath}_{k|k-1}\\ \mathbf{e}^{\jmath}_{k|k-1}\\ \mathbf{e}^{\kappa}_{k|k-1}\end{bmatrix}^H\right]$$
$$= \begin{bmatrix}\mathbf{P}_{1k|k-1} & \mathbf{P}_{2k|k-1} & \mathbf{P}_{3k|k-1} & \mathbf{P}_{4k|k-1}\\ \mathbf{P}^{\imath}_{2k|k-1} & \mathbf{P}^{\imath}_{1k|k-1} & \mathbf{P}^{\imath}_{4k|k-1} & \mathbf{P}^{\imath}_{3k|k-1}\\ \mathbf{P}^{\jmath}_{3k|k-1} & \mathbf{P}^{\jmath}_{4k|k-1} & \mathbf{P}^{\jmath}_{1k|k-1} & \mathbf{P}^{\jmath}_{2k|k-1}\\ \mathbf{P}^{\kappa}_{4k|k-1} & \mathbf{P}^{\kappa}_{3k|k-1} & \mathbf{P}^{\kappa}_{2k|k-1} & \mathbf{P}^{\kappa}_{1k|k-1}\end{bmatrix} \tag{6.34}$$

where $\mathbf{P}_{1k|k-1} = E[\mathbf{e}_{k|k-1}\left(\mathbf{e}_{k|k-1}\right)^H]$, $\mathbf{P}_{2k|k-1} = E[\mathbf{e}_{k|k-1}\left(\mathbf{e}_{k|k-1}\right)^{\imath H}]$, $\mathbf{P}_{3k|k-1} = E[\mathbf{e}_{k|k-1}\left(\mathbf{e}_{k|k-1}\right)^{\jmath H}]$, $\mathbf{P}_{4k|k-1} = E[\mathbf{e}_{k|k-1}\left(\mathbf{e}_{k|k-1}\right)^{\kappa H}]$.

Note that the product of two matrices with the augmented block matrix structure as above also has an augmented block matrix structure. Thus, only the first row of the product need to be computed to express the whole matrix. To formalise this product, we introduce the operator $f_{wl}(\cdot)$ which reconstructs a block matrix from its first row. The terms comprising $\mathbf{P}^a_{k|k-1}$ can now be calculated as

$$\mathbf{P}^{\alpha}_{k|k-1} = \mathbf{A}^{\alpha}_k f_{wl}\left(\mathbf{P}^{\alpha}_{k-1|k-1}\mathbf{A}^{aH}_k\right) + \mathbf{Q}^{\alpha}_{ww,k} \tag{6.35}$$

where $\mathbf{P}^{\alpha}_{k|k-1} = [\mathbf{P}_{1k|k-1}, \mathbf{P}_{2k|k-1}, \mathbf{P}_{3k|k-1}, \mathbf{P}_{4k|k-1}]$ and $\mathbf{Q}^{\alpha}_{ww,k} = [\mathbf{Q}_{ww,1k}, \mathbf{Q}_{ww,2k}, \mathbf{Q}_{ww,3k}, \mathbf{Q}_{ww,4k}]$

Remark#25: This form requires a quarter of the mathematical operations and only one quarter of the terms need be stored.

Equation (6.26)

$$\mathbf{S}^a_k = \mathbf{H}^a\mathbf{P}^a_{k|k-1}\mathbf{H}^{aH} + \mathbf{R}^a_{vv,k} \tag{6.36}$$

Following the same approach as that for (6.25), since $\mathbf{S}^a$ assumes the form

$$\begin{bmatrix} \mathbf{S}_{1k} & \mathbf{S}_{2k} & \mathbf{S}_{3k} & \mathbf{S}_{4k} \\ \mathbf{S}_{2k}^{\imath} & \mathbf{S}_{1k}^{\imath} & \mathbf{S}_{4k}^{\imath} & \mathbf{S}_{3k}^{\imath} \\ \mathbf{S}_{3k}^{\jmath} & \mathbf{S}_{4k}^{\jmath} & \mathbf{S}_{1k}^{\jmath} & \mathbf{S}_{2k}^{\jmath} \\ \mathbf{S}_{4k}^{\kappa} & \mathbf{S}_{3k}^{\kappa} & \mathbf{S}_{2k}^{\kappa} & \mathbf{S}_{1k}^{\kappa} \end{bmatrix} \tag{6.37}$$

this allows us to write

$$\mathbf{S}_k^{\alpha} = \mathbf{H}^{\alpha} f_{wl}\left(\mathbf{P}_{k|k-1}^{\alpha}\mathbf{H}^{aH}\right) + \mathbf{R}_{vv,k}^{\alpha} \tag{6.38}$$

where $\mathbf{S}_k^{\alpha} = [\mathbf{S}_{1k}, \mathbf{S}_{2k}, \mathbf{S}_{3k}\ \mathbf{S}_{4k}]$, $\mathbf{H}^{\alpha} = [\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{H}_4]$ and $\mathbf{R}_{vv,k}^{\alpha} = [\mathbf{R}_{vv,1k}, \mathbf{R}_{vv,2k}, \mathbf{R}_{vv,3k}, \mathbf{R}_{vv,4k}]$.

Remark#26: This form requires a quarter of the mathematical operations and only one quarter of the memory requirement.

Equation (6.27)

$$\mathbf{K}_k^a = \mathbf{P}_{k|k-1}^a \mathbf{H}^{aH} \mathbf{S}_k^{a^{-1}} \tag{6.39}$$

Knowing that $\mathbf{K}^a$ has an augmented block matrix structure we can follow the same approach as above to arrive at a more compact form given by

$$\mathbf{K}_k^{\alpha} = \mathbf{P}_{k|k-1}^{\alpha} f_{wl}\left(\mathbf{H}^{\alpha H} \mathbf{S}_k^{a^{-1}}\right)$$

where $\mathbf{K}_k^{\alpha} = [\mathbf{K}_{1k}, \mathbf{K}_{2k}, \mathbf{K}_{3k}, \mathbf{K}_{4k}]$.

The matrix inverse operation is computationally very intensive, however we can take advantage of the structure of matrix $\mathbf{S}^a$ to simplify this expression. The inverse of $\mathbf{S}$

has the same structure as $\mathbf{S}$, and can therefore be written as

$$\mathbf{S}_k^{a^{-1}} = \mathbf{C}^a = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{C}_3 & \mathbf{C}_4 \\ \mathbf{C}_2^{\iota} & \mathbf{C}_1^{\iota} & \mathbf{C}_4^{\iota} & \mathbf{C}_3^{\iota} \\ \mathbf{C}_3^{\jmath} & \mathbf{C}_4^{\jmath} & \mathbf{C}_1^{\jmath} & \mathbf{C}_2^{\jmath} \\ \mathbf{C}_4^{\kappa} & \mathbf{C}_3^{\kappa} & \mathbf{C}_2^{\kappa} & \mathbf{C}_1^{\kappa} \end{bmatrix} \tag{6.40}$$

By applying the block inversion lemma twice onto the matrix $\mathbf{S}$, the matrices $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$ and $\mathbf{C}_4$ can be evaluated as

$$\mathbf{C}_1 = (\mathbf{U}_1 - \mathbf{V}\mathbf{U}_3)^{-1} \tag{6.41}$$

$$\mathbf{C}_2 = -\mathbf{C}_1 V \tag{6.42}$$

$$\mathbf{C}_3 = -(\mathbf{C}_1\mathbf{O}_1 + \mathbf{C}_2\mathbf{O}_3) \tag{6.43}$$

$$\mathbf{C}_4 = -(\mathbf{C}_1\mathbf{O}_2 + \mathbf{C}_1\mathbf{O}_4) \tag{6.44}$$

where,

$$\mathbf{V} = \mathbf{U}_2\mathbf{U}_4^{-1} \tag{6.45}$$

$$\begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \\ \mathbf{U}_3 & \mathbf{U}_4 \end{bmatrix} = \mathbf{S}_a - \mathbf{O}\mathbf{S}_c \tag{6.46}$$

$$\begin{bmatrix} \mathbf{O}_1 & \mathbf{O}_2 \\ \mathbf{O}_3 & \mathbf{O}_4 \end{bmatrix} = \mathbf{O} = \mathbf{S}_b\mathbf{S}_d^{-1} \tag{6.47}$$

$$\begin{bmatrix} \mathbf{S}_a & \mathbf{S}_b \\ \mathbf{S}_c & \mathbf{S}_d \end{bmatrix} = \mathbf{S}_k^a \tag{6.48}$$

Remark#27: The above approach to obtaining the matrix inverse requires half the operations needed for inverting the whole of matrix $\mathbf{S}^a$.

Equation (6.28)

$$\hat{\mathbf{x}}^a_{k|} = \hat{\mathbf{x}}^a_{k|k-1} + \mathbf{K}^a_k[\mathbf{z}^a_k - \mathbf{H}^a\hat{\mathbf{x}}^a_{k|k-1}] \tag{6.49}$$

This can be simplified to

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}^\alpha_k(\mathbf{z}^a - \mathbf{H}^a\mathbf{x}^a_k) \tag{6.50}$$

where $\mathbf{K}^\alpha_k = [\mathbf{K}_{1k}, \mathbf{K}_{2k}, \mathbf{K}_{3k}, \mathbf{K}_{4k}]$.

Equation 6.29

$$\mathbf{P}^a_{k|k} = [\mathbf{I}^a - \mathbf{K}^a_k\mathbf{H}^a]\mathbf{P}^a_{k|k-1} \tag{6.51}$$

We can simplify this to

$$\mathbf{P}^\alpha_{k|k} = [[\mathbf{I}, \mathbf{0}, \mathbf{0}, \mathbf{0}] - \mathbf{K}^\alpha_k\mathbf{H}^a]\mathbf{P}^a_{k|k-1} \tag{6.52}$$

Algorithm 3 summarizes the efficient implementation of the quaternion widely linear Kalman filter.

6.6 Widely linear quaternion extended Kalman filter

For nonlinear systems, the state transition and the observation equations are nonlinear and are modeled using the extended Kalman filter, given by

$$\mathbf{x}_k = f[\mathbf{x}_{k-1}] + \mathbf{w}_k \tag{6.67}$$

$$\mathbf{z}_k = h[\mathbf{x}_k] + \mathbf{v}_k \tag{6.68}$$

where $f[\cdot]$ and $h[\cdot]$ are respectively the nonlinear state and observation functions. For their approximations, the extended Kalman filter uses the first order Taylor series expansions

(TSE) around the current state estimate. Calculating the quaternion derivative requires the function to be analytic in the CRF sense, however, this also imposes very strict conditions as only linear functions are analytic in the quaternion domain. Although nonlinear functions are not differentiable in the Cauchy-Riemann-Fueter sense, by exploiting the isomorphism between the quaternion domain $\mathbb{H}$ and real domain $\mathbb{R}^4$, the $\mathbb{HR}$-calculus allows for non-analytic functions to be differentiated [111]. This also guarantees the existence of the first order quaternion Taylor series expansion of a non-analytic quaternion function $f(q)$ about a point q_o, which assumes the form (see Appendix D)

$$f(q_o + dq) = f(q_o) + \frac{\partial f(q_o)}{\partial q} dq + \frac{\partial f(q_o)}{\partial q^{\imath}} dq^{\imath} + \frac{\partial f(q_o)}{\partial q^{\jmath}} dq^{\jmath} + \frac{\partial f(q_o)}{\partial q^{\kappa}} dq^{\kappa} \tag{6.69}$$

where for analytic functions the involution derivatives vanish.

Remark#28: Because quaternion nonlinear functions are nonanalytic (a consequence of CRF condition), the first order Taylor series in (6.69) will always be widely linear, due to the presence of the terms $dq^{\imath}$, $dq^{\jmath}$ and dq^{κ}. Therefore the quaternion extended Kalman filter is always widely linear. This is in contrast to the complex domain where the extended Kalman filter can be strictly linear if the nonlinear function is analytic [136].

Upon expanding $f[\cdot]$ around $\hat{\mathbf{x}}_{k-1|k-1}$ and $h[\cdot]$ around $\hat{\mathbf{x}}_{k|k-1}$ using the first order Taylor series in (6.69), the state and measurement equations become

$$\begin{aligned}
\mathbf{x}_k &= \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{x}^{\imath}_{k-1} + \mathbf{C}_k \mathbf{x}^{\jmath}_{k-1} + \mathbf{D}_k \mathbf{x}^{\kappa}_{k-1} + \mathbf{w}_k + \mathbf{d}_k \\
\mathbf{y}_k &= \mathbf{E}\mathbf{x}_{k-1} + \mathbf{F}\mathbf{x}^{\imath}_{k-1} + \mathbf{G}\mathbf{x}^{\jmath}_{k-1} + \mathbf{H}\mathbf{x}^{\kappa}_{k-1} + \mathbf{w}_k + \mathbf{l}_k
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{d}_k &= f(\hat{\mathbf{x}}_{k-1|k-1}) - \mathbf{A}_k\hat{\mathbf{x}}_{k-1|k-1} - \mathbf{B}_k\hat{\mathbf{x}}^{\imath}_{k-1|k-1} \\
&\quad -\mathbf{C}_k\hat{\mathbf{x}}^{\jmath}_{k-1|k-1} - \mathbf{D}_k\hat{\mathbf{x}}^{\kappa}_{k-1|k-1} \\
\mathbf{l}_k &= h(\hat{\mathbf{x}}_{k|k-1}) - \mathbf{E}\hat{\mathbf{x}}_{k|k-1} - \mathbf{F}\hat{\mathbf{x}}^{\imath}_{k|k-1} - \mathbf{G}\hat{\mathbf{x}}^{\jmath}_{k|k-1} \\
&\quad -\mathbf{H}\hat{\mathbf{x}}^{\kappa}_{k|k-1} \\
\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D} &= \frac{\partial f}{\partial x^{\varphi}_{k-1}}|_{\mathbf{x}^{\varphi}_{k-1}=\hat{\mathbf{x}}^{\varphi}_{k-1|k-1}} \text{ for } \varphi \in \{1,\imath,\jmath,\kappa\} \\
\mathbf{E},\mathbf{F},\mathbf{G},\mathbf{H} &= \frac{\partial h}{\partial x^{\varphi}_{k}}|_{\mathbf{x}^{\varphi}_{k}=\hat{\mathbf{x}}^{\varphi}_{k|k-1}} \text{ for } \varphi \in \{1,\imath,\jmath,\kappa\}
\end{aligned}$$

Observing that both the linearized process and observation equations are widely linear, the state and observation model of the extended Kalman filter can now be written as

$$\mathbf{x}_k = \mathbf{F}^a_k\mathbf{x}^a_{k-1}\mathbf{w}^a_k + \mathbf{d}^a_k \tag{6.70}$$

$$\mathbf{y}_k = \mathbf{H}^a\mathbf{x}^a_{k-1} + \mathbf{w}^a_k + \mathbf{l}^a_k \tag{6.71}$$

where $\mathbf{d}^a_k = [\mathbf{d}^T_k, \mathbf{d}^{\imath T}_k, \mathbf{d}^{\jmath T}_k, \mathbf{d}^{\kappa T}_k]^T$, $\mathbf{l}^a_k = [\mathbf{l}^T_k, \mathbf{l}^{\imath T}_k, \mathbf{l}^{\jmath T}_k, \mathbf{l}^{\kappa T}_k]^T$

$$\mathbf{F}^a_k = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k & \mathbf{C}_k & \mathbf{D}_k \\ \mathbf{B}^{\imath}_k & \mathbf{A}^{\imath}_k & \mathbf{D}^{\imath}_k & \mathbf{C}^{\imath}_k \\ \mathbf{C}^{\jmath}_k & \mathbf{D}^{\jmath}_k & \mathbf{A}^{\jmath}_k & \mathbf{B}^{\jmath}_k \\ \mathbf{D}^{\kappa}_k & \mathbf{C}^{\kappa}_k & \mathbf{B}^{\kappa}_k & \mathbf{A}^{\kappa}_k \end{bmatrix} \tag{6.72}$$

$$\mathbf{H}^a = \begin{bmatrix} \mathbf{E} & \mathbf{F} & \mathbf{G} & \mathbf{H} \\ \mathbf{F}^{\imath} & \mathbf{E}^{\imath} & \mathbf{H}^{\imath} & \mathbf{G}^{\imath} \\ \mathbf{G}^{\jmath} & \mathbf{H}^{\jmath} & \mathbf{E}^{\jmath} & \mathbf{F}^{\jmath} \\ \mathbf{H}^{\kappa} & \mathbf{G}^{\kappa} & \mathbf{F}^{\kappa} & \mathbf{E}^{\kappa} \end{bmatrix} \tag{6.73}$$

The widely linear quaternion extended Kalman filter is summarized in Algorithm 4.

6.7 Duality between the quaternion and real valued Kalman filter

Following on the definition of quaternions in (2.3) a one-to-one mapping exists between the points in $\mathbb{H}$ and $\mathbb{R}^4$. For the quaternion vector $\mathbf{q} = \mathbf{q}_r + \imath\mathbf{q}_\imath + \jmath\mathbf{q}_\jmath + \kappa\mathbf{q}_\kappa \in \mathbb{H}^n$, this duality is described by

$$\underbrace{\begin{bmatrix} \mathbf{q} \\ \mathbf{q}^\imath \\ \mathbf{q}^\jmath \\ \mathbf{q}^\kappa \end{bmatrix}}_{\mathbf{q}^a \in \mathbb{H}^{4n}} = \underbrace{\begin{bmatrix} \mathbf{I} & \imath\mathbf{I} & \jmath\mathbf{I} & \kappa\mathbf{I} \\ \mathbf{I} & \imath\mathbf{I} & -\jmath\mathbf{I} & -\kappa\mathbf{I} \\ \mathbf{I} & -\imath\mathbf{I} & \jmath\mathbf{I} & -\kappa\mathbf{I} \\ \mathbf{I} & -\imath\mathbf{I} & -\jmath\mathbf{I} & \kappa\mathbf{I} \end{bmatrix}}_{\mathbf{J}_n \in \mathbb{H}^{n\times n}} \underbrace{\begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_i \\ \mathbf{q}_j \\ \mathbf{q}_k \end{bmatrix}}_{\mathbf{q}^r \in \mathbb{R}^{4n}} \tag{6.80}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{J}$ describes the invertible mapping $\mathbf{J} : \mathbb{R}^4 \to \mathbb{H}$ (where $\mathbf{J}^{-1} = \frac{1}{4}\mathbf{J}^H$). Using this isomorphism, we can obtain a dual real quadrivariate version of the augmented quaternion state and measurement model in (6.22), in the form

$$\begin{aligned} \mathbf{x}_k^r &= \mathbf{A}_k^r \mathbf{x}_{k-1}^r + \mathbf{B}_k^r \mathbf{u}_k^r + \mathbf{w}_k^r \\ \mathbf{z}_k^r &= \mathbf{H}^r \mathbf{x}_{k-1}^r + \mathbf{v}_k^r \end{aligned} \tag{6.81}$$

where $\mathbf{x}_k^r = \mathbf{J}_n^{-1}\mathbf{x}_k^a$, $\mathbf{z}_k^r = \mathbf{J}_m^{-1}\mathbf{z}_k^a$, $\mathbf{F}_k^r = \mathbf{J}_n^{-1}\mathbf{F}_k^a\mathbf{J}_n$, $\mathbf{H}^r = \mathbf{J}_m^{-1}\mathbf{H}^a\mathbf{J}_n$, $\mathbf{w}_k^r = \mathbf{J}_n^{-1}\mathbf{w}_k^a$, $\mathbf{v}_k^r = \mathbf{J}_n^{-1}\mathbf{v}_k^a$ while, the error covariance matrix are related by $\mathbf{P}_k^r = \mathbf{J}_n^{-1}\mathbf{P}_k^a\mathbf{J}_n^{-H}$. We can now obtain an expression relating the MSEs of the quaternion and real valued Kalman filter in the form

$$\begin{aligned} Tr(\mathbf{P}_k^r) &= Tr(\mathbf{J}_n^{-1}\mathbf{P}_k^a\mathbf{J}_n^{-H}) \\ &= Tr(\mathbf{P}_k^a\mathbf{J}_n^{-H}\mathbf{J}_n^{-1}) \\ &= \frac{1}{4}Tr(\mathbf{P}_k^a) \end{aligned} \tag{6.82}$$

Due to the redundant information in the augmented pseudocovariance matrix $\mathbf{P}_k^a$

(see (6.34)), the MSE for the widely linear Kalman filter is $\frac{1}{4}Tr(\mathbf{P}_k^a)$ and not $Tr(\mathbf{P}_k^a)$, thus giving the same MSE as the real quadrivariate Kalman filter.

Remark#29: It follows from (6.82) that the performance of the widely linear quaternion Kalman filter is identical to that of the real Kalman filter. The duality between $\mathbb{H}$ and $\mathbb{R}^4$ only exists for widely linear models, reflecting the fact that strictly linear models are suboptimal, and are thus suitable only for a very restrictive class of circular data.

Remark#30: When the underlying signal generating model is strictly linear, both the real Kalman filter and WLQKF are over-parameterized and the quaternion Kalman filter provides better convergence performance.

6.8 Simulations

To illustrate the advantages offered by widely linear quaternion Kalman filters over their strictly linear counterparts, we considered the following scenarios: 1) filtering of a quaternion valued signal from noisy measurements where the state model is known and the first and second order statistics of both state and measurement noise are also known; 2) multiple step ahead prediction of a nonstationary three dimensional wind field; 3) estimating the position of a target using bearings only measurements.

6.8.1 Filtering of an autoregressive process

In the first experiment, the state model was described by a fourth order quaternion autoregressive process, AR(4), given by

$$x_k = 1.4x_{k-1} - 0.7x_{k-1} + 0.04x_{k-1} - 0.05x_{k-1} + w_k \tag{6.83}$$

where the driving noise w_k was quadruply white zero mean Gaussian noise with the variance and pseudovariance defined as

$$E[w_k w_n^*] = \alpha\delta_{k-n} \qquad E[w_k w_n^{\imath *}] = \beta\delta_{k-n}$$
$$E[w_k w_n^{\jmath *}] = \gamma\delta_{k-n} \qquad E[w_k w_n^{\kappa *}] = \eta\delta_{k-n}$$

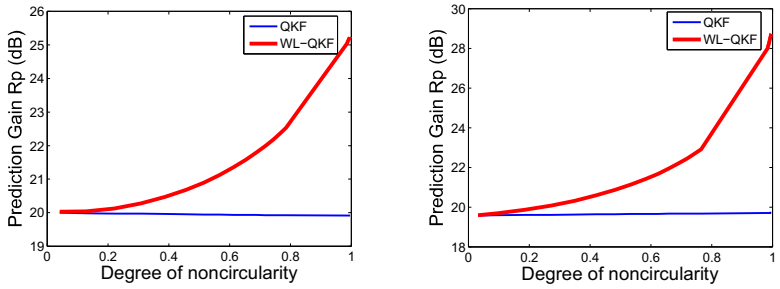
where δ is the discrete Dirac delta function. In the first experiment the state x_k was observed in the presence of additive quaternion valued white noise v_n of varying degree of circularity[2], that is

$$z_k = x_k + v_k \tag{6.84}$$

In the second experiment the state x_k was observed through a nonlinearity, given by

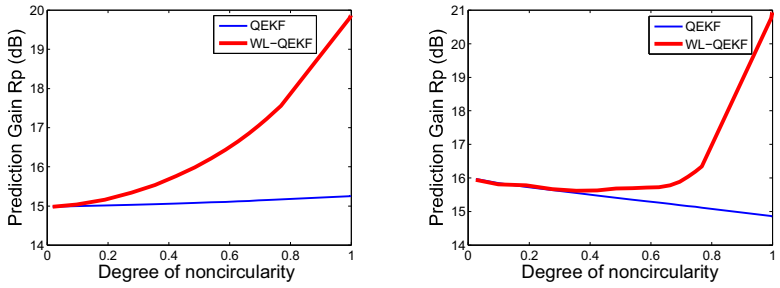
$$z_k = arctan(x_k) + w_k \tag{6.85}$$

The quantitative performance measure was the prediction gain $R_p = 10log(\sigma_y^2/\sigma_e^2)$. where σ_y^2 and σ_e^2 are respectively the powers of the input signal and the output error.



(a) Noncircular observation noise (b) Noncircular state noise

Figure 6.1: Performance comparison between QKF and WLQKF for the AR(4) process in (6.84), under varying degrees of noncircularity of the state and observation noise.



(a) Noncircular observation noise (b) Noncircular state noise

Figure 6.2: Performance comparison between QEKF and WLQEKF for the AR(4) process in (6.85), observed through output nonlinearity, for varying degrees of noncircularity of the state and observation noise.

[2]The noncircularity measure $r_s = \frac{\beta+\gamma+\eta}{3\alpha}$, where $r_s \in [0, 1]$ and the value $r_s = 0$ indicates a circular source while $r_s = 1$ indicates a highly noncircular source.

