DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

INDEPENDENT COMPONENT ANALYSIS IN BIOMEDICAL APPLICATIONS AND ACCELEROMETER DATA LOGGING SYSTEM

by
Taner AKKAN

September, 2009
İZMİR

INDEPENDENT COMPONENT ANALYSIS IN BIOMEDICAL APPLICATIONS AND ACCELEROMETER DATA LOGGING SYSTEM

**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in
Electrical and Electronics Engineering**

by

Taner AKKAN

September, 2009
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"INDEPENDENT COMPONENT ANALYSIS IN BIOMEDICAL APPLICATIONS AND ACCELEROMETER DATA LOGGING SYSTEM"** completed by **TANER AKKAN** under supervision of **ASST. PROF. DR. YAVUZ ŞENOL** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

....

Asst. Prof. Dr. Yavuz ŞENOL

Supervisor

....	....
Prof. Dr. Cüneyt GÜZELİŞ	Asst. Prof. Dr. Gökhan DALKILIÇ
Thesis Committee Member	Thesis Committee Member

....	....
Examining Committee Member	Examining Committee Member

Prof.Dr. Cahit HELVACI

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

Firstly, I would like express my sincere gratitude to my supervisor Asst. Prof. Dr. Yavuz Şenol for providing me with an interesting thesis topic to work with, and accepting me for supervision. Thanks for guiding me throughout this thesis work, enlightening me with advises and knowledge, and encouraging me.

I would like to thank Assoc. Prof. Dr. Murat Özgören for his invaluable ideas and contribution to the thesis work. Also, thanks to Asst. Prof. Dr. Adile Öniz and Dokuz Eylul University Biophysics Department staff for their contributions and EEG data.

I would like to thank Prof. Dr. Hasan Havıtçıoğlu for his suggestions, discussions and patience data labeling. Also, thanks to Dokuz Eylul University Orthopaedics and Traumatology Department and Biomechanics Department staff for their contributions and data recording preparation.

I thank thesis committee members Prof. Dr. Cüneyt Güzeliş and Asst. Prof. Dr. Gökhan Dalkılıç for their suggestions and contributions during thesis work. I thank Asst. Prof. Dr. Ahmet Özkurt and Asst. Prof. Dr. Nalan Özkurt for their valuable comments. Also, I am grateful to Mustafa Şakar for his support.

I would like to thank my university for giving us the opportunity to study at a stimulating atmosphere.

Last, but not least, I thank my parents for encouraging me for higher studies and assisting me by all means during my study period.

Taner AKKAN

INDEPENDENT COMPONENT ANALYSIS IN BIOMEDICAL APPLICATIONS AND ACCELEROMETER DATA LOGGING SYSTEM

ABSTRACT

In biomedical applications, the obtained biomedical data must be cleared from all undesired artifacts such as environmental and sensor noises prior to signal processing. If the frequency properties of artifacts are known, various appropriate filtering techniques can be used to remove them. However, if the information about artifact sources is limited or not exists, it is required to utilize various statistical methods to remove the artifacts. Independent Component Analysis (ICA) as the most known Blind Source Separation method can be used for filtering or detecting sources.

In this thesis study, ICA, ICA methods and the realization problems have been introduced and ICA has been applied to various biomedical applications. For ICA applications, two different novel data logging systems have been developed: simultaneous multi channel sound data logging system and simultaneous multi point multi dimensional accelerometer data logging system. The sound data logging system was used for general ICA applications and the ICA realization problems were examined. The accelerometer data logging system was used to collect knee vibroarthrographic (VAG) signals and for diagnosing knee problems. The obtained results were presented.

Moreover, ICA was applied to identification of artifact sources and artifact removing problems. Heart pulses and sweating artifacts in EEG recordings were identified and removed by ICA. Furthermore, the accelerometer data logging system has been used for filtering jaw motion artifacts from the EEG recordings.

Keywords: Blind Source Separation, Independent Component Analysis, EEG artifact removal, Signal filtering, Vibroarthrographic signals, Microcontroller based systems, Data Logging Systems, Accelerometer.

BİYOMEDİKAL UYGULAMALARDA BAĞIMSIZ BİLEŞEN ANALİZİ VE İVMEÖLÇER VERİ TOPLAMA SİSTEMİ

ÖZ

Biyomedikal uygulamalarda elde edilen biyomedikal veriler sinyal işleme öncesinde çevresel ve algılayıcı gürültüsü gibi istenmeyen bozucu etkilerden temizlenmelidir. Eğer bozucu etkilerin frekans özellikleri bilinirse, çeşitli uygun filtreleme teknikleri onları çıkarmak için kullanılabilir. Bununla birlikte, bozucu etkilerin kaynağı hakkında bilgimiz çok az veya hiç yoksa bozucu etkileri kaldırmak için istatistiksel metotlardan faydalanmamız gerekecektir. Gözü kapalı Kaynak Ayrıştırma yöntemleri arasında en bilineni olan Bağımsız Bileşen Analizi (BBA) bu amaçla filtreleme ve kaynak bulma için kullanılabilir.

Bu tez çalışmasında, BBA, BBA yöntemleri ve gerçekleştirme problemleri tanıtılmış ve BBA, çeşitli biyomedikal uygulama alanlarına uygulanmıştır. BBA uygulamaları için, eşzamanlı çoklu kanallı ses veri toplama sistemi ve eşzamanlı çok noktadan çok eksenli ivmeölçer veri toplama sistemi olmak üzere iki farklı özgün veri toplama sistemi geliştirilmiştir. Ses veri toplama sistemi genel BBA uygulamalarında kullanılmış ve BBA gerçekleştirme problemleri incelenmiştir. İvmeölçer veri toplama sistemi ise eklem titreşim grafisi (ETG) sinyallerini toplamak ve diz problemlerini tanılamakta kullanılmıştır. Elde edilen sonuçlar verilmiştir.

BBA ayrıca EEG bozunum kaynaklarının tanımlanması ve kaldırılması problemlerinde de uygulanmıştır. EEG kayıtlarındaki kalp atış bozunumu ve terleme bozunumu BBA kullanılarak tanımlanmış ve temizlenmiştir. Buna ek olarak, ivmeölçer veri toplama sistemi EEG kayıtlarından çene hareketi bozunumunun temizlenmesinde de kullanılmıştır.

Anahtar sözcükler: Kör Kaynak Ayırımı, Bağımsız Bileşen Analizi, EEG bozunum çıkarma, Sinyal filtreleme, Eklem Titreşim Grafi Sinyalleri, Mikrodenetleyici Tabanlı Sistemler, Veri Toplama Sistemleri, İvmeölçer.

