

UNIVERSITY OF SOUTHERN QUEESLAND

**ANALYSING EEG BRAIN SIGNALS USING
INDEPENDENT COMPONENT ANALYSIS TECHNIQUES**

A dissertation submitted by

Janett G. St. H. Williams, MS, MA

For the award of

Doctor of Philosophy

2011

Dedication

To my mother Gretel Georgia Walters

a love for education driven her believe in me. Mom you died two years too soon

To my family

who loves me unconditionally and believes in me no matter what.

Abstract

The use of electroencephalography (EEG) in the medical field is evident in the effect it has on diagnosis and treatment of patients who suffer from some form of brain problem. These signals however once collected are overlaid with artifacts. This thesis considers this problem and seeks to solve using popular methods in the form of Independent Component Analysis (ICA) and Wavelet Transform (WT).

Independent component analysis (ICA) is a popular blind source separation (BSS) technique that has proven to be promising for the analysis of EEG data. There are different estimators to developing these ICAs. Mutual Information is one of the most natural criteria when developing an estimator. Although utilized to some level it has always been difficult to calculate. In this thesis I present a new algorithm which utilizes a contrast function related to Mutual Information based on B-Spline functions. This thesis also investigates the creation of an algorithm which is based on a merger of Independent Component Analysis and Translation Invariant Wavelet Transform and goes on to merger the B-Spline ICA with the Translation Invariant Wavelet Transform. In addition I apply Unscented Kalman Filtering as it does not require any prior signal knowledge. Each algorithm will be examined and compared to ones in literature tackling the same EEG problems; results will be drawn on the base of comparative tests on both synthetic and real.

Certification of Dissertation

I certify that the ideas, experimental work, results, analyses, software and conclusions reported in this dissertation are entirely my own effort, except where otherwise acknowledged. I also certify that the work is original and has not been previously submitted for any other award, except where acknowledged.



November 24, 2011

Signature of Candidate

Date

ENDORSEMENT

Signature of Supervisor/s

Date

Acknowledgement

Completing a Doctor of Philosophy in a very challenging subject is usually a long and winding road; luckily I was not alone in this trip. The following were my travelling companions in this journey and I would like to say a big thanks, as this work would not have been possible without them.

Firstly I must thank God Almighty who completes things in the right season and who declared that this is my season. I am sincerely thankful to my supervisor Dr. Yan Li, for having believing in me and giving the necessary guidance and advice. In so doing she is responsible for helping me develop into the scientist that I am becoming.

Thanks to the University of Technology, Jamaica for funding half of my tuition for the time of my studies. Special thanks must be extended to the School of Computing and Information Technology, especially Mr. Arnett B. Campbell, Head of School, who provided me with the time and resources to conduct my research.

To my husband Marlon, my biggest supporter, reader and commenter, thank you for the many nights you sacrificed your sleep.

To my pastor and friend, Rev. Dexter E. Johnson whose support was deeply appreciated and who kept the church “at bay” while this thesis was being completed. I acknowledge my students who in supporting and believing in me made working pleasurable and at time a laughable experience.

Finally, thanks to my children – Melonie, MacAllister and Mahalia for their patience and love in the long distance and my mother-in-law Jean Williams for her help in taking care of them during my time of studies.

Table of Contents

Abstract	ii
Acknowledgement	iv
List of Figures	ix
List of Tables	xii
List of Algorithms	xiii
List of Abbreviations	xiv
List of Publications	xvii

Chapter 1 – Introduction

1.1	Background	1
1.2	Justification for the Research	4
1.3	Methodology	5
1.4	Outline of Dissertation	11
1.5	Proposed Contributions of the Dissertation	12
1.6	Summary	13

Chapter 2 - Electroencephalograph

2.1	Introduction	14
2.2	Electroencephalograph Measuring System	17
2.3	Wave Analysis of the EEG	19
2.4	Uses of EEG	26
2.5	Artifacts in EEG	27
2.6	Denoising EEG	35
2.7	Methods	36
2.8	Summary	38

Chapter 3 – Denoising Methods

3.1	Independent Component Analysis	40
3.2	Wavelet Analysis	57
3.3	Filtering	66
3.4	Performance Measure for Methods	70
3.5	Summary	74

Chapter 4 – B-Spline Mutual Information ICA

4.1	The Mutual Information Estimator	75
4.2	B-Spline Function	80
4.3	Newly Designed ICA	84
4.4	Summary	98

Chapter 5 – Reliability of BMICA

5.1	Reasons for Reliability Testing for ICA Algorithms	99
5.2	Previous Research on Reliability	100
5.3	The ICASSO Reliability Test	101
5.4	Comparison ICA Algorithm Tested – FastICA	104
5.5	Results	105
5.6	Summary	120

Chapter 6 – MI Algorithms vs Non-MI Algorithms

6.1	Introduction	121
6.2	Experiment Setup	121
6.3	Results	122
6.4	Discussion	130
6.5	Summary	131

Chapter 7 – Unscented Kalman Filter

7.1	Introduction	132
7.2	EEG, EKF and UKF	132
7.3	Experiment	135
7.4	Results	136
7.5	Discussions	140
7.6	Summary	142

Chapter 8 – Improving Translation Invariant Wavelet Transform (TIWT)

8.1	Introduction	143
8.2	Translation Invariant Algorithm	143
8.3	Mergering Filters and WT	153
8.4	TIWT and BMICA Merger	166
8.5	Summary	179

Chapter 9 – Discussion and Conclusion

9.1	Summary	182
9.2	Links to Dissertation Goals	182
9.3	Conclusion	187
9.4	Actual Contribution of the Dissertation	188
9.5	Implications	189
9.6	Further Work	190

References	192
------------	-----

Appendix A	211
------------	-----

Appendix B	213
------------	-----

Appendix C	213
Appendix D	214
Appendix E	214
Appendix F	215
Appendix G	216
Appendix H	218
Appendix I	220
Appendix J	221
Appendix K	222

List of Figures

1.1	Work Flow of Dissertation	6
1.2	Distribution of Electrodes	8
1.3	Sample EEG Signals from a Male Subject	9
2.1	Structure of Typical Mammalian Neurons	14
2.2	Placement of EEG Electrodes on a Patient	15
2.3	Different Amplifiers Which Produce A Signal EEG Recording/Signal	16
2.4	A Simple Example of an Electroencephalography Machine	16
2.5	The International 10-20 System	18
2.6	Four of the Six Frequency Bands found in EEG Signals	19
2.7	One Second Recording of EEG Alpha Wave	20
2.8	A Schematic View of the Human Head Viewed From Above	20
2.9	One Second Recording of EEG Mu Wave	21
2.10	One Second Recording of EEG Beta Wave	22
2.11	One Second Recording of EEG Theta Wave	22
2.12	One Second Recording of EEG Delta Wave	23
2.13	One Second Recording of EEG Gamma Wave	23
2.14	Set of EEG signals Showing 3 of the Six Bands	24
2.15	EEG Activity is Dependent on the Level of Consciousness	25
2.16	One second Recording of clean pure EEG Signal	27
2.17	Eye Blink Artifact	28
2.18	Eye Movement Artifact	29
2.19	EOG Signal containing both Eye Blink and Eye movement artifacts	29
2.20	EEG Contaminated with EOG Producing Spikes	30
2.21	Ten Seconds Cardiac Movement Artifact	30
2.22	EEG Signals corrupted with EKG and line signals	31

2.23	Muscle Activity Artifact - Chewing	31
2.24	Recording of a Glossokinetic Signal	32
2.25	One 60 Seconds Recording of a GSR Signal	33
2.26	Electrode Pop Artifact	34
2.27	Line Interference of 50Hz	34
3.1	Mathematical Model for ICA decomposition	43
3.2	Generalized ICA Algorithm	44
3.3	EEG Signals Being Broken Into ICs Using ICA	48
3.4	Difference between (A) Wave and (B) Wavelet	57
3.5	Several Different Families of Wavelets	59
3.6	Hard and Soft Thresholding Estimators Along With the Original Signal	63
3.7	Block Diagram of the Translation Invariant Wavelet Transform	63
3.8	Noisy EEG and its Wavelet Transform at Different Scales	66
4.1	Relationship between Mutual Information $I(X:Y)$ and Entropies $H(X)$ and $H(Y)$	76
4.2	Sample of Raw EEG Signal	87
4.3	EEG Signal After Denoised with BMICA algorithm	88
4.4	SIR Comparison (A) Fixed Point Algorithm (B) Non-fixed Point Algorithm	95
4.5	Amari Index (A) Fixed Point Algorithm (B) Non-fixed Point Algorithm	96
5.1	Sample of a Dendrogram	102
5.2	Cluster Plots for (A) BMICA (B) FastICA	106
5.3	Dendrogram for (A) BMICA (B) FastICA	108
5.4	Estimate Quality for (A) BMICA (B) FastICA	109
5.5	Estimates per cluster for (A) BMICA (B) FastICA	112
5.6	Centrotypes for (A) BMICA (B) FastICA	114
5.7	Cluster Plots for (A) FastICA(del-guas) (B) FastICA (sym-guas)	115
5.8	Cluster Plots for (A) FastICA(del-skew) (B) FastICA (sym-skew)	116
5.9	Cluster Plots for (A) FastICA(del-tanh) (B) FastICA (sym-tanh)	117

5.10	Cluster Plots for (A) FastICA(del-pow3) (B) FastICA (sym-pow3)	118
5.11	Cluster Plot for BMICA	119
6.1	Amari Index for both Mutual Information and Non Mutual Information Algorithms	124
7.1	First 14 Signals from Data Set Used	136
7.2	Channel 32 Showing First 150 Values With and Without Noise	137
7.3	True State of Signal and Estimates for (A) UKF (B) EKF	138
7.4	Estimation Errors and 3- σ Confidence Intervals for (A) UKF (B) EKF	139
7.5	Performance Comparison of the UKF and EKF filters.	141
8.1	EEG Signal Contaminated with EOG	145
8.2	Denoised EEG Signal for (A) TIWT (B) FastICA (C) RADICAL	146
8.3	SNR Comparison of EEG Signals	147
8.4	MSE Comparison of EEG Signals	148
8.5	PRD Comparison of EEG Signals	149
8.6	SIR Comparison with Six Other Algorithms	150
8.7	Amari Index Comparison with Five Other Algorithms	151
8.8	Proposed CTICA Artifacts Removal System	155
8.9	(A) EEG Signals with EOG (B) Denoised EEG Signals	157
8.10	Wave Coefficient (A) Before Denoising (B) After Denoising	158
8.11	SDR for 32 Real EEG Signals with EOG	162
8.12	Amari Results for the Four Algorithms	163
8.13	Raw EEG Signals	167
8.14	Denoised EEG using (A) WT (B) BMICW-WT	168
8.15	SIR Relations between BMICW-WT and TIWT	169
8.16	PSNR Relations between BMICW-WT and TIWT	171
8.17	Amari Index for BMICA-WT with Non-fixed Point Algorithms	173
8.18	Amari Index for BMICA-WT with Fixed Point Algorithms	174

8.19	SIR for BMICA-WT with Fixed Point Algorithms	175
8.20	SIR for BMICA-WT with Non-fixed Point Algorithms	175

List of Tables

4.1	MSE Comparison with (A) Fixed Point Algorithms (B) Non-Fixed Point Algorithms	89
4.2	PSNR Comparison with (A) Fixed Point Algorithms (B-C) Non-Fixed Point Algorithms	90
4.3	SNR Comparison with (A) Non-Fixed Point Algorithms (B) Fixed Point Algorithms	92
4.4	SDR Comparison with (A) Fixed Point Algorithms (B) Non Fixed Point Algorithms	93
6.1	SDR for 20 EEG Signals	122
6.2	SIR for 20 EEG Signals	123
6.3	MSE for 20 EEG Signals	125
6.4	SNR for 20 EEG Signals	126
6.5	PSNR for 20 EEG Signals	127
7.1	MSE for 8 Channels	140
8.1	PSNR for 11 Real EEG Signals	152
8.2	MSE for 11 Real EEG Signals	152
8.3	MSE for 20 Real EEG with EOG Noise	159
8.4	MSE for 19 EEG with Artificial Added Noise	159
8.5	PSNR for 19 Real EEG with EOG noise	160
8.6	PSNR for 19 EEG with Artificial Added Noise	161
8.7	SDR for 19 EEG with Artificial Added Noise	161
8.8	SDR for 19 EEG Signal Sets	169
8.9	MSE for 18 EEG Signal Sets	172
8.10	MSE for (A) Fixed Point Algorithms (B) Non-fixed Point Algorithms	176
8.11	PSNR for (A) Fixed Point Algorithms (B) Non-fixed Point Algorithms	177
8.12	SNR for (A) Fixed Point Algorithms (B) Non-fixed Point Algorithms	178

List of Algorithms

Algorithm 1	Algorithm for Uniform Cubic B-Spline Function	82
Algorithm 2	Algorithm to Generate B-Spline Estimated MI	84
Algorithm 3	Algorithm to Generate New ICA	86
Algorithm 4	Extended Kalman Filter	133
Algorithm 5	Unscented Kalman Filter	134

List of Abbreviations

α	Level of Confidence
σ	Standard Deviation
AL	Average Link
ARMA	Auto-Regressive Moving-Average
BMICA	B-Spline Mutual Information Independent Component Analysis
BMICA-WT	B-Spline Mutual Information Independent Component Analysis – Wavelet Transform
B-Spline	Basis Spline
BSS	Blind Source Separation
CCA	Cuvilinear Component Analysis
CS	Cycle Spinning
CT	Computed Tomography
CTICA	Cycle Spinning Wavelet Transform Independent Component Analysis
CubICA	Cumulant-based Independent Component Analysis
CRB	Cramér-Rao lower bound
CWT	Continuous Wavelet Transform
df	Degree of Freedom
DSS	Dynamic State Space
DWT	Discrete Wavelet Transform
ECG/EKG	Electrooculogram, -graphy
EEG	Electroencephalogram, -graphy
EFICA	Efficient FastICA
EKF	Extended Kalman Filter
EMG	Electromyogram, -graphy
EOG	Electroculogram, -graphy
FastICA	Fast fixed point Independent Component Analysis

GSR	Galvanic Skin Response
ICA	Independent Component Analysis
IMA	Infusion Motor Artifact
Infomax	Information Maximization
JADE	Joint Approximate Diagonalization Eigen Matrices
KDE	Kernel Density Estimator
KF	Kalman Filter
KL	Kullback Leibler
KNN	K Nearest Neighbour
Matlab	Matrix Laboratory
MEG	Magneto Encephalography
MI	Mutual Information
MILCA	Mutual Information Least-Dependent Component Analysis
ML	Maximum Likelihood
MMI	Minimum Mutual Information
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
p-value	Probability Value
PCA	Principal Component Analysis
PRD	Percentage Root Mean Square Difference
PSNR	Peak Signal to Noise Ratio
RADICAL	Robust, Accurate, Direct Independent Component Analysis Algorithm
REM	Rapid eye movement
SCCN	Swartz Center for Computational Neuroscience
SDR	Signal to Distortion Ratio
SIR	Signal to Interference Ratio
SNR	Signal to Noise Ratio
SOBI	Second Order Blind Identification

SOS	Second Order Statistics
SWS	Slow wave Sleep
SWT	Stationary Wavelet Transform
TIWT	Translation Invariant Wavelet Transform
TDSEP	Temporal Decorrelation Source Separation
TVAR	Time Varying Parameter Auto Regressive
UKF	Unscented Kalman Filter
UNGM	Univariate Nonstationary Growth Model
UT	Unscented Transformation
WF	Weiner Filter
WT	Wavelet Transform

List of Publications

Journal Papers

Janett Walters-Williams and Yan Li, *B-Spline Mutual Information Independent Component Analysis*. In Journal of Computer Science and Network Security, Vol 10, No. 7, 2010 pp. 129-141.

Janett Walters-Williams and Yan Li, *A New Approach to Denoising EEG Signals – Merger of Translation Invariant Wavelet and ICA*. In International Journal of Biometric and Bioinformatics (IJBB), Volume 5, Issue 2, May 2011, pp 130 – 148

Janett Walters-Williams and Yan Li, *Improving the Performance of Translation Wavelet Transform using BMICA*. In International Journal of Computer Science and Information Security (IJCSIS), Volume 9 No. 6 June 2011, pp 48-56

Janett Walters-Williams and Yan Li, *Performance Comparison of Known ICA Algorithms to a Wavelet-ICA Merger*. In Signal Processing: An International Journal (SPIJ), Volume 5, Issue 3. July 2011, pp 80-92

Janett Walters-Williams and Yan Li, *Using Invariant Translation to Denoise EEG Signals*, In American Journal of Applied Sciences, Volume 8, Issue 11, November 2011, pp 1122-1130

Janett Walters-Williams and Yan Li, *Investigating Nearest Neighbours Calculation: The Comparative Study of Distance Functions*. In Press, Journal of Communication and Computer, July 2011

Janett Walters-Williams and Yan Li, *Evaluation of an ICA-Filter-Wavelet Merger - A Case Study on Denoising EEG*, In Press, International Journal of Computer Science Issues ("IJCSI"), Vol 8 Iss 4, July 2011

Janett Walters-Williams and Yan Li, *BMICA-Independent Component Analysis Based On B-Spline Mutual Information Estimator*, In Press, Signal & Image Processing: An International Journal (SIPIJ), September 2011

Janett Walters-Williams and Yan Li, *BMICA-Independent Component Analysis based on B-Spline Mutual Information Estimation for EEG Signals*, Submitted to Medical Engineering & Physics Journal, May 2011

Janett Walters-Williams and Yan Li, *Reliability Testing of B-Spline Mutual Information Independent Component Analysis on EEG Signals*, Submitted to the International Journal of Systems Science, June 2011

Janett Walters-Williams and Yan Li, *Denoising EEG: Making the Correct Choice of ICA Algorithm*, Submitted to Clinical Neurophysiology, June 2011

Janett Walters-Williams and Yan Li, *BIMCA: Cleaner EEG Signals Equaling Quicker and Better Diagnosis of Epilepsy*, Submitted to Epilepsia, June 2011.

Conference Papers

Janett Walters-Williams and Yan Li, *Comparative Study of Distance Functions for Nearest Neighbours*. In Proceedings of International Conference on Systems, Computing Sciences and Software Engineering (SCS²), December 5-13, 2008, Published in Advanced Techniques in Computing Sciences and Software Engineering, Vol. XIV, 2010, pp 79-84

Janett Walters-Williams and Yan Li, *Estimation of Mutual Information: A Survey*. In Proceedings of the Fourth International Conference on Rough Set and Knowledge Technology (RSKT2009) July 14-16, 2009, Gold Coast, Australia, pp. 389-396

Janett Walters-Williams and Yan Li, *Comparison of Extended and Unscented Kalman Filters applied to EEG Signals*, In Proceedings of the 2010 IEEE/ICME International Conference on Complex Medical Engineering (CME2010) in Gold Coast, Australia, 13-15 July 2010, pp. 129-134

CHAPTER 1 - Introduction

1.1 Background

Biological processes are very complex mechanism, encompassing both neural and hormonal stimuli and responses, inputs and outputs in the most different forms, including physical material or information, and actions that could as well be mechanical, electrical or biochemical. Most of these processes are accompanied by or manifest themselves as signals that reflect their essential characteristics and qualities. Such signals are different in nature such as electrical and biochemical.

The development of diagnostic techniques based on these signal acquisition from the human body is commonly retained as one of the propelling factors of the advancements in medicine and biosciences. In fact, diseases or defects in biological systems almost always cause alterations in normal functions, giving birth to pathological processes that negatively impact on the performance and behavior of the systems themselves. If a good understanding of the system of interest is retained, in my case the neural network, it is possibly after the investigation of the signals and features originated by the system, to assess its state, discriminating between normal and abnormal responses. However, most physicians, like radiologists and neuroscientist, have to deal with additional problems when diagnosing the health state of the biological system from its signals. Like any acquisition systems, the instruments used for these biological systems are affected by non-idealities (artifacts) which by different degrees, negatively impact on the accuracy of these recordings. Accurate readings result once these artifacts have been removed.

In this research I study the neural network with emphasis on the brain as the biological system. The main interest in functional brain studies lays in the electrical activity of firing neurons, which cannot be directly investigated by any magnetic resonance imaging (MRI) procedure. The analysis of brain electrical activity is an increasingly important area of research for both understanding and modeling the

human brain, and for medical diagnosis and treatment as well, especially for developing automated patient monitoring and computer-aided diagnosis. Extraction of relevant information on brain activity from measured electrical signals, called Electroencephalogram or EEG, (measures electrical potentials on the scalp surface that occur as a result of dynamic brain function [80]) is affected by various artifacts due to volume conduction through cerebrospinal fluid, skull, and scalp, as well as generated by experimental imperfections. These artifacts include: EOG (Eye-induced) artifacts (includes eye blinks and eye movements); ECG/EKG (cardiac) artifacts; EMG (muscle activation-induced) artifacts; and Glossokinetic artifacts. Developing and understanding advanced signal processing techniques for the analysis of EEG signals is crucial in the area of biomedical research. The presence of these artifacts may cause different interpretations by users of the EEG signals [146] which may result in misdiagnosis in the case of some patients. Artifacts must therefore be eliminated or attenuated.

Extraction of these artifacts is based on different data analysis techniques. These are loosely dichotomized into (i) *hypothesis-driven* methods, like the general linear model (GLM) [37], and (ii) *data-driven* model-free methods, such as principal component analysis (PCA) and Independent Component Analysis (ICA) [118]. The estimations of the problem of determining the brain electrical sources from potential patterns recorded on the scalp surface are mathematically undetermined [103] however, ICA algorithms have been proven to be a reasonably fit technique in removing these artifacts.

1.1.1 Research Problems, Hypothesis and Contributions

Although there have been many researchers and algorithms, after 60 years artifacts contaminations remain a problem [71]. Krishnaveni *et al.* [90] in their research found that of the six algorithms tested the Robust, Accurate, Direct Independent Component Analysis Algorithm (RADICAL) was considered to be the most robust ICA algorithms in the presence of artifacts in mixed data. Pandey *et al.* [126] in their research of four algorithms including RADICAL found that although

RADICAL was the most robust, its performance was poor as all the algorithms assume the data to be homogeneous, which EEG is not, and operate based on the noise-free ICA model. Most if not all present algorithms therefore still do not remove all artifacts, resulting in degraded performances and interpretations that are still yet to be fully accurate.

Apart from the artifacts problem it has been found that most of the present algorithms perform best with certain data sizes e.g. Joint Approximate Diagonalization Eigen Matrices (JADE) which performs best on small signal set sizes [175]. There is presently none that can perform accurately given any data size therefore there is a problem of being adaptive.

Hyvarinen *et al.* [62] stated that the use of whitening in ICA helps to explain why the uses of Gaussian variables are forbidden. Most of the present ICA algorithms perform under the assumption that their data is non-gaussian; using ICA estimation can only be done up to an orthogonal transformation [61]. Separation can therefore fail when a Gaussian distribution is found within the data [120].

1.1.2 Dissertation Goals

This dissertation aims to present the design and implementation of robust ICA algorithms to separate Electroencephalography (EEG) signals from other signals described as artifacts. The new algorithms are expected to provide a more efficient way of removing these signals leaving EEG signals which can be interpreted by users. These algorithms will have the following features:

1. Robustness: The algorithms will be able to remove outliers from the signal data.
2. Accuracy: Removal of EOG and other underlying signals from the signal data.
3. Adaptive: Able to perform efficiently given any amount of data.

4. Convergence: Able to have a fast convergence.

1.2 Justification for the Research

Based on literature investigations most EEG correction techniques focus on removing artifacts based on contamination by eye movements and blinks often called ocular (EOG) artifacts [24, 42, 71, 77]. There has been relatively little work done on other forms of artifacts such as cardiac signals, muscle activities and electrode noise. In 2007 the most widely spread method was to reject the EEG segments containing these artifacts thus removal, especially when there are limited data available and/or many artifacts present, may lead to an unacceptable loss of valuable data [1147].

The ICA techniques in existence are mainly based on the basic noise-free ICA definition in [61] where the artifacts term is usually omitted i.e.

$$x(t) = A \times s(t) \quad (1.1)$$

where A is a 2x2 mixing matrix, $s(t)$ the desired signal and $x(t)$ is the observed signal in time (t). This is because this model seems to be sufficient for many applications and in many cases the number of the independent components (ICs) and observed mixtures may not be equal [62]. The algorithms based on this model produce results which are simpler and tractable [59]. These algorithms are perfect for artificial signal sets however real signals such as EEG always has some kind of artifacts present [62] resulting in Hyvärinen stating that canceling noise is a central yet an unsolved problem in EEG processing [59]. Literature therefore shows that the present algorithms either remove EEG data or allow for artifacts to remain after completion.

The level of performance for any ICA algorithms can be measured based on four areas:

1. How independent the “independent” components are

2. The uniqueness of the components.
3. The robustness of the estimated dependencies against outliers and artifacts.
4. The robustness of the estimated components.

Literature on ICA algorithms shows that areas 2 and 4 have often be utilized and answered, however areas 1 and 3 have not been investigated much. ICA algorithms have the need to exploit an independence measure. Literature on Mutual Information (MI) has shown that it is an obvious candidate for measuring this independence [89, 159] and a good contrast function [20, 62] thus answering area 1. MI, however, is not extensively used for measuring interdependence because estimating it from statistical samples is not easy.

In the ICA literature very crude approximations to MI based on cumulant expansions are popular because of their ease of use [89] and have been very successful [20]. Hyvarinen [59] stated however that in their present use MI algorithms is far from optimal as far as robustness and asymptotic variance are concerned. These algorithms were also sensitive to artifacts. As they are now MI-based algorithms cannot answer neither areas 1 or 3 although designed to measure the independence of the components and considered a natural criterion to estimating ICA algorithms. To the best of my knowledge no researcher has implemented an algorithm which seeks to tackle all 4 areas. In 2004 the closest algorithm, Mutual Information Least-Dependent Component Analysis (MILCA), was created but it is slower than algorithms like FastICA and JADE [188].

1.3 Methodology

I will use the B-Spline function to estimate MI which will then be used to create a new ICA algorithm. This new algorithm will be compared with known ICA algorithms as well as merged with Wavelet Transform to further create another form of denoising algorithm. Methodology flow charts can be seen in Figureure 1.1.

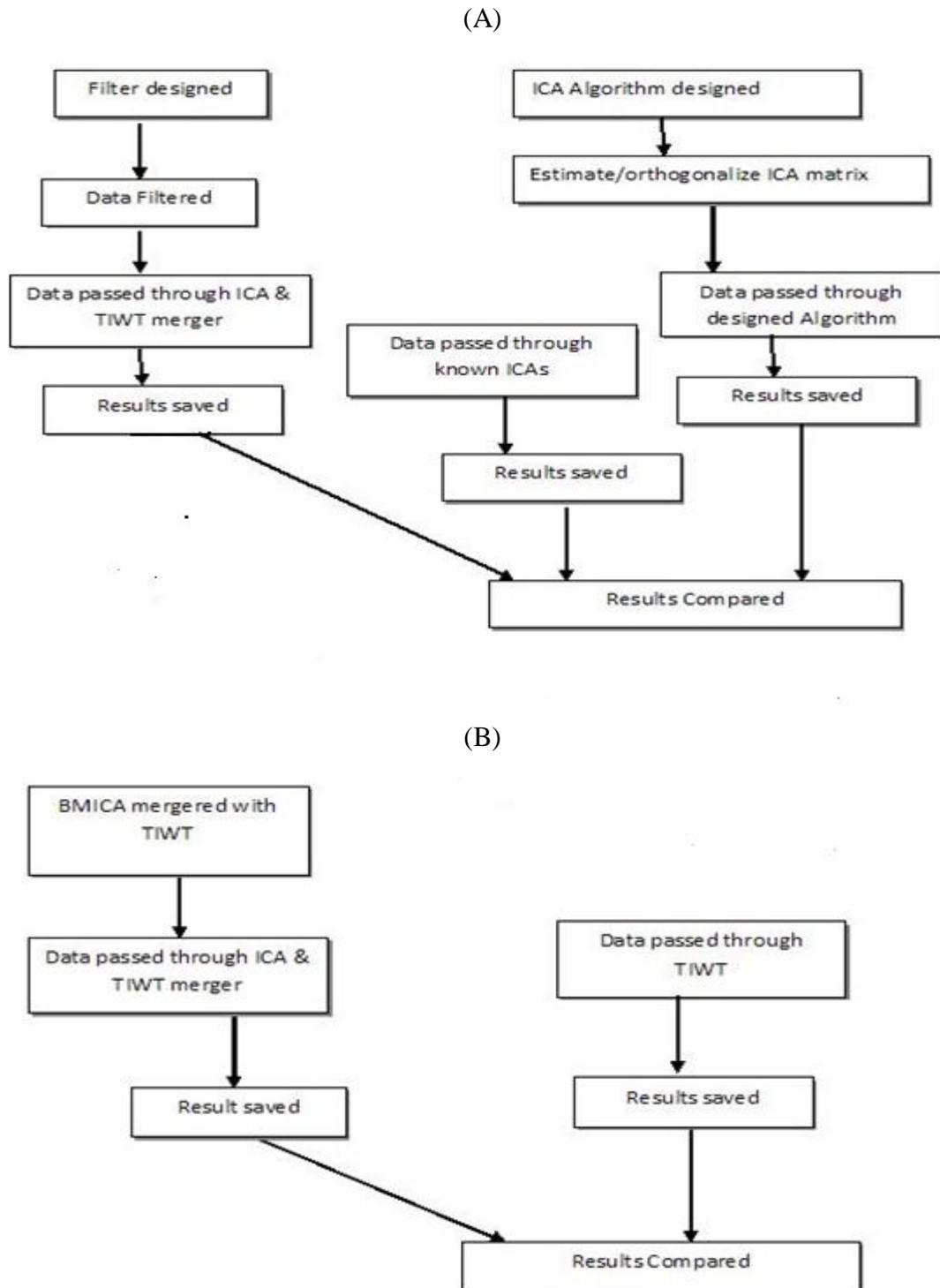


Figure 1.1. Work Flow of Dissertation (A) Algorithm Design using ICA (B) Algorithm Design using TIWT

1.3.1 Algorithm Environment

The algorithms will be implemented within the following environments:

- (i) Laptop Environment 1 - This environment was based on Matrix Laboratory (MATLAB) 7.8.0 (R2009) on a laptop with AMD Athlon 64x2 Dual-core Processor 1.80GHz.
- (ii) Laptop Environment 2 - This environment was based on Matrix Laboratory (MATLAB) 7.10.0.499 (R2010) on a laptop with AMD Athlon 64x2 Dual-core Processor 1.80GHz

Both MATLAB environments contain the ICA and EEGLAB toolboxes which provide interactive graphic user interfaces allowing users to flexibly and interactively process their high-density EEG and other dynamic brain data using ICA. These toolboxes offer wealth of methods for visualizing and modeling event-related brain dynamics, both at the level of individual EEG datasets and/or across a collection of datasets brought together in an EEG studyset. The labs offer extensible, open-source platforms through which new algorithms can be shared with the world research community by contributing 'plug-in' functions that appear automatically in the menus.

1.3.2 Datasets

There are two types of data that can be used in experiments – real data and synthetic data. In synthetic data the source signals are known as well as the mixing matrix A . In these cases the separation performance of the unmixing matrix W can be assessed using the known A and the quality of the unmixed signals y_i can be evaluated using the known source s_i . Biomedical signals however produce unknown source signals. In this dissertation to test and evaluate my algorithms I utilize real EEG data, of different sizes, collected from the following sites:

1.3.2.1 Dataset 1

This data set was obtained from the Swartz Center for Computational Neuroscience (SCCN) at the University of California, San Diego. The data have

been recorded with a sampling rate of 128 Hz from 32 different locations on the scalp, resulting in 32 separate EEG signals. Below is a diagram showing the 32 locations on the scalp and the placement of the 32 measuring tools called electrodes (Figure 1.2).

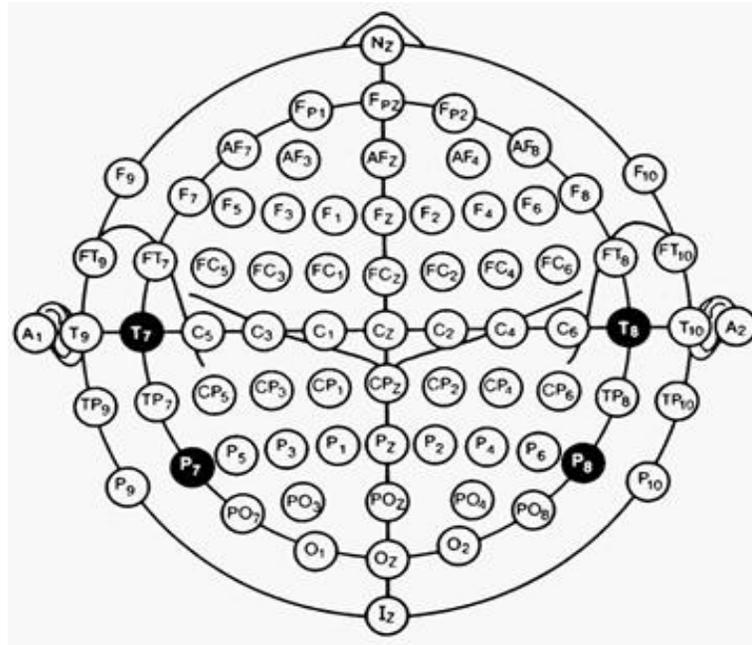


Figure 1.2. Distribution of Electrodes (adapted from *Practical Guide for Clinical Neurophysiologic Testing: EEG*, Thoru Yamada, Elizabeth Meng, Lippincott Williams & Wilkins, 2009)

Each electrode contains 30,504 values. All data are real comprised of EEG signals from both human and animals. Data were of different types namely:

- Data set acquired is a collection of 32-channel data from one male subject who performed a visual task. Figure 1.3 shows 10 signals from this dataset as represented in Matlab.
- Human data based on five disabled and four healthy subjects. The disabled subjects (1-5) were all wheelchair-bound but had varying communication and limb muscle control abilities. The four healthy subjects (6-9) were all male PhD students, age 30 who had no known neurological deficits. Signals

were recorded at 2048 Hz sampling rate from 32 electrodes placed at the standard positions of the 10-20 international system.

- Data set is a collection of 32-channel data from 14 healthy subjects (7 males, 7 females, mean age 26 ranging from age 22 to 46) with normal or corrected to normal vision. They performed a go-nogo categorization task and a go-no recognition task on natural photographs presented very briefly (20 ms). Each subject responded to a total of 2500 trials. The data is CZ referenced, sampled at 1000 Hz using the 10-20 international system. Data focuses on two groups of electrodes (i) frontal (Fz, FP1, FP2, F3, F4, F7, F8) and (ii) occipital (O1, O2, O1', O2', Oz, I, PO9, PO10, PO9', PO10' where the differential activity reached the highest amplitude.

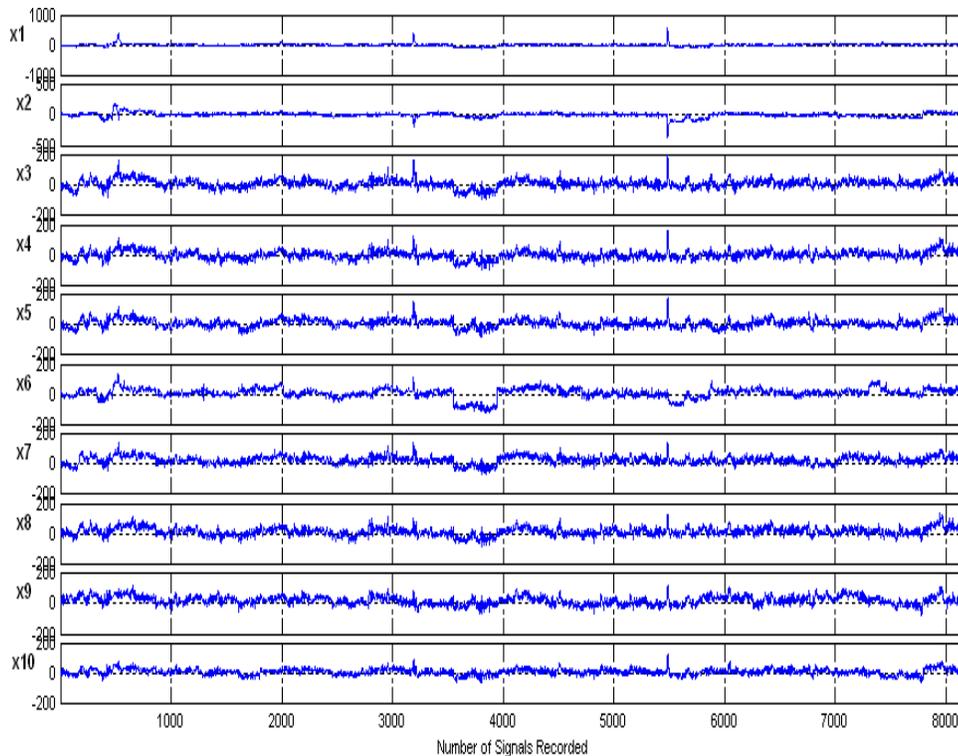


Figure 1.3. Sample EEG signals from a male subject

1.3.2.2 Data Set 2

<http://www.cs.tut.fi/~gomezher/projects/eeg/databases.htm>. Data here contains

- Two EEG recordings (linked-mastoids reference) from a healthy 27-year-old male in which the subject was asked to intentionally generate artifacts in the EEG.
- Two 35 years-old males where the data was collected from 21 scalp electrodes placed according to the international 10-20 System with addition electrodes T1 and T2 on the temporal region. The sampling frequency was 250 Hz and an average reference montage was used. The electrocardiogram (ECG) for each patient was also simultaneously acquired and is available in channel 22 of each recording.

1.3.2.3 Data Set 3

<http://www.meb.uni-bonn.de/epileptologie/science/physik/eegdata.html>. Five data sets containing quasi-stationary, noise-free EEG signals. Each data set contains 100 single channel EEG segments of 23.6 sec duration recorded with a 128-channel amplifier system using an average common reference (omitting electrodes containing pathological activity). These segments, selected and cut out from continuous multichannel EEG recordings, were obtained from (i) five healthy relaxed volunteers using a standardized electrode placement and (ii) five epileptic subjects in seizure activity. These signals were artificially contaminated.

1.3.2.4 Data Set 4

<http://idiap.ch/scientific-research/resources/>. Data here comes from 3 normal subjects during non-feedback sessions. The subjects sat in a normal chair, relaxed arms resting on their legs. All sessions of each subject were acquired on the same day, each lasting 4 minutes with 5-10 minutes breaks in between them. The data is not splitted in trials since the subjects were continuously performing any of the

mental tasks. Data was provided in two ways: raw EEG signals and data with precomputed features.

1.3.2.5 Data Set 5

sites.google.com/site/projectbci. Data here is from a 21 age year old right-handed male with no medical conditions. EEG consists of actual random movement of left and right hand recordings with eyes closed. Each row represents one electrode. The order of electrode is FP1, FP2, F3, F4, C3, C4, P3, P4, O1, O2, F7, F8, T3, T4, T5, T6, F2, CZ, PZ. Recording was done at 500Hz using Neurofax EEG system.

1.4 Outline of Dissertation

This dissertation is focused on artifacts separation of real world EEG signals. In this attempt, my solutions are based on statistical methods called ICA and Wavelet Transformation (WT). In Chapter 2, I establish the basic background information on EEG needed for my analysis. Here I establish the need for separation techniques for the EEG signals. Subsequently, I analyze the techniques utilized in this research showing the relationship between them and the EEG signals in Chapter 3. In this chapter I focus on ICA, WT and Filtering as well as an overview of different ICA algorithms and performance measures utilized within the dissertation.

Chapter 4 focuses on the development of the B-Spline Mutual Information estimator and the resulting algorithm created B-Spline Mutual Information Independent Component Analysis (BMICA). Chapter 5 concentrates on discussing the reliability of BMICA while comparing it to known ICA algorithms.

Chapter 6 discusses the performance of MI based algorithms versus the performance of non MI based algorithms and Chapter 7 discusses the effect of denoising EEG using filters.

Chapter 8 focuses on the effect of denoising using Translation Invariant Wavelet Transform (TIWT) and discusses merging TIWT with ICA and Unscented Kalman Filter (UKF) to create the algorithm named Cycle Spinning Wavelet Transform ICA (CTICA). In this chapter I also focus on improving TIWT with the merger of BMICA to produce BMICA-WT.

I conclude in Chapter 9 by outlining several of the issues that were introduced in this dissertation. Emphasis is given to the novel ideas presented throughout the text. In addition future areas of improvements and development are presented.

Several appendices have been included to present detailed information on the development of the algorithms created.

1.5 Proposed Contributions of the Dissertation

The scientific contributions of this dissertation should include the following.

- Experimental results are given using TIWT, UKF and an ICA method as a method of artifact reduction.
- The use of B-Spline Mutual Information estimator to create a new ICA algorithm.
- The merger of the new ICA and TIWT as a method of artifact reduction.
- The creation of a bridge by comparing the performance of EKF to UKF when applied to EEG signals, especially since there were only investigations on the accuracy of UKF for nonlinear, nonstationary systems not including EEG.

1.6 Summary

Since 1995 when the first algorithm, Infomax was introduced, ICA has been used to identify both temporally and functionally independent source signals in multi-channel EEG. It characteristically separates several important classes of non-brain EEG artifact activity from the rest of the EEG signal into separate sources including eye blinks, eye movement potentials, electromyographic (EMG) and electrocardiographic (ECG) signals, line noise, and single-channel noise. This important benefit of ICA makes it very important to the field of medicine. The removal of artifacts would present cleaner signals thus making it possible to detect with EEG the asymmetries connected to disorders in blood circulation, general disorders connected to poisoning as well as epileptic disorders.

This dissertation will therefore have major contributions in that it will provide new approaches which seek to address the sensitivity of most present ICA algorithms to artifacts thus tackling the robustness problem and presenting solutions. The implementation of these algorithms will provide more accurate EEG recordings to allow for more accurate analysis and interpretations. In denoising data the algorithms also seek to present more accurate analysis thus better interpretation.

CHAPTER 2- Electroencephalograph

2.1 Introduction

The human brain weighs approximately 3 lbs., and it is 3 lbs. of the most complex software on earth. It is so sophisticated it makes the most ultra modern super-computer look like an abacus in comparison. The brain is the boss of the body and consists of about 100 billion cells. Most of these cells are called neurons which are apart of the nervous system. Neurons communicate by sending an electrical charge (potential) down the axon and across the synapse to the next neuron. Because the neurons are not physically connected, chemical messengers called neurotransmitters cross the synaptic gap to get the message to the next neuron [14]. These neurotransmitters then activate corresponding receptors in the post synaptic neuron and generate post synaptic currents which then passes on to next synapse and so on (Figure 2.1). Communication is therefore both electrical and chemical.

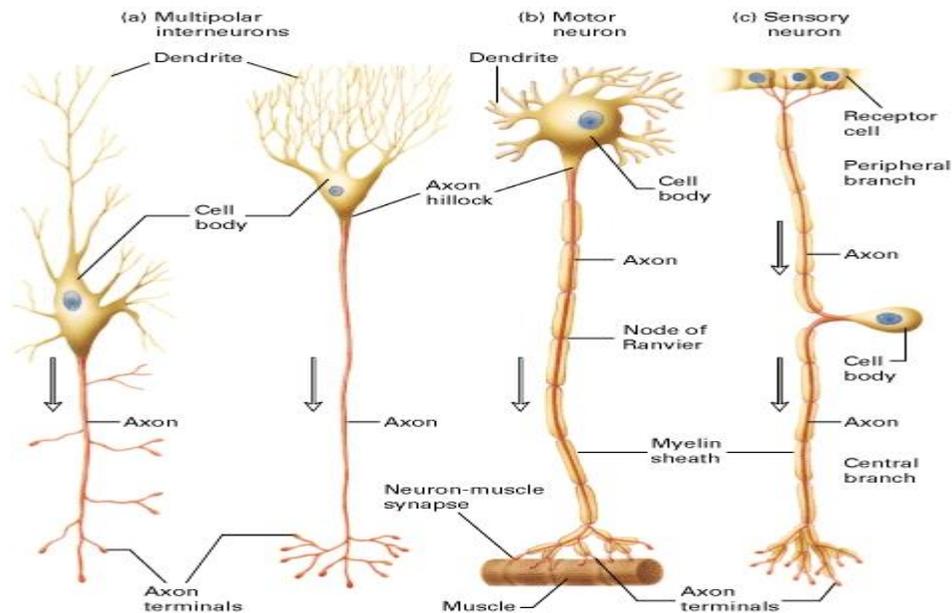


Figure 2.1: Structure of Typical Mammalian Neurons (adapted from *Molecular Cell Biology*, 4th edition. Lodish H, Berk A, Zipursky SL, et al. New York: W. H. Freeman; 2000)

Doctors have learned that measuring this electrical activity can tell how the brain is working and this activity is really a superposition of a large number of electrical potentials arising from several sources (including brain cells i.e. neurons and artifacts) [148]. However, the potentials arising from independent neurons inside the brain, not their superposition, are of main interest to the physicians and researchers to describe the cerebral activity. Direct measurements from the different centers in the brain require placing electrodes inside the head, which needs surgery. This is not acceptable because it causes pain and risk for the subject [161]. A better solution is to calculate the signals of interest obtained on the scalp as seen in Figure 2.2.



Figure 2.2: Placement of EEG electrodes on a patient, monitoring the different sectors of the brain for activities. (adapted from *Hamlet on the Holodeck: the Future of Narrative in Cyberspace*, Janet Horowitz Murray, New York: Free Press, 1997)

These signals are the weighed sums of the neurons activity, the weights depending on the signal path from the brain cell to the electrodes. Because the same potential is recorded from more than one electrode, the signals from the electrodes are supposed to be highly correlated [161]. Researchers therefore collect recordings by attaching tens or hundreds of electrodes, positioned in pairs, on different

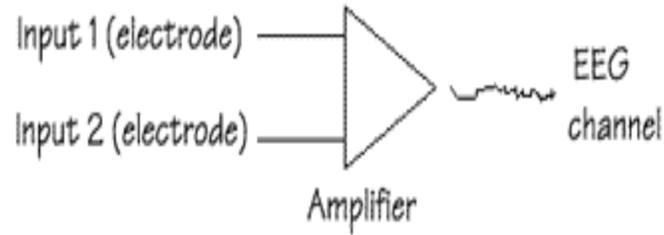


Figure 2.3: Differential Amplifier which produces a signal EEG recording/signal (*adapted from ebme.co.uk*)

locations on the surface of the head. These potentials are simultaneously tested through individuals' amplifiers or channels. Recordings from any one channel does not represent total discharge from a single underlying segment of the brain but represent the difference in potential between two (2) areas under each pair of electrodes (Figure 2.3) [16]. The machine used is called an electroencephalograph (Figure 2.4) and the recordings collected are called electroencephalogram (EEG) signals. From these recordings an accurate appraisal of departures from norms can be made.



Figure 2.4: A Simple Example of an Electroencephalograph Machine (*adapted from http://www.shaktitechnology.com/granqvist_persinger.htm*)

History

Studies about EEG began as early as 1870, but these studies had been carried out in animals. It was five years later that an English physician, Richard Caton discovered the presence of electrical current in the human brain. The information was recorded by the physician but no further research was done with it until Hans Berger. In 1924 Hans Berger, a German neurologist, took Caton's information and put it to a test. He used his ordinary radio equipment to amplify the brain's electrical activity so that he could observe the results on graph paper. Berger noticed that rhythmic changes (brain waves) varied with the individual's state of consciousness and called the recorded signals *Elektroenkephalogram* [33]. This EEG represents complex irregular signals that may provide information about underlying neural activities in the brain.

2.2 Electroencephalograph Measuring System

There are different types of electroencephalographs; however the internationally standardized 10-20 system is the most widely used method to describe the location of scalp electrodes. It is based on the relationship between the location of an electrode and the underlying area of cerebral cortex and usually employs 21 electrodes. The positions are determined by dividing the skull into perimeters by connecting few reference points on human head.

Each perimeter has a letter (to identify the lobe) and a number or another letter to identify the hemisphere location. The letters used are: "F"-Frontal lobe, "T"-Temporal lobe, "C"-Central lobe, "P"-Parietal lobe, "O"-Occipital lobe. Even numbers (2,4,6,8) refer to the right hemisphere and odd numbers (1,3,5,7) refer to the left hemisphere. "Z" refers to an electrode placed on the midline; the smaller the number, the closer the position to the midline. Figure 2.5 shows the actual electrode placement on the head and from these points, the skull perimeters are measured in the transverse and median planes [64]. In this system the "10" and "20" refer to the fact that the actual distances between adjacent electrodes are either 10% or 20% of

the three main measurements: *nasion* (the delve at the top of the nose, level with the eyes)–*inion* (bony lump at the base of the skull on the midline at the back of the head) preauricular points and circumference of the head.

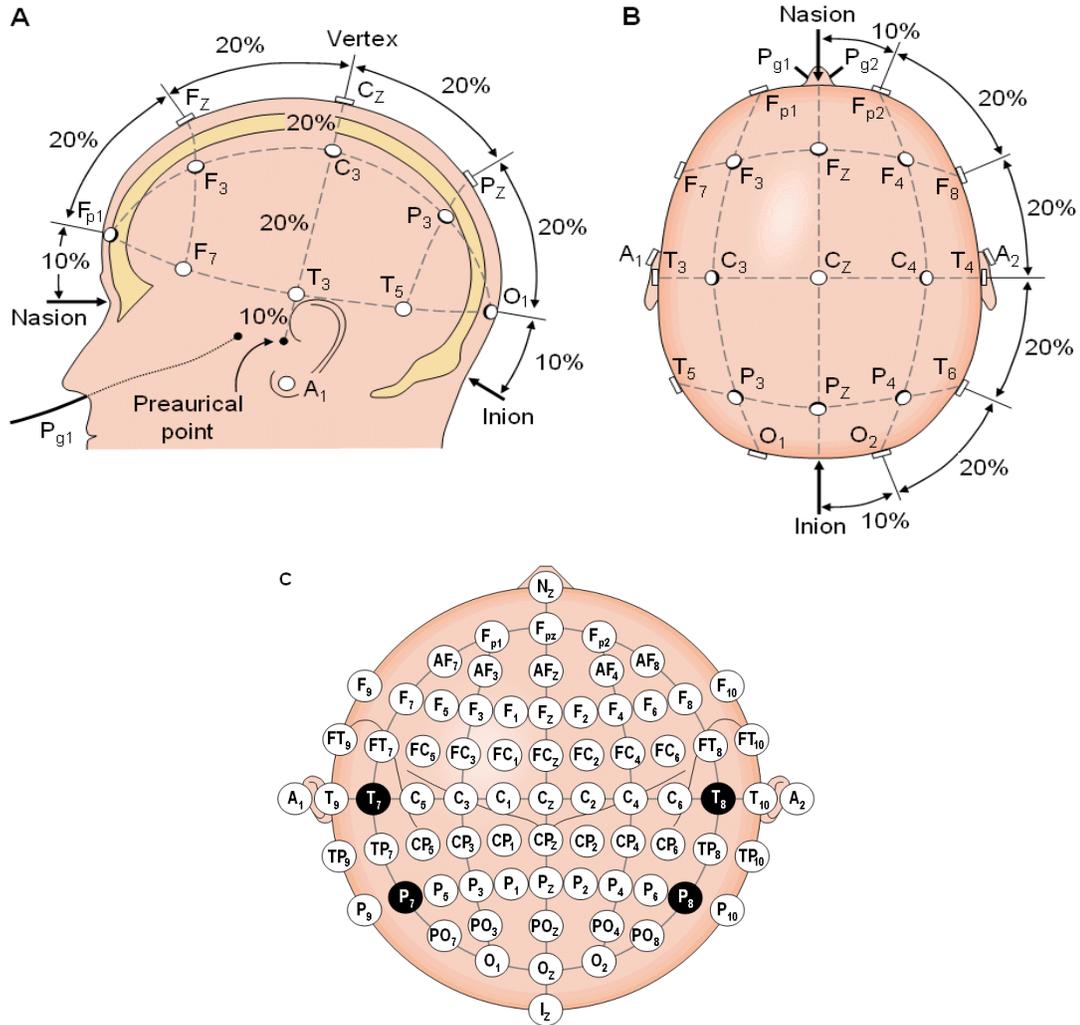


Figure 2.5: The international 10-20 system seen from (A) left and (B) above the head. A = Ear lobe, C = central, Pg = nasopharyngeal, P = parietal, F = frontal, Fp = frontal polar, O = occipital. (C) Location and nomenclature of the intermediate 10% electrodes, as standardized by the American Electroencephalographic Society. (adapted from *Bioelectromagnetism Principles and Applications of Bioelectric and Biomagnetic Fields* Jaakko Malmivuo, Robert Plonsey, Oxford University Press, 1995)

2.3 Wave Analysis of the EEG

In the brain the more neurons that work in synchrony, the larger the potential (amplitude) of the electrical oscillations measured in microvolts (mV) and the faster the neurons work together, the higher the frequency of the oscillations measured in Hertz (Hz). These two parameters: *amplitude* and *frequency* are the primary characteristics of brain waves. EEGs are the recordings of these tiny electrical potentials or waves which are generally less than $300\mu\text{V}$ [142].

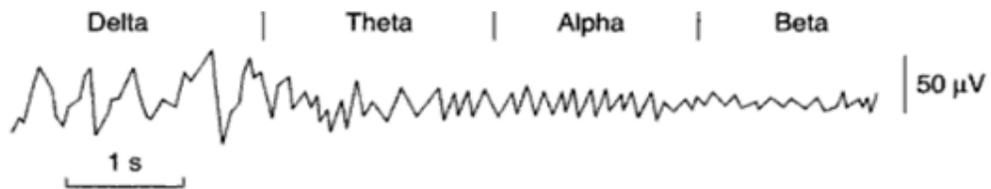


Figure 2.6: Four of the Six Frequency Bands found in EEG Signals (*adapted from Introduction to Biomedical Instrumentation Mandeep Singh, PHI Learning Private Ltd. 2010*)

The waves basically have small amplitudes typically ranging from $1\ \mu\text{V}$ to $100\ \mu\text{V}$ in a normal adult and are approximately $10\ \text{mV}$ to $20\ \text{mV}$ when measured with subdural electrodes such as needle electrodes on the surface of the brain. The frequencies of these EEG waves, emitted from various regions of the brain, range from $0.5\ \text{Hz}$ to $100\ \text{Hz}$. This has presented a great deal of difficulty to researchers trying to interpret the large amount of data they receive from even one EEG recording as depending on the frequency a recording can present six classical categories or bands for the EEG waves - delta, theta, alpha, beta, mu and gamma as described below. Figure 2.6 shows the frequency band for some of the waves.

2.3.1 Alpha (α) waves

These waves, seen in Figure 2.7, were the first to be discovered (around 1908 by Hans Berger) hence why they are called "Alpha waves". These waves predominantly originate from the occipital lobe during wakeful relaxation with closed eyes having frequency ranges of $8\text{--}12\text{Hz}$ with $30\text{--}50\mu\text{V}$ amplitude.

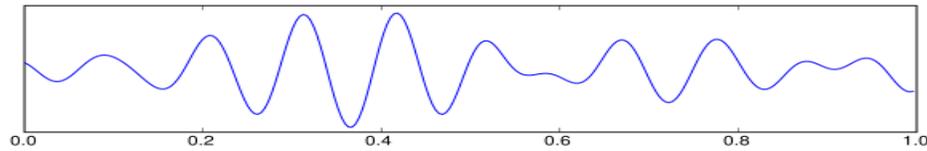


Figure 2.7: One second Recording of EEG Alpha Waves (*adapted from Introduction to Biomedical Instrumentation Mandeep Singh, PHI Learning Private Ltd. 2010*)

Alpha waves are not a measure of peace and serenity, nor are they indicative of an altered state of consciousness. They are indicative of lack of visual processing and lack of focus: the less visual processing and the more unfocused, generally the stronger the alpha waves. When a person closes his eyes and does not do any deep thinking or concentrating on vivid imagery, alpha waves are usually quite strong.

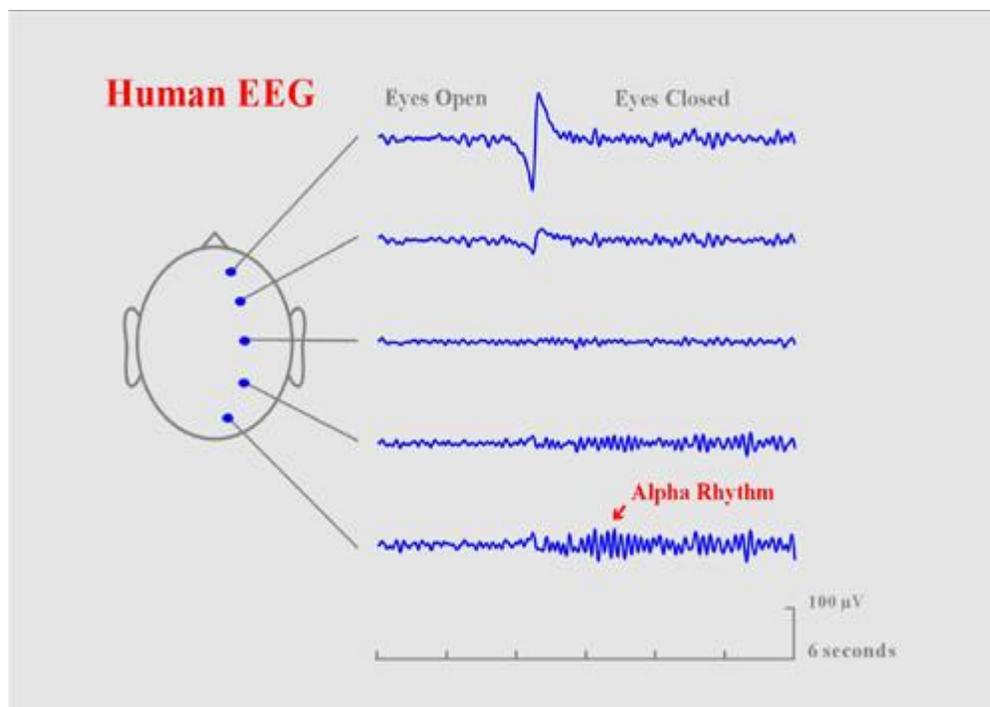


Figure 2.8: A schematic view of the human head viewed from above with the nose at the top. When the eyes are open, the EEG signals show low-voltage random activity. When the eyes close (the time of the large signal in the electrode near the eyes) an alpha rhythm with a frequency of 11 cycles per second occurs at the back of the head. (*adapted from Electroencephalogram (EEG), Terence W. Picton, M.D, Ph.D Rotman Research Institute, 2011*)

The waves generated here, called the occipital alpha waves, are the strongest EEG brain signals (Figure 2.8) usually being detected with the naked eye. Alpha waves generally are seen in all age groups but are most common in adults. Alpha activity disappears normally with attention (e.g., mental arithmetic, stress, opening eyes).