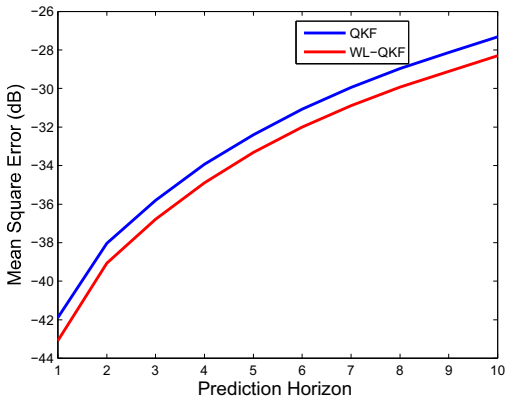


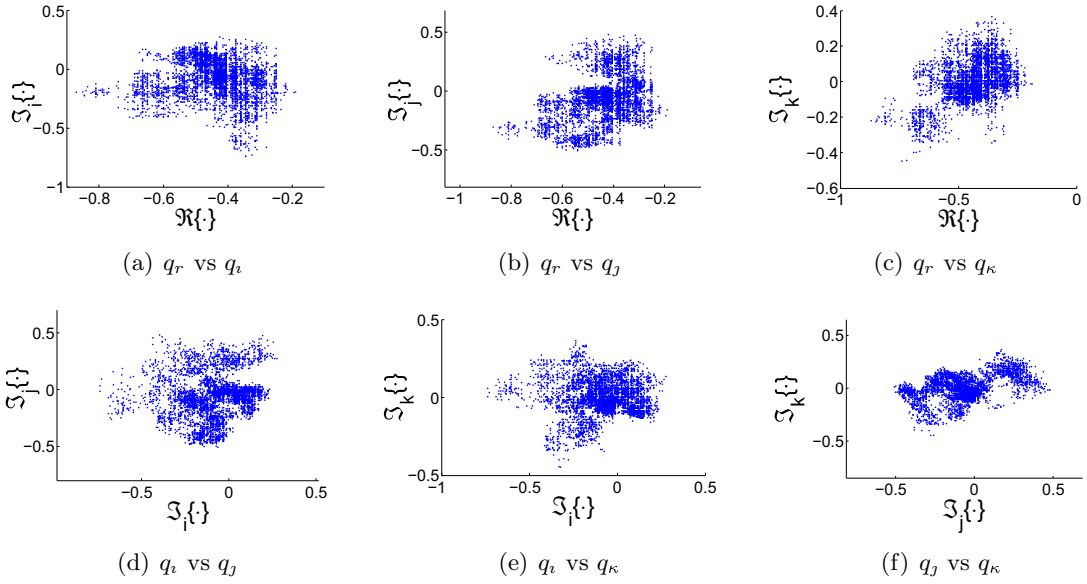
Figure 6.3: Multistep ahead prediction of real-world four dimensional wind data using QKF and WLQKF.

Figure 6.1 compares the performances of the QKF and its widely linear counterpart, the WLQKF, for the signal in (6.84). Figure 6.1(a) illustrates the results for a circular state noise and an observation noise of varying degree of a circularity, while in Figure 6.1(b) the observation noise was circular and the state noise noncircular. Conforming with the analysis, for both sets of experiments, when the noises were circular, the strictly linear and widely linear QKF had the same prediction gain. However, for noncircular noises, the QKF was inferior to the WLQKF, as it did not cater for the full second order statistics of the noises. This was also reflected by the performance of the QKF being independent of the noise noncircularity.

Figure 6.2 compares the performances of the QEKF and WLQEKF for the nonlinear signal in (6.85). Similarly to the results for the linear observation model in the previous experiment, we observe that the WLQEKF outperformed the QEKF when either the observation noise or state noise were noncircular. However, unlike the previous experiment, for noncircular state noise, the performance of the strictly linear QEKF was dependent on the degree of noncircularity.

6.8.2 Multistep ahead prediction

The performances of the QKF and WLQKF were next investigated for the multistep ahead prediction of a real world noncircular and nonstationary four-dimensional wind



(a) q_r vs q_i (b) q_r vs q_j (c) q_r vs q_κ

(d) q_i vs q_j (e) q_i vs q_κ (f) q_j vs q_κ

Figure 6.4: Geometric view of the wind signal noncircularity via the six scatter plots of quaternion components.

field[3]. Figure 6.3 illustrates the prediction performance in terms of the mean square error (MSE) for both the QKF and WLQKF. Observe that, owing to its ability to utilize all the second order information available, the WLQKF was able to capture the underlying dynamics of the wind signal better than the QKF. This confirms that the wind signal is inherently noncircular (as shown in Figure 6.4) and so should be processed using the widely linear Kalman filter. This is also true for most real world scenarios, including orientation tracking in aeronautics.

6.8.3 Bearings only tracking

In the bearings only tracking problem the aim is to estimate the position of an object based only on noisy bearings measurements of the object from one or more sensors. Bearings only tracking is useful in military applications, for example in submarine tracking by passive sonar. Since the range of the target must be estimated from bearings measurements, the problem is inherently nonlinear. To estimate the position (x_k,y_k,z_k) and velocity $(\dot{x}_k,\dot{y}_k,\dot{z}_k)$

[3]The wind speed measurements in the North, East, and vertical direction formed the imaginary part of the quaternion while the temperature was incorporated in the real part to form a full quaternion. The dataset was recorded using the WindMaster, a 3D Gill Instruments ultrasonic anemometer, which was resampled at 5Hz for simulation purposes.

of a target using a system of L observers located at $(x^0_{i,k}, y^0_{i,k}, z^0_{i,k})$ for $i = 1, ..., L$, the WLQEKF can be used with the state and observation model described as follows

- $\mathbf{x}_k = [\imath x_k + \jmath y_k + \kappa z_k, \imath \dot{x}_k + \jmath \dot{y}_k + \kappa \dot{z}_k]$
- Matrices $\mathbf{F}$ and $\mathbf{B}$ are defined as

$$\mathbf{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix}$$

  where T is the sampling interval.

- $\mathbf{y}_k$ are the noise corrupted bearing measurements and $\mathbf{h}[\mathbf{x}_k]$ is a nonlinear defined as

$$\mathbf{h}[\mathbf{x}_k] = \begin{bmatrix} \theta_1, k + \imath\theta_{\frac{L}{2}+1,k} + \jmath\phi_{1,k} + \kappa\phi_{\frac{L}{2}+1,k} \\ \theta_{2,k} + \imath\theta_{\frac{L}{2}+2,k} + \jmath\phi_{2,k} + \kappa\phi_{\frac{L}{2}+2,k} \\ \vdots \\ \theta_{\frac{L}{2},k} + \imath\theta_{L,k} + \jmath\phi_{\frac{L}{2},k} + \kappa\phi_{L,k} \end{bmatrix}$$

  where

$$\begin{aligned} \theta_{i,k} &= arctan\left(\frac{y_k - y^0_{i,k}}{x_k - x^0_{i,k}}\right) \\ \phi_{i,k} &= arctan\left(\frac{z_k - z^0_{i,k}}{((x_k - x^0_{i,k})^2 + (y_k - y^0_{i,k})^2)^{\frac{1}{2}}}\right) \end{aligned}$$

  are respectively the azimuth and elevation angles.

- $w_k = \ddot{x}_k + \jmath\ddot{x}_k + \kappa\ddot{x}_k$ is the zero mean noncircular state noise (accounting for the target accelerations) and $\mathbf{v}_k = [v_{1,k} v_{2,k} \dots v_{L,k}]$ is the zero mean

The function $\mathbf{h}[x_k]$ is nonholomorphic and therefore the first order approximation using the quaternion Taylor series is widely linear. To illustrate the necessity of the widely linear model we simulated a scenario where two static sensors are located at (-1200,1300,0) and (1000,1500,100). The sampling interval of $T = 0.1$ was used and the target was made to have an initial position of (200,100,300) and velocity (2,1,0.5). The statistics of the

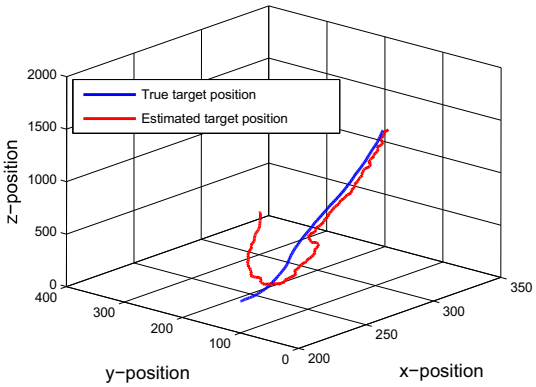
state and observation noise were as follows

$$\begin{aligned} w_k &= \mathcal{N}(0, 10, 10, 10, 10) \\ v_k &= \mathcal{N}(0, 0.001, 0.001, 0.001, 0.001) \end{aligned}$$

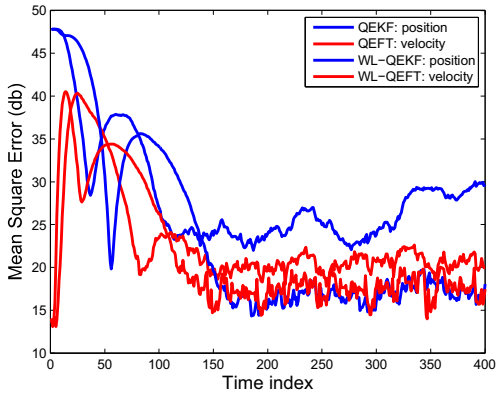
In the notation used above to describe the Gaussian distribution, the first two terms have their usual meaning, while the last tree terms are the pseudovariances of the noise. Figure 6.5(a) shows the trajectory of the target and the position estimated by the WLQEKF. The MSE shown in Figure 6.5(b) shows that the WLQEKF achieved a better MSE performance than the QEKF, for both the position and velocity estimation of the target. This conforms with Remark 28 where we showed that due to quaternion nonlinear functions always being nonholomorphic, the extended Kalman filter must always be widely linear since the first order quaternion Taylor series is also always widely linear.

6.9 Summary

The quaternion Kalman (QKF) filter and its widely linear counterpart (WLQKF) have been introduced for the processing of three- and four- dimensional signals in the quaternion domain. In this way, both second order circular (proper) and noncircular (improper) signals are catered for, giving a natural solution in a division algebra where the data reside, as opposed to the existing approaches which use the duality that exists between $\mathbb{H}$ and $\mathbb{R}^4$ to convert the quaternion quantities into real vectors. Both strictly linear and widely linear Kalman and extended Kalman filters have been developed, which was made possible by the recently introduced $\mathbb{HR}$-calculus. We have also illuminated that owing to the widely linear nature of the first order quaternion Taylor series, the quaternion extended Kalman filter is in general widely linear, an issue that has so far prevented its development. The proposed class of quaternion Kalman filters fully operate in the quaternion domain where the data reside, and are a generic extension of their real- and complex-valued counterparts, simplifying into the existing strictly and widely linear complex solutions when operating on two-dimensional data [136]. The duality with the quadrivariate real Kalman filters



(a) trajectory of an object tracked in a three dimensional space.



(b) MSE performance

Figure 6.5: Bearings only tracking using the widely linear quaternion extended Kalman filter.

and some computational issues have also been addressed. Simulations have shown that for a range of signals, for which either the state model is widely linear or the noises are noncircular, the WLQKF filter and the WLQEKF offer better performance over their strictly linear counterparts.

Algorithm 3: An efficient implementation of the quaternion widely linear Kalman filter

Model output

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k^{\alpha T}\hat{\mathbf{x}}_{k-1|k-1}^{a} \tag{6.53}$$

$$\mathbf{P}_{k|k-1}^{\alpha} = \mathbf{A}_k^{\alpha} f_{wl}\left(\mathbf{P}_{k-1|k-1}^{\alpha}\mathbf{A}^{aH}\right) + \mathbf{Q}_{ww,k}^{\alpha} \tag{6.54}$$

Measurement output

$$\mathbf{S}_k^{\alpha} = \mathbf{H}^{\alpha} f_{wl}\left(\mathbf{P}_{k|k-1}^{\alpha}\mathbf{H}^{aH}\right) + \mathbf{R}_{vv,k}^{\alpha} \tag{6.55}$$

$$\mathbf{K}^{\alpha} = \mathbf{P}_{k|k-1}^{\alpha} f_{wl}\left(\mathbf{H}^{\alpha H}\mathbf{C}^{a}\right) \tag{6.56}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k^{\alpha}(\mathbf{z}_k^{a} - \mathbf{H}^{a}\mathbf{x}_{k|k-1}^{a}) \tag{6.57}$$

$$\mathbf{P}_{k|k}^{\alpha} = [[\mathbf{I},\mathbf{0},\mathbf{0},\mathbf{0}] - \mathbf{K}_k^{\alpha}\mathbf{H}^{a}]\mathbf{P}_{k|k-1}^{a} \tag{6.58}$$

where

$$\mathbf{C}_1 = (\mathbf{U}_1 - \mathbf{U}_2\mathbf{U}_4^{-1}\mathbf{U}_3)^{-1} \tag{6.59}$$

$$\mathbf{C}_2 = -\mathbf{C}_1\mathbf{U}_2\mathbf{U}_4^{-1} \tag{6.60}$$

$$\mathbf{C}_3 = -(\mathbf{C}_1\mathbf{O}_1 + \mathbf{C}_2\mathbf{O}_3) \tag{6.61}$$

$$\mathbf{C}_4 = -(\mathbf{C}_1\mathbf{O}_2 + \mathbf{C}_1\mathbf{O}_4) \tag{6.62}$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \\ \mathbf{U}_3 & \mathbf{U}_4 \end{bmatrix} = \mathbf{S}_a - \mathbf{O}\mathbf{S}_c \tag{6.63}$$

$$\mathbf{O} = \begin{bmatrix} \mathbf{O}_1 & \mathbf{O}_2 \\ \mathbf{O}_3 & \mathbf{O}_4 \end{bmatrix} = \mathbf{S}_b\mathbf{S}_d^{-1} \tag{6.64}$$

while the superscript α in $\mathbf{M}^{\alpha}$ denotes the augmented array of matrix $\mathbf{M}$

$$\mathbf{M}^{\alpha} = [\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4] \tag{6.65}$$

The superscript a in $\mathbf{M}^{a}$ defines the augmented matrix of matrix $\mathbf{M}$, given by

$$\mathbf{M}^{a} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{M}_2 & \mathbf{M}_3 & \mathbf{M}_4 \\ \mathbf{M}_2^{\imath} & \mathbf{M}_1^{\imath} & \mathbf{M}_4^{\imath} & \mathbf{M}_3^{\imath} \\ \mathbf{M}_3^{\jmath} & \mathbf{M}_4^{\jmath} & \mathbf{M}_1^{\jmath} & \mathbf{M}_2^{\jmath} \\ \mathbf{M}_4^{\kappa} & \mathbf{M}_3^{\kappa} & \mathbf{M}_2^{\kappa} & \mathbf{M}_1^{\kappa} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_a & \mathbf{M}_b \\ \mathbf{M}_c & \mathbf{M}_d \end{bmatrix} \tag{6.66}$$

where $f_{wl}(\mathbf{M}^{\alpha}) = \mathbf{M}^{a}$ for any matrix $\mathbf{M}$.

Algorithm 4: The Widely Linear Quaternion Extended Kalman Filter

Model output

$$\hat{\mathbf{x}}^a_{k|k-1} = f^a_k[\hat{\mathbf{x}}^a_{k-1|k-1}] + \mathbf{B}^a_k \mathbf{u}^a_k \tag{6.74}$$

$$\mathbf{P}^a_{k|k-1} = \mathbf{F}^a_k \mathbf{P}^a_{k-1} \mathbf{F}^{aH}_k + \mathbf{Q}^a_{ww,k} \tag{6.75}$$

Measurement output

$$\mathbf{S}^a = \mathbf{H}^a \mathbf{P}^a_{k|k-1} \mathbf{H}^{aH} + \mathbf{R}^a_{vv,k} \tag{6.76}$$

$$\mathbf{K}^a = \mathbf{P}^a_{k|k-1} \mathbf{H}^{aH} \mathbf{S}^{a^{-1}} \tag{6.77}$$

$$\hat{\mathbf{x}}^a_{k|k} = \hat{\mathbf{x}}^a_{k|k-1} + \mathbf{K}^a[\mathbf{z}^a_k - h^a[\hat{\mathbf{x}}^a_{k|k-1}]] \tag{6.78}$$

$$\mathbf{P}^a_{k|k} = [\mathbf{I} - \mathbf{K}^a \mathbf{H}^a] \mathbf{P}^a_{k|k-1} \tag{6.79}$$

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we have developed a novel unified framework, consisting of the $\mathbb{HR}$-calculus, quaternion gradient and I-gradient, enabling the development of quaternion valued adaptive signal processing algorithms. Using this framework, the development of the $\mathbb{HR}$-calculus has proved invaluable, not only to simplify the derivation of algorithms, but also to enable the differentiation of non-analytic functions in $\mathbb{H}$, such as the real-valued cost functions which are often encountered in quaternion valued optimization problems. This is particularly important in the quaternion domain, where the CRF condition (equivalent to the Cauchy-Riemann condition in the complex domain) restricts differentiability to only constants and linear functions. Using this unified framework, we have derived a class of adaptive signal processing algorithms (based on both the least squares and stochastic gradient approach) suitable for the processing of the generality of quaternion-valued signals (both proper and improper) and evaluated their performances both analytically and experimentally, on synthetic and real world signals. The contributions in this thesis are summarised as follows

1. The development of the $\mathbb{HR}$-calculus, enabling the differentiation of non-analytic functions in the quaternion domain. Since all quaternion functions, other than constants and linear functions are non holomorphic, the $\mathbb{HR}$-calculus has proved to

be an invaluable tool. We have highlighted key differences with the $\mathbb{CR}$-calculus, including the fact that the $\mathbb{HR}$-calculus places a strict restriction on the type of dependent variables taken by the function that is differentiated. This has helped explain unusual results, such as $\frac{q^*}{q} = -\frac{1}{2}$, and simplify the derivation of algorithms.

2. The complex gradient has been extended to the quaternion domain. It has been shown that, like the complex domain, it is the conjugate gradient that gives the direction of steepest descent. To obtain generic extensions of complex valued algorithms in the quaternion domain, we also proposed the I-gradient, which makes use of the involution derivatives introduced by the $\mathbb{HR}$-calculus.

3. We have derived a generic extension of the CLMS (called the IQLMS) using the I-gradient. We have shown the equivalence between different expressions for the QLMS and the IQLMS. In addition to requiring half the numerical operations of the QLMS, analysis in the mean also showed that the IQLMS offers faster convergence rate and wider stability range. Analysis in the mean square of the IQLMS and WLIQLMS has demonstrated the advantage of the WLIQLMS over the IQLMS when the underlying signal generating model is widely linear. This was confirmed via simulations on both synthetic signals and a four-dimensional wind field.

4. To speed up the convergence of the QLMS for coloured input signals, we have derived the widely linear quaternion affine projection algorithm and the widely linear quaternion recursive least squares.

5. We have derived the quaternion Kalman filter and widely linear quaternion Kalman filter. We have taken advantage of the redundancy existing in the quaternion Kalman filter to derive an efficient implementation of the widely linear Kalman filter. Since Kalman filters are often used to track positions and orientations in three dimensional space, we have derived an efficient widely linear model suitable for processing three dimensional signals. We have also shown that the quaternion Taylor series is widely linear and hence the quaternion extended Kalman filter is also, always widely linear. Finally, we have shown how the widely linear quaternion extended Kalman filter can

be used for bearings only tracking.

7.2 Further work

There are numerous ways that the work carried out for this thesis can be further explored or extended, including

- *Efficient implementations of quaternion valued algorithms.* The quaternion widely linear model, like the complex widely linear model, is overparametrized. This is more so the case in the quaternion domain, since in the complex domain, half of the terms in the augmented covariance matrix are independent, while in the quaternion domain, it is only one quarter. Efficient implementation of complex valued algorithms have started to gain interest, particularly for the LMS and RLS [40], but further work remains to be done, particularly in terms of convergence analysis of the efficient implementations (Appendix H summarises the results of a recently submitted paper addressing some of these issues).

- *Extending the HR-calculus.* The HR-calculus consists of eight derivatives, six of which are with respect to involutions of a quaternion variable. Since involutions are rotations of 180 degrees about an axis of rotation (e.g. $q^\imath = -\imath q \imath$), it should be possible to generalise the HR-calculus to consider the derivative of functions with respect to the rotation of a quaternion variable around any axis of rotation, that is, $q^r = -rqr$. Appendix G gives the abstract of a recently submitted that proposes a generalization of the HR-calculus.

- *Extending other algorithms, such as the unscented Kalman filter, particle filters and distributed algorithms to the quaternion domain.* The extended Kalman filter allows the estimation of non linear state models by linearising all non linear models, thus allowing the Kalman filter equations to be applied. While widely used as an estimator for nonlinear state models [137], the linearisation can result in unstable filters as the assumption of local linearity breaks down and the Jacobian matrices can be difficult to calculate and implement. To overcome these problems, the unscented transfor-

mation can be used to obtain the unscented Kalman filter [138], which uses a set of carefully sampled points to parameterise the means and covariances of probability distributions. While computationally more intensive than the extended Kalman filter, the unscented Kalman filter does perform better for highly nonlinear state equations as it is able to capture higher order moments of distribution than linearisation. For highly non linear state models and non-Gaussian noises, the particle filter [139](a sequential Monte Carlo method that approximates the Bayesian posterior distribution in nonlinear and non-Gaussian envirnonments) is able to capture higher order moments and thus offer better performance than the unscented Kalman filter.

- *Distributed algorithms* [47] [51] could also benefit from the use of quaternions. Advances in sensor technology and wireless communication have led to a growing use of sensor networks in practical applications, ranging from environmental monitoring to target localization. These networks typically comprise a number of interconnected nodes which are able to communicate with one another in order to estimate some parameter of interest from noisy measurements. Within distributed solutions to estimation problems, every node in the network communicates only with a subset of the nodes, while processing is distributed among all nodes in the network. Since many networks use three or four dimensional sensors, the algorithms could be conveniently extended into the quaternion domain (see Appendix F).

- *Using octonions to extend algorithms to an eight dimensional space.* Octonions are a normed division algebra, forming an octet of real numbers. While quaternions are noncommutative, octonions are also nonassociative, making them harder to manipulate algebraically. For this reason they are less studied than complex and quaternion numbers, but have been found useful in quantum logic, special relativity, string theory and supersymmetry [140]. It is, however, the opinion of the author that the nonassociative nature of octonions is likely to negate any advantages obtained in working in a division algebra.

- *Applications.* While quaternions have been extensively used in graphic design to

perform three dimensional rotations, they are nevertheless, scarcely used in other applications. The mathematical framework and algorithms developed for this thesis open up the possibility to using quaternions in a wider range of applications, such as

1. Navigation: The three dimensional output of inertial navigation sensors can be conveniently processed using quaternions. Quaternions also prevent an undesired effect known as gimbal lock which can result when, for example in pitch/roll/yaw navigational systems, the roll is rotated 90 degrees to the right or left, so that pitch and yaw no longer move independently, and a degree of freedom is lost. In a gimbal-based inertial navigation system, this could result in serious consequences if, for example, the aircraft is in a steep dive or ascent.
2. Robotics: Movements of limbs on a robot usually happen in three dimensional space, and often require a complex sequence of rotations to perform a task. Quaternions are therefore ideally suited for such tasks, and have already attracted some interest [83–86].
3. Medical imaging and colour image processing: Quaternions are used in MRI (Magnetic Resonance Images) and CAT (Computed Axial Tomography). The MRI imaging technique employs a strong magnetic field in conjunction with radio frequency waves to produce cross-sectional slice images of the human body, while the CAT imaging technique utilizes x-rays. MRI and CAT scans perform rigid transformations in three dimensional space, that can be represented using quaternions [141]. Due to the three dimensional nature of colour, quaternions are also naturally suited to the processing of colour images [56] [57] [58].
4. Tracking of body sensor networks: Recent advances in sensing technology have brought to light data sources which are almost invariably three- and four-dimensional, such as measurements from inertial bodysensors. The quaternion domain offers a convenient means to process three- and four-dimensional signals under the same umbrella.

Bibliography

[1] B. Widrow and M. A. Lehr, "Noise canceling and channel equalization," *The Handbook of Brain Theory and Neural Networks*, 1995.

[2] N. Thakor and Y.-S. Zhu, "Applications of adaptive filtering to ecg analysis: Noise cancellation and arrhythmia detection," *IEEE Transactions on Biomedical Engineering*, vol. 38, no. 8, pp. 785–794, 1991.

[3] R. Plackett, "Some theorems in least squares," *Biometrika*, p. 1490157, Jun 1950.