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

Blind Source Separation (BSS) is a technique that finds the components of a mixture of various signals without knowing any or less information about them. BSS methods aim to find the components which are hidden in a real world mixture of signals using underlying factors (Figure 1.1). The most popular blind source separation method is Independent Component Analysis (ICA) and it is very popular among researchers because of wide usage on the signal processing application areas. In this study, ICA was applied on biomedical artifacts filtering such as heart pulses and sweating artifacts in EEG and knee vibroarthrographic (VAG) signal analysis.

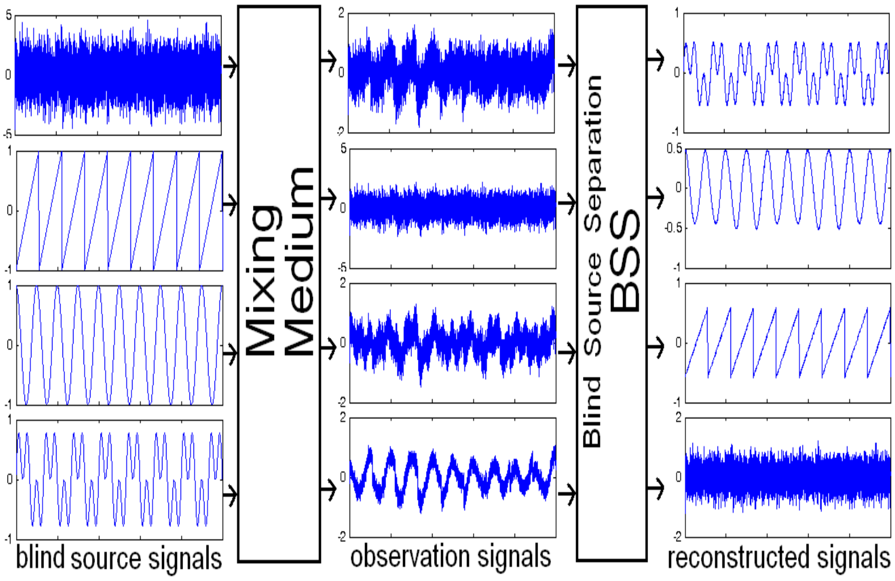


Figure 1.1 The Blind Source Separation procedure (left column) the source signals are mixed into a medium (middle column) depicts the observed signals and (left column) the reconstruction is achieved through the BSS algorithm.

Decomposing a signal into its independent components first used by Jutten and Herault (1991) in a blind source separation study, then ICA term was expressed by Comon (1994) in his paper which shows the theory of linear ICA. This research field became popular with the paper on Infomax principle written by Bell and Sejnowski (1995). Later Amari (1998) simplified Infomax learning rule with introducing natural gradient concept which was discovered by Cardoso. The original infomax ICA algorithm with sigmoidal nonlinearities was only suitable for super-Gaussian sources. An efficient extended infomax ICA algorithm for non-Gaussian signals was developed by Lee, Girolami, & Sejnowski (1999a).

Most popular FastICA algorithm that uses negentropy and kurtosis contrast functions was developed by Hyvärinen (1999a). After that many kind of ICA algorithms proposed such as geometric ICA, batch cumulant-based ICA algorithms JADE and SHIBBS, Efficient ICA and many others. Different ICA approaches use principles like maximum likelihood, Bussgang methods based on cumulants, projection pursuit and negentropy methods. All of these methods are in the infomax framework. A survey on ICA is a very useful study by Hyvärinen (1999b).

ICA can be used to find all the important components in a linearly mixed signal sources. These components can be used to analyze the nature of the mixed signal sources, or filtered out from the other undesired components. ICA performance heavily depends on the statistical independency criterion between source signals.

ICA has many applications such as speech filtering, speech recognition, image noise filtering, signal enhancements and preprocessing stage for neural networks. In addition, ICA is also very popular in biomedical applications such as EEG signal filtering and source localization problems. EEG – fMRI artifact filtering, EEG – fMRI source localization, speech and face recognition, sound localization, kinematic signals covering knee VAG signals filtering applications shows that ICA is getting more popular and still needs to be improved by researchers.

Removal of noise and detection of some components in a signal are all related to usage of ICA. Noise filtering is used in biomedical applications especially EEG analysis. Analysis of knee related problems are similar to the EEG signal analysis except less sensors are required. EEG recording systems are available and they are complex devices, because multi channel EEG instrumentation amplifiers and data capturing devices are needed. For VAG signal analysis, two or three channel recordings are sufficient and there is no specific device available for recording multi channel signals. In this study, two kinds of data logging systems were built for this purpose. Multi channel sound and multi channel acceleration data logging systems were designed and built on USB interface. The developed systems would be also suitable for various applications in addition to the thesis main subjects of sound signal processing and knee signal diagnosis.

A low-cost multi-channel simultaneous sound capturing system was developed by Akkan & Senol (2008a) for ICA (Figure 1.2). This system makes easy and cost-effective data logging and thus real time ICA applications can be developed on MATLAB environment. This system first especially thought to be used to log vibration data from the human body skin. Especially knee related problems can be detected with the microphone and stethoscope combinations inspired from some medical doctors that still use auscultation method for pre-diagnosis. But, knee researches on that topic show that accelerometers are more suitable than the microphones.

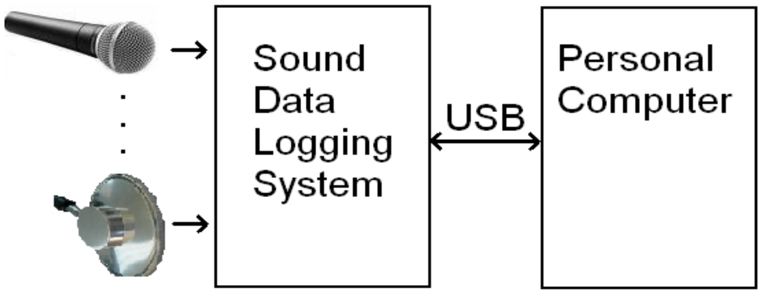


Figure 1.2 The schematic representation of the Sound Data Logging system developed in our thesis work.

Rapid development on the accelerometer technology makes possible to build multi channel simultaneous acceleration data logging. Accelerometer data logging system was built using PIC microcontroller and USB interface (Akkan & Şenol, 2008b). Accelerometer data logging system is shown in Figure 1.3.

USB sound and acceleration data logging systems were used to implement ICA algorithms in real life applications. Some sound separation applications are made using sound data logging system and problems of real life ICA applications and their solutions were investigated on this platform.