2.3.2 Mu (μ) waves

Mu waves (Figure 2.9) produce oscillations in the 8-13 Hz being located in the motor and sensorimotor cortex. It partly overlaps with other frequencies and reflects the synchronous firing of motor neurons in rest state. The amplitude of Mu varies when the subject performs movement consequently it is also known as the “sensorimotor rhythm”.

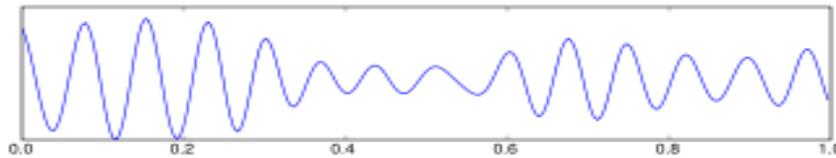


Figure 2.9: One Second Recording of EEG Mu Wave (*adapted from, T. Trott, Electroencephalogram <http://weirddreams.net/electroencephalogram-eeeg/>. 2009*)

2.3.3 Beta (β) waves

Beta waves (Figure 2.10) are in the frequency range of human brain activity 12-30Hz with low voltage 5-30 μ V and are usually split into three sections: High Beta Waves (19Hz+); Beta Waves (15-18Hz); and Low Beta Waves (12-15Hz). These waves are observed in all ages and are the usual waking rhythm of the brain associated with active thinking, active attention, and focus on the outside world or solving concrete problems thus they are typically produced by the left hemisphere of the brain. It has been found that drugs, such as barbiturates and benzodiazepines, augment beta waves.

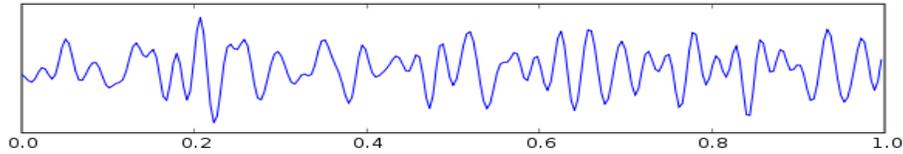


Figure 2.10: One second Recording of EEG Beta Wave (*adapted from Introduction to Biomedical Instrumentation, Mandeep Singh, PHI Learning Private Ltd. 2010*)

2.3.4 Theta (θ) waves

Theta Waves are the second slowest frequency of brain waves with frequency in the 4–7 Hz range (Figure 2.11), regardless of their source. They are associated with the early stages of sleep, the process of day-dreaming, drowsy, states of enhanced creativity, “Super Learning,” deeper relaxation or meditation, and sleep-dream activity. These waves are of high amplitude and appear during states of arousals and powerful surges of emotion. In awake adults, these waves are abnormal if they occur in excess. Several types of brain pathology can give rise to abnormally strong or persistent cortical theta waves.

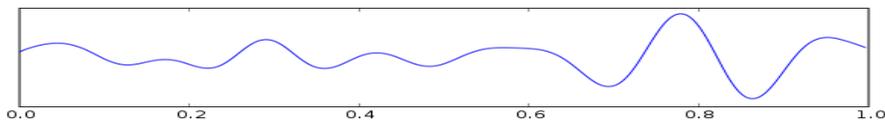


Figure 2.11: One second Recording of EEG Theta Wave (*adapted from Introduction to Biomedical Instrumentation, Mandeep Singh, PHI Learning Private Ltd. 2010*)

2.3.5 Delta (δ) waves

A delta wave (Figure 2.12) is a high amplitude brain wave in humans with a frequency of 1–4Hz and is usually associated with slow-wave sleep (SWS) i.e. deep sleep for all ages. Their activity occurs most frequently during stage N3 SWS, accounting for 20% or more of the EEG record during this stage. Delta waves are abnormal in the awake adult as they are associated with being completely unconscious. These waves are believed to originate in the thalamus in coordination

with the reticular formation and are responsible for the slowest form of mental processing.

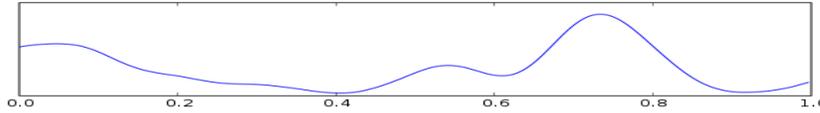


Figure 2.12: One second Recording of EEG Delta Wave (*adapted from Introduction to Biomedical Instrumentation, Mandeep Singh, PHI Learning Private Ltd. 2010*)

Theta and delta waves are known collectively as slow waves.

2.3.6 Gamma (γ) waves

Gamma waves (Figure 2.13) have a frequency between 25 to 100 Hz, though 40 Hz is prototypical. They were initially ignored before the development of digital electroencephalography as analog electroencephalography is restricted to recording and measuring rhythms usually less than 25Hz. Gamma waves are thought to represent binding of different populations of neurons into a network for the purpose of carrying out certain cognitive or motor functions and have long been considered the brain's information and sensory-binding brainwave. They are usually associated with perception, consciousness, higher mental and reasoning activities, high levels of intelligence, compassion, high self-control, and feelings of natural happiness. They have also been linked to having a great memory and an increased perception of reality.

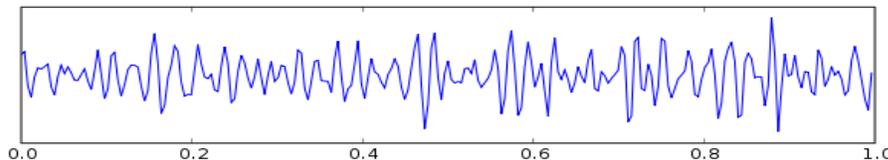


Figure 2.13: One second Recording of EEG Gamma Wave (*adapted from Introduction to Biomedical Instrumentation, Mandeep Singh, PHI Learning Private Ltd. 2010*)

Although none of these waves is ever emitted alone as seen in Figure 2.14, the state of consciousness of the individual may make one frequency more pronounced than the others. For example an alert person displays a low amplitude EEG of mixed frequencies, while a relaxed person produces large amounts of sinusoidal waves, in the 8Hz to 13Hz frequency range, which are particularly prominent at the back of the head.

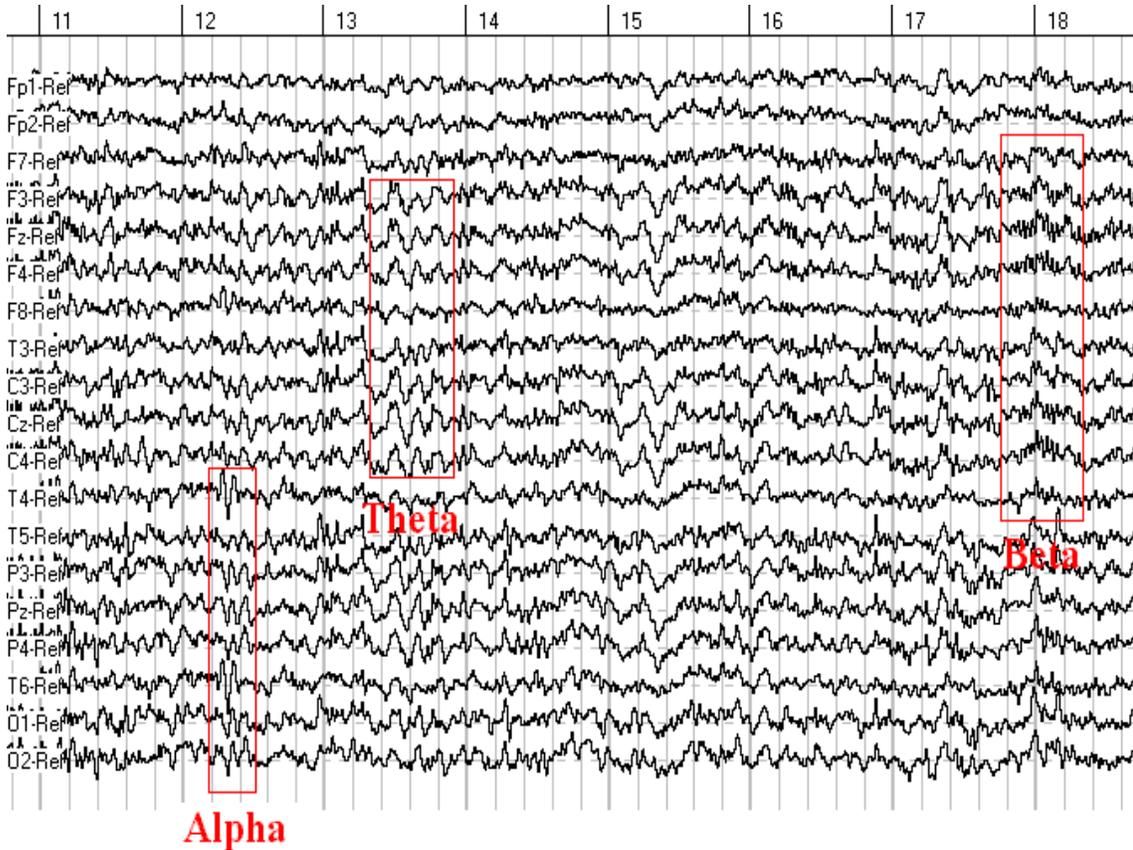


Figure 2.14: Set of EEG Signals showing three of the six bands (adapted from *Brainwaves and EEG examples* http://www.121neurofeedback.com/brainwaves_eeg_examples.html)

2.3.7 Flow of EEG Waves

The EEG signal is closely related to the level of consciousness of a person. As the activity increases, the EEG shifts to higher dominating frequency and lower amplitude. When the eyes are closed, the alpha waves begin to dominate the EEG. When the person falls asleep, the dominant EEG frequency decreases resulting in theta waves. In a certain phase of sleep, rapid eye movement called Rapid eye movement (REM) sleep occurs, the person dreams and has active movements of the eyes, which can be seen as a characteristic EEG signal. In deep sleep, the EEG has large and slow deflections called delta waves. No cerebral activity can be detected from a patient with complete cerebral death. Examples of the above-mentioned waveforms are given in Figure 2.15.

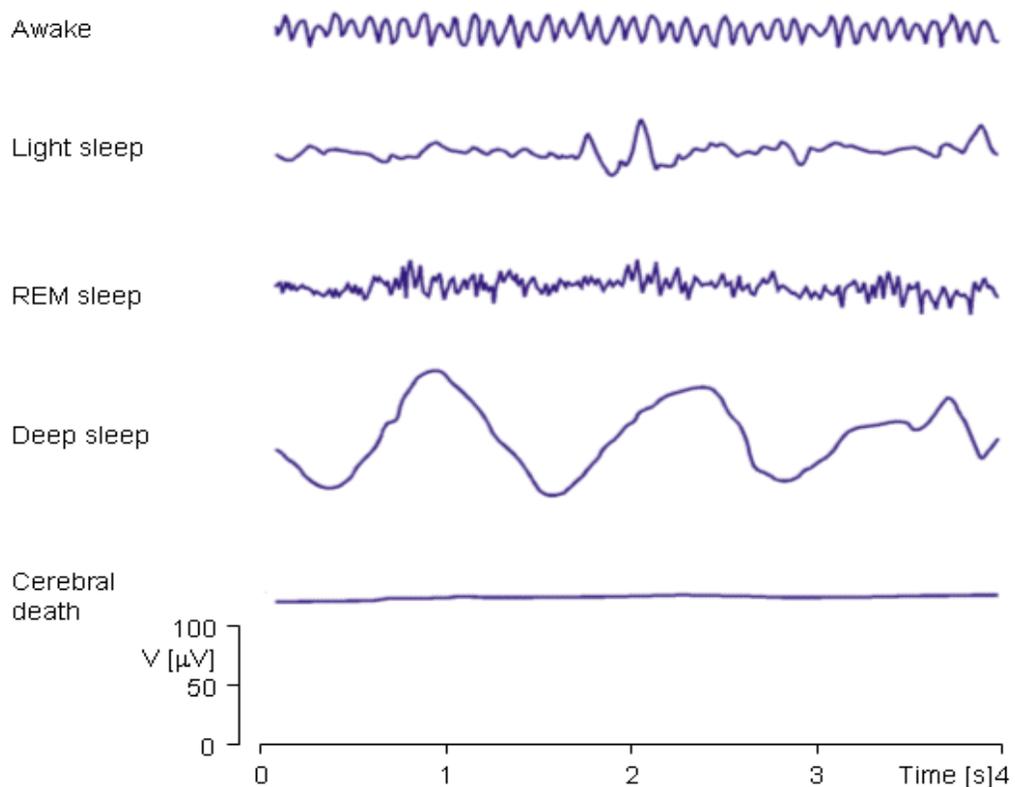


Figure 2.15: EEG activity is dependent on the level of consciousness. (adapted from *Bioelectromagnetism Principles and Applications of Bioelectric and Biomagnetic Fields* Jaakko Malmivuo, Robert Plonsey, Oxford University Press, 1995)

2.4 Uses of EEG

Understanding the brain is a huge part of Neuroscience, and the development of EEG was for the elucidation of such a phenomenon. The analysis of EEG waves has been the subject of several studies since EEG itself represents the brain activity for a subject and gives an objective mode of recording brain stimulation. In neurology EEG is used to:

- Diagnose and Confirm Epilepsy
- Distinguish and characterize seizures for treatment purposes such as epileptic, psychogenic non-epileptic, syncope (fainting), migraine and sub-cortical movement disorders
- Localize the region of brain from which a seizure originates for work-up of possible seizure surgery
- Check for problems with loss of consciousness or dementia.
- Help find out a person's chance of recovery after a change in consciousness.
- Serve as an adjunct test for brain-death.
- Study sleep disorders, such as narcolepsy.
- Monitor the depth of general anesthesia and the amobarbital effect during the Wada test
- Monitor for non-convulsive seizures/non-convulsive status epilepticus
- Help find out if a person has a physical (in the brain, spinal cord, or nervous system) or mental health problem.
- Detect diseases such as Creutzfeldt-Jakob diseases (CJD), Alzheimer's, and Schizophrenia.
- Differentiate "organic" encephalopathy or delirium from primary psychiatric syndromes such as catatonia
- Act as an indirect indicator of cerebral perfusion in carotid endarterectomy
- Prognosticate, in certain instances, in patients with coma

EEG used to be a first-line method for the diagnosis of tumors, stroke and other focal brain disorders, but this use has decreased with the advent of anatomical imaging techniques such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT).

In cognitive neuroscience EEG is used to investigate the neural correlates of mental activities from low-level perceptual and motor processes to high-order cognition such as attention, memory, and reading. In cognitive psychology it is used to get a better understanding of how the brain influences the way a person thinks, feels and acts.

2.5 Artifacts in EEG

EEG is widely used by physicians and scientists to study brain function and to diagnose neurological disorders. Any misinterpretations can lead to misdiagnosis. These signals must therefore present a true and clear picture about brain activities as seen in Figure 2.16. The poor spatial resolution of scalp EEG (limited to 1 centimeter) is due to the low conductivity of the skull, the cerebrospinal fluid and the meninges, which cause a reduction and dispersion of the activity originated in the cortex.

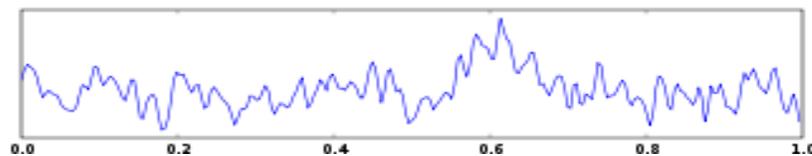


Figure 2.16: One second Recording of clean pure EEG Signal (*adapted from Introduction to Biomedical Instrumentation, Mandeep Singh, PHI Learning Private Ltd. 2010*)

Scalp EEG is also very sensitive to subject movement and external noise such as activation of the head, musculature, eye movements, interference from nearby electric devices, and changing conductivity in the electrodes due to the movements of the subject or physicochemical reactions at the electrode sites [20].

All of these activities that are not directly related to the current cognitive processing of the subject are collectively referred to as background activities. EEG signals are therefore highly attenuated and mixed with these non-cerebral impulses called **artifacts** or **noise** which fall into two categories – physiologic and extra-physiologic [95]. A true diagnosis can only be seen when all these noises are removed.

2.5.1 Physiologic Artifacts

Any source in the body which has an electrical dipole or generates an electrical or magnetic field is capable of producing physiologic artifacts. They are generated by some biological activities in the human body. Physiological signals have widely different sources. Below we discuss those which tend to overlay the EEG signals

2.5.1.1 Electrooculogram (EOG)

Eye artifacts are often measured more directly in the electrooculogram (EOG), where pairs of electrodes are placed above and around the eyes. Unfortunately, these measurements are contaminants of the EEG signals of interest and so simple subtraction is not a removal option even if an exact model of EOG diffusion across the scalp is available [71]. In the frequency domain, ocular artifacts increase the power of EEG signals from 2Hz to 20 Hz. These artifacts are of two types:

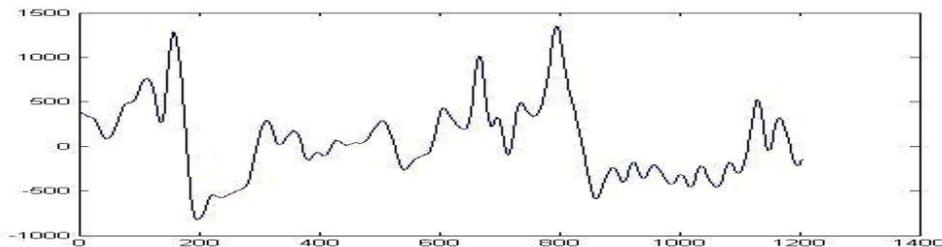


Figure 2.17: Eye Blink (adapted from L F Araghi, A New Method for Artifact Removing in EEG Signals, Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol 1 (IMECS 2010))

Eye Blinking: The eye blink artifact (Figure 2.17) is very common in EEG data. It produces a high amplitude signal that can be many times greater than the EEG signals of interest. Because of its high amplitude an eye blink can corrupt data on all electrodes, even those at the back of the head.

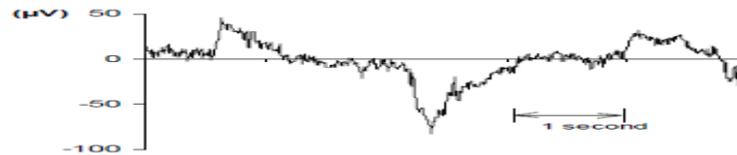


Figure 2.18: Eye Movement (*adapted from M. van de Velde, Signal Validation in Electroencephalography Research, PhD Thesis, Eindhoven University of Technology, 2000*)

Eye Movement: Eye movement artifacts (Figure 2.18) are caused by the reorientation of the retinocorneal dipole [71, 125]. This artifact's diffusion across the scalp is stronger than that of the eye blink artifact.

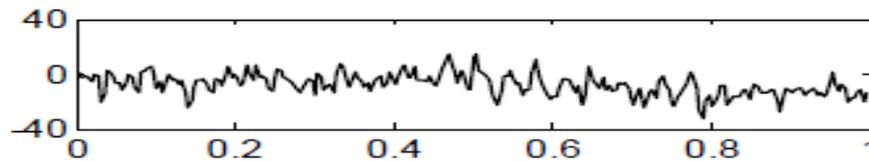


Figure 2.19: EOG Signal containing both Eye Blink and Eye movement artifacts (*adapted from JJM Kierkels, Validating and Improving the Correction of Ocular Artifacts in Electroencephalography, Dissertation Abstracts International, 2007*)

Eye blinks and movements often occur at close intervals producing effects shown in Figure 2.19 and Figure 2.20.

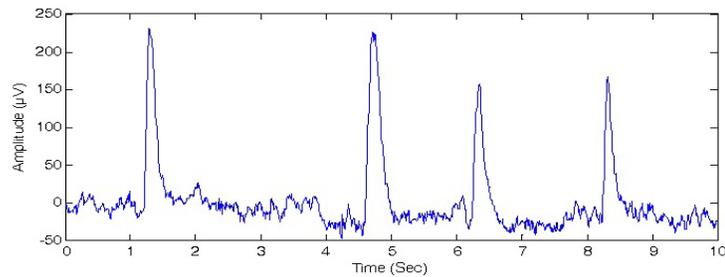


Figure 2.20: EEG contaminated with EOG producing spikes (*adapted from P. Senthil Kumar, R. Arumuganathan, K. Sivakumar, C. Vimal, An Adaptive method to remove ocular artifacts from EEG signals using Wavelet Transform Journal of Applied Sciences Research, 5(7): 741-745, 2009*)

2.5.1.2 Cardiograph (ECG/ EKG)

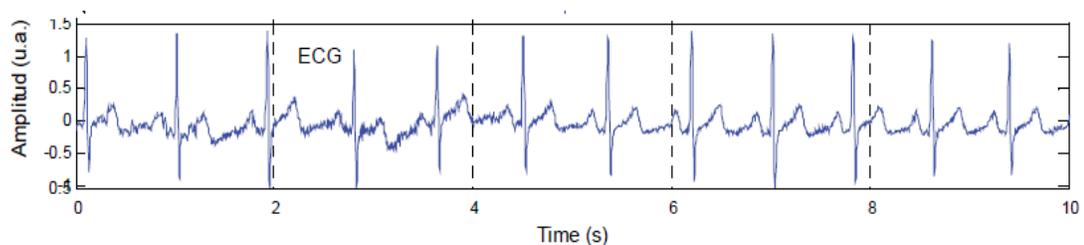


Figure 2.21: Ten Seconds Cardiac Movement Artifact (*adapted from Noise Removal from EEG Signals in Polysomnographic Records Applying Adaptive Filters in Cascade, M. Agustina Garcés Correa, et al. Adaptive Filtering Applications, 2011*)

The pulse, or heart beat artifact (Figure 2.21), normally measured by a cardiograph, occurs when an electrode is placed on or near a blood vessel [95]. The expansion and contraction of the vessel introduce voltage changes into the recordings. The artifact signal has a frequency near 1.2Hz, but can vary with the state of the patient. This artifact can appear as a sharp spike or smooth wave [13]. EKG artifacts get more prevalent with aging. An example of an EEG mixed with EKG and corrupted with line interference is illustrated in Figure 2.22.

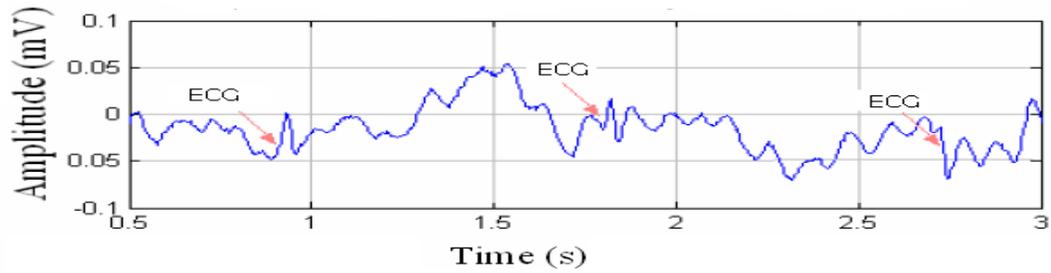


Figure 2.22: EEG Signal corrupted with ECG/EKG (adapted from *Artifact Removal from EEG Signals using Adaptive Filters In Cascade*, A Garcés Correa et al, *Journal of Physics: Conference Series* 90, 2007)

2.5.1.3 Electromyogram (EMG)

Muscle activity can be caused by activity in different muscle groups including the neck and facial muscles. Frontalis and temporalis muscles (e.g., clenching of jaw muscles) are common causes. Generally, the potentials generated in the muscles are of shorter duration than those generated in the brain (Figure 2.23) and are identified easily on the basis of duration, morphology, and rate of firing (i.e., frequency) [95]. Particular patterns of electromyogram (EMG) artifacts can occur in some movement disorders. Essential tremor and Parkinson disease can produce rhythmic 4-Hz to 6-Hz sinusoidal artifacts that may mimic cerebral activity.

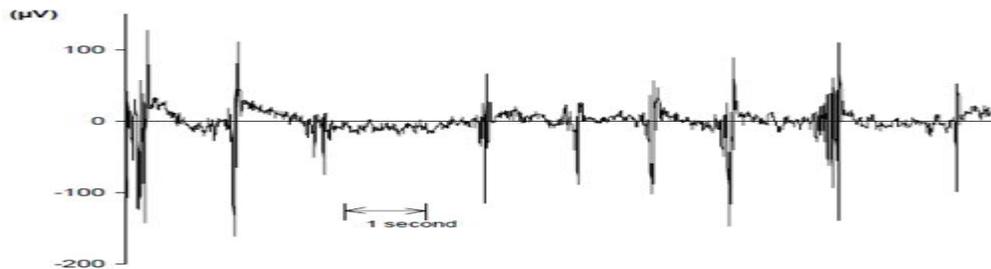


Figure 2.23: Muscle Activity Wave - Chewing (adapted from M. van de Velde, *Signal Validation in Electroencephalography Research*, PhD Thesis, Eindhoven University of Technology, 2000)

2.5.1.4 Chewing and Sucking Movement – Glossokinetic

In addition to muscle activity, the tongue (like the eyeball) functions as a dipole, with the tip negative with respect to the base. In this case, the tip of the tongue is the most important part because it is more mobile. The artifact produced (Figure 2.24) by the tongue has a broad potential field that drops from frontal to occipital areas, although it is less steep than that produced by eye movement artifacts [95]. Chewing and sucking can produce similar artifacts. These are commonly observed in young patients, however, they also can be observed in patients with dementia or those who are uncooperative. Minor tongue movements can contaminate the EEG, especially in Parkinsonian and Tremor disorders.

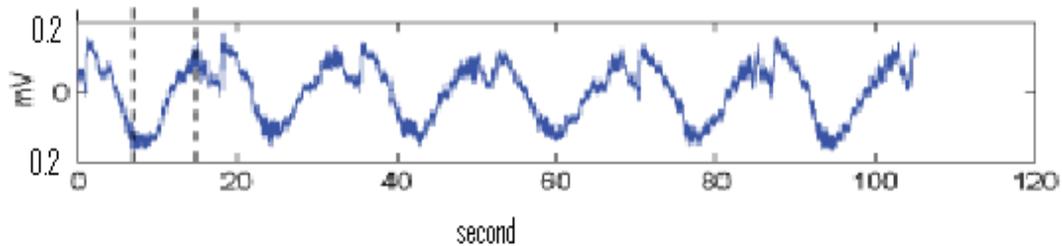


Figure 2.24: Recording of a Glossokinetic Signal (*adapted from Machine Learning Group, Department of Computer Science Pohang University of Science and Technology <http://mlg.postech.ac.kr/research/bci.html>*)

2.5.1.5 Galvanic Skin Response (GSR)

All tissues in the human body, including skin, have the ability to conduct electricity. This is how the nerves communicate. The skin also has electrical activity, which is in constant, slight variation, and can be measured and charted. The skin's electrical conductivity fluctuates based on certain bodily conditions, and this fluctuation is called the Galvanic Skin Response (GSR) (Figure 2.25). Sudden changes in emotion, such as fright, can trigger GSR, as can other types of changes, such as the hot flashes that are characteristic of menopause. Changes in EEG

recordings occur as these changes produce Sodium Chloride (NaCl) and Lactic Acid ($C_3H_6O_3$) from sweating which react with the metals of the electrodes resulting in huge slow baseline sways. Significant asymmetry also can be observed when a collection of sweat (e.g. subgaleal hematoma) is under or in the skin.

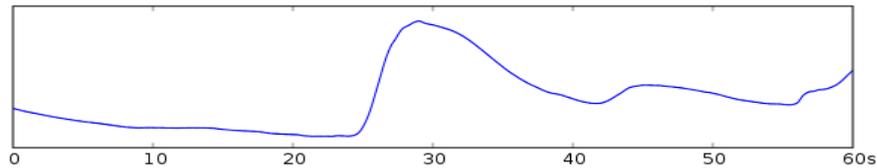


Figure 2.25: One 60 Seconds recording of a GSR Signal (*adapted from Circuit Surgery* <http://www.circuitsurgery.co.za/>)

2.5.2 Extra-physiologic Artifacts

Extra-physiologic artifacts include interference from electric equipment, kinesiological artifacts caused by body or electrode movements, and mechanical artifacts caused by body movement.

2.5.2.1 The Electroencephalograph Machine

Movement by the patient, or even just the settling of electrodes, may cause *electrode pops* - changes of the conduction between electrodes and skin. Morphologically this appears as single or multiple sharp waveforms (Figure 2.26) due to abrupt impedance change [95]. It is identified easily by its characteristic appearance (i.e. abrupt vertical transient that does not modify the background activity) and its usual distribution, which is limited to a single electrode. In general, sharp transients that occur at a single electrode should be considered artifacts until proven otherwise. . At other times, the impedance change is less abrupt, and the artifact may mimic a low-voltage arrhythmic delta wave [95].

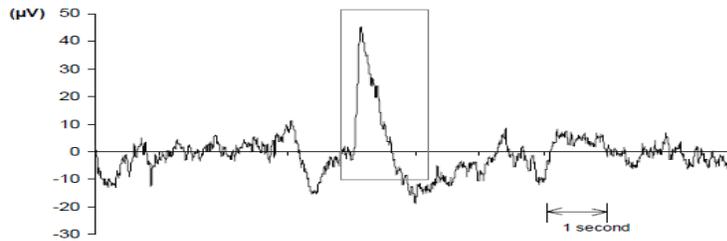


Figure 2.26: Example of Electrode Pop: a sudden sharp edge in the recorded signal, followed by an exponential decay, obscuring the EEG (*adapted from M. van de Velde, Signal Validation in Electroencephalography Research, PhD Thesis, Eindhoven University of Technology, 2000*)

2.5.2.2 Power Lines – Alternating Currents (50-60Hz)

Strong signals from Alternating Current (A/C) power supplies (Figure 2.27) can corrupt EEG data as it is transferred from the scalp electrodes to the recording device. The problem usually arises when the impedance of one of the active electrodes becomes significantly large between the electrodes and the ground of the amplifier. In this situation, the ground becomes an active electrode that, depending on its location, produces the 50-60-Hz artifact. This artifact is often filtered by notch filters, but for lower frequency line noise and harmonics this is often undesirable. If the line noise or harmonics occur in frequency bands of interest they interfere with EEG that occurs in the same band [74]. Notch filtering at these frequencies can remove useful information. Line noise can corrupt the data from some or all of the electrodes depending on the source of the problem.



Figure 2.27: Line Interference of 50Hz (*adapted from Artifact Removal from EEG Recordings – An Overview, Rohtash Dhiman, et al., National Conference on Computational Instrumentation, CSIO 2010*)

2.5.2.3 Infusion Motor Artifact (IMA)

An Intra Venous (IV) drip within a person can cause rhythmic, fast, low-voltage bursts, which may be confused for spikes. With the increasing use of automatic electric infusion pumps, a new type of artifact - IMA, has arisen. Morphologically, IMA appears as very brief spiky transients, sometimes followed by a slow component of the same polarity.

2.6 Denoising EEG Signals

Contamination of EEG data can occur at many points during the recording process. Most of the artifacts considered here are biologically generated by sources external to the brain [83]. Improving technology can decrease externally generated artifacts, such as line noise, but biological artifact signals must be removed after the recording process, thus denoising procedures must be introduced to remove these biological overlays from the EEG signals.

Denoising stands for the process of removing noise i.e. unwanted information, present in an unknown signal. Real EEG recordings are a combination of artifacts (noise) and the pure EEG signal. Mathematically it is defined as:

$$E(t) = S(t) + N(t) \quad (2.1)$$

where $S(t)$ is pure EEG signal, $N(t)$ is the artifact, $E(t)$ represents the recorded signal and t is the time recording was taken. The presence of these artifacts, as seen before, introduces spikes which can be confused with neurological rhythms. They also mimic EEG signals, overlaying these signals resulting in signal distortion (Figures 2.19, 2.20, 2.22). Correct analysis is therefore impossible, resulting, as stated in Chapter 1, in misdiagnosis in the case of some patients. Noise must be eliminated or attenuated.

Numerous methods have been proposed by researchers to achieve this denoising process in EEG and are even reviewed in [25, 77]. A brief write-up about the existing techniques for denoising EEG is described in the next section.

2.7 Methods

The removing of artifacts from an EEG recording can be categorized into two groups:

- (i) Artifact rejection - removes the EEG signal which contains the artifact and
- (ii) Artifact correction - removes the artifacts from the EEG signal while keeping the pure EEG signal.

The most popular methods for each category are described below.

2.7.1 Artifact Rejection

2.7.1.1 Basic Artifact Rejection

One common denoising strategy is to reject all EEG epochs containing artifacts larger than some arbitrarily selected EEG voltage level. This is artifact rejection. When limited data are available, or artifacts, such as EOG, occur too frequently the rejection of epochs contaminated with the artifacts usually results in a considerable loss of information and may be impractical for clinical data. Since EEG and some artifacts occupy the same frequency band, this method is ineffective [42].

2.7.1.2 Regression

Traditional artifacts correction procedures use a regression based approach. Widely used methods are based on regression in time domain or frequency domain techniques [177]. Regression analyses rely on a clean measure of the artifact signal to be subtracted out [177]. One concern often raised about the regression approach is bidirectional contamination. If artifacts potentials can contaminate EEG recordings,

then brain electrical activity can also contaminate the artifacts recordings. Therefore, subtracting a linear combination of the recorded artifacts from the EEG may not only remove artifacts but also interesting cerebral activity. Review of the technique is in [24-25].

2.7.1.3 Filtering

In order to reduce the cerebral activity, Lins *et al.* [100] suggested low-pass filtering of the artifacts. They recognized however that low-pass filtering removes all high frequency activity from the EOG signal, both of cerebral and ocular origins. The use of adaptive filtering, such as Bayesian [28, 170], prior to applying regression correction may substantially reduce problems from bidirectional contamination. Use of adaptive digital filters for artifacts removal, however requires a suitable reference model for training the filter.

2.7.2 Artifact correction

2.7.2.1 Principal Component Analysis (PCA)

PCA is a class of methods based on decomposing the EEG and artifacts into spatial components, identifying artifactual components and reconstructing the EEG without the artifactual components. Lagerlund *et al.* [94] used PCA to identify the artifactual components. Statistically, PCA decomposes the signals into uncorrelated, but not necessarily Independent Components (ICs) that are spatially orthogonal and thus it cannot deal with higher-order statistical dependencies. PCA also cannot completely separate artifacts from brain signals especially when they both have comparable amplitudes [94].

2.7.2.2 Independent Component Analysis (ICA)

ICA was developed in the context of blind source separation (BSS) problems to form components that are as independent as possible [23, 69]. Scott Makeig *et al.* [103] reported the first application of ICA for EEG data analysis by using the algorithm of Bell and Sejnowski [8] for ICA. They showed that ICA can separate

neural activity from muscle and blink artifacts in spontaneous EEG data. Jung *et al.*, [71] also showed using an extended version of the Infomax algorithm [8] that ICA can effectively detect, separate and remove activity in EEG records from a wide variety of artifactual sources. Vigon *et al.* [165] compared four methods of artifact removal and found that the two ICA methods, using Infomax and JADE were significantly better than PCA and simple EOG subtraction. The limitation of present ICA algorithms is that there is no guarantee that any particular algorithm can capture the individual source signals in its components [91].

2.7.2.3 Wavelet Transform

The newest form of denoising method is Wavelet Transform (WT). It has been used to study EEG signals [11, 111, 130, 134] successfully because of its good localization properties in time and frequency domain [40]. There have been many approaches to denoising using WT; those based on shrinkage are the most popular [110] where the EEG signals are decomposed into wavelets and noise removal done using thresholding and shrinkage. Akin [1] in his research compared WT with fast Fourier transform and found that WT was better in detecting brain diseases. His research was confirmed by Hermann *et al* [54]. Unser *et al* [163] showed that wavelet is good at denoising EEG signals as well as other biomedical signals. WT's capability in transforming a time domain signal into time and frequency localization helps to understand the behaviour of a signal better. WT however has limitations such as Gibbs phenomena [21].

2.8 Summary

The advantage of artifact rejection is the complete elimination of artifacts however the disadvantage is the loss of EEG signals. Relatively clean EEG signals can be obtained from artifacts contaminated EEG signal by applying an artifact correction method. However, artifacts may not be removed thoroughly. The relatively clean EEG signal may still contain artifacts. A combination of both

categories could alleviate disadvantages and so in this dissertation we will focus on denoising EEG, the desired signals from the artifacts utilizing three of the aforementioned approaches. These approaches are from both categories – artifacts rejection and correction, and will be discussed in details in the following chapter.

Chapter 3 – Denoising Methods

Computer based methods for analysis and interpretation of biological signals have been the subject of intense research. It is obvious that automated systems for biological signal processing such as noise removal considerably improve or support the judgment of physicians that perform the signal analysis. The methods used to produce these helpful signals are therefore important. This chapter focuses on the three methods utilized within this research. It will also describe the performance measures utilized as well as the different ICA algorithms.

3.1 Independent Component Analysis

It is often said that we suffer from “information overload” but we actually suffer from “data overload”. This is because we have access to a large amount of data containing relatively small amount of useful information [152]. We suffer this way in my daily lives as well as in the science disciplines. There is therefore the need to extract this useful information from data.

In the sciences a set of measured signals is essentially a mixture of underlying factors which are the driving forces to the signal set. Here ICA promises to reveal these forces which are underlying i.e. ICA will extract these factors called source signals which are buried within a measured signal [152].

3.1.1 ICA and Its History

3.1.1.1 What is ICA?

ICA belongs to a class of Blind Source Separation (BSS) methods for separating data into underlying informational components. A method is considered to be “blind” when it can separate data into source signals even if there is very little known about the nature of those source signals [152]. The ICA name suggests a separation of a set of signal mixtures into a corresponding set of statistically

independent component signals or source signals [39]. The mixtures can be sound, electrical signals such as EEG or image such as fMRI. For ICA to work as defined there are some assumptions made:

1. Independence: Whereas electrical signals (s) are statistically independent their mixture is not i.e.

$$\langle s_1^a, s_2^b \rangle = \langle s_1^a \rangle \langle s_2^b \rangle \quad \text{for all } a \text{ and } b \quad (3.1)$$

This is because each source signal is shared between both mixtures such that the resultant commonality between signal mixtures ensures that they cannot be independent [152]. This mathematical definition of independence means that we can generate the additional information necessary to recover the original signals.

2. Complexity: The temporal complexity of any mixture is greater than or equals to its simplest complexity i.e. its least complexity constituted source signal. This ensures that extracting the least complex signal from a set of signal mixtures yield a source signal [152].

Based on the assumption that if different signals are from different sources then those signals are statistically independent ICA works on the implication of the reversal of this assumption resulting in assumption 3:

3. If statistically independent signals can be extracted from signal mixtures then these extracted signals must be from different sources.

ICA therefore separates the source into statistically independent signals. If the assumption is valid then each of the signals extracted by ICA will have been generated by a different source and will therefore be a desired signal.

3.1.1.2 History of ICA

The method ICA, although not the name, formulated in 1983 by Herault and Jutten in an attempt to solve the BSS problem in signal processing [68, 75]. Unlike pervious methods the crucial step was to assume that the underlying signals were

independent of each other. They therefore suggested that the BSS problem could be solved by forcing the data towards independence. All through the 1980's the method was utilized by mostly French researchers with limited influence internationally. It was during this time that Cardoso and Comon produced their early ICA papers [62]. The method was put on a sound theoretical basis in 1991 by Jutten and Herault [69] and Comon *et al.* [70]. Jutten coined the name ICA in analogue with PCA where the latter analysed the mathematical foundation underpinning the procedure.

ICA was formally defined in 1994 by Comon [23] in which he proposed Mutual Information (MI) as a natural measure of independence albeit a difficult one in practice. He showed that ICA can be applied beyond Neural Networks to applications such as identifying stock market trends and separating body rhythms in biomedical applications. The method attained wider attention during this time and research improved from being small and narrow. Research continued to grow to the point that in 1999 the 1st international workshop on ICA was held in Aussois, France.

3.1.2 ICA Model

The basic ICA model is a discrete time mode in which M sources $s_m(t)$ are instantaneously mixed and the resulting mixture, corrupted by other sources (artifacts), is observed. Writing the source signal at instant t in vector form where $s(t)=[s_1(t),s_2(t),s_3(t)...s_n(t)]$, the N-dimensional $x(t)=[x_1(t),x_2(t),x_3(t)...x_n(t)]^T$ and a possibly nonlinear mixture corrupted by additive or noise $n(t)$ results in :

$$x(t) = f \ s(t) + n(t) \tag{3.2}$$

where $f: R^M \rightarrow R^N$ is an unknown function [127]. The BSS goal is to invert the mixing function f and recover the sources. The quantifier “blind” signifies that little is known about the quantities on the right hand side (RHS) of the equation; the mixing function $f()$, noise $n(t)$ and of course the source $s(t)$ themselves are unknown and must be estimated.

Traditional treatments of ICA make assumption that the source $s(t)$ are linearly mixed by a mixing matrix $A \in R^{N \times M}$. This observation is assumed to be generated by [62]

$$x(t) = As(t) + n(t) \quad (3.3)$$

This is called the generative or noisy model [62]. $S(t)$ is still unknown because there are $N+M$ unknown signals (M sources and N noise) and N known signals (the observations). Most known ICA algorithms however are based on the noise-free model as defined in Eq (1.1) [61].

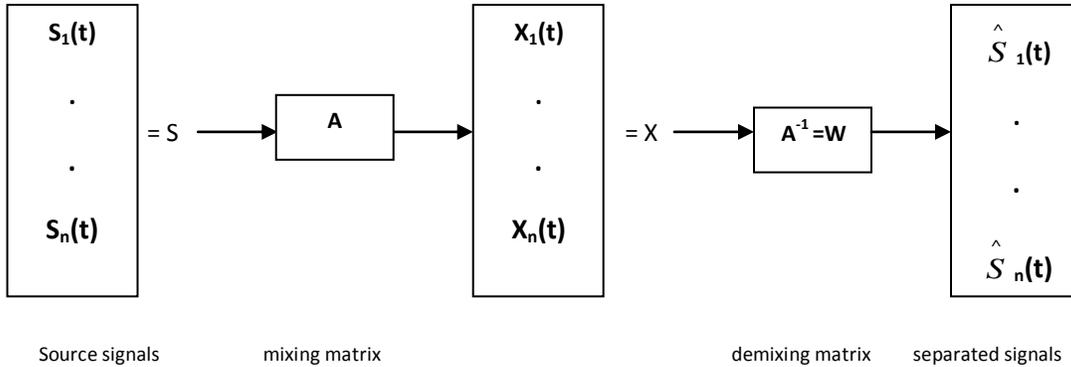


Figure 3.1: Mathematical model for ICA decomposition

3.1.3 Estimating ICA Model

Using assumption 3, described in the afore page where independent signals can be extracted from a mixture, the estimated ICs of the ICA model can be:

$$y(t) = As(t) \rightarrow \hat{s}(t) = Wx(t) \quad (3.4)$$

where W , the demixing matrix is an estimate of the (pseudo)inverse of A i.e

$$W = A^{-1} \quad (3.5)$$

This process is described by Eq. (3.4) and a schematic illustration of the mathematical model as shown in Figure 3.1.

Whether noisy or noise-free, the ICA model amounts to selecting an adequate algorithm which can be decomposed into two parts:

- (i) Objective (contrast) function and
- (ii) Optimization function

Figure 3.2 shows the relationship between (i) and (ii)

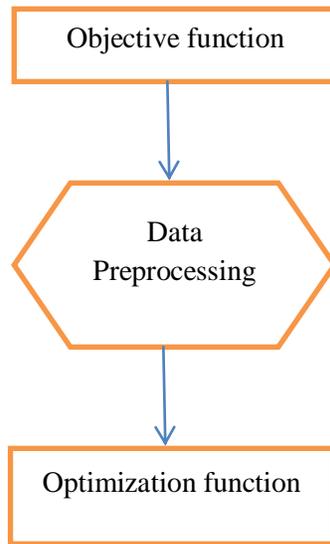


Figure 3.2: Generalized ICA Algorithm

3.1.3.1 Objective Function

The objective function, F , e.g. entropy is a real function of the probability distribution used to measure the mutual independence of the extracted ICs. It is used to estimate the data model which is then minimized or maximized. The choice of this function determines the statistical properties (i.e. robustness, consistency and asymptotic variance) of the ICA method [58, 62]. There are different approaches namely:

- (i) Likelihood as used in Infomax

- (ii) Information-Theoretic measure e.g. Mutual Information used in Pearson-ICA and
- (iii) Tensorial methods as used in JADE

The objective function utilized in this dissertation is based on (ii) and described in Chapter 4.

3.1.3.2 Data Processing

Signals often contain noise $n(t)$ which affects the ICA algorithm resulting in an estimated $x(t)$ which is different from the original $s(t)$. Most ICA algorithms contain some form of preprocessing steps which are used to decrease the dimensionality of a multi-linearity data set. These steps are defined in [61] as:

Centering. Centering is removing the mean from each source [166]. If x has to be centered, the mean of the data is subtracted from the actual data to make it zero mean, i.e.

$$\mathbf{x}_c = \mathbf{x} - \mathbf{E}[\mathbf{x}] \quad (3.6)$$

After the estimation of the mixing matrix A , the mean is added back to the data.

Whitening or sphering. Whitening is done to make the mixed signals uncorrelated. The aim of ICA lies in finding a linear transformation W such that the output signals are as independent as possible thus a linear transform is applied to the data x so that the covariance matrix of the transformed data x_w equals unity:

$$\mathbf{E} \mathbf{x}_w \mathbf{x}_w^T = \mathbf{I} \quad (3.7)$$

This transformation is always possible, for example by using the eigenvalue decomposition of the covariance matrix $\mathbf{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{E}\mathbf{D}\mathbf{E}^T$ to transform the observed data according to:

$$\mathbf{x}_w = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T \mathbf{x} \quad (3.8)$$

where \mathbf{E} is the orthogonal matrix of eigenvectors of the covariance matrix of the data, \mathbf{D} is the diagonal matrix of associated eigenvalues $\mathbf{D} = \text{diag}(d_1, \dots, d_m)$ and $\mathbf{D}^{1/2} = \text{diag}(d_1^{-1/2}, \dots, d_m^{-1/2})$ [166].

Dimensionality reduction. When sphering, we can at the same time reduce the dimensionality of the data by discarding those eigenvalues of the covariance matrix that are too small as is done in PCA. Reducing the dimensions of the data can help in suppressing noise and preventing over learning of the ICA algorithm. After sphering we can determine the demixing matrix \mathbf{B} where

$$\mathbf{E} \mathbf{v} \mathbf{v}^T = \mathbf{B} \mathbf{E} \mathbf{s} \mathbf{s}^T \quad \mathbf{B}^T = \mathbf{B} \mathbf{B}^T = \mathbf{I} \quad (3.9)$$

Once \mathbf{B} is found Eq (3.8) is used to find the ICs from the observed \mathbf{v} by

$$\hat{\mathbf{s}} = \mathbf{B}^T \mathbf{v} \quad (3.10)$$

3.1.3.3 Optimization function

Once the objective function is chosen there is the need for a practical method for its implementation i.e. a method, such as maximum likelihood, that is used to maximize or minimize F . This function determines the algorithmic properties (i.e. numerical stability of the ICA, convergence speed and memory requirement) [59, 62].

3.1.4 Assumptions for the ICA model

The following assumptions ensure that the ICA model estimates the ICs meaningfully. Actually the first assumption is the only true requirement which ICA demands. The other assumptions ensure that the estimated ICs are unique.

- (1) The ICs are statistically independent and the mixing is linear.
- (2) There is no more than one gaussian signal among the ICs. In theory multiple Gaussian processes cannot be separated by ICAs [71]. The ICs also have cumulative density function not much different from a *logistic sigmoid*.

- (3) The number of observed signals, m , is greater than or equal to the number of ICs, n (i.e. $m \geq n$). If $n > m$, we come to a special category of ICA called *ICA with over-complete bases*. In such a case the mixed signals do not have enough information to separate the ICs. There have been attempts to solve this particular problem but no rigorous proofs exist as of yet. If $m > n$ then there is redundancy in the mixed signals. The ICA model works ideally when $n = m$ [127].
- (4) The mixing matrix is of full column rank, which means that the rows of the mixing matrix are linearly independent. If the mixing matrix is not of full rank then the mixed signals will be linear multiples of one another.
- (5) The propagation delay of the mixing medium is negligible [127].
- (6) Spatial projections of the components are fixed across time and conditions.

3.1.5 The ICA model applied to EEG Data

Assumptions

In the case of EEG signals we have m -scalp electrodes picking up correlated brain signals where we would like to know what effectively independent brain sources produced these signals. The ICA model appears well suited for this scenario because it satisfies most of the model assumptions considered above. Start with assuming that EEG data can be modeled as a collection of statistically independent brain signals. Assumption (5) is valid since volume conduction in the brain is effectively instantaneous and assumption (2) is plausible. In this research, we will attempt to separate the m -observed EEG signals into n -statistically ICs (thus satisfying assumption (3) and (4)). However, it is questionable to assume whether EEG data recorded from m -electrodes is made up of *exactly* n statistically ICs since it ultimately cannot know the exact number of ICs embedded in the EEG data [143]. Nonetheless, this assumption is usually enough to identify and separate artifacts that are concentrated in certain areas of the brain such as eye, temporal, and occipital artifacts. The ICA model tends to have a more difficult time in separating artifacts that are more spaced out over the scalp such as muscle artifacts. Assumption (6) is

met because as nonstationary EEG produces signals that tend to be transient (localized in time), restricted to certain ranges of temporal and spatial frequencies (localized in scale) and prominent over certain scalp regions (localized in space) [146].

Mathematically

Mathematically ICA and EEG are identical and can be written as:

$$E(t) = As(t) + n(t) \tag{3.11}$$

Where $As(t)$ is the matrix representation of the pure EEG signal in Eq (2.1) and x in Eq (3.2) is the mixed recorded signal $E(t)$ in Eq (3.11). Determining W therefore will generate $S(t)$ as seen in Figure 3.3.

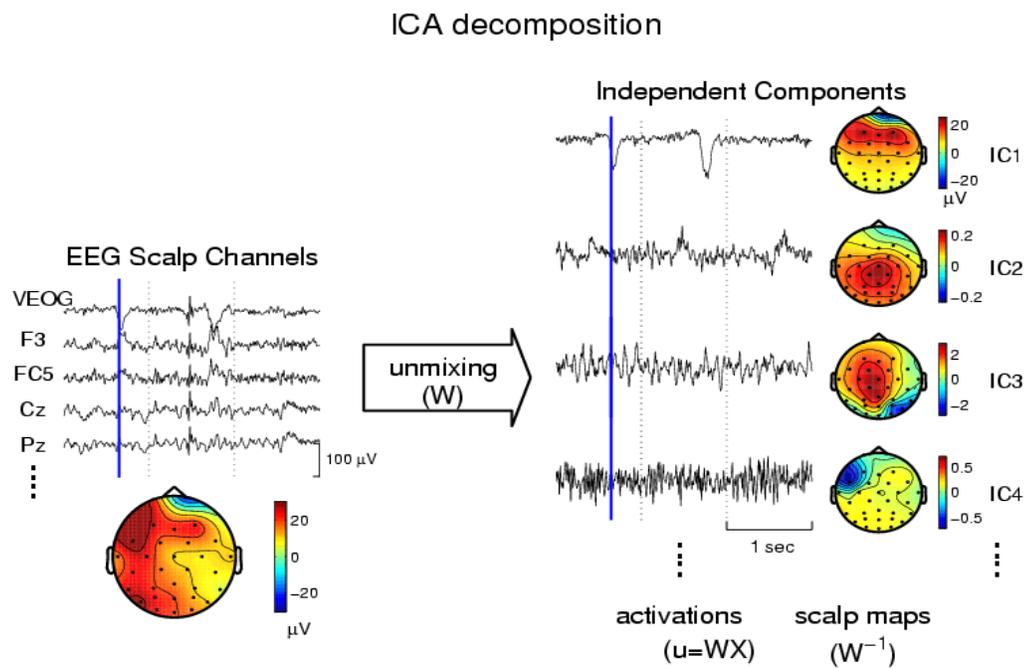


Figure 3.3: EEG Signals Beign Broken into ICs using ICA (adapted from Scott Makeig, *ICA Toolbox Tutorial – Removing Artifacts from Singale Trials*, <http://sccn.ucsd.edu/~scott/tutorial/icatutorials8.html#NIPS97>)

3.1.6 ICA Performance at Denoising EEG Signals

Different types of ICA algorithms have been proposed in the last 10 to 12 years, most of which assume that the sources are stationary and are based explicitly or implicitly on high order statistics computation. Therefore, Gaussian sources cannot be separated, as they don't have higher than 2 statistic moments. Other types of algorithms do not make the stationarity hypothesis, and use the non stationary structure of the signals (i.e. their time or frequency structure) to separate them. These methods use the second order statistics (SOS) only, and are called SOS algorithms. As EEG signals are highly non-stationary, these type of algorithms are the most widely used. Krishnaveni *et al.* [92] used Joint Approximate Diagonalization for Eigen-matrices (JADE) and Neural Network to successfully remove EOG artifacts from EEG signals. Hoffmann *et al.* [57] compared two ICA methods - both are able to remove artifacts although at different levels.

Keith *et al.* [80] investigated parallel ICA methods applied to EEG and found that there were superposition of underlying signals from various sources within the brain and extra-brain. Nicolaou *et al.* [120] supported this finding when they compared standard and temporal ICA, stating that these signals contaminated the EEG signals. Sadovský *et al.* [138-139] found that these artifacts have bigger amplitude when they overlay EEG signals.

Keith *et al.* [80] investigated the analysis of EEG signals using FastICA and Infomax. They found that both are very processor intensive especially with large data sets. They went on to say that FastICA failed to converge on a solution especially for large number of channels. Ziehe *et al.* [185] in their investigation found that JADE performed well for small blind separation problems while Temporal Decorrelation source SEPARation (TDSEP) performed better in large numbers of sources. Glass *et al.* [43] stated that when removing artifacts from multichannel EEG both Infomax and FastICA were highly accurate in separating [91]. They also investigated artifacts removal using MI estimator to investigate the performance of six (6) algorithms. They found that the Robust, Accurate, Direct Independent Component Analysis aLgorithm (RADICAL) performed the best at

separating the original sources from the observed signals. They also found that JADE outperformed four (4) other algorithms including Fast fixed-point Independent Component Analysis (FastICA), Information Maximization (Infomax) and TDSEP. They stated that JADE was considered best for small problems while RADICAL was best for large problems.

Researchers have shown that ICA can be used to denoise EEG signals. The algorithms however perform better in different circumstances such as data size, processor speed, and the number of channels. The question is why and can there be an algorithm that performs in all circumstances?

Comparison with other Denoising Techniques

ICA has several advantages when compared with other artifact removal methods:

1. The algorithm is computationally efficient and the computational requirements are not excessive even for fairly large EEG data sets.
2. ICA is generally applicable for removal of a wide variety of EEG artifacts. It simultaneously separates both the EEG and its artifacts into ICs based on the statistics of the data, without relying on the availability of one or more “clean” reference channels for each type of artifacts. This avoids the problem of mutual contamination between regressing and regressed channels.
3. Unlike regression-based methods, no arbitrary thresholds (usually variable across sessions) are needed to determine when artifact correction should be performed.
4. Separate analyses are not required to remove different classes of artifacts. Once the training is complete, artifact-free EEG records in all channels can then be derived by simultaneously eliminating the contributions of various identified artifactual sources in the EEG record.
5. The ICA artifact subtraction method preserves and recovers more brain activities than regression and PCA.

6. The same ICA approach should be equally applicable to other types of multichannel biomedical data for which linear summation can be assumed (e.g., MEG, EOG, ECG/EKG, EMG, etc.). In addition to artifact removal, ICA decomposition can be highly useful for observing changes in the spatial structure of ongoing or averaged EEG activity in multiple brain areas, networks, or neural populations [72-73, 104-106].

3.1.7 Algorithms used in Research

ICA techniques have been dominating research in the field. The algorithmic approaches have emphasized improving the speed and separation performance of the method by finding new and clever ways of estimating independence. All these algorithms seek to maximize or minimize some potential energy function related to the entropy or information contents of the signals. These methods led to categoring ICA algorithms in four (4) approaches namely:

- (i) maximizing Mutual Information (MI)
- (ii) maximizing non- gaussianity
- (iii) using second order correlation
- (iv) using joint diagonalization

In this dissertation we utilize at least one algorithm from each approach for experimental comparisons. These are algorithms known to be utilized in biosignal denoising. They are: FastICA [60], Pearson ICA [79], Infomax [8] (maximize MI), CubICA [12], TDSEP [181], JADE [15] (joint diagonalization), SOBI [9] (second order correlation) and EFICA [93] (maximize non-gaussianity) discussed below.

The original Matlab codes provided by the authors were used, whenever possible, and a brief description of each is given below. Some of these algorithms are not designed to estimate a rectangular $m \times n$ unmixing matrix W so in order to reduce computational complexity preprocessing is done by means of PCA as described in Chapter 4.

3.1.7.1 FastICA

Hyvarinen developed a fixed-point ICA algorithm in 1997 called the fast fixed-point Independent Component Analysis (FastICA). According to the central limit theorem, sum of two independent random variables usually has a distribution that is closer to gaussian than of the two original random variables, thus, maximizing the non-gaussianity yields ICs. The non-gaussianity is measured with the differential entropy J , called *negentropy* [23] which is defined as the difference between the entropy of a Gaussian random variable y_{gauss} and the entropy of y :

$$J(y) = H(y_{\text{gauss}}) - H(y) \quad (3.12)$$

where the entropy H is given by

$$H(y) = - \int f(y) \log(f(y)) dy \quad (3.13)$$

Since Gaussian random variables have the largest entropy H among all random variables having equal variance, maximizing $J(y)$ leads to the separation of independent source signal.

Originally FastICA was presented in a deflation mode where the IC's were extracted recursively i.e. one after another [174]. Presently there is also the symmetric mode where the ICs are extracted simultaneously. FastICA uses simple estimates of negentropy based on the maximum entropy principle which maximizes a “non-gaussianity” measure. This measure requires the use of appropriate nonlinearities for the learning rule of the neural network and the standard nonlinearities implemented are *pow3*, *tanh*, *gaus*, and *skew*. Of the four, *tanh* has been described as a “good general purpose contrast function” [58] and one which is better at producing a more robust algorithm [124]. In this dissertation we utilize *tanh*.

3.1.7.2 Pearson ICA

The Pearson-ICA algorithm is a MI-based method for blind separation of statistically independent source signals based on the Pearson system defined by the differential equation

$$f'(x) = \frac{(x-a)f(x)}{b_0 + b_1x + b_2x^2} \quad (3.14)$$

where a , b_0 , b_1 , and b_2 are parameters of the distribution. In this method the underlying source distributions are estimated through the marginal distributions by fitting them to the Pearson system. Fitting to the Pearson system is done iteratively until the optimization algorithm converges [79]. The algorithm combines two (2) well known techniques: maximum likelihood and fixed non-linear contrast functions. In order to extract the independent component sources for a demixing matrix W that minimizes the mutual information of the sources W_{k+1} is calculated as

$$W_{k+1} = W_k + D E \varphi(y)y^T - \text{diag} E \varphi(y_i)y_i \quad W_k \quad (3.15)$$

where $D = \text{diag}(1/E\{\varphi(y_i)y_i\} - E\{\varphi(y_i)\})$.