[4] B. Widrow and S. D. Steards, *Adaptive Signal Processing*. Prentice Hall, 1985.

[5] B. Widrow and M. E. Hoff, "Adaptive Switching Circuits," in *WESCON Convention Record, Part 4*, pp. 96–104, IRE, 1960.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME Journal of Basic Engineering*, no. 82 (Series D), pp. 35–45, 1960.

[7] M. Grewal and A. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]," *IEEE Control Systems*, vol. 30, no. 3, pp. 69–78, 2010.

[8] D. Godard, "Channel equalization using a Kalman filter for fast data transmission," *IBM Journal of Research and Development*, vol. 18, no. 3, pp. 267–273, 1974.

[9] A. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18–60, 1994.

[10] L. Ljung, M. Morf, and D. Falconer, "Fast calculation of gain matrices for recursive estimation schemes," *International Journal of Control*, vol. 27, no. 1, pp. 1–19, 1978.

[11] M. Morf, T. Kailath, and L. Ljung, "Fast algorithms for recursive identification," in *Proceedings of the 1976 IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes,*, vol. 15, pp. 916–921, 1976.

[12] D. Falconer and L. Ljung, "Application of fast Kalman estimation to adaptive equalization," *IEEE Transactions on Communications*, vol. 26, no. 10, pp. 1439–1446, 1978.

[13] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE Transactions on Automatic Control,*, vol. 12, no. 3, pp. 282–287, 1967.

[14] M. Tarrab and A. Feuer, "Convergence and performance analysis of the normalized LMS algorithm with uncorrelated gaussian data," *IEEE Transactions on Information Theory*, vol. 34, no. 4, pp. 680–691, 1988.

[15] R. Bitmead and B. D. O. Anderson, "Performance of adaptive estimation algorithms in dependent random environments," *IEEE Transactions on Automatic Control,*, vol. 25, no. 4, pp. 788–794, 1980.

[16] N. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 793–806, 1986.

[17] A. H. Sayed, *Fundamentals of adaptive filtering.* Wiley-Blackwell, 2003.

[18] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communnications in Japan*, vol. 67-A, no. 5, pp. 19–27, 1984.

[19] S. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 3023–3026, 1995.

[20] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, third ed., 1996.

[21] B. Widrow, J. McColl, and M. Ball, "The complex LMS algorithm," *Proceedings of the IEEE*, vol. 63, pp. 719– 720, 1975.

[22] N. Bershad, "On the real and complex least mean square adaptive filter algorithms," *Proceedings of the IEEE*, vol. 69, no. 4, pp. 469–470, 1981.

[23] Y. Bar-Ness and F. Haber, "Interference canceler array with reference generating loop," *IEEE Transactions on Aerospace and Electronic Systems,*, vol. AES-21, no. 5, pp. 654–662, 1985.

[24] L. Horowitz and K. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Transactions on Circuits and Systems*, vol. 28, no. 6, pp. 562–576, 1981.

[25] M. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," *Proceedings of the IEEE*, vol. 66, no. 12, pp. 1658–1659, 1978.

[26] J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 14–37, 1992.

[27] S. Narayan and A. Peterson, "Frequency domain least-mean-square algorithm," *Proceedings of the IEEE*, vol. 69, no. 1, pp. 124–126, 1981.

[28] B. D. H. Brandwood, "A complex gradient operator and its applications in adaptive array theory," *IEE Proceedings: Communications, Radar and Signal Proceedings*, vol. 130, no. 1, pp. 11–16, 1983.

[29] W. Wirtinger, "Zur formalen theorie der funktionen von mehr komplexen veränderlichen," *Mathematische Annalen*, vol. 97, pp. 357–375, December 1927.

[30] R. Remmert, *Theory of complex functions*. Springer, 1992.

[31] F. Neeser and J. Massey, "Proper complex random processes with applications to information theory," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1293–1302, 1993.

[32] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Transactions on Signal Processing*, vol. 43, no. 8, pp. 2030–2033, 1995.

[33] B. Picinbono, "Wide-sense linear mean square estimation and prediction," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95.,*, vol. 3, pp. 2032–2035, 1995.

[34] P. Chevalier and B. Picinbono, "Complex linear-quadratic systems for detection and array processing," *IEEE Transactions on Signal Processing*, vol. 44, no. 10, pp. 2631–2634, 1996.

[35] Y. Yoon and H. Leib, "Maximizing snr in improper complex noise and applications to cdma," *IEEE Communications Letters*, vol. 1, no. 1, pp. 5–8, 1997.

[36] G. Gelli, L. Paura, and A. Ragozini, "Blind widely linear multiuser detection," *IEEE Communications Letters*, vol. 4, no. 6, pp. 187–189, 2000.

[37] R. Schober, W. Gerstacker, and L. H.-J. Lampe, "Data-aided and blind stochastic gradient algorithms for widely linear MMSE MAI suppression for DS-CDMA," *IEEE Transactions on Signal Processing*, vol. 52, no. 3, pp. 746–756, 2004.

[38] S. Javidi, M. Pedzisz, S. L. Goh, and D. P. Mandic, "The augmented complex least mean square algorithm with application to adaptive prediction problems," *in Proceedings of the 1st IARP Workshop on Cognitive Information Processing*, pp. 54 – 57, 2008.

[39] S. C. Douglas, "Widely-linear recursive least-squares algorithm for adaptive beamforming," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2041–2044, 2009.

[40] F. Neto, V. Nascimento, and M. T. M. Silva, "Reduced-complexity widely linear adaptive estimation," in *Proceedings of the 7th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 399–403, 2010.

[41] Y. Xia, C. Took, S. Javidi, and D. Mandic, "A widely linear affine projection algorithm," in *Proceedings of the IEEE/SP 15th Workshop on Statistical Signal Processing*, pp. 373–376, 2009.

[42] W. Dang and L. Scharf, "Extensions to the theory of widely linear complex Kalman filtering," *IEEE Transactions on Signal Processing,*, vol. 60, no. 12, pp. 6669–6674, 2012.

[43] D. Dini, D. Mandic, and S. Julier, "A widely linear complex unscented Kalman filter," *IEEE Signal Processing Letters*, vol. 18, no. 11, pp. 623–626, 2011.

[44] D. H. Dini and D. Mandic, "Widely linear complex extended Kalman filters," in *Sensor Signal Processing for Defence (SSPD)*, pp. 1–5, 2011.

[45] D. P. Mandic and V. S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models.* John Wiley and Sons Ltd, 2009.

[46] S. L. Goh and D. Mandic, "An augmented extended Kalman filter algorithm for complex-valued recurrent neural networks," in *IEEE International Conference on Acoustics (ICASSP)*, vol. 5, pp. V–V, 2006.

[47] F. Cattivelli and A. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[48] Y. Xia, D. Mandic, and A. Sayed, "An adaptive diffusion augmented CLMS algorithm for distributed filtering of noncircular complex signals," *Signal Processing Letters, IEEE*, vol. 18, no. 11, pp. 659–662, 2011.

[49] F. Cattivelli, C. Lopes, and A. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, 2008.

[50] B. Rao and H. Durrant-Whyte, "Fully decentralised algorithm for multisensor Kalman filtering," *IEE Proceedings on Control Theory and Applications*, vol. 138, no. 5, pp. 413–420, 1991.

[51] F. Cattivelli and A. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control,*, vol. 55, no. 9, pp. 2069–2084, 2010.

[52] D. Dini and D. Mandic, "Cooperative adaptive estimation of distributed noncircular complex signals," in *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, pp. 1518–1522, 2012.

[53] H.-D. Schutte and J. Wenzel, "Hypercomplex numbers in digital signal processing," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1557–1560, 1990.

[54] K. Ueda and S. Takahashi, "Digital filters with hypercomplex coefficients," in *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS*, pp. 479–482, 1993.

[55] H. Toyoshima, "Computationally efficient implementation of hypercomplex digital filters," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing,*, vol. 3, pp. 1761–1764, 1998.

[56] S. Sangwine, "Fourier transforms of colour images using quaternion or hypercomplex, numbers," *Electronics Letters*, vol. 32, no. 21, pp. 1979–1980, 1996.

[57] S.-C. Pei and C.-M. Cheng, "A novel block truncation coding of color images using a quaternion-moment-preserving principle," *IEEE Transactions on Communications*, vol. 45, no. 5, pp. 583–595, 1997.

[58] S. Sangwine and T. Ell, "Hypercomplex auto- and cross-correlation of color images," in *Proceedings of the International Conference on Image Processing*, vol. 4, pp. 319–322, 1999.

[59] T. Ell, "Quaternion-fourier transforms for analysis of two-dimensional linear time-invariant partial differential systems," in *Proceedings of the 32nd IEEE Conference on*, vol. 2, 1993.

[60] S. Sangwine, "The discrete quaternion fourier transform," in *Proceedings of the Sixth International Conference on Image Processing and Its Applications*, vol. 2, pp. 790–793, 1997.

[61] M. Bahri, E. S. Hitzer, A. Hayashi, and R. Ashino, "An uncertainty principle for quaternion Fourier transform," *Computers and Mathematics with Applications*, vol. 56, no. 9, pp. 2398 – 2410, 2008.

[62] N. Le Bihan and S. Sangwine, "Quaternion principal component analysis of color images," in *Proceedings of the International Conference on Image Processing*, vol. 1, pp. I–809–12, 2003.

[63] S. Miron, N. Le Bihan, and J. Mars, "Quaternion-MUSIC for vector-sensor array processing," *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1218–1229, 2006.

[64] C. Cheong Took and D. P. Mandic, "Augmented second-order statistics of quaternion random signals," *Signal Processing*, vol. 91, no. 2, pp. 214–224, 2011.

[65] J. Via, D. Ramirez, and I. Santamaria, "Properness and widely linear processing of quaternion random vectors," *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3502 –3515, 2010.

[66] C. Cheong Took and D. P. Mandic, "The quaternion LMS algorithm for adaptive filtering of hypercomplex processes," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1316–1327, 2009.

[67] C. C. Took and D. P. Mandic, "A quaternion widely linear adaptive filter," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4427–4431, 2010.

[68] X. Yun, M. Lizarraga, E. Bachmann, and R. Mcghee, "An improved quaternion-based Kalman filter for real-time tracking of rigid body orientation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1074–1079, 2003.

[69] D. Choukroun, I. Bar-Itzhack, and Y. Oshman, "Novel quaternion Kalman filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 174–190, 2006.

[70] W. Krlikowski, "On 4-dimensional locally conformally flat almost Kahler manifolds," *Archivum Mathematicum*, vol. 42, no. 3, pp. 215–223, 2006.

[71] A. Sudbery, "Quaternionic analysis," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 85, no. 2, pp. 199–225, 1977.

[72] W. R. Hamilton, *Elements of Quaternions*. Longmans, Green, co., 1866.

[73] A. M. Bork, "Vectors versus quaternions," *American Journal of Physis*, vol. 34, p. 202, 4 1966.

[74] A. J. Hanson, *Visualizing Quaternions*. Morgan-Kaufmann/Elsevier, 2006.

[75] K. Shoemake, "Animating rotation with quaternion curves," *SIGGRAPH Computer Graphics*, vol. 19, pp. 245–254, July 1985.

[76] J. C. Hart, G. K. Francis, and L. H. Kauffman, "Visualizing quaternion rotation," *ACM Transactions on Graphics*, vol. 13, pp. 256–276, July 1994.

[77] R. Mukundan, "Quaternions: From classical mechanics to computer graphics, and beyond," in *Proceedings of the 7 th Asian Technology Conference in Mathematics*, 2002.

[78] J. Kuang and S. Tan, "Gps-based attitude determination of gyrostat satellite by quaternion estimation algorithms," *Acta Astronautica*, vol. 51, no. 11, pp. 743 – 759, 2002.

[79] C. Song, S.-J. Kim, S.-H. Kim, and H. Nam, "Robust control of the missile attitude based on quaternion feedback," *Control Engineering Practice*, vol. 14, no. 7, pp. 811 – 818, 2006.

[80] D. Andreis and E. Canuto, "Orbit dynamics and kinematics with full quaternions," in *Proceedings of the American Control Conference*, vol. 4, pp. 3660–3665, 2004.

[81] Y. Yang, "Quaternion based model for momentum biased nadir pointing spacecraft," *Aerospace Science and Technology*, vol. 14, no. 3, pp. 199 – 202, 2010.

[82] A.-M. Zou, K. Kumar, and Z.-G. Hou, "Quaternion-based adaptive output feedback attitude control of spacecraft using Chebyshev neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 9, pp. 1457 –1471, 2010.

[83] A. Ude, "Filtering in a unit quaternion space for model-based object tracking," *Robotics and Autonomous Systems*, vol. 28, no. 2, pp. 163 – 172, 1999.

[84] R. Campa, K. Camarillo, and L. Arias, "Kinematic modeling and control of robot manipulators via unit quaternions: Application to a spherical wrist," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 6474–6479, 2006.

[85] V. Ntouskos, P. Papadakis, and F. Pirri, "A comprehensive analysis of human motion capture data for action recognition," *In Proceedings of the International Conference on Computer Vision Theory and Applications, Rome, Italy*, 2012.

[86] "Filtering in a unit quaternion space for model-based object tracking," *Robotics and Autonomous Systems*, vol. 28, no. 23, pp. 163 – 172, 1999.

[87] C. F. Karney and Karney, "Quaternions in molecular modeling," *Journal of Molecular Graphics and Modelling*, vol. 25, no. 5, pp. 595 – 604, 2007.

[88] J. Ward, *Quaternions and Cayley Numbers*. Kluwer Academic Publishers, 1997.

[89] T. A. Ell and S. J. Sangwine, "Quaternion involutions and anti-involutions," *Computers and Mathematics with Applications*, vol. 53, no. 1, pp. 137–143, 2007.

[90] B. Picinbono, "On circularity," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3473–3482, 1994.

[91] B. Picinbono and P. Bondon, "Second-order statistics of complex signals," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 411 –420, 1997.

[92] A. van den Bos, "Complex gradient and Hessian," *Vision, Image and Signal Processing, IEE Proceedings* -, vol. 141, pp. 380–383, Dec 1994.

[93] P. Schreier and L. Scharf, "Second-order analysis of improper complex random vectors and processes," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 714–725, 2003.

[94] T. Adali, P. Schreier, and L. Scharf, "Complex-valued signal processing: The proper way to deal with impropriety," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5101–5125, 2011.

[95] S. M. Kay, *Fundmantals of statistical signal processing*, vol. 1. Prentice Hall, 1993.

[96] J. H. Mathews and R. W. Howell, *Complex Analysis for mathematics and engineering*. Jones and Bartlett Publishers, 4 ed., 2001.

[97] E. Freitag and R. Busam, *Complex Analysis*. Springer, 2005.

[98] K. Kreutz-Delgado, "The complex gradient operator and the $\mathbb{CR}$-calculus," technical report, University of California, San Diego, 2006.

[99] R. A. Horn and C. A. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.

[100] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. John Wiley and Sons, Inc, 1996.

[101] C. Brooks, *Introductory Econometrics for Finance*. Cambridge university press, 2 ed., 2008.

[102] D. Biamino, G. Cannata, M. Maggiali, and A. Piazza, "Mac-eye: A tendon driven fully embedded robot eye," in *Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots*, pp. 62–67, 2005.

[103] A. Sabatini, "Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346 –1356, 2006.

[104] S.-C. Pei and C.-M. Cheng, "Color image processing by using binary quaternion-moment-preserving thresholding technique," *IEEE Transactions on Image Processing*, vol. 8, no. 5, pp. 614–628, 1999.

[105] N. Rehman and D. P. Mandic, "Empirical mode decomposition for trivariate signals," *IEEE Transactions on Signal Processing,*, vol. 58, no. 3, pp. 1059 –1068, 2010.

[106] P. O. Amblard and N. L. Bihan, "On properness of quaternion valued random variables," *Proceedings of the Interational Conference of Mathematics (IMA) Signal Processing*, 2004.

[107] S. Javidi, C. Cheong Took, C. Jahanchahi, N. Le Bihan, and D. P. Mandic, "Blind extraction of improper quaternion sources," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, no. 5, pp. 3708 –3711, 2011.

[108] S. Javidi, C. Cheong Took, and D. Mandic, "Fast Independent Component Analysis Algorithm for Quaternion Valued Signals," *IEEE Transactions on Neural Networks in Press*, 2011.

[109] J. Via, D. Palomar, L. Vielva, and I. Santamaria, "Quaternion ICA from second-order statistics," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1586 –1600, 2011.

[110] N. L. Bihan and J. Mars, "Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing," *Signal Processing*, vol. 84, no. 7, pp. 1177 – 1199, 2004.

[111] D. P. Mandic, C. Jahanchahi, and C. Cheong Took, "A quaternion gradient operator and its applications," *IEEE Signal Processing Letters*, vol. 18, no. 1, pp. 47 –50, 2011.

[112] C. Took, D. Mandic, and J. Benesty, "Study of the quaternion LMS and four-channel LMS algorithms," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 3109–3112, 2009.

[113] N. Yousef and A. Sayed, "A unified approach to the steady-state and tracking analyses of adaptive filters," *IEEE Transactions on Signal Processing*, vol. 49, no. 2, pp. 314 –324, 2001.

[114] B. Che Ujang, C. Cheong Took, and D. P. Mandic, "Quaternion-valued nonlinear adaptive filtering," *IEEE Transactions on Neural Networks*, vol. 22, no. 8, pp. 1193–1206, 2011.

[115] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind Energy Explained: Theory, Design and Application*. Wiley, 2002.

[116] X. Zhang, J. Zhang, Y. Li, and R. Zhang, "Short-term forecasting of wind speed based on recursive least squares," in *Proceedings of the International Conference on Electrical and Control Engineering (ICECE)*, pp. 367–370, 2011.

[117] K. C. Chou and R. B. Corotis, "Simulation of hourly wind speed and array wind power," *Solar Energy*, vol. 26, no. 3, pp. 199 – 212, 1981.

[118] J. Torres, A. Garca, M. D. Blas, and A. D. Francisco, "Forecast of hourly average wind speed with ARMA models in Navarre (Spain)," *Solar Energy*, vol. 79, no. 1, pp. 65 – 77, 2005.

[119] L. Kamal and Y. Z. Jafri, "Time series models to simulate and forecast hourly averaged wind speed in Quetta, Pakistan," *Solar Energy*, vol. 61, no. 1, pp. 23 – 32, 1997.

[120] S. Sangwine and N. L. Bihan, "Quaternion toolbox for matlab 2005." http://qtfm.sourceforge.net/.

[121] C. Jahanchahi, C. C. Took, and D. P. Mandic, "The widely linear quaternion recursive least squares filter," in *Proceedings of the 2nd International Workshop on Cognitive Information Processing (CIP)*, pp. 87–92, 2010.

[122] C. Jahanchahi, C. Cheong-Took, and D. P. Mandic, "$\mathbb{HR}$ calculus and quaternion analyticity," Technical Report, Imperial College London, Jan 2010.

[123] H.-C. Shin and A. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 90 – 102, 2004.

[124] M. St-Pierre and D. Gingras, "Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 831–835, 2004.

[125] F. Heimes, "Extended Kalman filter neural network training: experimental results and algorithm improvements," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 2, pp. 1639–1644, 1998.

[126] L. Fortuna, G. Muscato, and M. Xibilia, "A comparison between HMLP and HRBF for attitude control," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 318–328, 2001.

[127] X. Wang and Y. Huang, "Convergence study in extended Kalman filter-based training of recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 22, no. 4, pp. 588–600, 2011.

[128] R. Sharaf and A. Noureldin, "Sensor integration for satellite-based vehicular navigation using neural networks," *IEEE Transactions on Neural Networks*, vol. 18, no. 2, pp. 589–594, 2007.

[129] S. Canan, R. Akkaya, and S. Ergintav, "Extended Kalman filter sensor fusion and application to mobile robot," in *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference*, pp. 771–774, 2004.

[130] M. Carminati, G. Ferrari, R. Grassetti, and M. Sampietro, "Real-time data fusion and mems sensors fault detection in an aircraft emergency attitude unit based on Kalman filtering," *IEEE Sensors Journal*, vol. 12, no. 10, pp. 2984–2992, 2012.

[131] J. Marins, X. Yun, E. Bachmann, R. Mcghee, and M. Zyda, "An extended Kalman filter for quaternion-based orientation estimation using marg sensors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, pp. 2003–2011, 2001.

[132] M. D. Shuster, "Approximate algorithms for fast optimal attitude computation," in *Proceedings of the Guidance and Control Conference*, pp. 88–95, Mar. 1978.

[133] W. Grace, "Least squares estimate of spacecraft attitude," *SIAM Review*, vol. 7, no. 3, pp. 384–386, 1965.

[134] I. Bar-Itzhack and Y. Oshman, "Attitude determination from vector observations: Quaternion estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-21, no. 1, pp. 128 –136, 1985.

[135] H. Rughooputh and S. Rughooputh, "Extended Kalman filter learning algorithm for hyper-complex multilayer neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 1824 –1828, 1999.

[136] D. H. Dini and D. Mandic, "Class of widely linear complex Kalman filters," *IEEE Transactions on Neural Networks and Learning Systems,*, vol. 23, no. 5, pp. 775–786, 2012.

[137] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley-Interscience, 2001.

[138] S. Julier and J. Uhlmann, "A new extension of the Kalman filter to non-linear systems," *Proc. AeroSense 11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, pp. 182–193, 1997.

[139] J. H. Kotecha and P. Djuric, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, 2003.

[140] J. H. Conway and D. Smith, *On quaternion and octonions*. A K Peters/CRC Press, 2003.

[141] J. Flifla, J. L. Dillenseger, P. Haigron, and J. L. Coatrieux, "Quaternion based registration method in medical imaging," in *Proceedings of the International Biomedical Engineering Days*, pp. 161–165, 1992.

[142] F. Neto and V. Nascimento, "A novel reduced-complexity widely linear qlms algorithm," in *IEEE Statistical Signal Processing Workshop (SSP)*, pp. 81–84, June 2011.

[143] S. Bayin, *Mathematical methods in science and engineering.* John Wiley and Sons, 2006.

[144] C. Jahanchahi, C. C. Took, and D. P. Mandic, "On gradient calculation in quaternion adaptive filtering," *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3773–3776, 2012.

[145] B. Jelfs, D. P. Mandic, and S. C. Douglas, "An adaptive approach for the identification of improper complex signals," *Signal Processing*, vol. 92, no. 2, pp. 335–344, 2012.

[146] J. Arenas-Garcia and A. H. S. A. R. Figueiras-Vidal, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 1078–1090, 2006.

[147] Y. Gong and C. F. N. Cowan, "An LMS style variable tap-length algorithm for structure adaptation," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2400–2407, 2005.

[148] Y. Zhang, N. Li, J. A. Chambers, and A. H. Sayed, "Steady-state performance analysis of variable tap-length LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 839–845, 2008.

[149] B. C. Ujang, C. C. Took, and D. P. Mandic, "Identification of improper processes by variable tap-length complex valued adaptive filters," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, 2010.

[150] J. Navarro-Moreno, "ARMA prediction of widely linear systems by using the innovations algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3061–3068, 2008.

[151] B. Jelfs, S. Javidi, P. Vayanos, and D. P. Mandic, "Characterisation of signal modality: Exploiting signal nonlinearity in machine learning and signal processing," *Journal of Signal Processing Systems*, vol. 61, no. 1, pp. 105–115, 2010.

[152] E. Ollila, "On the circularity of a complex random variable," *IEEE Signal Processing Letters*, vol. 15, pp. 841–844, 2008.

Appendix A

Duality Between Four-Channel Real Valued LMS Filter and the QLMS

To show the equivalence between the WLIQLMS and four-channel real LMS we compare the signal output and weight update of both filters and show that they are identical.