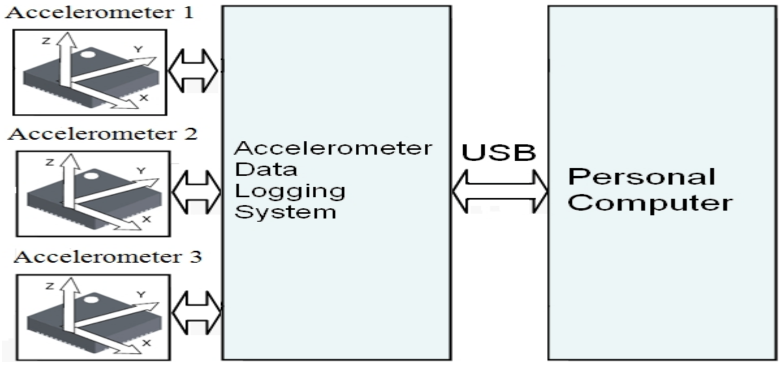


Figure 1.3 The schematic representation of the Accelerometer Data logging system developed in our thesis work. The accelerometers are shown with their three degrees of freedom for each unit forming 9 channel axes information.

1.1 Objectives of the thesis

The objective of the study is introducing BSS and the most popular ICA method, discussing the advantages and the disadvantages of ICA, exhibiting real world ICA implementation problems and some solution to these problems, showing ICA benefits on biomedical applications.

Biomedical applications such as EEG and VAG analysis need recording of multi channel long time duration data. Some of these data are related to our aim of the study; on the other hand, some of these data are redundant and noisy. Analysis like ICA can be the answer to find these important data and filter out from the undesired data. Here, the aim is to find suitable ICA algorithms and pre-processing and post processing techniques to improve the signal processing performance.

In EEG application of ICA, our aim was to find deterministic components such as heart pulse signals and sweating artifact, and then extract from spontaneous EEG recordings by using various ICA methods. The reason to use spontaneous EEG recordings is to provide continuity of data for implementing ICA method easily and efficiently. Then, evoked potential EEG recordings would also be analyzed by ICA. In this study, electrocardiogram signal and sweating artifact were both identified using kurtosis criteria from ten channel recordings by using Fast ICA, Efficient Fast ICA and Maximum Likelihood ICA techniques (Akkan & Şenol, 2009).

EEG data are easily available from many related sources. But, multi channel VAG signals must be collected using a specific data logging system. For this purpose, a multi-point multi-dimensional accelerometer data logging system was designed and built. Simultaneous data recording from multi points is needed to achieve comparison of multi-points data with each other in the same time instances. For the vibration data analysis, ICA will be used to remove noise and unwanted signal forms from knee recordings to extract the desired signal source(s). Then, the obtained desired signals can be used for diagnosing purposes. Our accelerometer data logging hardware and software are suitable for ICA analysis which, preferably, needs to have simultaneously recorded multi-point data.

The system uses popular Universal Serial Bus (USB 2.0) to provide fast data collection, and the microcontroller triggers the specially selected digital accelerometer via serial peripheral interface (SPI) to meet the requirements for the synchronous data logging. VAG data were collected from twenty patients including healthy ones and with the ones having knee problems. It has been observed that the recorded VAG signals show different statistical behaviour for normal and abnormal knees. There are numerous knee related problems and this requires creating a large data base from different cases to obtain data features for developing a convenient classification algorithm.

Another application area was filtering of jaw artefacts from EEG recordings using simultaneously recorded vibration data from human jaw using acceleration data logging system. These supporting data would help to filter jaw motion based artefacts effectively from EEG recordings without losing the important information on the EEG signals. This study is currently on the beginning phase.

1.2 Thesis Outline

Thesis study consists of six chapters and two appendices. This introduction chapter explains the thesis concept and the objectives. In the second chapter, ICA methods are explained and some ICA methods are given. Third chapter discusses real world ICA problems and investigates some solutions. In the fourth chapter, sound and vibration data logging hardware are introduced. Fifth chapter gives biomedical applications on ICA. The result of the thesis are concluded and discussed in Chapter six. In Appendix A, some useful MATLAB codes developed for the thesis study are given. In Appendix B, healthy and problematic knee signals are given graphically.

CHAPTER TWO
INDEPENDENT COMPONENT ANALYSIS

Blind Source Separation (BSS) methods are very popular in especially extracting a signal from signal mixture and in finding the underlying features or signals in a signal mixture. Our inspiration is the success of the human brain to separate sounds and images using two sensors (ears and eyes). This means brain is a very sophisticated statistical engine. So, statistical methods are most popular methods in that topic. Also these methods can be a preprocessing stage mostly to reduce input dimension of the neural network circuits. The Cocktail Party Problem is a main example of the separation of mixed signals. In this problem, many source signals are mixed in a media with delays and reverberation effects. Media can be air, water, a solid material, human brain, etc.

Some BSS methods are listed below:

a) Bayesian Approach: Forming a model that describes a particular source separation problem. The result is a mixing matrix. The algorithm is known from Independent Component Analysis (ICA).

b) TDSEP (Temporal Decorrelation source separation): Temporal structure of signals is used in order to compute the time-delayed 2nd order correlation for the source separation. The best results are achieved if the autocorrelations are as different as possible. Algorithm makes a rotation in order to simultaneously diagonalize the set of time-lagged correlation matrices. This algorithm sometimes delivers better results than ICA, especially in Gaussian signals. Compared to ICA, it is computation load is low.

c) Blind Separation of disjoint orthogonal signals: It uses only 2 mixtures of N sources, but the sources have to be disjointly orthogonal. The algorithms are based on the Short Time Fourier Transform.

d) Principal component analysis (PCA): PCA transforms a number of correlated variables into a smaller number of uncorrelated variables called principal components (PC). The first principal component variability in the data is very much, and the other components variability is less. Main use of PCA is to reduce the dimensionality of the data set.

PCA use second-order methods in order to reconstruct the signal in the mean-square error sense. The results are independent in the second order statistics. PCA basis vectors are mutually orthogonal.

The uncorrelated principal components are estimated from the eigenvectors of the covariance or correlation matrix of the original variables. PCA is mostly in use of eigenvalue decomposition of covariance of a signal, or singular value decomposition of a signal.

PCA was first introduced by Pearson (1901) and used mostly in data analysis. Some application areas of PCA are noise reduction, data compression, visualization of high dimensional data, dimension reduction, and filtering of undesired components from a signal mixture etc.

e) Independent Component Analysis: Most popular blind source separation technique is independent component analysis. ICA especially finds wide application areas in biomedical such as source separation, artifact finding and filtering, source localization.

In this chapter, firstly the background theory for independent component analysis was given. Especially gaussianity criteria, gaussian signal types are given in that manner. Then, the independent component analysis was introduced and explained.
Later and the some ICA algorithms were given and the useful test results are explained.

2.1 Independent Component Analysis Background Theory

Before understanding independent component analysis, the theory behind the analysis must be known well. Theory of ICA comes mostly from statistics and information theory.

2.1.1 Probability Distribution Function

Probability distribution term identifies either the probability of each value of an unidentified random variable (when the variable is discrete), or the probability of the value falling within a particular interval (when the variable is continuous) (Everitt, 2006). Thus, a random variable can represent some possible values and the probability of this random variable can be defined in this possible values range.