3.1.7.3 Infomax

Bell and Sejnowski [8] have proposed an adaptive learning algorithm that blindly separates mixtures, $X(t)$ of independent sources, $S(t)$ using information maximization (infomax). The information maximization is attained by maximizing the joint entropy of a transformed vector $z=g(Wx)$, where g is a point wise sigmoidal nonlinear function. This function provides all higher order statistics necessary to establish independence. The point entropy $H(y)$ can be written as:

$$H(y) = -E[\ln f(y)] \quad (3.16)$$

where $f(y)$ is the multivariate joint density function of y . The relation between $f(y)$ and $f(x)$ is expressed by

$$f(y) = \frac{f(x)}{|J_w|} \quad (3.17)$$

where $|J_w|$ denotes the absolute value of the Jacobian matrix

$$J_w = \det \left[\frac{\partial y_i}{\partial x_j} \right]_{ij} \quad (3.18)$$

Consequently, $H(y)$ can also be written as

$$H(y) = -E[\ln f(y)] \quad (3.19)$$

and maximization of $H(y)$ can be achieved by adapting W to maximize only the first term $E[\ln|J_w|]$.

3.1.7.4 CubICA

The Cumulant-based Independent Component Analysis (CubICA) is a method that uses the diagonalization of cumulant tensors of higher order to search for independent sources. It is based on Comon's algorithm but it takes third $C_{\alpha\beta\gamma}^{(y)}$ and fourth-order $C_{\alpha\beta\gamma\delta}^{(y)}$ cumulant tensors into account simultaneously. It is based on the third-order and fourth-order cumulant tensors [12]. A proper contrast function Ψ_{34} is used to achieve an approximation simultaneous diagonalization of the two tensors:

$$\Psi_{34}(y) = \frac{1}{3!} \sum_{\alpha\beta\gamma \neq \alpha\alpha\alpha} (C_{\alpha\beta\gamma}^{(y)})^2 + \frac{1}{4!} \sum_{\alpha\beta\gamma\delta \neq \alpha\alpha\alpha\alpha} (C_{\alpha\beta\gamma\delta}^{(y)})^2 \quad (3.20)$$

CubICA estimates the ICs and the unmixing matrix W by maximizing Ψ_{34} .

3.1.7.5 TDSEP

Temporal Decorrelation source SEPARation (TDSEP) uses the time structure of the sources to separate the components. It performs a simultaneous approximate diagonalization of several time-delayed correlation matrices using time delays $\tau=1,2,3,\dots$, which are based on the expectation on the temporal structure of the signals. The cross-covariance function of the signals is obtained from

$$E x(t)x^T(t, \tau) = C_x^\tau = AC_x^\tau A^T \quad (3.21)$$

where the source cross-covariance functions C_s^τ are a set of diagonal matrices due to the statistical independence of the sources. In order to estimate an unmixing matrix W , TDSEP uses rotation of the mixtures and whitening [105], and may require that a set of time delays be arbitrarily selected or manually tuned.

3.1.7.6 JADE

The Joint Approximate Diagonalization for Eigen-matrices (JADE) is an algorithm based on the joint diagonalization of cumulant matrices under the assumption that the sources have non-Gaussian distributions. It involves transformations of the cumulant tensors of the second and the fourth orders. The independence of the sources is obtained through the reduction of zero of the second-order cumulant $C_{\alpha\beta}^{(y)}$, which is the condition for the uncorrelation of the source estimates, and the reduction to zero of the fourth-order cumulant $C_{\alpha\beta\gamma\delta}^{(y)}$, which implies that the sources are also statistically independent. A fast optimization of the algorithm is obtained with the simultaneous diagonalization of two cumulants, based on Comon [23], using the contrast function Ψ_{24}

$$\Psi_{24}(y) = \frac{1}{4} \sum_{\alpha\beta \neq \alpha\alpha} (C_{\alpha\beta}^{(y)})^2 + \frac{1}{48} \sum_{\alpha\beta\gamma\delta \neq \alpha\alpha\alpha\alpha} (C_{\alpha\beta\gamma\delta}^{(y)})^2 \quad (3.22)$$

In general, JADE estimates very rapidly the unmixing matrix W , which may be square or rectangular. JADE works in batch mode, has no problem updating weights and has no need for manual parameter tuning.

3.1.7.7 SOBI

The Second Order Blind Identification (SOBI) algorithm exploits joint diagonalization of time delayed second order correlation matrices. In this algorithm the cross-covariance matrix $R_s(\tau)$ is assumed to be diagonal for each time shift τ . The decomposition problem here is to find the matrix A that simultaneously diagonalizes all the matrices $R_s(\tau)$ for any τ , or more formally, to find A such that

$$R_x(\tau) = A^T R_s(\tau) A \quad (3.23)$$

and $R_s(\tau)$ is diagonal for any τ . SOBI determines the unmixing matrix W by approximating the set of covariance matrices.

3.1.7.8 EFICA

Efficient FastICA (EFICA) is an improved version of the FastICA algorithm which combines the idea of the generalized symmetric FastICA with an adaptive choice of the function g . The algorithm A is based on the following observations: (i) the symmetric FastICA algorithm can be run with different nonlinearity for different sources but in this dissertation we use $g(x) = \tanh(x)$ (ii) In the symmetrization step of each iteration, it is possible to introduce an auxiliary constants, that can be tuned to minimize mean square estimation error in one (say k -th) row of the estimated de-mixing matrix [93]. These estimations can be performed in parallel for all rows - to obtain an estimate of the whole de-mixing matrix that achieves the corresponding Cramér-Rao lower bound (CRB), if the nonlinearities correspond to score functions of the sources. (iii) The algorithm remains to be asymptotically efficient (attaining the CRB) if the theoretically optimum auxiliary constants in the algorithm are replaced by their consistent estimates. In this algorithm CRB is defined as:

$$CRB(G_{kl}) = \frac{1}{N} \frac{K_k}{K_k K_l - 1} \quad (3.24)$$

where $K_k = E[\psi_k^2(s_k)]$

3.1.7.9 RADICAL

Miller and Fisher [113] proposed the Robust, Accurate, Direct Independent Component Analysis aLgorithm (RADICAL) algorithm based on the neighborhood density estimator. The algorithm uses the well-known minimum maginal entropy constrast function. Denoting y_i the i th estimated independent component this contrast function is given by:

$$\phi^{ME}(y) = \sum_{i=1}^n H(y_i) \approx \sum_{i=1}^n \hat{H}_T(y_i) \quad (3.25)$$

where n is the number of estimated independent component and $\hat{H}_T(y_i)$ is the estimation of the entropy of y_i using T samples of y_i . The final entropy estimator can be expressed as:

$$\hat{H}(y) = \frac{1}{T-m} \sum_{i=1}^{T-m} \log \left(\frac{T+1}{m} y^{(i+m)} - y^{(i)} \right) \quad (3.26)$$

where m is the spacing order $y^{(i+m)} - y^i$ for $1 \leq i < i+m \leq T$

3.2 Wavelet Analysis

An EEG signal is a wave which is a periodic oscillating function of time or space. In contrast, a wavelet is a waveform of limited duration which has energy concentrated in time – having a beginning and an end. Each wavelet is irregular, of limited duration, and often non-symmetrical. They are therefore better at describing anomalies, pulses and other events that start and stop within a signal [38]; as a result they provide a versatile mathematical tool to analyse transient, non-stationary or time-varying phenomena that are not statistically predictable. Figure 3.4 shows the difference between a wave and a wavelet.

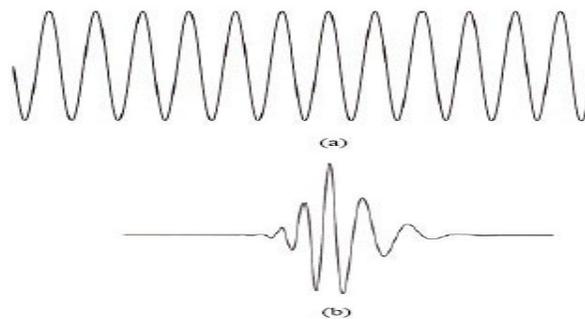


Figure 3.4: Difference between (a) wave and (b) wavelet. Notice that the wave has an easily discernible frequency while the wavelet has a *pseudo frequency* in that the frequency varies slightly over the length of the wavelet. (adapted from D.L. Fugal. 2009. *Conceptual Wavelets in Digital Signal Processing: An in depth Practical Approach for the Non-Mathematician, Space & Signals Technologies LLC*)

There are three categories of wavelets – crude, orthogonal and biorthogonal. In this dissertation we focus on the use of the orthogonal wavelets as they:

- (i) are not symmetrical as biorthogonal so it is easy to see the approximation image or signal at higher levels and
- (ii) do not require conversion to wavelet filters like crude wavelets to be used with digital signals such as biosignals [38].

A set of wavelets are employed to approximate a wave or signal. This wavelet expansion of $s(t)$ is the representation of the wave or signal in terms of an orthogonal collection of real-valued functions generated by applying suitable transformations to the original given wavelet and defined as:

$$s(t) = \sum_j \sum_k a_{j,k} \psi_{j,k}(t) + \sum_k c_{j,k} \varphi_{j,k}(t) \quad (3.27)$$

These functions are called “daughter” wavelets while the original wavelet is dubbed the “mother” wavelet defined in [147] as:

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad (3.28)$$

The variables j and k are integers that scale and dilate the mother function to generate wavelets, such as a Daubechies wavelet family. The scale index j indicates the wavelet's width, and the location index k gives its position. Notice that the mother functions are rescaled, or “dilated” by powers of two, and translated by integers [45]. The collection of coefficients $a_{j,k}$ is based on the subset of j and k called the discrete wavelet transform (DWT) of $s(t)$ and represents the “details”.

The second term in Eq. (3.25) is the “approximation” based on the scaling function:

$$\varphi_{j,k}(t) = 2^{j/2} \varphi\left(\frac{t - 2^j k}{2^j}\right) \quad (3.29)$$

A signal can be analyzed better with an irregular wavelet. These are employed to approximate a signal and each element in the wavelet set is constructed from the mother wavelet, by shifting (translating or delaying) and scaling (dilating or compressing) it.

3.2.1 Wavelet Families

There are a number of basis functions that can be used as the mother wavelet for Wavelet Transformation. Since the mother wavelet produces all wavelet functions used in the transformation through translation and scaling, it determines the characteristics (filter length, vanishing order, filter magnitude response, smoothness and group delay difference) of the resulting Wavelet Transform (WT) [149]. Therefore, the details of the particular application should be taken into account and the appropriate mother wavelet should be chosen in order to use the WT effectively.

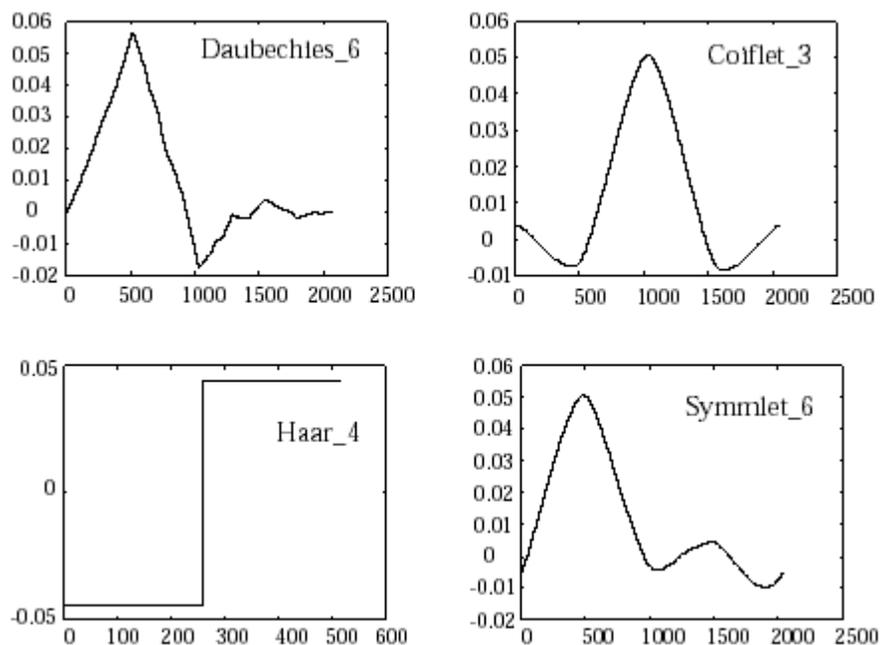


Figure 3.5: Examples of Wavelets Families. The number next to the wavelet name represents the num of vanishing moments for the subclass of wavelet (*adapted A. Graps 1995. An Introduction to Wavelets. IEEE Journal of Computational Science and Engineering 2(2):1-17.*)

Figure 3.5 illustrates some of the commonly used wavelet families. Haar wavelet is one of the oldest and simplest wavelet. Therefore, any discussion of wavelets starts with the Haar wavelet. Daubechies wavelets are the most popular

wavelets. They represent the foundations of wavelet signal processing and are used in numerous applications. These are also called Maxflat wavelets as their frequency responses have maximum flatness at frequencies 0 and R . This is a very desirable property in some applications. The Haar, Daubechies, Symmlets and Coiflets are compactly supported orthogonal wavelets. These wavelets along with Meyer wavelets are capable of perfect reconstruction. The Meyer, Morlet and Mexican Hat wavelets are symmetric in shape.

The different families make trade-offs between how compactly the basis functions are localised in space and how smooth they are [45]. Within each family of wavelets (such as the Daubechies family) are wavelet subclasses distinguished by the number of filter coefficients and the level of iteration. Wavelets are most often classified within a family by the number of *vanishing moments* [45]. This is an extra set of mathematical relationships for the coefficients that must be satisfied and indicates the smoothness of the wavelet function as well as the flatness of the frequency response of the wavelet filters (filters used to compute the DWT) [45]. The extent of compactness of signals depends on the number of vanishing moments of the wavelet function used thus wavelets with a high number of vanishing moments lead to a more compact signal representation and are hence useful in coding applications. With all these the actual wavelets are chosen based on their shape and their ability to analyze the signal in a particular application as the objective of the wavelet is to minimise reconstructed error variance and maximise signal to noise ratio (SNR). For this dissertation I utilize the Symmlet family with vanishing moments of 8 [149]. The Symmlet family is utilized because when these wavelets are applied to signals the said signals perform better and the SNR of the reconstructed or denoised signals is improved [17]. The reason for the choice of vanishing moment is explained in Chapter 8.

3.2.2 Denoising Using Wavelet

The use of wavelets for noise removal was first introduced by Donoho and Johnstone in 1995 [29]. The general procedure involves three steps:

- (i) Decompose – a wavelet is chosen with a level N and the signal is decomposed at N using DWT to give coefficients at different scales having different magnitudes
- (ii) Noise Removal – here for each level 1 to N noise is removed from the detail coefficients using one of two processes:
 - Wavelet transforms maxima where noise is eliminated and maximizes the information of the original signal. The process of calculation is however unstable and the amount of calculation is great
 - Wavelet thresholding proposed by Donoho which was used in this research. When threshold is applied coefficients are categorized. Noise normally produces coefficients with magnitudes smaller than those of the natural signal and according to Donoho and Johnstone [29] basic wavelet denoising is performed by taking the WT of the noise-corrupted $s[t]$ and then zeroing out the detail coefficients that fall below a certain threshold - noise. The other coefficients that are larger are usually caused by the desired signal and kept (hard-thresholding) or shrunk (soft-thresholding) [186].
- (iii) Reconstruct – denoised signals are reconstructed from the wavelet coefficients by an inverse wavelet transform which is applied to the thresholded signal to yield an estimate for the true signal, as:

$$\hat{x}[t] = D s[t] = W^{-1} \Lambda_t W s[t] \quad (3.30)$$

where Λ_t is the diagonal thresholding operator that zeroes out wavelet coefficients less than the threshold, t .

Thesholding

Thresholding is an important concept in denoising and compression, because a few detail coefficients hold the signal information when the wavelet basis selected is well matched to signal characteristics, while the effect of additive white gaussian noise (AWGN) on the signal is the same over all the coefficients at each scale. Note

that the approximation coefficients that do not contain signal energy often do not reside at or near zero, as do their parent detail coefficients. Hence, thresholding schemes will be limited to detail coefficients. In this dissertation we focus on the thresholding methods that are utilized.

The first step in the denoising process is to obtain the WT of the signal $x(n)$ using a suitable basis function. Then, a threshold is obtained using one of the above mentioned thresholding methods. Once an appropriate threshold is determined we must decide how to apply it. This work discusses the hard and soft thresholding techniques as they pertain to a given threshold.

Hard thresholding zeroes out, or shrinks, the coefficients that have magnitudes below the threshold, and leaves the rest of the coefficients unchanged:

$$\hat{d}_j(i) = \begin{cases} d_j(i) & |d_j(i)| > T \\ 0 & |d_j(i)| \leq T \end{cases} \quad (3.31)$$

Soft thresholding extends hard thresholding by shrinking the magnitude of the remaining coefficients by threshold T , producing a smooth rather than abrupt transition to zero

$$\hat{d}_j(i) = \begin{cases} \text{sign}[d_j(i)](|d_j(i)| - T) & |d_j(i)| > T \\ 0 & |d_j(i)| \leq T \end{cases} \quad (3.32)$$

The smooth transition to zero results in noticeably fewer artifacts upon reconstruction, especially when dealing with image denoising. Hence, soft thresholding is generally better for denoising [133] due to its inherent smoothing, whereas hard thresholding is better suited for data compression. For comparison purposes Figure 3.6 illustrates the two threshold methods against the original signal. In either case, perfect reconstruction is not possible since some of the signal components are thrown away with the undesired noise. Another drawback to thresholding is that noise affecting the remaining signal coefficients is not removed.

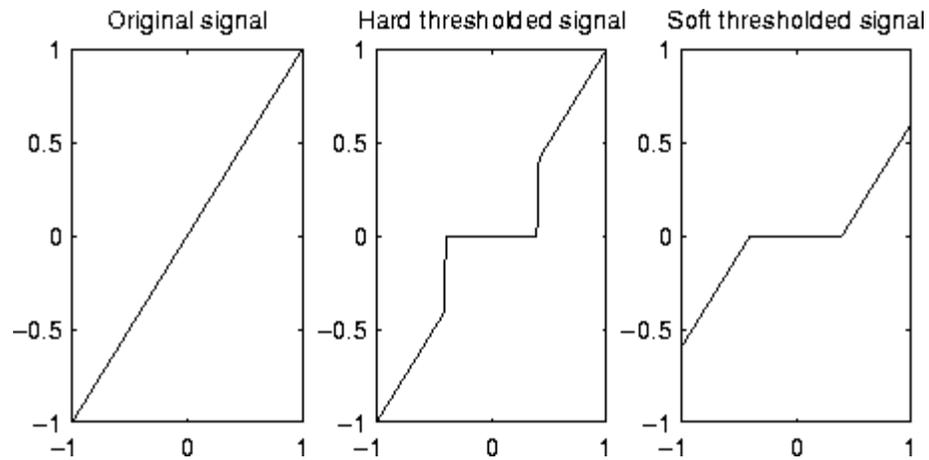


Figure 3.6: Hard and Soft Thresholding Estimators along with the Original Signal (*adapted from Denoising Signals and Images, Wavelet Toolbox, MathWorks, <http://www.mathworks.com/products/wavelet/demos.html?file=/products/demos/shipping/wavelet/denoising/signalsdemo.htm>*)

Although simple and easy to use, research has shown that each thresholding method exhibits problems:

- (i) Hard thresholding leads to the oscillation of the reconstructed signal [155] and
- (ii) Soft thresholding reduces the amplitude of the signal waveform [155, 183].

Furthermore, traditional thresholding de-noising methods based on DWT may cause Pseudo-Gibbs phenomena in the reconstructed signal. The next section introduces an important concept as an extension of the orthogonal WT denoising which improves these limitations.

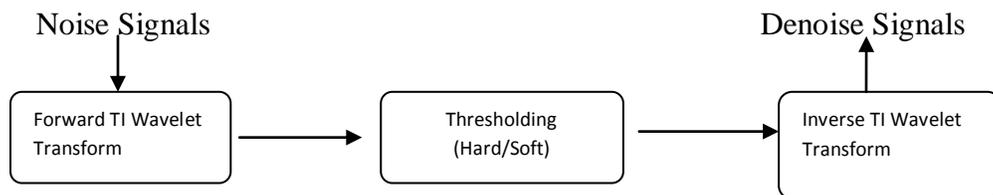


Figure 3.7: Block Diagram of the TI Wavelet Transform

3.2.3 Translation-Invariant

Note that orthogonal WT transform is not translation-invariant, i.e., the wavelet coefficients change when the signal is transformed using different time shifts, thus the application of DWT may result in a blur of the signal energy over several transform details of smaller amplitude which may be masked in the noise. The result – the detail been subsequently truncated when it falls below the threshold. These truncations can result in overshooting and undershooting around discontinuities similar to the Gibbs phenomena in the reconstructed denoised signal. [6, 21]. Coifman and Donoho proposed a solution by designing a denoising scheme to suppress such truncations and artifacts by averaging over the denoised signals of all circular shifts. They achieved by designing a cycle spinning denoising algorithm which:

- (i) shifts the signal by collection of shifts, within the range of cycle spinning
- (ii) denoise each shifted signal using a threshold (hard or soft)
- (iii) inverse-shift the denoised signal to get a signal in the same phase as the noisy signal
- (iv) Averaging the estimates.

The Gibbs artifacts of different shifts partially cancel each other, and the final estimate exhibits significantly weaker artifacts [21]. This method, described in Figure 3.7, is called the translation-invariant (TI) denoising scheme.

Experimental results in [2] show that single TI wavelet denoising performs better than the traditional single wavelet denoising. Research has also shown that TI produces smaller approximation error when approximating a smooth function as well as mitigating Gibbs artifacts when approximating a discontinuities function.

Cycle Spinning

WT is not time invariant, consequently, if the noisy signal is shifted in time, denoised, and then shifted back, the result will, in general, be different from the estimate obtained from denoising without shifting. Cycle Spinning (CS) was proposed by Coifman & Donoho [21] as a simple yet efficient method that utilizes

the periodic time-invariance of WT in fixing the noise found in wavelet coefficients. CS calls for the suppression of these noises by shifting the signals in time and computing the estimate. Using different shifts produce different estimates which are not completely independent. Consequently linearly averaging these estimates results in a reduction in the noise generated in each shift as the errors in the estimates are not completely dependent [34]. This results in the denoising of all possible unique circularly shifted version of the signal. Mathematically CS is defined in [21] as:

$$\hat{s} = \frac{1}{k_1 k_2} \sum_{i=1, j=1}^{k_1 k_2} S_{-i, -j} T^{-1} \theta \left[T S_{i, j}(x) \right] \quad (3.33)$$

where (k_1, k_2) are maximum number of shifts, T the shift variant transform, $S_{i, j}$ is the circulant shift, and θ the threshold operator.

3.2.4 WT and EEG

Since wavelet analysis uses bases that are localized in time as well as frequency it can represent non-stationary signals such as EEG more effectively; so, it's more compact and easier to implement. WT utilizes the distinguishing features of the noise, however. Once wavelet coefficient are created, noise can be identified. Mallat in [2] stated that as scales increase the WT of EEG and noise present different inclination. Noise concentrates on scale 2^1 , decreasing significantly when the scale increases, while EEG concentrates on the 2^2 - 2^5 scales. Elimination of the smaller scales denoises the EEG which is then reconstructed by the other scales and the useful EEG signals are reserved while the noise are removed efficiently [10]. Using the dyadic WT, EEG signals can be successfully decomposed into the alpha, beta, theta, and delta waves as shown in Figure 3.8. WT removes any overlapping of noise and EEG signals that ICA cannot filter out.

Denoising is applied only on the detail coefficients of the WT and it has been shown that this algorithm offers the advantages of smoothness and adaptation.

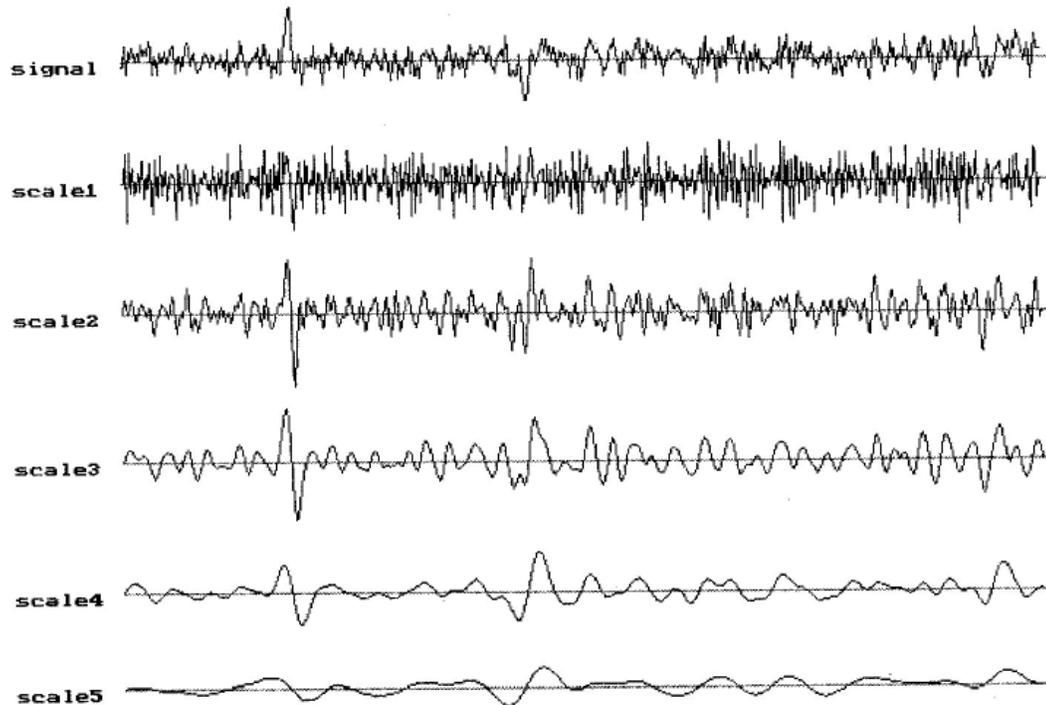


Figure 3.8: Noisy EEG and its Wavelet Transform at different scales. Scale1 to scale5: dyadic wavelet transform of EEG signal from scale 2^1 to 2^5 (adapted from Weidong Z., Yingyuan, L. 2001. *EEG Multi-resolution Analysis using Wavelet Transform*, 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE/EMBS) 2001)

3.3 Filtering

Linear filtering has been found to be very effective at denoising signals and images. There are many types of filter; however, the most used is the Weiner Filter (WF). Li *et al.* [97] found that when data noise is converted into Gaussian noise with a nearly constant variance of 0.25 it enables WF to effectively treat the noise. They also found that although WF has achieved a noticeable success, its strong dependence on the stationary random process (variance of noise constant) limits further improvement.

Unlike WF the Kalman Filter (KF), an important generalization of WF, has the ability to adapt itself to non-stationary environments. Kovalchuk *et al.* [88]

found that the accuracy of WF is one of tenth that of KF. In a series of experiments, Wu *et al.* [178-180] made a comparative study of the relative performance of the two filters, and reported a superior decoding performance by KF.

The basic KF is limited to a linear assumption; however biosignals, such as EEG, are non-linear. The most widely used filter for these systems is the Extended Kalman Filter (EKF). This filter applies KF to nonlinear systems by simply linearising all nonlinear models (i.e., process and measurement models) so that the KF equations can be applied [66]. Unfortunately, it has two major flaws:

- (i) Derivation of Jacobian matrices often complex and nontrivial - implementation difficulties
- (ii) Linearization approximations can introduce large errors - instabilities [66]

In order to address these flaws the Unscented Kalman Filter (UKF) was developed [31, 172]. Instead of linearizing using Jacobian matrices, UKF uses a deterministic sampling approach to capture the mean and covariance estimates with a minimal set of sample points, thus having superior implementation properties to EKF [66]. Wan *et al.* [172] found that UKF represents a derivative-free alternative to EKF and provides a superior performance at an equivalent computational complexity.

3.3.1 Extended Kalman Filter

Extended Kalman Filter was created as an extension of the KF to solve problems which are non-linear. This algorithm deals with nonlinear data by linearizing about the current mean and covariance so that KF can be applied [67]. This produces an estimated state instead of the real state and Gaussian pdfs instead of the non-Gaussian in the real pdf. These produce flaws in the algorithm and do not guarantee convergence. The nonlinear stochastic system used for the algorithm based on [172] is:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + v_k \\ y_k &= Hx_k + w_k \end{aligned} \tag{3.34}$$

where A and H are the known and constant matrices respectively, x_k is the unobserved state of the system, u_k is a known exogenous input, y_k is the observed measurement signal, v_k is the process noise and w_k is the measurement noise.

3.3.2 Unscented Kalman Filter

Unscented Kalman Filter (UKF) is a Bayesian filter which uses minimum mean-squared error (MMSE) as the criterion to measure optimality [7, 154]. For highly nonlinear systems, the linear estimate of the nonlinear model does not provide a good approximation of the model, and the Extended Kalman Filter (EKF) will not track signals around sharp turning points [141] thus performs poorly in highly nonlinear system. Research has shown also that if the initial estimate of the state is wrong, or the process is modeled incorrectly, EKF may diverge quickly, due to its linearization. Lastly, it was found that with EKF the estimated covariance matrix tends to underestimate the true covariance matrix and therefore risks becoming inconsistent in the statistical sense without the addition of "stabilising noise". UKF was found to address these flaws.

UKF uses the same nonlinear stochastic system used for EKF in Eq (3.34) and involves the Unscented Transformation (UT), a method used to calculate the first and second order statistics of the outputs of nonlinear systems with Gaussian.

UKF uses the intuition that it is easier to approximate a probability distribution function rather than to approximate an arbitrary nonlinear function or transformation [4]. Following this intuition, a set of sample points, called sigma points, are generated around the mean, which are then propagated through the nonlinear map to get a more accurate estimation of the mean and covariance of the mapping results. In this way, it avoids the need to calculate the Jacobian, which for complex functions can be a difficult task in itself (i.e., requiring complicated derivatives if done analytically or being computationally costly if done numerically). UKF therefore replaces linearization with sigma points.

3.3.3 Filter Configuration

The filters described above can be used for state estimation, parameter estimation, or for the simultaneous estimation of the state and the parameters. These require different filter configurations, which are summarized below based on [41].

3.3.3.1 State tracking

The model parameters are assumed to be known. The goal of state tracking is to determine the pdf $p(x_k/y_{1:k})$ for every k . The state configuration is the actual stochastic system in Eq (3.34).

3.3.3.2 Parameter tracking

The model states and measurements are assumed to be known. The state-space model is formed for the evolution of the model parameters $x_{par;k} = w_k$, which is often assumed to be a random walk with noise $v_{par;k}$: The measurement is written as a function of the system state x_k and the model parameters $x_{par;k}$; and a “state tracking” filter is run for $x_{par;k}$:

$$\begin{aligned}x_{par,k} &= x_{par,k-1} + v_{par,k-1} \\y_k &= g'(x_k, x_{par,k}, n_k)\end{aligned}\tag{3.35}$$

3.3.3.3 Joint estimation

In joint estimation both the system state and the model parameters are estimated simultaneously. To this end an augmented state vector is defined consisting of both the system state and the model parameters, $x_{aug,k} = [x_k^T, x_{par,k}^T]^T$. Based on Eqs (3.34)- (3.35) a new state-space system is formed on which the filter is run.

3.3.3.4 Dual estimation

Similarly to joint estimation, in dual estimation the system state and the model parameters are estimated simultaneously. However, here the state system in

Eq (3.34) and the parameter system Eq (3.35) are kept separately, and two filters are run, one for the state estimation, one for the parameter estimation. For each sample step k the result of the state estimation of the previous sample step x_{k-1} is used as an input for the parameter estimator, and vice versa, the result of the parameter estimator of the previous sample step $x_{par;k-1}$ is used in the state estimator.

3.4 Performance Measures for Methods

The analysis of the denoising algorithms performance consists of estimating (1) the accuracy with which each algorithm is able to separate components, and (2) the speed with which each algorithm is able to reproduce EEG signals [107]. For (1) experiments are mainly aimed at assessing the algorithms' ability to perform ICA (extraction of ICs) and not blind source separation (recovery of original sources). The performance measures that will be used throughout this dissertation are based on two categories of calculation:

- (i) Separation Accuracy Measures - Amari Performance Index, Signal to Interference Ratio (SIR), and Signal to Distortion Ratio (SDR),
- (ii) Noise/Signal Measures - Signal to Noise Ratio (SNR), Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), and Percentage Root Mean Square Difference (PRD)

For (2) we compare the computational complexity of the denoising methods.

3.4.1 Separation Accuracy Measures

3.4.1.1 Amari Index

The most widely studied BSS situation is the non degenerate case where there are at least as many mixtures as there are sources (i.e., $n \geq m$). When that is the case the accuracy of a BSS algorithm can be assessed from its ability to estimate the

mixing matrix. The most widely used measure for assessing the accuracy of the estimated mixing matrix is the Amari performance index P_{err} [3]:

$$P_{err} = \frac{1}{2m} \sum_{i,j=1}^m \left(\frac{|p_{ij}|}{\max_k |p_{ik}|} + \frac{|p_{ij}|}{\max_k |p_{kj}|} \right) - 1 \quad (3.36)$$

where $p_{ij} = (BA)_{ij}$. It assesses the quality of the de-mixing matrix W for separating observations generated by the mixing matrix A . When the separation is perfect, the Amari index is equal to zero. In the worst case, i.e. when the estimated sources contain the same proportion of each original source signal, the Amari index is equal to $m/2-1$. This is most likely the case when we try to separate the underlying brain sources from EEG recordings using a reduced set of electrodes.

3.4.1.2 Signal to Interference Ratio

The most common situation in many applications is the degenerate BSS problem, i.e. $n < m$. This is most likely the case when we try to separate the underlying brain sources from electroencephalographic (EEG) or magnetoencephalographic (MEG) recordings using a reduced set of electrodes. In degenerate demixing, the accuracy of a BSS algorithm cannot be described using only the estimated mixing matrix. In this case it becomes of particular importance to measure how well BSS algorithms estimate the sources with adequate criteria. The most commonly used index to assess the quality of the estimated sources is the Signal to Interference Ratio (SIR) [46]:

$$SIR(dB) = \frac{1}{n} \sum_{i=1}^n \left(\sum_j \frac{|p_{ij}|}{\max_k |p_{ij}|} - 1 \right) \quad (3.37)$$

SIR takes into account the fact that, in general, BSS is able to recover the sources only up to (a permutation and) a gain factor α . It is easy to check that if $s_i = \alpha s_i$ the SIR is infinite. By contrary, when the estimated source is orthogonal to the true source, the SIR is equal to zero.

3.4.1.3 Signal to Distortion Ratio

While SIR assesses the quality of the estimated sources, and the Amari Index assess the accuracy of the estimated mixing matrix, the accuracy of the separation of an ICA algorithm in terms of the signals (i.e. the overall separation performance) is calculated by the total Signal to Distortion Ratio (SDR) defined in [44] as:

$$SDR(x_i, y_i) = \frac{\sum_{n=1}^L x_i(n)^2}{\sum_{n=1}^L y_i(n) - x_i(n)^2} \quad i = 1, \dots, m, \quad (3.38)$$

where $x_i(n)$ is the original source signal and $y_i(n)$ is the reconstructed signal. The SDR is expressed in decibels (dB). The higher the SDR value, the better the separation of the signal from the noise. When the SDR is calculated if it is found to be below 8-10dB the algorithm is considered to have failed separation.

3.4.2 Noise/Signal Measures

3.4.2.1 Signal to Noise Ratio

A good measure of detectability of a noise is its signal-to-noise ratio (SNR) defined as [129]:

$$SNR(dB) = 20 \log_{10} \frac{\sum_{n=0}^N s^2(n)}{\sum_{n=0}^N x^2(n)} \quad (3.39)$$

The SNR provides a comparison of the amount of signal with the amount of background noise in a particular signal, such that at sufficiently high SNR's, the large coefficients of the signals are only slightly distorted by the noise coefficients, and the estimation of the unmixing matrix is almost not affected by the presence of noise. For performance, the greater the ratio, evidenced by a larger number, the less noise and the more easily it can be filtered out. Biosignals such as EEG commonly has below 0dB SNR therefore the highest SNR would be 0dB

3.4.2.2 Peak Signal to Noise Ratio

Peak Signal-to-Noise Ratio, often abbreviated as PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale and defined as [110]:

$$PSNR = 10 \times \log_{10} \left(\frac{MAX^2}{MSE} \right). \quad (3.40)$$

In this research MAX takes the value of 255. Unlike MSE which represents the cumulative squared error between the denoised and mixed signal, PSNR represents a measure of the peak error i.e. when the two signals are identical the MSE will be equal to zero, resulting in an infinite PSNR. The higher the PSNR therefore, the better the quality of the reconstructed signal i.e. a higher PSNR indicates that the reconstruction is of a higher quality and therefore the algorithm is considered good.

3.4.2.3 Mean Square Error

The Mean Square Error (MSE) measures the average of the square of the “error” which is the amount by which the estimator differs from the quantity to be estimated. Mathematically it is defined as in [110] as:

$$MSE = \frac{1}{N} \sum_{y=1}^N [I(x, y) - I'(x, y)]^2. \quad (3.41)$$

The difference occurs because of the randomness or because the estimator doesn't account for information that could produce a more accurate estimate. MSE thus assesses the quality of an estimator in terms of its variation and unbiasedness. Note that the MSE is not equivalent to the expected value of the absolute error. Since MSE is an expectation, it is a scalar, and not a random variable. It may be a function of the unknown parameter θ , but it does not depend on any random quantities. However, when MSE is computed for a particular estimator of θ the true value of

which is not known, it will be subject to estimation error. In a Bayesian sense, this means that there are cases in which it may be treated as a random variable.

3.4.2.4 Percentage Root Mean Square Difference

Percentage Root Mean Square Difference (PRD) measures the square difference average between the original and reconstructed signals i.e. it measures the level of the distortion between the original signal and the reconstructed signal as is defined according to [129] as:

$$PRD = \sqrt{\frac{\sum_{n=0}^N (s(n) - x(n))^2}{\sum_{n=0}^N s^2(n)}} \times 100\% \quad (3.42)$$

The method determines the deformation percent in the denoised signal. Since the variability of the signal around its baseline is what should be preserved and not the baseline itself, the performance measure used to reveal the accuracy of the algorithm was the variance of the error with respect to the variance of the signal.

3.5 Summary

Biosignals are very important in the fields of research and medicine to enhance understanding and the development of cures to ailments in the human body. Pure signals are therefore a necessity. In this chapter I discussed three family of methods used to produce signals that are as closer to pure as possible – ICA, WT and Filtering. The chapter also looked at different algorithms in each method on which will be utilized in this dissertation as well as the linkage between them and the biosignal of choice – EEG. To determine each algorithm’s performance on denoising EEG the chapter discussed the mathematical measures which will be utilized to evaluate them.

Chapter 4 – B-Spline Mutual Information Independent Component Analysis

4.1 The Mutual Information Estimator

We can differentiate between two types of objective functions depending on how the ICs are estimated. The *multi-unit* objective function estimates all the ICs at the same time while the *one-unit* objective function enables the estimation of single ICs. Amongst the latter we can find Negentropy and higher-order cumulants such as kurtosis. Amongst the former, some examples are likelihood and Infomax, Mutual Information (MI) and higher-order cumulants.

Of all these objective functions Hyvarinen *et al.* [62] considered MI to be the most natural and realistic approach to denoising, as it does not assume anything about the data, It is considered the most appealing because it is:

- (i) A strict dependence measure
- (ii) Always non-negative
- (iii) Zero if only the estimated components are statistically independent
- (iv) Based on Shannon's entropy
- (v) Insensitive to invertible transformations of the components

4.1.1 What is Mutual Information?

Mutual Information (MI), also known as transinformation, was first introduced by Shannon in 1948. It is considered to be a non parametric measure of relevance [35] that measures the mutual dependence of two variables, both linear and non linear for which it has a natural generalization. Represented as $I(X:Y)$, MI measures the reduction in uncertainty in X which results from knowing Y i.e. it

indicates how much information Y conveys about X . This relationship can be seen in Figure 4.1 which shows that MI has the following properties [85].

- (i) It is symmetric: $I(X:Y)=I(Y:X)$
- (ii) It is always non-negative between X and Y ; the uncertainty of X cannot increase by learning of Y i.e. $I(X:Y)\geq 0$

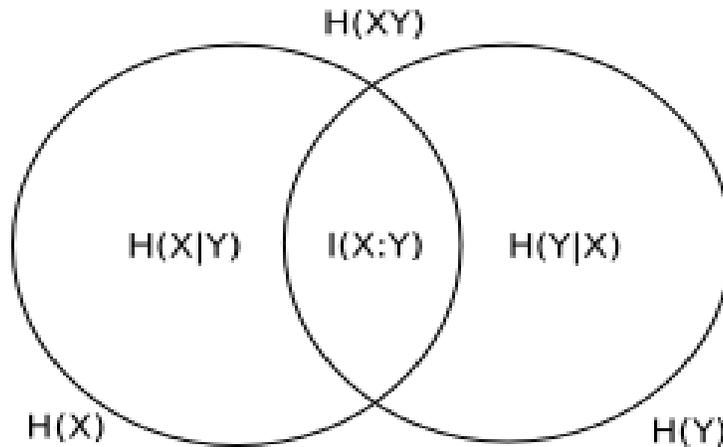


Figure 4.1: Relationship between Mutual information $I(X:Y)$ and entropies $H(X)$ and $H(Y)$ (adapted from M.A. Nielsen, I.L. Chuang, 2000, *Quantum Computation and Quantum Information*, Cambridge University Press)

It also has the following properties:

- (iii) The information X contains about itself is the entropy of X :
 $I(X:X)=H(X)$
- (iv) The information variables contain about each other can never be greater than the information in the variables themselves: $I(X:Y)\leq H(Y)$
 $I(X:Y)\leq H(X)$
- (v) The information in X is in no way related to Y as no knowledge is gained about X when Y is given and visa versa. Here the joint distribution of

the X and Y values hold the exact quality of information; X and Y are considered separately i.e. they are independent: $I(X:Y)=0$

4.1.2 Calculating Mutual Information

There are three different ways to determine MI, namely:

- Using Entropy, considered to be the best way
- Using Probability Density and
- Using Kullback-Leibler Divergence

4.1.2.1 Entropy

The uncertainty of a variable can be estimated using entropy and the relationship between two variables estimated using conditional entropy as in Eq (4.1). MI is therefore defined based on entropy as [78]:

$$\begin{aligned}
 I(X : Y) &= H(X) + H(Y) - H(X, Y) \\
 &= H(X) - H(X | Y) \\
 &= H(Y) - H(Y | X)
 \end{aligned} \tag{4.1}$$

with $H(X, Y)$ being the joint entropy, $H(X)$, $H(Y)$ the entropies of X and Y , and $H(X|Y)$ and $H(Y|X)$ the conditional entropies of X given Y and of Y given X , respectively. These entropies are defined based on probability as:

$$\begin{aligned}
 H(X) &= -\sum_i p(x_i) \log p(x_i) \\
 H(X, Y) &= -\sum_{i,j} p(x_i, y_j) \log p(x_i, y_j) \\
 H(X | Y) &= -\sum_{i,j} p(x_i, y_j) \log p(x_i | y_j)
 \end{aligned} \tag{4.2}$$

Eq. (4.1) contains the term $-H(X, Y)$, which means that maximizing MI is related to minimizing joint entropy.

4.1.2.2 Probability

MI can be defined in terms of probabilities as

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (4.3)$$

where p is the joint probability distribution function of X and Y , and f and g are the marginal probability distribution functions (pdf) of X and Y respectively. In the continuous case, it can be defined with a definite double integral:

$$I(X;Y) = \int \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dx dy \quad (4.4)$$

where p is now the joint probability density function (pdf) of X and Y , and f and g are the marginal pdf of X and Y respectively.

4.1.2.3 Kullback-Leibler (KL) Divergence

MI can also be expressed in terms of the Kullback-Leibler divergence between the joint distribution of two random variables X and Y and the product of their marginal distributions. If $q(x, y) = f(x) \times g(y)$; then

$$I(X;Y) = KL(p,q). \quad (4.5)$$

By definition KL is the measure of the distance between two probability distributions functions $p(i)$ and $q(i)$ and can be written according to [160] as:

$$\sum_i p(i) \log \frac{p(i)}{q(i)} \quad (4.6)$$

MI can therefore be defined in the form of measuring the distance between the joint distribution and the joint distribution in the case of independent variables as:

$$I(X : Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \quad (4.7)$$

4.1.3 Estimating Mutual Information

MI is considered to be very powerful yet there is computational difficulty to estimate it [89, 108-109, 159]. This is because MI requires the pdf of each variable and involves the integration of functions for other functions which can lead to high computational complexity – one problem [113]. To use the definition of entropy in Eq. (4.7), the joint density has to be estimated which demands a duly large amount of data for an acceptable accuracy – another problem. Estimation can therefore be unreliable, noisy and even bias. These problems have severely restricted the use of MI in ICA estimation and many other applications.

Approaches

In recent years researchers have designed different ways of estimating MI. Some researchers have used approximations of MI based on polynomial density expansions, which led to the use of higher-order cumulants. The approximation is valid, however, only when it is not far from the Gaussian density function, and may produce poor results when this is not the case. More sophisticated approximations of MI have been constructed however. Other researchers have estimated MI by binning the coordinate axes [164], the use of histograms [81] as well as the applications of wavelets. All have however, sought to estimate a density $P(x)$ given a finite number of data points x^N drawn from that density function. There are two basic approaches to estimation:

- (i) Parametric which include Bayesian, Edgeworth, maximum likelihood (ML), and least square estimators
- (ii) Nonparametric which includes histogram based, adaptive partitioning of the XY plane, kernel density (KDE), B-Spline, nearest neighbour and wavelet density estimators.

MI Estimators

ICA literature presents mostly crude approximations to MI based on cumulant expansions which became popular because of their ease of use [89] and

they have been very successful [23]. One of the main differences among the various MI-based ICA methods however is the way in which this estimation is dealt with. For example the ICA method using minimum mutual information (MMI) was constructed by Shannon's MI where the difference between the marginal entropy and the joint entropy of different information sources was accumulated. The one difficulty of this method however is the estimation of marginal entropy.

Although all these algorithms exist, Hyvarinen [58] stated that in their present use these algorithms are far from optimal as far as robustness and asymptotic variance are concerned. These algorithms are also sensitive to artifacts. Presently the newest MI estimators for ICA algorithms are using KDE and k -nearest neighbour (k NN) statistics [150]. Krishnaveni *et al.* [90-91] found that a MI estimated using k NN distance outperforms many of the known ICA algorithms. B-Spline estimators according to my research [168] have been shown to be one of the best nonparametric approaches, second to only wavelet density estimators thus better than Krishnaveni's k NN estimator.

For my research I define MI based on the definition of entropy using the second best nonparametric approach – Basis Spline (B-Spline) described below.

4.2 B-Spline Function

Since the early paper by Hou and Andrews, which provides a detailed analysis of cubic spline interpolation, the use of B-Spline representations has had limited applications in signal processing [162]. It would appear that the main reason is that the conventional approach to B-Spline interpolation or approximation is computationally quite expensive for it involves explicit matrix inversions and multiplications. In Signal Processing however where spacing between data points are constant, there is yet to be a simpler approach [126].

B-Spline is a flexible mathematical formulation for curve fitting due to the number of desirable properties [128]. Under the smoothness constraint, B-Spline

gives the “optimal” curve fitting in terms of minimum mean-square error [102, 132]. A 2D B-spline curve can be defined mathematically as:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f(t) \\ g(t) \end{pmatrix} = \sum_{i=1}^{m-1} B_{j,k}(t) \begin{pmatrix} \tilde{x}_i \\ \tilde{y}_i \end{pmatrix}, \quad t_{\min} \leq t < t_{\max} \quad (4.8)$$

where $\left\{ \begin{pmatrix} \tilde{x}_i \\ \tilde{y}_i \end{pmatrix} \mid i=1,2, \dots, m-1 \right\}$ are $m-1$ control points assigned from data samples, t is a parameter and is in the range of maximum and minimum values of the element in a knot vector. A knot vector, $t_1, t_2, \dots, t_{k+(m-1)}$, is specified for giving a number of control points $m-1$ and B-Spline order k . It is necessary that $t_i \leq t_{i+1}$, for all i . For an open curve, open uniform knot vector t is defined according to [27] as:

$$t_i = \begin{cases} 0 & \text{if } i < k \\ i-k+1 & \text{if } k \leq i \leq M-1 \\ m-1-k+2 & \text{if } i > m-1 \end{cases} \quad (4.9)$$

where M is the total number of bins and i is an index into the knot vector.

The shape of the B-Spline functions, in my method, is determined by their order k ($1..M$), which is a parameter of the method. With B-Spline order 1, each point is assigned to exactly one bin and the method is equivalent to simple equidistant binning. The proposed method is thus a fixed binning scheme extended with a preprocessing step designed to reduce the variance. The B-Spline functions are defined and evaluated recursively, based on the Cox-de Boor Recursion formula [27]:

$$B_{i0}(z) = \begin{cases} 1 & \text{if } t_i \leq z < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (4.10)$$

$$B_{i,k}(z) = B_{i,k-1}(z) \frac{z - t_i}{t_{i+k-1} - t_i} + B_{i+1,k-1}(z) \frac{t_{i+k} - z}{t_{i+k} - t_{i+1}} \quad (4.11)$$

where z is an element in the domain of the function i.e. $z \in [0, M-k+1]$. The probability of each bin is estimated by

$$p(x_i) = \frac{1}{N} \sum_{j=1}^N B_{i,k}(f_{M,k}(x_j)) \quad (4.12)$$

where $f_{M,k}(x)$ is a linear transformation which maps the values of x onto the domain of the B-Spline functions. $B_{i,k}$ is a B-Spline function of order k evaluated at bin i ; and $\sim x_j$ is an appropriately scaled data sample mapping the values of x into the domain of the B-Spline function. In two dimensions the joint *pdf* is computed as

$$p(x_i, y_j) = \frac{1}{N} \sum_{l=1}^N B_{i,k}(f_x(x_l)) B_{j,k}(f_y(y_l)) \quad (4.13)$$

The algorithm to determine an Uniform Cubic B-Spline Function (based on [27]) is summarized in Algorithm 1 and code is found in Appendix A.

(a) Determine the validity of variable x

(b) Calculate D_1 based on Cheney and Kincaid (1994)

(c) Determine D_1 with

$$s(x_i) = D_i B_{i-2}(x_i) + D_{i+1} B_{i-1}(x_i) \quad \dots \dots \dots (4.14)$$

where

$$y_i = D_i \frac{h_i}{h_i + h_{i-1}} + D_{i+1} \frac{h_{i-1}}{h_i + h_{i-1}} \quad \dots \dots \dots (4.15)$$

(d) Determine data interval for x

(e) Calculate $B(x)$ with

$$B(x) = \sum_{i=1}^{n+1} D_i B_{i-k}(x) \quad \dots \dots \dots (4.16)$$

Algorithm 1: Algorithm for Uniform Cubic B-Spline Function

4.2.1 Use of B-Spline with Mutual Information

Recently, B-Spline has been widely used in microarray data analysis, including inference of genetic networks, estimation of MI, and modeling of time-series gene expression data [5, 10, 27, 52, 62, 101-102, 153]. In numerical estimation of MI from continuous microarray data [27], a generalized indicator function based on B-Spline has been proposed to get more accurate estimation of probabilities. Klien *et al.* [82] in their research found that the maximisation of MI, in combination with a deformation field parameterised by cubic B-Spline, has been shown to be robust and accurate in many applications. In 2003 Rueckert *et al.* [137] presented MI schemes using B-Spline to help represent the deformation field. Daub *et al.* [27] went on to actually estimate MI using B-Spline. They found that since MI is defined in “terms of discrete variables” B-Spline can be used to perform a numerical estimation to give more accurate estimation of probabilities. Their algorithm avoided the time-consuming numerical integration steps for which KDE are noted. They showed that B-Spline estimated MI outperforms all the other known algorithms for gene expression analysis. Rossi *et al.* [136] stated that B-Spline estimated MI reduces feature selection. It is a good choice as it is non- parametric and model-independent. The other newest form of estimating MI –kNN has a total complexity of $O(n^3p^2)$ while B-Spline worst-case complexity is still less at $O(n^3p)$ thus having a smaller computation time [168]. Rossi *et al.* [136] also stated that B-Spline does not require samples that grow exponentially to provide accurate estimations when estimating joint densities, unlike other estimation methods.

The Estimator

MI as discussed in section 4.1.3 can be estimated based on three different methods. In this dissertation we utilize the Entropy method. This results in Algorithm 2 with code in Appendices A through F.

<p>1. Calculation of marginal entropy for variable x</p> <p>(a) Determine the B-Spline of variable x according to Algorithm 1</p> <p>(b) Sum over all x_u and determine $p(a_i)$ for each bin a_i from</p> $p(a_i) = \frac{1}{N} \sum_{u=1}^N \tilde{B}_{i,k}(x_u) \quad \dots \dots \dots (4.17)$ <p>(c) Determine entropy $H(x)$ according to Eq. (4.2) using 1 variable</p> <p>2. Calculation of joint entropy of two variables x and y</p> <p>(a) Apply steps 1 (a) and (b) to both variables x and y, independently</p> <p>(b) Calculate joint probabilities $p(a_i, b_j)$ for all $M_x \times M_y$ bins according to</p> $p(a_i, b_j) = \frac{1}{N} \sum_{u=1}^N \tilde{B}_{i,k}(x_u) \times \tilde{B}_{j,k}(y_u) \quad \dots \dots \dots (4.18)$ <p>(c) Calculate the joint entropy $H(x,y)$ according to Eq. (4.2)</p> <p>3. Calculate the mutual information $I(x,y)$ according to</p> $I_{M,k}(X : Y) = H_{M,k}(X) + H_{M,k}(Y) - H_{M,k}(X, Y) \quad \dots \dots \dots (4.19)$

Algorithm 2: Algorithm to generate B-Spline estimated MI

4.3 Newly Designed ICA

The aim of all ICA algorithms is to determine S from Eq (3.1). The approach estimates A using the separation/demixing matrix W which is the inverse of A , i.e. $W = A^{-1}$ resulting in the equation below which produces the ICs (ICs), u :

$$u(t) = Wx(t) = WAs(t), \tag{4.20}$$

My ICA algorithm is a fixed point algorithm because these algorithms converge faster than other algorithms [122]. Unlike the gradient descent method, there is no need for the adjustment of learning steps or other adjustable parameters

and the rate of convergence is therefore fixed without regard to the changing environment. Fixed-point algorithms also tend to be much more stable than other algorithms [114]. Like all fixed-point algorithms I have a two-step approach – prewhitening and rotation of the observation vector.

4.3.1 PreWhitening

Prewhitening is a popularly used preprocessing technique in ICA literature which speeds algorithms up substantially. For example many famous ICA algorithms such as FastICA, and JADE, have used this pre-processing technique. It is the actual whitening of a signal ahead of some processing i.e. removing bias and unwanted autocorrelations derived from both internal and external processes, so that all parts of the signal enter the next stage of processing on a level playing field. This amounts to a PCA of the observations. The removal of these autocorrelations is necessary to the interpretation of other potential relationships. Here the matrix W is usually decomposed into two factors

$$W = RV \quad (4.21)$$

where the prewhitening V transforms the covariance matrix into

$$C' = VCV^T = 1 \quad (4.22)$$

and R is a pure rotation. Whitening is done by multiplication with the transformation matrix P .

$$P = VC^{-\frac{1}{2}}V^T \quad (4.23)$$

$$\tilde{W} = P\hat{W} \quad (4.24)$$

4.3.2 Algorithm

Once prewhitening is done the ICA problem is reduced to finding a suitable rotation for the prewhitened data. My algorithm aims to minimize MI between the output signals under the pure rotation R . The performance of my algorithm depends on the nonquadratic nonlinear function G used. The four most used are *pow3*, *tanh*, *gaus*, and *skew*. Since *tanh* is a “good general purpose contrast function” [128] and described as the best at producing a more robust algorithm [131] my algorithm

therefore uses it. As this is a symmetric algorithm, independent sources are extracted simultaneously from the mixed signal. The algorithm is defined in Algorithm 3. The resulting algorithm is called B-Spline Mutual Information Independent Component or BMICA. The actual Matlab code is found in Appendices G and H.

- (a) Center the data to make its mean zero
- (b) Whiten the data according to Eq (15) using B calculated in
- $$y = z' \times A \quad (4.25)$$
- (c) Choose m , the number of independent components to estimate
- (d) Choose initial values for the w_i , $i=1, \dots, m$ each of unit norm. Orthogonalize matrix A as in (f) below.
- (e) For every $i = 1, \dots, m$ let
- $$B = (zg(y)' / m - \sum (1 - g(y)^2)' \times I) / m \quad (4.26)$$
- where g is defined as
- $$g(y) = \tanh(y) \quad (4.27)$$
- and I is calculated based on Algorithm 3
- (f) Do a symmetric orthogonolization of the matrix B according to
- $$B = (BB^T)^{-1/2} B \quad (4.28)$$
- (g) If not converged repeat (e)
- (h) Compute W using B
- (i) Determine u according to Eq (4.20)
- Output: u**

Algorithm 3: Algorithm to generate new ICA

4.3.3 Results

How well does BMICA perform when denoising EEG signals? I tested it using Data sets 1, 2, and 3 and compared its results with the following categories of benchmark ICAs:

- (i) fixed-point- FastICA, Pearson_ICA and EFICA
- (ii) non fixed-point – Infomax, SOBI, and JADE

Each algorithm and performance measure utilized is described also in Chapter 3.