A.1 The four-channel real valued LMS filter

The output of the four-channel real valued LMS filter is given by:

$$\underbrace{\begin{bmatrix} y_r(k) \\ y_\imath(k) \\ y_\jmath(k) \\ y_\kappa(k) \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} \mathbf{w}_{11}^T(k) & \mathbf{w}_{12}^T(k) & \mathbf{w}_{13}^T(k) & \mathbf{w}_{14}^T(k) \\ \mathbf{w}_{21}^T(k) & \mathbf{w}_{22}^T(k) & \mathbf{w}_{23}^T(k) & \mathbf{w}_{24}^T(k) \\ \mathbf{w}_{31}^T(k) & \mathbf{w}_{32}^T(k) & \mathbf{w}_{33}^T(k) & \mathbf{w}_{34}^T(k) \\ \mathbf{w}_{41}^T(k) & \mathbf{w}_{42}^T(k) & \mathbf{w}_{43}^T(k) & \mathbf{w}_{44}^T(k) \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} \mathbf{x}_r(k) \\ \mathbf{x}_\imath(k) \\ \mathbf{x}_\jmath(k) \\ \mathbf{x}_\kappa(k) \end{bmatrix}}_{\mathbf{x}} \tag{A.1}$$

where $\{\mathbf{w}_{11}, \ldots, \mathbf{w}_{44}\} \in \mathbb{R}^{N\times 1}$ are the 16 filter weights, $\{y_r, y_\imath, y_\jmath, y_\kappa\} \in \mathbb{R}^{1\times 1}$ are the channel outputs and $\{\mathbf{x}_r, \mathbf{x}_\imath, \mathbf{x}_\jmath, \mathbf{x}_\kappa\} \in \mathbb{R}^{N\times 1}$ are the input channels. The error for each

data channel is given by:

$$\begin{aligned} e_r(k) &= d_r(k) - y_r(k) \qquad e_\imath(k) = d_\imath(k) - y_\imath(k) \\ e_\jmath(k) &= d_\jmath(k) - y_\jmath(k) \qquad e_\kappa(k) = d_\kappa(k) - y_\kappa(k) \end{aligned}$$

and the objective function to be minimized

$$J = \frac{1}{2}(e_r^2 + e_\imath^2 + e_\jmath^2 + e_\kappa^2) \tag{A.2}$$

Based on (A.1), the weight updates are obtained from $\mathbf{W}(k+1) = \mathbf{W}(k) - \mu\nabla_{\mathbf{W}}J$, where the gradient $\nabla_{\mathbf{W}}J$ is given by

$$\nabla_{\mathbf{W}}J = \begin{bmatrix} \mathbf{e}_r(k)\mathbf{x}_r(k) & \mathbf{e}_r(k)\mathbf{x}_\imath(k) & \mathbf{e}_r(k)\mathbf{x}_\jmath(k) & \mathbf{e}_r(k)\mathbf{x}_\kappa(k) \\ \mathbf{e}_\imath(k)\mathbf{x}_r(k) & \mathbf{e}_\imath(k)\mathbf{x}_\imath(k) & \mathbf{e}_\imath(k)\mathbf{x}_\jmath(k) & \mathbf{e}_\imath(k)\mathbf{x}_\kappa(k) \\ \mathbf{e}_\jmath(k)\mathbf{x}_r(k) & \mathbf{e}_\jmath(k)\mathbf{x}_\imath(k) & \mathbf{e}_\jmath(k)\mathbf{x}_\jmath(k) & \mathbf{e}_\jmath(k)\mathbf{x}_\kappa(k) \\ \mathbf{e}_\kappa(k)\mathbf{x}_r(k) & \mathbf{e}_\kappa(k)\mathbf{x}_\imath(k) & \mathbf{e}_\kappa(k)\mathbf{x}_\jmath(k) & \mathbf{e}_\kappa(k)\mathbf{x}_\kappa(k) \end{bmatrix}$$

A.2 Duality between the four-channel real LMS and the WLIQLMS

The output of the WLIQLMS is given by

$$y(k) = \mathbf{u}^T(k)\mathbf{x}(k) + \mathbf{v}^T(k)\mathbf{x}^i(k) + \mathbf{g}^T(k)\mathbf{x}^j(k) + \mathbf{h}^T(k)\mathbf{x}^k(k) = \mathbf{w}^{aT}(k)\mathbf{x}^a(k) \tag{A.3}$$

where $\mathbf{x}^a(k) = [\mathbf{x}^T(k), \mathbf{x}^{iT}(k), \mathbf{x}^{jT}(k), \mathbf{x}^{kT}(k)]^T$ and $\mathbf{w}^a(k) = [\mathbf{u}^T(k), \mathbf{v}^T(k), \mathbf{g}^T(k), \mathbf{h}^T(k)]^T$

We can compare the output of the WLIQLMS to that of the four-channel real LMS filter by writing the output $y(k)$ of the WLIQLMS in the same form as the output of the four-channel real LMS, that is, in terms of $y_r(k)$, $y_\imath(k)$, $y_\jmath(k)$ and $y_\kappa(k)$. In doing so we

obtain

$$\begin{aligned} y_r(k) &= (\underbrace{\mathbf{u}_r(k)+\mathbf{v}_r(k)+\mathbf{g}_r(k)+\mathbf{h}_r(k)}_{\mathbf{w}_{11}(k)})^T\mathbf{x}_r + (\underbrace{-\mathbf{u}_\imath(k)-\mathbf{v}_\imath(k)+\mathbf{g}_\imath(k)+\mathbf{h}_\imath(k)}_{\mathbf{w}_{12}(k)})^T\mathbf{x}_\imath \\ &+ (\underbrace{-\mathbf{u}_\jmath(k)+\mathbf{v}_\jmath(k)-\mathbf{g}_\jmath(k)+\mathbf{h}_\jmath(k)}_{\mathbf{w}_{13}(k)})^T\mathbf{x}_\jmath + (\underbrace{-\mathbf{u}_\kappa(k)+\mathbf{v}_\kappa(k)+\mathbf{g}_\kappa(k)-\mathbf{h}_\kappa(k)}_{\mathbf{w}_{14}(k)})^T\mathbf{x}_\kappa \end{aligned}$$

$$\begin{aligned} y_\imath(k) &= (\underbrace{\mathbf{u}_r(k)+\mathbf{v}_r(k)-\mathbf{g}_r(k)-\mathbf{h}_r(k)}_{\mathbf{w}_{21}(k)})^T\mathbf{x}_\imath + (\underbrace{\mathbf{u}_\imath(k)+\mathbf{v}_\imath(k)+\mathbf{g}_\imath(k)+\mathbf{h}_\imath(k)}_{\mathbf{w}_{22}(k)})^T\mathbf{x}_r \\ &+ (\underbrace{\mathbf{u}_\jmath(k)-\mathbf{v}_\jmath(k)-\mathbf{g}_\jmath(k)-\mathbf{h}_\jmath(k)}_{\mathbf{w}_{23}(k)})^T\mathbf{x}_\kappa + (\underbrace{-\mathbf{u}_\kappa(k)+\mathbf{v}_\kappa(k)-\mathbf{g}_\kappa(k)+\mathbf{h}_\kappa(k)}_{\mathbf{w}_{24}(k)})^T\mathbf{x}_\jmath \end{aligned}$$

$$\begin{aligned} y_\jmath(k) = & (\underbrace{\mathbf{u}_r(k)-\mathbf{v}_r(k)+\mathbf{g}_r(k)-\mathbf{h}_r(k)}_{\mathbf{w}_{31}(k)})^T\mathbf{x}_\jmath + (\underbrace{-\mathbf{u}_\imath(k)+\mathbf{v}_\imath(k)+\mathbf{g}_\imath(k)-\mathbf{h}_\imath(k)}_{\mathbf{w}_{32}(k)})^T\mathbf{x}_\kappa \\ &+ (\underbrace{\mathbf{u}_\jmath(k)+\mathbf{v}_\jmath(k)+\mathbf{g}_\jmath(k)+\mathbf{h}_\jmath(k)}_{\mathbf{w}_{33}(k)})^T\mathbf{x}_r + (\underbrace{\mathbf{u}_\kappa(k)+\mathbf{v}_\kappa(k)-\mathbf{g}_\kappa(k)-\mathbf{h}_\kappa(k)}_{\mathbf{w}_{34}(k)})^T\mathbf{x}_\imath \end{aligned}$$

$$\begin{aligned} y_\kappa(k) = & (\underbrace{\mathbf{u}_r(k)-\mathbf{v}_r(k)-\mathbf{g}_r(k)+\mathbf{h}_r(k)}_{\mathbf{w}_{41}(k)})^T\mathbf{x}_\kappa + (\underbrace{\mathbf{u}_\imath(k)-\mathbf{v}_\imath(k)+\mathbf{g}_\imath(k)-\mathbf{h}_\imath(k)}_{\mathbf{w}_{42}(k)})^T\mathbf{x}_\jmath \\ &+ (\underbrace{-\mathbf{u}_\jmath(k)-\mathbf{v}_\jmath(k)+\mathbf{g}_\jmath(k)+\mathbf{h}_\jmath(k)}_{\mathbf{w}_{43}(k)})^T\mathbf{x}_\imath + (\underbrace{\mathbf{u}_\kappa(k)+\mathbf{v}_\kappa(k)+\mathbf{g}_\kappa(k)+\mathbf{h}_\kappa(k)}_{\mathbf{w}_{44}(k)})^T\mathbf{x}_r \end{aligned}$$

By comparing the equations above with (A.1), we can identify the filter coefficient vectors $\mathbf{w}_{11}(k)$ to $\mathbf{w}_{44}(k)$, and the two filters therefore have the same output.

To show that the weight update for the WLIQLMS is identical to the weight update of the four-channel real LMS, consider the WLIQLMS weight update, given by

$$\begin{aligned} \mathbf{u}(k+1) &= \mathbf{u}(k)+\mu\left(\frac{1}{2}e(k)\mathbf{x}^*(k)\right) \qquad \mathbf{v}(k+1)=\mathbf{v}(k)+\mu\left(\frac{1}{2}e(k)\mathbf{x}^{i^*}(k)\right) \\ \mathbf{g}(k+1) &= \mathbf{g}(k)+\mu\left(\frac{1}{2}e(k)\mathbf{x}^{j^*}(k)\right) \qquad \mathbf{h}(k+1)=\mathbf{h}(k)+\mu\left(\frac{1}{2}e(k)\mathbf{x}^{k^*}(k)\right) \end{aligned} \quad \text{(A.4)}$$

We can compare (A.4) with the weight update for the four-channel real LMS. For illustration, we consider the real part of the weight updates, that is, $\mathbf{u}_r$, $\mathbf{v}_r$, $\mathbf{g}_r$ and $\mathbf{h}_r$ and show that the real part of the weight update $\Delta\mathbf{w}_{11}(k)$ is related by:

$$\Delta\mathbf{w}_{11}(k) = \frac{1}{2}\big(\Delta\mathbf{u}_r(k) + \Delta\mathbf{v}_r(k) + \Delta\mathbf{g}_r(k) + \Delta\mathbf{h}_r(k)\big) \tag{A.5}$$

To show this, we can write the weight update in (A.4) in terms on the real and imaginary components, as

$$\mathbf{u}_r(k+1) = \mathbf{u}_r(k) + \mu\left(\frac{1}{2}e_r(k)\mathbf{x}_r(k) + \frac{1}{2}e_\imath(k)\mathbf{x}_\imath(k) + \frac{1}{2}e_\jmath(k)\mathbf{x}_\jmath(k) + \frac{1}{2}e_\kappa(k)\mathbf{x}_\kappa(k)\right)$$

$$\mathbf{v}_r(k+1) = \mathbf{v}_r(k) + \mu\left(\frac{1}{2}e_r(k)\mathbf{x}_r(k) + \frac{1}{2}e_\imath(k)\mathbf{x}_\imath(k) - \frac{1}{2}e_\jmath(k)\mathbf{x}_\jmath(k) - \frac{1}{2}e_\kappa(k)\mathbf{x}_\kappa(k)\right)$$

$$\mathbf{g}_r(k+1) = \mathbf{g}_r(k) + \mu\left(\frac{1}{2}e_r(k)\mathbf{x}_r(k) - \frac{1}{2}e_\imath(k)\mathbf{x}_\imath(k) + \frac{1}{2}e_\jmath(k)\mathbf{x}_\jmath(k) - \frac{1}{2}e_\kappa(k)\mathbf{x}_\kappa(k)\right)$$

$$\mathbf{h}_r(k+1) = \mathbf{h}_r(k) + \mu\left(\frac{1}{2}e_r(k)\mathbf{x}_r(k) - \frac{1}{2}e_\imath(k)\mathbf{x}_\imath(k) - \frac{1}{2}e_\jmath(k)\mathbf{x}_\jmath(k) + \frac{1}{2}e_\kappa(k)\mathbf{x}_\kappa(k)\right)$$

Observing that

$$\frac{1}{2}\left(\Delta\mathbf{u}_r(k) + \Delta\mathbf{v}_r(k) + \Delta\mathbf{g}_r(k) + \Delta\mathbf{h}_r(k)\right) = \frac{1}{2}\left(2e_r(k)\mathbf{x}_r(k)\right) = \Delta\mathbf{w}_{11}(k) \tag{A.6}$$

this illustrates that the four-channel LMS and WLIQLMS are equivalent when the four-channel LMS has a step size twice as large as that of the WLIQLMS.

A.3 Mathematical complexity

We have shown that the IQLMS is identical to the four-channel LMS. However, due to the over-parameterization of the quaternion augmented covariance matrix, the WLIQLMS requires four times as many mathematical operations as the four-channel LMS. However,

by developing efficient implementations of the WLIQLMS, the mathematical complexity can be reduced to that of the four-channel LMS, while maintaining the advantages inherent in working in a division algebra. This is addressed in [142] and in a recently submitted paper, summarised in Appendix H.

Appendix B

QLMS Convergence Proofs

B.1 Proof for the equation (3.64)

Substitute for $e(k)$ in (3.59) into (3.63) to obtain

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{1}{2}\mu\mathbf{w}_o^T\mathbf{x}(k)\mathbf{x}^*(k) + \frac{1}{2}\mu n(k)\mathbf{x}^*(k) - \frac{1}{2}\mu\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^*(k) - \frac{1}{4}\mu\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}_o^* - \frac{1}{4}\mu\mathbf{x}(k)n^*(k) + \frac{1}{4}\mu\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{w}^*(k)$$

We can now subtract the optimal weight vector $\mathbf{w}_o$ from both sides to obtain the weight error vector $\mathbf{r}(k) = \mathbf{w}(k) - \mathbf{w}_o(k)$ in the form

$$\mathbf{r}(k+1) = \mathbf{r}(k) - \frac{1}{2}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^*(k)\right) + \frac{1}{4}\mu\left(\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{r}^*(k)\right) + \frac{1}{2}\mu n(k)\mathbf{x}^*(k) - \frac{1}{4}\mu\mathbf{x}(k)n^*(k)$$

The Hermitian transpose of the term $\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{r}^*(k)$ can be written as $\left(\mathbf{x}(k)\mathbf{x}^H(k)\mathbf{r}^*(k)\right)^H = \mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)$ and noting also that $\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^*(k) = (\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k))^T$, we have

$$\mathbf{r}(k+1) = \mathbf{r}(k) - \frac{1}{2}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right)^T + \frac{1}{4}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right)^H + \frac{1}{2}\mu n(k)\mathbf{x}^*(k) - \frac{1}{4}\mu\mathbf{x}(k)n^*(k)$$

Taking the transpose of both sides gives

$$\mathbf{r}^T(k+1) = \mathbf{r}^T(k) - \frac{1}{2}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right) + \frac{1}{4}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right)^* + \frac{1}{2}\mu n(k)\mathbf{x}^H(k) - \frac{1}{4}\mu\mathbf{x}^T(k)n^*(k)$$

B.2 Expression for the convergence of the QLMS

Upon applying the statistical expectation operator to (3.64) we obtain (the noise term vanishes as it is zero mean and independent from other terms)

$$E[\mathbf{r}^T(k+1)] = E[\mathbf{r}^T(k)] - \frac{1}{2}\mu\left(E[\mathbf{r}^T(k)]\mathbf{R}_{xx}\right) + \frac{1}{4}\mu\left(E[\mathbf{r}^T(k)]\mathbf{R}_{xx}\right)^* \tag{B.1}$$

For $\mathbf{r}^T(k) = \mathbf{r}_r^T(k) + \imath\mathbf{r}_\imath^T(k) + \jmath\mathbf{r}_\jmath^T(k) + \kappa\mathbf{r}_\kappa^T(k)$, we can write

$$\begin{aligned}
E[\mathbf{r}_r^T(k+1)] &= E[\mathbf{r}_r^T(k)] - \frac{1}{2}\mu\left[E[\mathbf{r}_r^T(k)][\mathbf{R}_{xx}]_r + \imath E[\mathbf{r}_\imath^T(k)]\imath[\mathbf{R}_{xx}]_\imath + \jmath E[\mathbf{r}_\jmath^T(k)]\jmath[\mathbf{R}_{xx}]_\jmath\right. \\
&+ \left.\kappa E[\mathbf{r}_\kappa^T(k)]\kappa[\mathbf{R}_{xx}]_r\right] + \frac{1}{4}\mu\left[E[\mathbf{r}_r^T(k)][\mathbf{R}_{xx}]_r + \imath E[\mathbf{r}_\imath^T(k)]\imath[\mathbf{R}_{xx}]_\imath\right. \\
&+ \left.\jmath E[\mathbf{r}_\jmath^T(k)]\jmath[\mathbf{R}_{xx}]_\jmath + \kappa E[\mathbf{r}_\kappa^T(k)]\kappa[\mathbf{R}_{xx}]_r\right] \\
&= E[\mathbf{r}_r^T(k)] - \frac{1}{4}\mu\left[E[\mathbf{r}_r^T(k)][\mathbf{R}_{xx}]_r + \imath E[\mathbf{r}_\imath^T(k)]\imath[\mathbf{R}_{xx}]_\imath + \jmath E[\mathbf{r}_\jmath^T(k)]\jmath[\mathbf{R}_{xx}]_\jmath\right. \\
&+ \left.\kappa E[\mathbf{r}_\kappa^T(k)]\kappa[\mathbf{R}_{xx}]_r\right]
\end{aligned}$$

and similarly

$$\begin{aligned}
\imath E[\mathbf{r}_\imath^T(k+1)] &= \imath E[\mathbf{r}_\imath^T(k)] - \frac{3}{4}\mu\left[E[\mathbf{r}_r^T(k)]\imath[\mathbf{R}_{xx}]_\imath + \imath E[\mathbf{r}_\imath^T(k)][\mathbf{R}_{xx}]_r + \jmath E[\mathbf{r}_\jmath^T(k)]\kappa[\mathbf{R}_{xx}]_\kappa\right. \\
&+ \left.\kappa E[\mathbf{r}_\kappa^T(k)]\jmath[\mathbf{R}_{xx}]_\jmath\right]
\end{aligned}$$

$$\begin{aligned}
\jmath E[\mathbf{r}_\jmath^T(k+1)] &= \jmath E[\mathbf{r}_\jmath^T(k)] - \frac{3}{4}\mu\left[E[\mathbf{r}_r^T(k)]\jmath[\mathbf{R}_{xx}]_\jmath + \imath E[\mathbf{r}_\imath^T(k)]\kappa[\mathbf{R}_{xx}]_\kappa + \jmath E[\mathbf{r}_\jmath^T(k)][\mathbf{R}_{xx}]_r\right. \\
&+ \left.\kappa E[\mathbf{r}_\kappa^T(k)]\imath[\mathbf{R}_{xx}]_\imath\right]
\end{aligned}$$

$$\begin{aligned}
\kappa E[\mathbf{r}_\kappa^T(k+1)] &= \jmath E[\mathbf{r}_\jmath^T(k)] - \frac{3}{4}\mu\left[E[\mathbf{r}_r^T(k)]\kappa[\mathbf{R}_{xx}]_\kappa + \imath E[\mathbf{r}_\imath^T(k)]\jmath[\mathbf{R}_{xx}]_\jmath + \jmath E[\mathbf{r}_\jmath^T(k)]\imath[\mathbf{R}_{xx}]_\imath\right. \\
&+ \left.\kappa E[\mathbf{r}_\kappa^T(k)][\mathbf{R}_{xx}]_r\right]
\end{aligned}$$

B.3 Convergence comparison of the QLMS, HR-QLMS and IQLMS

To compare the eigenvalues of the covariance matrix $\mathbf{R}_b$ within the HR-QLMS and $\mathbf{R}_c$ within the QLMS to those of the matrix $\mathbf{R}_{xx}$ within the IQLMS, we re-write the IQLMS weight error vector in (3.60) in terms of its $r-$, $\imath-$, $\jmath-$ and $\kappa-$parts

$$E[\boldsymbol{\omega}(k+1)] = (\mathbf{I} - \frac{3}{4}\mu\mathbf{R}_a)E[\boldsymbol{\omega}(k)] \tag{B.2}$$

where $\boldsymbol{\omega}(k) = [\mathbf{v}_r^T(k)\ ,\ \imath\mathbf{v}_\imath^T(k)\ ,\ \jmath\mathbf{v}_\jmath^T(k)\ ,\ \kappa\mathbf{v}_\kappa^T(k)]^T$ and

$$\mathbf{R}_a = \begin{pmatrix} [\mathbf{R}_{xx}]_r & \imath[\mathbf{R}_{xx}]_\imath & \jmath[\mathbf{R}_{xx}]_\jmath & \kappa[\mathbf{R}_{xx}]_\kappa \\ \imath[\mathbf{R}_{xx}]_\imath & [\mathbf{R}_{xx}]_r & \kappa[\mathbf{R}_{xx}]_\kappa & \jmath[\mathbf{R}_{xx}]_\jmath \\ \jmath[\mathbf{R}_{xx}]_\jmath & \kappa[\mathbf{R}_{xx}]_\kappa & [\mathbf{R}_{xx}]_r & \imath[\mathbf{R}_{xx}]_\imath \\ \kappa[\mathbf{R}_{xx}]_\kappa & \jmath[\mathbf{R}_{xx}]_\jmath & \imath[\mathbf{R}_{xx}]_\imath & [\mathbf{R}_{xx}]_r \end{pmatrix}$$

Notice that the matrix $\mathbf{R}_b$ is identical to the matrix $\mathbf{R}_a$, except for the first column of block matrices of $\mathbf{R}_a$ being scaled by 3, allowing us to express $\mathbf{R}_a$ in terms of $\mathbf{R}_b$ as

$$\mathbf{R}_b = \mathbf{R}_a\mathbf{D} \tag{B.3}$$

where $\mathbf{D}$ is the diagonal matrix $diag(\frac{1}{3}\mathbf{I}_N, \mathbf{I}_N, \mathbf{I}_N, \mathbf{I}_N)) \in \mathbb{H}^{4N\times 4N}$ and $\mathbf{I}_N$ is the identity matrix of size $N \times N$. We now make the assumption that the cross-covariance of the $r-$, $i-$, $j-$ and $k-$ part of signal vector $\mathbf{x}$ is either symmetric or zero. This condition may at first appear over-restrictive, but a signal satisfying this condition is circular and the QLMS and IQLMS are only optimal for circular signals, and so it is reasonable to make this assumption. Making this assumption, we have $[\mathbf{R}_{xx}]_\imath = [\mathbf{R}_{xx}]_\jmath = [\mathbf{R}_{xx}]_\kappa = \mathbf{0}$, and matrices $\mathbf{R}_a$ and $\mathbf{R}_b$ and $\mathbf{R}_c$ become block diagonal. Using the following:

- The eigenvalues of a block diagonal matrix are the eigenvalues of its blocks;
- Scaling a matrix by a factor α scales the eigenvalues by the same factor;

we can further write

$$\lambda_{max}(\mathbf{R_a}) = \lambda_{max}(\mathbf{R}_{xx}) \qquad \lambda_{min}(\mathbf{R_a}) = \lambda_{min}(\mathbf{R}_{xx}) \tag{B.4}$$

$$\lambda_{max}(\mathbf{R_b}) = \lambda_{max}(\mathbf{R}_{xx}) \qquad \lambda_{min}(\mathbf{R_b}) = \frac{1}{3}\lambda_{min}(\mathbf{R}_{xx}) \tag{B.5}$$

$$\lambda_{max}(\mathbf{R_c}) = \frac{5}{6}\lambda_{max}(\mathbf{R}_{xx}) \qquad \lambda_{min}(\mathbf{R_c}) = \frac{1}{2}\lambda_{min}(\mathbf{R}_{xx}) \tag{B.6}$$

B.4 Proof for the equation (3.70)

Substituting for $e(k)$ into (3.52), we obtain

$$\begin{aligned}\mathbf{w}(k+1) = \mathbf{w}(k) &+ \frac{1}{2}\mu\mathbf{w}_o^T\mathbf{x}(k)\mathbf{x}^*(k) + \frac{1}{2}\mu n(k)\mathbf{x}^*(k) - \frac{1}{2}\mu\mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^*(k) \\ &-\frac{1}{4}\mu\mathbf{x}^*(k)\mathbf{x}^H(k)\mathbf{w}_o^* - \frac{1}{4}\mu\mathbf{x}^*(k)n^*(k) + \frac{1}{4}\mu\mathbf{x}^*(k)\mathbf{x}^H(k)\mathbf{w}^*(k)\end{aligned}$$

We can now subtract $\mathbf{w}_o$ to obtain the weight error vector $\mathbf{r}(k) = \mathbf{w}(k) - \mathbf{w}_o(k)$, in the form

$$\mathbf{r}(k+1) = \mathbf{r}(k) - \frac{1}{2}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^*(k)\right) + \frac{1}{4}\mu\left(\mathbf{x}^*(k)\mathbf{x}^H(k)\mathbf{r}^*(k)\right) + \frac{1}{2}\mu n(k)\mathbf{x}^*(k) - \frac{1}{4}\mu\mathbf{x}^*(k)n^*(k)$$