The cumulative distribution function (cdf) completely describes the probability distribution of a random variable *X* that has real values. The cumulative distribution function F_x is defined as in Equation 2.1;

$$F_x(x_0) = P(x \leq x_0) \tag{2.1}$$

where x_0 take values -∞ to ∞, so cdf is calculated for all values of x. Cdf is a nonnegative function for continuous random variables and values of cdf are in the range of $0 \leq F_x(x) \leq 1$. For example, if x has an uniform distribution on the interval of [0,1], the cdf is given by in Equation 2.2;

$$F(x) = \begin{cases} 0 : x < 0 \\ x : 0 \leq x \leq 1 \\ 1 : 1 < x \end{cases} \tag{2.2}$$

The probability density function (pdf) is the derivative of its cdf.

$$p_x(x_0) = \left.\frac{dFx(x)}{dx}\right|_{x=x_0} \tag{2.3}$$

Assuming x is an n-dimensional random vector;

$$x = (x_1, x_2, ..., x_n)^T \tag{2.4}$$

where the components $x_1, x_2, ..., x_n$ are continuous random variables.

Joint density $p_{x,y}(x, y)$ of x and y is defined as

$$p_{x|y}(x|y) = \frac{p_{x,y}(x, y)}{p_y(y)} \tag{2.5}$$

where $p_y(y)$ is the marginal density.

2.1.2 Expectation and the moments

In data analysis and processing, expectation of a function of a random variable is very important. The expectation of $g(x)$ is denoted by $E\{g(x)\}$ is defined as

$$E\{g(x)\} = \int_{-\infty}^{+\infty} g(x) p_x(x) dx \tag{2.6}$$

where $g(x)$ is a random variable either scalar or a vector or a matrix.

Usually the probability density of a random vector is unknown, but often a set of n samples $x_1, x_2, ..., x_n$ from x is available, as for example in the case of data measured in real world applications. The expectation can then be estimated by averaging over the samples using

$$E\{g(x)\} \cong \frac{1}{n} \sum_{i=1}^{n} g(x_i) \tag{2.7}$$

If g(x) is the form of x^n, the nth moment is defined as

$$\alpha_n = E\{x^n\} = \int_{-\infty}^{+\infty} x^n p_x(x) dx \tag{2.8}$$

But the central moments form μ_n is more useful and they are computed around the mean:

$$(m_x = \alpha_1 = 1^{\text{st}} \text{ moment}), \quad \mu_n = E\{(x - m_x)^n\} = \int_{-\infty}^{+\infty} (x - m_x)^n p_x(x)dx \quad (2.9)$$

From Equation 2.9, we found $\mu_0 = 1$ due to normalization and $\mu_1 = 0$ due to the removal of mean. The higher moments are valuable and called as higher order statistics. Second, third and fourth moments are defined as;

$$\sigma^2 = \mu_2 \text{ Second moment (the variance of x)} \quad (2.10)$$

$$\gamma = \frac{\mu_3}{\sigma^3} \text{ Third moment (the skewness of x)} \quad (2.11)$$

$$\kappa = \frac{\mu_4}{\sigma^4} - 3 \text{ Fourth moment (the kurtosis of x)} \quad (2.12)$$

The variance gives an estimate of distribution width, the skewness asymmetry of the distribution and the kurtosis an estimate of the deviation from a Gaussian distribution or in other words gaussianity. For example, the Gaussian distribution kurtosis is zero, and the skewness is also zero. In one dimension pdf of Gaussian function as;

$$p_x(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \quad (2.13)$$

Gaussian pdf can be written in another form as ;

$$p_x(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{\left[\frac{-(x-\overline{X})^2}{2\sigma^2}\right]} \quad (2.14)$$

A Gaussian distribution function can be shown in Figure 2.1. The graphic was created in MATLAB. Variance is equal to 1. $\overline{X}$ denotes the mean value and equals to 0.

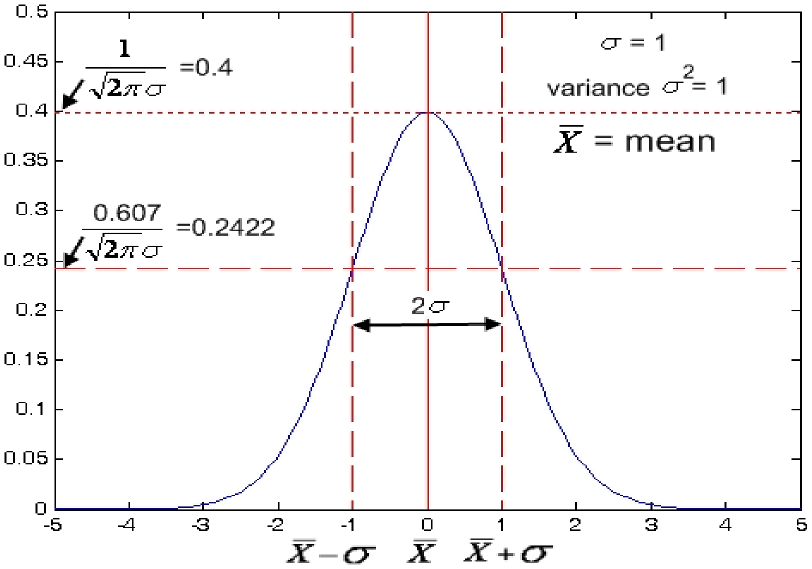


Figure 2.1 Gaussian distribution function. Variance is 1 and mean is 0.

The Gaussian distribution types according to the variances (σ^2) are shown in Figure 2.2. The graphic was created in MATLAB. Variance shows gaussianity here, if it is small that means supergaussianity occurs.

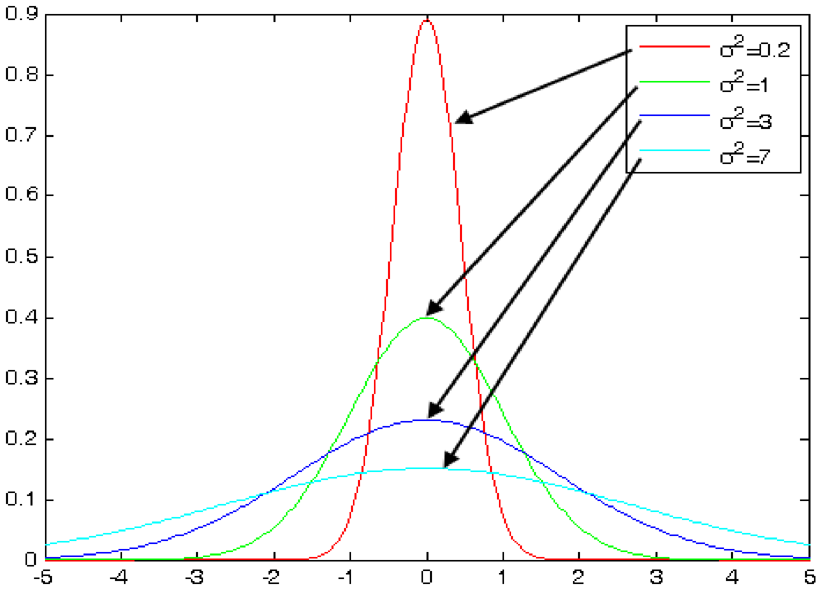


Figure 2.2 Gaussian distribution types according to variances. As variance increases, sub Gaussian distribution occurs. As variance decreases, super Gaussian distribution occurs.