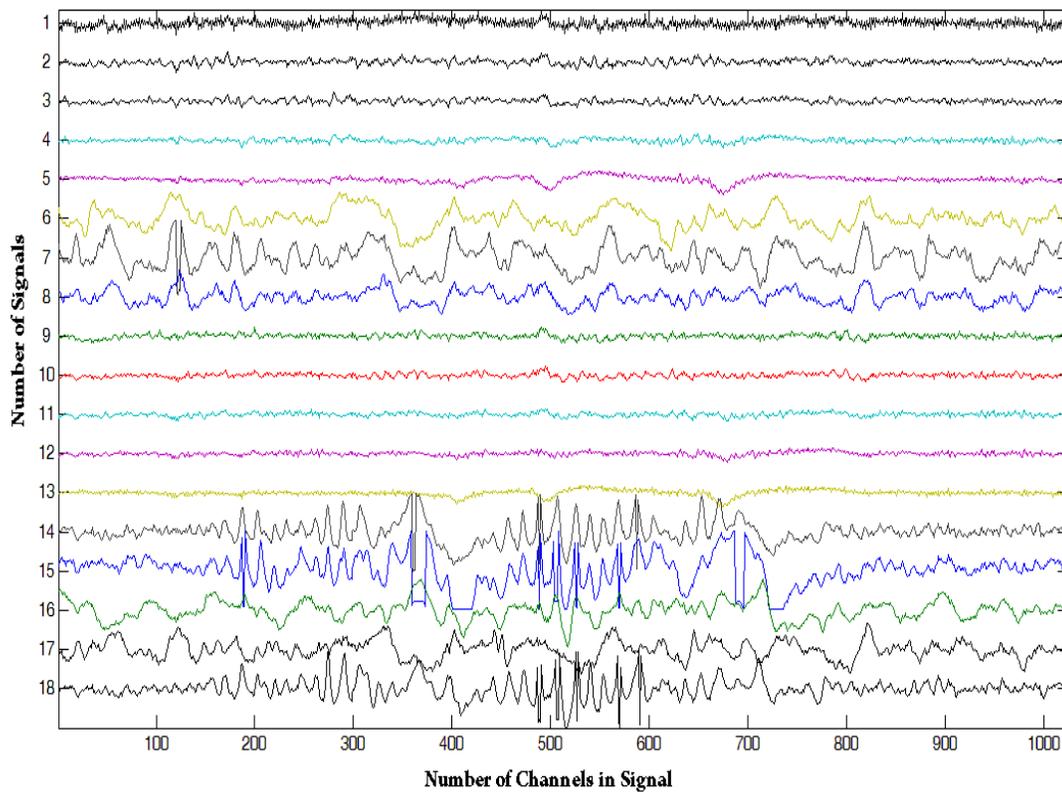


Figure 4.2: Sample of Raw EEG Signals

Figure 4.2 shows one mixed EEG signal set where overlays in signals are in signals 2, 6-8 and 14-18. Figure 4.3 shows the same signal set after the application

of BMICA showing that the overlays have been minimized – noise has been removed.

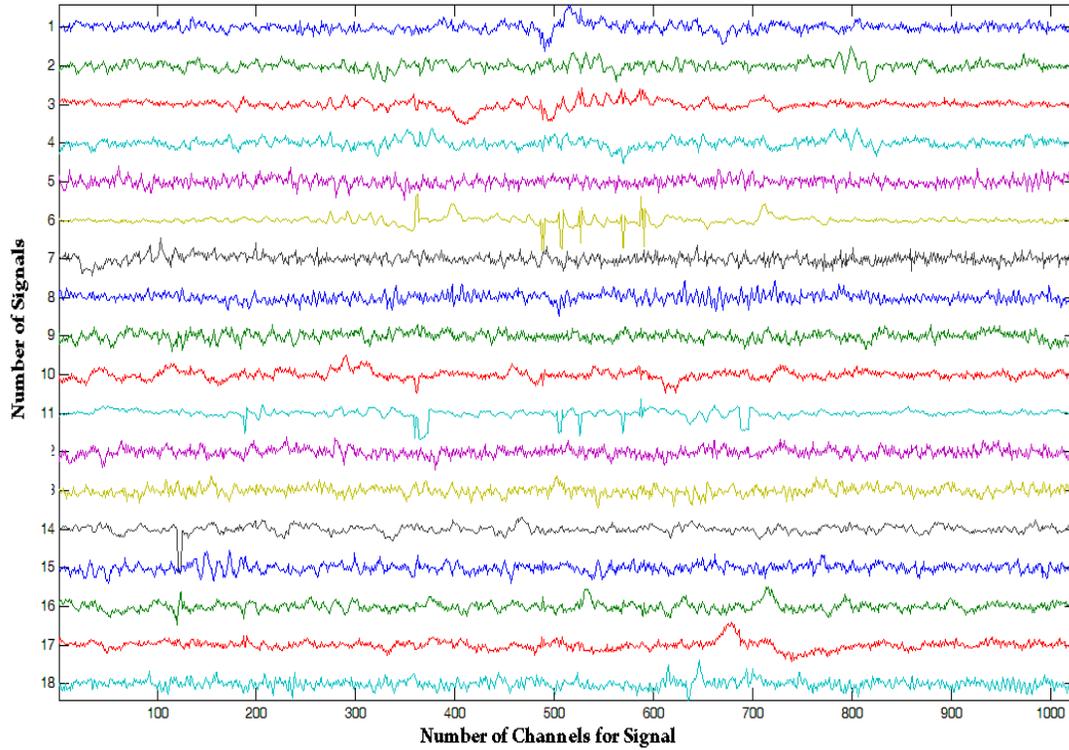


Figure 4.3: EEG Signals after denoised with BMICA algorithm

4.3.3.1 Performance Comparison

BMICA was compared to both categories in Table 4.1. Table 4.1A shows that on average all tested fixed-point algorithms have similar MSE. Further investigations however show that BMICA has the lowest MSE 75% of the time when there are differences in the MSE. Examination of Table 4.2B shows Infomax to have the lowest MSE on average. BMICA performed best in 10 of 15 experiments of the other algorithms.

Table 4.1: MSE comparison with (A) fixed-point algorithms (B) non fixed-point algorithms

BMICA	FASTICA	PEARSON	EFICA
1.66E+03	1.67E+03	1.68E+03	1.69E+03
1.27E+03	1.30E+03	1.27E+03	1.28E+03
1.16E+03	1.17E+03	1.21E+03	1.21E+03
1.81E+03	2.01E+03	2.02E+03	2.00E+03
1.11E+03	1.12E+03	1.12E+03	1.11E+03
1.17E+03	1.53E+03	1.55E+03	1.55E+03
3.14E+03	3.12E+03	3.11E+03	3.11E+03
1.28E+04	1.29E+04	1.29E+04	1.28E+04
4.91E+05	4.91E+05	4.91E+05	4.92E+05
4.63E+05	4.63E+05	4.63E+05	4.63E+05
3.30E+05	3.30E+05	3.30E+05	3.30E+05
9.41E+02	9.63E+02	9.22E+02	9.62E+02
8.79E+02	9.18E+02	9.52E+02	9.82E+02
7.51E+02	7.73E+02	7.57E+02	7.86E+02
6.70E+02	6.65E+02	6.68E+02	6.68E+02
7.09E+02	7.04E+02	7.25E+02	7.17E+02
5.97E+02	5.92E+02	5.95E+02	5.85E+02
4.59E+02	4.62E+02	4.70E+02	4.70E+02
7.30E+04	7.30E+04	7.30E+04	7.30E+04

(A)

BMICA	SOBI	INFOMAX	JADE
1.66E+03	1.67E+03	1.61E+03	1.66E+03
1.27E+03	1.29E+03	1.26E+03	1.31E+03
1.16E+03	1.19E+03	1.12E+03	1.22E+03
1.81E+03	1.01E+03	2.40E+03	2.02E+03
1.11E+03	1.11E+03	1.06E+03	1.08E+03
1.17E+03	1.54E+03	1.49E+03	1.54E+03
3.14E+03	2.13E+03	3.03E+03	3.13E+03
1.28E+04	1.29E+04	1.27E+04	1.29E+04
4.91E+05	4.91E+05	4.91E+05	4.91E+05
4.63E+05	4.63E+05	4.62E+05	4.63E+05
3.30E+05	3.30E+05	3.29E+05	3.30E+05
9.41E+02	9.56E+02	8.96E+02	9.43E+02
8.79E+02	9.18E+02	8.60E+02	9.57E+02
7.51E+02	7.65E+02	7.21E+02	7.58E+02
6.70E+02	6.65E+02	6.22E+02	6.69E+02
7.09E+02	7.16E+02	6.75E+02	7.23E+02
5.97E+02	5.93E+02	5.56E+02	5.85E+02
4.59E+02	4.66E+02	4.32E+02	4.70E+02
7.30E+04	7.29E+04	7.28E+04	7.30E+04

(B)

Table 4.2: PSNR comparison with (A) fixed-point algorithms (B-C) non fixed-point algorithms

BMICA	SOBI	INFOMAX	JADE
15.9207	15.9025	16.0672	15.9205
17.0666	17.0139	17.1141	16.9518
18.3936	18.3244	18.6072	18.3857
18.6811	18.502	18.7848	18.3204
19.3730	19.2944	19.5525	19.3362
17.4924	17.3896	17.6247	17.3567
19.8679	19.9030	20.1919	19.8743
19.6233	19.5816	19.8367	19.5386
17.6643	17.6829	17.8611	17.7787
13.1659	13.1822	13.3197	13.1756
16.2423	16.2448	16.4067	16.2435
20.3691	20.4020	20.6802	20.4578
21.5157	21.4500	21.7757	21.4110
30.0699	30.2565	30.8013	30.1489
29.6448	29.7112	30.1369	29.6404
19.6727	19.6561	19.9174	19.6360

(A)

BMICA	SOBI	INFOMAX	JADE
15.9207	15.9025	16.0672	15.9205
17.0666	17.0139	17.1141	16.9518
18.3936	18.3244	18.6072	18.3857
18.6811	18.502	18.7848	18.3204
19.3730	19.2944	19.5525	19.3362
17.4924	17.3896	17.6247	17.3567
19.8679	19.9030	20.1919	19.8743
19.6233	19.5816	19.8367	19.5386
17.6643	17.6829	17.8611	17.7787
13.1659	13.1822	13.3197	13.1756
16.2423	16.2448	16.4067	16.2435
20.3691	20.4020	20.6802	20.4578
21.5157	21.4500	21.7757	21.4110
30.0699	30.2565	30.8013	30.1489
29.6448	29.7112	30.1369	29.6404
19.6727	19.6561	19.9174	19.6360

(B)

BMICA	FASTICA	PEARSON	EFICA
-8.7827	-8.7836	-8.7842	-8.7889
-8.5257	-8.5226	-8.5206	-8.5211
7.0480	7.0282	7.0285	7.0510
-7.0496	-7.0490	-7.0501	-7.0493
-4.3275	-4.3318	-4.3316	-4.3271

(C)

The higher the PSNR the better the quality of the reconstructed signal i.e. a higher PSNR indicates that the reconstruction is of a higher quality and therefore the algorithm is considered good. Examination of Table 4.2 shows that Infomax is the algorithm that has the highest PSNR then BMICA. Table 4.2C shows BMICA presenting more PSNR values which are higher than the others. BMICA therefore presents more signal than noise in its denoised results than SOBI, JADE FastICA, EFICA and Pearson_ICA resulting in the second best performance. This follows as this is the same behavior with the MSE investigations.

When calculating SNR the greater the ratio, evidenced by a larger number, the less noise and the more easily it can be filtered out. A SNR of 0 however means that noise and signal levels are the same. Although signals contain non-random intelligence and can be isolated and separated, with a 0 SNR, it would be extremely difficult to isolate the signal in real time. On average in Table 4.3 BMICA has the second highest SNR.

Examination of the table shows that there are high variabilities in values, especially with SOBI and BMICA, for example BMICA has a value $-4.90E-03$ while SOBI has a value $-4.48E-07$. This indicates that SOBI has a very small SNR in comparison to BMICA. The aim is to have a high SNR indicating better performance. Further examination shows that BMICA when compared to the other fixed-point algorithms has the highest SNR. The table also shows that of the seven algorithms only Infomax has a higher SNR thus less obtrusive background noise and better signal performance.

Table 4.3: SNR comparison with (A) non fixed-point algorithms (B) fixed-point algorithms

BMICA	SOBI	INFOMAX	JADE
2.26E-02	1.80E-03	1.14E-01	-9.30E-03
6.69E-02	8.86E-04	1.03E-01	-2.12E-02
9.60E-02	8.96E-04	1.93E-01	1.18E-01
2.38E-01	6.15E-04	2.00E-01	-2.14E-01
1.16E-01	1.93E-02	1.88E-01	2.21E-02
-1.24E-02	2.35E-02	1.84E-01	-1.04E-01
3.16E-02	7.04E-05	1.25E-01	7.50E-03
-5.87E+00	2.14E-02	3.24E+01	-5.85E+00
5.83E-02	-3.02E-04	1.60E-01	1.20E-03
-4.01E-02	-3.50E-03	1.17E-01	1.02E-01
8.90E-03	2.62E-04	8.60E-02	-2.81E-02
5.92E-05	-9.32E-07	5.30E-03	7.23E-05
-4.90E-03	-4.48E-07	4.90E-03	5.33E-05
1.05E-02	2.37E-04	3.86E-02	3.20E-03
1.35E-04	2.30E-08	3.60E-03	5.89E-04
4.20E-03	3.53E-04	8.74E-02	7.60E-03
-3.61E-02	2.35E-04	1.50E-01	8.36E-02
5.82E-02	-7.14E-05	1.77E-01	-4.68E-02
-2.92E-01	3.65E-03	1.91E+00	-3.29E-01

(A)

BMICA	FASTICA	PEARSON	EFICA
2.26E-02	7.37E-02	-7.73E-02	-3.43E-02
6.69E-02	-4.05E-02	8.92E-02	2.99E-02
9.60E-02	-7.84E-02	1.25E-01	-5.81E-02
2.38E-01	2.00E-03	-1.43E-01	-2.86E-01
1.16E-01	-5.47E-02	7.69E-02	-1.25E-01
-1.24E-02	6.94E-02	-7.17E-02	7.82E-02
3.16E-02	2.80E-03	-2.56E-02	-2.76E-02
-5.87E+00	-5.85E+00	-5.84E+00	-5.10E+00
5.83E-02	9.86E-02	-7.51E-02	1.92E-02
-4.01E-02	-1.58E-02	-1.90E-03	1.32E-02
8.90E-03	2.85E-02	3.87E-02	4.80E-03
5.92E-05	1.66E-04	-6.12E-05	-5.30E-03
-4.90E-03	-1.70E-03	9.82E-07	-2.72E-05
1.05E-02	-2.50E-03	-6.20E-03	2.39E-02
1.35E-04	8.68E-04	-8.42E-05	3.27E-04
4.20E-03	4.47E-02	1.76E-02	-2.46E-02
-3.61E-02	2.50E-03	-1.20E-02	9.87E-02
5.82E-02	1.37E-02	8.97E-04	-3.53E-02
-2.92E-01	-3.17E-01	-3.28E-01	-3.02E-01

(B)

Table 4.4: SDR comparison with (A) fixed-point algorithms (B) non fixed-point algorithms

BMICA	FASTICA	PEARSON	EFICA
3.46E-01	1.02E-02	3.19E-01	3.38E-01
3.24E-01	3.28E-01	3.19E-01	3.21E-01
5.63E+00	5.68E+00	5.57E+00	5.37E+00
5.46E+00	5.56E+00	5.49E+00	5.55E+00
1.15E-01	1.12E-01	1.14E-01	1.12E-01
7.56E-02	7.47E-02	7.53E-02	7.33E-02
1.55E-01	1.43E-01	1.49E-01	1.41E-01
1.86E-01	3.89E-04	1.69E-01	1.63E-01
1.16E-01	1.10E-01	1.14E-01	1.10E-01
1.75E-01	1.31E-01	1.29E-01	1.30E-01
3.92E-01	3.87E-01	3.79E-01	3.80E-01
1.92E-01	1.95E-01	1.96E-01	1.95E-01
3.39E+00	4.47E+00	3.22E+00	4.42E+00
1.63E-02	1.63E-02	1.63E-02	1.63E-02
3.10E-01	3.07E-01	3.05E-01	3.01E-01
1.13E+00	1.17E+00	1.10E+00	1.17E+00

(A)

BMICA	SOBI	INFOMAX	JADE
3.46E-01	1.43E-04	3.37E-01	3.31E-01
3.24E-01	2.75E-04	3.32E-01	3.26E-01
5.63E+00	3.67E-05	5.83E+00	5.60E+00
5.46E+00	3.73E-05	5.85E+00	5.75E+00
1.15E-01	9.68E-05	1.14E-01	1.10E-01
7.56E-02	4.63E-05	5.43E-02	7.42E-02
1.55E-01	6.61E-05	1.50E-01	1.54E-01
1.86E-01	5.56E-05	1.78E-01	1.64E-01
1.16E-01	1.83E-07	1.13E-01	1.08E-01
1.75E-01	1.44E-07	1.33E-01	1.27E-01
3.92E-01	4.19E-06	3.85E-01	3.89E-01
1.92E-01	1.81E-04	2.01E-01	1.93E-01
3.39E+00	3.60E-03	4.40E+00	4.70E+00
1.63E-02	7.88E-06	1.63E-02	1.63E-02
3.10E-01	-1.16E-05	-2.40E-02	-2.40E-02
1.13E+00	3.02E-04	1.21E+00	1.20E+00

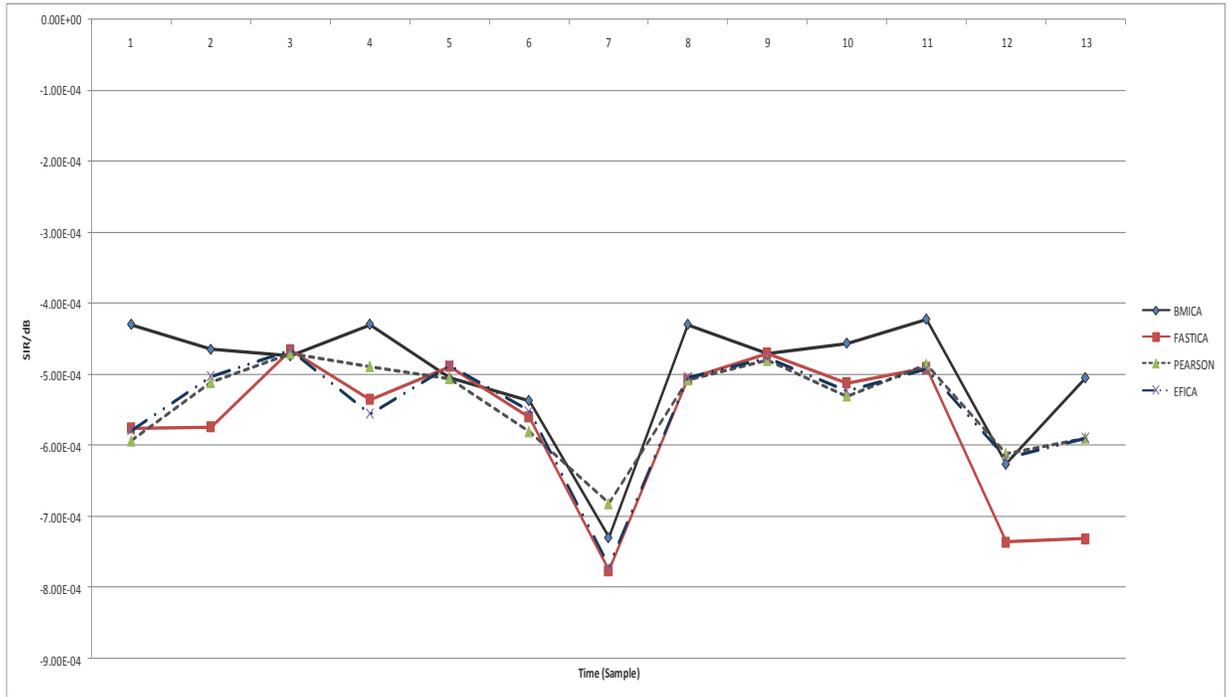
(B)

Consider Table 4.4A; this shows that of the four fixed-point algorithms BMICA has the highest average SDR indicating that BMICA performed the best at separating the EEG from the noise. When BMICA is compared with the three non fixed-point algorithms it was seen that the SDR for my algorithm was superior to the others as Table 4.4B shows.

The lower the SIR, the better the achieved separation and a SIR index of 0 implies a perfect separation. Examination of the algorithms' SIR shows that of the seven algorithms BMICA displays the SIR index nearest to 0, implying a good separation as seen in Figure 4.4.

The Amari indexes obtained for the different algorithms and for different sample sizes are presented in Figure 4.5. From observation it can be seen that BMICA has an Amari separation pattern similar to all the other algorithms as there is basically. An overlay of each algorithm for similar sizes

(A)



(B)

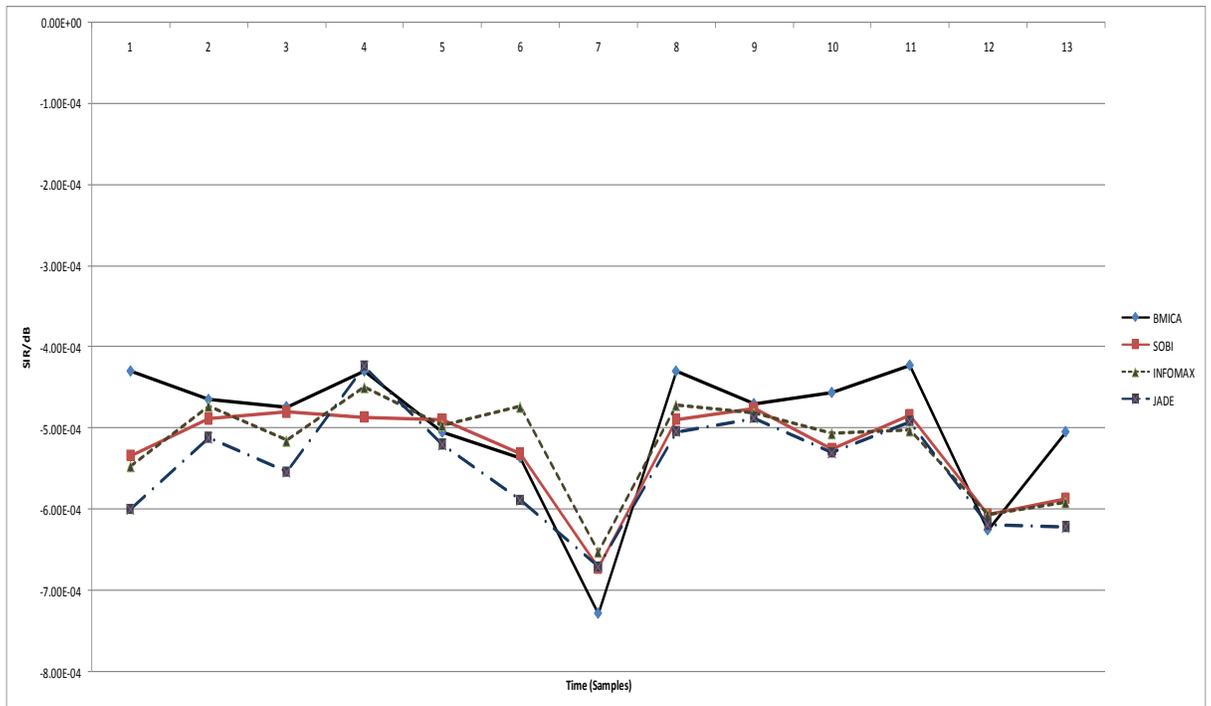
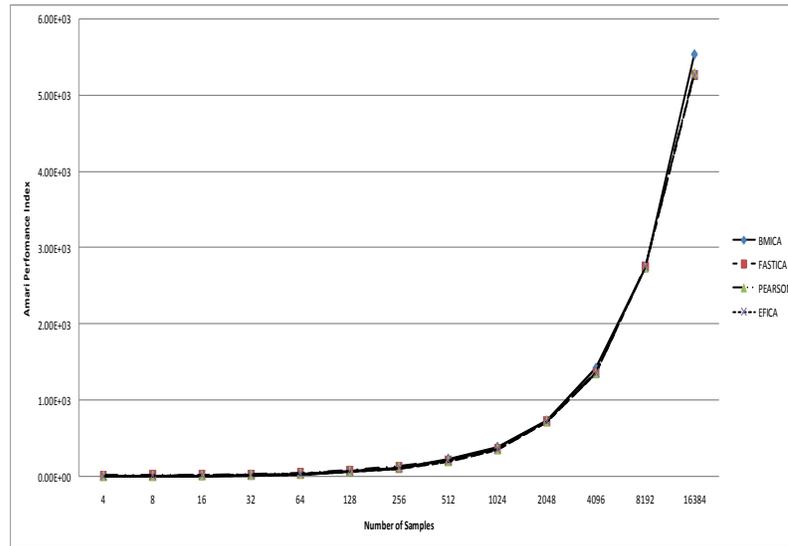


Figure 4.4: SIR comparison with (A) fixed-point algorithms (B) non fixed-point algorithms

(A)



(B)

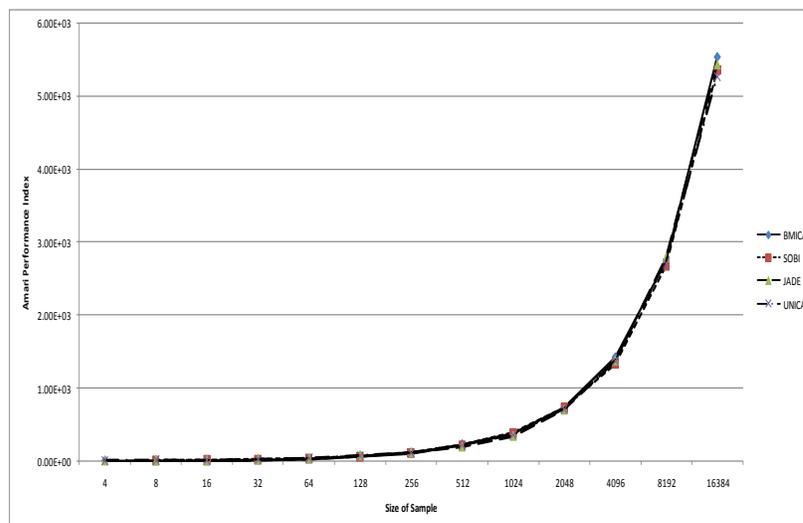


Figure 4.5: Amari Index for (A) fixed-point algorithms (B) non fixed-point algorithms

4.3.3.2 Computational Cost

Although the ultimate goal of a signal separation approach is the quality of such a separation, reflected on the estimated source signals, it is interesting to relate the various ICA approaches from a numeral complexity viewpoint. Therefore we

determined the computational complexity of BMICA as shown below and compare it to the other algorithms.

Let N denote the number of samples, and m denote the number of sources. M is the maximum number of iteration. I assume that $m \leq N$.

- Performing preprocessing is $O(N)$
- Run the Iterations for algorithm is $O(M)$
- Determining the contrast function is $O(N^2)$ – calculating the loop for the matrix containing N signals is $O(N)$ and determine the MI for each signal of m size is $O(N)$
- Determining the matrix to calculate W is $O(N/2)$

BMICA has a complexity of $O(M) + O(N^2) + O(N/2)$ resulting in an overall complexity of $O(N^2M)$. When compared to other ICA algorithms it was found that

- (i) FastICA and Infomax, both have a complexity on the order of $O(N^3M)$ [159],
- (ii) JADE algorithm is on the order of $O(N^4M)$ [159], and
- (iii) EFICA has a computational complexity only slightly (about three times) higher than that of the standard symmetric FastICA [93]

This shows that BMICA has comparable complexity.

It has been shown experimentally that FastICA has outperformed most of the commonly used ICA algorithms [123] and its convergence speed has been the topic of much research [122-123]. We have therefore compared the convergence speed of BMICA with FastICA. FastICA is quadratic in general and cubic for symmetric distributions [58]. Investigations have shown that my algorithm has a slower speed than FastICA. Comparison of the running time has also shown that FastICA completes its analysis much faster than BMICA. In both instances I conclude that the presence of the B-Spline iterative steps influences the speed of the BMICA as on average B-Spline calculates slowly.

4.4 Summary

My creation of BMICA shows that it is possible to create an ICA algorithm based on a B-Spline MI estimator. Throughout this chapter I showed the quality of the BMICA algorithm based on as sessions based on performance, computational complexity and convergence speed. Since it was designed as a fixed-point algorithm its performance as an ICA algorithm was compared to the most widely used fixed-point algorithms - FastICA, Pearson_ICA and EFICA (created as an improvement of FastICA [78]). It was also compared to SOBI, Infomax and JADE, three well used nonfixed-point algorithms, to show overall performance. Assessment shows that BMICA has:

- (i) The best Separation Accuracy as it has
 - the highest SDR
 - the lowest SIR and
 - similar Amari Performance Index to the other six algorithms
- (ii) Relatively good Noise/Signal ratio as it has
 - The highest SNR for fixed point algorithms and third overall next to Infomax and SOBI
 - the highest PSNR for fixed point algorithms and second overall to Infomax
 - the lowest MSE fixed point algorithms and second overall to Infomax

Investigations of the computational complexity of BMICA, when compared to the chosen benchmark algorithms of JADE, EFICA and FastICA, finds BMICA having the smallest complexity. Hence, my algorithm is computationally more intensive. It has been found however that it converges slowly due to the presence of B-Spline. The presence of B-Spline also affects the convergence speed.

Chapter 5 – Reliability of BMICA

Now that we see from Chapter 4 that BMICA is comparable to other ICA algorithms, are the ICS which are produced trust worthy i.e. are they reliable? In this chapter we seek to answer this question.

5.1 Reasons for Reliability Testing for ICA algorithms

ICA is an iterative algorithm that aims at producing a new system from the old with the first being the most non-gaussian moving to the most gaussian. Because it is iterative however, the results in most algorithms tend to differ when executed multiple times on the same dataset. The problem is that most ICA algorithms are based on methods related to gradient descent. The basic principle is to start in some initial point, and then make steps in a direction that decreases the objective function, until one finds a point in which the objective function is locally minimized. Depending on the point where the search was started (the “initial point”), the algorithm will find different local minima [55]. Results from each algorithm therefore have a level of uncertainty, which components should be considered seriously is not clear.

There is no set size for datasets used in ICA algorithms. This generates a problem, even if the algorithm is deterministic, i.e. finding the global optimum always [47]. Finite datasets induce statistical errors in the estimation. Secondly once the dataset contains real data it never follows the ICA model, defined in Eq. (3.2), exactly thus resulting in the estimations having many local minima which are all equally acceptable. This means that for real data it is not clear which components of the output should be interpreted meaningfully. Complications can be avoided however by assessing the reliability of the ICs of a signal i.e. how well does the ICA model in Eq. (3.2) fits the actual signal. The testing of the reliability of a signal’s ICs allows for the tuning of the application of the ICA to enhance reliability e.g. it can

be used to determine the necessary amount of data and/or how best to pre-process the data. Focusing on reliable ICs also prevents the unreliable ICs from obfuscating subsequent analysis [47].

5.2 Previous Research on Reliability

Several techniques have been developed to assess ICs' reliability. They vary from investigating the frequency of ICs repeating to assessment of individual ICs. Of the two, individual testing of ICs has become more valuable, so in this research we focused on tests based on this mode.

Reliability testing goes as far back as 1987 when Friedman [36] proposed a robust structural measure to arrange the ICA basis vectors so as to rank them based on their closest to the desired results.

In 2002 Meinecke *et al.* [112] proposed a bootstrap resampling method that estimates the reliability and grouping of ICs produced for ICA. Although this method utilized the bootstrap theory, some algorithms had problems trying to define a resampling strategy that preserves the statistical structure relevant to the considered ICA algorithms. It was also only applicable to deterministic algorithms which were not stochastic; most ICAs are non-deterministic and stochastic. A year later Harmeling *et al.* [49] presented a new method that constructively injected noise to assess the reliability and the group structure of empirical ICA components. Although they showed empirically and with toy experiments that the method worked, there was no known developed mathematical theory to support the approach.

Himberg *et al.* [55] applied bootstrapping, a well-known computational method for computing statistical reliability and clustering. They tested this method, called ICASSO, on the popular FastICA and found that it could be used to successfully determine the reliability of the tested algorithm. Stogbauer *et al.* [150] proposed a dual testing method looking at (i) how each ICA model conformed

to the standard ICA model and (ii) the uniqueness of each IC based on rotation in a 2-dimensional sub-space. They showed that this method is applicable to both artificial and real-data. Both Himberg's and Stogbauer's methodologies were applied to only one algorithm however, so the performance with other algorithms was unknown until Ylipaavalniemi and Vigario in 2004 [182]. In this work, the Infomax algorithm was applied to dimension reduced data at different orders for multiple Monte Carlo trials and the ICA estimation results at each order were analyzed with the ICASSO software package.

The newest method was developed in 2009 when Groppe *et al.* [47] tested the ICs reliability by running an ICA on user-defined split-halves of the data. This method is comparable to the other methods and confirms the importance of checking the reliability of ICs.

5.3 The ICASSO Reliability Test

ICASSO was developed in 2003 as a reliability testing method for ICA algorithms [55]. It is based on estimating a large number of candidate ICs by running an ICA algorithm many times, and visualizing their clustering in the signal space. Each estimated IC is one point in the signal space. If an IC is reliable, (almost) every run of the algorithm should produce one point in the signal space that is very close to the "real" component. Thus, reliable ICs correspond to clusters that are small and well separated from the rest of the estimates [55-56]. The method consists of the following steps:

1. Selection of parameters for the ICA estimation algorithm.
2. Execution of the ICA algorithm n times using the selected training parameters. Each time the data is bootstrapped and/or the initial point of the optimization is changed.
3. Clustering of estimates (estimated components) according to their mutual similarities. The measure of similarity is the absolute value of the linear

correlation coefficient between the ICs.

4. Visualization of clusters as a 2-D plot and a dendrogram.
5. Retrieval of estimates belonging to certain cluster(s) for further analysis and visualization [55].

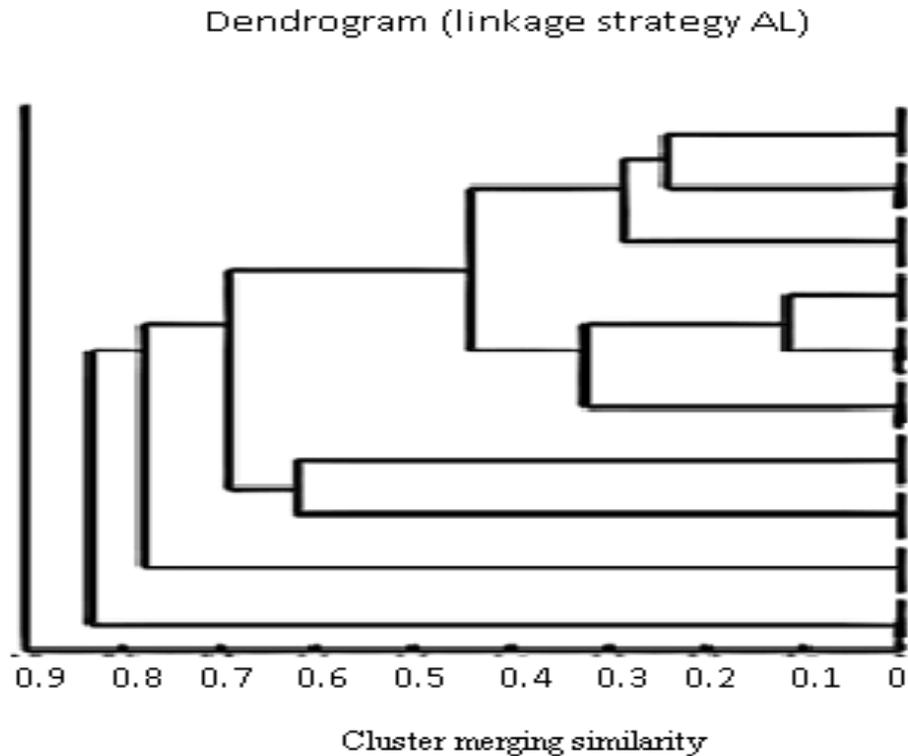


Figure 5.1: Sample of a Dendrogram. Points are successively joined into clusters when moving upwards in the dendrogram. The horizontal axis gives the dissimilarity for which the clusters are merged. Clustering can be performed at any level. (adapted from Himberg, J. and Hyvarinen, A. 2003. *ICASSO: Software for Investigating The Reliability Of ICA Estimates By Clustering And Visualization. IEEE Workshop on Neural Networks for Signal Processing (NNSP2003):259-268*)

Once the method has been engaged the ICs are partitioned into a number of disjoint clusters using agglomerative hierarchical clustering. The tree-like hierarchy (dendrogram) produced by agglomeration is intuitively appealing in the sense that all clusters implied by lower levels of the tree are always subsets of clusters at the

higher levels [56]. To have the best results the agglomeration strategy utilized is average-link (AL), which was used to identify the cluster of IC estimates attributing to the same underlying independent source. A dendrogram based on AL is illustrated in Figure. 5.1.

To direct the attention of the user to those clusters that seem to be the most compact and interesting, the method presents three indices:

- (i) I_q , a stability quality index that reflects the compactness and isolation of a cluster. It is computed as the difference between the average intra-cluster similarities and average extra-cluster similarities and defined below as:

$$I_q(C_m) = \frac{1}{|C_m|^2} \sum_{i,j \in C_m} \sigma_{ij} - \frac{1}{|C_m||C_{-m}|} \sum_{i \in C_m} \sum_{j \in C_{-m}} \sigma_{ij}, \quad (5.1)$$

where C_m is the set of indices of all the estimated components belonging to the m -th cluster, $|C_m|$ is the size of the m -th cluster and C_{-m} is the set of indices that do not belong to the m -th cluster.

- (ii) I_R , a quantitative index, that suggests the clustering that is best fits to the “natural” structure of the data i.e. the best partitions and defined as:

$$I_R = \frac{1}{L} \sum_{m=1}^L \frac{S_m^{in}}{S_m^{ex}} \quad (5.2)$$

where L is the number of clusters,

$$S_m^{in} = \frac{1}{|C_m|^2} \sum_{i,j \in C_m} d_{ij} \quad \text{and} \quad S_m^{ex} = \min_{m' \neq m} \frac{1}{|C_m||C_{m'}|} \sum_{i \in C_m} \sum_{j \in C_{m'}} d_{ij}$$

- (iii) A trustworthiness index based on Curvilinear Component Analysis (CCA), a powerful non-linear mapping algorithm which efficiently unfolds high dimensional data structures towards their mean manifold reproducing the topology of the original data in the projection space without fixing in a static way the configuration of the topology [48,185].

5.4 Comparison ICA Algorithm Tested - FastICA

FastICA is one of the most popular ICA algorithms. It has been found that this is because of:

- (i) its simplicity
- (ii) its satisfactory performance in several applications [185] and
- (iii) its convergence speed [182]

The FastICA was described in Chapter 3.

5.4.1 Reasons for Comparison

5.4.1.1 Performance Testing of BMICA

In Chapter 4 BMICA was discussed and investigations done on its performance described. Experiments were mainly aimed at assessing the method's ability to perform ICA (extraction of ICs). The complexity of BMICA was also compared that those of JADE, EFICA and FastICA showing that it has the smallest overall complexity. Further experiments also showed that BMICA has a slower convergence speed than FastICA [169]. The algorithmic and statistical reliability of the estimated ICs from BMICA however were not investigated.

5.4.1.2 Why compare to FastICA

It has been shown experimentally that FastICA has outperformed most of the commonly used ICA algorithms such as JADE [79, 112] and its convergence speed has been the topic of many research [18, 96, 107]. Mantini *et al.* [107] compared six known algorithms including FastICA and concluded that FastICA had the best overall performance in terms of both separation quality and computation times. In 2009 Glass *et al.* [43] presented a method for evaluating ICA separation of artifacts from EEG data which was tested using FastICA and Infomax. They found that both were highly accurate in their ability to separate out the simulated blinks from the EEG. My test for BMICA is based on EEG signals.

Performance testing in literature has shown that of all the algorithms applied to biomedical signals Infomax is the only algorithm which tends to compare with FastICA in performance [36, 79, 112]. For my research however FastICA is used instead of Infomax because:

1. FastICA has been tested for its reliability using ICASSO [55] the software of choice
2. FastICA had been tested using Friedman's reliability testing and when compared to other algorithms such as EASI, Infomax and Pearson-ICA was found to be the most reliable especially with symmetric orthogonalization and exponential nonlinearity [119]
3. Reliability testing of the ICA decomposition for Infomax, FastICA, JADE and EGLD ML were done in 2007. Results showed that FastICA outperformed the other algorithms in five(5) of the seven(7) test points, resulting in the highest performance/reliability [115].

Reliability testing is the final test for the BMICA algorithm. Since FastICA has undergone many reliability testings with most of the other well known algorithms, it was used as a bench mark for the reliability testing for the proposed BMICA algorithm. A comparison test with FastICA I think would therefore highlight BMICA's performance especially since it has outperformed this algorithm in all other forms of performance testing.

5.5 Results

In this study, I used Laptop environment 1 (MATLAB 7.8.0 (R2009) on a laptop with AMD Athlon 64x2 Dual-core Processor 1.80GHz) to implement ICASSO [48-54] with 10 runs using both bootstrapping and random initial conditions, with a low epsilon threshold (0.0000001) so that optimization would not converge too soon before reaching extrema. I ran ICASSO for the same data. Each time the number of randomization was 10. I employed FastICA [60] in both modes, applying the four types of linearity mentioned in 4.1, to the same data and conditions.

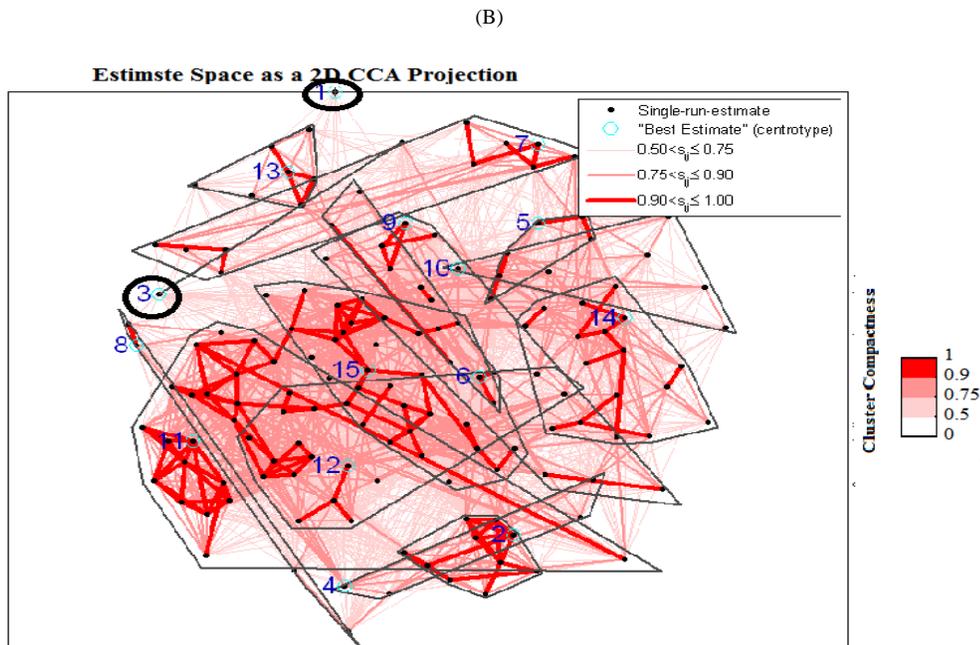
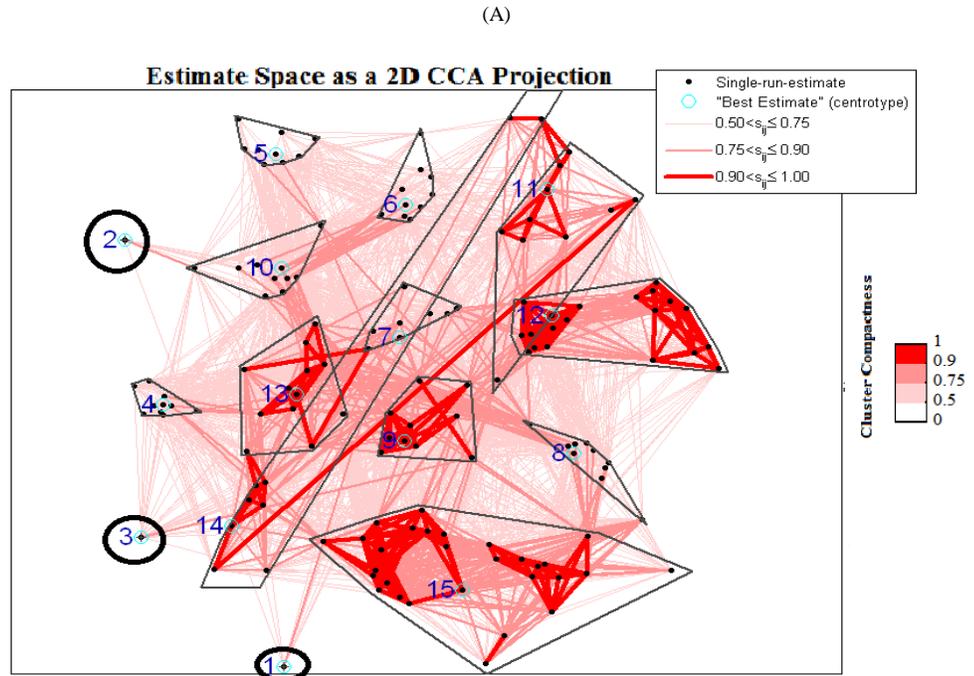


Figure 5.2: Cluster plots for (A) BMICA (clusters 1, 2, 3 are ideal) and (B) FastICA (clusters 1 and 3 are ideal). S_{ij} denotes the correlation between estimates occurring in other clusters. Clustering is based on the similarity between the components; the 2D projection is based on the Euclidean distance as a metric of the dissimilarity between the components. Compact and isolated hulls indicate reliable estimates

5.5.1 General Analysis Using ICASSO

In the following we present the results for one of the datasets tested (as described above) using the number of clusters, L , to equal 15 which is the dimension of the data. Since BMICA has been designed to have a linearity of \tanh and a symmetric orthogonalization, the FastICA algorithm tested also had the same.

5.5.1.1 Clustering the Estimates

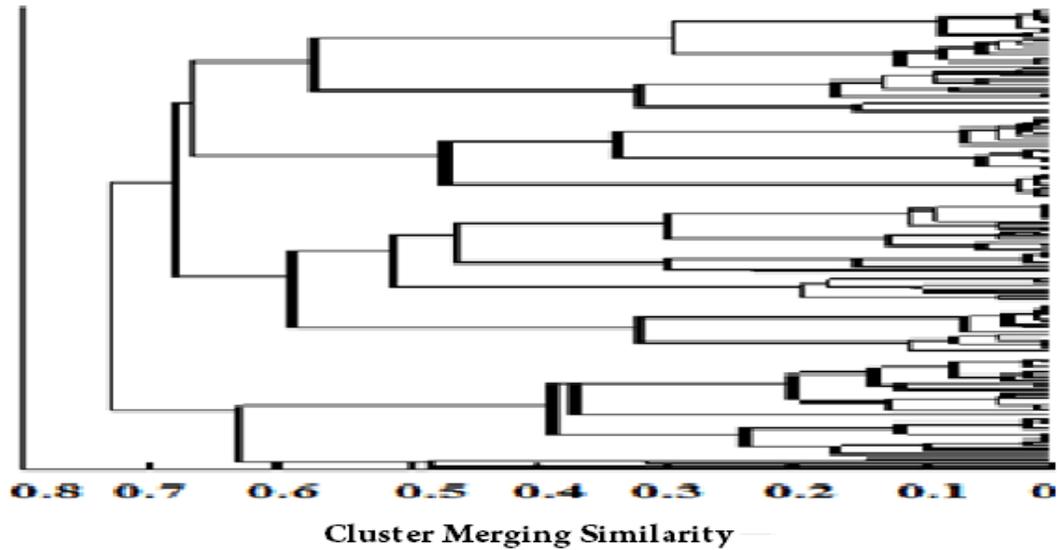
The clustering of the estimated components is expected to yield information on the reliability (robustness) of estimation. In line with the stability indices, Figure 5.2 shows the 2D projections of the clustered IC estimates for BMICA and FastICA respectively. Each IC cluster is prescribed by a convex hull with the black dots representing the individual IC estimates. In the figure the lightly colored lines connect estimates whose similarity is larger than a threshold and the darker lines indicate that there are stronger similarities.

A compact, tight cluster emerges when a similar component repeatedly comes up despite the randomization. In Figure 5.2 the smaller clusters are indicated by smaller, tighter convex hulls and the tightness of a cluster corresponds to a component's reliability. It can be seen that the clusters for BMICA are smaller than those of FastICA indicating that BMICA produces more reliable ICs.

If an IC is reliable then every run of the algorithm produces one point that is very close to the real component thus producing a tight cluster. The tighter a cluster, the smaller it becomes resulting in a single point. This is an ideal cluster [55]. There are ideal clusters in #1, #2 and #3 for BMICA and #1 and #3 for FastICA. Analysis therefore shows that BMICA produces more reliable ICs than FastICA. It has been observed also that some of the clusters run into each other for FastICA.

(A)

Dendrogram (linkage strategy : AL)



(B)

Dendrogram (linkage strategy: AL)

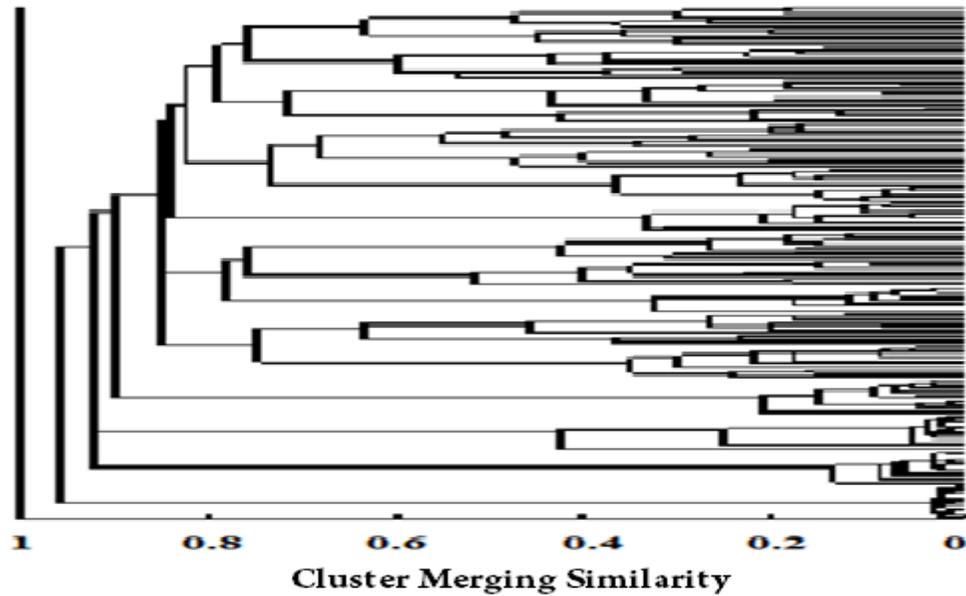
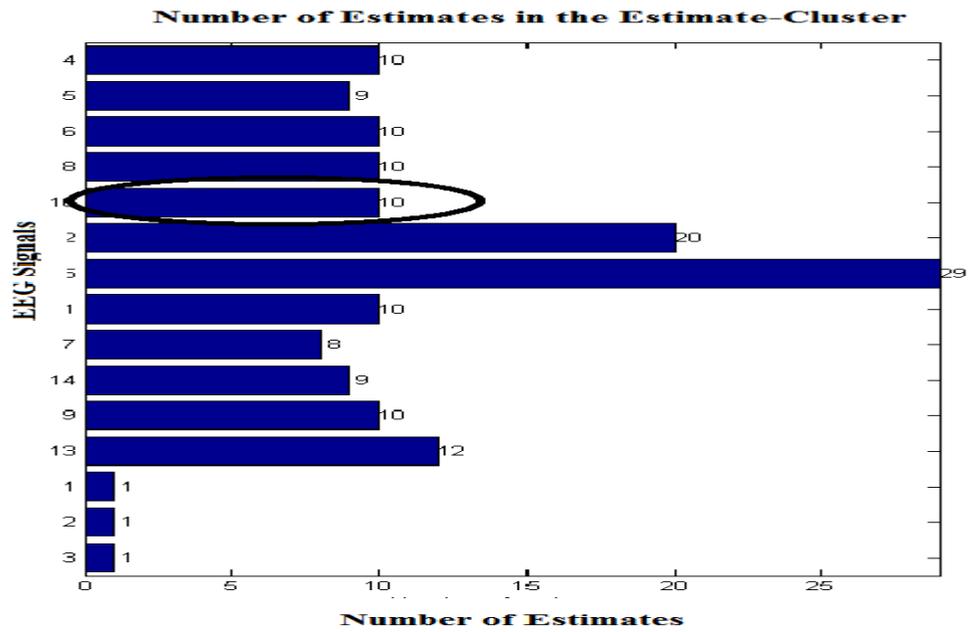


Figure 5.3: Dendrogram for (A) BMICA and (B) FastICA. Dendrogram illustrates the arrangement in 15 clusters (as suggested by the R-index) of the ICA-estimates. The horizontal axis represents the dissimilarity values at which clusters are merged at each possible partition level. The vertical axis indexes ICA-estimates

(A)



(B)

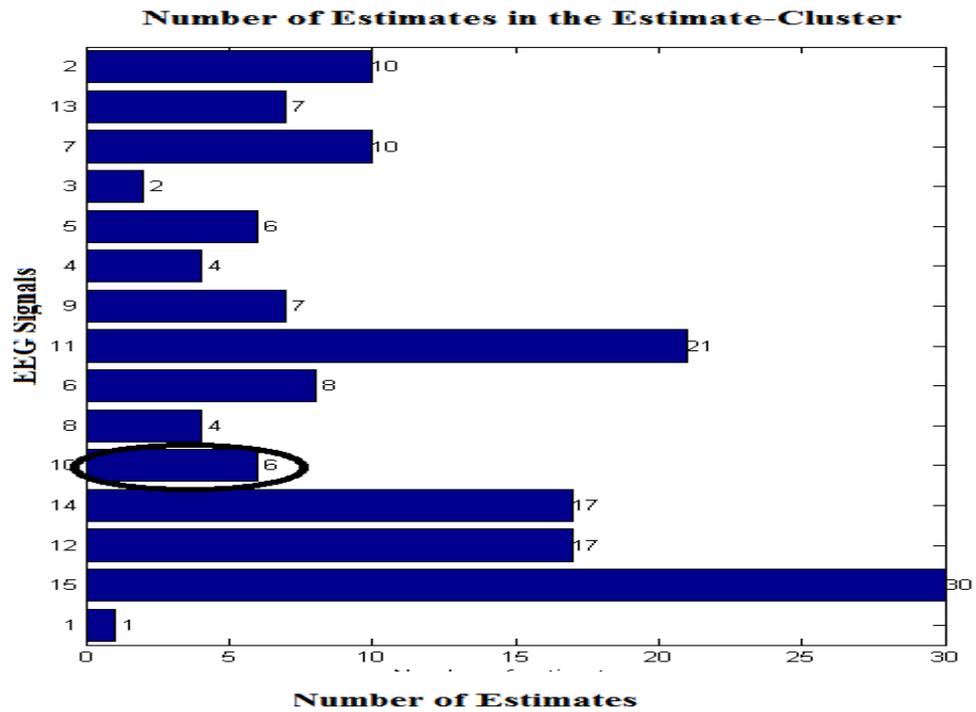


Figure 5.4: Estimates per cluster for (A) BMICA and (B) FastICA. The number of estimates in each component indicates the similarities between the points – the more estimates the more similarities.

5.5.1.2 Using a Dendrogram

ICASSO creates a tree of clusters called the dendrogram. Examination of Figure 5.3 confirms the cluster diagrams, for fifteen(15) clusters, in Figure 5.2. Here it can be seen that the fragment boxes corresponding to a single cluster in each dendrogram are joined by reasonably low tie bars i.e. have high similarity. The dendrograms show good clustering as there are large separations between the tie bars that link fragments within a cluster and the tie banks that link separate clusters. In the dendrogram shown in Figure 5.2 it can be seen that there are more fragment boxes for BMICA which join to form clusters. Figure 5.4 confirms that there were actually fifteen (15) clusters.

The distance measure between two clusters is calculated using the formula

$$D = 1 - C \quad (5.3)$$

where D is distance and C is the correlation between joined clusters. If the clusters are highly correlated, they will have a correlation value close to 1 and so $D=1-C$ will have a value close to zero. Therefore, highly correlated clusters are nearer the bottom of the dendrogram. In Figure 5.2 the clusters for BMICA are closer to the bottom of the dendrogram than those of FastICA indicating that the clusters for BMICA are higher in correlation.

5.5.1.3 The Quality Index (I_q)

ICASSO returns a stability (quality) index (I_q) for each estimate-cluster. This gives a rank for the corresponding ICA estimate. In the ideal case of m one-dimensional ICs, the estimates are concentrated in m compact and close-to-orthogonal clusters. In this case the index to all estimate-clusters is (very close) to one producing an ideal cluster. The value drops when the clusters grow wider and mix up. This implies that the larger the size of the diameter of the convex hulls, the smaller the I_q .

Figure 5.5A shows the I_q from BMICA. The I_q of each cluster is calculated resulting in stability indices of most of the clusters ranging from 0.45 to 0.7 indicating that the clusters were getting more compact thereby indicating that the estimates are more consistent thus more reliable. The same pattern exists in Figure 5.5B for FastICA; however I_q increases from 0.35 to 0.6. Further analysis of the I_q also reveals that 67% of ICs for BMICA have I_q of 0.5 and over while for FastICA is only 20%. The I_q for FastICA are smaller thus indicating that the clusters are wider and mixed up. This confirms that the clusters in BMICA are more compact than those of FastICA thus the ICs are more reliable.

The points found within the convex hulls are connected with lines whose thickness represents the similarities between them. Figure 5.4 shows the number of IC estimates within each cluster (each cluster is labeled on the y-axis). This indicates the thickness of the lines in the clusters i.e. more estimates indicates thicker lines. When the lines within clusters are compared for BMICA and FastICA it can be seen that the lines in BMICA are thicker indicating that there are more similarities between points. It can be seen for example, that for cluster #10 BMICA has 10 estimates while FastICA has 6. In total FastICA only produced 4 (27%) clusters of the 15 which were thicker than BMICA.

It is observed in the figures also that clusters #1, #2 and #3 for BMICA possess a similar number of estimates to produce ideal clusters and clusters #1 and #3 for FastICA.