The Hermitian of the term $\mathbf{x}^*(k)\mathbf{x}^H(k)\mathbf{r}^*(k)$ can be written as $\left(\mathbf{x}^*(k)\mathbf{x}^H(k)\mathbf{r}^*(k)\right)^H = \mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)$ and noting also that $\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^*(k) = (\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k))^T$, we have

$$\mathbf{r}(k+1) = \mathbf{r}(k) - \frac{1}{2}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right)^T + \frac{1}{4}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\right)^H + \frac{1}{2}\mu n(k)\mathbf{x}^*(k) - \frac{1}{4}\mu\mathbf{x}^*(k)n^*(k)$$

Taking the transpose of both sides gives

$$\mathbf{r}^T(k+1) = \mathbf{r}^T(k) - \frac{1}{2}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^H(k)\right) + \frac{1}{4}\mu\left(\mathbf{r}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\right)^* + \frac{1}{2}\mu n(k)\mathbf{x}^H(k) - \frac{1}{4}\mu\mathbf{x}^H(k)n^*(k)$$

B.5 Proof for $E[\mathbf{x}\mathbf{x}^T] = -\frac{1}{2}E[\mathbf{x}\mathbf{x}^H]$

Take the transpose of $\mathbf{x} = \frac{1}{2}(\mathbf{x}^{i*} + \mathbf{x}^{j*} + \mathbf{x}^{k*} - \mathbf{x}^*)$ and premultiply by $\mathbf{x}$ to obtain

$$\mathbf{x}\mathbf{x}^T = \frac{1}{2}\mathbf{x}(\mathbf{x}^{i*} + \mathbf{x}^{j*} + \mathbf{x}^{k*} - \mathbf{x}^*)^T = \frac{1}{2}\mathbf{x}(\mathbf{x}^{iH} + \mathbf{x}^{jH} + \mathbf{x}^{kH} - \mathbf{x}^H)$$

Taking the expectation of both sides gives

$$\mathbf{C}_x = \frac{1}{2}(\mathbf{P}_x + \mathbf{S}_x + \mathbf{T}_x - \mathbf{R}_x)$$

where $\mathbf{C}_x = E[\mathbf{x}\mathbf{x}^T]$. For circular signals, $\mathbf{P}_x = \mathbf{S}_x = \mathbf{T}_x = \mathbf{0}$, and therefore

$$\mathbf{C}_x = -\frac{1}{2}\mathbf{R}_x$$

B.6 MSE expression for the IQLMS

For an improper teaching signal $d(k) = \mathbf{w}_o^H \mathbf{x}(k) + \mathbf{v}_o^H \mathbf{x}^\imath(k) + \mathbf{g}_o^H \mathbf{x}^\jmath(k) + \mathbf{h}_o^H \mathbf{x}^\kappa(k) + n(k)$ the error can be written as

$$\begin{aligned} e(k) &= \mathbf{d}(k) - \mathbf{w}^H(k)\mathbf{x}(k) \\ e(k) &= \mathbf{w}_o^H \mathbf{x}(k) + \mathbf{v}_o^H \mathbf{x}^\imath(k) + \mathbf{g}_o^H \mathbf{x}^\jmath(k) + \mathbf{h}_o^H \mathbf{x}^\kappa(k) + n(k) - \mathbf{w}^H(k)\mathbf{x}(k) \\ e(k) &= \mathbf{r}^H \mathbf{x}(k) + \mathbf{v}_o^H \mathbf{x}^\imath(k) + \mathbf{g}_o^H \mathbf{x}^\jmath(k) + \mathbf{h}_o^H \mathbf{x}^\kappa(k) + n(k) \\ e(k) &= e_a(k) + \mathbf{w}^{cH} \mathbf{x}^c(k) + n(k) \end{aligned} \tag{B.7}$$

where

$$\mathbf{x}^c(k) = [\mathbf{x}^{\imath T}(k), \mathbf{x}^{\jmath T}(k), \mathbf{x}^{\kappa T}(k)]^T \qquad \mathbf{w}^c = [\mathbf{v}_o^T, \mathbf{g}_o^T, \mathbf{h}_o^T]^T$$

The mean square error can now be written as

$$MSE = \lim_{k\to\infty} E\left[\|e(k)\|\right] = \lim_{k\to\infty} E\left[\|e_a(k) + n(k) + \mathbf{w}^{cH}\mathbf{x}^c(k)\|^2\right]$$

Making the usual independent assumption that

A.1 The terms $e_a(k)$, $n(k)$ and $\mathbf{x}(k)$ are statistically independent and $\lim_{k\to\infty} E[e_a(k)] = E[n(k)] = 0$

the MSE takes the form

$$\begin{aligned} MSE &= \lim_{k\to\infty} E\left[\|e_a(k)\|^2\right] + \lim_{k\to\infty} E\left[\|\mathbf{w}^{cH}\mathbf{x}^c\|^2\right] + \lim_{k\to\infty} E\left[\|n(k)\|^2\right] \\ &= EMSE + \lim_{k\to\infty} E\left[\|\mathbf{w}^{cH}\mathbf{x}^c\|^2\right] + \sigma^2 \end{aligned}$$

where EMSE is the excess mean square error and is defined as

$$EMSE = \lim_{k\to\infty} E[\|e_a(k)\|^2]$$

B.7 Proof for the conservation of weight error energy

Following the approach in [123], substitute for $\mathbf{r}(k) = \mathbf{w}_o - \mathbf{w}(k)$ into the IQLMS weight update (where the filter takes the form $\mathbf{w}^H\mathbf{x}$) to give

$$\mathbf{r}(k+1) = \mathbf{r}(k) - \mu\mathbf{x}(k)e^*(k)$$

Taking the Hermitian of both sides we have

$$\mathbf{r}^H(k+1) = \mathbf{r}^H(k) - \mu e(k)\mathbf{x}^H(k) \tag{B.8}$$

and by post-multiplying both sides by $\mathbf{x}(k)$ we obtain

$$\mathbf{r}^H(k+1)\mathbf{x}(k) = \mathbf{r}^H(k)\mathbf{x}(k) - \mu e(k)\|\mathbf{x}(k)\|^2 \tag{B.9}$$

The a priori error $e_a(k)$ and a posteriori error $e_p(k)$ can now be defined

$$e_a(k) = \mathbf{r}^H(k)\mathbf{x}(k) \qquad\qquad e_p(k) = \mathbf{r}^H(k+1)\mathbf{x}(k)$$

Substitute into (B.9) to give

$$\begin{aligned} e_p(k) &= e_a(k) - \mu e(k)\|\mathbf{x}(k)\|^2 \\ \mu e(k) &= (e_a(k) - e_p(k))\|\mathbf{x}(k)\|^{-2} \end{aligned}$$

allowing us to rewrite (B.8) as

$$\mathbf{r}^H(k+1) = \mathbf{r}^H(k) - \big(e_a(k) - e_p(k)\big)\|\mathbf{x}(k)\|^{-2}\mathbf{x}^H(k)$$

Upon rearranging the terms above

$$\mathbf{r}^H(k+1) + e_a(k)\|\mathbf{x}(k)\|^{-2}\mathbf{x}^H(k) = \mathbf{r}^H(k) + e_p(k)\|\mathbf{x}(k)\|^{-2}\mathbf{x}^H(k)$$

Evaluating the energy ($\|\cdot\|^2$) of both sides, the energy conservation relationship can be written as [123]

$$\|\mathbf{r}(k+1)\|^2 + \frac{\|e_a(k)\|^2}{\|\mathbf{x}(k)\|^2} = \|\mathbf{r}(k)\|^2 + \frac{\|e_p(k)\|^2}{\|\mathbf{x}(k)\|^2}$$

Appendix C

The Discrepancy Between the Measured MSE of the ARMA Process in (5.39) and Theoretical MSE in (5.36)

Because the affine projection algorithm operates on an adaptive FIR filter, it was assumed that the signal is generated by an FIR filter. This then allowed us to assume that the noise vector $\mathbf{v}(k)$ in (5.12) is independent of the data matrix $\mathbf{A}(k)$ in the analysis (see footnote 2). While this assumption is valid for an FIR filter when $\mathbf{v}(k)$ is white noise, it is only valid for an AR model (and also ARMA model) when $N = 1$. To illustrate this, take the simple AR(1) model

$$x(k) = 0.9x(k-1) + v(k) \tag{C.1}$$

where $v(k)$ is white Gaussian noise. When the filter length $M = 3$ and the number of constraints $N = 1$, then the data matrix $\mathbf{A}(k) = [x(k-1), x(k-2), x(k-3)]^T$ and $\mathbf{v}(k) = [v(k)]$. In this case we see that it is accurate to assume that $v(k)$ is independent

from $\mathbf{A}(k)$. However, For $N = 2$ we have

$$\mathbf{A}(k) = \begin{bmatrix} x(k-1) & x(k-2) \\ x(k-2) & x(k-3) \\ x(k-3) & x(k-4) \end{bmatrix}$$

and $\mathbf{v}(k) = [v(k), v(k-1)]$. In this case $\mathbf{v}(k)$ and $\mathbf{A}(k)$ are correlated because of the presence of the term $v(k-1)$ in $\mathbf{v}(k)$ and $x(k-1)$ in $\mathbf{A}(k)$. As the number of constraints N increases there are more and more terms in $\mathbf{A}(k)$ that are correlated with $\mathbf{v}(k)$ and so the assumption of independence between the noise vector $\mathbf{v}(k)$ and data matrix $\mathbf{A}(k)$ is no longer valid.

Appendix D

Proof for the First Order Quaternion Taylor Series

The Taylor series of a real quadrivariate function $u(q_a, q_b, q_c, q_d)$ is given by [143]

$$u(q_r, q_\imath, q_\jmath, q_\kappa) = u(q_{r0}, q_{\imath 0}, q_{\jmath 0}, q_{\kappa 0}) + \sum_{n=1}^{\infty} \frac{1}{n} \left[\Delta q_r \frac{\partial}{\partial q_r} + \Delta q_\imath \frac{\partial}{\partial q_\imath} + \Delta q_\jmath \frac{\partial}{\partial q_\jmath} + \Delta q_\kappa \frac{\partial}{\partial q_\kappa} \right]^n u(q_{r0}, q_{\imath 0}, q_{\jmath 0}, q_{\kappa 0}) \tag{D.1}$$

where $\Delta q_\eta = q_\eta - q_{\eta 0}$, for $\eta = \{r, \imath, \jmath, \kappa\}$. Using the identity

$$\begin{bmatrix} \frac{\partial f}{\partial q_r} \\ \frac{\partial f}{\partial q_\imath} \\ \frac{\partial f}{\partial q_\jmath} \\ \frac{\partial f}{\partial q_\kappa} \end{bmatrix}^T = \begin{bmatrix} \frac{\partial f}{\partial q} \\ \frac{\partial f}{\partial q^\imath} \\ \frac{\partial f}{\partial q^\jmath} \\ \frac{\partial f}{\partial q^\kappa} \end{bmatrix}^T \begin{bmatrix} 1 & \imath & \jmath & \kappa \\ 1 & \imath & -\jmath & -\kappa \\ 1 & -\imath & \jmath & -\kappa \\ 1 & -\imath & -\jmath & \kappa \end{bmatrix} \tag{D.2}$$

to related the real partial derivatives to quaternion partial derivatives, we next derive the quaternion Taylor series. For a real function of quaternion variables, we have $f(q, q^\imath, q^\jmath, q^\kappa) = u(q_r, q_\imath, q_\jmath, q_\kappa)$ and so substituting (D.2) into (D.1) and using the identities $\Delta q_r = \frac{1}{4}\left(\Delta q + \Delta q^\imath + \Delta q^\jmath + \Delta q^\kappa\right)$, $\Delta q_\imath = \frac{1}{4\imath}\left(\Delta q + \Delta q^\imath - \Delta q^\jmath - \Delta q^\kappa\right)$, $\Delta q_\jmath = \frac{1}{4\jmath}\left(\Delta q - \Delta q^\imath + \Delta q^\jmath - \Delta q^k\right)$ and $\Delta q_\kappa = \frac{1}{4\kappa}\left(\Delta q - \Delta q^\imath - \Delta q^\jmath + \Delta q^\kappa\right)$, we obtain the quaternion Taylor series

$$f(q, q^\imath, q^\jmath, q^\kappa) = f(q_0, q_0^\imath, q_0^\jmath, q_0^\kappa) + \sum_{n=1}^{\infty} \frac{1}{n} \left[\Delta q \frac{\partial}{\partial q} + \Delta q^\imath \frac{\partial}{\partial q^\imath} + \Delta q^\jmath \frac{\partial}{\partial q^\jmath} + \Delta q^\kappa \frac{\partial}{\partial q^\kappa} + \right]^n f(q_0, q_0^\imath, q_0^\jmath, q_0^\kappa)$$

For a multivariate quaternion function, we thus have

$$f(\mathbf{q}, \mathbf{q}^\imath, \mathbf{q}^\jmath, \mathbf{q}^\kappa) = f(\mathbf{q}_0, \mathbf{q}_0^\imath, \mathbf{q}_0^\jmath, \mathbf{q}_0^\kappa) + \sum_{n=1}^{\infty} \frac{1}{n} \left[\Delta \mathbf{q}^T \frac{\partial}{\partial \mathbf{q}} + \Delta \mathbf{q}^{\imath T} \frac{\partial}{\partial \mathbf{q}^\imath} + \Delta \mathbf{q}^{\jmath T} \frac{\partial}{\partial \mathbf{q}^\jmath} + \Delta \mathbf{q}^{\kappa T} \frac{\partial}{\partial \mathbf{q}^\kappa} + \right]^n f(\mathbf{q}_0, \mathbf{q}_0^\imath, \mathbf{q}_0^\jmath, \mathbf{q}_0^\kappa)$$

Using the notation $\mathbf{v} = [\mathbf{q}, \mathbf{q}^\imath, \mathbf{q}^\jmath, \mathbf{q}^k]^T$ and $\frac{\partial f}{\partial \mathbf{q}} = [\frac{\partial f}{\partial \mathbf{q}}, \frac{\partial f}{\partial \mathbf{q}^\imath}, \frac{\partial f}{\partial \mathbf{q}^\jmath}, \frac{\partial f}{\partial \mathbf{q}^\kappa}]^T$, the first order quaternion Taylor series can be written as

$$f(\mathbf{v}) = f(\mathbf{v}_0) + \Delta \mathbf{v}^T \frac{\partial f}{\partial \mathbf{v}} \tag{D.3}$$

Appendix E

Adaptive Convex Combination Approach for the Identification of Improper Quaternion Processes

E.1 Introduction

To deal with evolving environments, we here introduce a class of quaternion-valued fractional tap-length algorithms; for rigour, this is achieved by considering the full second-order (augmented) quaternion statistics of the signal. The quaternion-valued algorithms considered are the recently introduced Quaternion Least Mean Square (QLMS) [66] and Widely Linear Quaternion Least Mean Square (WLQLMS) [67], which cater, respectively, for strictly linear and widely linear quaternion-valued processes. We also consider a collaborative architecture comprising of QLMS and WLQLMS, which equips us with the ability to identify both the model order and the second-order circular (proper) or noncircular (improper) nature of general quaternion-valued systems and processes. The collaborative combination of QLMS and WLQLMS (CCQLMS), therefore promises a flexible tool for the modelling of the generality quaternion-valued processes. We also show that the evolution of the convex mixing parameter within the proposed architecture indicates the degree of widely linear nature of a given system (degree of improperness for quaternion-valued

signals).

This Appendix is organized as follows. Section E.2 describes the operation of the proposed model order identification algorithms. The convergence of the convex mixing parameter is presented in Section E.3, followed by the steady-state analysis in Section E.4. In Section E.5, simulations supporting the proposed approaches are shown. The chapter concludes in Section E.6.

E.2 Model order identification

The proposed algorithms for the identification of the widely linear systems (and noncircularity of a signal) comprise of two parts: an adaptive finite impulse response (FIR) filter which optimises the adaptive weight coefficients, followed by the FT algorithm that adapts the tap-length of the filter to an optimal length, all performed for any given time k. We first review the FIR filter weight updates and proceed to illustrate how the FT algorithm can be exploited within quaternion-valued adaptive systems.

E.2.1 Filter weight updates

Quaternion-valued adaptive filtering algorithms are based on optimizing a real-valued cost function of quaternion variables

$$E(k) = e_r^2(k) + e_\imath^2(k) + e_\jmath^2(k) + e_\kappa^2(k) = e(k)e^*(k) = \|e(k)\|_2^2 \tag{E.1}$$

where the error $e(k) = d(k) - y(k)$, and $d(k)$ and $y(k)$ respectively the desired and output signal. The terms $e_r(k)$, $e_\imath(k)$, $e_\jmath(k)$ and $e_\kappa(k)$ denote respectively the error component in the real part, $\imath$ part, $\jmath$ part, and κ part of a quaternion variable.

Quaternion-valued adaptive filtering algorithms minimize the cost function (E.1) through a gradient descent weight update specified by

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \nabla_{\mathbf{w}} E(k) \tag{E.2}$$

where the gradient $\nabla E(k)$ is given by

$$\nabla E(k) = \frac{\partial E(k)}{\partial \mathbf{w}^*} = \frac{\partial E(k)}{\partial \mathbf{w}_r} + \frac{\partial E(k)}{\partial \mathbf{w}_\imath}\imath + \frac{\partial E(k)}{\partial \mathbf{w}_\jmath}\jmath + \frac{\partial E(k)}{\partial \mathbf{w}_\kappa}\kappa \tag{E.3}$$

The issue of a quaternion gradient has been a stumbling block in the derivation of adaptive learning algorithms, and has been solved only recently with the emergence of $\mathbb{HR}$-calculus [144].

The strictly linear Quaternion Least Mean Square (QLMS) is based on gradient-descent (E.2) and is described by [66]

$$\begin{aligned} e_{\mathrm{l}}(k) &= d(k) - y_{\mathrm{l}}(k) \\ y_{\mathrm{l}}(k) &= \mathbf{w}^T(k)\mathbf{x}(k) \\ \mathbf{w}(k+1) &= \mathbf{w}(k) + \mu\Big(2e_{\mathrm{l}}(k)\mathbf{x}^*(k) - \mathbf{x}^*(k)e_{\mathrm{l}}^*(k)\Big) \end{aligned} \tag{E.4}$$

where $\mathbf{w}(k)$ is the weight vector, $\mathbf{x}(k)$ is the filter input, $e_{\mathrm{l}}(k)$ is the QLMS error, $y_{\mathrm{l}}(k)$ is the QLMS output and μ is a real-valued learning rate.

The widely linear QLMS (WLQLMS) utilizes the widely linear model and optimises the same cost function in (E.1), and is given by [67]

$$\begin{aligned} e_{\mathrm{w}}(k) &= d(k) - y_{\mathrm{w}}(k) \\ y_{\mathrm{w}}(k) &= \mathbf{w}^{\mathrm{aT}}(k)\mathbf{x}^{\mathrm{a}}(k) \\ \mathbf{w}^{\mathbf{a}}(k+1) &= \mathbf{w}^{\mathbf{a}}(k) + \mu\Big(2e_{\mathrm{w}}(k)\mathbf{x}^{a*}(k) - \mathbf{x}^{a*}(k)e_{\mathrm{w}}^*(k)\Big) \end{aligned} \tag{E.5}$$

where $e_{\mathrm{w}}(k)$ is the WLQLMS error, $y_{\mathrm{w}}(k)$ the WLQLMS output, $\mathbf{w}^a(k)$ the augmented weight vector, and $\mathbf{x}^a(k)$ the augmented filter input vector. It was recently shown [144] that using the so-called I-gradient, it is possible to obtain the expressions for QLMS and WLQLMS which are generic extensions of real- and complex-valued counterparts. Since the performances of the algorithms in (E.4) - (E.5) and those obtained through I-gradient are identical, we shall consider the original forms.

The collaborative adaptive estimator, shown in Figure 1, consists of two indepen-

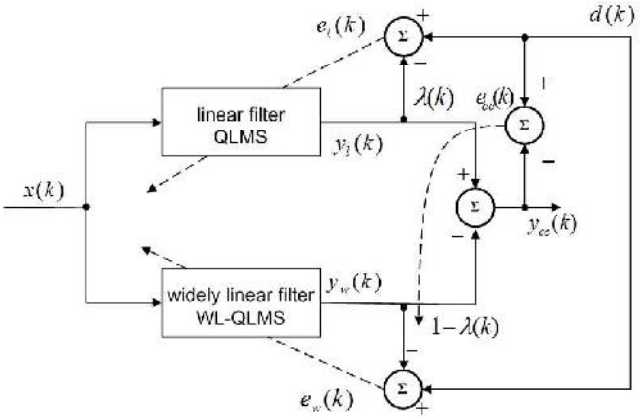


Figure E.1: Hybrid filter structure.

dent subfilters sharing the common input $\mathbf{x}(k)$ and desired signal $d(k)$. In [145], it was shown that a hybrid combination of a collaborative strictly linear and widely linear estimators has the ability to identify in real time both the system order and the degree of noncircularity of signals/systems, in the context of complex-valued widely linear modelling. Following that approach, we here employ a convex combination of QLMS and WLQLMS, termed the Convex Combination QLMS (CC-QLMS), to form the overall output $y_{cc}(k)$ given by

$$y_{\mathrm{cc}}(k) = \lambda(k)y_{\mathrm{l}}(k) + \big(1-\lambda(k)\big)y_{\mathrm{w}}(k) \tag{E.6}$$

where $\lambda(k)$ is a real-valued convex mixing parameter, whose update is governed by

$$\lambda(k+1) = \lambda(k) - \mu_\lambda \nabla_\lambda E(k) \tag{E.7}$$

where μ_λ and $\nabla_\lambda E(k)$ represent the real-valued stepsize and the error gradient.
The error gradient $\nabla_\lambda E(k)$ can be evaluated as

$$\begin{aligned}
\nabla_\lambda E(k) &= e_{\mathrm{cc}}(k)\frac{\partial e^*_{\mathrm{cc}}(k)}{\partial \lambda(k)} + \frac{\partial e_{\mathrm{cc}}(k)}{\partial \lambda(k)} e^*_{\mathrm{cc}}(k) \\
&= e_{\mathrm{cc}}(k)\big(y_{\mathrm{l}}(k) - y_{\mathrm{w}}(k)\big)^* \\
&+ \big(y_{\mathrm{l}}(k) - y_{\mathrm{w}}(k)\big)e^*_{\mathrm{cc}}(k) \\
&= 2\Re\big\{e_{\mathrm{cc}}(k)\big(y_{\mathrm{l}}(k) - y_{\mathrm{w}}(k)\big)^*\big\}
\end{aligned} \tag{E.8}$$

where $e_{\text{cc}}(k) = d(k) - y_{\text{cc}}(k)$ is the error of the CC-QLMS algorithm, and symbol $\Re\{\cdot\}$ denotes the real part of a quaternion variable. Finally, the weight update of the convex mixing parameter $\lambda(k)$ has the form

$$\lambda(k+1) = \lambda(k) - \mu_\lambda \left(2\Re\left\{ e_{\text{cc}}(k) \big(y_{\text{l}}(k) - y_{\text{w}}(k) \big)^* \right\} \right) \tag{E.9}$$

Due to the convex nature of the CC-QLMS and given the range of the mixing parameter $\lambda(k)$ (within $[0,1]$), the CC-QLMS converges as long as one of the subsystems in Figure 1 converges [146]; with this in mind, in the simulations the value of $\lambda(k)$ is hard bounded to $\lambda \in [0,1]$.

E.2.2 Tap length adaptation

The adaptation of model order in real time, also termed "tap-length adaptation" is enabled by the adaptive nature of the models considered, and has the ability to identify the evolving changes in signal nature for time-varying system parameters. The tap-length adaptation is governed by the FT algorithm given by [147]

$$\eta_f(k+1) = (\eta_f(k) - \alpha) - \gamma \cdot \left[\left(E_p^{(p)}(k) \right) - \left(E_{p-\Delta}^{(p)}(k) \right) \right] \tag{E.10}$$

where η_f is the pseudo fractional tap-length which can take only positive real value, α and γ are the leaky factor and tap-length learning rate, small positive real parameters that satisfy $\alpha \ll \gamma$. Symbols $E_p^{(p)}(k)$ and $E_{p-\Delta}^{(p)}(k)$ denote respectively the instantaneous square errors for the tap-lengths of order p and $\big(p-\Delta\big)$, symbol $p(k)$ denotes the "true" tap-length at a discrete time instant 'k', and Δ is a real positive integer such that $\min\{p(k)-\Delta\} > 0$. The instantaneous square output errors for filters of lengths p and $(p-\Delta)$ are given by

$$\begin{aligned} E_p^{(p)}(k) &= \big(e_p^{(p)}(k)\big)\big(e_p^{(p)}(k)\big)^* \\ E_{p-\Delta}^{(p)}(k) &= \big(e_{p-\Delta}^{(p)}(k)\big)\big(e_{p-\Delta}^{(p)}(k)\big)^* \end{aligned} \tag{E.11}$$

and are based on the errors $e_p^{(p)}(k)$ and $e_{p-\Delta}^{(p)}(k)$. These errors can be evaluated as

$$e_q^{(p)}(k) = d(k) - y_q^{(p)}(k) = d(k) - \mathbf{w}_q^{(p)T}(k)\mathbf{x}_q^{(p)}(k) \tag{E.12}$$

where $1 \leq q \leq p$, while $\mathbf{w}_q^{(p)}(k)$ and $\mathbf{x}_q^{(p)}(k)$ are the vectors consisting of the first q coefficients of $\mathbf{w}^{(p)}(k)$ and $\mathbf{x}^{(p)}(k)$, respectively.