A Sub-Gaussian pdf is typically flatter than the Gaussian pdf. Example: signals mainly “on”, e.g. 50/60 Hz electrical mains supply, but also eye blinks. A Super-

Gaussian pdf has typically a longer tail and a sharper peak than Gaussian pdf. Example: infrequent signals of short duration e.g. evoked brain signals. Also music has Super-Gaussian pdf. Some examples are in Figure 2.3.

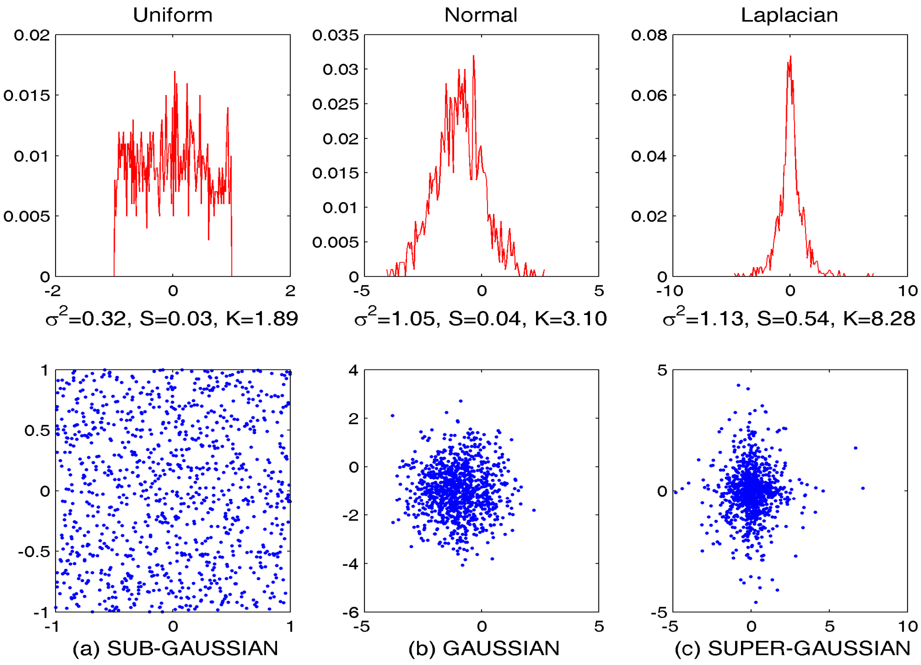


Figure 2.3 Histogram and observation basis examples of Gaussian distribution types. Kurtosis (K) values differentiate the Gaussian distribution types. a) Uniform distribution b) Normal distribution c) Laplacian distribution.

2.1.3 Correlation and independence

The correlation between ith and jth component of a random vector of x is;

$$r_{ij} = E\left\{x_i x_j\right\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_i x_j p_{xi,xj}(xi,xj)dx_j dx_i \tag{2.15}$$

If this correlation value is zero, the two variables are uncorrelated random variables. The correlation matrix for random vector x can be calculated as in Equation 2.16.

$$R_x = E\left\{xx^T\right\} \tag{2.16}$$

The covariance matrix is a kind of correlation matrix and calculated using the mean removal from vector x.

$$C_x = E\left\{(x - m_x)(x - m_x)^T\right\} \tag{2.17}$$

For two random vectors x and y, cross-correlation and cross-covariance matrices;

$$R_{xy} = E\left\{xy^T\right\},\ C_{xy} = E\left\{(x - m_x)(y - m_y)^T\right\} \tag{2.18}$$

x and y are uncorrelated if $C_{xy} = 0$. But independence requires also the joint probability distribution must be;

$$p_{x,y}(x, y) = p_x(x).p_y(y) \tag{2.19}$$

The independence is much stronger property than uncorrelatedness. If random variables have Gaussian distributions and uncorrelated at the same time, this means these variables are also independent. If a mixture includes more than one Gaussian component, ICA fails.

2.2 Independent Component Analysis

Independent Component Analysis (ICA) is the identification & separation of mixtures of sources with little or no prior information. ICA is sometimes known as blind signal separation. ICA finds underlying factors or components from multidimensional statistical data and searches for both statistically independent and non-gaussian components.

ICA algorithm minimizes the mutual information between the statistically independent components. In contrast to correlation-based transformations such as Principal Component Analysis (PCA), ICA not only decorrelates the signals (2nd-order statistics) but also reduces higher-order statistical dependencies and makes the signals more independent. ICA does not constrain the axes to be orthogonal as in PCA and attempts to place them in the directions of statistical dependencies in the data. In Figure 2.4, chirp and train sounds (8192 Hz sample frequency) were mixed linearly. The scatter plot of mixtures, PCA and ICA result shows that ICA achieved to recover sources successfully, but PCA couldn't find the exact rotation angle to recover the sources back. In Mixtures scatter plot, perpendicular PCA vectors doesn't match with the correct mixtures vectors which can be easily used by ICA.

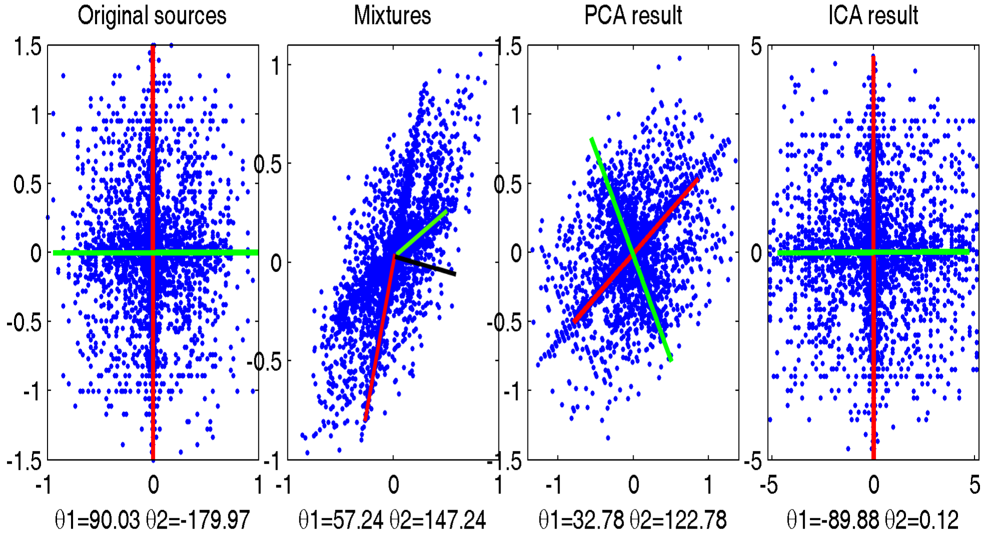


Figure 2.4 PCA and ICA comparison. Original sources on the left mixed to form the mixtures. PCA and ICA results are on the right.