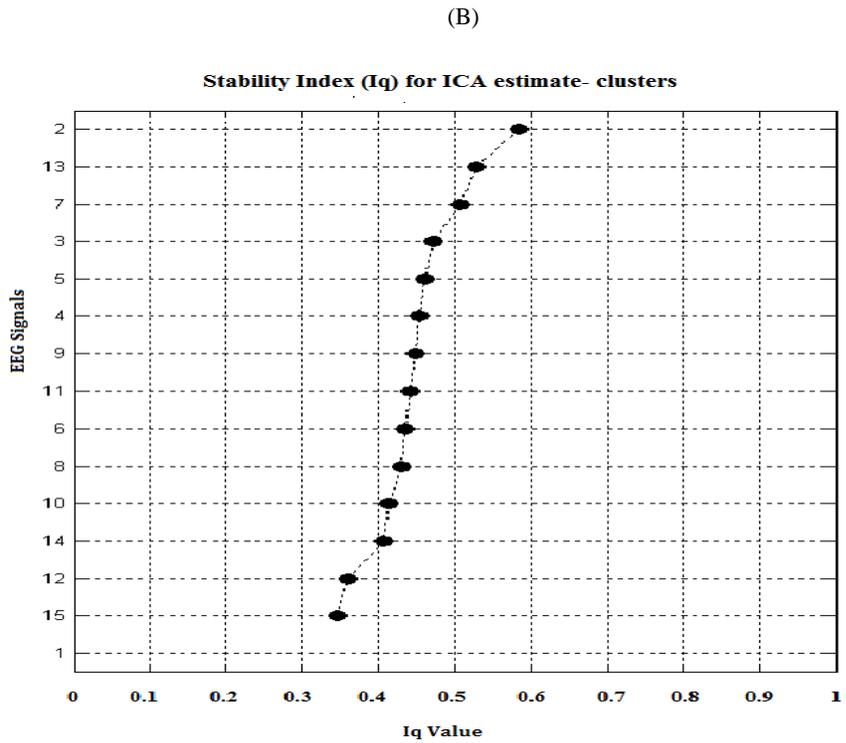
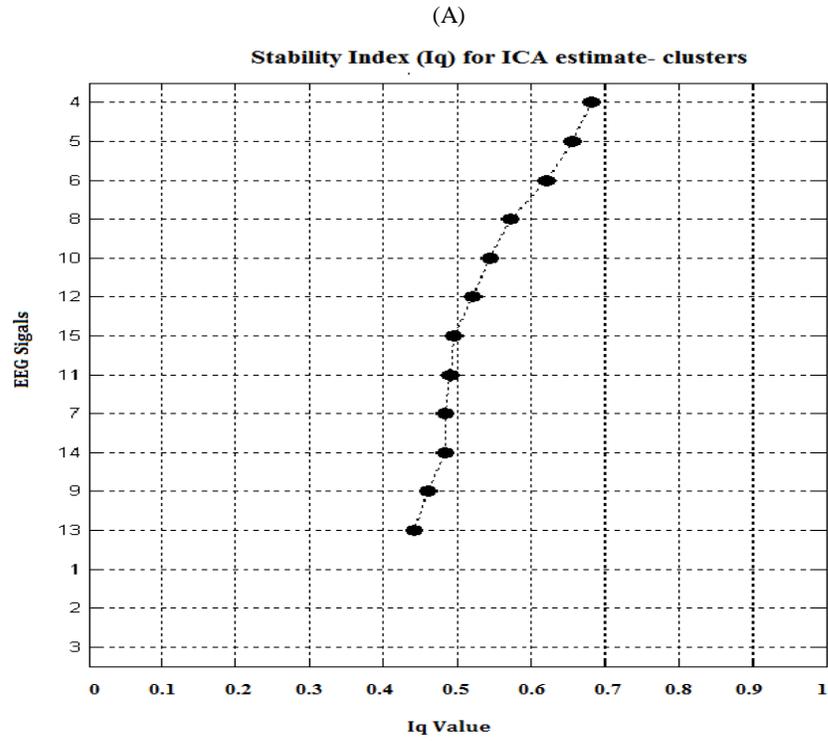


Figure 5.5: Estimate Quality for (A) BMICA and (B) FastICA. Iq shows how signals, after 10 runs, appeared. The closer to 1 a signal Iq implies more consistent estimates thus more reliable signal

5.5.1.4 Centrotype Analysis

A centroid (calculated center point) of a cluster is considered a more reliable estimate than any estimate from an arbitrary run. Instead of an average as a centroid, ICASSO visualizes and returns a centrotype from each cluster. This is one of the original estimates that is most similar to other estimates in the same cluster. Each centrotype is identified as the representative independent component for each cluster. Each is defined as the point in the cluster that has the maximum sum of similarities to the other points in the cluster. The centrotypes associated to the clusters are presented in Figure 5.6 for BMICA and FastICA respectively. Examination of these figures show that in repeated experiments the centrotypes are presented in quality rank order, i.e. the centrotypes are in reducing stability order with the top estimates. The centrotypes were always presented in the same order in every experiment. Further examination of Figure 5.6 shows that for BMICA the ranking order is similar to the I_q placement in Figure 5.5. This is not the same in FastICA. BMICA therefore reflects more consistency.

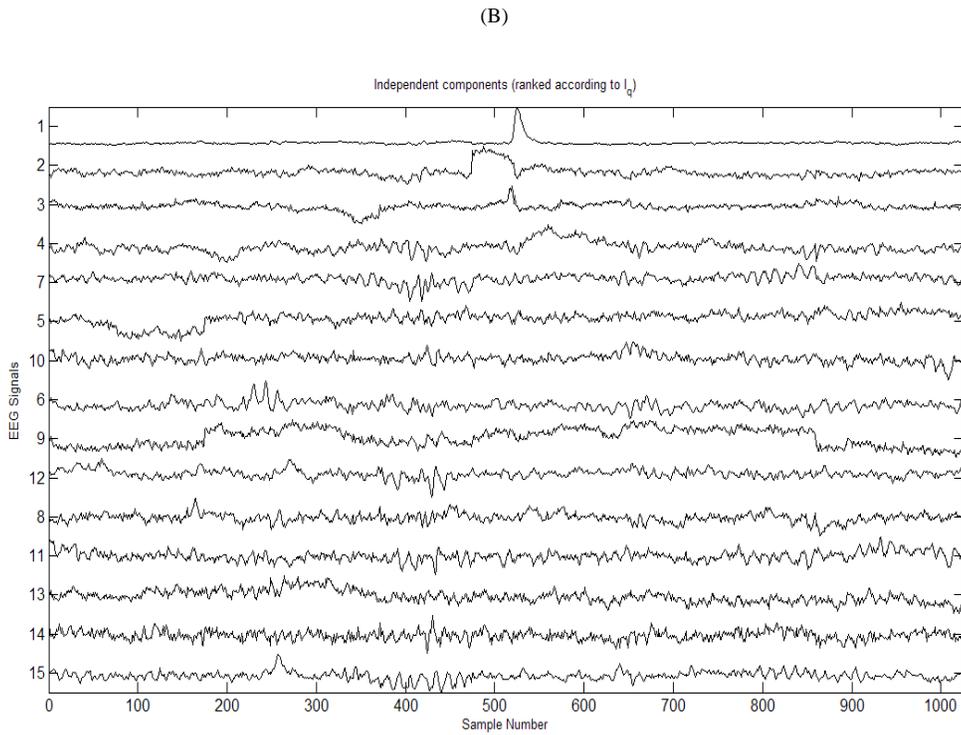
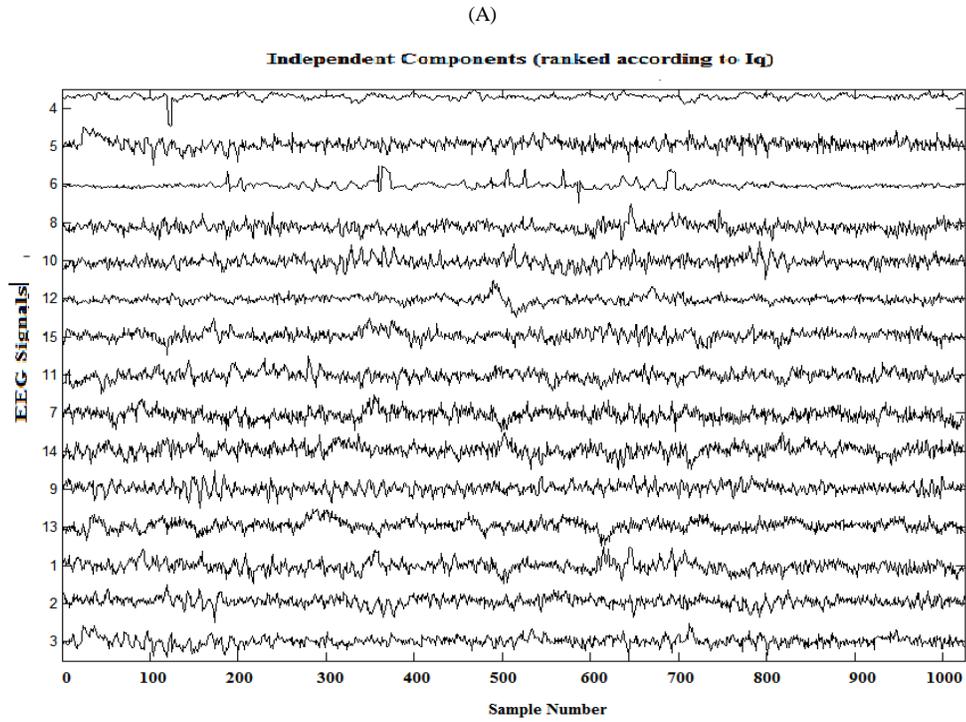
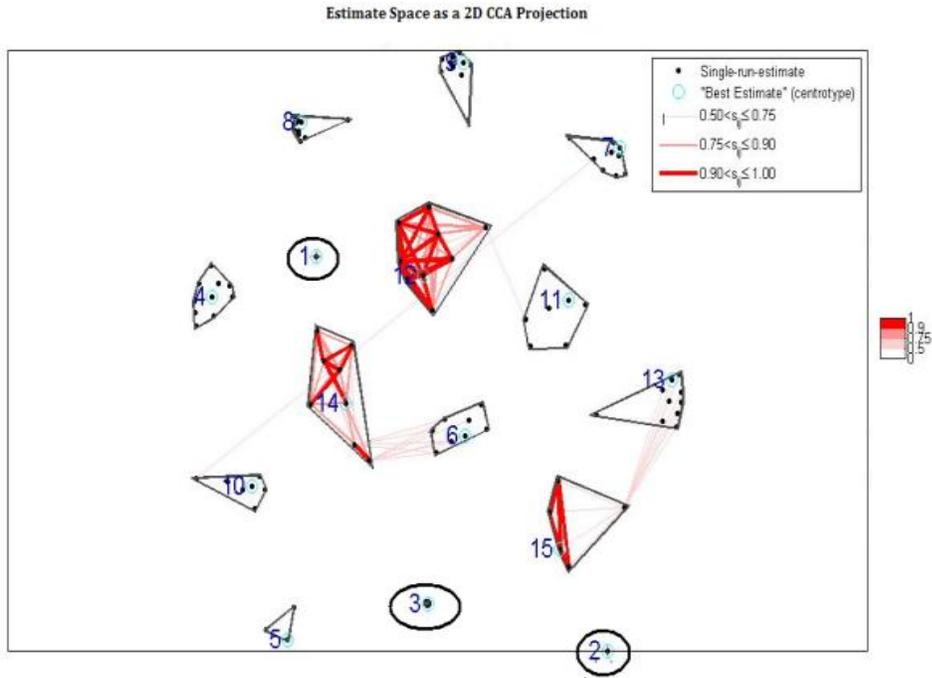


Figure 5.6: Centrotypes for (A) BMICA and (B) FastICA. Each cluster was uniquely represented by a single centertype ICA-estimate, which is just the estimate in the cluster that has the maximum sum of similarities to other points in the cluster

(A)



(B)

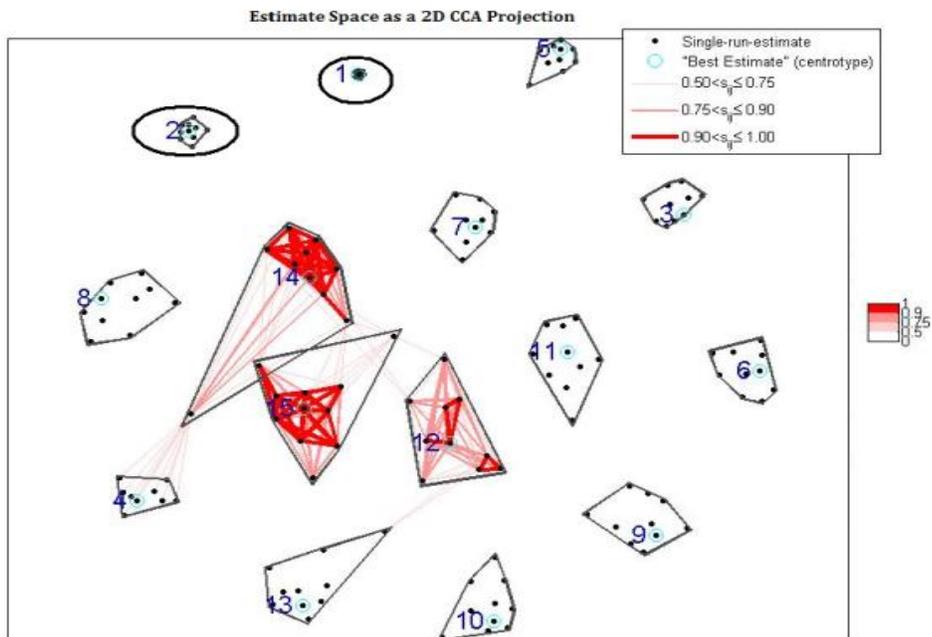
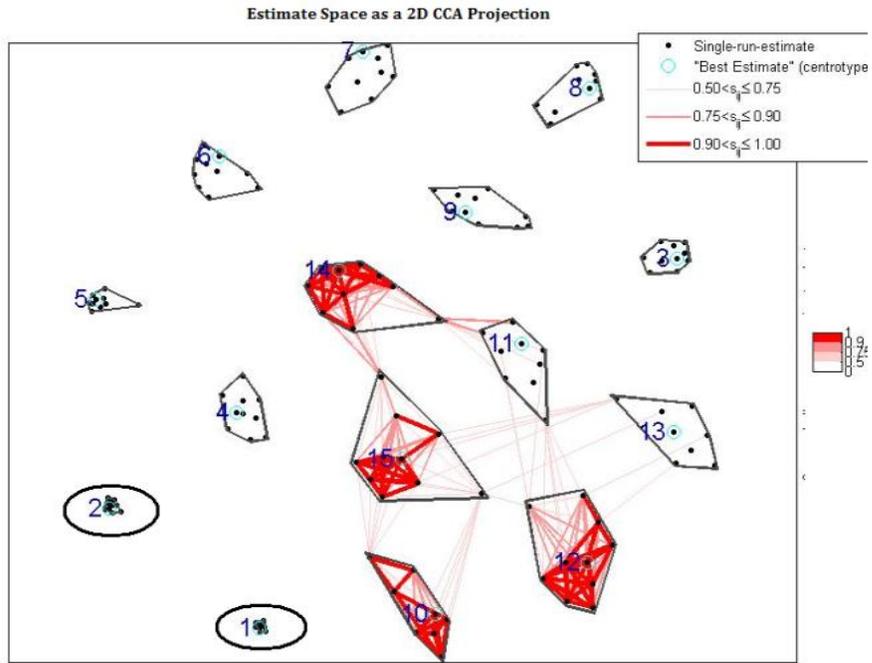


Figure 5.7: Cluster plots for (A) FastICA-*defl,gauss* (clusters 1, 2, 3 are ideal) and (B) FastICA-*symm,gauss* (clusters and 3 are ideal).

(A)



(B)

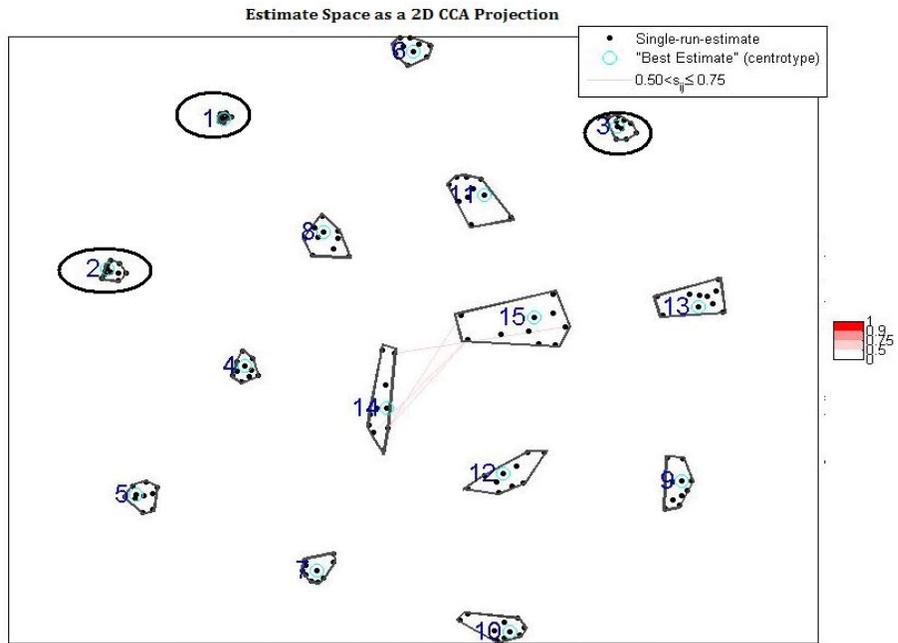


Figure 5.8: Cluster plots for (A) FastICA-*defl,skew* (clusters 1, and 2 are ideal) and (B) FastICA-*symm,skew* (clusters 1, 2, and 3 are ideal).

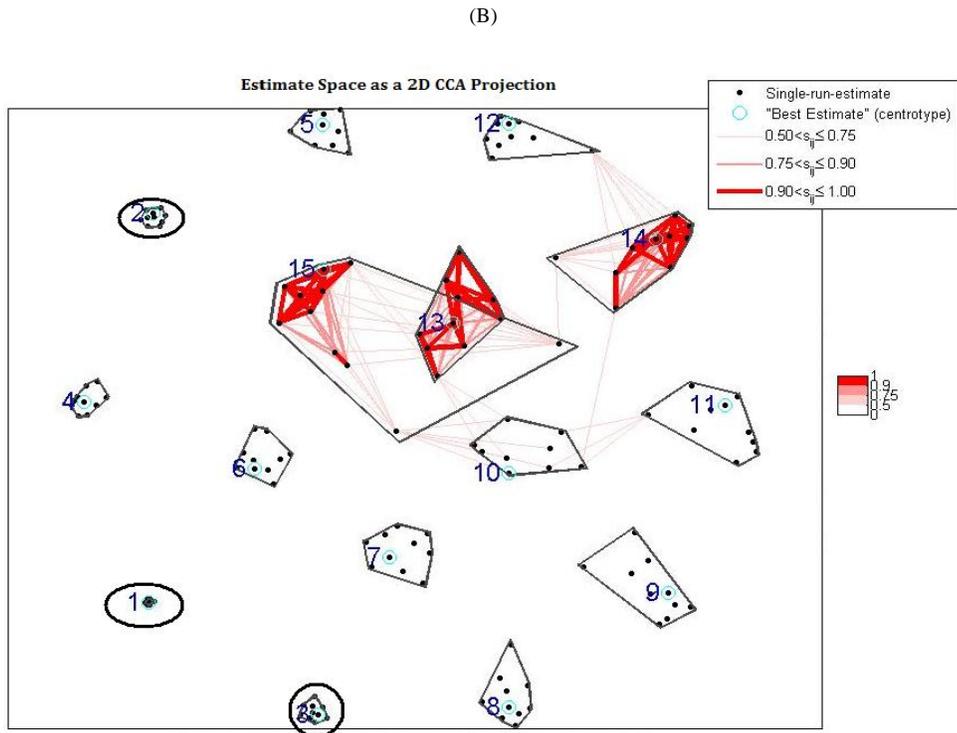
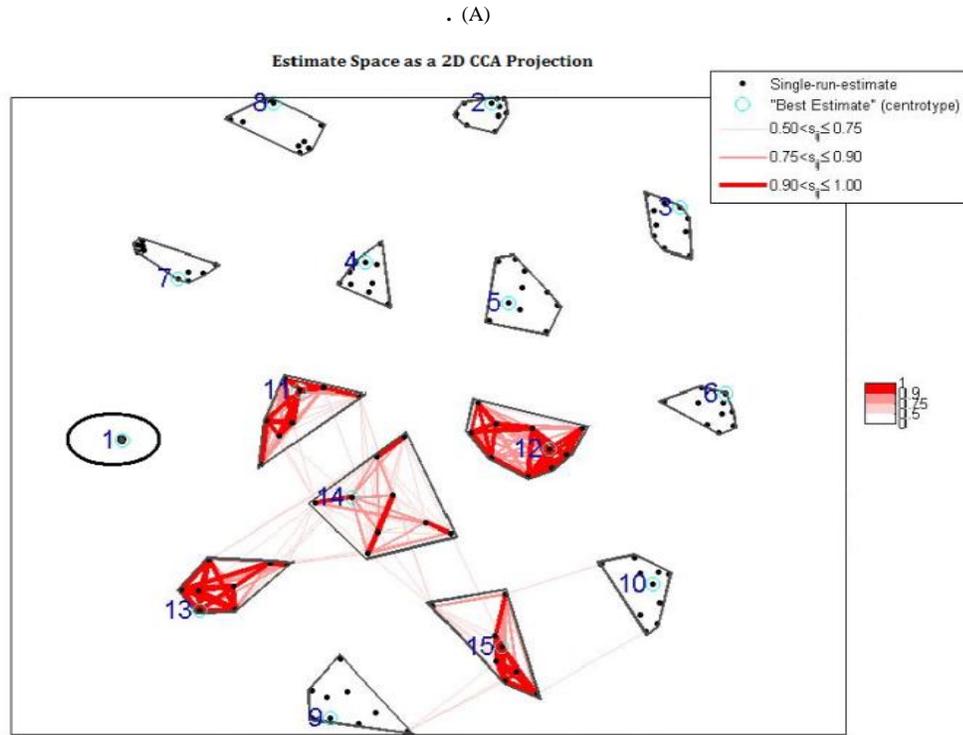
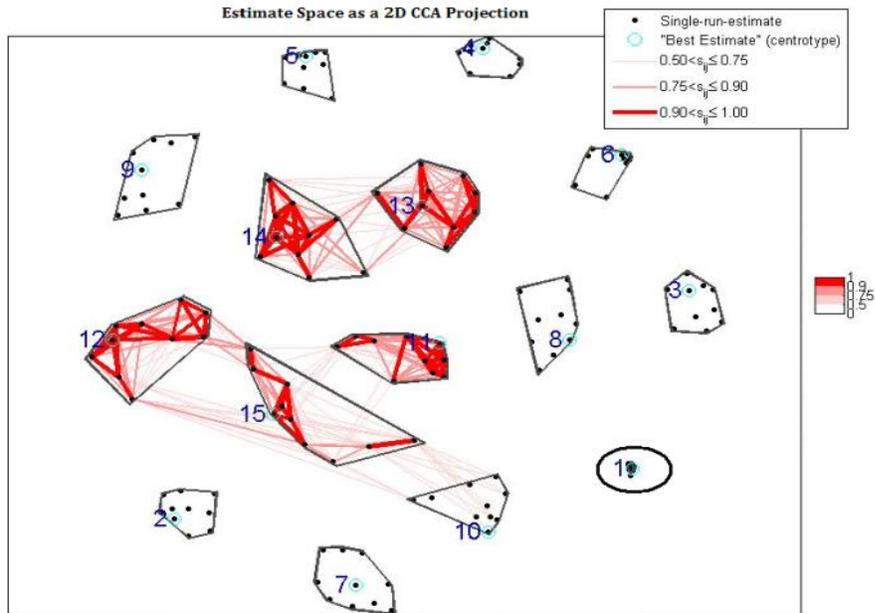


Figure 5.9: Cluster plots for (A) FastICA-*defl,tanh* (cluster 1 is ideal) and (B) FastICA-*symm,tanh* (clusters 1,2, 3 are ideal).

(A)



(B)

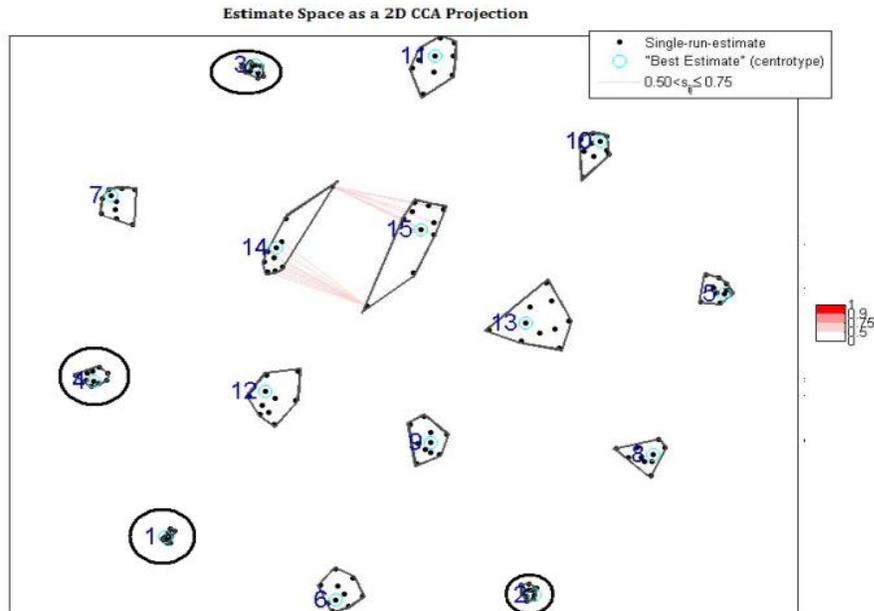


Figure 5.10: Cluster plots for (A) *FastICA-defl,pow3* (cluster 1 is ideal) and (B) *FastICA-symm,pow3* (clusters 1, 2, 3 and 4 are ideal)

5.5.2 Comparing Clusters using Different Linearity and orthogonalization

In my research I tested both BMICA and FastICA using the datasets described above. So far we have seen BMICA outperforming FastICA with a linearity of \tanh and a symmetric orthogonalization. Does BMICA outperform the other modes for FastICA? Below I show the results using a different dataset from the sets above. This was used to show how BMICA performs in different datasets with different linearity and modes.

Figure 5.7 through Figure 5.10 show the clusters for FastICA in both modes – deflation and symmetric, applying the four types of linearity – pow3 , \tanh , gauss and skew . Examination shows that FastICA (symm , pow3) has the best clustering, having 4 ideal clusters. This indicates that symmetric FastICA with pow3 linearity is the most reliable. Investigations also show that FastICA (symm , \tanh), and FastICA(defl , \tanh) had the least amount of similarities.

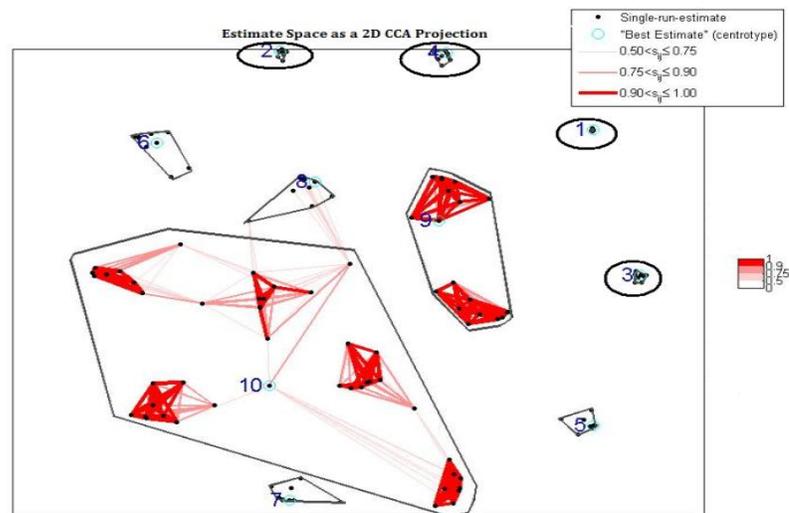


Figure 5.11: Cluster plots for BMICA (clusters 1, 2, 3, 4 are ideal)

Figure 5.11 shows the cluster results of BMICA for the same EEG data set. Comparison of both algorithms shows that both BMICA and FastICA (*symm, pow3*) have the same number of ideal clusters – 4. BMICA therefore produces more reliable components than 7 of the eight FastICA algorithms.

5.6 Summary

In this chapter I wanted to assess the reliability of the ICs produced from BMICA while comparing results to those of FastICA. After executing each algorithm many times and administering cluster analysis on the real world datasets, ICASSO showed that:

- (i) BMICA produced more ideal clusters
- (ii) BMICA produced more compact clusters
- (iii) Quality Index, I_q , for BMICA was closer to 1 for more clusters than FastICA (67% versus 30%) indicating that BMICA produced more clusters closer to the ideal which results from consistent thus more reliable ICs.

Cluster Analysis using ICASSO shows therefore that the new algorithm BMICA produces ICs which can be interpreted meaningfully. It also shows that FastICA with symmetric orthogonalization and exponential nonlinearity is the most reliable form of the FastICA algorithm, thus confirming the conclusions in [119]. This form is also the one most comparable with BMICA. One therefore deduces that BMICA is comparable to FastICA (all forms) when tested for reliability, however appearing more reliable.

Chapter 6 – MI Algorithms vs. Non MI Algorithms

6.1 Introduction

A number of analyses have been published [15, 46, 60, 113-121, 185] looking at different contrast function ICAs applied to biomedical signals such as EEG and ECG. Although this has been done there is nothing in literature which has performance comparison of MI-based methods with other contrast functions. In this chapter I evaluate the performance of seven ICA algorithms commonly used for biomedical analysis along with my newly created algorithm BMICA. The algorithms' performance when applied to EEG signals were analysed based on Noise/Signal and Separation Accuracy Measures as well as their convergence speed.

Analysis was done to see if the use of MI estimation produced better ICA algorithms as well as where BMICA follows within this spectrum.

6.2 Experiment Setup

As the need for accuracy in biomedical resources increase so does the need for improvement in the tools utilized. ICA offers a means to produce more accurate signals. Which category is best for biosignals? I tried to answer this question by looking at eight algorithms (based on their use of information maximization), designed for denoising these signals with focus on EEG signals using the datasets and performance measures described in Chapter 3. The algorithms are:

- (i) MI-based Methods – FastICA, Pearson ICA, Infomax and BMICA
- Non MI-based methods- JADE, SOBI, CuBICA and TDSEP.

I also compared the computational complexity of the two most popular algorithms from both categories – FastICA, Infomax, SOBI and JADE and examined the result patterns for each algorithm using the t-Test.

Table 6.1: SDR for 20 EEG Signals

Signal	BMICA	FASTICA	INFOMAX	PEAR	JADE	CUBICA	SOBI	TDSEP
1	-9.38E+01	-9.79E+01	-9.37E+01	-8.91E+01	-9.76E+01	-9.36E+01	-9.36E+01	-1.51E+00
2	-5.56E+01	-5.39E+01	-5.49E+01	-5.62E+01	-5.81E+01	-5.62E+01	-	2.55E+00
3	-5.03E+01	-5.09E+01	-5.11E+01	-	-4.95E+01	-4.92E+01	-2.45E-04	-1.91E+00
4	-8.47E+01	-8.67E+01	-7.79E+01	-8.36E+01	-8.65E+01	-8.50E+01	-8.50E+01	1.94E+00
5	-1.32E+02	-1.35E+02	-1.42E+02	-1.37E+02	-1.32E+02	-1.33E+02	-1.33E+02	1.29E+01
6	-3.33E+02	-3.37E+02	-3.55E+02	-3.36E+02	-3.40E+02	-3.36E+02	-3.36E+02	1.87E+01
7	-2.27E+02	-2.25E+02	-2.28E+02	-2.14E+02	-2.16E+02	-2.26E+02	-2.26E+02	2.06E+02
8	-5.41E+01	-5.90E+01	-5.86E+01	-5.79E+01	-5.53E+01	-5.65E+01	-5.65E+01	1.85E+00
9	-6.44E+01	-6.38E+01	-6.70E+01	-6.16E+01	-6.29E+01	-6.49E+01	-6.49E+01	3.88E+00
10	-6.53E+01	-6.45E+01	-6.67E+01	-6.46E+01	-6.29E+01	-6.49E+01	-6.49E+01	3.88E+00
11	-6.16E+01	-6.01E+01	-6.42E+01	-5.82E+01	-6.03E+01	-6.07E+01	-6.07E+01	-2.54E+00
12	-4.60E+01	-4.61E+01	-4.76E+01	-4.69E+01	-4.43E+01	-4.49E+01	-4.49E+01	1.92E+00
13	-9.38E+01	-9.79E+01	-9.37E+01	-8.91E+01	-9.76E+01	-9.36E+01	-9.36E+01	-1.51E+00
14	-5.56E+01	-5.39E+01	-5.49E+01	-5.62E+01	-5.81E+01	-5.62E+01	-	2.55E+00
15	-5.03E+01	-5.09E+01	-5.11E+01	-	-4.95E+01	-4.92E+01	-4.92E+01	-1.91E+00
16	-9.19E+00	-9.91E+00	-9.57E+00	-9.81E+00	-9.60E+00	-9.63E+00	-9.63E+00	-
17	8.40E+02	8.80E+02	8.88E+02	8.62E+02	8.20E+02	8.42E+02	8.42E+02	4.15E+01
18	4.46E+02	4.52E+02	4.81E+02	4.64E+02	4.51E+02	4.81E+02	4.81E+02	-1.31E+01
19	8.27E+02	7.72E+02	7.98E+02	8.10E+02	7.51E+02	7.76E+02	7.76E+02	8.95E+01
20	7.99E+02	6.91E+02	1.03E+03	9.28E+02	8.50E+02	9.08E+02	9.08E+02	-2.51E+01
Average	7.18E+01	6.52E+01	8.40E+01	9.47E+01	6.96E+01	7.63E+01	9.38E+01	1.79E+01

6.3 Results

6.3.1 Performance Measures

Examinations of both categories of algorithms show that the MI-based estimated algorithms produce higher SDRs on average such as in Table 6.1 MI-based algorithms average were 7.89E+01 compared to the others of 6.44E+01. This

indicates that the MI-based algorithms produced more accurate separations. Further examination shows that every non-MI-based algorithm had a MI-based algorithm which produced a better SDR e.g. Infomax 8.4E+01 vs. CubICA 7.63E+01; Pearson 9.47E+01 vs. SOBI 9.38E+01; FastICA 6.52E+01 vs. TDSEP 1.79E+01; and BMICA 7.18E+01 vs JADE 6.96E+01. This indicates that for every Non-MI algorithms tested there is a MI algorithm which produces a more accurate separation of signal and noise indicating that MI-based algorithms possess better overall separation performance.

Table 6.2: SIR for 20 EEG Signals

Signal	BMICA	FASTICA	INFOMAX	PEAR	JADE	CUBICA	SOBI	TDSEP
1	5.16E-01	5.31E-01	5.28E-01	5.38E-01	5.21E-01	3.62E-01	4.01E-01	4.33E-01
2	4.92E-01	4.85E-01	4.35E-01	5.71E-01	4.44E-01	4.52E-01	5.74E-01	4.89E-01
3	5.22E-01	5.11E-01	4.44E-01	5.71E-01	4.83E-01	2.50E-01	6.28E-01	5.71E-01
4	5.78E-01	5.22E-01	3.96E-01	5.24E-01	5.03E-01	3.96E-01	4.35E-01	4.81E-01
5	4.92E-01	3.60E-01	3.62E-01	4.04E-01	3.76E-01	3.49E-01	3.12E-01	3.32E-01
6	2.24E-01	3.34E-01	2.64E-01	3.34E-01	3.52E-01	3.25E-01	3.63E-01	6.63E-02
7	1.47E-01	2.93E-01	1.52E-01	2.01E-01	2.68E-01	2.49E-01	2.83E-01	1.87E-01
8	-2.39E-04	-2.56E-04	-2.65E-04	-2.46E-04	-2.60E-04	-2.73E-04	-2.42E-04	-1.54E-04
9	-2.61E-04	-2.48E-04	-2.55E-04	-2.51E-04	-2.45E-04	-2.58E-04	-2.47E-04	-2.26E-04
10	-2.61E-04	-2.59E-04	-2.54E-04	-2.51E-04	-2.45E-04	-2.58E-04	-2.47E-04	-2.26E-04
11	4.08E-01	4.48E-01	4.11E-01	7.08E-01	5.03E-01	3.33E-01	4.84E-01	3.02E-01
12	5.73E-01	5.63E-01	6.91E-01	6.00E-01	7.09E-01	3.65E-01	3.61E-01	2.13E-01
13	4.51E-01	2.96E-01	2.35E-01	2.87E-01	1.87E-01	1.66E-01	3.14E-01	4.03E-01
14	3.67E-01	3.72E-01	3.61E-01	3.67E-01	3.85E-01	3.55E-01	3.34E-01	0.00E+00
15	3.26E-01	3.53E-01	3.56E-01	3.53E-01	3.92E-01	4.02E-01	4.66E-01	0.00E+00
16	3.86E-01	4.33E-01	3.84E-01	3.77E-01	4.16E-01	4.51E-01	4.46E-01	0.00E+00
17	4.19E-01	4.64E-01	4.38E-01	4.10E-01	4.43E-01	4.67E-01	4.49E-01	2.50E-01
18	4.87E-01	4.31E-01	3.63E-01	5.15E-01	4.91E-01	4.87E-01	3.01E-01	6.16E-01
19	4.50E-01	4.65E-01	4.59E-01	4.68E-01	4.68E-01	5.44E-01	2.78E-01	5.11E-01
20	-3.14E-01	-2.34E-01	-8.41E-01	-3.51E-01	-1.12E-01	-1.33E-05	-2.00E-06	-2.59E+01
Average	3.42E-01	3.44E-01	3.14E-01	3.65E-01	3.46E-01	2.78E-01	3.37E-01	2.17E-01

Examination of Table 6.2 shows that the Non-MI algorithms produce smaller SIR results except for JADE which produces higher SIR than all four MI algorithms. This suggests that the Non-MI-based algorithms are better at estimating the sources thus producing better quality. All MI algorithms however produced SIR in a similar range.

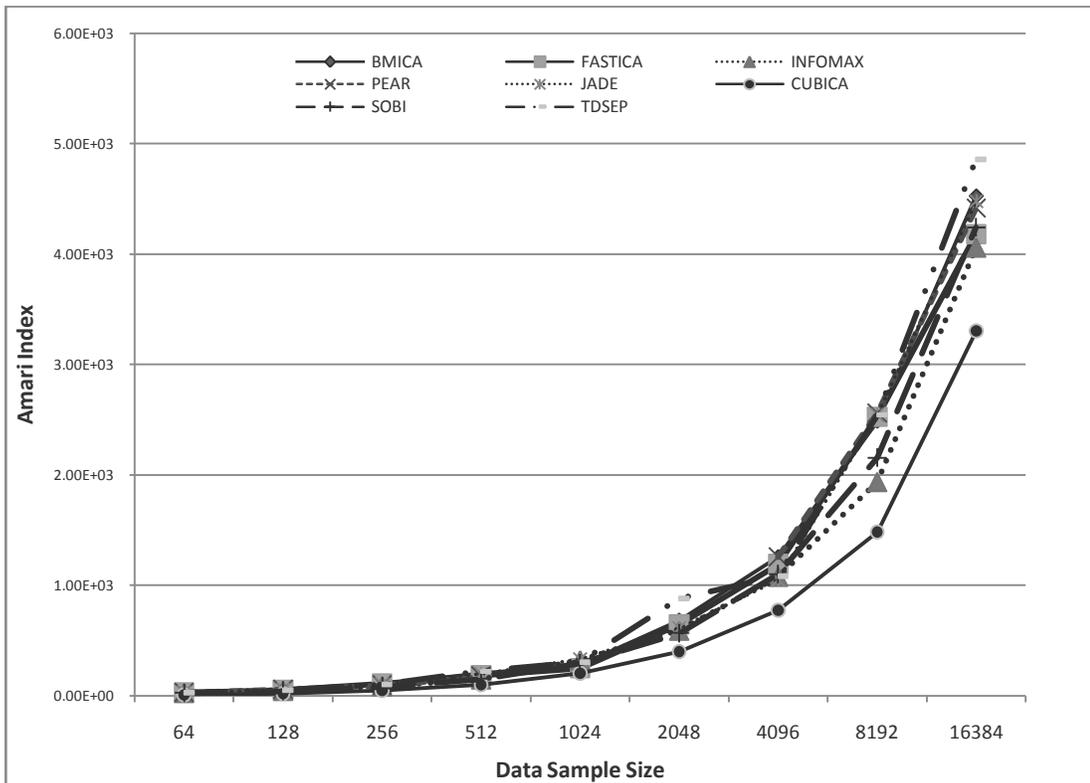


Figure 6.1: Amari Index for both MI and Non-MI Algorithms

How accurate is the estimated mixing matrix? Examination of the Amari Index answers this question. In the simulations it was found that the Amari indexes obtained for all algorithms were similar even when different sample sizes were used. This can be seen in Figure 6.1. This indicates that when the quality of A (determined from $W=A^{-1}$) is assessed all algorithms have similar performance i.e. how well the demixing transformation W agrees with the true mixing matrix A , for

all algorithms do not have any mark deviation from each other. Closer examination of Figure 6.1 shows however the MI-based algorithms were more consistent and closely knitted.

Table 6.3: MSE for 20 EEG Signals

Signal	BMICA	FASTICA	INFOMAX	PEAR	JADE	CUBICA	SOBI	TDSEP
1	7.20E+02	7.16E+02	7.04E+02	7.20E+02	7.56E+02	7.16E+02	7.27E+02	1.46E+05
2	6.29E+02	6.28E+02	5.78E+02	6.42E+02	6.10E+02	6.17E+02	6.20E+02	4.97E+05
3	8.61E+02	8.54E+02	8.14E+02	8.38E+02	8.89E+02	8.70E+02	8.72E+02	3.77E+05
4	1.11E+03	1.13E+03	1.05E+03	1.08E+03	1.08E+03	1.10E+03	1.09E+03	4.71E+05
5	1.03E+03	9.54E+02	9.34E+02	9.58E+02	1.00E+03	9.68E+02	9.79E+02	8.23E+05
6	7.11E+02	7.19E+02	7.02E+02	7.48E+02	7.56E+02	7.16E+02	7.27E+02	1.46E+05
7	6.65E+02	6.69E+02	6.82E+02	6.69E+02	6.69E+02	6.65E+02	6.65E+02	6.09E+05
8	7.15E+02	7.19E+02	6.97E+02	7.06E+02	7.23E+02	7.02E+02	7.16E+02	1.41E+05
9	1.16E+03	1.11E+03	1.09E+03	1.10E+03	1.08E+03	1.10E+03	1.11E+03	6.25E+05
10	1.06E+03	1.04E+03	1.01E+03	1.06E+03	1.02E+03	1.04E+03	1.04E+03	1.79E+06
11	7.37E+02	7.45E+02	7.53E+02	7.20E+02	7.21E+02	7.29E+02	7.36E+02	5.97E+05
12	7.57E+02	7.61E+02	7.39E+02	-	7.63E+02	7.64E+02	7.67E+02	4.04E+05
13	3.12E+03	3.09E+03	3.09E+03	3.13E+03	3.13E+03	3.12E+03	3.13E+03	1.84E+06
14	1.16E+03	1.18E+03	1.17E+03	1.17E+03	1.17E+03	1.17E+03	1.18E+03	0.00E+00
15	1.54E+03	1.54E+03	1.54E+03	1.54E+03	1.50E+03	1.53E+03	1.54E+03	0.00E+00
16	3.30E+05	0.00E+00						
17	7.62E+02	7.75E+02	7.53E+02	7.76E+02	7.71E+02	7.64E+02	7.67E+02	6.12E+05
18	4.38E+02	4.38E+02	4.37E+02	4.34E+02	4.28E+02	4.25E+02	4.27E+02	2.40E+05
19	5.88E+02	6.04E+02	6.04E+02	5.95E+02	6.03E+02	6.04E+02	5.95E+02	4.53E+05
20	5.87E+04	5.76E+04	6.61E+04	5.92E+04	5.60E+04	5.46E+04	5.46E+04	2.13E+07
Average	2.16E+04	2.16E+04	2.16E+04	2.30E+04	2.16E+04	2.16E+04	2.16E+04	5.30E+04

The experiment examines the quality of the different categories of the estimators for ICA algorithms so to determine the quality of the MSE of each algorithm must be known. Examination of simulations shows that the MSE for all algorithms were similar on average. Table 6.3 shows the results of a number of simulations; here the overall average is 2.16e+04 except for Pearson ICA, which did not compute for one signal, and TDSEP. On closer examination however the Non-

MI algorithms tend to produce higher MSE approximately 50% of the time. This indicates that the MI-based estimator was of better quality in terms of variation and unbiasedness.

Table 6.4: SNR for 20 EEG Signals

Signal	BMICA	FASTICA	INFOMAX	PEAR	JADE	CUBICA	SOBI	TDSEP
1	-4.98E-02	2.03E+02	1.90E-01	-9.06E-02	5.58E-02	2.01E-02	3.59E-04	-3.37E+01
2	-1.96E-02	-9.22E-02	-3.79E-02	1.13E-01	7.72E-02	-3.79E-02	6.73E-04	-2.97E+01
3	7.92E-02	9.27E-02	1.89E-01	-	2.82E-02	2.28E-02	-2.45E-04	-1.91E+00
4	-2.24E-02	-2.30E-03	-1.38E-01	-2.52E-02	7.50E-03	-9.61E-05	-7.82E-05	-3.18E+01
5	-3.96E-02	-1.01E-01	-1.27E-02	5.88E-02	1.20E-03	8.65E-02	2.96E-04	-2.15E+01
6	-1.58E-02	-4.70E-03	5.33E-02	2.20E-02	1.02E-01	5.56E-02	3.50E-03	-2.80E+01
7	-5.79E-02	1.10E-01	2.77E-01	2.21E-02	2.73E-02	-7.09E-02	2.00E-03	-2.62E+01
8	-2.24E-01	1.77E-01	2.94E-01	2.94E-01	-8.15E-02	5.35E-02	2.93E-04	2.94E-01
9	9.19E-02	5.51E-02	1.72E-01	1.59E-01	-1.66E-01	8.63E-02	6.33E-04	1.72E-01
10	-8.30E-02	-1.66E-01	1.58E-01	3.11E-02	3.55E-02	8.63E-02	6.33E-04	-2.26E+01
11	4.08E-01	-2.06E-02	-	1.24E-01	6.05E-02	3.59E-01	9.08E-04	-2.95E+01
12	7.41E-01	8.10E-02	3.46E-01	1.81E-01	-7.15E-02	2.55E-02	7.26E-04	-2.66E+01
13	9.80E-03	7.59E-02	7.59E-02	-2.53E-02	-2.81E-02	2.29E-02	3.36E-04	-2.85E+01
14	-4.72E-02	-5.00E-03	1.83E-02	1.34E-02	1.75E-02	-6.70E-03	-1.25E-04	0.00E+00
15	2.42E-02	-4.60E-03	-8.00E-03	-2.22E-02	7.60E-03	2.42E-02	3.05E-05	0.00E+00
16	5.44E-04	-1.94E-04	2.22E-04	-2.28E-05	5.89E-04	-1.50E-05	2.30E-08	0.00E+00
17	-2.54E-02	4.64E-01	4.38E-01	4.10E-01	-8.70E-03	-5.06E-02	2.30E-03	-3.18E+01
18	-1.40E-01	-1.03E-01	-5.16E-02	-1.67E-01	4.58E-03	7.93E-02	9.12E-04	-2.92E+01
19	7.16E-02	-8.91E-02	-2.40E-01	2.15E-02	-5.76E-02	-3.24E-02	5.13E-04	-3.09E+01
20	-3.14E-01	-2.34E-01	-8.41E-01	-3.51E-01	-1.12E-01	-1.33E-05	-2.00E-06	-2.59E+01
Average	3.69E-02	2.59E-02	9.56E-02	6.22E-02	6.30E-04	3.81E-02	7.19E-04	-1.95E+01

I have examined the overall separation performance; the quality of the estimated mixing matrix, estimator and source signal. How is the relationship between the signal and noise? SNR and PSNR answer this question. Examination of Table 6.4 shows the results of 20 SNR for all algorithms. I found that TDSEP produces the lowest SNR. Further examination showed that although CuBICA had the most consistent line the MI algorithms especially BMICA and Infomax produced the

highest SNR. This indicates that MI-based algorithms after denoising produce signals that are only slightly distorted by noise as well as unmixing matrix that is almost not affected by noise.

Table 6.5: PSNR for 20 EEG Signals

Signal	BMICA	FASTICA	INFOMAX	PEAR	JADE	CUBICA	SOBI	TDSEP
1	1.79E+01	1.80E+01	1.81E+01	1.79E+01	1.80E+01	1.80E+01	1.80E+01	-3.51E+00
2	1.95E+01	1.94E+01	1.94E+01	1.96E+01	1.96E+01	1.95E+01	1.95E+01	-8.83E+00
3	1.93E+01	1.93E+01	1.94E+01		1.93E+01	1.93E+01	1.93E+01	-7.63E+00
4	1.99E+01	1.99E+01	1.98E+01	1.99E+01	1.99E+01	1.99E+01	1.99E+01	-8.60E+00
5	1.96E+01	1.96E+01	1.97E+01	1.96E+01	1.95E+01	1.97E+01	1.96E+01	-1.10E+01
6	1.77E+01	1.77E+01	1.78E+01	1.77E+01	1.78E+01	1.77E+01	1.77E+01	-3.51E+00
7	1.77E+01	1.76E+01	1.79E+01	1.78E+01	1.78E+01	1.77E+01	1.77E+01	-9.72E+00
8	1.80E+01	1.83E+01	1.84E+01	1.83E+01	1.81E+01	1.83E+01	1.82E+01	-3.36E+00
9	1.96E+01	1.96E+01	1.97E+01	1.94E+01	1.93E+01	1.96E+01	1.95E+01	-9.83E+00
10	1.96E+01	1.96E+01	1.97E+01	1.96E+01	1.93E+01	1.96E+01	1.95E+01	-1.44E+01
11	2.01E+01	2.02E+01	2.05E+01	2.01E+01	2.03E+01	2.02E+01	2.01E+01	-9.63E+00
12	1.88E+01	1.88E+01	1.90E+01	1.89E+01	1.86E+01	1.87E+01	1.87E+01	-7.93E+00
13	1.32E+01	-1.45E+01						
14	1.74E+01	1.74E+01	1.74E+01	1.74E+01	1.75E+01	1.74E+01	1.74E+01	1.74E+01
15	1.62E+01	1.63E+01	1.63E+01	1.62E+01	1.62E+01	1.63E+01	1.62E+01	1.61E+01
16	-7.05E+00	-7.09E+00						
17	1.93E+01	1.93E+01	1.94E+01	1.92E+01	1.93E+01	1.93E+01	1.93E+01	-9.74E+00
18	2.17E+01	2.17E+01	2.17E+01	2.18E+01	2.18E+01	2.18E+01	2.18E+01	-5.67E+00
19	2.04E+01	2.03E+01	2.03E+01	2.04E+01	2.03E+01	2.03E+01	2.04E+01	8.43E+00
20	4.47E-01	5.28E-01	-7.30E-02	4.10E-01	6.46E-01	7.56E-01	7.56E-01	-2.51E+01
Average	1.73E+01	1.73E+01	1.74E+01	1.72E+01	1.73E+01	1.73E+01	1.73E+01	-5.91E+00

Investigations into PSNR calculations, as seen in Table 6.5, show that most of the algorithms appeared to perform within similar ranges except for TDSEP for most signals. The table shows that five of algorithms generate an average of 1.73E+01 for their PSNR with two others with only ± 0.01 difference. This indicates that for seven of the eight algorithms the reconstruction of the signal was of a high quality resulting in algorithms considered to be good.

6.3.2 T-Test Investigations

From section 6.3.1 it can be seen that TDSEP has the overall worst performance. Did this have anything to do with the chosen datasets? Was the population behind TDSEP dataset different from the other algorithms to allow for the difference seen? I applied the t-test to answer these questions. A t-test can be performed to see if the population means behind two data sets (samples) are similar enough to conclude that they could have come from the same population. The test was used to test the following competing hypotheses:

H1 (Test Hypothesis): The population means behind the two samples are different.

H0 (Null Hypothesis): The population means behind the two samples are the same.

For every test the chosen level of significance, α is 0.05, (standard for biological tests) which corresponds to a level of confidence of 95% and a probability of 1 in 20. The number of distribution tail is 2, the sample groups have equal variance, the determined degree of freedom, df is 38 and the critical t-value collected from the t-value table at <http://www.stattools.net/tTest Tab.php> is 2.024. Samples were based on the values displayed in tables 6.1 though to 6.5. Below the example utilizes the BMICA and TDSEP values from Table 6.5

	BMICA	TDSEP
Mean	16.465	-4.583
Standard Deviation	7.096	10.654
Standard Error	1.588	4.027
t Stat	8.196	
p (two tail)	3.518E-6	

Here the computed t-value is greater than the critical t-value. It can also be seen that

the p-value is less than α . I can therefore reject the null hypothesis and conclude that:

- the population means behind the two samples are different because there is a small chance (1 in 20) that the population means are the same
- the samples are not “equivalent”
- the experiment is not repeatable

The pattern appears no matter the sample algorithm utilized with TDSEP as when compared to FastICA the two-tailed p-value is less than 0.0001 and the computed t-value is 8.2064. The result is the same. The pattern is similar throughout all tables e.g. in Table 6.1 when BMICA and TDSEP are tested the two-tailed p-value is 0.513 and the calculated t-value is -2.972. The computed t-value is still larger than the critical t-value.

The t-test for the other algorithms however produced differences not considered to be statistically significant. For example when BMICA is compared with SOBI the two-tailed p-value is 0.9928, and calculated t-value is 0.0091. It is found that the computed t-value is less than the critical t-value. I therefore, based on these results, fail to reject the null hypothesis (H_0) that the population means behind the two samples are the same as I could not conclude that

- the population means are different
- the samples are not “equivalent”
- the experiment is not repeatable

6.3.3 Computational Complexity

I investigated the computational complexity of four algorithms. Using N as the number of sources and M the sample size investigations have found the following: FastICA and Infomax, both have a complexity on the order of $O(N^3M)$ [137], while JADE [137] and SOBI [158] algorithms are on the order of $O(N^4M)$.

Investigations of the above named four algorithms show that FastICA’s convergence is quadratic in general and cubic for symmetric distributions [58]. The

JADE algorithm uses Givens rotations, the convergence of which is known to be cubic [165]. Infomax is gradient algorithm with non-adaptive step sizes and converges linearly [165]. SOBI also converges linearly.

6.4 Discussion

Examination of the results for the three noise/signal measure tools show that although all except TDSEP has similar PSNR showing that they all produce similar reconstructions. This was further supported by the t-test results. For the quality of the estimator utilized the MI based algorithms produce better results i.e. producing lower MSE indicating that they produce results closer to the actual data. MI algorithms produced higher SNRs indicating also that they are better at separating noise from the signals. So are MI algorithms better at denoising than Non-MI algorithms? Examination of the separation accuracy measure tools helps to answer this question.

When the accuracy of the separations was examined it was found that there was no consistency in the conclusion as in the other measurement tools. I found that the Non-MI algorithms achieves better separations i.e. quality of the estimated sources were better as showed in the SIR results however these separations were not the most accurate indicated by the SDR results. The Amari Index assesses the accuracy of the estimated mixing matrix and for all algorithms the performances were similar, indicating that this tool did not show much difference in the performance of the estimators used.

Although the ultimate goal of a signal separation approach is the quality of such a separation, reflected on the estimated source signals, it is interesting to relate the various ICA approaches from a numeral complexity viewpoint. When the computational complexity is compared it is found that MI class of algorithms produce smaller complexity. This indicates that these algorithms compute quicker. As far as convergence speed is concerned MI-based algorithms were both linear and

cubic just as the Non MI-based. This indicates that convergence speed would not be a strong deciding factor in ICA choices.

6.5 Summary

Researchers over the years have always claimed that MI estimators produced more accurate ICA algorithms. In this chapter I evaluated the performance of eight ICA algorithms. It was observed that MI algorithms – BMICA, FastICA, Infomax, and Pearson_ICA

- (i) were better at separating noise from the EEG signals and
- (ii) produced the most accurate EEG signals.

This showed that my experiments supported the claim of other researchers – MI estimation produces better ICA algorithms. So which category is better for biosignals denoising? I conclude – MI-based algorithms.

Chapter 7 – Unscented Kalman Filter

7.1 Introduction

The recognition and elimination of noise attached to EEG signals is complicated but essential to patients' diagnosis and several separation techniques have been employed including mimetic, template matching, rule –base and predictive filtering.

Cohen *et al.* [22] in their paper suggested that EEG epochs can be stationary, EEG signals, however, are not usually stationary, exhibiting marked patterns of changes over time in frequency structure and amplitudes. This influences the model and method used to analyze the signals. Research has shown the use of time-varying autoregressive moving average (ARMA) model [157], and time-varying parameter autoregressive (TVAR) model [176] have been used to analyze EEG signals. In the TVAR model, adaptive algorithms such as the Kalman filter (KF) has been utilized [144]. Although it has been proven that the Kalman filter is more effective than a stationary filter such as the Wiener filter [116], it is limited by its assumptions and its application to linear systems. EEG is non linear. This leads to the development of suboptimal filters of which this research utilized two – Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). Both are described in Chapter 3.

7.2 EEG, EKF and UKF

My research sought to investigate the performance of the EKF and UKF when applied to EEG noise reduction. According to recent developments, UKF is an interesting alternative to the EKF for nonlinear systems, since it has a higher accuracy [65, 172]. Noise reduction in EEG was previously done using KF [7] as well as EKF [135]. Although several papers have investigated the accuracy of UKF for nonlinear, nonstationary systems none has addressed the accuracy when applied

to EEG. My research bridges this gap by comparing the performance of EKF to UKF when applied to EEG signals. Although both algorithms have time and measurement updates examination of UKF in Algorithm 5 and EKF in Algorithm 4 showed that UKF requires more computation. Does this mean that UKF will perform better than EKF when denoising EEG signals? I sought to answer this question in the rest of the chapter.