To calculate the optimal model length, the tap-length parameter $p(k)$ is made adaptive according to [147]

$$p(k+1) = \begin{cases} \lfloor \eta_f(k) \rfloor, \ |p(k) - \eta_f(k)| \geq \delta \\ p(k), \ \text{otherwise} \end{cases} \tag{E.13}$$

where δ is a predefined integer threshold and symbol $\lfloor \cdot \rfloor$ denotes the floor operator. The coefficient adaptation for the FT collaborative architecture in Figure 1 is summarised in Algorithm 5.

Algorithm 5: Adaptation of the collaborative adaptive system identification architecture

---

Filter Weight Algorithms
CC-QLMS, QLMS:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\Big(2e_{\mathrm{l}}(k)\mathbf{x}^*(k) - \mathbf{x}^*(k)e_{\mathrm{l}}^*(k)\Big)$$

CC-QLMS, WLQLMS:

$$\mathbf{w}^{\mathbf{a}}(k+1) = \mathbf{w}^{\mathbf{a}}(k) + \mu\Big(2e_{\mathrm{w}}(k)\mathbf{x}^{a*}(k) - \mathbf{x}^{a*}(k)e_{\mathrm{w}}^*(k)\Big)$$

CC-QLMS:

$$\lambda(k+1) = \lambda(k) - \mu_\lambda\Big(2\Re\big\{e_{\mathrm{cc}}(k)\big(y_{\mathrm{l}}(k) - y_{\mathrm{w}}(k)\big)^*\big\}\Big)$$

Fractional Tap-Length Algorithm

$$\eta_f(k+1) = (\eta_f(k) - \alpha) - \gamma \cdot \left[\left(E_p^{(p)}(k)\right) - \left(E_{p-\Delta}^{(p)}(k)\right)\right]$$

$$p(k+1) = \begin{cases} \lfloor \eta_f(k) \rfloor, \ |p(k) - \eta_f(k)| \geq \delta \\ p(k), \ \text{otherwise} \end{cases}$$

E.3 Convergence of the mixing parameter

To address the convergence of the mixing parameter λ within CC-QLMS when processing both strictly linear and widely linear signals, consider first a strictly linear model given by

$$d(k) = \mathbf{w}^{oT}\mathbf{x}(k) + v(k) \tag{E.14}$$

where $\mathbf{w}^o$ the optimal weight coefficients and v is a $\mathbb{H}$-circular quaternion-valued quadruply white Gaussian noise.

At the steady state, we employ the standard assumptions that $\mathbf{w}^o = \mathbf{w} = \mathbf{g}$ and due to circularity $\mathbf{h} = \mathbf{u} = \mathbf{v} = \mathbf{0}$. Enforcing these assumptions, the overall instantenous error e_{cc} at the steady-state becomes

$$\begin{aligned} e_{\mathrm{cc}}(k) &= \mathbf{w}^{oT}\mathbf{x}(k) + v(k) - \Big(\lambda(k)\big(\mathbf{w}^{oT}(k)\mathbf{x}(k)\big) \\ &\quad + \big(1-\lambda(k)\big)\big(\mathbf{w}^{oT}(k)\mathbf{x}(k)\Big) \\ &= v(k) \end{aligned} \tag{E.15}$$

Substituting (E.15) into the expression for evolution of the mixing parameter λ in (E.9) will yield

$$\begin{aligned} \lambda(k+1) &= \lambda(k) - \mu_\lambda\Big(2\mathcal{R}\big\{e_{\mathrm{cc}}(k)\big(y_{\mathrm{l}}(k) - y_{\mathrm{w}}(k)\big)^*\big\}\Big) \\ &= \lambda(k) - \mu_\lambda\Big(2\mathcal{R}\big\{v(k)\big(\mathbf{w}^T(k)\mathbf{x}(k) \\ &- \mathbf{g}^T(k)\mathbf{x}(k)\big)^*\big\}\Big) \\ &= \lambda(k) \end{aligned} \tag{E.16}$$

where symbol $\Re\{\cdot\}$ denotes the real part of a quaternion variable.

Since the QLMS converges faster than the WLQLMS (due to the smaller number of weight terms), the CC-QLMS will favour the QLMS trained subfilter leading to $\lambda(k) \to 1$.

Remark#31: The mixing parameter $\lambda \to 1$ within the CC-QLMS indicates a strictly

linear quaternion-valued model, or equivalently a second order circular (proper) signals for which the power in all the components are equal. Current models all assume properness, however, this is not justified for real world data.

Consider next the widely linear teaching signal (model) given by

$$d(k) = \mathbf{g}^{oT}\mathbf{x}(k) + \mathbf{h}^{oT}\mathbf{x}^{\imath}(k) + \mathbf{u}^{oT}\mathbf{x}^{\jmath}(k) + \mathbf{v}^{oT}(k)\mathbf{x}^{\kappa}(k) + v(k) \tag{E.17}$$

where $\mathbf{g}^o$, $\mathbf{h}^o$, $\mathbf{u}^o$ and $\mathbf{v}^o$ are the optimal weight coefficients. At the steady-state, in this case $\mathbf{g} = \mathbf{g}^o \neq \mathbf{w}$ and $\mathbf{h} = \mathbf{h}^o$, $\mathbf{u} = \mathbf{u}^o$. $\mathbf{v} = \mathbf{v}^o$. Proceeding in a similar manner as above, the error e_{cc} becomes

$$\begin{aligned}
e_{\mathrm{cc}}(k) &= \mathbf{g}^{oT}\mathbf{x}(k) + \mathbf{h}^{oT}\mathbf{x}^{\imath}(k) + \mathbf{u}^{oT}\mathbf{x}^{\jmath}(k) \\
&+ \mathbf{v}^{oT}(k)\mathbf{x}^{\kappa}(k) + v(k) - \Big(\lambda(k)\big(\mathbf{w}^T(k)\mathbf{x}(k)\big) \\
&+ \big(1-\lambda(k)\big)\big(\mathbf{g}^{oT}(k)\mathbf{x}(k) + \mathbf{h}^{oT}\mathbf{x}^{\imath}(k) \\
&+ \mathbf{u}^{oT}\mathbf{x}^{\jmath}(k) + \mathbf{v}^{oT}\mathbf{x}^{\kappa}(k)\big)\Big) \\
&= v(k) + \lambda(k)\big(\mathbf{g}^{oT}(k)\mathbf{x}(k) + \mathbf{h}^{oT}\mathbf{x}^{\imath}(k) \\
&+ \mathbf{u}^{oT}\mathbf{x}^{\jmath}(k) + \mathbf{v}^{oT}\mathbf{x}^{\kappa}(k) - \mathbf{w}^T(k)\mathbf{x}(k)\big)
\end{aligned} \tag{E.18}$$

Replacing e_{cc} in (E.9) with (E.18) will result in the update of the mixing parameter

$$\begin{aligned}
\lambda(k+1) &= \lambda(k) - \mu_\lambda\Big(2\mathcal{R}\big\{e_{\mathrm{cc}}(k)\big(y_{\mathrm{l}}(k) - y_{\mathrm{w}}(k)\big)^*\big\}\Big) \\
&= \lambda(k) - \mu_\lambda\Big(2\mathcal{R}\big\{\big(v(k) + \lambda(k)\big(\mathbf{g}^T(k)\mathbf{x}(k) \\
&+ \mathbf{h}^T\mathbf{x}^{\imath}(k) + \mathbf{u}^T\mathbf{x}^{\jmath}(k) + \mathbf{v}^T\mathbf{x}^{\kappa}(k) \\
&- \mathbf{w}^T(k)\mathbf{x}(k)\big)\big)\big(\mathbf{w}^T(k)\mathbf{x}(k) - \mathbf{g}^T(k)\mathbf{x}(k) \\
&- \mathbf{h}^T\mathbf{x}^{\imath}(k) - \mathbf{u}^T\mathbf{x}^{\jmath}(k) - \mathbf{v}^T\mathbf{x}^{\kappa}(k)\big)^*\big\}\Big)
\end{aligned} \tag{E.19}$$

To simplify this further, we shall employ the standard "independence" assumptions [20]

- the filter weights are independent of one another;
- the error and the input vector are statistically independent of one another.

Enforcing these assumptions and applying the statistical expectation operator, (E.19) can be further simplified into

$$\begin{aligned} E\{\lambda(k+1)\} &= E\{\lambda(k)\} - 2\mu_\lambda E\{\lambda(k)\}\Big(\mathbf{g}^{oT}\mathbf{R}_{xx}\mathbf{g}^{o*} \\ &+ \mathbf{h}^{oT}\mathbf{P}_{xx}\mathbf{h}^{o*} + \mathbf{u}^{oT}\mathbf{S}_{xx}\mathbf{u}^{o*} \\ &+ \mathbf{v}^{oT}\mathbf{T}_{xx}\mathbf{v}^{o*} + \mathbf{w}^{T}\mathbf{R}_{xx}\mathbf{w}^{*}\Big) \end{aligned} \tag{E.20}$$

For further insight, define [64]

$$\mathbf{R}_{xx} = \mathbf{P}_{xx} = \mathbf{S}_{xx} = \mathbf{T}_{xx} = \sigma_x^2\mathbf{I} \tag{E.21}$$

where σ_x^2 is the variance of the signal, and plug into (E.20) to give the evolution of λ in the form

$$\begin{aligned} E\{\lambda(k+1)\} &= E\{\lambda(k)\}\Big(1 - 2\mu_\lambda\|\mathbf{g}\|^2\sigma_x + \|\mathbf{h}\|^2\sigma_x \\ &+ \|\mathbf{u}\|^2\sigma_x + \|\mathbf{v}\|^2\sigma_x + \|\mathbf{w}\|^2\sigma_x\Big) \\ &= E\{\lambda(k)\}\Big(1 - 2\mu_\lambda\Delta e^2\Big) \end{aligned} \tag{E.22}$$

where Δe^2 denotes the performance advantage of the widely linear model over the standard linear model. Since both the learning rate μ_λ and Δe^2 are positive, for the widely linear model or improper signal, the mixing parameter $\lambda(k)$ converges towards zero, favouring the WLQLMS subfilter within the collaborative architecture in Figure 1.

Remark#32: The convergence of the mixing parameter within CC-QLMS to $\lambda \to 0$ indicates a quaternion-valued widely linear model or equivalently a noncircular signal.

E.4 Steady-state analysis of FT based algorithms

To illustrate the robustness of the proposed approach, and to convey the information about both the second-order model order and its linear or widely linear nature, we next provide a rigorous steady-state analysis of the of the FT-QLMS, FT-WLQLMS and FT-CCQLMS

algorithms for two cases: a) the desired signal is widely linear and noncircular b) the desired signal is strictly linear and circular. We first consider the case of widely linear teaching signal and the FT-WLQLMS algorithm, where the desired (teaching) signal $d(k)$ is defined as

$$\begin{aligned} d(k) = &\mathbf{g}_{Lopt}^{oT}\mathbf{x}_{Lopt}(k) + \mathbf{h}_{Lopt}^{oT}\mathbf{x}_{Lopt}^{\imath}(k) + \mathbf{u}_{Lopt}^{oT}\mathbf{x}_{Lopt}^{\jmath}(k) \\ &+ \mathbf{v}_{Lopt}^{oT}(k)\mathbf{x}_{Lopt}^{\kappa}(k) + v(k) \end{aligned} \tag{E.23}$$

where $\mathbf{g}_{Lopt}^{o}$, $\mathbf{h}_{Lopt}^{o}$, $\mathbf{u}_{Lopt}^{o}$ and $\mathbf{v}_{Lopt}^{o}$ are the optimal weight coefficients of the optimal tap lengths *Lopt* of the widely linear model, and $v(k)$ is a $\mathbb{H}$-circular quaternion-valued quadruply white Gaussian noise.

The output of the FT-WLQLMS algorithm is then given as

$$y_{\mathrm{w}}(k) = \underbrace{\mathbf{g}^{T}(k)\mathbf{x}(k)}_{\text{standard part}} + \underbrace{\mathbf{h}^{T}(k)\mathbf{x}^{\imath}(k) + \mathbf{u}^{T}(k)\mathbf{x}^{\jmath}(k) + \mathbf{v}^{T}(k)\mathbf{x}^{\kappa}(k)}_{\text{augmented part}} \tag{E.24}$$

and the output error $e_{\mathrm{w}}(k)$ can be expressed in terms of the optimal tap weights as

$$\begin{aligned} e_{\mathrm{w}}(k) &= \mathbf{g}^{oT}\mathbf{x}_{Lopt}(k) + \mathbf{h}^{oT}\mathbf{x}_{Lopt}^{\imath}(k) + \mathbf{u}^{oT}\mathbf{x}_{Lopt}^{\jmath}(k) \\ &+ \mathbf{v}^{oT}\mathbf{x}_{Lopt}^{\kappa}(k) + v(k) - \mathbf{g}^{T}(k)\mathbf{x}(k) - \mathbf{h}^{T}(k)\mathbf{x}^{\imath}(k) \\ &- \mathbf{u}^{T}(k)\mathbf{x}^{\jmath}(k) - \mathbf{v}^{T}(k)\mathbf{x}^{\kappa}(k) \end{aligned} \tag{E.25}$$

Proceeding in a manner similar to the analysis in [148], the optimal weight vector coefficients can be split into three parts

$$\begin{aligned} \mathbf{g}_{Lopt}^{o} &= \begin{bmatrix} \mathbf{g}^{\prime o} \\ \mathbf{g}^{\prime\prime o} \\ \mathbf{g}^{\prime\prime\prime o} \end{bmatrix} \quad \mathbf{h}_{Lopt}^{o} = \begin{bmatrix} \mathbf{h}^{\prime o} \\ \mathbf{h}^{\prime\prime o} \\ \mathbf{h}^{\prime\prime\prime o} \end{bmatrix} \\ \mathbf{u}_{Lopt}^{o} &= \begin{bmatrix} \mathbf{u}^{\prime o} \\ \mathbf{u}^{\prime\prime o} \\ \mathbf{u}^{\prime\prime\prime o} \end{bmatrix} \quad \mathbf{v}_{Lopt}^{o} = \begin{bmatrix} \mathbf{v}^{\prime o} \\ \mathbf{v}^{\prime\prime o} \\ \mathbf{v}^{\prime\prime\prime o} \end{bmatrix} \end{aligned} \tag{E.26}$$

where $\mathbf{g}'^o$, $\mathbf{h}'^o$, $\mathbf{u}'^o$, $\mathbf{v}'^o$ are the coefficients modelled by tap-length 1 to $(p-\Delta)$, $\mathbf{g}''^o$, $\mathbf{h}''^o$, $\mathbf{u}''^o$, $\mathbf{v}''^o$ are the coefficients modelled by the tap-length $(p-\Delta+1)$ to p, and $\mathbf{g}'''^o$, $\mathbf{h}'''^o$, $\mathbf{u}'''^o$, $\mathbf{v}'''^o$ are the undermodelled coefficients.

For convenience, we denote the coefficient error vectors of the FT-WLQLMS as

$$\begin{aligned} \tilde{\mathbf{g}}(k) &= \mathbf{g}^o - \mathbf{g}_p(k); \quad \tilde{\mathbf{h}}(k) = \mathbf{h}^o - \mathbf{h}_p(k) \\ \tilde{\mathbf{u}}(k) &= \mathbf{u}^o - \mathbf{u}_p(k); \quad \tilde{\mathbf{v}}(k) = \mathbf{v}^o - \mathbf{v}_p(k) \end{aligned} \tag{E.27}$$

where $\mathbf{g}_p(k)$, $\mathbf{h}_p(k)$, $\mathbf{u}_p(k)$ and $\mathbf{v}_p(k)$ are the weight vectors of length p.

Similar to (E.26), the weight error vectors can also be split up into three parts

$$\begin{aligned} \tilde{\mathbf{g}}(k) &= \begin{bmatrix} \tilde{\mathbf{g}}'(k) \\ \tilde{\mathbf{g}}''(k) \\ \tilde{\mathbf{g}}'''(k) \end{bmatrix} \tilde{\mathbf{h}}(k) = \begin{bmatrix} \tilde{\mathbf{h}}'(k) \\ \tilde{\mathbf{h}}''(k) \\ \tilde{\mathbf{h}}'''(k) \end{bmatrix} \\ \tilde{\mathbf{u}}(k) &= \begin{bmatrix} \tilde{\mathbf{u}}'(k) \\ \tilde{\mathbf{u}}''(k) \\ \tilde{\mathbf{u}}'''(k) \end{bmatrix} \tilde{\mathbf{v}}(k) = \begin{bmatrix} \tilde{\mathbf{v}}'(k) \\ \tilde{\mathbf{v}}''(k) \\ \tilde{\mathbf{v}}'''(k) \end{bmatrix} \end{aligned} \tag{E.28}$$

We can now rewrite the errors $e_p^{(p)}(k)$ and $e_{p-\Delta}^{(p)}(k)$ to be (the time index 'k' has been dropped due to space limitations)

$$\begin{aligned} e_p^{(p)} &= \begin{bmatrix} \tilde{\mathbf{g}}' \\ \tilde{\mathbf{g}}'' \\ \mathbf{g}'''^{o} \end{bmatrix}^T \begin{bmatrix} \mathbf{x}' \\ \mathbf{x}'' \\ \mathbf{x}''' \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{h}}' \\ \tilde{\mathbf{h}}'' \\ \mathbf{h}'''^{o} \end{bmatrix}^T \begin{bmatrix} \mathbf{x}'^{h} \\ \mathbf{x}''^{h} \\ \mathbf{x}'''^{h} \end{bmatrix} \\ &+ \begin{bmatrix} \tilde{\mathbf{u}}' \\ \tilde{\mathbf{u}}'' \\ \mathbf{u}'''^{o} \end{bmatrix}^T \begin{bmatrix} \mathbf{x}'^{\jmath} \\ \mathbf{x}''^{\jmath} \\ \mathbf{x}'''^{\jmath} \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{v}}' \\ \tilde{\mathbf{v}}'' \\ \mathbf{v}'''^{o} \end{bmatrix}^T \begin{bmatrix} \mathbf{x}'^{\kappa} \\ \mathbf{x}''^{\kappa} \\ \mathbf{x}'''^{\kappa} \end{bmatrix} + \upsilon \end{aligned} \tag{E.29}$$

$$\begin{aligned} e_{p-\Delta}^{(p)} &= \begin{bmatrix} \tilde{\mathbf{g}}' \\ \mathbf{g}''^{o} \\ \mathbf{g}'''^{o} \end{bmatrix}^T \begin{bmatrix} \mathbf{x}' \\ \mathbf{x}'' \\ \mathbf{x}''' \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{h}}' \\ \mathbf{h}''^{o} \\ \mathbf{h}'''^{o} \end{bmatrix}^T \begin{bmatrix} \mathbf{x}'^{h} \\ \mathbf{x}''^{h} \\ \mathbf{x}'''^{h} \end{bmatrix} \\ &+ \begin{bmatrix} \tilde{\mathbf{u}}' \\ \mathbf{u}''^{o} \\ \mathbf{u}'''^{o} \end{bmatrix}^T \begin{bmatrix} \mathbf{x}'^{\jmath} \\ \mathbf{x}''^{\jmath} \\ \mathbf{x}'''^{\jmath} \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{v}}' \\ \mathbf{v}''^{o} \\ \mathbf{v}'''^{o} \end{bmatrix}^T \begin{bmatrix} \mathbf{x}'^{\kappa} \\ \mathbf{x}''^{\kappa} \\ \mathbf{x}'''^{\kappa} \end{bmatrix} + \upsilon \end{aligned} \tag{E.30}$$

In order to ensure mathematical tractability, we shall employ the following usual independence assumptions [148]:

- Both the input signal $\mathbf{x}(k)$ and the noise $v(k)$ are i.i.d. zero mean white jointly Gaussian with the respective variances σ_x^2 and σ_v^2;
- At the steady state, the input signal $\mathbf{x}(k)$ is independent of the weight vectors;
- The tap-length parameter has converged at steady-state, hence $E\{\eta_f(k+1)\} = E\{\eta_f(k)\}$, leading to the undermodelled error vectors vanishing.

We proceed by applying the statistical expectation operator to the steady-state MSE to yield

$$E\left\{\left(E_p^{(p)}(k)\right) - \left(E_{p-\Delta}^{(p)}(k)\right)\right\} < |\frac{\alpha}{\gamma}| \tag{E.31}$$

Following the definition in (E.11), we can rewrite (E.31) to give

$$\begin{aligned}
&E\{\|\tilde{\mathbf{g}}''^{T}(k)\mathbf{x}''(k)\|_2^2 + \|\tilde{\mathbf{h}}''^{T}(k)\mathbf{x}''^{\imath}(k)\|_2^2 \\
&+\|\tilde{\mathbf{u}}''^{T}(k)\mathbf{x}''^{\jmath}(k)\|_2^2 + \|\tilde{\mathbf{v}}''^{T}(k)\mathbf{x}''^{\kappa}(k)\|_2^2 \\
&-\|\mathbf{g}''^{oT}(k)\mathbf{x}''(k)\|_2^2 - \|\mathbf{h}''^{oT}(k)\mathbf{x}''^{\imath}(k)\|_2^2 \\
&-\|\mathbf{u}''^{oT}(k)\mathbf{x}''^{\jmath}(k)\|_2^2 - \|\mathbf{v}''^{oT}(k)\mathbf{x}''^{\kappa}(k)\|_2^2\} < |\frac{\alpha}{\gamma}|
\end{aligned} \tag{E.32}$$

Remark#33: The FT-WLQLMS incorporates the errors from both the standard and augmented parts of the quaternion widely linear model in adapting the tap-length, thus ensuring efficient modelling of the general widely linear quaternion-valued systems, that is, incorporating both those with strictly and widely linear system dynamics.
In order to obtain the steady-state of the FT-QLMS algorithm, we first set the augmented part in (E.24) to zero which gives

$$y_{\mathrm{l}}(k) = \mathbf{w}^T(k)\mathbf{x}(k) \tag{E.33}$$

Proceeding in a similar fashion to obtain the steady-state of FT-WLQLMS yields

$$E\{\|\mathbf{w}''^{T}(k)\tilde{\mathbf{x}}''(k)\|_2^2 - \|\mathbf{g}''^{oT}(k)\mathbf{x}''(k)\|_2^2\} < |\frac{\alpha}{\gamma}| \tag{E.34}$$

Remark#34: The FT-QLMS only considers the error from the standard part of the quaternion widely linear model in adapting the tap-length, proving to be inadequate for the modelling of widely linear quaternion-valued systems and improper quaternion-valued processes, that is those with different powers in the components, a typical case in real world scenarios.

To derive the steady-state of the FT-CCQLMS algorithm, consider the output of FT-CCQLMS, given by

$$\begin{aligned}
y_{\mathrm{cc}}(k) = \lambda(k)\mathbf{w}^T(k)\mathbf{x}(k) + \Big(1-\lambda(k)\Big)\Big(&\mathbf{g}^T(k)\mathbf{x}(k) \\
&+ \mathbf{h}^T(k)\mathbf{x}^{\imath}(k) + \mathbf{u}^T(k)\mathbf{x}^{\jmath}(k) + \mathbf{v}^T(k)\mathbf{x}^{\kappa}(k)\Big)
\end{aligned} \tag{E.35}$$

Proceeding in a similar manner as above to obtain FT-WLQLMS and FT-QLMS, the final steady-state becomes

$$
\begin{aligned}
&E\Bigg\{\Big(1-\lambda(k)\Big)\Big(\|\tilde{\mathbf{g}}^{\prime\prime T}(k)\mathbf{x}^{\prime\prime}(k)\|_2^2+\|\tilde{\mathbf{h}}^{\prime\prime T}(k)\mathbf{x}^{\prime\prime\imath}(k)\|_2^2\\
&+\|\tilde{\mathbf{u}}^{\prime\prime T}(k)\mathbf{x}^{\prime\prime\jmath}(k)\|_2^2+\|\tilde{\mathbf{v}}^{\prime\prime T}(k)\mathbf{x}^{\prime\prime\kappa}(k)\|_2^2\Big)+\lambda(k)\\
&\cdot\|\tilde{\mathbf{w}}^{\prime\prime T}(k)\mathbf{x}^{\prime\prime}(k)\|_2^2-\|\mathbf{g}^{\prime\prime oT}(k)\mathbf{x}^{\prime\prime}(k)\|_2^2-\|\mathbf{h}^{\prime\prime oT}(k)\mathbf{x}^{\prime\prime\imath}(k)\|_2^2\\
&-\|\mathbf{u}^{\prime\prime oT}(k)\mathbf{x}^{\prime\prime\jmath}(k)\|_2^2-\|\mathbf{v}^{\prime\prime oT}(k)\mathbf{x}^{\prime\prime\kappa}(k)\|_2^2\Bigg\}<|\frac{\alpha}{\gamma}|
\end{aligned}
\tag{E.36}
$$

As stated in Remark 32, for the processing of widely linear systems, $\lambda \to 0$ simplifying (E.36) to an expression similar to the steady-state of FT-WLQLMS in (E.32).