There are three different signal types in ICA: signals have Non-Gaussian distribution, signals that are non-stationary and have slowly changing power spectrum, and signals have time correlation and have different power spectrums.

When a signal mixture is represented as a linear combination of the original sources at very time instant, it is defined as an instantaneous mixing model. This model is the simplest form for separation. Practically, when the signals are recorded in an ideal environment, i.e., no reverberation, the mixture of the signals recorded by the mixing system can be considered as an instantaneous mixture.

ICA instantaneous mixing model can be expressed as

$$x = As + n \tag{2.20}$$

where $x = (x_1, x_2, ..., x_n)^T$ and $s = (s_1, s_2, ..., s_m)^T$ are the vectors of observed random variables and the independent components respectively. Here, n is the additive noise and A is an unknown mixing matrix.

Following conditions are listed for ICA types due to number of mixed signals (n) and number of independent components (m) :

- Complete ICA (quadratic ICA) , (m=n) : This is the general case.
- Less-Complete ICA (undercomplete ICA), (m>n): Here, our aim is to find more number of independent components using less mixtures.
- Over-Complete ICA, (n<m) : Here, it is desired to found few independent components with more than enough sensors.

ICA estimates A so that we can recover the sources via $A^{-1} = W$. Recovered sources are not the exact copy of original sources (Amari, Cichocki, & Yang, 1996; Hyvärinen, Patrik, & Mika, 2001b). A general case of ICA model is shown in Figure 2.5.

Although instantaneous mixing models are very handy, it fails to model real life situations, for instance recording in a real room. The microphones in this environment pick up not only the signals of the sources, but also the delayed and attenuated versions of the same signals due to reverberation. Hence, this can be viewed as microphones receiving a filtered version of different signals, and can be modeled as follows:

$$x(n) = \sum_{p=0}^{p} A(p).s(n-p) + n(n) \tag{2.21}$$

where $A(p)$ is the multipath multi-channel filter.

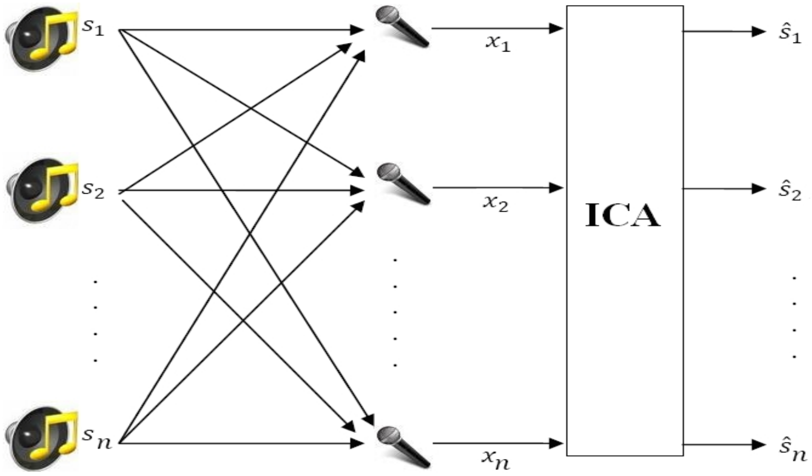


Figure 2.5 Instantaneous ICA Schematic Diagram. Sources are mixed and observed with the sensors enters the ICA algorithm to get the recovered sources.

The source mixtures are called convolved mixtures since acoustic signals recorded simultaneously in a reverberant environment can be described as the sums of differently convolved sources. Figure 2.6 shows convolutive mixing system model.

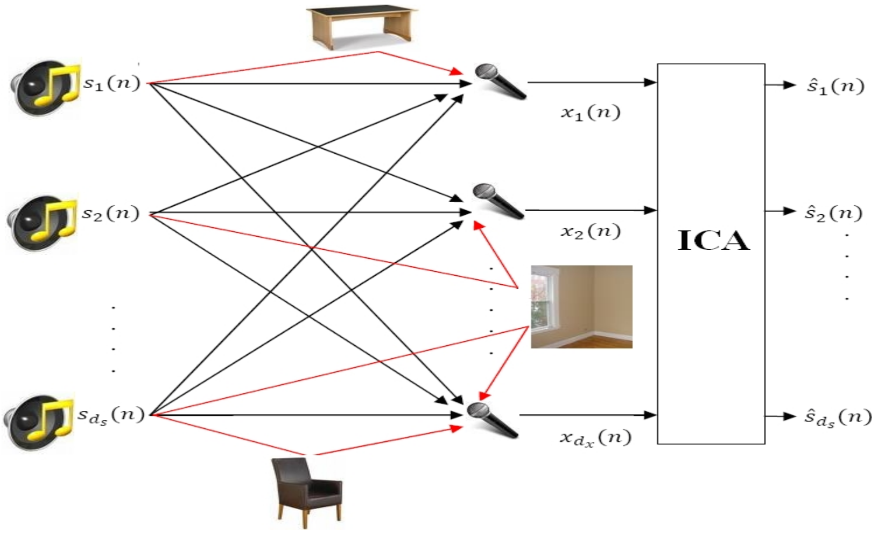


Figure 2.6 Convolutive ICA Schematic Diagram. Sources are mixed with delays and reflections, and the observed signals enter to the ICA algorithm. Recovered sources are obtained at the outputs of ICA.

As we know before, ICA tries to find the independent components with only observing the mixture, and no need to have prior knowledge about the original sources. But, in the convolutive mixing situation, knowing prior information about sources helps the algorithm be successful.

This problem is known as “cocktail party” problem. It is not a big problem for human beings to concentrate on listening to one voice in the room, even when there are lots of other sound sources, such as other conversations, music, background noise, etc. present in the room. The "cocktail party" problem can be described as the ability to focus one's listening attention on a single talker among a mixture of conversation and background noise; also known as "cocktail party effect". For a long time, it has been recognized as an interesting and a challenging problem.

Experiments by Payne (1969) have shown that owls are sensitive to the sounds and they must be able to accurately localize both the azimuth and the elevation of the sound source. On the other hand, the human sound separation system starts with the filter banks mechanism of the cochlea and ends with a neural network circuit, but it is not fully understood that human separation system is still superior to the artificial systems, but the human hearing system has the ratio of higher frequency to lower frequency is at least 1000:1 and the strongest signal to lowest signal ratio is 32 trillion to 1 (Ludwig, 2009).