Initialization mean $x_0 = E[x_0]$ and covariance matrix

$$P_0 = E \left[(x_0 - \hat{x})(x_0 - \hat{x})^T \right] \quad (7.1)$$

Time Update:

$$X_{k|k-1} = F X_{k-1}, u(k) \quad (7.2)$$

$$P_k^- = A_k P_k A_k^T + Q_k \quad (7.3)$$

Measurement Update:

$$\kappa_k = P_k^- H_k^T [H_k P_k^- H_k^T + R_k]^{-1} \quad (7.4)$$

$$\hat{x}_k = \hat{x}_k^- + \kappa_k (y_k - h(\hat{x}_k^-, w)) \quad (7.5)$$

$$P_k = (I - \kappa_k H_k) P_k^- \quad (7.6)$$

where Q_w =process noise cov., R_v =measurement noise cov.,

$$A_k = \left. \frac{\partial F(x, w)}{\partial x} \right|_{\hat{x}_k} \quad H_k = \left. \frac{\partial h(x, w)}{\partial x} \right|_{\hat{x}_k} \quad (7.7)$$

Algorithm 4: Extended Kalman Filter

Initialization mean $x_0 = E[x_0]$ and covariance matrix

$$P_0 = E[(x_0 - \hat{x})(x_0 - \hat{x})^T] \quad (7.8)$$

Calculate the sigma points for $1, 2, \dots, k = n$

$$X_{k-1} = \begin{bmatrix} \hat{x}_{k-1} & \hat{x}_{k-1} + \sqrt{(L+\lambda)P_{k-1}} & \hat{x}_{k-1} - \sqrt{(L+\lambda)P_{k-1}} \end{bmatrix} \quad (7.9)$$

Time Update:

$$X_{k|k-1} = F X_{k-1}, u(k) \quad (7.10)$$

$$Y_{k|k-1} = H [X_{k|k-1}, u(k)] \quad (7.11)$$

$$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^m Y_{i,k|k-1} \quad \hat{x}_k = \sum_{i=0}^{2L} W_i^m X_{i,k|k-1} \quad (7.12)$$

$$P_k^- = \sum_{i=0}^{2L} W_i^c [X_{i,k|k-1} - \hat{x}_k^-] [X_{i,k|k-1} - \hat{x}_k^-]^T + Q_w \quad (7.13)$$

Measurement Update:

$$P_{y_k y_k}^- = \sum_{i=0}^{2L} W_i^c [Y_{i,k|k-1} - \hat{y}_k^-] [Y_{i,k|k-1} - \hat{y}_k^-]^T \quad (7.14)$$

$$P_{x_k y_k} = \sum_{i=0}^{2L} W_i^c [X_{i,k|k-1} - \hat{x}_k^-] [Y_{i,k|k-1} - \hat{y}_k^-]^T + R_v \quad (7.15)$$

$$K = P_{x_k y_k} P_{y_k y_k}^{-1} \quad (7.16)$$

$$\hat{x}_k = \hat{x}_k^- + K(y_k - \hat{y}_k^-) \quad P_k = P_k^- - K P_{y_k y_k}^- K^T \quad (7.17)$$

where Q_w =process noise cov., R_v =measurement noise cov.

Algorithm 5: UKF Algorithm

7.3 Experiment

7.3.1 Chosen Model for Filters

The EEG analysis problem consists of estimating a possibly dynamic state of a nonlinear stochastic system, based on a set of noisy observations. It can therefore be written in the form of the so-called dynamic state space (DSS) model. We presented my EEG data using this model. These models are usually characterized with additive Gaussian noises, but the functions that appear in the process and observation equations are nonlinear functions. The general DDS model can be written in two basic equations – process and observation as seen below [86-87]:

$$\begin{aligned}x_n &= f(x_{n-1}) + u_n && \text{(process equation)} \\y_n &= h(x_n) + v_n && \text{(observation equation)}\end{aligned}\tag{7.18}$$

where u_n and v_n are random noise vectors. I carried out my experiments using a special type of DSS model called the Univariate Nonstationary Growth Model (UNGM.) We choose this model because it is highly nonlinear in both process and observation equations and is bimodal in nature. This model's DSS equations are:

$$\begin{aligned}x_n &= \alpha x_{n-1} + \beta \frac{x_{n-1}}{1 + x_{n-1}^2} + \gamma \cos(1.2(n-1)) + u_n \\y_n &= \frac{x_n^2}{20} + v_n, \quad n = 1, \dots, N\end{aligned}\tag{7.19}$$

For my experiment to be executed in UNGM we assumed the following:

$$x_0 = 0.1, \quad P_0 = 1, \quad \alpha=0.5, \quad \beta=25 \quad \gamma=8.$$

7.3.2 Filter Configuration

As described in Chapter 3 there are four types of filter configurations – State, Parameter, Joint and Dual. For my research we utilized the State configuration since UKF was originally designed for this estimation [172]. Although many filters have already been investigated in this configuration [59], there is none for EEG investigations.

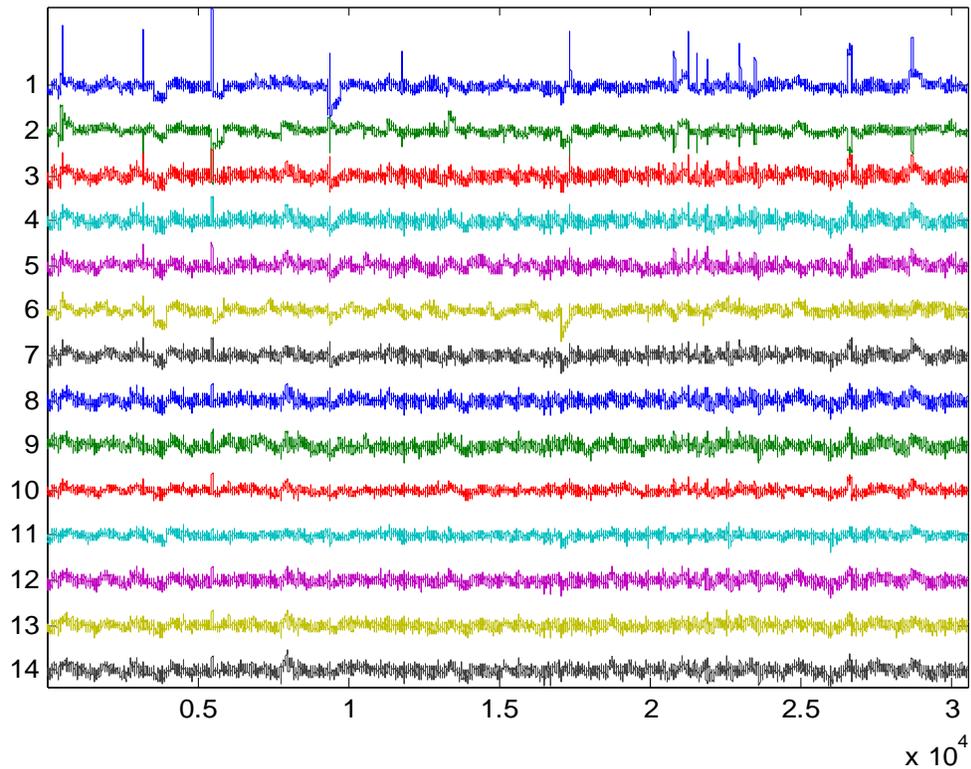


Figure 7.1: First 14 signals from the Data Set used

7.4 Results

All experiments were conducted using the Laptop environment 1 (MATLAB 7.8.0 (R2009)) on a laptop with AMD Athlon 64x2 Dual-core Processor 1.80GHz) and all results displayed are based on dataset 1 which was collected from the Swartz Center for Computational Neuroscience (SCCN). The dataset contains 32 EEG signals of which Figure 7.1 shows the first 14. This figure shows the dataset before filtering is performed. Each of the 32 signals (channels) contains 30,504 values i.e. the number of times data was collected at the electrode location.

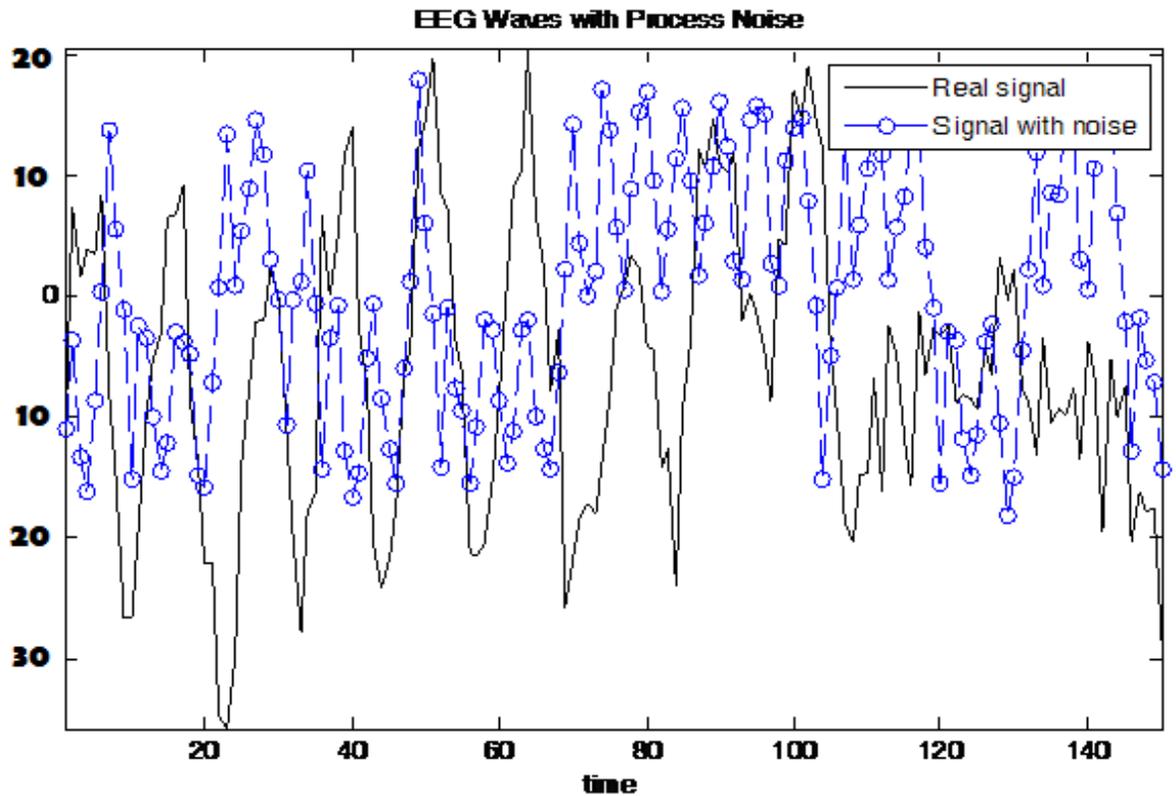
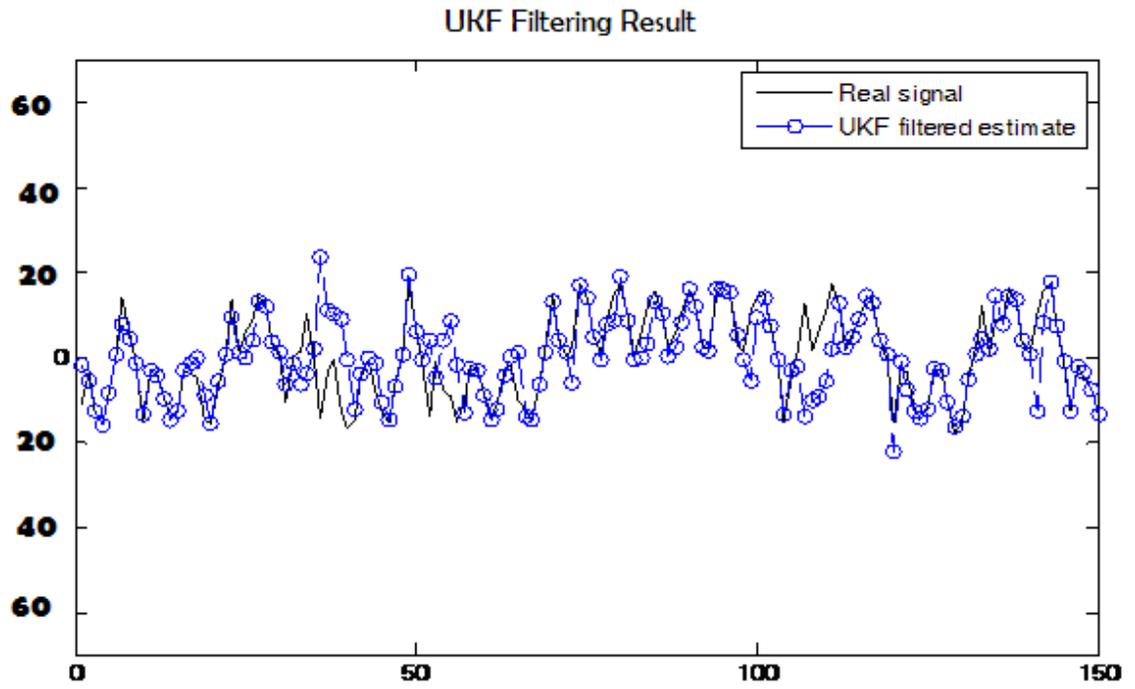


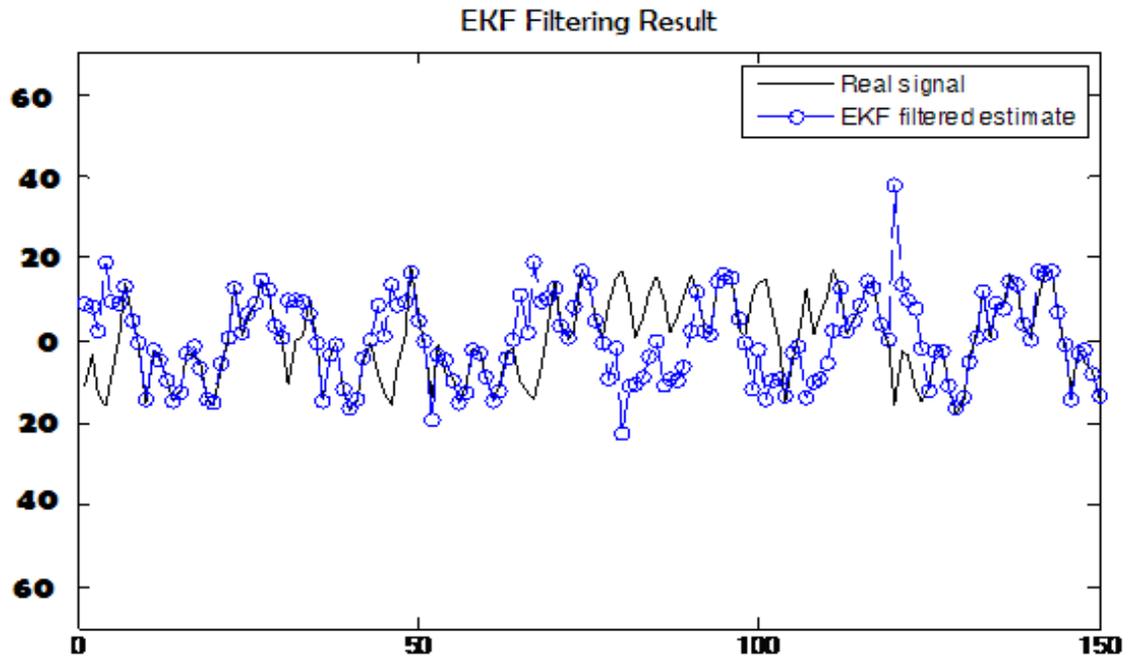
Figure 7.2: Channel 32 showing first 150 values with and without noise

To each signal in the dataset we added noise. Figure 7.2 shows the first 150 values of signal (channel) 32, reflecting the original signal and the signal after added noise. Once noise was added the dataset was then filtered by both UKF and EKF. Figure 7.3 shows the filtered estimate of the signal shown in Figure 7.2.

(A)



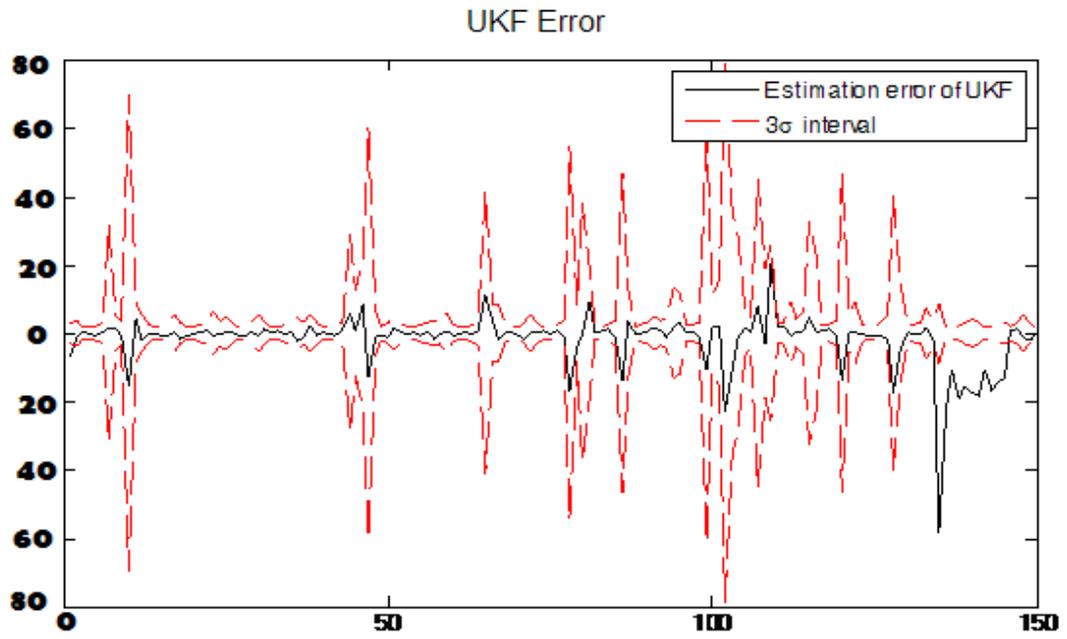
(B)



.ll.

Figure 7.3: True State of Signal and Estimates for (A) UKF (B) EKF

(A)



(B)

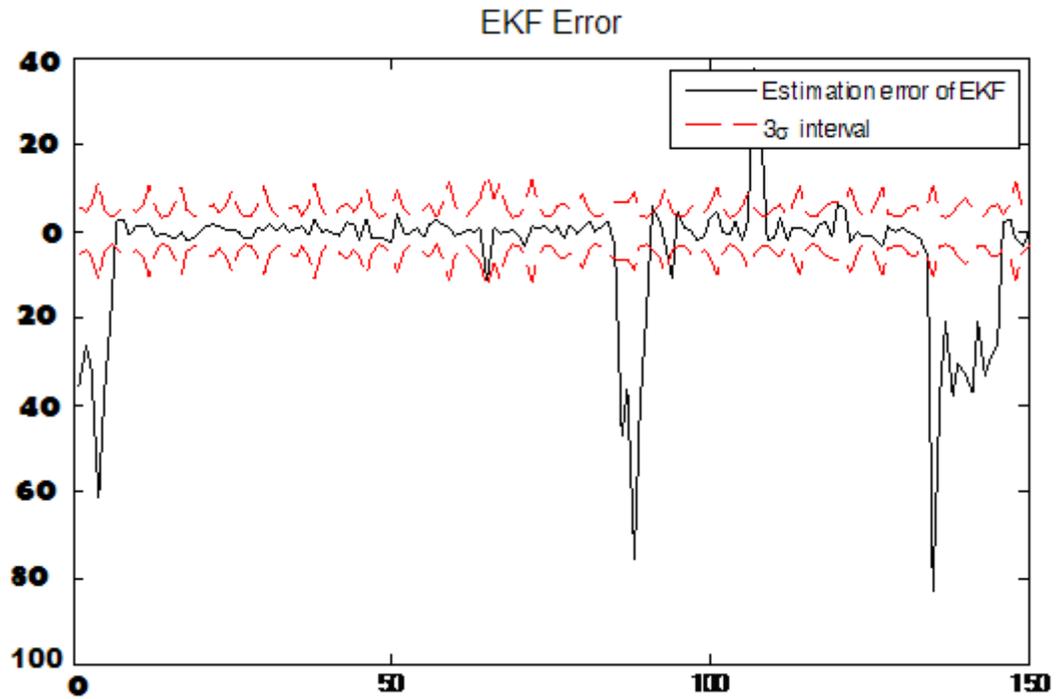


Figure 7.4: Estimation Errors and 3- σ confidence intervals for (A) UKF (B) EKF

7.5 Discussions

Examination of Figure 7.3 reveals that when filtered using UKF the resulting signals is very close to the original in all points unlike EKF which although reduced the noise level produced

- (i) signal values below the original
- (ii) signal values greater than the original

indicating that it either removed some of the original signal (i) or left too much noise (ii). EKF estimation seems to have a poorer performance. The performance of both is confirmed in Figure 7.4 when the estimated error and 3- σ intervals were determined. These are calculated once the updated state covariance for each signal was determined. It is shown that the UKF performs noticeably better, i.e., more accurate and more consistent. The behaviour of UKF maybe due to the higher degree of accuracy of the mean and covariance estimates.

Table 7.1: MSE Calculation for 8 EEG Signals(Channels)

Channel	UKF MSE	EKF MSE
4	62.4363	113.7740
8	64.7571	121.1993
12	59.9900	123.1414
16	61.7373	116.5613
20	59.8916	121.2223
24	63.0880	117.5134
28	58.3766	108.9822
32	59.9209	120.5497

Although the graphs show that UKF estimates are nearer to the original signal than EKF, to determine how well the EKF and UKF algorithms are performing, we need comparison data. This data was determined by means of the Mean Square Error (MSE), defined in Chapter 3 which is used to measure the optimality of the filters.

Table 7.1 shows the estimates of UKF and EKF for eight channels. Examination shows that the MSE for UKF is lower than that of EKF. Figure 7.5 confirms this UKF's MSE is nearly one times smaller than that of EKF. Error performance for UKF is superior to the EKF. This supports the findings which state that UKF has improved the error performance of the EKF in state as well as parameter estimations [66, 171-172].

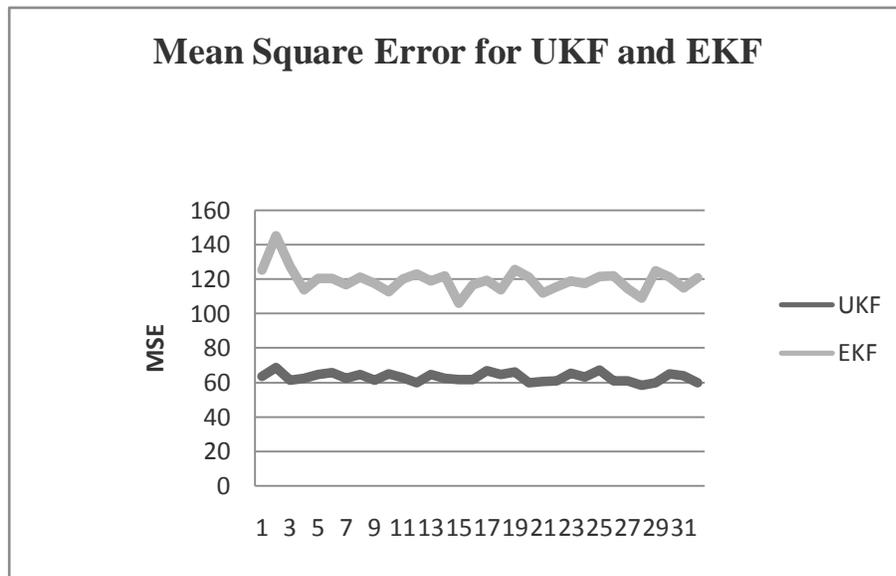


Figure 7.5: Performance Comparison of the UKF and EKF filters. The MSEs across 32 runs

Comparison with MSE is not the only comparison; we recorded the running times for each algorithm. On average, the EKF algorithm took approximately 9 seconds per estimate while the UKF algorithm took approximately 33 seconds per

estimate. The reason the UKF algorithm takes significantly longer to make an estimate is because it has to handle all the sigma points using the unscented transformation. Since the estimation accuracy of the UKF was found to be better than the EKF, this additional computational overhead is warranted.

7.6 Summary

Many researchers have found that UKF performs better than EKF for non-linear signals [26, 76, 84, 154]. Rohál'ová *et al.* [135] went on in their experiment to show that EKF is also very good at filtering noise from EEG Signals. When comparison of UKF and EKF was done on simpler signals in the form of ECG it was found that UKF outperforms EKF, particularly around the sharp turning points of the signal [140].

While the results were achieved with artificially generated noise, without any specific assumption on the noise origin, the test shows that UKF outperforms EKF. This also supports all the researches already conducted.

Chapter 8 – Improving Translation Invariant Wavelet Transform

8.1 Introduction

Research has shown that when denoising EEG signals using Wavelet Transform it has been based on Discrete Wavelet Transform (DWT) [139] and Stationary Wavelet Transform (SWT) [139]. Research also shows that Wavelet Transform is best suited for denoising as far as performance goes because of its properties like sparsity, multiresolution and multiscale nature. Non-orthogonal wavelets such as UDWT and Multiwavelets improve the performance at the expense of a large overhead in their computation [117].

In recent years researchers have used both ICA algorithms and WT to denoise EEG signals. In this dissertation I draw attention to Coifman and Donoho's Translation invariant Wavelet Transform and its application to denoising EEG signals. There has been none done using the Translation-Invariant (TI) form of wavelet transform (proposed by Coifman and Donoho [21]), in denoising these signals. In this chapter I sought to

- (i) denoise EEG using TIWT
- (ii) improve TIWT performance by
 - a. mergering with UKF and an ICA and
 - b. mergering with my algorithm, BMICA.

8.2 Translation Invariant Algorithm

In Chapter 3 it was noted that TI was created to improve the limitations of normal DWT and SWT. It calls for the cycle spinning technique and results in the following algorithm:

Signal Collection

This algorithm is designed to denoise both natural and artificially noised signals. They should therefore be mathematically defined based on Eq. (3.3).

Apply translation invariant forward wavelet transform to signal

- a. The number of time shifts is determined; in so doing signals are forcibly shifted so that their features change positions removing the undesirable oscillations which result in pseudo-Gibbs phenomena. The circulant shift by h is defined as:

$$S_h f(n) = f(n+h) \bmod N \quad (8.1)$$

where $f(n)$ is the signal, S is time shift operator and N is the number of signals. The time-shift operator S is unitary and therefore invertible i.e. $(S_h)^{-1} = S_{-h}$

- b. The signals are decomposed into different levels of DWT using the chosen wavelet function – Sym8 to separate noise and true signals.
 - Symmlets are orthogonal and its regularity increases with the increase in the number of moments [32]. Symmlets are also capable of perfect reconstruction [45] which allows for all signals to be reconstituted for analysis.
 - Since the number of vanishing moments determines the order of the polynomial that can be approximated and is useful for compression purposes I had to choose the most effective for my algorithm. Literature has shown that Symmlets for 1-3 are the same as the Daubechies wavelets of the same order [45] thus not giving the desired symmetry. After experiments using vanishing moments 4 – 14, I found that Sym8 was better as it reconstructed signals with less time, and showed maximum response for random noise for both hard and soft thresholds.

Choose and Apply Threshold Value

Denoise using the soft-thresholding method discarding all coefficients below the threshold value using VisuShrink based on the universal threshold defined by Donoho & Johnstone [29] given as:

$$T = \sqrt{2\sigma^2 \log N} \quad (8.2)$$

where N is the number of samples and σ^2 is the noise power.

Reconstruction of denoised EEG signal using inverse translation invariant forward wavelet transform

Revert signals to their original time shift and average the results obtained to produce the denoised EEG signals. The proposed algorithm based on time shifts can be expressed as $Avg [Shift - Denoise - Unshift]$ i.e. using Eq. (8.1) it is defined as:

$$avg_{h \in H} S_{-h} T S_h (f) \quad (8.3)$$

where H is the range of shifts, T is the wavelet shrinkage denoising operator, h the circular shift and the maximum of H is the length of the signal N from Eq. (8.1).

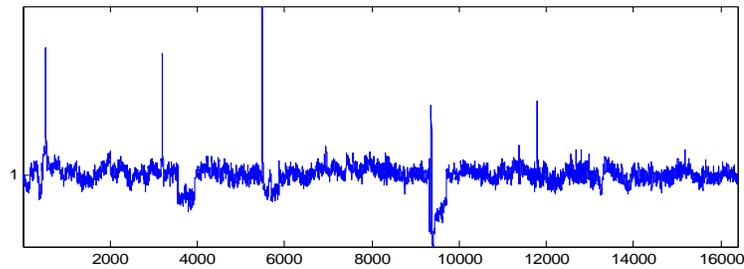
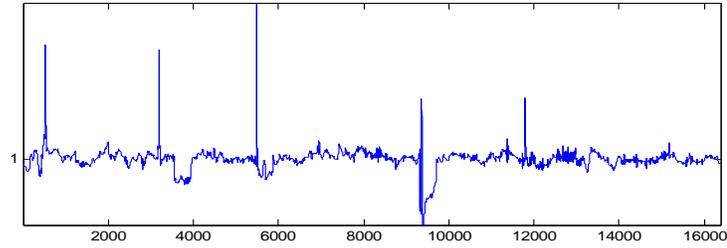
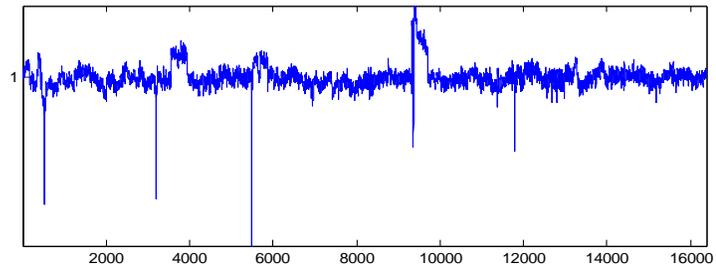


Figure 8.1: EEG signal contaminated with EOG

(A)



(B)



(C)

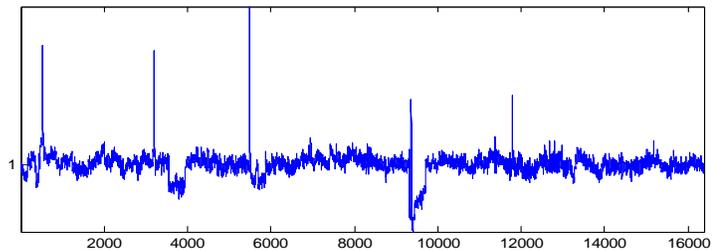


Figure 8.2: Denoised EEG signal for (A) TIWT (B) FastICA (C) RADICAL

8.2.1 Testing TIWT

Performance of TIWT was determined using datasets 1 and 2. Comparison was done using two successful ICA algorithms - FastICA and, RADICAL. Both algorithms were downloaded from the web sites of the respective authors. In the case of FastICA, a symmetric orthogonal view based on the \tanh gradient function was utilized.

8.2.2 Results

8.2.2.1 Performance Comparison Test 1

Figure 8.1 shows a contaminated EEG signal. This is a sample segment from the datasets utilized in my experiment. These datasets were denoised using FastICA, and RADICAL ICA algorithms along with TIWT. Figure 8.2 shows the denoised results of the EEG in Figure 8.1.

Examination of both figures allow for the assumption that TIWT produces better results. How accurate is this assumption? To answer this question I utilized three of the performance measures described in Chapter 3 – MSE, SNR and PRD.

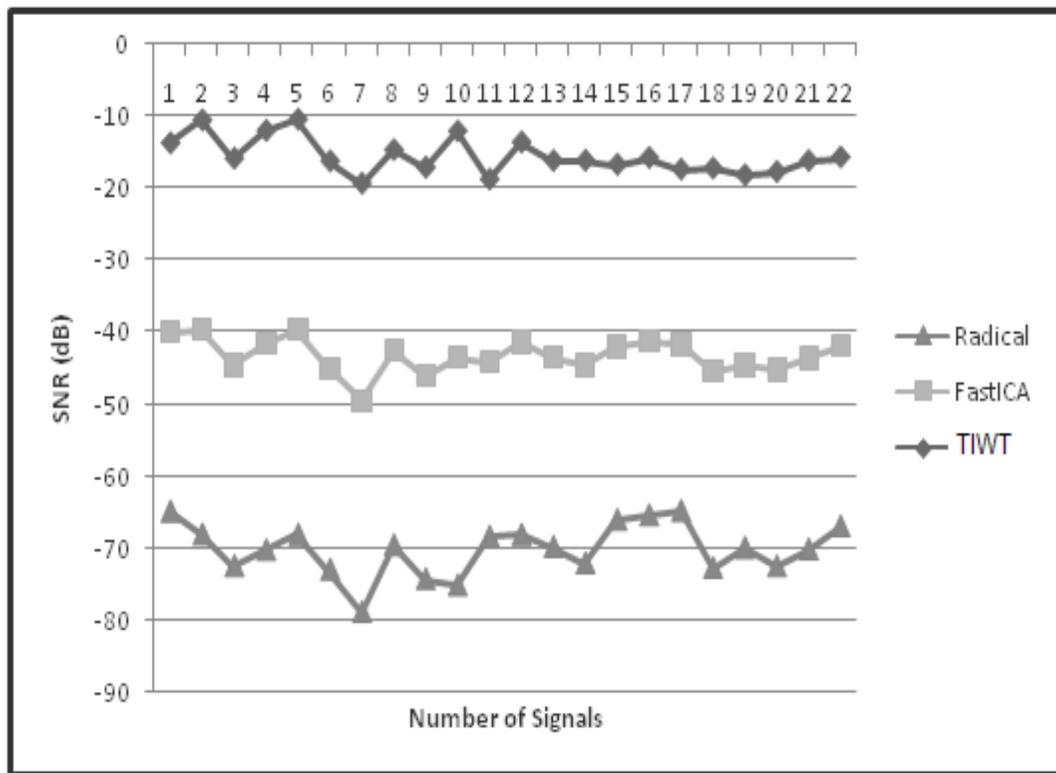


Figure 8.3: SNR Comparison of EEG Signals

Figure 8.3 shows the results of the SNR calculations (in negative values). Close examination shows that TIWT has the SNR nearest to 0, falling between -10 and -20. RADICAL on the other hand had the worst falling between -60 and -80. Having a SNR far from 0 indicates that the algorithm's performance deteriorates i.e. it ranges from low to moderate to high noise conditions. Having the highest SNR clearly demonstrates that TIWT has filtered out more noise than the other algorithms and therefore produces cleaner signals.

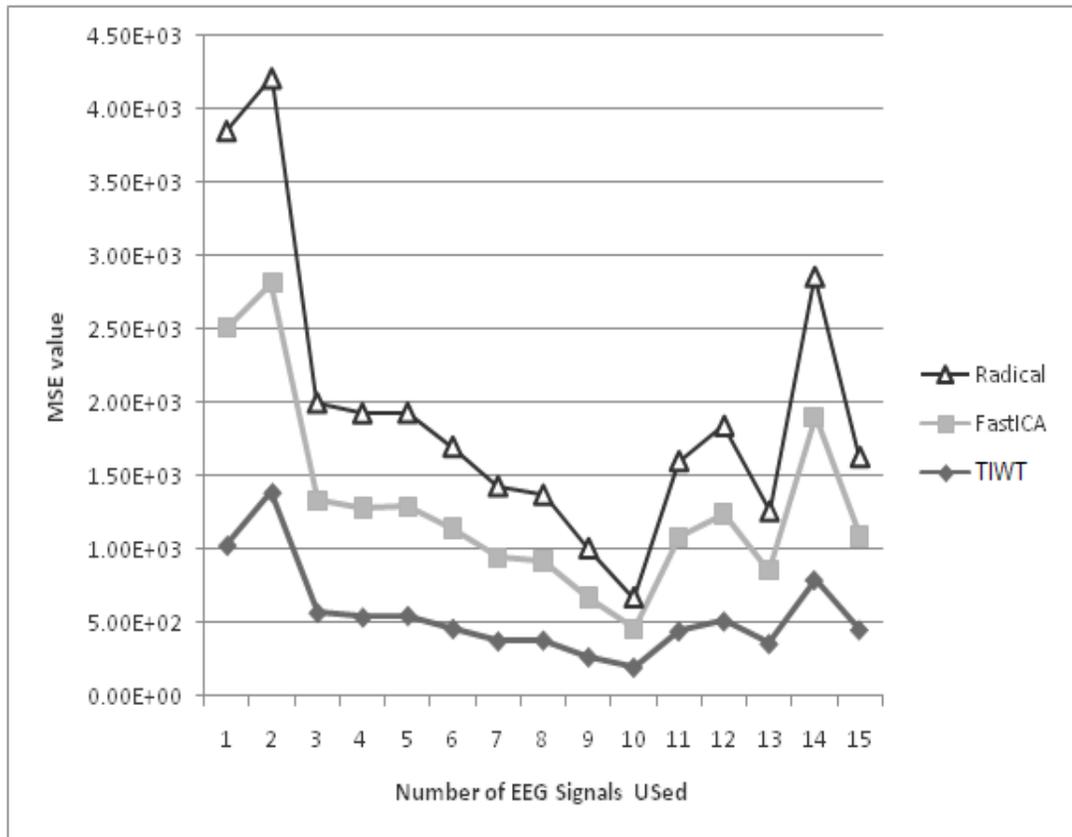


Figure 8.4: MSE Comparison of EEG Signals

MSE measures the average of the square of the “error” which is the amount by which the estimator differs from the quantity to be estimated. The difference occurs because of randomness or because the estimator doesn’t account for information that could produce a more accurate estimate. For a perfect fit, $I(x,y) = I'(x,y)$ and $MSE = 0$; so, the MSE index ranges from 0 to infinity, with 0 corresponding to the ideal. The smaller the MSE therefore the closer the estimator is to the actual data. A small mean squared error means that the randomness reflects the data more accurately than a larger mean squared error. Figure 8.4 presents the MSE of all three algorithms where TIWT has the smallest MSE; this indicates that TIWT has produced the signal which is nearest to the pure signal.

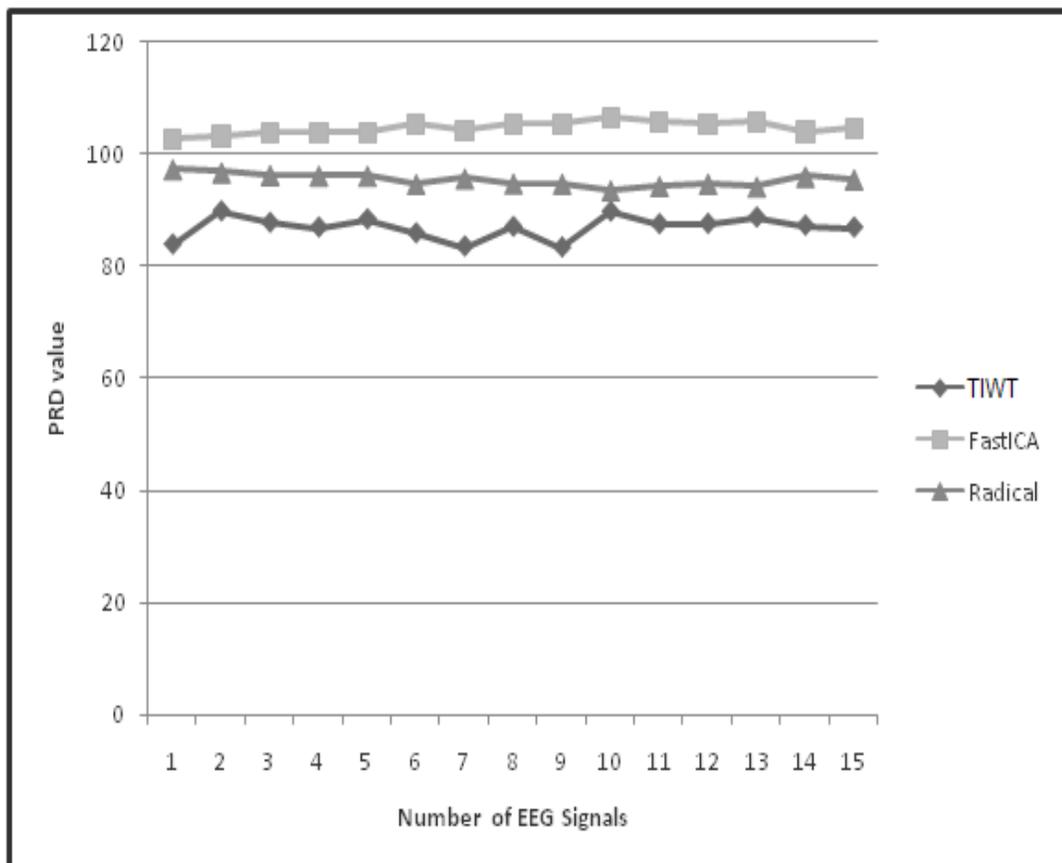


Figure 8.5: PRD Comparison of EEG Signals

The level of the distortion between the original signal and the reconstructed signal is determined by the PRD; the smaller the value the better the algorithm's performance. In Figure 8.5 FastICA and RADICAL both have higher PRD values than TIWT indicating that in these cases, the performances are weaker due to the presence of noise.

8.2.2.2 Performance Comparison Test 2

After testing with FastICA and RADICAL I went on to investigate TIWT's performance against other ICA algorithms. It is known that the lower the SIR of an algorithm, the better the achieved separation and a SIR index of 0 implies a perfect separation. Examination of Figure 8.6 which displays the SIR of the seven algorithms tested shows that TIWT displays the lowest SIR - below 0.5 which is the nearest to 0, implying the best separation.

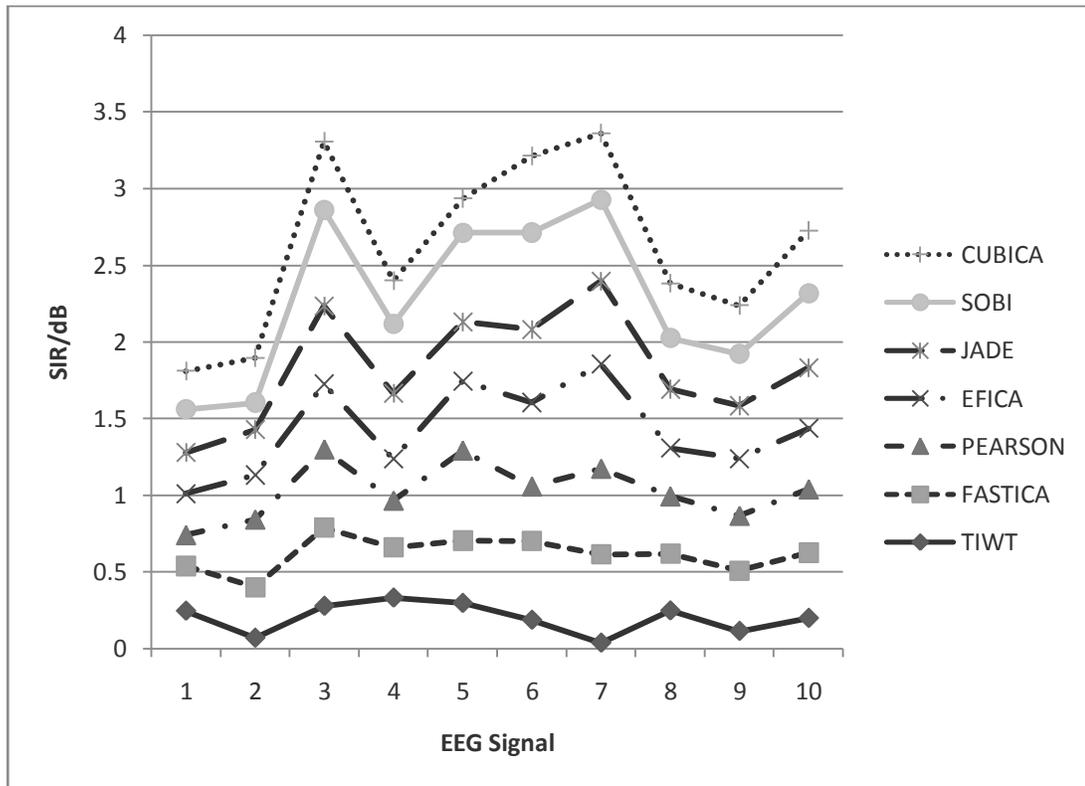


Figure 8.6: SIR Comparison with Six Other Algorithms

Figure 8.7 shows the results of an Amari Index test. When the separation is perfect, the Amari index for an algorithm is equal to zero. From the figure no algorithm has perfect separation; however TIWT has the lowest index for all signal sizes indicating that it has the best overall separation performance.

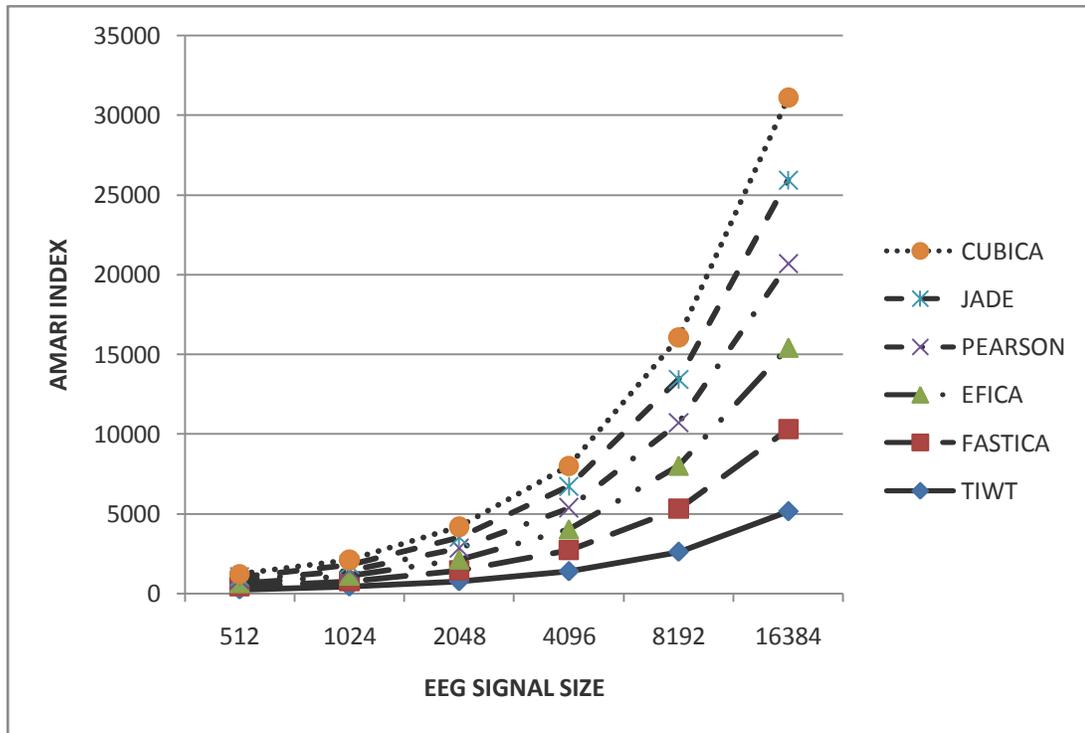


Figure 8.7: Amari Index Comparison with Five Other Algorithms

Further investigations showed the results of PSNR comparisons in Table 8.1 and MSE in Table 8.2. Table 8.1 shows that TIWT produces the highest PSNR on average of the seven algorithms tested indicating that it has a better quality for the reconstructed signal. Once PSNR is high then the MSE should be low. Table 8.2 confirms this as TIWT has the lowest MSE of the algorithms.

Table 8.1. PSNR for real 11 real EEG signals

TIWT	FASTICA	PEARSON	EFICA	JADE	SOBI	CUBICA
17.0	16.0	15.9	16.0	15.9	15.9	15.9
19.4	17.1	16.9	16.9	17.0	17.0	17.0
20.3	18.2	18.4	18.4	18.4	18.3	18.3
20.3	18.3	18.3	18.4	18.3	18.5	18.5
21.2	19.3	19.3	19.4	19.3	19.3	19.3
19.6	17.4	-	17.5	17.5	17.5	17.6
19.0	16.9	16.8	17.0	17.0	17.0	17.0
20.2	18.3	-	18.2	18.0	18.3	18.3
18.5	17.4	17.4	17.4	17.5	17.4	17.4
16.2	14.9	14.9	14.9	14.8	14.9	14.9
17.2	16.0	16.0	15.9	16.0	16.0	16.0
19.0	17.2	17.1	17.3	17.2	17.3	17.3

Table 8.2. MSE for real 11 real EEG signals

TIWT	FASTICA	PEARSON	EFICA	JADE	SOBI	CUBICA
1106.9	1646.4	1671.4	1650.7	1663.5	1670.9	1675.3
739.1	1273.5	1322.1	1324.3	1311.9	1293.6	1295.7
609.2	977.7	937.5	937.6	943.0	936.3	960.4
608.0	956.1	959.4	945.3	957.3	917.8	918.7
491.2	765.0	757.4	747.8	757.6	765.6	753.9
709.2	1194.5	-	1168.1	1167.8	1146.3	1134.3
818.4	1322.5	1341.4	1308.4	1305.0	1307.0	1304.6
615.3	972.8	-	989.1	1032.6	967.3	965.2
917.3	1187.0	1189.6	1183.1	1167.4	1177.2	1171.1
1566.8	2118.0	2124.8	2106.5	2139.0	2122.1	2125.9
1242.1	1646.3	1647.4	1657.3	1651.3	1649.8	1642.9
856.7	1278.2	1327.9	1274.4	1281.5	1268.5	1268.0

8.2.2.3 Computational Cost

Examination of the TIWT algorithm resulted in the determination of its computational complexity. Research has declared that the TIWT algorithm has a complexity of $O(N^2 \log N)$ [99]. From Chapter 4 it was seen that FastICA has a

complexity on the order of $O(N^3M)$, JADE has one of $O(N^4M)$ [159] and EFICA has a computational complexity only slightly (about three times) higher than that of the standard symmetric FastICA. Researchers have also found that Pearson_ICA has a complexity similar to FastICA [98] and RADICAL has a complexity of $O(MN \log N)$ [115]. This shows that TIWT has a comparable complexity to known ICA algorithms.

Further performance comparison was done when I investigated the speed of analysis. It was found that there was not much difference in the running time of all algorithms.

8.2.3 Conclusion

I conclude therefore that while TIWT had similar running time and comparable complexity it outperformed all tested algorithms by possessing

- (i) the smallest MSE
- (ii) the smallest PRD when tested with FastICA and RADICAL
- (iii) the highest SNR and PSNR and
- (iv) the lowest Amari Index for all tested signal sizes.

The translation invariant method of wavelet is therefore an efficient technique for improving the quality of EEG signals. But can I continue improving? I will try to answer this question in the following sections – 8.3 and 8.4.

8.3 Merging Filters and WT

In Chapter 7 and section 8.2 I focused on the use of two denoising techniques which were applied to EEG signals – Filters and WT. In this section I seek to create a new algorithm which calls for the merger of these techniques with ICA. The section seeks to answer the question posed in 8.1 as well as the question “Can the

merger of these three techniques improve the quality of the results produced after denoising EEG signals?”

8.3.1 Reasons for Merger

8.3.1.1 Improving WT with Filters

Researchers have shown that improvements of WT have used Wavelet-transform-based soft-thresholding as the pre-processing of Kalman Filtering (KF) in denoising. In 2006 the combination of WT and KF was a novel idea. In the experiments, researchers [156] found that the combination effectively correct overlapped spectra and reduce noise. Mastriani *et al.* [111] created the KalmanShrink for the WT; simulations showed that the threshold had better performance than the most commonly used filters [111]. The use of KF and WT combination therefore improved denoising techniques.

The idea of Wiener filtering of individual wavelet coefficients arose from the fact that wavelet transforms tend to decorrelate data. An improved wavelet domain denoising technique was therefore proposed that utilizes the Wiener filtering of wavelet coefficients [40]. Research shows that this technique has superior performance over other denoising algorithms using thresholding or shrinkage of wavelet coefficients and has motivated the analysis of many denoising algorithms in terms of optimal filtering of noisy wavelet coefficients. Research has also shown that the Kalman Filter (KF) outperforms the Wiener Filter when applied to WT [130], and that UKF is an advancement on KF. In this dissertation I seek to see if the application of UKF also improves denoising in the wavelet domain.

8.3.1.2 Problem with ICA

Although ICA is popular and for the most part does not result in much data loss; its performance depends on the size of the data set i.e. the number of signals. The larger the set, the higher the probability that the effective number of sources will overcome the number of channels (fixed over time), resulting in an over complete ICA. This algorithm might not be able to separate noise from the signals. Another problem with ICA algorithms has to do with the signals in frequency domain.

Although noise has different distinguishing features, once they overlap the EEG signals ICA cannot filter them without discarding the true signals as well. This results in data loss.

8.3.1.3 Comparison of WT and ICA

Each technique has produced excellent results as researchers have shown. Each however produces limitations. Recently there has been research comparing the denoising techniques of both ICA and WT. It was found that

- (i) if noise and signals are nearly the same or higher amplitude, wavelets had difficulties distinguishing them. ICA, on the other hand, looks at the underlying distributions thus distinguishing each [30] and
- (ii) ICA gives high performance when datasets are large. It suffers however from the trade off between a small data set and high performance [63].

Research therefore shows that ICA and wavelets complement each other, removing the limitations of each [145]. Since then research has been done applying a combination of both with ICA as a pre- or post- denoising tool. Inuso *et al.* [63] used them where ICA and wavelets are joint. They found that their method outperformed the pre- and post- ICA models.

8.3.1.4 Reasons Summary

Investigations have shown that each of the chosen techniques – filters, WT and ICA aim at improving the others in that

- (i) WT removes overlapping of noise signals that ICA cannot filter out.
- (ii) ICA can distinguish between noise and signals that are nearly the same or higher amplitudes which WT has difficulty with.
- (iii) WT exhibits serious problems such as pseudo-Gibbs phenomena which CS eliminates by creating the TIWT method and
- (iv) Combination of filters and WT effectively correct overlapped spectra

I propose a merger of all methodologies to create the Cycle Spinning Wavelet Transform ICA (CTICA) as represented as a block diagram in Figure 8.8.

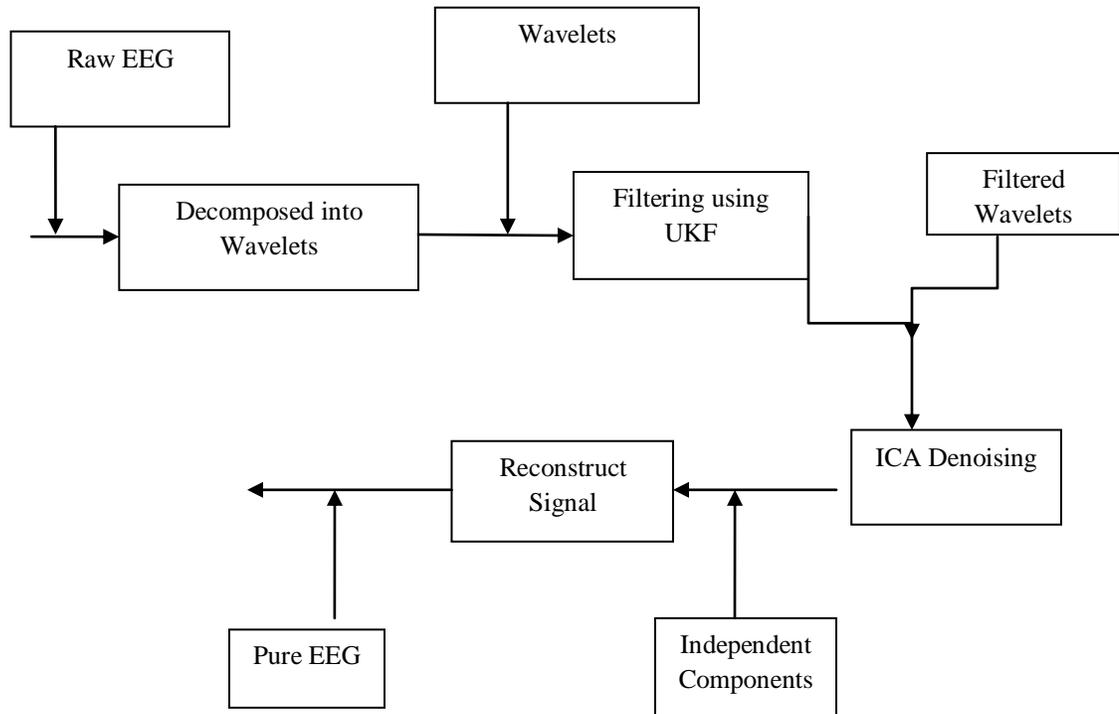


Figure 8.8: Proposed CTICA - Artifacts Removal System

8.3.2 Methodology for Merger

CTICA is the merger of TIWT, UKF and ICA into one to improve the denoising of EEG signals. The algorithm utilizes the TIWT algorithm described in 8.2 as the base, with the following modification after the decomposition of signals:

Filter Coefficients

Perform UKF on the coefficients to filter out some noise reducing the shrinkage threshold.

Choose and Apply Threshold Value (based on description in 8.2)

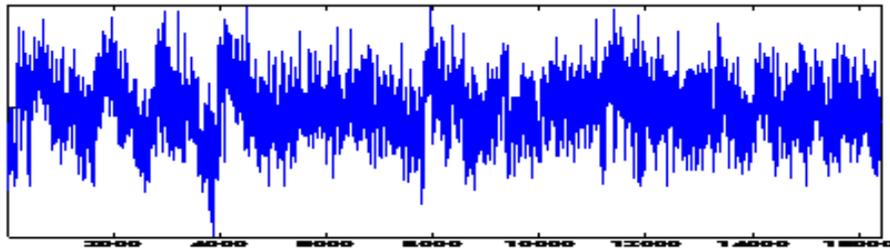
Apply ICA algorithm

Signals and noise may have nearly the same frequency characteristics and overlap in time thus producing noisy coefficients such as beta activity and muscle noise, that

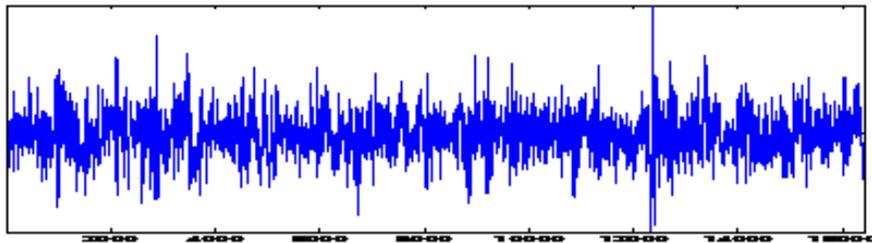
WT has not been able to distinguish and remove. ICA is able to look at the underlying distributions thus distinguish noise and remove them. Research has shown that ICA is a robust denoising method where its performance is not affected by the severity of the mixing signals [30]. We implemented a symmetrical fixed-point ICA algorithm based on the Hyvarinen model [62] where the gradient function is:

$$g(y) = \tanh(a, y) \quad (8.1)$$

A fixed-point algorithm has a cubic or at least a quadratic convergence, is not linear and no parameters have to be chosen for usage which makes it a better choice than other ICA models. Once this is done EEG signals are reconstructed using inverse TIWT.



(a)



(b)

Figure 8.9: (a) EEG Signal with EOG (b) Denoised EEG Signal

8.3.3 Results & Discussions

In order to do the study effectively data was collected for analysis from datasets 1 and 2 in Laptop 1 environment. Noisy signals were generated by adding noise to the original noise-free signals and the length of all signals, N , were truncated to lengths of power of twos i.e. 2^x .

Figure 8.9 shows the results of the above algorithm on one EEG signal contaminated with EOG. Investigations on the wavelet coefficients (Figure. 8.10) also show that there are major changes in the wavelets - some wavelets have been zeroed because of their identification to noise.