Remark#35: For widely linear systems (improper signals), as $\lambda \to 0$, the steady-state of FT-CCQLMS is similar to that of the FT-WLQLMS when identifying widely linear systems.

Consider next a strictly linear model given by

$$
d(k)=\mathbf{w}_{Lopt}^{oT}\mathbf{x}_{Lopt}(k)+v(k) \tag{E.37}
$$

which gives a similar steady-state expression for FT-QLMS and FT-WLQLMS, given by

$$
\begin{aligned}
\textbf{FT-QLMS}: &E\{\|\tilde{\mathbf{w}}^{\prime\prime T}(k)\mathbf{x}^{\prime\prime}(k)\|_2^2\\
&-\|\mathbf{w}^{\prime\prime oT}(k)\mathbf{x}^{\prime\prime}(k)\|_2^2\}<|\frac{\alpha}{\gamma}|
\end{aligned}
\tag{E.38}
$$

$$
\begin{aligned}
\textbf{FT-WLQLMS}: &E\{\|\tilde{\mathbf{g}}^{\prime\prime T}(k)\mathbf{x}^{\prime\prime}(k)\|_2^2\\
&-\|\mathbf{w}^{\prime\prime oT}(k)\mathbf{x}^{\prime\prime}(k)\|_2^2\}<|\frac{\alpha}{\gamma}|
\end{aligned}
\tag{E.39}
$$

Remark#36: Both the FT-QLMS and FT-WLQLMS take into account the error from the standard (strictly linear) part of the quaternion linear model in adapting their tap-lengths, demonstrating their suitability for the modelling of strictly linear quaternion-valued systems (circular signals).

Similarly, the steady-state expression for FT-CCQLMS becomes

$$E\Bigg\{\lambda(k)\|\tilde{\mathbf{w}}''^{T}(k)\mathbf{x}''(k)\|_2^2+\Big(1-\lambda(k)\Big)\cdot\Bigg(\|\tilde{\mathbf{g}}''^{T}(k)\mathbf{x}''(k)\|_2^2+\|\tilde{\mathbf{h}}''^{T}(k)\mathbf{x}''^{\imath}(k)\|_2^2+\|\tilde{\mathbf{u}}''^{T}(k)\mathbf{x}''^{\jmath}(k)\|_2^2+\|\tilde{\mathbf{v}}''^{T}(k)\mathbf{x}''^{\kappa}(k)\|_2^2\Bigg)-\|\mathbf{w}''^{oT}(k)\mathbf{x}''(k)\|_2^2\Bigg\}<|\frac{\alpha}{\gamma}| \tag{E.40}$$

For optimal processing of the linear model, $\lambda \approx 1$ (as stated by Remark 31) resulting in a similar expression to the steady-state of FT-QLMS in (E.38).

Remark#37: The FT-CCQLMS will have similar behaviour to the FT-QLMS in (E.34) for the modelling of strictly linear quaternion-valued systems, when $\lambda \to 1$.

The analysis sets the scene for the assessment of the changes in some fundamental parameters of quaternion-valued systems and signals (widely linear nature, noncircularity) in real-time, and in time-varying environments, a typical case in real world applications.

E.5 Simulations

Simulations were conducted in the system identification setting and performances of FT-QLMS, FT-WLQLMS and FT-CCQLMS were evaluated for a range of systems both benchmark and real world. For benchmark systems, the quaternion quadruply white circular Gaussian noise (QWGN) defined by

$$\epsilon(k) = \epsilon_a(k) + \epsilon_b(k)\imath + \epsilon_c(k)\jmath + \epsilon_d(k)\kappa \tag{E.41}$$

served as the driving input, where ϵ_a, ϵ_b, ϵ_c and ϵ_d are realisations of real-valued independent white Gaussian noises (WGN).

The QWGN was first fed through a filter defined by $h(k) = 0.35\epsilon(k) + \epsilon(k-1) + 0.35\epsilon(k-2)$ which slows down the convergence of the algorithms. The output of $h(k)$ was

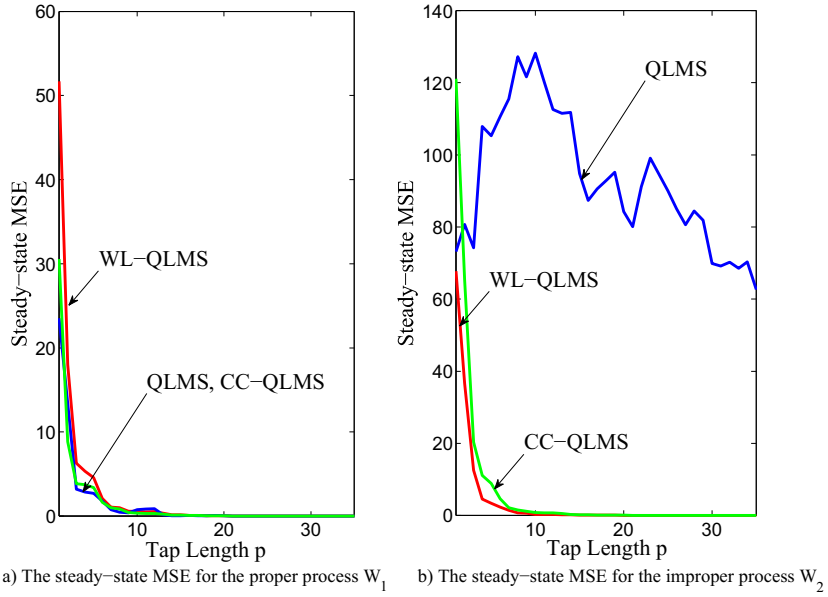


Figure E.2: The steady-state MSE for the circular process W_1 and the noncircular process W_2 with respect to the tap-length (model order).

then fed into the systems defined by

$$\begin{aligned} W_1(k) &= 1.79W_1(k-1) - 1.85W_1(k-2) \\ &\quad + 1.27W_1(k-3) - 0.41W_1(k-4) + \epsilon(k) \end{aligned} \tag{E.42}$$

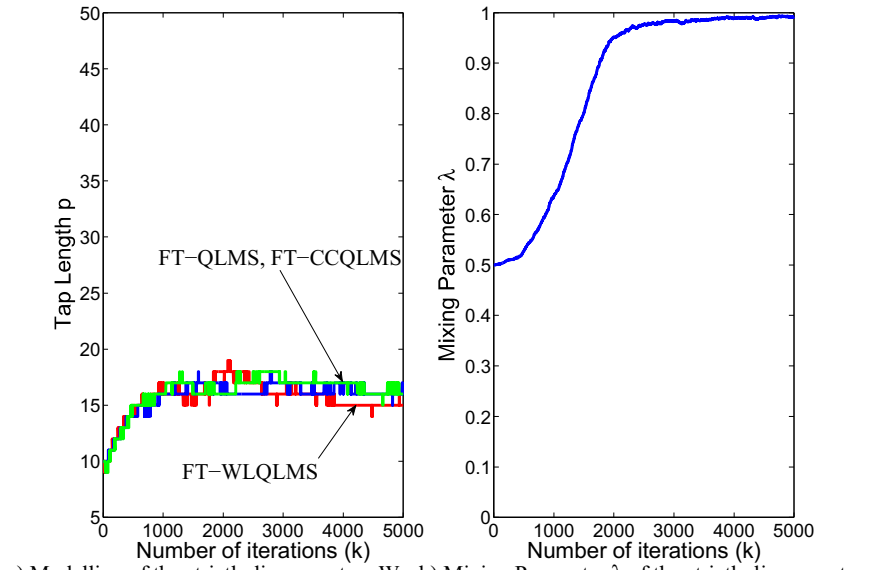
$$\begin{aligned} W_2(k) &= 1.79W_2(k-1) - 1.85W_2(k-2) \\ &\quad + 1.27W_2(k-3) - 0.41W_2(k-4) + \epsilon(k) \\ &\quad + 0.5\epsilon^*(k) + 0.9\epsilon^*(k-1) \end{aligned} \tag{E.43}$$

where W_1 is a strictly linear AR(4) system [45] and W_2 a widely linear WLAR(4) system [149]. System W_2 is constructed by combining W_1 with the augmented part of W given by [150]

$$W(k) = \mathrm{e}^{\imath}W(k-1) + \epsilon(k) + 0.5\epsilon^*(k) + 0.9\epsilon^*(k-1) \tag{E.44}$$

where $\imath$ is the imaginary unit.

This way the system/signal W_1 was circular and W_2 widely linear (improper). The real world process were 3D wind and 3D inertial body motion data.



a) Modelling of the strictly linear system W_1 b) Mixing Parameter λ of the strictly linear system W_1

Figure E.3: The evolution of the optimal filter length parameter p and the mixing parameter λ for the modelling of the strictly linear system W_1.

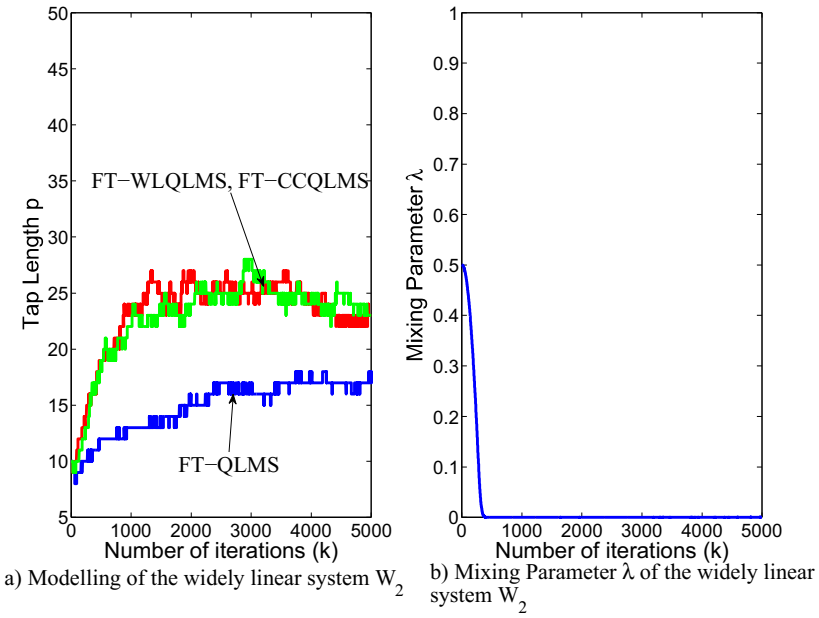
E.5.1 Optimal tap-length

The optimal tap-lengths for both systems were determined by the steady-state MSE criterion [147]

$$\bar{\varepsilon}(k) = \lambda_{MSE}\bar{\varepsilon}(k-1) + (1-\lambda_{MSE})E(k) \tag{E.45}$$

where $\bar{\varepsilon}$ is the estimated steady-state MSE and $\lambda_{MSE} = 0.9$.

Figure E.2 depicts the steady-state MSE for both the strictly linear system W_1 and widely linear system W_2, using the QLMS, WLQLMS and CCQLMS algorithms with $\mu = 10^{-3}$. From Figure E.2a), it can be seen that all the three MSE curves were monotonically non-increasing functions of the tap-length and as such the optimal tap-lengths for all algorithms were found to be $p_0 = \{15, 16\}$. Figure E.2b) shows that the shape of the MSE curve for QLMS did not asymptotically converge, thus proving the inability of the strictly linear QLMS to model the widely linear system W_2. On the other hand, the MSE curves for the WLQLMS and CC-QLMS converged indicating their ability to model W_2 for which the optimal tap-length was found to be $p_0 = \{21, 22\}$. The optimal tap-lengths for both systems were not a single integer due to the use of feedforward filters, which can



a) Modelling of the widely linear system W_2

b) Mixing Parameter λ of the widely linear system W_2

Figure E.4: The evolution of the optimal filter length parameter p and mixing parameter λ for the modelling of the widely linear system W_2.

only give approximations of the considered autoregressive (AR) infinite impulse response (IIR) system.

E.5.2 Modelling of quaternion-valued systems

Figure E.3 depicts the evolution of the optimal tap length parameter p for the FT-QLMS, FT-WLQLMS and FT-CCQLMS algorithms when employed for the modelling of strictly linear AR(4) system W_1 along with the evolution of the mixing parameter λ of FT-CCQLMS. These algorithms were initialized with the following parameters: $\alpha = 0.03$, $\gamma = 1$, δ=1, Δ=4, $\mu = 1 \times 10^{-5}$, $\mu_l = 5 \times 10^{-4}$, the initial mixing parameter $\lambda(0) = 0.5$ and the initial tap length $p(0) = 10$. From Figure E.3a), it is evident that the performances of all three algorithms considered were similar as they converged to the optimal tap-length at around the same number of iterations. This conforms with Remark 36 and Remark 37, which give justification for their similar performances. Figure E.3b) shows that the mixing parameter λ of FT-CCQLMS $\lambda \to 1$ for the modelling of linear system W_1, conforming with Remark 31.

Similarly, Figure E.4 shows the modelling for the widely linear system W_2; Figure

WGN	Noncircular
ϵ_a	$\mathcal{N}(0,1)$
ϵ_b	$-0.6\epsilon_a + \mathcal{N}(0,1)$
ϵ_c	$0.8\epsilon_b + \mathcal{N}(0,1)$
ϵ_d	$0.8\epsilon_a - 0.4\epsilon_b + \mathcal{N}(0,1)$

Table E.1: Noncircular quadruply white quaternion Gaussian noise used.

E.4a) illustrates that the strictly linear FT-QLMS was unable to model the widely linear system W_2, whereas FT-WLQLMS and FT-CCQLMS converged to the optimal tap-length. This is justified by Remark 33, Remark 34 and Remark 35. Figure E.4b) illustrates that for the modelling of widely linear systems, value of $\lambda \to 0$ conforming with Remark 32.

E.5.3 Evolution of model parameters in nonstationary systems

To illustrate the real time tracking ability of the proposed algorithms, we next consider a system consisting of three separate subsystems of different natures. The first subsystem was the linear system W_1 for the intervals of $1 \leq k \leq 3000$, followed by widely linear system W_2 for $3001 \leq k \leq 6000$, and a linear noncircular W_1 for the interval of $6001 \leq k \leq 9000$. The linear noncircular W_1 was the original system W_1 (E.42) fed with a noncircular QWGN as the driving input. The noncircular QWGN was constructed by referring to Table E.1.

Figure E.5 shows the steady-state MSE for the so-produced noncircular linear system W_1 using the QLMS, WLQLMS and CCQLMS algorithms with $\mu = 10^{-4}$. As desired, all three of the MSE curves were monotonically non-increasing functions of the tap-length and the optimal tap length was found to be $p_o = \{21, 22\}$, illustrating the learning ability of the proposed architecture.

Figure E.6 shows the evolution of the optimal filter length parameter p for the FT-QLMS, FT-WLQLMS and FT-CCQLMS employed for the modelling of the system W_1 (interval of $1 \leq k \leq 3000$), W_2 (interval of $3001 \leq k \leq 6000$), and the noncircular W_1 (interval of $6001 \leq k \leq 9000$). These algorithms were intialized as follows: $\alpha = 0.03$, $\gamma = 1$, δ=1, Δ=4, $\mu = 1 \times 10^{-5}$, $\mu_l = 5 \times 10^{-4}$, the initial mixing parameter $\lambda(0) = 0.5$ and the initial tap length $p(0) = 25$. From the figure, the FT-WLQLMS was able to converge to the optimal tap-length of the system W_1 for the interval $1 \leq k \leq 3000$ and adapts to

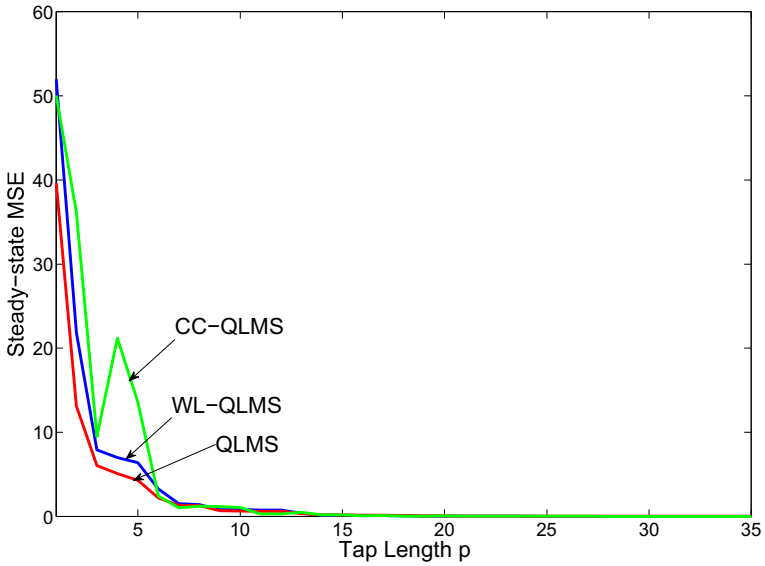


Figure E.5: The steady-state MSE for the process noncircular W_1 with respect to tap-length.

the system W_2 for $3001 \leq k \leq 6000$. The FT-WLQLMS was unable to model efficiently the noncircular W_1 for interval $6001 \leq k \leq 9000$. As for the FT-QLMS, it was incapable to adapt to the system W_2 during the interval of $3001 \leq k \leq 6000$ but was able to model W_1 and W_2. FT-CCQLMS was able to model all three systems owing to its theoretical advantages and the robust adaptation of the mixing parameter λ.

Figure E.7 depicts the evolution of the mixing parameter λ of FT-CCQLMS for the modelling of subsystems W_1, W_2 and noncircular W_1. For the modelling of linear system W_1, the parameter $\lambda \to 1$ for the interval of $1 \leq k \leq 3000$ making FT-QLMS dominant over FT-WLQLMS. As for the widely linear system W_2 in the interval $3001 \leq k \leq 6000$, the parameter $\lambda \to 0$ resulting in FT-WLQLMS to be superior. For the processing of noncircular linear system W_1, parameter $\lambda \to 1$ favouring the linear model. This corroborates with earlier findings in [151, 152].

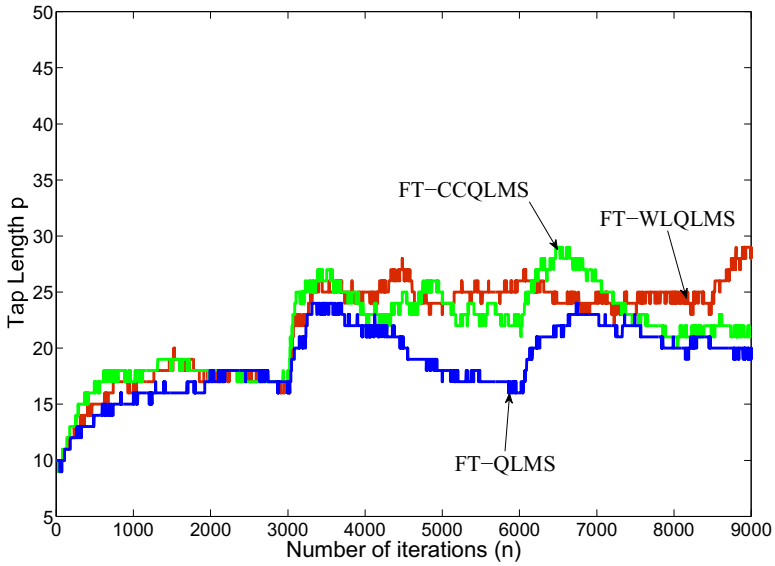


Figure E.6: The evolution of the mixing parameter p for the modelling of the linear system W_1 for the intervals $1 \leq k \leq 3000$, widely linear W_2 for $3001 \leq k \leq 6000$ and noncircular W_1 for $6001 \leq k \leq 9000$.

E.6 Summary

We have introduced the fractional tap-length (FT) optimisation into quaternion-valued adaptive modelling and have demonstrated the advantages it offers in model order selection and the identification of circular or noncircular nature of the data. The collaborative adaptive FT-CCQLMS has been shown to be able to model efficiently both widely linear and strictly linear quaternion-valued systems and to have a number of theoretical and practical advantages. The analysis on the convergence of the mixing parameter and the relationship between the steady-state error and tap-length has been established, giving a mathematical justification to the modelling capabilities of all algorithms. Simulations on model order selection and the identification of quaternion-valued circular natures for the problems in renewable energy and human-oriented computing support the approach.

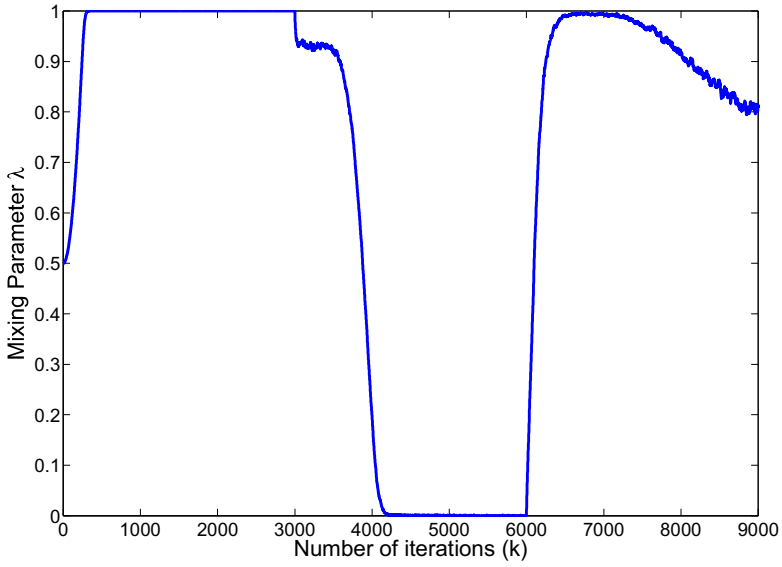


Figure E.7: The evolution of the optimal filter length parameter p for the modelling of the system W_1 for the intervals $1 \leq k \leq 3000$, noncircular W_1 $6001 \leq k \leq 9000$ and W_2 for $3001 \leq k \leq 6000$.

Appendix F

An Adaptive Diffusion Quaternion LMS Algorithm for Distributed Networks of 3D and 4D Vector Sensors

F.1 Introduction

In this Appendix, the IQLMS will be used as a platform for deriving the diffusion IQLMS (D-IQLMS) and diffusion WLIQLMS (D-WLIQLMS), allowing cooperative adaptive estimation of both circular and noncircular quaternion valued signals. The advantage of the D-WLIQLMS over the strictly linear D-IQLMS is illustrated over simulations on benchmark signals. In Section F.2 we derive the D-IQLMS, while in Section F.3 we use the widely linear model to obtain the D-WLIQLMS. Section F.4 presents stability bounds, thus ensuring convergence in the mean. We conclude the work with representative simulations.

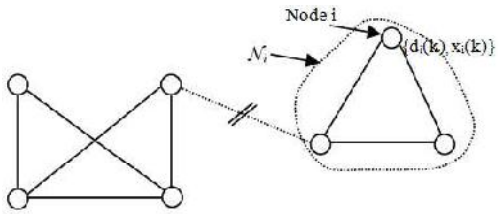


Figure F.1: Topology of a distributed network, highlighting the neighbourhood of node i.

F.2 Derivation of D-IQLMS

Consider a network of N nodes distributed over some area, as shown in Figure F.1. At time instant k, each node i has access to the local teaching signal $d_i(k)$ and a regressor input vector, defined as $\mathbf{x}_i(k) = [x_i(k-1), ..., x_i(k-L)]^T$. In addition to each nodes having access to their own data, nodes can also access data from their direct neighbours, where the neighbourhood $\mathcal{N}_i$ is defined as the set of all nodes linking to node i, including itself. A diffusion protocol describes how data from the neighbourhood $\mathcal{N}_i$ is combined.