The "cocktail party problem" is also called the "multichannel blind deconvolution” problem. It is aimed at separating a set of mixtures of convolved signals, detected by an array of microphones. This is performed extremely well by the human brain, and over the years attempts have been made to capture this functionality by using assemblies of abstracted neurons or adaptive processing units. Some of the other interesting application areas of BSS include speech enhancement in multiple microphones, cross talk removal in multi-channel communication, Direction of Arrival (DOA) estimation in sensor arrays and improvement over microphone directivity for audio and passive sonar.

Two important application of ICA are blind source separation and feature extraction. Recently, blind source separation applications has received attention because of its potential applications in signal processing such as in speech recognition systems, telecommunications and medical signal processing.

General application areas are:

- Blind source separation (Bell&Sejnowski, TeWon Lee, Girolami, Hyvarinen,etc.)
- Image denoising (Hyvarinen)
- Medical signal processing – fMRI, ECG, EEG (Mackeig)
- Modelling of the hippocampus and visual cortex (Lorincz, Hyvarinen)
- Feature extraction, face recognition (Bartlett)
- Compression, redundancy reduction
- Watermarking (D Lowe)
- Clustering (Girolami, Kolenda)
- Time series analysis (Back, Valpola)
- Topic extraction (Kolenda, Bingham, Kaban)
- Scientific Data Mining (Kaban, etc.)
- Vibration Data Analysis.

2.3 Algorithms of Independent Component Analysis

First step in ICA is whitening. Whitening is done usually applying Principal Component Analysis. Then using the contrast function ICA seeks proper rotation. All of the ICA algorithms differ with their rotation algorithms. After rotation has done, the sources are recovered. There is one case when rotation doesn't matter. This case cannot be solved by basic ICA when sources have gaussian distributions. In Figure 2.7, ICA steps are shown graphically.

For problems that mixture signals more than sources (m>n), the problems can be easily reduced to quadratic case by applying PCA. Because more number of linear mixture combinations do not give any further information for separating the sources. But if less mixture equations are available (n>m), this means we don't have enough knowledge to find the sources on a overcomplete basis. Overcomplete ICA methods (first presented by Lewicki and Sejnowski, 1998) are needed to solve these ICA problems. Later, blind source separation of more sources than mixtures using overcomplete representations has been published (Lee, Lewicki, Girolami, & Sejnowski, 1999).

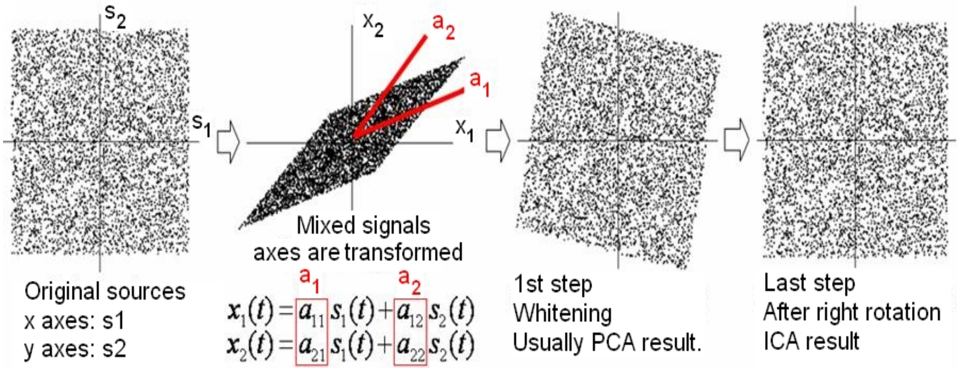


Figure 2.7 ICA steps.

Latest ICA algorithms can be found in Cardoso (2009). Another survey is helpful by Hyvärinen & Oja (2000). The study by Azzerboni, Ipsale, Foresta, Mammone, & Morabito (2006) is a good information source of ICA algorithms for biomedical applications.

ICA methods have two categories for the properties: the statistical properties (e.g., consistency, asymptotic variance, robustness) of the ICA method depend on the choice of the objective function, and the algorithmic properties (e.g., convergence speed, memory requirements, and numerical stability) depend on the optimization algorithm.

There are several approaches in determining independent components. The approaches and some related algorithms will be briefly given in the following subsections.

2.3.1 Maximization of non-gaussianity

According to the central limit theorem of statistical theory, the distribution of a sum of independent random variables tends toward a gaussian distribution, under certain conditions. Loosely speaking, a sum of two independent random variables usually has a distribution that is closer to gaussian than any of the two original random variables (Hyvärinen, Karhunen, & Oja, 2001a). Therefore, some algorithms to maximize the non-gaussianity have been developed. These algorithms depend on two main gaussianity measures. One of them is kurtosis which is defined in Eq. 2.12. However, the kurtosis is very sensitive to outliers since it is related to fourth order moment, so it is not robust in noisy situations. Thus, the negentropy, an objective function which is based on information theory, can be used for maximization of non-gaussianity. The negentropy basically show the how far the entropy of the distribution from the entropy of a gaussian variable.

Most known algorithm is FastICA algorithm because of computation cost is very low and very faster (10 to 100 times) than the conventional gradient descent based methods. The most important reason to be fast is fixed point iteration mechanism. FastICA also can be used for applying projection pursuit which is a general purpose data analysis method based on finding low-dimensional projections of multivariate data that show highly nongaussian distributions. Projection pursuit (Friedman & Tukey, 1974; Friedman, 1987; Huber, 1985; Jones & Sibson, 1987) finds meaningful projections in a multidimensional data and this gives us visualization of data, density estimation and regression analysis. Here, the most gaussian sources are not interesting because these sources are barrier to find the other components. ICA, here interests with least gaussian sources to deal and more popular than projection pursuit.

Fast ICA algorithm is in Figure 2.8. The algorithm is deflationary, so all the independent components are estimated one by one, not all of them in one calculation.