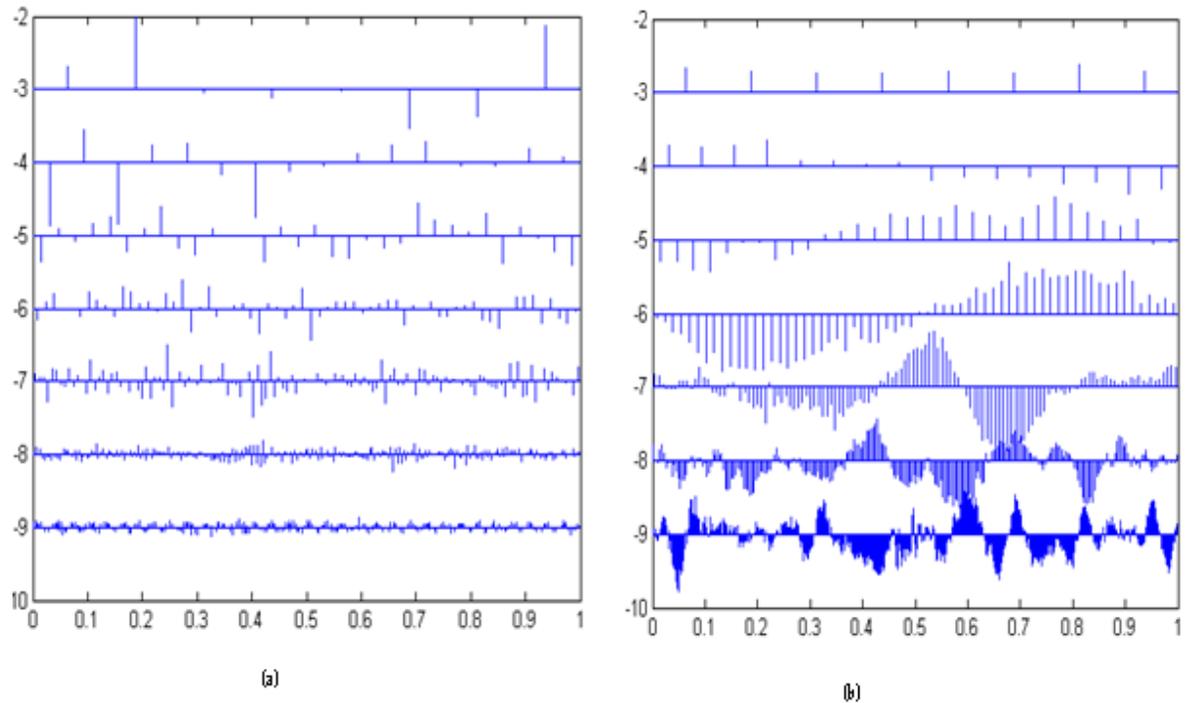


Figure 8.10: Wave Coefficient (a) before denoising (b) after denoising

8.3.3.1 Performance Comparison

To quantify the ability of my proposed algorithm in recovering the desired signal we compared its performance with several state-of-the-art fixed-point ICA

algorithms - FastICA, EFICA, and Pearson-ICA. All the algorithms were downloaded from the web sites of the respective authors. In all cases a symmetrical view based on the *tanh* score function was used.

Table 8.3. MSE for real 15 real EEG signals with EOG noise

	CTICA	FASTICA	EFICA	PEARSON_ICA
1	1411.0	1489.7	1412.4	1337.2
2	950.1	1014.8	951.7	890.7
3	714.5	769.9	715.3	662.7
4	693.7	749.0	694.7	642.4
5	757.9	815.2	758.8	704.5
6	858.6	921.0	859.3	799.9
7	849.2	923.8	849.6	778.5
8	788.2	850.9	788.8	729.0
9	875.4	941.8	876.2	813.1
10	369.9	410.6	370.3	332.0
11	297.6	333.6	298.1	264.6
12	526.5	574.2	527.1	482.0
13	636.1	696.3	636.4	579.3
14	1031.7	1116.7	1032.0	950.7
15	236.4	269.1	236.9	206.8
Average	733.1	791.8	733.8	678.2

Table 8.4: MSE for 19 EEG signals with artificially added noise

	CTICA	FASTICA	EFICA	PEARSON_ICA
1	26351	26678	26352	26028
2	12824	13051	12823	12597
3	6449.3	6610	6447.8	6287.7
4	5493.4	5642.2	5492.6	5345.2
5	6221.4	6379.9	6220.6	6063.5
6	106.5	87.3	107.0	87.4
7	2481.9	2583.6	2482.6	2383.8
8	6457.5	6618.9	6456.8	6296.9
9	6451.6	6612.9	6450.7	6290.7
10	12811	13038	12810	12584
11	5508.1	5657.2	5507.5	5359.8
12	16004	16265	16008	15755
13	3839.7	3968.9	3842.1	3717.9
14	1999.4	20921	2000.2	1910.7
15	451.7	494.8	451.3	410.0
16	470.5	517.9	471.6	427.6
17	13454	13694	13458	13255
18	13089	13324	13092	12863
19	3850.6	3979.8	3853	3728.6
Average	7595.5	8743.4	7596.2	7441.7

Table 8.3 shows the MSE when tested on real EEG signals naturally contaminated with EOG while Table 8.4 shows the results for artificially contaminated signals. In Table 8.3 CTICA tended to have smaller MSE than FASTICA and EFICA. Table 8.4 also shows that CTICA has a smaller MSE than both algorithms for artificially contaminated signals. This indicates that of the four algorithms tested only Pearson_ICA produced smaller MSE. This indicates that only Pearson_ICA outperforms CTICA in both real and artificially contaminated EEG signals.

Table 8.5. PSNR for 20 real EEG signals with EOG noise

	CTICA	FASTICA	EFICA	PEARSON_ICA
1	16.6	16.4	16.6	16.9
2	18.3	18.1	18.3	18.6
3	19.6	19.3	19.6	19.9
4	19.7	19.4	19.7	20.1
5	19.3	19.0	19.3	19.7
6	18.8	18.5	18.8	19.1
7	18.8	18.5	18.8	19.2
8	19.2	18.8	19.2	19.5
9	18.7	18.4	18.7	19.0
10	22.5	22.0	22.4	22.9
11	23.4	22.9	23.4	23.9
12	20.9	20.5	20.9	21.3
13	20.1	19.7	20.1	20.5
14	18.0	17.7	18.0	18.4
15	24.4	23.8	24.4	25.0
16	22.3	21.9	22.3	22.7
17	20.0	20.4	20.0	20.4
18	19.7	19.4	19.7	20.1
19	20.5	20.0	20.5	21.0
20	21.6	21.1	21.6	22.1
Average	20.1	19.8	20.1	20.5

Table 8.5 shows the PSNR for EOG contaminated signals and Table 8.6 shows those with artificially contaminated noise. If the PSNR is high then the ratio of signal to noise is higher and therefore the algorithm is considered good. After experiments it can be seen that the PSNR values for all four algorithms are very close. CTICA on average however, has the second best performance with Pearson_ICA presenting a slightly better performance.

Table 8.6. PSNR for 19 EEG signals with artificially added noise

	CTICA	FASTICA	EFICA	PEARSON
1	3.9	3.9	3.9	4.0
2	7.1	7.0	7.1	7.1
3	10.0	10.0	10.0	10.2
4	10.7	10.6	10.7	10.9
5	10.2	10.1	10.2	10.3
6	27.9	28.7	27.8	28.7
7	14.2	14.0	14.2	14.4
8	10.0	10.0	10.0	10.1
9	10.0	10.0	10.0	10.1
10	7.1	7.0	7.1	7.1
11	10.7	10.6	10.7	10.8
12	6.1	6.0	6.1	6.2
13	12.3	12.1	12.2	12.4
14	15.1	14.9	15.1	15.3
15	21.6	21.2	21.6	22.0
16	21.4	21.0	21.4	21.8
17	6.8	6.8	6.8	7.0
18	7.0	6.9	7.0	7.0
19	12.3	12.1	12.3	12.4
Average	11.8	11.7	11.8	12.0

Table 8.7. SDR for 19 with artificially added noise

	CTICA	FASTICA	EFICA	PEARSON
1	10089.0	9916.5	9977.6	10039.0
2	1900.1	1867.7	1884.2	1901.0
3	930.9	911.9	923.3	935.0
4	1141.0	1112.7	1127	1143.1
5	966.0	945.5	957.5	969.9
6	600.6	503.9	455.2	503.9
7	-663.6	-660.6	-673.9	-687.7
8	1266.6	1237.2	1252.6	1268.4
9	961.9	942.0	965.9	965.9
10	1371.9	1351.6	1363.6	1375.8
11	1074.2	1048.2	1062.4	1076.9
12	901.3	888.2	895.3	902.5
13	371.5	361.4	367.3	373.3
14	258.6	249.0	254.7	260.6
15	-2441.1	-6579.9	-6889.7	-7228.6
16	80.2	75.0	78.6	82.5
17	771.0	759.2	765.8	772.5
18	848.9	8354.0	842.3	849.8
19	384.0	373.4	379.5	385.8
Average	1095.4	849.4	841.5	836.3

Table 8.7 showed that CTICA produced the largest SDR on average for the artificially contaminated signals. This shows that on average CTICA had the best separation of signal from noise than the other algorithms. Figure 8.11 shows all four algorithms having SDR above 8dB; there is not much differentiation in the graph for the algorithms however. Where there were difference in the SDR calculations CTICA had the most consistent.

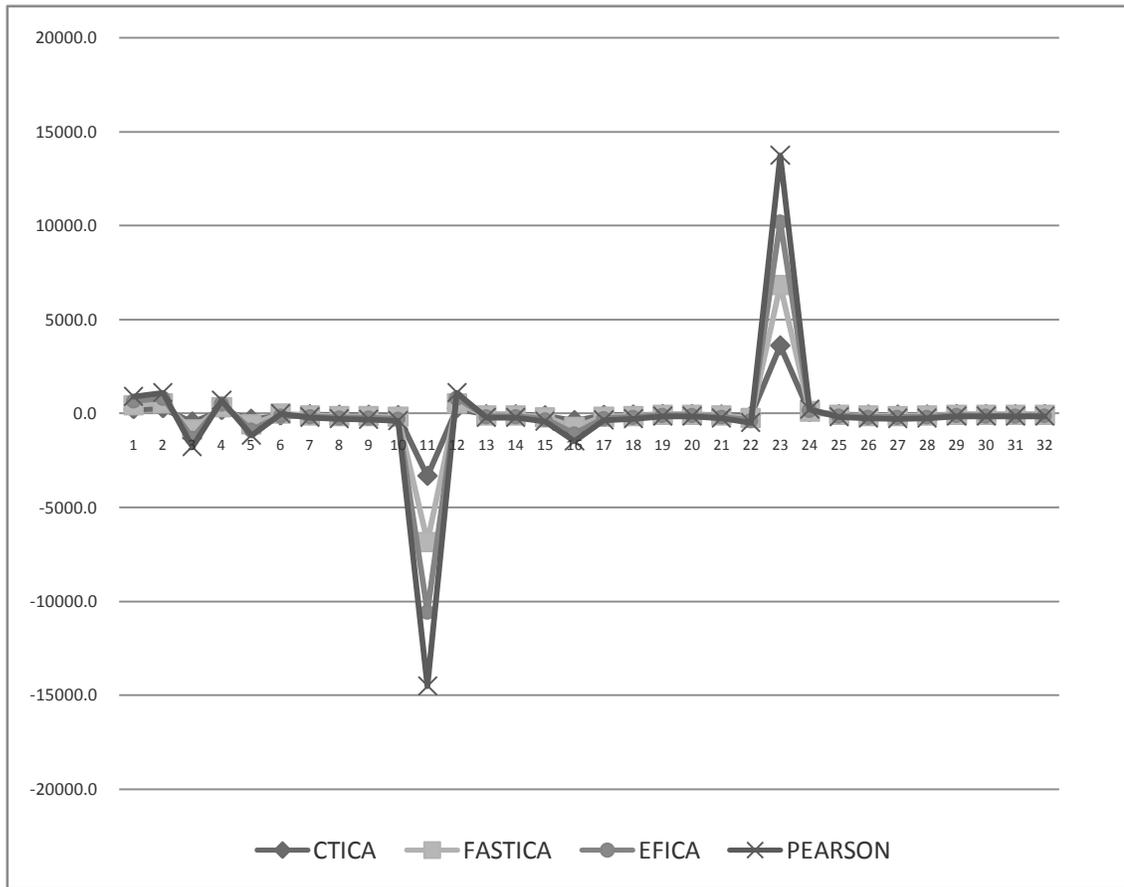


Figure 8.11: SDR for 32 Real EEG signals with EOG

Observations of results show that the Amari indexes for my method is lower for sample sizes greater than 2^7 i.e. it clearly outperforms the other algorithms with

sample size greater than 128. Figure 8.12 shows that unlike the other algorithms, the Amari index for CTICA is inversely proportional to a sample size.

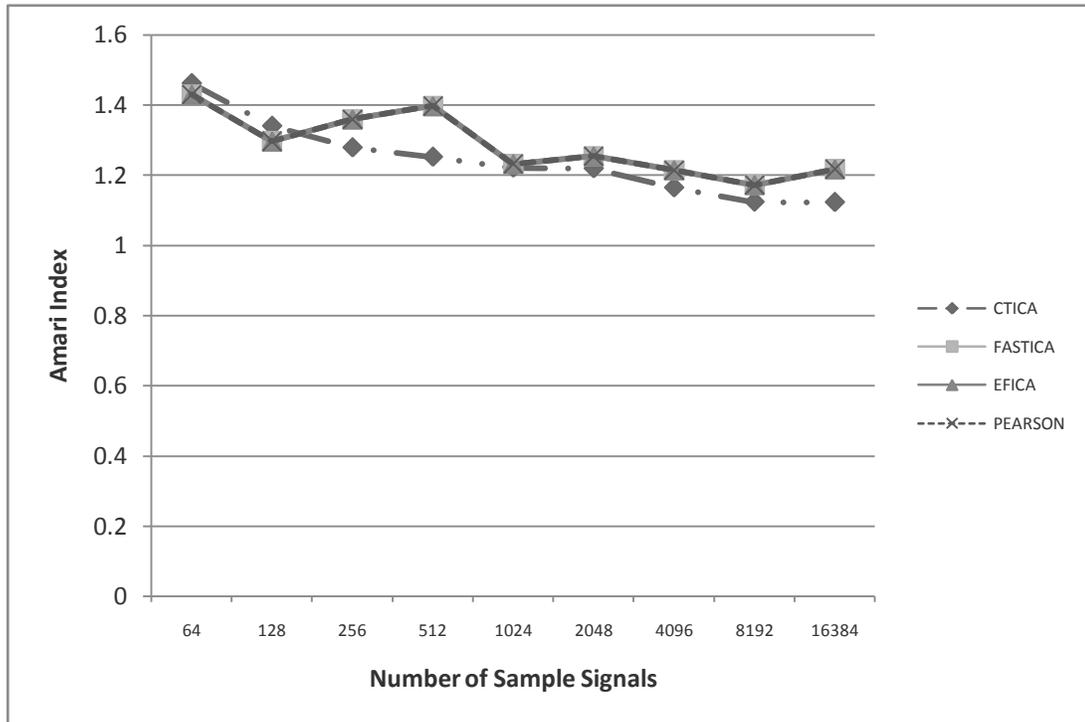


Figure 8.12: Amari Results for the four algorithms

Statistical Testing of datasets

To determine the relationship between the datasets in each algorithm I applied the t-test as I did in Chapter 6. For every test the chosen level of significance is 0.05, the number of distribution tail is 2, and the sample groups have equal variance; however the degree of freedom (*df*) varies depending on the table tested. For example the *df* for Table 8.3 is 28, for Table 8.4 - 36 and for Table 8.5 - 38.

Below are the t-test results for all algorithms in Table 8.3. Here the critical t-value based on the t-value table at http://www.stattools.net/tTest_Tab.php is 2.0484

and the mean, standard deviation (σ) and standard error for each algorithm remains constant for each test.

	CTICA	FastICA	Pearson-ICA	EFICA
Mean	7595.5053	8743.7895	7441.6737	7596.1474
σ	6664.8855	7231.8622	6590.8995	6665.3413
Standard Error	1529.0296	1659.1030	1512.0560	1529.1342

CTICA compared to Pearson-ICA - two-tailed p-value=0.6141, computed t-value=0.5099.

CTICA compared to FastICA - two-tailed p-value=0.60656, computed t-value=0.5210.

CTICA compared to EFICA - two-tailed p-value=0.9948, computed t-value=0.0065.

FastICA compared to Pearson-ICA - two-tailed p-value=0.3119, computed t-value=0.5164.

FastICA compared to EFICA - two-tailed p-value=0.6110, computed t-value=0.5144.

EFICA compared to Pearson-ICA - two-tailed p-value=0.6096, computed t-value=0.5164.

In all tests the computed t-value is less than the critical t-value. I therefore, based on these results, fail to reject the null hypothesis (H_0) that the population means behind the two samples are the same as I could not conclude that

- the population means are different
- the samples are not “equivalent”
- the experiment is not repeatable

This behavior in Table 8.3 is the same in all the other tables which leads me to conclude that the comparison between each algorithm is considered statistically non-significant.

8.3.3.2 Computational Cost

Using the same parameters as defined in Chapter 4 where N denotes the number of samples, m denotes the number of sources and M is the maximum number of iteration I determined the complexity of CTICA. Again I assume $m \leq N$.

- Performing an TIWT is $O(N^2 \log N)$
- Performing Filtering using UKF is $O(N^2)$
- Performing ICA denoising is $O(N^3 M)$

CTICA has a complexity of $O(N^3 M) + O(N^2 \log N) + O(N^2)$ resulting in an overall complexity of $O(N^3 M)$. From Chapter 4 it was seen that FastICA has a complexity on the order of $O(N^3 M)$ and EFICA has a computational complexity only slightly (about three times) higher than that of the standard symmetric FastICA. Researchers have found that Pearson_ICA has a complexity similar to FastICA [98]. It can be seen therefore that CTICA has a similar complexity in comparison to the known algorithms.

Although CTICA has a similar complexity when tested with the different datasets it takes a longer time to complete the denoising process than the other algorithms. It takes approximately 19.36s on a real data with 15 signals with 2048 channels. On the same set symmetric FastICA with tanh nonlinearity was the fastest with 8.8s

8.3.4 Conclusion

In recent years researchers have used both ICA algorithms and WT to denoise EEG signals. In this section I proposed a new method – Cycle Spinning Wavelet Transform ICA (CTICA). From the experiments I can conclude CTICA overall performance is better than the three known ICA algorithms tested namely FastICA,

EFICA and Pearson_ICA. The t-test results show that all algorithms have similar patterns therefore I can conclude that CTICA is the most consistent and robust denoising method.

8.4 TIWT and BMICA Merger

In 8.3 I sought to improve the basic DWT by merging WT, Filters and ICA. In Chapter 3 I discussed that TIWT is an improvement on WT. In this section I seek to answer the questions:

- (i) Can the merger of TIWT and ICA improve the denoising quality of the reconstructed signals? and
- (ii) Can the merger of BMICA and TIWT produce better quality results than basic TIWT?

All experiments were conducted using the Laptop environment 2 (MATLAB 7.10.0.499 (R2010) on a laptop with AMD Athlon 64x2 Dual-core Processor 1.80GHz) using data sets 1, 2, 4 and 5.

To seek answers for the above questions I designed the B-Spline Mutual Information Independent Component Analysis – Wavelet Transform (BMICA-WT) algorithm. This is a merger of TIWT and BMICA. The algorithm utilizes the TIWT algorithm of 8.2 with the execution of the BMICA algorithm after the application of thresholding.

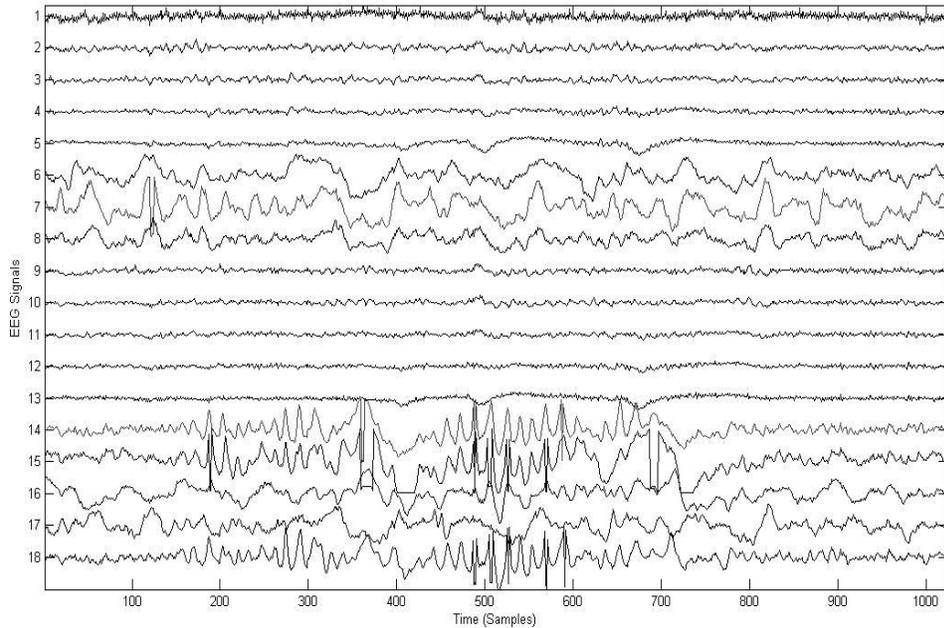
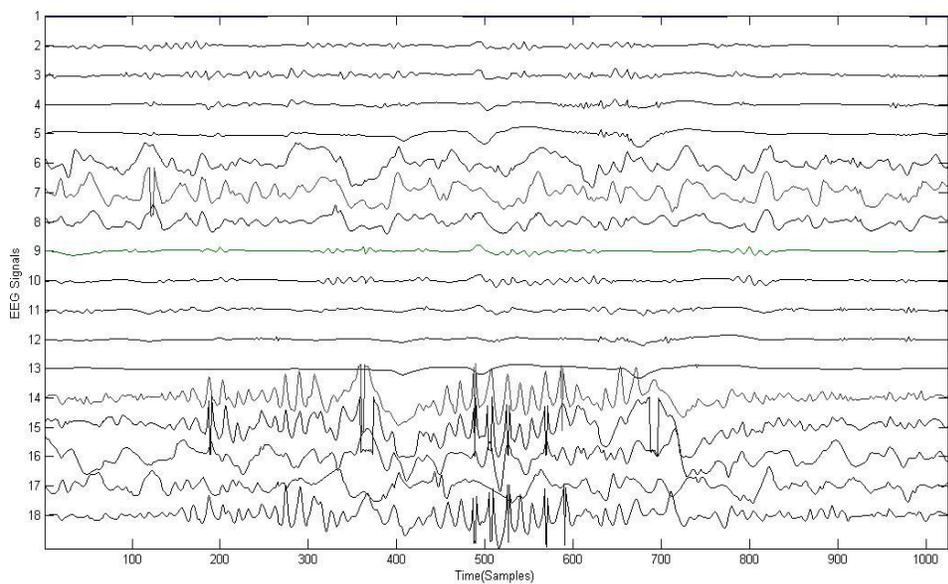


Figure 8.13: Raw EEG

8.4.1 Results & Discussions

To answer my first question I examined the original signals such as the sample shown in Figure 8.13. This figure shows one mixed EEG signal set which is contaminated having overlays in signals Nos. 6-8 and Nos. 14-18. These datasets were then denoised using TIWT and BMICA-WT. Figure 8.14 shows the signal set found in Figure 8.13 after applying TIWT and BMICA-WT. The figure shows that the overlays in all eight signals (Nos. 6-8 and Nos. 14-18) have been minimized – noise has been removed. Further analysis of the figure shows that with BMICA-WT it appears to have eliminated more noise especially in signals 14-18 than TIWT. Can we therefore conclude that BMICA has improved the denoising quality of TIWT? The answer is No. To draw this conclusion we must conduct performance comparisons using SIR, SDR, PSNR and MSE as presented below.

(A)



(B)

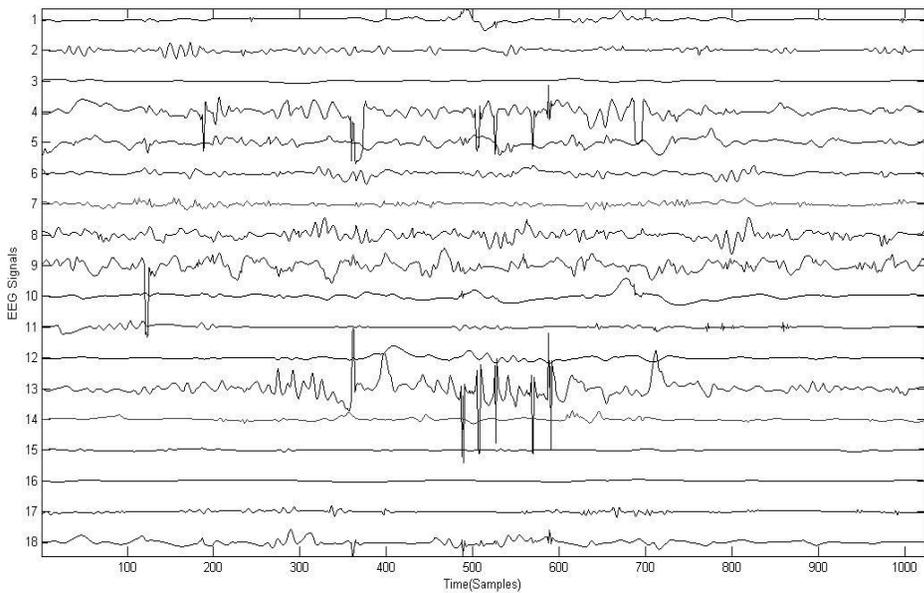


Figure 8.14: Denoising EEG with (A) TIWT (B) BMICA-WT

8.4.1.1 Performance Comparison with TIWT

Separation Accuracy

Investigations on the EEG data sets described above showed that BMICA-WT produced higher SIR calculations than TIWT. Using the samples of 18 signals in Figure 8.13 it can be seen that BMICA-WT produces SIR with higher value than TIWT 94% of the time. Literature states that the estimated source is more orthogonal to the true source as its value approaches zero. This suggests that without merging with BMICA, TIWT achieved better separation of EEG signals. So is TIWT better at denoising? I sought to answer using the other measures.

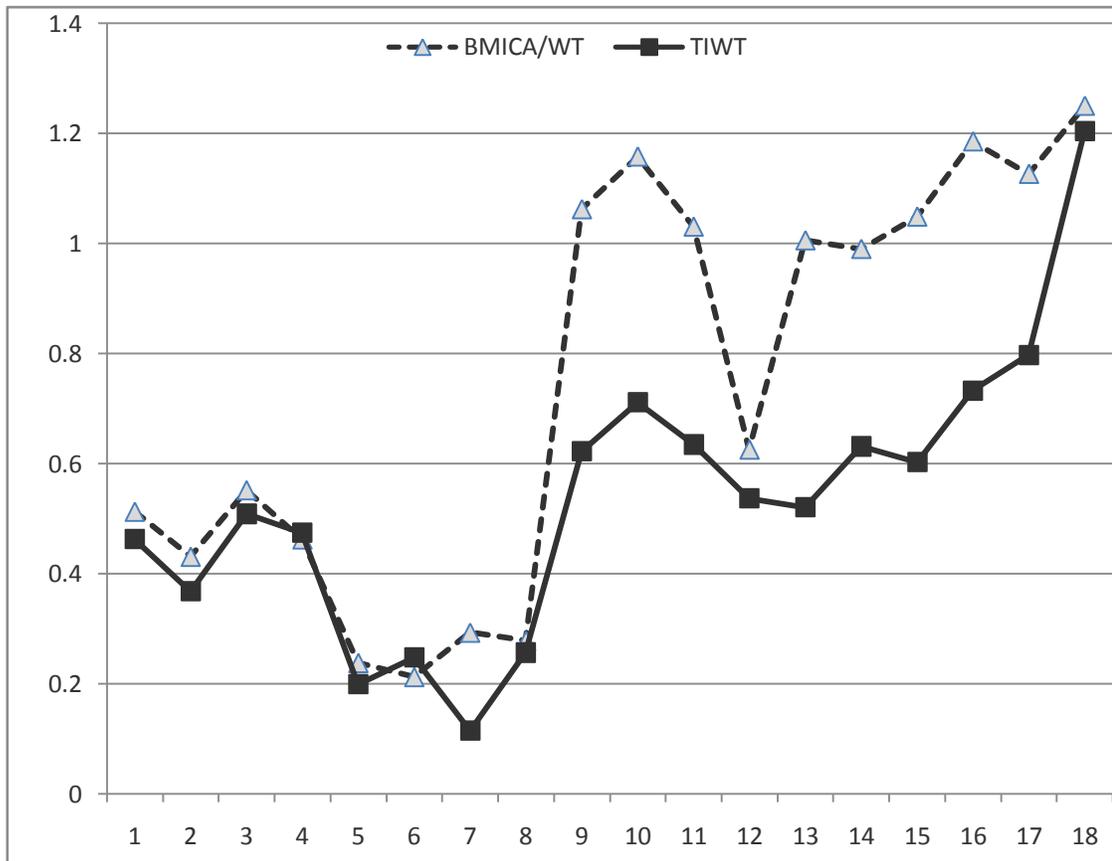


Figure 8.15: SIR relations between BMICA-WT and TIWT

I examined both algorithms using another Separation Accuracy measure in SDR. Investigations shows that BMICA-WT tends to produce higher SDRs. In Table 8.8 it can be seen that BMICA-TIWT produces higher SDR 65% of the time. This indicates that in almost every TIWT testing there is a BMICA-TIWT test which produces a more accurate overall separation of signals and noise.

Table 8.8: SDR for 19 EEG Signal Sets

BMICA-WT	TIWT
3535.0	2135.6
-88.8	-127.5
-57.4	-80.3
-112.4	-121.5
-564.5	-641.0
-217.7	-260.3
-2481.3	-3396.1
-8.6228.0	-85729.0
27.1	-0.0
47676.0	1390.0
680.2	767.0
2382.2	786.6
2.73.2	269.6
1827.2	1657.8
1.1	707.9
649.7	997.0
855.1	880.9
4.73.8	994.9
2.7062.0	21253.0

Noise/Signal

Investigations using Separation Accuracy is inconclusive since SIR and SDR favours TIWT and BMICA-WT respectively. I therefore sought measures which looked at the noise/signal relationship to answer the questions posed.

The higher an algorithm's PSNR values the better the quality of the reconstructed signals indicating that the algorithm is considered increasingly good. Figure 8.16 shows a sample of the relationship between BMICA-WT and TIWT for

PSNR. Close examinations show that for all 18 signal sets the PSNR for BMICA-TIWT were higher than those of TIWT. BMICA-TIWT therefore produces a better quality of the reconstructed signal, producing more signals than noise in its denoised signals. This implies that BMICW-WT can be considered a better algorithm for denoising.

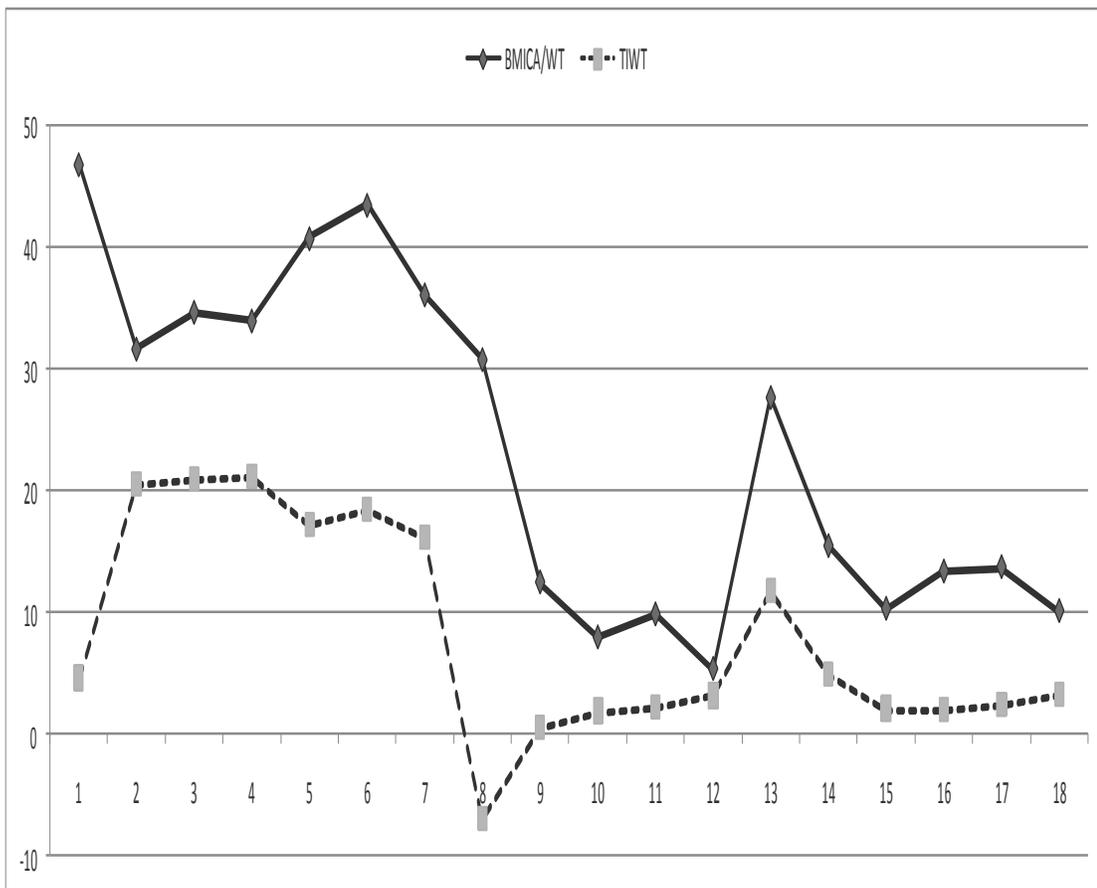


Figure 8.16: PSNR relations between BMICA-WT and TIWT

I went further by examining the MSE of the algorithms as a good algorithm will have a small MSE and a large PSNR. This is because MSE is indirectly proportional to PSNR i.e. when MSE calculated is equal to zero, then PSNR is infinite. Examination of the experiments produced results as seen in Table 8.9.

Examination shows that BMICA-WT produces a smaller MSE than TIWT. Investigations therefore show that BMICA-TIWT produces a smaller MSE and a larger PSNR than TIWT – better algorithm as it produces results closer to the actual data.

Table 8.9: MSE for 18 EEG Signal Sets

BMICAWT	WT
1.325.3	22716.0
44.0	583.1
21.9	530.0
25.8	501.2
5.4	1242.1
2.9	917.3
15.7	1566.8
53.8	324950.0
3667.6	59350000000.0
10440.0	42383.0
67438.5	40128.0
19007.0	31641.0
110.5	4325.6
18407.0	21273.0
6.055.5	40453.0
2984.1	40826.0
2745.3	37181.0
6321.3	31500.0

8.4.1.2 Comparison with ICA Algorithms

Investigations have so far shown that BMICA-WT outperforms TIWT. I wished however to see how it performed against known ICA algorithms. Does BMICA improve the denoising quality of TIWT so that it even outperforms these ICA algorithms? To answer this question I conducted performance comparisons

using SIR, AMARI, SNR, PSNR and MSE on Data sets 1, 2, and 3. BMICA-WT's results are compared with the following categories of benchmark ICAs:

(i) fixed-point- FastICA, Pearson_ICA and EFICA

(ii) non fixed-point – CubICA, SOBI, and JADE

Each algorithm and performance measure utilized is described also in Chapter 3.

Separation Accuracy

Since the Amari index is the most widely used measure for assessing the accuracy of the estimated mixing matrix I plotted the algorithms' performance. Examination of Figure 8.17 shows that BMICA-WT has the lowest Amari Index for

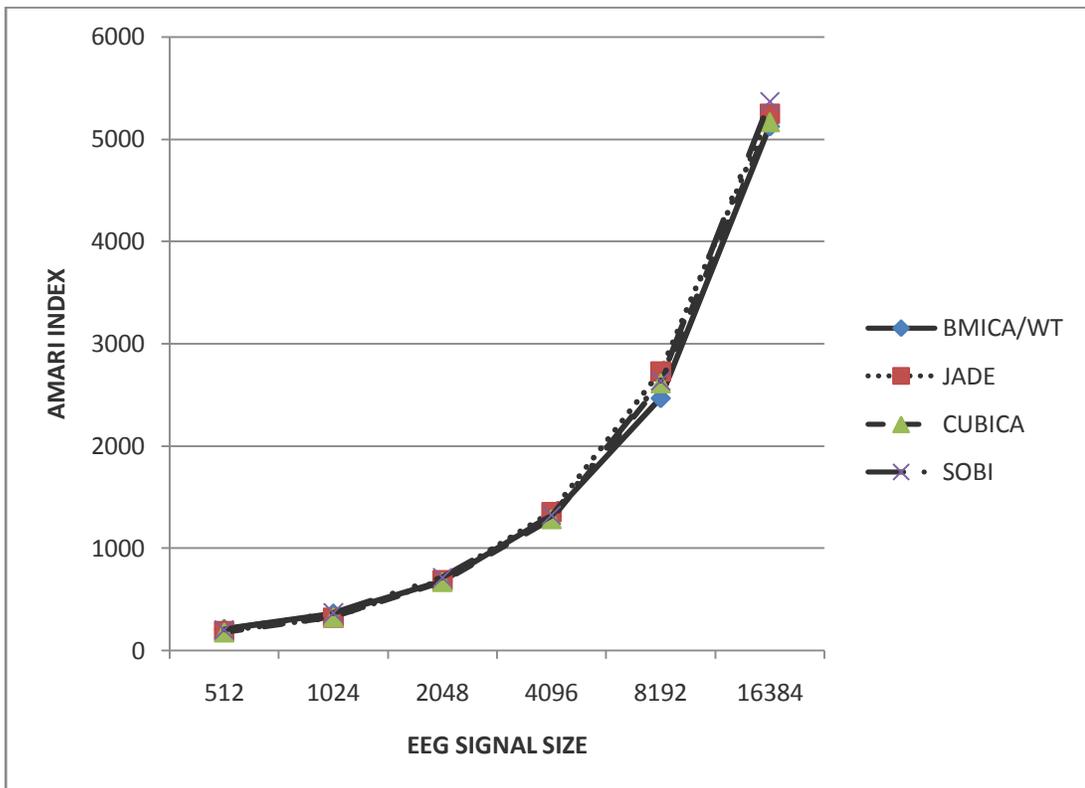


Figure 8.17: Amari Index for BMICA-WT and Non-fixed Algorithms

all signals sizes. Further examination into Figure 8.18 shows that BMICA-WT has a similar behavior pattern with fixed-point algorithms as with non-fixed-point algorithms. BMICA-WT clearly outperforms the other algorithms in all sample sizes.

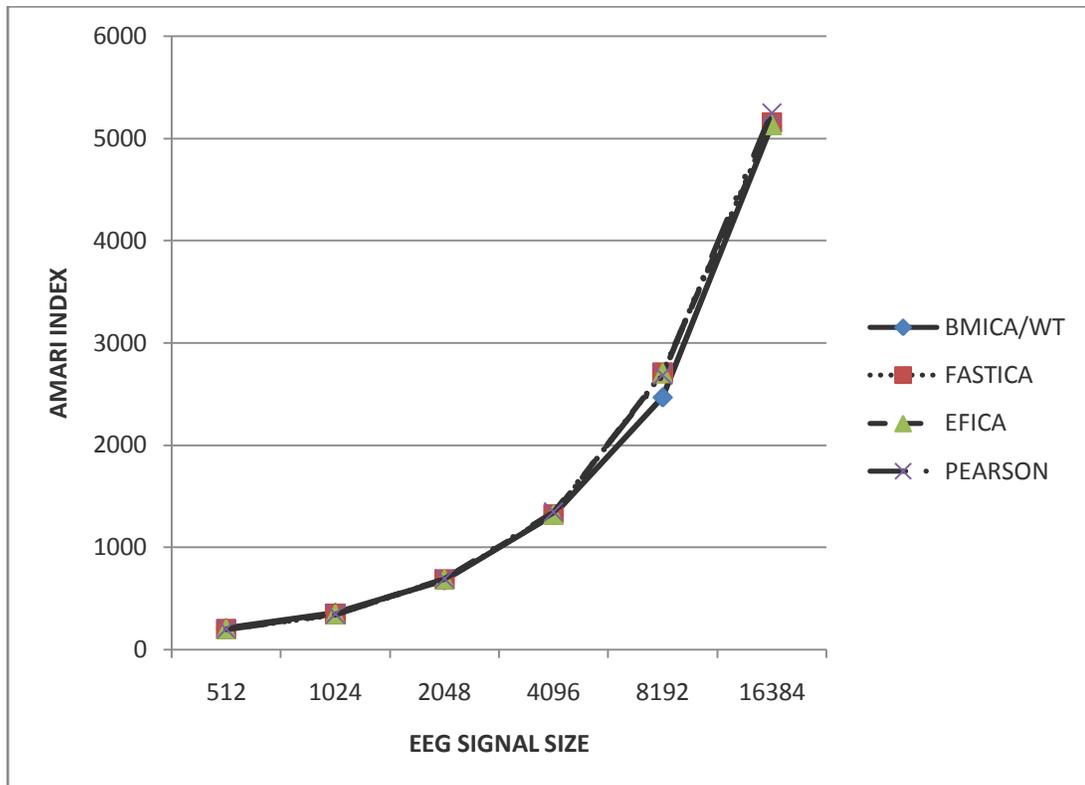


Figure 8.18: Amari Index for BMICA-WT and Fixed-point Algorithms

The Amari index shows that all algorithms have the same pattern for signal separation. I needed to investigate more and so I determined the SIR of the algorithms which are shown in Figures 8.19 and 8.20. The lower the SIR, the better the achieved separation and a SIR index of 0 implies a perfect separation. Examination of the algorithms' SIR shows that of the seven algorithms BMICA displays the SIR index nearest to 0, implying a good separation.

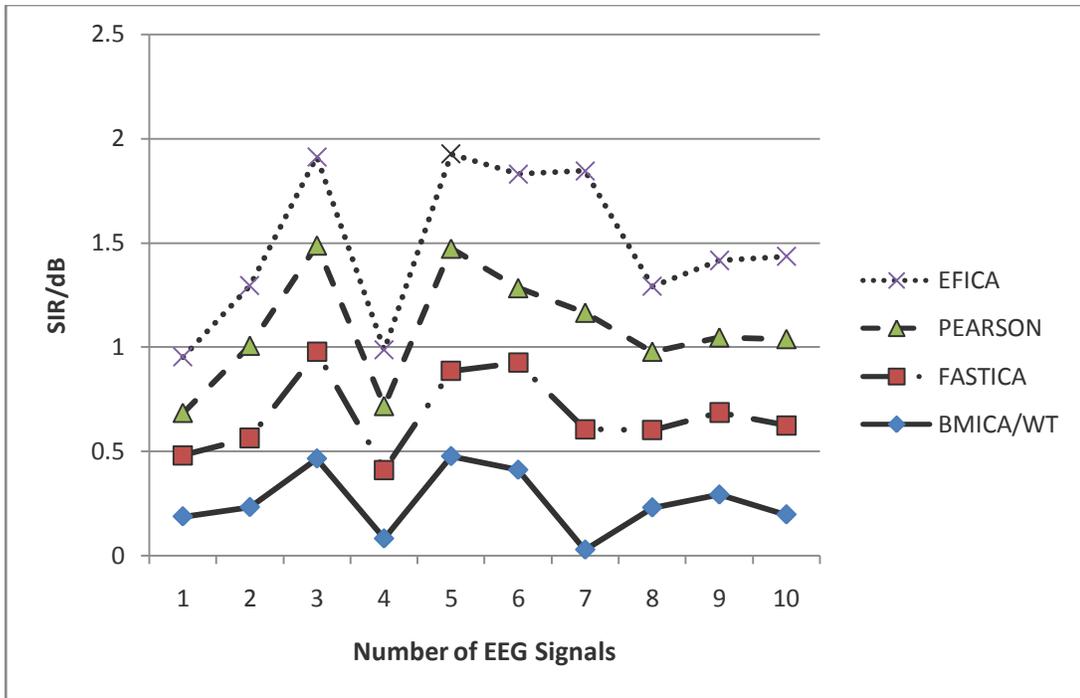


Figure 8.19: SIR for BMICA-WT and Fixed-point Algorithms

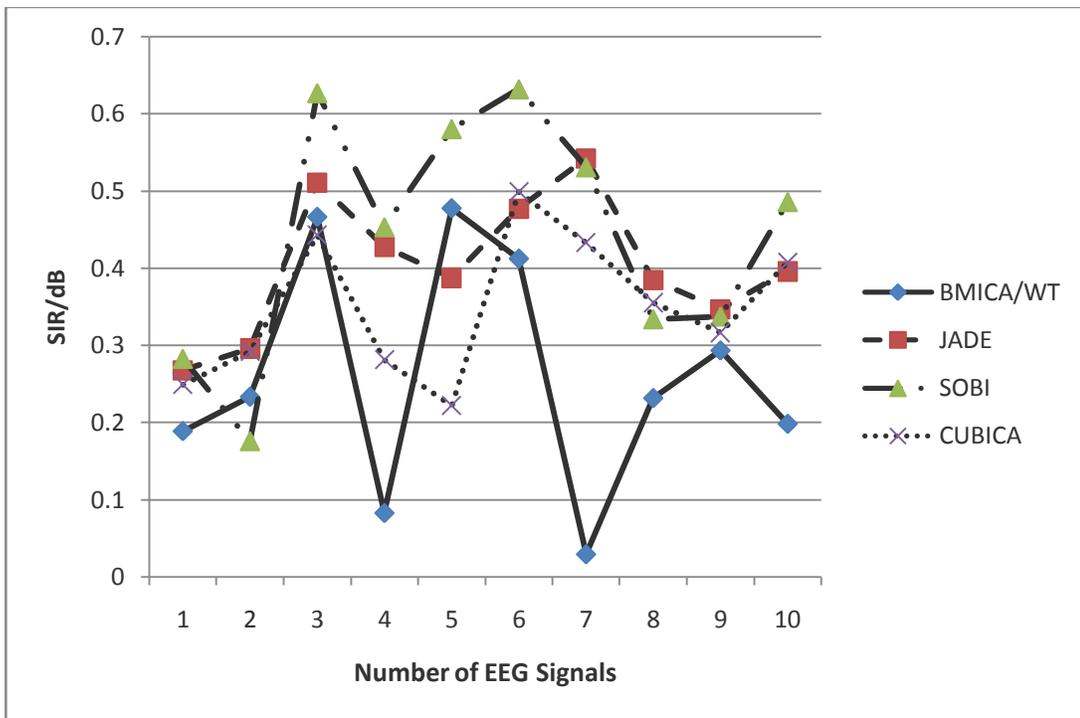


Figure 8.20: SIR for BMICA-WT and Non Fixed-point Algorithms

Noise/Signal

For an algorithm to have a good noise/signal relationship it should have a high PSNR and SNR and a low MSE. Does BMICA-WT possess these qualities? Table 8.10 shows the MSE comparisons for both fixed and non fixed algorithms.

Table 8.10: MSE for (A) Fixed-Point Algorithms (B) Non-fixed Point Algorithms

BMICA/WT	FASTICA	PEARSON	EFICA
23.5	1646.4	1671.4	1650.7
35.0	1273.5	1322.1	1324.3
17.0	977.7	937.5	937.6
44.8	956.1	959.4	945.3
13.7	765.0	757.4	747.8
20.3	1194.5	-	1168.1
38.4	1322.5	1341.4	1308.4
37.4	972.8	-	989.1
2.9	1187.0	1189.6	1183.1
15.7	2118.0	2124.8	2106.5
4.9	1646.3	1647.4	1657.3
23.1	1278.2	1327.9	1274.4

(A)

BMICA/WT	JADE	SOBI	CUBICA
23.5	1663.5	1670.9	1675.3
35.0	1311.9	1293.6	1295.7
17.0	943.0	936.3	960.4
44.8	957.3	917.8	918.7
13.7	757.6	765.6	753.9
20.3	1167.8	1146.3	1134.3
38.4	1305.0	1307.0	1304.6
37.4	1032.6	967.3	965.2
2.9	1167.4	1177.2	1171.1
15.7	2139.0	2122.1	2125.9
4.9	1651.3	1649.8	1642.9
23.1	1281.5	1268.5	1268.0

(B)

Both tables show that BMICA-WT possesses the smallest MSE. On average CubICA has the closest MSE to this algorithm. Further investigations were done on PSNR shown in Table 8.11. Examination shows that BMICAW-WT has the highest PSNR indicating that the reconstruction is of a higher quality and therefore considered a good algorithm.

Table 8.11: PSNR for (A) Fixed-Point Algorithms (B) Non-fixed Point Algorithms

BMICA/WT	FASTICA	PEARSON	EFICA
34.4	16.0	15.9	16.0
32.7	17.1	16.9	16.9
35.8	18.2	18.4	18.4
31.6	18.3	18.3	18.4
36.8	19.3	19.3	19.4
35.0	17.4	-	17.5
32.3	16.9	16.8	17.0
32.4	18.3	-	18.2
43.5	17.4	17.4	17.4
36.2	14.9	14.9	14.9
41.2	16.0	16.0	15.9
35.6	17.2	17.1	17.3

(A)

BMICA/WT	JADE	SOBI	CUBICA
34.4	15.9	15.9	15.9
32.7	17.0	17.0	17.0
35.8	18.4	18.3	18.3
31.6	18.3	18.5	18.5
36.8	19.3	19.3	19.3
35.0	17.5	17.5	17.6
32.3	17.0	17.0	17.0
32.4	18.0	18.3	18.3
43.5	17.5	17.4	17.4
36.2	14.8	14.9	14.9
41.2	16.0	16.0	16.0
35.6	17.2	17.3	17.3

(B)

When calculating SNR the greater the ratio, evidenced by a larger number, the less noise and the more easily it can be filtered out. Does BMICA-WT produce the highest SNR of all the algorithms? Table 8.12 shows data that seeks to answer the question. On average BMICA-WT produces the highest SNR of all the seven algorithms.

Table 8.12: SNR for (A) Fixed-Point Algorithms (B) Non-fixed Point Algorithms

BMICA/WT	FASTICA	PEARSON	EFICA
1.89	0.11	-0.08	0.04
2.43	0.02	0.03	0.10
1.98	-0.12	-0.03	-0.09
1.78	-0.14	0.01	0.07
1.88	-0.07	0.02	-0.03
2.10	-0.13	0.20	0.02
2.10	-0.09	-0.12	0.01
1.13	0.08	-0.05	0.01
1.32	0.03	0.00	-0.06
1.29	0.01	0.00	-0.01
1.79	-0.03	0.00	0.01

(A)

BMICA/WT	JADE	SOBI	CUBICA
1.89	0.03	0.00	-0.07
2.43	-0.08	0.00	0.05
1.98	-0.17	0.00	0.09
1.78	-0.10	0.00	0.08
1.88	0.10	0.00	0.04
2.10	0.07	0.00	0.00
2.10	0.05	0.00	-0.04
1.13	0.01	0.00	-0.01
1.32	-0.02	0.00	0.01
1.29	0.00	0.00	0.03
1.79	-0.01	0.00	0.02

(B)

8.4.1.3 Computational Cost

I showed in Chapter 4 that BMICA has a complexity of $O(N^2M)$ resulting in a complexity comparable to FastICA, Infomax, JADE and EFICA. Section 8.2 showed that the complexity of TIWT is $O(N^2 \log N)$. When these two algorithms are merged as described in this chapter the complexity results in $O(N^2M) + O(N^2 \log N) = O(N^2 \log N)$ – the complexity of TIWT.

Investigations on running time show however that TIWT completes the denoising process faster than BMICA-WT. Like in Chapter 4 I conclude that the presence of the B-Spline iterative steps influences the speed of the BMICA step in the BMICA-WT algorithm.

8.5 Summary

Research has found that WT is the best suited for denoising as far as performance goes because of its properties like sparsity, multiresolution and multiscale nature. Non-orthogonal wavelets such as UDWT and Multiwavelets improve the performance at the expense of a large overhead in their computation [117]. Research also shows that TIWT is considered to be an improvement on WT, removing Gibbs phenomena. In this work I investigated the performance of TIWT when denoising EEG. I found that TIWT performed better than the ICA algorithms tested for all performance measures examined.

Further improvement of the TIWT with the merger with UKF and ICA produced CTICA. Although CTICA had a longer running time, it also produced better performance than the tested ICA algorithms as

- (i) It has outperformed FastICA and EFICA as far as MSE, and PSNR were concerned for both real and artificial signals
- (ii) It has the best SDR for both real and artificial signals
- (iii) It has the best Amari index for signals greater than 2^7 in size which decreases as sample size increases and

(iv) Its complexity is similar to the other algorithms.

When investigating the effects of the addition of BMICA to TIWT I found

- (i) BMICA-WT produces a better quality reconstructed signal 100% of the time than all algorithms tested for PSNR and MSE
- (ii) BMICA-WT produces higher SDR 65% of the time TIWT produces lower SIR 94% of the time.
- (iii) BMICA-WT produces higher SNR and lower SIR than all the ICA algorithms tested
- (iv) BMICA-WT produces a smaller Amari Index for all tested signal sizes
- (v) Time complexity of BMICA-WT is the same as TIWT which is better than the tested ICA algorithms and
- (vi) The running time of BMICA-WT is higher than TIWT due to the presence of the B-Spline function.

I cannot therefore be conclusive of BMICA-WT as far as separation accuracy was concerned when comparing with TIWT but overall examination of all four measures show BMICA-WT outperforming TIWT in three measures – SDR, MSE and PSNR. This indicates that it is a better denoising algorithm. I found however that for SNR, MSE, PSNR, Amari Index and SIR BMICA-WT outperformed all tested ICA algorithms.

To answer the questions posed in the introduction of section 8.4 therefore I found that the merger of TIWT and ICA improve the denoising quality of the reconstructed signals. Investigations went on to confirm that the merger of BMICA and TIWT produced better quality results than basic TIWT.

In this chapter I investigated the performance of TIWT in denoising EEG signals as well as improving its performance by mergers. I have found that TIWT is an efficient technique and it can be improved when it becomes CTICA and BMICA-WT. Although both are slower than the tested ICA algorithms, they produce more

accurate results. Users will now have the choice to trade either “*speed for accuracy*” or “*accuracy for speed*”.

Chapter 9 – Discussion and Conclusion

9.1 Summary

Eye movements, eye blinks, cardiac signals, muscle noise, sucking movement, and line noise present serious problems for EEG interpretation and analysis when rejecting contaminated EEG segments results in an unacceptable data loss. Many methods have been proposed to remove artifacts from EEG recordings. Often regression in the time or frequency domain is performed on parallel EEG and artifacts recordings to derive parameters characterizing the appearance and spread of artifacts in the EEG channels. Because EEG and artifacts activity mix bidirectionally, regressing out artifacts inevitably involves subtracting relevant EEG signals from each record as well. Regression methods become even more problematic when a good regressing channel is not available for each artifact source, as in the case of muscle artifacts. Use of PCA has been proposed to remove the artifacts from multichannel EEGs. However, PCA cannot completely separate artifacts from brain signals, especially when they have comparable amplitudes. In this dissertation, I proposed three new and generally applicable methods for removing a wide variety of artifacts from EEG records based on BSS by ICA and WT.

9.2 Link to Dissertation Goals

9.2.1 Algorithm Features

The methods created in dissertation were designed to remove artifacts from EEG signals and were to have the following features as stated in Chapter 1:

robustness, accuracy, adaptability, and convergence. I produced BMICA, CTICA and BMICA-WT which presented these features.

9.2.1.1 BMICA

The use of B-Spline to create a MI estimator resulted in a MI estimator which according to Klien *et al* [82] is robust and accurate. Daub *et al.* [27] also showed that a B-Spline based MI estimator produces more accurate results than other MI estimator. A B-Spline MI estimator does not need to grow exponentially to provide accurate estimations [136]. Research shows therefore that B-Spline MI should provide more accurate processing which implies that BMICA should perform better than other ICA algorithms.

Robustness

To be robust BMICA should remove noise from the EEG data. Chapter 4 shows that this is done. It out performs most of the other known algorithms used to denoise EEG signals (FastICA, Pearson_ICA, EFICA, SOBI, Infomax and JADE) when tested using SNR, PSNR and MSE. BMICA is robust.

Accuracy

Separation accuracy measres usng Amari Indx, SIR and SDR indicated that BMICA out performed known ICA algorithms. Further testing based on ICASSO described in Chapter 5 showed that BMICA produced clearer and more reliable denoised signals than FastICA, a reknown ICA algorithm. BMICA produces accurate signals. BMICA is accurate.

Adaptability

BMICA was tested on datasets of different sizes ranging from 4 to 16384; all in powes of 2 i.e. 2^x . It performed efficiently in all sizes. Amari Index (Figure 4.5) shows that BMICA performed in a similar behavior pattern as the other tested ICA algorithms. BMICA is adaptive.

Convergence Speed

FastICA's convergence speed has been the topic of much research [122-123] and in my research I have therefore compared the convergence speed of BMICA with it. FastICA is quadratic in general and cubic for symmetric distributions [57]. Investigations have shown that my algorithm has a slower speed than FastICA. Comparison of the running time has also shown that FastICA completes its analysis much faster than BMICA. In both instances I conclude that the presence of the B-Spline iterative steps influences the speed of the BMICA as on average B-Spline calculates slowly. This means that although BMICA is able to converge, its speed is slower than that of FastICA.

The complexity of BMICA $O(N^2M)$ I found however is comparable to FastICA, Infomax $O(N^3M)$ and JADE $O(N^4M)$. All this shows that BMICA converges.

9.2.1.2 CTICA

The creation of CTICA created another algorithm which was robust as well as accurate. Chapter 8 shows that this algorithm does remove noise from the presented EEG signals indicating that the algorithm is robust. Investigations against other known denoising algorithms showed that this algorithm is not only robust but also accurate.

Robustness

Chapter 8 shows that CTICA is successful in removing noise from the EEG datasets tested as seen in Figure 8.7. Testing and comparison with known ICA algorithms using noise/signal measures of MSE and PSNR confirms this denoising. These also show that CTICA outperform other ICA algorithms. CTICA is robust.

Accuracy

To confirm the accuracy of the CTICA the SDR and Amari Index were determined. In both cases CTICA out performed the other algorithms. CTICA is accurate.

Adaptability

Like BMICA this algorithm's adaptability was tested using Amari Index and compared with those of known algorithms. This showed that as the sample size increased CTICA produced more accurate results. CTICA is adaptive.

Convergence Speed

Because CTICA has to perform a filter operation as well as an ICA algorithm it cannot converge as fast as a single ICA algorithm or even the basic TIWT. It however has a complexity that is similar to FastICA and Infomax - $O(N^3M)$. CTICA converges.

9.2.1.3 BMICA-WT

This algorithm is a merger of the new BMICA and TIWT. This algorithm has been shown to pose all the features of the basic BMICA algorithm.

Robustness

As a merger of TIWT and BMICA, algorithms shown in this dissertation to be robust it can be assumed that it follows for BMICA-WT. I tested this assumption however in Chapter 8 using MSE and PSNR to confirm. BMICA-WT is robust.