For a global (where all data is available to all nodes) least squares solution to the network in Figure F.1, the centralized IQLMS (c-IQLMS) weight update can be written as [47]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \sum_{i=1}^{N} e_i(k)\mathbf{x}_i^*(k) \tag{F.1}$$

where the $e_i(k) = d_i(k) - y_i(k)$ is the error between the teaching signal $d_i(k)$ and output signal $y_i(k)$ at node i. In a distributed system, only data from the neighbourhood of a node is available and so the global weight update in (F.1) is replaced by local weight updates

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \mu \sum_{l \in \mathcal{N}_i} c_{l,i} e_l(k)\mathbf{x}_l^*(k) \tag{F.2}$$

where the weighting coefficient $c_{l,i}$ is always greater than 0, being equal to 0 when nodes l and k are not connected, and satisfies $\sum_{l \in \mathcal{N}_i} c_{l,i} = 1$. The weighting coefficient $c_{l,i}$ can be conveniently described as a matrix $\mathbf{C}$ where $\mathbb{1}^T\mathbf{C} = \mathbb{1}^T$ and $\mathbf{C}\mathbb{1} = \mathbb{1}$. The weight update

at node i can be interpreted as a weighted linear combination of neighbouring estimates of the local cost function gradient. The advantage of this local weight update over the global weight update is that it is robust to node and link failures. The local weight estimates $w_i(k)$ in (F.2) can be further improved by combining the local weight updates iteratively as follows

$$\begin{aligned} \boldsymbol{\phi}_i(k) &= \mathbf{w}_i(k-1) + \mu \sum_{l \in \mathcal{N}_i} c_{l,i} e_l(k) \mathbf{x}_l^*(k) \\ \mathbf{w}_i(k) &= \sum_{l \in \mathcal{N}_i} a_l(k) \boldsymbol{\phi}_l(k) \end{aligned} \tag{F.3}$$

where $\sum_{l \in \mathcal{N}_i} a_l = 1$ (more conveniently written as$\mathbf{A}\mathbb{1} = \mathbb{1}$).

F.3 The D-WLIQLMS

The D-IQLMS in (F.3) is only adequate when the underlying model is strictly linear. To process a signal generated from a widely linear system, or from noncircular statistics, we employ the widely linear model to augment the input vector. The D-WLIQLMS can then be compactly written as

$$\boldsymbol{\phi}_i^a(k) = \mathbf{w}_i^a(k-1) + \mu \sum_{l \in \mathcal{N}_i} c_{l,i} \mathbf{x}_l^{a*}(k) e_l(k) \tag{F.4}$$

$$\mathbf{w}_i^a(k) = \sum_{l \in \mathcal{N}_i} a_l(k) \boldsymbol{\phi}_l^a(k) \tag{F.5}$$

where $\mathbf{x}_l^a = [\mathbf{x}_l^{\imath T}, \mathbf{x}_l^{\imath T}, \mathbf{x}_l^{\jmath T}, \mathbf{x}_l^{\kappa T}]^T$.

F.4 Stability analysis of D-WLIQLMS

To investigate the convergence in the mean of the D-WLIQLMS algorithm, we rewrite the weight update in (F.5) as

$$\boldsymbol{\phi}^{aT}(k) = \mathbf{w}^{aT}(k-1) + \left(\mathbf{d}^T - \mathbf{w}^{aT}(k-1)\mathbf{X}^a(k)\right) \mathbf{X}^{aH}\mathbf{M}\mathbf{D} \tag{F.6}$$

where

$$\begin{aligned}
\boldsymbol{\phi}^a(k) &= [\boldsymbol{\phi}_1^{aT}(k), \ldots, \boldsymbol{\phi}_N^{aT}(k)] \\
\mathbf{w}^a(k) &= [\mathbf{w}_1^{aT}(k), \ldots, \mathbf{w}_N^{aT}(k)]^T \\
\mathbf{X}^a(k) &= diag\{[\mathbf{x}_1^{aT}(k), \ldots, \mathbf{x}_N^{aT}(k)]^T\} \\
\mathbf{d}(k) &= [d_1(k), \ldots, d_N(k)]^T \\
\mathbf{D} &= diag\{[\mu_1 \mathbf{I}_{4L}, \ldots, \mu_N \mathbf{I}_{4L}]\} \\
\mathbf{M} &= \mathbf{C} \otimes \mathbf{I}_{4L}
\end{aligned}$$

and $\otimes$ denotes the Kronecker product operator. To evaluate the performance of the D-WLIQLMS without loss in generality, consider an improper teaching signal $d(k)$, given by

$$d(k) = \mathbf{w}_o^{aT}(k)\mathbf{X}^a(k) + \mathbf{n}(k) \tag{F.7}$$

where $\mathbf{w}_o^{aT}(k)$ is the optimal (but unknown) local augmented weight vector, and $\mathbf{n}(k)$ is quadruply white Gaussian noise with zero mean and variance σ_n^2, uncorrelated with $\mathbf{X}(k)$. Substituting the teaching (F.7) into the weight update (F.6) and multiplying through we obtain

$$\begin{aligned}
\boldsymbol{\phi}^{aT}(k) = \mathbf{w}^{aT}(k-1) + \big(&\mathbf{w}_o^{aT}(k) \\
&- \mathbf{w}^{aT}(k-1)\big)\mathbf{X}^a(k)\mathbf{X}^{aH}\mathbf{MD} + \mathbf{n}(k)\mathbf{X}^{aH}\mathbf{MD}
\end{aligned}$$

while a further multiplication of both sides by -1 and addition of $\mathbf{w}_o^{aT}(k)$ yields

$$\begin{aligned}
\widetilde{\boldsymbol{\phi}}^{aT}(k) &= \widetilde{\mathbf{w}}^{aT}(k-1) - \widetilde{\mathbf{w}}^{aT}(k-1)\left(\mathbf{X}^a(k)\mathbf{X}^{aH}\mathbf{MD}\right. \\
&+ \left.\mathbf{n}(k)\mathbf{X}^{aH}\mathbf{MD}\right)
\end{aligned}$$

where the weight error vectors $\widetilde{\boldsymbol{\phi}} = \boldsymbol{\phi}_o(k) - \boldsymbol{\phi}(k)$ and $\widetilde{\mathbf{w}} = \mathbf{w}_o(k) - \mathbf{w}(k)$. Substitution for $\widetilde{\boldsymbol{\phi}}$ using (F.5) we have

$$\widetilde{\mathbf{w}}^{aT}(k-1) = \widetilde{\mathbf{w}}^{aT}(k-1)\mathbf{P} - \widetilde{\mathbf{w}}^{aT}(k-1)\left(\mathbf{X}^{a}(k)\mathbf{X}^{aH}\mathbf{MD} + \mathbf{n}(k)\mathbf{X}^{aH}\mathbf{MD}\right)\mathbf{P}$$

where $\mathbf{P} = \mathbf{A} \otimes \mathbf{I}_{4L}$. Upon applying the statistical expectation operator we have

$$E[\widetilde{\mathbf{w}}^{aT}(k-1)] = E[\widetilde{\mathbf{w}}^{aT}(k-1)]\left(\mathbf{I} - \mathbf{R}^{a}_{xx}\mathbf{MD}\right)\mathbf{P} \tag{F.8}$$

where

$$\mathbf{R}^{a}_{xx} = E[\mathbf{X}^{a}(k)\mathbf{X}^{aH}] = diag\{E[\mathbf{x}^{a}_{1}\mathbf{x}^{aH}_{1}], \ldots, E[\mathbf{x}^{a}_{N}\mathbf{x}^{aH}_{N}]\}$$

It then follows from (F.8) that to ensure that the power of the weight error vector $E[\widetilde{\mathbf{w}}^{aT}(k-1)]$ converges to 0 as $k \to \infty$, the following condition must hold

$$||\left(\mathbf{I} - \mathbf{R}^{a}_{xx}\mathbf{MD}\right)\mathbf{P}||_2 < 1$$

Since the 2-norm is sub-multiplicative and $||P|| = 1$ (this results from the fact that the rows and columns add to 1) we can write

$$||\left(\mathbf{I} - \mathbf{R}^{a}_{xx}\mathbf{MD}\right)\mathbf{P}||_2 < ||\left(\mathbf{I} - \mathbf{R}^{a}_{xx}\mathbf{MD}\right)||_2$$

and obtain a more conservative condition for convergence in the form

$$||\left(\mathbf{I} - \mathbf{R}^{a}_{xx}\mathbf{MD}\right)||_2 < 1$$

For the case where all nodes have the same step size, i.e. matrix $\mathbf{D}$ is a scaled identity matrix, the above convergence condition becomes $||\{(\mathbf{I} - \mu\mathbf{R}^{a}_{xx}\mathbf{M})\}||_2 < 1$, yielding the following bound for the step size

$$\mu < \frac{2}{\lambda_{max}(\mathbf{R}^{a}_{xx}\mathbf{M})} \tag{F.9}$$

F.5 Simulations

The one step ahead prediction performances of the proposed D-WLIQLMS and D-IQLMS were compared to the global IQLMS (where a centralized processing unit has access to all

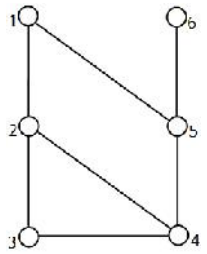


Figure F.2: Network topology.

nodes) and to the non-cooperative IQLMS, where nodes do not share information with neighbouring nodes (i.e. where $\mathbf{M} = \mathbf{I}$). The network topology consisted of 6 nodes and is shown in Figure F.2. Numerous methods exist to select the combination coefficients $c_{l,i}$, , such as the Metropolis [47], Laplacian and nearest neighbour rule. For our simulations the Metropolis rule was used, described by

$$c_{l,i} = \begin{cases} \frac{1}{max(n_i, n_l)} & \text{if } l \neq i \text{ are connected} \\ 0 & \text{if } l \neq i \text{ are not connected} \\ 1 - \sum_{l \in N_i} c_{l,i} & \text{if } l = i \end{cases} \tag{F.10}$$

where n_i and n_l are respectively the numbers of nodes in the neighbourhood of nodes i and l. Following the Metropolis rule in (F.10), the combination matrix $\mathbf{C}$ for the network topology in Figure F.2 becomes

$$\mathbf{C} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

The simulations were carried out for two signals, one generated by a strictly linear AR(4) model and the other generated by the Lorenz system.

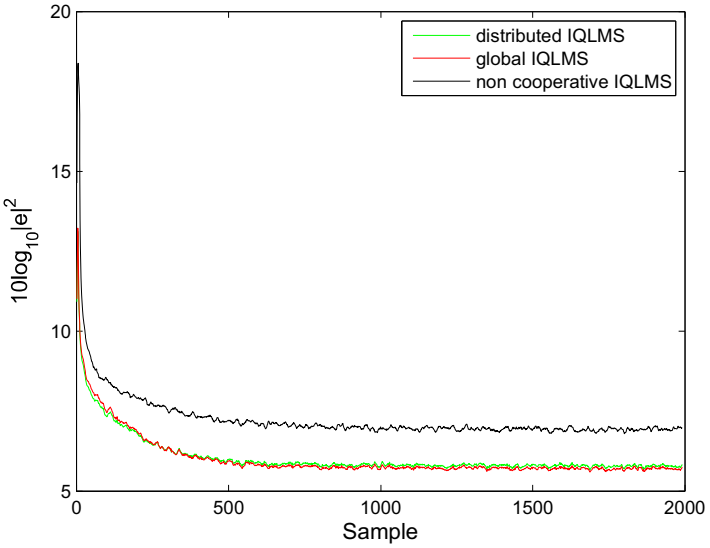


Figure F.3: Performance comparison of the QLMS, HR-QLMS and IQLMS on a circular AR(4) signal, for all the algorithms having the same step size.

F.5.1 A circular AR(4) process

The benchmark circular quaternion signal used in simulations was generated by an AR(4) model described by

$$\begin{aligned} y(k) &= 1.79y(k-1) - 1.85y(k-2) + 1.27y(k-3) \\ &\quad -0.41y(k-4) + n(k) \end{aligned}$$

where the driving noise $n(k)$ was quaternion valued circular white Gaussian noise. Figure F.3 shows the evolution of the mean square error (MSE) for the AR(4) process. Observe that as expected the distributed QLMS achieves significantly better steady state performance compared to the case where there is no cooperation between the nodes. It is also worth noting that the steady state performance of the distributed QLMS is only marginally worse than that of the global QLMS, while being more robust to failures in the network and requiring fewer mathematical operations.

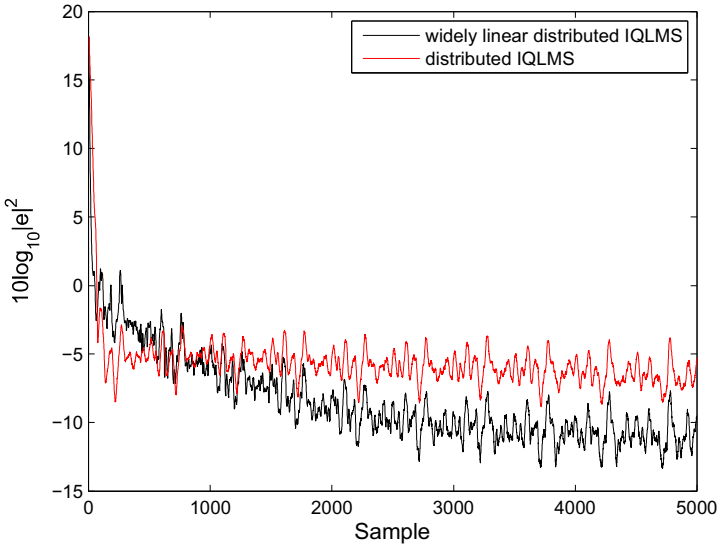


Figure F.4: Comparison of MSEs of the D-WLIQLMS and DIQLMS on the noncircular Lorenz signal.

F.5.2 Noncircular Lorenz attractor

The Lorenz signal is the output of a three-dimensional nonlinear system, originally used to model atmospheric turbulence, but also to model lasers, dynamos and the motion of waterwheel. Fig. F.4 shows the evolution of the MSE for the D-IQLMS and D-WLIQLMS. Observe that due to the widely linear nature of the Lorenz signal, the D-WLIQLMS has significantly lower MSE at steady state.

F.6 Summary

We have introduced a diffusion WLQLMS (D-WLQLMS) algorithm for the distributed processing of general quaternion valued signals (both circular and noncircular) in a co-operative fashion. The advantage of the diffusion topology over non-cooperative and fully-cooperative (global-QLMS) networks in terms of robustness, convergence speed and steady state performance has been shown in analysis and simulations on synthetic signals. The proposed strategy is particularly useful for data coming from vector sensors, such as 3D anemometers and 3D inertial body sensors.

Appendix G

Quaternion Derivatives: GHR Calculus

The quaternion derivatives in mathematical literatures of quaternion analysis are often defined only for analytic (regular) functions. However, in engineering problems, the functions of interest often are not analytic because they are often real-valued functions, such as the standard cost function. While the HR calculus is a convenient way to calculate the formal derivatives of analytic and non-analytic functions of quaternion variables, in our recently submitted paper "Quaternion Derivatives: GHR Calculus", we derive a generalization of the HR-calculus, based on a general orthogonal system. The generalization, called here generalized HR (GHR) calculus, encompasses not just the left and right-handed versions of quaternion derivative but also generates some surprising results, such as the novel product rule, the chain rule, the mean-valued theorem and Taylors theorem. The core of the novel product rule is the quaternion rotation, a generalisation of the quaternion involution. Taylors theorem is presented in a compact and familiar form, using the HR derivatives. To make the GHR-derivatives more accessible, we present the derivatives of many familiar quaternion functions in tabular form (shown below). Several examples taken from recently published material show how signal processing and communication systems can be optimized using the GHR-calculus

Table G.1: Important results of the GHR derivatives

$f(q)$	$\frac{\partial f}{\partial q^{\mu}}\mu$	$\frac{\partial f}{\partial q^{\mu^*}}\mu$	Note
q	$\Re(\mu)$	$-\frac{1}{2}\mu^*$	$--$
ωq	$\omega\Re(\mu)$	$-\frac{1}{2}\omega\mu^*$	$\forall\omega\in\mathbb{H}$
$q\nu$	$\Re(\nu\mu)$	$-\frac{1}{2}(\nu\mu)^*$	$\forall\nu\in\mathbb{H}$
$\omega q\nu+\lambda$	$\omega\Re(\nu\mu)$	$-\frac{1}{2}\omega(\nu\mu)^*$	$\forall\omega,\nu,\lambda\in\mathbb{H}$
q^*	$-\frac{1}{2}\mu^*$	$\Re(\mu)$	$--$
ωq^*	$-\frac{1}{2}\omega\mu^*$	$\omega\Re(\mu)$	$\forall\omega\in\mathbb{H}$
$q^*\nu$	$-\frac{1}{2}(\nu\mu)^*$	$\Re(\nu\mu)$	$\forall\nu\in\mathbb{H}$
$\omega q^*\nu+\lambda$	$-\frac{1}{2}\omega(\nu\mu)^*$	$\omega\Re(\nu\mu)$	$\forall\omega,\nu,\lambda\in\mathbb{H}$
$\alpha=\lvert V_q\rvert$	$-\frac{1}{4}\mu\hat{q}$	$\frac{1}{4}\mu\hat{q}$	$V_q=\Im(q)$
$\hat{q}=\frac{V_q}{\lvert V_q\rvert}$	$\frac{2\mu^*+\mu-\mu^{\hat{q}}}{4\alpha}$	$-\frac{2\mu^*+\mu-\mu^{\hat{q}}}{4\alpha}$	$\alpha=\lvert V_q\rvert$
$\arctan\left(\frac{\lvert V_q\rvert}{S_q}\right)$	$-\frac{\mu\hat{q}q^*}{4\lvert q\rvert^2}$	$\frac{\mu\hat{q}q}{4\lvert q\rvert^2}$	$S_q=\Re(q)$
q^{-1}	$-q^{-1}\Re(q^{-1}\mu)$	$\frac{1}{2}q^{-1}\mu^*(q^*)^{-1}$	$--$
$(q^*)^{-1}$	$\frac{1}{2}(q^*)^{-1}\mu^*q^{-1}$	$-(q^*)^{-1}\Re((q^*)^{-1}\mu)$	$--$
$(\omega q\nu+\lambda)^{-1}$	$-f\omega\Re(\nu f\mu)$	$\frac{1}{2}f\omega(\nu f\mu)^*$	$\forall\omega,\nu,\lambda\in\mathbb{H}$
$(\omega q^*\nu+\lambda)^{-1}$	$\frac{1}{2}f\omega(\nu f\mu)^*$	$-f\omega\Re(\nu f\mu)$	$\forall\omega,\nu,\lambda\in\mathbb{H}$
q^2	$q\Re(\mu)+\Re(q\mu)$	$-\frac{1}{2}q\mu^*-\frac{1}{2}(q\mu)^*$	$--$
$(q^*)^2$	$-\frac{1}{2}q^*\mu^*-\frac{1}{2}(q^*\mu)^*$	$q^*\Re(\mu)+\Re(q^*\mu)$	$--$
$(\omega q\nu+\lambda)^2$	$g\omega\Re(\nu\mu)+\omega\Re(\nu g\mu)$	$-\frac{1}{2}g\omega(\nu\mu)^*-\frac{1}{2}\omega(\nu g\mu)^*$	$g=\omega q\nu+\lambda$
$(\omega q^*\nu+\lambda)^2$	$-\frac{1}{2}g\omega(\nu\mu)^*-\frac{1}{2}\omega(\nu g\mu)^*$	$g\omega\Re(\nu\mu)+\omega\Re(\nu g\mu)$	$g=\omega q^*\nu+\lambda$
$\Re(q)$	$\frac{1}{4}\mu$	$\frac{1}{4}\mu$	$--$
$\Re(\omega q\nu+\lambda)$	$\frac{1}{4}\mu\nu\omega$	$\frac{1}{4}\mu\omega^*\nu^*$	$\forall\omega,\nu,\lambda\in\mathbb{H}$
$\Re(\omega q^*\nu+\lambda)$	$\frac{1}{4}\mu\omega^*\nu^*$	$\frac{1}{4}\mu\nu\omega$	$\forall\omega,\nu,\lambda\in\mathbb{H}$
$\frac{q}{\lvert q\rvert}$	$\frac{1}{\lvert q\rvert}\Re(\mu)-\frac{1}{4\lvert q\rvert^3}q\mu q^*$	$-\frac{1}{2\lvert q\rvert}\mu^*-\frac{1}{4\lvert q\rvert^3}q\mu q$	$--$
$\frac{q^*}{\lvert q\rvert}$	$-\frac{1}{2\lvert q\rvert}\mu^*-\frac{1}{4\lvert q\rvert^3}q^*\mu q^*$	$\frac{1}{\lvert q\rvert}\Re(\mu)-\frac{1}{4\lvert q\rvert^3}q^*\mu q^*$	$--$
$\frac{\omega q\nu+\lambda}{\lvert\omega q\nu+\lambda\rvert}$	$\frac{\omega}{2\lvert g\rvert}\Re(\nu\mu)+\frac{g}{4\lvert g\rvert^3}\nu^*(\omega^*g\mu)^*$	$-\frac{\omega}{4\lvert g\rvert}(\nu\mu)^*-\frac{g}{2\lvert g\rvert^3}\nu^*\Re(\omega^*g\mu)$	$g=\omega q\nu+\lambda$
$\frac{\omega q^*\nu+\lambda}{\lvert\omega q^*\nu+\lambda\rvert}$	$-\frac{\omega}{2\lvert g\rvert}(\nu\mu)^*-\frac{f}{\lvert g\rvert}\frac{\partial\lvert g\rvert}{\partial q^{\mu}}\mu$	$\frac{\omega}{\lvert g\rvert}\Re(\nu\mu)-\frac{f}{\lvert g\rvert}\frac{\partial\lvert g\rvert}{\partial q^{\mu^*}}\mu$	$g=\omega q^*\nu+\lambda$
$\lvert q\rvert$	$\frac{1}{4\lvert q\rvert}\mu q^*$	$\frac{1}{4\lvert q\rvert}\mu q$	$--$
$\lvert q\rvert^2$	$\frac{1}{2}\mu q^*$	$\frac{1}{2}\mu q$	$--$
$\lvert\omega q\nu+\lambda\rvert$	$\frac{g^*}{2\lvert g\rvert}\omega\Re(\nu\mu)-\frac{1}{4\lvert g\rvert}\nu^*(\omega^*g\mu)^*$	$-\frac{g^*}{4\lvert g\rvert}\omega(\nu\mu)^*+\frac{1}{2\lvert g\rvert}\nu^*\Re(\omega^*g\mu)$	$g=\omega q\nu+\lambda$
$\lvert\omega q^*\nu+\lambda\rvert$	$\frac{g}{2\lvert g\rvert}\nu^*\Re(\omega^*\mu)-\frac{1}{4\lvert g\rvert}\omega(\nu g^*\mu)^*$	$-\frac{g}{4\lvert g\rvert}\nu^*(\omega^*\mu)^*+\frac{1}{2\lvert g\rvert}\omega\Re(\nu g^*\mu)$	$g=\omega q^*\nu+\lambda$
$\lvert\omega q\nu+\lambda\rvert^2$	$g^*\omega\Re(\nu\mu)-\frac{1}{2}\nu^*(\omega^*g\mu)^*$	$-\frac{1}{2}g^*\omega(\nu\mu)^*+\nu^*\Re(\omega^*g\mu)$	$g=\omega q\nu+\lambda$
$\lvert\omega q^*\nu+\lambda\rvert^2$	$g\nu^*\Re(\omega^*\mu)-\frac{1}{2}\omega(\nu g^*\mu)^*$	$-\frac{1}{2}g\nu^*(\omega^*\mu)^*+\omega\Re(\nu g^*\mu)$	$g=\omega q^*\nu+\lambda$

Appendix H

Split Complex Estimation: A Novel Method for Designing Efficient Widely Linear Adaptive Filters

In complex valued least squares estimation, the cost function is usually made to be the absolute value of the error function, that is, the sum of the real and imaginary errors squared. Conventional linear estimators try to minimise the sum of these two mean squared errors with a single set of parameters and this in general, leads to a sub-optimal solution. Our method, called split complex estimation and introduced in our recently submitted publication "Split Complex Estimation: A Novel Method for Designing Efficient Widely Linear Adaptive Filters", separates the mean squared error into two separate cost functions, and finds optimal linear estimators that minimise each one separately. We show that this approach to complex valued estimation is equivalent to widely linear estimation, while offering a number of important advantages. For example, developing algorithms using this approach allows us to create computationally more efficient adaptive filters and obtain a greater insight into the transient properties of the widely linear adaptive filters. This is shown by introducing three novel least mean square (LMS) algorithm, called, the

dual channel complex LMS (DC-CLMS), complex LMSr and complex LMSi. Conditions for equivalence between these filters and the CLMS and ACLMS (augmented CLMS) are shown and convergence in the mean and mean square are carried out.