Accuracy

Chapter 8 showed testing using Separation Accuracy Measures SIR and SDR. This shows that BMICA-WT does remove noise from the tested EEG datasets. BMICA-WT is accurate.

Adaptability

I have shown that both BMICA and TIWT are both robust and accurate and I have shown the same for BMICA-WT. Adaptability therefore follows the same pattern. BMICA-WT is adaptive.

Convergence Speed

Chapter 8 shows that BMICA-WT has the same complexity as TIWT – $O(N^2 \log N)$. Convergence speed is based on the B-Spline influence as described in Chapter 4. BMICA-WT converges.

9.2.1.4 Conclusion on Algorithm Features

I have created three algorithms – BMICA, CTICA and BMICA-WT, which possess the four features, described in Chapter 1 of robustness, accuracy, adaptability and possess a convergence speed. These features are necessary for more efficient ICA algorithms. I have therefore met my dissertation goals.

9.2.2 Level of Algorithm Performance

The three algorithms were also able to produce a level of performance that answers the described four areas in Chapter 1 namely:

1. How independent the “independent” components are
2. The uniqueness of the components.
3. The robustness of the estimated dependencies against outliers and artifacts.
4. The robustness of the estimated components.

I found in my research that BMICA, CTICA and BMICA-WT, like all the known ICA utilized here, answered areas 2 and 4. BMICA and BMICA-WT were created using MI estimations. Literature has shown that MI is an obvious candidate for measuring this independence [89, 159] thus answering area 1. BMICA and BMICA-WT therefore should provide an answer for area 1. Hyvarinen [59] stated that the present MI estimated ICA algorithms are far from optimal as far as robustness and

asymptotic variance are concerned. These algorithms are also sensitive to artifacts thus as they are now these algorithms cannot answer neither areas 1 or 3.

I created BMICA using B-Spline which in my research was found to be the second best non-parametric approach to MI-estimation, outperforming MI estimators such as k NN and KDE. B-Spline focuses on each IC allowing the MI estimation to create ICs in the ICA algorithm allowing my algorithm to answer area 1. Comparison of my algorithm with other MI estimated algorithms in Chapter 6 showed that it performed better in both separation accuracy and noise/signal measures. My research also showed that it was better in both areas as well when compared to non-MI based algorithms. B-Spline therefore allows for a better performance as well as robustness thus answering area 3.

Testing of BMICA using ICASSO showed that it produces more robust ICs than FastICA, which is more reliable than EASI, Infomax, Pearson_ICA, JADE, EGLD and ML. BMICA and ultimately BMICA-WT produces more robust and clearer estimated components answering area 3. My algorithms BMICA and BMICA-WT therefore answered all four areas.

The use of WT answers areas 1 and 2. The use of UKF and a robust ICA increases the algorithms performance. The algorithm of choice for the creation of CTICA answers 2 and 4. In my research it was found that this new algorithm now answers area 3. CTICA therefore answers like BMICA, BMICA-WT all four areas used to determine the level of performance for any ICA algorithm.

9.3 Conclusion

Comparison of my three algorithms with known ICA algorithms such as FAsTICA, SOBI and JADE have shown that they have

- (i) Better separation accuracy measures
- (ii) Better noise/signal measures and

(iii) Comparable complexity

BMICA and BMICA-WT however have slower convergence speed than the other algorithms. My algorithms are therefore more robust, accurate and adaptive. The three algorithms also address the four areas of algorithm performance described in Chapter 1. They all produce a level of performance that answers all four areas. Users can now choose ICA algorithms that are accurate, reliable and possess all the desired level of performance.

9.4 Actual Contribution of Dissertation

The scientific contributions of this dissertation include the following:

- A fast fixed-point ICA algorithm for separating mixed complex valued EEG source signals is presented and the local consistency of the estimator given by the algorithm is proved.
- The use of B-Spline Mutual Information estimator to create a new ICA algorithm. Empirical validation is presented in and compared with known ICA algorithms. This has not been presented elsewhere.
- The reliability of the new fast fixed-point ICA algorithm is proved using empirical validation.
- Experimental results are given on the performance of MI based ICA algorithms versus non MI based ICA algorithms on biosignals (EEG).
- Empirical validation of using TIWT as a denoising method for biosignals (EEG) is given.
- The creation of a bridge by comparing the performance of EKF to UKF when applied to EEG signals, especially since there were only investigations on the accuracy of UKF for nonlinear, nonstationary systems not including EEG.

- Experimental results are given using TIWT, UKF and an ICA method as a method of artifact reduction. These experimental results have not been presented elsewhere.
- The merger of the new B-Spline MI based ICA algorithm with TIWT as a method of artifact reduction. Empirical validation is presented using biosignals in the form of EEG.
- The presentation of different ways of improving the performance of TIWT. Empirical validation is presented using biosignals in the form of EEG.

9.5 Implications

ICA denoising is important for the detection of epileptic seizures which range from the shortest lapse in attention to severe, frequent convulsions. They can occur from several times a day to once every few months. These seizures are caused by bursts of excessive electrical activity in the brain and these epileptic components are morphologically very similar to ECG artifact in EEG. ECG artifacts occur as a consequence of cardiac electrical field that affects the surface potential near the scalp. Therefore it is important to distinguish between ECG artifact and epileptic components by extracting ECG artifact from EEG signals. The running of an inadequate ICA algorithm might lead to removal of important EEG components, false or missed epileptic detection. My algorithm BMICA and its derivative have been found to produce clearer EEG signals than the other known ICA algorithms used to denoise EEG signals. This indicates that these denoised signals should be more accurate for the detection of epileptic seizures. This also stands for all other diseases that are detected by EEG recordings.

I found that the B-Spline based MI approach to ICA is extremely rewarding, its superior performance over existing methods is very encouraging. BMICA therefore shows that B-Spline is a good method in creating more accurate denoising techniques.

I found also that although TIWT is an improvement on the use of WT when denoising it was not the final stage. My research shows that it can be further improved to produce more accurate and clearer signals which can help in diagnosis and neuro- research.

9.6 Further Work

From the point of view of the ICA community, the estimator for the contrast function has been studied extensively during the past years and it is natural to set the focus on new problem settings. The extensions presented in this dissertation are by far not the only possible ones, nor are they yet conclusively studied here.

In this study I introduced a new ICA algorithm based on MI estimation based on B-Spline. All investigations for this fixed point ICA algorithm have been based on the linearity $g(u) = \tanh$ and a symmetric orthogonalization, resulting in all tests executed on only one configuration. Fixed point algorithms have four linearities and two modalities resulting in eight configurations. Further research can focus on investigations of BMICA performance using the other seven configurations.

The field of biosignal processing has been growing at a very fast pace. This field focuses on all biosignals such as ECG and EMG. BMICA, CTICA and BMICA-WT however were created and tested using only EEG signals. These algorithms show a great improvement in the resulting EEG signals. The performance of these algorithms on other biosignals was not the research of this work. These offer another avenue for research, as these other biosignals produce vital information of medical and research importance yet like EEG are contaminated with noise.

Further study is needed in the area of WT. Recall that denoising was only performed using wavelet Sym8. It maybe possible to implement CTICA using different wavelet families with changing vanishing moments. In addition, it maybe also possible to research results using different types of thresholds, including hybrids, other than the universal threshold. They were untouched in my study.

The application of recursive cycle spinning was not researched in this work. This offers another avenue of research. Recursive cycle-spinning can be combined with filters as another avenue of promise. Optimization of the recursive cycle-spinning technique for each level of decomposition may also be a viable area of study.

CTICA and BMICA-WT both have shown that ICA algorithms greatly improve the results of WT and TIWT respectively. I leave for further work a hybrid scheme that has the better characteristics of both.

REFERENCES

1. Akin M., 2002. Comparison of Wavelet Transform and FFT Methods in the Analysis of EEG Signals, *Journal of Medical Systems* 26(3): 241-247
2. Alfaouri, M., and Daqrouq, K. 2008. ECG Signal Denoising By Wavelet Transform Thresholding. *American Journal of Applied Sciences* 5(3):276-281
3. Amari, S., Chichocki, A., and Yang, H.H., 1996. A new learning algorithm for blind source separation. *Advances in Neural Information Processing (NIPS'95)*:757–763.
4. Banani, S.A., and M. A. Masnadi-Shirazi, 2007. A New Version of Unscented Kalman Filter World Academy of Science, Engineering and Technology 26:192-197.
5. Bar-Joseph, Z., Gerber, G.K., Gifford, D.K., Jaakkola, T.S. and Simon, I., 2003. Continuous representations of time-series gene expression data. *Journal of Computational Biology*, 10(3-4): 341–356.
6. Barsanti Jr. R.J., 1996. “Denoising of ocean acoustic signals using wavelet-based techniques”, MSc Thesis, Naval Postgraduate School, Monterey, California.
7. Bartoli, F., and Cerutti, S., 1983. An optimal linear filter for the reduction of noise superimposed to the EEG signal, *Journal of Biomedical Engineering* 5 (4):274-280.
8. Bell, A.J. and Sejnowski, T.J., 1995. An Information-Maximization Approach to Blind Separation and Blind Deconvolution”, *Neural Computation*, 7(6):1129-1159.
9. Belouchrani, A., Abed-Meraim, K., Cardoso, J.F., and Moulines, E. 1993. Second-Order Blind Separation of Temporally Correlated Sources”. *International Conference on Digital Signal Processing*:346–351.
10. Bhasi, K., Forrest, A., and Ramanathan, M., 2005. SPLINDID: a semiparametric, model-based method for obtaining transcription rates and gene regulation parameters from genomic and proteomic expression profiles, *Bioinformatics* 21(20):3873–3879

11. Bhatti, M.I., Pervaiz, A., and Baig, M.H., 2001. EEG Signal Decomposition and Improved Spectral Analysis Using Wavelet Transform, *23rd Engineering in Medicine and Biology Society* 2:1862-1864.
12. Blaschke T. and Wiskott, L., 2004. CuBICA: Independent Component Analysis By Simultaneous Third- And Fourth-Order Cumulant Diagonalization, *IEEE Transactions on Signal Processing* 52 :1250-1256,
13. Blume, W.T., Kaibara, M. and Young, G.B., 2002. Atlas of Adult Electroencephalography. Eds; Lippincott Williams and Wilkins, Philadelphia.
14. Bownds, M. D. 1999. Biology of the Mind: Origins and Structures of Mind, Brain, and Consciousness, eds *John Wiley & Sons*
15. Cardoso, J. 1998. Blind Signal Separation: Statistical Principles. In *Proceedings of the IEEE* 86(10):2009-2025.
16. Carson M. and Walter. R.D. 1961. The Electroencephalogram in Pediatrics: The Indications for Use and the Limitations. *California Medicine* 94(4):225-229
17. Chavan, M., Mastorakis, N., Chavan, M.S., and Gaikwad, M.S., 2011. Implementation of Symlet Wavelets to Removal of Gaussian Additive Noise from Speech Signal”, *Recent Researches in Communication, Automation, Signal Processing, Nanotechnology, Astronomy and Nuclear Physics*:37-41
18. Chien, J.T., and Chen B.C. 2006. A New Independent Component Analysis for Speech Recognition and Separation, *IEEE Transactions on Audio, Speech and Language Processing* 14(4):1245-1254.
19. Chien, J., Hsin-Lung H., and Furui, S., 2008. A New Mutual Information Measure for Independent Component Analysis, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*:1817 – 1820
20. Chiappa, S. 2006, Analysis and Classification of EEG Signals using Probabilistic Models for Brain Computer Interfaces, PhD Thesis
21. Coifman, R.R., and Donoho, D.L. 1995. Translation-Invariant De-Noising, *Lecture Notes in Statistics: Wavelets and Statistics*:125-150.

22. Cohen, B.C., and Sances Jr., A., 1997. Stationarity of the Human Electroencephalogram, *Medical and Biology Engineering and Computing* 15:513-518.
23. Comon, P., 1994. Independent Component Analysis, a New Concept? *Signal Processing, Elsevier*, 36(3):287-314
24. Croft R.J., and Barry, R.J. 2000. EOG Correction: Which Regression Should We Use? *Psychophysiology* 37:123-125.
25. Croft, R.J., and Barry, R.J., 2000. Removal of Ocular Artifacts from the EEG: A Review, *Clinical Neurophysiology* 30(1):5-19
26. Cruz, J., Pedroza, J., Altamirano, L., and Olivera, I., 2006. A Performance Comparison of Estimation Filters for Adaptive Imagery Tracking, *Signal Processing, Pattern Recognition, and Applications* (SPPRA 2006):20-25.
27. Daub C.O., Steuer, R., Selbig, J., and Kloska, S. 2004. Estimating Mutual Information Using B-Spline Functions—An Improved Similarity Measure for Analysing Gene Expression Data. *BMC Bioinformatics*, 5(1): 118.
28. DiMatteo, I., Genovese, C.R., and Kass, R.E., 2001, “Bayesian Curve Fitting with Free-Knot Splines,” *Biometrika* 88, 1055– 1073.
29. Donoho, D.L., and Johnstone, I.M., 1995. Adapting to Unknown Smoothness via Wavelet Shrinkage, *Journal of the American Statistical Association*, 90(32):1200-1224.
30. Emuir, U.E., Akin, A., Ertuzun, A., Sankur B., and Harmanci, K., 2003. Wavelet Denoising vs ICA Denoising for Functional Optical Imaging, *Proc. 1st Int. IEEE EMBS Conference on Neural Engineering*:384-387.
31. Errity, A., McKenna, J., and Isard. S., 2004. Unscented Kalman Filtering of Line Spectral Frequencies. In the *Proceedings of 8th International Conference of Spoken Language Processing (Interspeech-ICSLP 2004)*:2697-2700.
32. Ferguson, B., and Abbott, D. 2001. Denoising Techniques for Terahertz Response of Biological Samples, *Microelectronics Journal* 32:943-953.
33. Ferrez. P.W. 2007. Error-Related EEG Potentials in Brain-Computer Interfaces, Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland.

34. Fletcher, A.K., Ramchandran, K. and Goyal, V.K. 2002. Wavelet Denoising by Recursive Cycle Spinning, 2002. *IEEE International. Conference on Image Processing 2*, pp. 873–876.
35. Francois, D., Rossi, F., Wertz, V., and Verleysen, M., 2007. Resampling Methods for Parameter Free and Robust Feature Selection with Mutual Information, *Neurocomputing* 70:1276-1288.
36. Friedman J.H., 1987. Exploratory Projection Pursuit, *Journal of the American Statistical Association* 82(397):249-266.
37. Friston K.J., Holmes A.P., Worsley K.J., Poline J.-P., Frith C.D., and Frackowiak R.S.J. 1995. Statistical Parametric Maps in Functional Imaging: A General Linear Approach, *Human Brain Mapping* 2:189–210.
38. Fugal, D. L. 2009. Conceptual Wavelets in Digital Signal Processing: An indepth Practical Approach for the Non-Mathematician, *Space & Signals Technologies LLC*.
39. Georgescu V. and Matei, A.D. 2011. Seeking for Fundamental Factors behind the Co-Movement of Foreign Exchange Time Series, using Blind Source Separation Techniques, In *Recent Researches in Artificial Intelligence, Knowledge Engineering and Data Bases*:243-248
40. Ghael, S.P., Sayeed A.M., and Baraniuk, R.G. 1997. “Improved Wavelet Denoising via Empirical Wiener Filtering”, *Proc. of SPIE*, 3169:389-399
41. Girimonte, D., Izzo D. and Bergamin, L. 2006. Reasoning Under an Uncertain Thermal State *Proceedings of the 57th International Astronautical Congress*.
42. Girton D.G., and Kamiya J, 1973. A Simple On-Line Technique For Removing Eye Movement Artifacts From The EEG, *Electroencephalography and Clinical Neurophysiology* 34:212-216
43. Glass, K., Frishkoff, G.A., Frank, R.M., Davey, C., Dien, J., Malony, A.D., and Tucker D.M. 2004. A Framework for Evaluating Independent Component Analysis Methods of Artifacts Removal from Multichannel EEG. *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA2004)*, *Lecture Notes in Computer Science* 3195/2004:1033-1040.
44. Gómez-Herrero, G., and Egiazarian, K., 2005. Independent Component Analysis by a Resampling Strategy, *Technical Report 2005*, Retrieved

45. Graps, A. 1995. An Introduction to Wavelets. *IEEE Journal of Computational Science and Engineering* 2(2):1-17.
46. Gribonval, R., Vincent, E., and Févotte, C. 2003. Proposals for Performance Measurement in Source Separation. *4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*:763–768.
47. Groppe, D.M., Makeig, S., and Kutas, M. 2009. Identifying Reliable Independent Component via Split-half Comparisons, *Neuroimage* 45:1199-1211.
48. Guerin-Dugue, A., Teissier, P., Gafaro, G.D., and Herautt, J. 1999. Curvilinear Component Analysis for High Dimensional Data Representation: II Examples of Additional Mapping Constraints in Specific Applications, *Engineering Applications of BioInspired Artificial Neural Networks* 1607:635-644.
49. Harmeling, S., 2004. Injecting Noise for Analysing the Stability of ICA Components, *Signal Processing*, 84(2):255-266.
50. Hastie, T. and R. Tibshirani, 2002. Independent Component Analysis through Product Density Estimation, *Neural Information Processing Systems (NIPS)*.
51. Hawwar, Y.M. Reza, A.M., and Turney, R.D. 2002. Filtering(Denoising) in the Wavelet Transform Domain, Unpublished, Department of Electrical Engineering And Computer Science, University of Wisconsin-Milwaukee.
52. He, W., 2002. A Spline Function Approach for Detecting Differentially Expressed Genes in Microarray Data Analysis. *Bioinformatics* 20(17):2954–2963.
53. Hegyi, A., Girimonte, D., Babuška, R., and Deschutter, B. 2006. A Comparison of Filter Configuration for Freeway Traffic State Estimation, *IEEE Intelligent Transportation Systems Conference (ITSC 06)*:1029-1034.
54. Herrmann, C.S., Grigutsch M., and Busch, N.A. 2005. EEG oscillations And Wavelet Analysis, Event-related potentials: A Methods Handbook eds MIT Press:229-259

55. Himberg, J. and Hyvarinen, A. 2003. ICASSO: Software for Investigating The Reliability Of ICA Estimates By Clustering And Visualization. *IEEE Workshop on Neural Networks for Signal Processing (NNSP2003)*:259-268.
56. Himberg, J., Hyvarinen, A. and Esposito, F. 2004. Validating the Independent Components Of Neuroimaging Time-Series via Clustering and Visualization, *Neuroimage* 22(3):1214-1222.
57. Hoffman, S., and Falkenstein, M. 2008. "The Correction of Eye Blink Artefacts in the EEG: A Comparison of a Two Prominent Methods", *PLoS One* 3(8):e3004.
58. Hyvarinen, A., 1999. Fast and robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks* 10 (3): 626 – 634.
59. Hyvärinen, A. 1999. Survey on Independent Component Analysis. *Neural Computing Surveys* 2: 94-128.
60. Hyvärinen, A., and Bingham, E. 2000. A Fast, Fixed-Point Algorithm for Independent Component Analysis of Complex Valued Signals. In the *Proceedings of the International Journal of Neural Systems* 10(1):1-8.
61. Hyvärinen, A., and Oja E. 2000. Independent Component Analysis: Algorithms and Applications. In *Neural Computation* 13: 411-430.
62. Hyvarinen, A., Karhunen J., and Oja, E. 2001. Independent Component Analysis, eds. Wiley & Sons
63. Inuso, G., La Foresta, F. Mammone, N., and Morabito, F.C. 2007. Wavelet-ICA Methodology for Efficient Artifact Removal from Electroencephalographic Recordings, *International Joint Conference on Neural Networks* :1524-1529
64. Jasper, H.H. 1958. Report of the Committee on Methods of Clinical Examination in Electroencephalography. *Electroencephalography and Clinical Neurophysiology*. 10:370-373.
65. Julier, S., Uhlmann, J.K., and Durrant-White, H. 2000. A New Method for Nonlinear Transformation of Means and Covariances in Filters and Estimators, *IEEE Transactions on Automatic Control*, 45:477-482.

66. Julier, S., and Uhlmann, J.K, 1997. A New Extension of the Kalman Filter to Nonlinear Systems” *Proc. AeroSense: 11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*:182-193.
67. Julier, S., and Uhlmann, J.K, 2004, Unscented Filtering and Nonlinear Estimation”, *Proc. IEEE* 92(3): 401-421.
68. Jutten, C., and Herault, J. 1986. Space or Time Adaptive Signal Processing by Neural Models, *Proceedings of AIP Conference: Neural Networks for Computing* 151:206-211.
69. Jutten, C., and Herault, J., 1991, “Blind Separation Of Sources: Part I: Anadaptive Algorithm Based on Neuromimetic Architecture,” *Signal Processing* 24:1 –10.
70. Jutten C., Herault, J. and Comon P. 1991. Blind Separation of Sources Part II: Problem Statement, *Signal Processing* 24:11-20
71. Jung, T.P., Humphries, C., Lee, T., Makeig, S., McKeown, M.J., Iragui, V., Sejnowski, T.J. 1998. Removing Electroencephalographic Artifacts: Comparison between ICA and PCA, *IEEE Workshop on Neural Network for Signal Processing VIII*:63-72.
72. Jung, T.P., Makeig, S., Bell, A.J., and Sejnowski, T. J. 1997. Independent Component Analysis of Electroencephalographic And Event-Related Potential Data. In P. Poon & J. Brugge (Eds.), *Central Auditory Processing and Neural Modeling*:189–197.
73. Jung, T.P., Makeig, S., Westerfield, M., Townsend, J., Courchesne, E., and Sejnowski, T.J. 2001. Analyzing and Visualization of Single-Trial Eventrelated Potentials. In *Human Brain Mapping* 14:166-185
74. Jung, T.P. Humphries, C., Lee, T.W., McKeown, M.J., Iragui, V., Makeig, S., and Sejnowski, T.J. 2000. Removing Electroencephalographic Artifacts by Blind Source Separation. *Psychophysiology*,37:163-178.
75. Kachenource, A., Albera, L. Senhadji, L., Comon, P. 2008. ICA: A potential Tool for BCI Systems, *IEEE Signal Processing Magazine*:57-68.
76. Kallapur, A. Anavatti, S., and Garratt, M. 2008. Extended and Unscented Kalman Filters for Attitude Estimation of an Unmanned Aerial Vehicle, *Proc. 27th IASTED Int. Conf. Modelling, Identification, and Control (MIC 2008)*.

77. Kandaswamy, A., Krishnaveni, V., Jayaraman, S., Malmurugan N., and Ramadoss, K. 2005. Removal of Ocular Artifacts from EEG - A Survey” IETE Journal of Research, 51(2):121-130.
78. Karmeshu, J. 2003, Entropy Measures, Maximum Entropy Principle and Emerging Applications (Studies in Fuzziness and Soft Computing), Springer-Verlag
79. Karvanen, J., Eriksson, J., and Pearson, K.V. 2000. System Based Method for Blind Separation. *2nd International Workshop on Independent Component Analysis and Blind Signal Separation*:585 - 590.
80. Keith, D.B., Hoge, C.C., Frank, R.M., and Malony, A.D. Parallel Independent Component Analysis Methods for EEG Neuroimaging. *Technical Report*, University of Oregon, Neuroinformatics Center
81. Khosla, R., Howlett, R.J., Jain, L. C. 2005. Knowledge-based intelligent information and engineering systems: 9th international conference, Springer Science & Business
82. Klein, S., Marius Staring, M., and Josien, P.W. Pluim.2005. Comparison of gradient approximation techniques for optimisation of mutual information in nonrigid registration. *SPIE, (Medical Imaging:Image Processing)* 5747:192-203
83. Knight, J.N. 2003. Signal Fraction Analysis and Artifact Removal in EEG Thesis, Department of Computer Science, Colorado State University.
84. Konatowski, S., and Pieniężny, A.T. A Comparison of Estimation Accuracy by the use of KF, EKF & UKF Filters. *Computational Methods and Experimental Measurements XIII*
85. Koski, T., 2001, Hidden Markov Models for Bioinformatics, Springer, pp 30-33
86. Kotecha, J.H., and Djurić P.M. 2003. Gaussian Particle Filtering, *IEEE Transactions on Signal Processing* 51(10):2592-2601.
87. Kotecha, J.H., and Djurić P.M. 2003. Gaussian Sum Particle Filtering, *IEEE Transactions on Signal Processing* 51(10):2602-2612.

88. Kovalchuk, I.A., and Gorbunov, A.P., 1977. Comparison of the Efficiency of Wiener and Kalman Filters in Problems of Secondary Signal-Processing. In *Radioelektronika*, 20: 99-102.
89. Kraskov, A., Stogbauer, H., and Grassberger, P. 2004. Estimating Mutual Information, In *Physical Review E*. 69(6) 066138
90. Krishnaveni, V., Jayaraman, S., and Ramadoss, K. 2006. Application of Mutual Information based Least dependent Component Analysis (MILCA) for Removal of Ocular Artifacts from Electroencephalogram *International Journal of Biomedical Sciences*, 1(1): 63-74
91. Krishnaveni, V., S. Jayaraman P.M. Manoj Kumar, K. Shivakumar, and K. Ramadoss. 2005. Comparison of Independent Component Analysis Algorithms for Removal of Ocular Artifacts from Electroencephalogram. *Measurement Science Review Journal – Measurement in Biomedicine*, 5(2):68-78
92. Krishnaveni, V., Jayaraman, S., Gunasekaran, A., and Ramadoss, K. 2007. Automatic Removal of Ocular Artifacts using JADE Algorithm and Neural Network. *International Journal of Biomedical Sciences*, 2(1):10-21
93. Koldovský, Z., and Tichavský, P. 2007. Time-Domain Blind Audio Source Separation Using Advanced ICA Methods. *8th Annual Conference of the International Speech Communication Association (Interspeech 2007)*:846-849.
94. Lagerlund, T.D., Sharbrough, F.W., and Busacker, N.E., 1997, “Spatial Filtering Of Multichannel Electroencephalographic Recordings Through Principal Component Analysis By Singular Value Decomposition,” *Journal of Clinical Neurophysiology* 14 (1), 73–82
95. Lee-Chiong, T.L. 2006. Sleep: A Comprehensive Handbook eds John Wiley & Sons
96. Lennon, M., Mercier, G., Mouchot, M.C., and Hubert-Moy, L. 2001. Curvilinear Component Analysis for Non-Linear Dimensionality Reduction of Hyperspectral Images, *Proc. of SPIE Symposium on Remote Sensing Conference on Image and Signal Processing for Remote Sensing VII* 4541:157.
97. Li, X., Lu, H., Han, G., and Liang, Z. 2001. A Noise Reduction Method for Non-Stationary Noise Model of SPECT Sinogram based on Kalman Filter. In *IEEE Nuclear Science Symposium Conference Record*:2134-2138

98. Li, X., and Adali, T. 2010. "Independent Component Analysis by Entropy Bound Minimization" *IEEE Transaction on Signal Processing*, 58(10):5151-5164
99. Liang, J., and Parks, T.W. 1994. "A Two-dimensional Translation Invariant Wavelet Representation and its Applications". *IEEE Conference on Image Processing (ICIP 94) Vol. 1*:66-70
100. Lins, O.G., Picton, T.W., Berg, P., and Scherg, M., 1993, Ocular Artifacts in Recording EEGs and Event-Related Potentials: II. Source Dipoles and Source Components," *Brain Topography* 6 (1):65– 78
101. Luan Y., and Li, H. 2003. Clustering of Time-Course Gene Expression Data Using Amixed-Effects Model with B-Splines, *Bioinformatics*, 19(4):474–482.
102. Ma, P., Castillo-Davis, C.I., Zhong, W., and Liu. J.S. 2006. A Data Driven Clustering Method for Time Course Gene Expression Data. *Nucleic Acids Research*, 34(4):1261–1269.
103. Makeig, S., Jing, T., Bell, A.J., Sejnowski, T.J. 1996. Independent Component Analysis of Electroencephalographic data. In *Advances in Neural Information Processing Systems 8*, MIT Press Cambridge MA 8: 145-151.
104. Makeig, S., Jung, T.P., Ghahremani, D., and Sejnowski, T. J. 1996. Independent Component Analysis of Simulated ERP Data. *Technical Report. INC (Institute for Neural Computation, University of California San Diego)*, No. 9606.
105. Makeig, S., Jung, T.P., Bell, A. J., Ghahremani, D., and Sejnowski, T. J. 1997. Blind Separation of Event-Related Brain Responses into Independent Components. *Proceedings of the National Academy of Sciences of the United States of America*, 94:10979–10984.
106. Makeig, S., Jung, T.P., Westerfield, M., Covington, J., Townsend, J., Sejnowski, T. J., and Courchesne, E. 1999. Independent Components of the Late Positive Response Complex in a Visual Spatial Attention Task. *Journal of Neuroscience*, 19: 2665–2680.
107. Mantini, D., Hild II, K.E., Alleva G., and Comani, S., 2006. Performance Comparison on Independent Component Analysis Algorithms for Fetal Cardiac Signal Reconstruction: A Study on Synthetic fMCG Data. *Physics in Medicine and Biology* 51:1033-1046

108. Marques, G.C., and Almeida, L.B. 1996. An Objective Function for Independence, *International Conference on Neural Network*:453-457.
109. Marques, G.C., and Almeida, L.B, 1999. Separation of Nonlinear Mixtures Using Pattern Repulsion, *1st International Workshop on Independent Component Analysis and Signal Separation*:277-282.
110. Mastriani M. 2009. Denoising and Compression in Wavelet Domain via Projection onto Approximation Coefficients, *International Journal of Signal Processing* 5(1):20-30
111. Mastriani, M., and Giraldez, A.E. 2006. Kalman's Shrinkage for Wavelet-Based Despeckling of SAR Images, *International Journal of Intelligent Systems and Technologies* 1(3):190-196
112. Meinecke, F., Ziehe, A., Kawanabe, M., Muller, K.R. 2002. A Resampling Approach to Estimate The Stability Of One- Or Multidimensional Independent Components. *IEEE Transactions on Biomedical Engineering* 4(12):1514-1525.
113. Miller, E., and Fisher, J., 2003. ICA using Spacings Estimates of Entropy. *Journal of Machine Learning Research* 4:1271-1295.
114. Mitianoudis, N. and Davies, M. 2001. New Fixed-Point ICA Algorithms for Convolved Mixtures. *3rd International Conference on Independent Component Analysis and Blind Source Separation*:633-638.
115. Molgedey L., and Schuster, H., 1994. "Separation Of Independent Signals Using Time-Delayed Correlations," *Physical Review Letters*, 72(23):3634-3637.
116. Morbidi, F., Garulli, A., Prattichizzo, D., Rizzo, C., and Rossi, S. 2007. A Kalman Filter Approach to Remove TMS-induced Artifacts from EEG Recordings, In *Proceedings of the European Control Conference* :2201-2206
117. Motwani, M.C., Gadiya M.C., Motwani, R.C., and Harris Jr., F.C. 2004. Survey of Image Denoising Techniques, In *Proceedings of the Global Signal Processing Expo and Conference (GSPx)*: 27-30
118. Mutihac R., 2005. *Exploratory versus Confirmatory Analysis of fMRI data*, *ESMRMB 2005*, Functional MRI, Basel, Switzerland: S203-204.

119. Mutihac R. and Mutihac R.C., 2007. A Comparative Study of Independent Component Analysis Algorithms for Electroencephalography, *Romanian Reports in Physics* 59(3):831-860.
120. Nicolaou, N. and Nasuto, S.J., 2003. Comparison of Temporal and Standard Independent Component Analysis Algorithms for EEG Analysis. In the *Proceedings of the Joint 13th International Conference on Artificial Neural Networks and 10th International Conference on Neural Information Processing (ICANN/ICONIP03)*.
121. Nikolaev, N., and Gotchev, A., 2000. ECG Signal Denoising using Wavelet Domain Wiener Filtering, *10th European Signal Processing Conf. (EUSIPCO 2000)*.
122. Oja, E., 2002. Convergence of the Symmetrical FastICA Algorithm. *9th International Conference on Neural Information Processing (ICONIP '02)* 3:1368 – 1372.
123. Oja, E., and Z. Yuan, Z. 2006. The FastICA Algorithm Revisited: Convergence Analysis. *IEEE Transactions on Neural Networks* 17(6): 1370-1381.
124. Ollila, E. 2009. On the Robustness of the Deflation-based FastICA Estimator, *IEEE/SP 15th Workshop on Statistical Signal Processing*:673-679
125. Overton D.A., and Shagass, C. 1969. Distribution of Eye Movement and Eye Blink Potentials over the Scalp. *Electroencephalography and Clinical Neurophysiology*, 27:546.
126. Pandey, S., Biller, N., and Turkmen, A. 2005. The Effects of Outliers in Independent Component Analysis. In the *Proceedings of the 12th International Conference on Statistics, Combinatorics, Mathematics and Applications*.
127. Paulchamy, B. Ilavennila, Jaya, J., Saravanakumar, R. 2010. Comparative Evaluation of Various Independent Components (ICA) Tech for the Removal of Artifacts of EEG Signals, In *IJCSNS International Journal of Computer Science and Network Security*, 10(3):226-234.
128. Pham, D. T., and Garrat, P. 1997. Blind Separation Of Mixture Of Independent Sources Through a Quasi-Maximum Likelihood Approach. *IEEE Transactions on Signal Processing* 45(7):1712-1725.

129. Poornachandra, S. and Kumaravel, N. 2006. Subband-Adaptive Shrinkage for Denoising of ECG Signals, 2006, In *EURASIP Journal on Advances in Signal Processing* 2006:1-10
130. Postalcioglu, S., Erkan, K., and Bolat, E.D. 2005. "Comparison of Kalman Filter and Wavelet Filter for Denoising", *International. Conference on Neural Networks and Brain 2005* 2:951 – 954
131. Prasad, R., Saruwatari, H., Lee, A., and Shikano, K. 2003. A Fixed-Point ICA for Convolved Speech Signal Separation. *4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*:579-584.
132. Prautzsch, H., Boehm, W., and Paluszny, M. 2002. B´ezier and B-Spline Techniques, Springer 1st edition, ISBN-10: 3642078427.
133. Qian, J. 2000. Denoising By Wavelet Transform, *Department of Electrical Engineering, Rice University*
134. Ramachandran, N., and Chellappa, A.K. 2006. Feature Extraction from EEG using Wavelets: Spike Detection Algorithm, *7th International Conference on Mathematics in Signal Processing*
135. Rohál'ová, M., Sykacek, P., Koska, M., and Dorffner, G. 2001. Detection of the EEG Artifacts by the means of the (Extended) Kalman Filter *Measurement Science Review* 1(1):59-62
136. Rossi, F., Fracois, D., Wertz, V., Meurens, M., and Verleysen, M. 2007. Fast Selection of Spectral Variables with B-Spline Compression". *Chemometrics and Intelligent Laboratory Systems* 86:208-218.
137. Rueckert D., Frangi, A.F., and Schnabel, J.A. 2003. Automatic Constructions of 3D Statistical Deformation Models of the Brain using Non-rigid Registration". *IEEE Transaction on Medical Imaging*, 22(8):1014-1025.
138. Sadovský, P. 2003. Removing Artifacts from EEG Signals using ICA. In *Radioelektronika 2003*:197-200.
139. Sadovský, P., Čermák, D., and Rozman, J. 2003. Removing Artifacts from EEG Signals. In *13th International Czech - Slovak Scientific Conference Radioelektronika 2003 Conference Proceedings*:197-200

140. Sameni, M. B. Shamsollahi, C. Jutten, G. D. Clifford A Nonlinear Bayesian Filtering Framework for ECG Denoising, *IEEE Transactions on Biomedical Engineering* 54(12):2172-2185
141. Sameni, R., Shamsollahi, M.B., and Jutten, C. 2005. Filtering Electrocardiogram Signals using the Extended Kalman Filter”, *27th IEEE Engineering in Medicine and Biology (EMBS) Annual Conference*: 5639-5642.
142. Sanei, S., and Chambers, J. 2007. “Introduction of EEG”, *EEG Signal Processing, John Wiley & Sons, ISBN-10: 0470025816*
143. Sasankar, A., Bawane, N.G., Bodkhe, S., Bawane, M.N. 2010. Artifact Extraction and Removal from EEG Data Using ICA In *IJCSNS International Journal of Computer Science and Network Security*, 10(3):96-103
144. Schl ogel, A., Anderer, P., Roberts, S.J., Preenzer, M., and Pfurtscheller, G. 1999. Artefact Detection in Sleep EEG by the use of Kalman Filtering. In *Proceedings EMBEC'99, Part II*:1648-1649.
145. Senthil Kumar, P., Arumuganathan, R., Sivakumar, K., and Vimal, 2008. A Wavelet Based Statistical Method for De-noising of Ocular Artifacts in EEG Signals, *International Journal of Computer Science and Network Security (IJCSNS)* 8(9):87-92.
146. Senthil Kumar, P., Arumuganathan, R., Sivakumar, K., and Vimal, 2008. Removal of Ocular Artifacts in the EEG through Wavelet Transform without using an EOG Reference Channel, *International Journal of Open Problems in Computer Science and Mathematics (IJOPCM)* 1(3):188-200
147. Senthil Kumar, P., Arumuganathan, R., Sivakumar, K., and Vimal, C. 2009. An Adaptive Method to Remove Ocular Artifacts from EEG signals using Wavelet, *Journal of Applied Sciences Research* 5(7):741-745.
148. Sivasankari. N and Dr. K. Thanushkodi, 2009. Automated Epileptic Seizure Detection in EEG Signals Using FastICA and Neural Network, *International Journal of Advances in Soft Computing and Its Applications*, 1(2):91-104.
149. Sripathi, D. 2003. Efficient Implementations of Discrete Wavelet Transforms Using FPGAs, Master Thesis, Department of Electrical and Computer Engineering - The Florida State University College Of Engineering

150. Stogbauer, H., A. Kraskov, S.A. Astakhov, P. Grassberger, 2004. Least Dependent Component Analysis based on Mutual Information. *Physical Review E* 70 (6) 066123.
151. Stögbauer, H., Andrzejak, R.G., Kraskov A., and Grassberger, P. 2004. Reliability of ICA Estimates with Mutual Information, ICA 2004, *Lecture Notes in Computer Science* 3195:209-219.
152. Stone, J.V. 2004. Independent Component Analysis: A Tutorial Introduction, MIT Press 2004
153. Storey, J. D., Xiao, W., Leek, J.T., Tompkins, R.G., and Davis, R.W. 2005. Significance Analysis of Time Course Microarray Experiments, In *Proceedings of the National Academy of Sciences of the United States of America*, 102(36):12837–12842.
154. St-Pierre, M., and Gingras, D. 2004. Comparison Between The Unscented Kalman Filter And The Extended Kalman Filter For The Position Estimation Module of An Integrated Navigation Information System *IEEE Intelligent Vehicles Symposium*:831- 835.
155. Su L., and Zhao, G. 2005. Denoising of ECG Signal Using Translation Invariant Wavelet Denoising Method with Improved Thresholding, In the 27th Annual Conference IEEE Engineering in Medicine and Biology:5946-5949
156. SuWen, L., WenQing, L., PinHua, X., and YuJui, Z. 2006. Application of Kalman Filtering and Wavelet Transform in DOAS”, 2006 IEEE Int. Conf. Information Acquisition:748-753.
157. Tarvainen, M.P., Hiltunen, J.K., Ranta-aho, P.O., and Karjalainen, P.A. 2004. Estimation of Nonstationary EEG with Kalman Smoother Approach: An Application to event-related synchronization (ERS), *IEEE Transactions on Biomedical Engineering* 51(3):516-524.
158. Tichavsky, P., Doron, E., Yeredor, A., and Nielsen. J. 2006. A Computationally Affordable Implementation of an Asymptotically Optimal BSS Algorithm for AR Sources, *Proceedings of the XIV European Signal Processing Conference (EUSIPCO 2006)*
159. Torkkola, K. 2003. Feature Extraction by Non-parametric Mutual Information Maximization, *Journal of Machine Learning Research* 3:1415-1438
160. Ulrych, T.J., Sacchi, M.D. 2005, Information-based Inversion And Processing With Applications Elsevier Publishers

161. Ungureanu, M., Bigan, C., Strungaru, R., and Lazarescu, V. 2004. Independent Component Analysis Applied in Biomedical Signal Processing", in proceedings of *Measurement Science Review* 4(2).
162. Unser, M., Aldroubi, A and Eden, M. 1993. B-Spline Signal Processing: Part 1 – Theory. *IEEE Transactions on Signal Processing* 41(2):821-832
163. Unser, M., and Aldroubi, A., 1996. A Review of Wavelets in Biomedical Applications”, *Proc. IEEE* 84(4):626-638.
164. Van Hulle. M.M., 2005. Edgeworth Approximation of Multivariate Differential Entropy. *Neural Computation* 17(9):1903-1910.
165. Van Hulle. M.M., 2008. Sequential Fixed-Point ICA Based on Mutual Information Minimization. *Neural Computation* 20(5):1344-1365.
166. Vara prasad Naraharisetti, K. 2010. Removal of Ocular Artifacts from EEG Signal using Joint Approximate Diagonalization of Eigen Matrices (JADE) and Wavelet Transform, *Canadian Journal on Biomedical Engineering & Technology* Vol. 1, No. 4, July 2010, 56-60
167. Vigon L.R., Saatchi, M.R., Mayhew, J.E.W., and Fernandes, R. 2000. Quantitative Evaluation of techniques for Ocular Artefacts Removal, *IEEE Proc Science Measurement and Technology* 147 (5):219-228.
168. Walters-Williams, J., and Y. Li. 2009. Estimation of Mutual Information: A Survey. *4th International Conference on Rough Set and Knowledge Technology (RSKT2009)*:389-396.
169. Walters-Williams, J., and Li, Y. 2010. B-Spline Mutual Information Independent Comonent Analysis, In *Journal of Computer Science and Network Security*, 10(7):129-141.
170. Wallstrom, G., Kass, R., Miller, A., Cohn, J., and Fox, N., 2002, “Correction of Ocular Artifacts in the EEG using Bayesian Adaptive Regression Splines,” *Case Studies in Bayesian Statistics, 6. Springer-Verlag, New York, NY.*

171. Wan E.A., van der Merwe, R. and Nelson, A.T. 2000. Dual Estimation and the Unscented Transformation in *Neural Information Processing Systems 12*:666–672
172. Wan E.A., and van der Merwe, R. 2000. The Unscented Kalman Filter for Nonlinear Estimation, *IEEE Symposium on Adaptive Systems for Signal Processing, Communication and Control (ASSPCC)*:153-158
173. Wan E.A., and van der Merwe, R. 2001. The Square-Root Unscented Kalman Filter for State and Parameter Estimation, In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Singal Processing (ICASSP '01)* 6:3461-3465.
174. Wang, B., and Lu, W. 2005. An in-depth Comparison of FastICA, CuBICA and IC-FastICA, *Advances in Natural Computations, Lecture Notes on Computer Science* 3611:410-414.
175. Welling, M., Zemel, R.S., and Hinton, G.E., 2003. Probabilistic Sequential Independent Component Analysis, *Technical Report, Department of Computer Science, University of Toronto*.
176. West, M., Prado, R., and Krystal, A.D. 1999. Evaluation and Comparison of ECG traces: Latent Structures in Nonstationary Time Series, *Journal of the American Statistical Association* 94(448).
177. Woestenburg, J.C., Vertaten, M.N., and Slangen J.L. 1983. The Removak of the Eye Movement Artifact from the EEG by regression Analysis in the Frequency Domain, *Biological Physiology* 16:127-147.
178. Wu, W., Gao, Y., Bienenstock, E., Donoghue, J., and Black, M and Serruya, M. 2002. Inferring Hand Motion from Multi-Cell Recordings in Motor Cortex Using a Kalman Filter. *SAB - workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices*: 66–73.
179. Wu, W., Gao, Y., Bienenstock, E., Donoghue, J., Black, M Serruya, M., and Shaikhouni, A. 2003. Neural Decoding of Cursor Motion Using a Kalman Filter. In *Advances in Neural Information Processing Systems 15*: 117–124.
180. Wu, W., Gao, Y., Bienenstock, E., Donoghue, J., and Black, M. 2006. Bayesian Population Decoding of Motor Cortical Activity Using a Kalman Filter. In *Neural Computation*, 18(1): 80–118.

181. Xu, D., Principe, J., Fisher, J., and Wu, H.C. 1998. A Novel Measure for Independent Component Analysis (ICA), In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 2:1161-1164.
182. Ylipaavalniemi, J., and Vigario, R. 2004. Analysis of Auditory fMRI Recordings via ICA: A Study on Consistency. In *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IJCNN'2004)*.
183. Yup Han, J., Lee, S.K., and Park, H.B. 2009. Denoising ECG using Translation Invariant Multiwavelet, *International Journal of Electrical Computer and Systems Engineering* 3(3):138-142
184. Zarzoso, V., Comon, P., and Kallel, M. 2006. How fast is FastICA? *14th European Signal Processing Conference (EUSIPCO-2006)*.
185. Ziehe, A., and Müller, K. 1998. TDSEP - An Efficient Algorithm For Blind Separation Using Time Structure, *Proceedings of the 8th International Conference on Artificial Neural Networks (ICANN '98)*:675-680
186. Zhang, W., and Song, G., 2003. A Translation-Invariant Wavelet Denoising Method based on a new Thresholding Function, *2nd Int. Conf. Machine Learning and Cybernetics*:2341-2345.
187. Zhou, W. and Gotman, J. 2004. Removal of EMG and ECG Artifacts from EEG Based on Wavelet Transform and ICA. *26th Annual International Conference on the IEEE EMBS*:392-395.
188. Zhou, W. and Gotman, J. 2005. Removing Eye-movement Artifacts from the EEG during the Intracarotid Amobarbital Procedure *Epilepsia* 46(3):409-411.
189. Zouridakis, G., and Iyer, D. 2004. Comparison between ICA and Wavelet-based Denoising of single-trial evoked potentials. *26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 87-90.

APPENDICES

APPENDIX A Calculating B-Spline (adapted)

```
function [xx,yy] = bSpline1(x, y, n)

    % Finds n points along xx,yy given the b-spline
    % defined by the control points in (x,y).
    %
    % [x(1),y(1)] & [x(end),y(end)] are the two on-curve
    % end points- the rest are the off-curve control point(s).
    %
    % The final curve is rendered as a set of Bezier curves
    % defined by 3 point (two on-curve end points and one
    % off-curve control point).
    % If the 3 points of each Bézier curve are
    % (Ax, Ay), (Bx, By) and (Cx, Cy), then
    %   xx = (1-t)^2.Ax + 2t(1-t).Bx + t^2.Cx
    %   yy = (1-t)^2.Ay + 2t(1-t).By + t^2.Cy
    % Varying t from 0 to 1 produces all the points on the curve.
    %
    % If there is more than 1 off-curve control point specified,
    % then intermediate on-curve control points are assumed to be
    % at the midpoint connecting consecutive off-curve points.
    %
    % 99.05.21 RFD bobd@stanford.edu

    if length(x) ~= length(y)
        error('x and y must be equal length vectors!');
    end
    if length(x) < 3
        error('need at least 3 points in x and y!');
    end

    if length(x) == 3
        % special case- a one Bezier segment contour
        t = linspace(0,1,n);
        xx = (1-t).^2.*x(1) + 2.*t.*(1-t).*x(2) + t.^2.*x(3);
        yy = (1-t).^2.*y(1) + 2.*t.*(1-t).*y(2) + t.^2.*y(3);
    else
        nSegments = length(x) - 2;
        segn = ceil(n/nSegments);
        t = linspace(0,1,segn);
        xx = [];
        yy = [];
        for ii=1:length(x)%-1
            if ii==1
                segx1 = x(ii);
                segy1 = y(ii);
                segx3 = (x(ii)+x(ii+1))/2;
                segy3 = (y(ii)+y(ii+1))/2;

            elseif ii==length(x)%-1
                segx1 = (x(ii)+x(ii-1))/2;
                segy1 = (y(ii)+y(ii-1))/2;
                segx3 = x(ii);
            end
        end
    end
end
```

```

        segy3 = y(ii);
    else
        segx1 = (x(ii)+x(ii-1))/2;
        segy1 = (y(ii)+y(ii-1))/2;
        segx3 = (x(ii)+x(ii+1))/2;
        segy3 = (y(ii)+y(ii+1))/2;
    end
    xx = [xx (1-t).^2.*segx1 + 2.*t.*(1-t).*x(ii) +
t.^2.*segx3];
    yy = [yy (1-t).^2.*segy1 + 2.*t.*(1-t).*y(ii) +
t.^2.*segy3];
    end
    end
    return;

```

APPENDIX B Calculating the probability of a single vector – P(X)

```
function [P]=single_prob(x)
% Function that calculate the probability of X
%
% -INPUT-
% x: VECTOR 1
%
% -OUTPUT-
% P: probability of x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N = length(x);
P = [1,N];

for i=1:N;
    P(i) = 1/N * sum(sum(x(i)));
end
end
```

APPENDIX C Calculating the joint probability of X and Y - P(X,Y)

```
function [JB] =joint_prob(x,y)

% Function that calculate the joint probability of X and Y
%
% -INPUT-
% x: VECTOR 1
% y: VECTOR 2
%
% -OUTPUT-
% JP: joint probability
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N = length(x);
JB = [1,N];

for i=1:N;
    % j(i) = sum(sum(x(i) * y(i)));
    JB(i) = 1/N * sum(sum(x(i) * y(i)));
end
end
```

APPENDIX D Calculating The Marginal Entropy- H(X)

```
function Hx=mar_entropy(X)
%   Function that calculate marginal entropy of X
%
%   -INPUT-
%   X:   VECTOR
%
%   -OUTPUT-
%   Hx:  marginal entropy
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
row = length(X);
Hx = zeros(1,row);
for i=1:row;
    if X(i)==0
        %do nothing
    else
        [F,E] = (log2(X(i)));
        Hx(i) = Hx(i) + -(X(i)* F);
    end
end
end
```

APPENDIX E Calculating the MI of X and Y

```
function I = mutual(Hx,Hy,Pxy)
%   Function that calculates the mutual information X and Y
%
%   -INPUT-
%   Hx:   VECTOR 1
%   Hy:   VECTOR 2
%   Pxy:  Joint Probability of x and y
%
%   -OUTPUT-
%   I:  mutual information
%
% Takes a pair of signals and returns the mutual information Ixy
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculate joint entropy for vectors 1 and 2
[F,E] = (log2(Pxy + (Pxy==0)));
Hxy = -sum(sum(Pxy.* F));
%calculate mutual information
I = Hx + Hy - Hxy;
end
```

APPENDIX F Determining the B-Spline based MI estimator

```
function I = bs_mi(x,y)
% Function that estimates the Mutual Information (MI) of X and Y
% with estimated joint probability distribution P(x,y). The
% estimation is performed using a quadratic B-spline function
%
% -INPUT-
% x: VECTOR that represents signal 1
% y: VECTOR that represents signal 2
%
% -OUTPUT-
% I: estimated Mutual Information
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculate b-spline of input vectors
[Bx,By] = bSpline1(x, y, 1);

% calculate probabilities for each variable
Px = single_prob(Bx);
Py = single_prob(By);

% calculate joint probabilities
Pxy = joint_prob(Bx,By);

% calculate marginal entropies
Hx = mar_entropy(Px);
Hy = mar_entropy(Py);

% calculate mutual information using entropy
I = mutual(Hx,Hy,Pxy);
end
```

APPENDIX G Determining B-Spline ICA Components

```
function B = bsica(whitesig,numOfSig,numSamples)
% Function that determines the Bspline ICA components
%
% -INPUT-
% whitesig:    signal used to determine mutual information
% numOfSig:   Number of signals
%
% -OUTPUT-
% B: vector used to calculate W
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Declare variables to be used
maxNumIterations = 1000;
epsilon = 0.0002;           %Stop criterion

% Take uniform orthonormal initial signals
B = orth(randn(numOfSig) - .5); % create a nxn matrix
BOld = B;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Loop to determine ICA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for round = 1:maxNumIterations + 1,
    if round == maxNumIterations + 1,
        fprintf('No convergence after %d steps\n', maxNumIterations);
        break;
    end

    wsignal = whitesig' * B;           % Calculate ICA components

    fprintf ('Calculating B Spline Mutual Information.\n');
% Calculate Mutual Information
    for i = 1:numOfSig,
        if i == numOfSig
            MI(i,:) = bs_mi(wsignal(i,:),wsignal(i-numOfSig+1,:));
        else
            MI(i,:) = bs_mi(wsignal(i,:),wsignal(i+1,:));
        end
    end

% Calculate nonlinearities for each signals
    for i = 1:size(wsignal,2),
        gradient(i,:) = tanh1(wsignal(:,i));
    end

    B1 = whitesig * gradient'/numSamples; % create a nxn matrix
    B2 = sum((1-gradient.^2)'*MI);
```

```

    %B3 = ones(numOfSig,1) * B2;
    B = B1-ones(numOfSig,1) * B2.*B/numSamples;

% Symmetric orthogonalization
    B = symm(B);

% Test termination conditions
    minAbsCos = min(abs(diag(B'*BOld)));
    fprintf('Step no. %d, change in value of estimate: %.6f \n',...
        round,1-minAbsCos);

    if (1 - minAbsCos > epsilon)
        fprintf('Convergence after %d steps\n', round);
    end
    BOld = B;
end
end

```

APPENDIX H BMICA algorithm

```
function [whitesig,wMatrix,dMatrix] = preprocess(signal)
%   Function that whitened the EEG signals and reduces dimension
%
%   -INPUT-
%   signal:      EEG mixed signal
%
%   -OUTPUT-
%   wsignal: whitened signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Remove mean from signal
nomean = signal - mean(signal')' * ones(1,size(signal,2));

% Determine the whitened EEG signal
wMatrix = inv(real(sqrtm(cov(nomean'))));
whitesig = wMatrix * nomean;

% Just some security...
if ~isreal(whitesig)
    error ('Whitened vectors have imaginary values. ');
end

covarianceMatrix = cov(whitesig', 1);
[E, D] = eig (covarianceMatrix);
dMatrix = E * sqrt (D);

% Print some information to user
fprintf ('\nCheck: covariance differs from identity by [ %g ].\n',
...
max (max (abs (cov (whitesig', 1) - eye (size (whitesig, 1))))));
end

function [desig,W,A] = bmica(signal)
%   Function that determines the Bspline ICA components
%
%   -INPUT-
%   signal:      EEG mixed signal
%
%   -OUTPUT-
%   denoised: Denoised EEG signals
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Check some basic requirements of the data
if nargin == 0,
    error ('You must supply the mixed data as input argument. ');
end

if ~isreal(signal)
```

```

    error('Input has an imaginary part.');
```

end

```

if length (size (signal)) > 2,
    error ('Input data can not have more than two dimensions.');
```

end

```

if any (any (isnan (signal))),
    error ('Input data contains NaN's.');
```

end

```

if ~isa (signal, 'double')
    fprintf ('Warning: converting input data into regular (double)
precision.\n');
```

signal = double (signal);

end

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[numOfSig,numSamples] = size(signal);

% print information about data
fprintf('Number of signals: %d\n', numOfSig);
fprintf('Number of samples: %d\n', numSamples);

% Check if the data has been entered the wrong way, warn only
if numOfSig > numSamples
    if b_verbose
        fprintf('Warning: ');
        fprintf('The signal matrix may be oriented in the wrong
way.\n');
```

fprintf('In that case transpose the matrix.\n');

end

end

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Whitening data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf('PreprocessingSignal . . .');
```

[whitesig,whiteMat,deMat] = preprocess(signal);

fprintf('done.\n');

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calcultating ICA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf('Determining ICs...\n');
```

B = bsica(whitesig,numOfSig,numSamples);

W = B' * whiteMat;

A = deMat * B;

desig = W * signal;

end

APPENDIX I Determine UKF

```
function MM_UKF = Ukf(data)

x_0 = .1; % Initial state
P_0 = 1;

[row,len_vec] = size(data); % # of vectors and
inputs
data1 = data(row:len_vec);

%----- Initialization of Data -----

M = x_0;
P = P_0;
Y = zeros(1,len_vec); % Space for
measurements

% Strengths of perturbations
u_n = 1;
v_n = 1;

% Handles to dynamic and measurement model functions, and their
derivatives
f_func = @ungm_f;
h_func = @ungm_h;

%----- Assign data to true states with process noise -----

data1 = zeros(1,len_vec); % Matrix for storing
state estimates

%----- Unscented Kalman Filter -----

MM_UKF = zeros(size(M,1),len_vec);
PP_UKF = zeros(size(M,1),size(M,1),len_vec);

% Filtering loop for UKF
for k = 1:size(Y,2)
    [M,P,X_s,w] = ukf_predict(M,P,f_func,u_n,v_n,k);
    [M,P] = ukf_update(M,P,Y(:,k),h_func,v_n,X_s,w,[]);

    MM_UKF(:,k) = M;
    PP_UKF(:,:,k) = P;
end

end
```

APPENDIX J FWT Shrink

```
function [nsig] = FWT_Shrink(signal,L)
% FWTShrink -- Soft Threshold Shrinkage Applied to Wavelet
% Coefficients
% Usage
%   [nsig] = FWTShrink(signal,L)
% Inputs
%   signal    1-d signal. length(y)= 2^J
%             Normalized to noise level 1! (See NoiseNorm)
%   L         Low-Frequency cutoff for shrinkage (e.g. L=4)
%             Should have L << J!
% Outputs
%   nsig      estimate, obtained by applying soft thresholding on
%             wavelet coefficients
%
% Description
%   FWTShrink smooths noisy data presumed to have noise level 1
%   by transforming it into the wavelet domain, applying soft
%   thresholding to the wavelet coefficients and inverse
%   transforming.

qmf = MakeONFilter('Symmlet',8);           % Quadrature Mirror Filter
[signal,y] = NormNoise(signal,qmf);
[n,J] = dyadlength(signal);
wcoef = FWT_PO(signal,L,qmf);             % Apply Wavelet Transform
fprintf('\n UKF running...                ');
wcoef = Ukf(wcoef);                       % Apply UKF filter
fprintf('done\n');
wcoef = NeighSSdenoise(wcoef, L);         % Apply shrinkage
wcoef = pearson_ica(wcoef);               % Apply ICA algorithm
nsig = IWT_PO(wcoef,L,qmf);
end
```

APPENDIX K CTICA Algorithm

```
function fout = runwave_2d(data)

    [out] = zeros(size(data));
    L = 5;

    nspin = 16;
    for i=0:(nspin-1),
        for j=1:(nspin-1),
            [cycle] = cyclespin2(data, i,j);
            [denoise] = FWT2d_Shrink(cycle,L);
            [cspin] = cyclespin2(denoise, -i,-j);
            [out] = out + cspin;
        end
    end
    [fout] = out/nspin;
end

function [MSE,PSNR,Xclean] = ctica(Xnoise)
% determine denoised signal for a signal

    Xclean = ones(size(Xnoise));
    Xclean = runwave_2d(Xnoise);

% Estimate the denoising effect (i.e. computing MSE and PSNR)
[MSE, PSNR] = Calc_MSE_PSNR(Xnoise,Xclean);
end
```