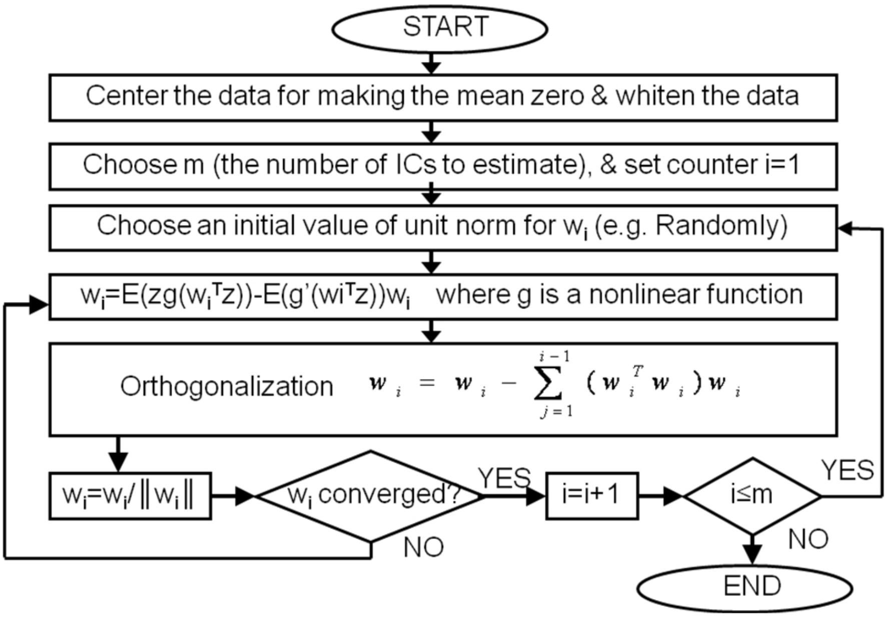

START
Center the data for making the mean zero & whiten the data
Choose m (the number of ICs to estimate), & set counter i=1
Choose an initial value of unit norm for w_i (e.g. Randomly)
$w_i=E(zg(w_i^Tz))-E(g'(wi^Tz))w_i$ where g is a nonlinear function
Orthogonalization $w_i = w_i - \sum_{j=1}^{i-1} (w_i^T w_i) w_i$
$w_i=w_i/\|w_i\|$
w_i converged?
YES
NO
i=i+1
i≤m
YES
NO
END

Figure 2.8 Fast ICA algorithm.

Efficient ICA (EFICA) is an improved version of FastICA algorithm, and it is asymptotically efficient (Koldovsky, Tichavsky, & Oja, 2006). Accuracy of the algorithm is high because the residual error variance goes to Cramér-Rao lower bound. EFICA is more complex than FastICA algorithm and it is very efficient to separate linearly mixed speech signals.

2.3.2 Maximum likelihood estimation and ML ICA

Another way of determining the most independent directions is to use the maximum likelihood estimation. It is used to estimate the distribution parameters. The idea is to use estimates, which give the highest probability ("likelihood") for the observations. The expectation maximization (EM) algorithm is one of the most common methods for determination. It has been showed that the fixed point algorithm for maximum likelihood approach which is implemented in FastICA package gives an almost identical optimization problem with maximization of nangaussianity approach. Also, there is a close relationship with the infomax principle of neural networks. This is based on maximizing the output entropy, or information flow, of a neural network with nonlinear outputs. It is seen that the output entropy is of the same form as the expectation of the likelihood. This means that infomax is equivalent to maximum likelihood estimation.

Here there is assumption about parametric density $p(x|\theta)$ for an observation vector x. Thus, the likelihood function for the samples of $x(1), x(2),..., x(T)$ is;

$$p(x(1), x(2),..., x(T)|\theta) = \prod_{j=1}^{T} p(x(j)|\theta) \tag{2.22}$$

The ML estimate for θ maximizes function in Eq. 2.22 and it is consistent especially T goes to infinity and asymptotically efficient because asymptotically estimation error is minimized down to the Cramer-Rao lower bound (Hyvärinen, Karhunen, & Oja, 2001a). The practical maximization is done with the Natural Gradient Algorithm. A gradient ascent algorithm (Figure 2.9) is easy to derive for ML ICA.

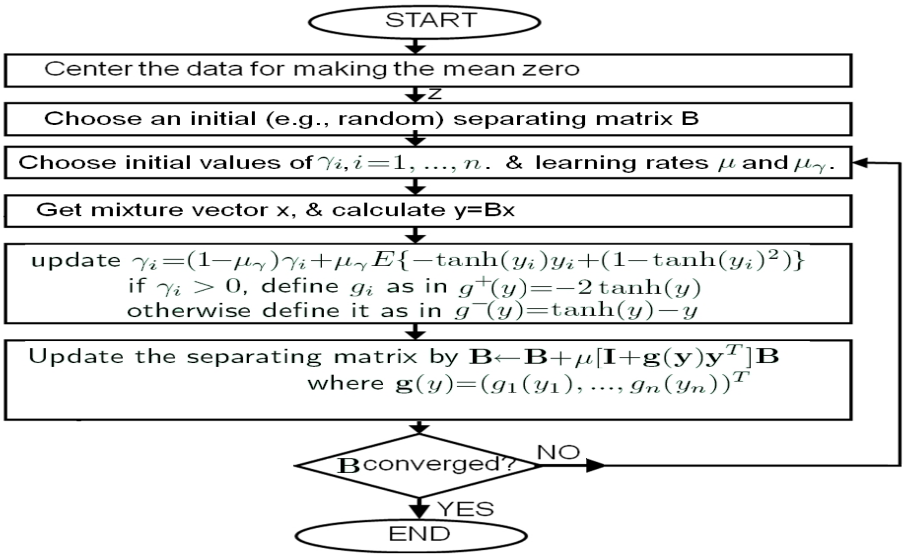

START
Center the data for making the mean zero
z
Choose an initial (e.g., random) separating matrix B
Choose initial values of $\gamma_i, i=1, \ldots, n.$ & learning rates μ and μ_γ.
Get mixture vector x, & calculate y=Bx
update $\gamma_i=(1-\mu_\gamma)\gamma_i+\mu_\gamma E\{-\tanh(y_i)y_i+(1-\tanh(y_i)^2)\}$
if $\gamma_i > 0$, define g_i as in $g^+(y)=-2\tanh(y)$
otherwise define it as in $g^-(y)=\tanh(y)-y$
Update the separating matrix by $\mathbf{B}\leftarrow\mathbf{B}+\mu[\mathbf{I}+\mathbf{g}(\mathbf{y})\mathbf{y}^T]\mathbf{B}$
where $\mathbf{g}(y)=(g_1(y_1), \ldots, g_n(y_n))^T$
$\mathbf{B}$converged?
NO
YES
END

Figure 2.9 ML ICA algorithm.

2.3.3 Minimizing mutual information

The mutual information is a natural measure of the dependence between random variables. It is always nonnegative, and zero if and only if the variables are statistically independent. Mutual information takes into account the whole dependence structure of the variables, and not just the covariance, like principal component analysis (PCA) and related methods. Therefore, it can be used as a criterion for finding the ICA representation. It is showed that ICA estimation by minimization of mutual information is equivalent to maximizing the sum of nongaussianities of the estimates of the independent components, when the estimates are constrained to be uncorrelated. However, deflationary approach is not possible in this method.

2.3.4 Tensorial methods

One approach for estimation of independent component analysis consists of using higher-order cumulant tensor. Cumulant tensors are then generalizations of the covariance matrix. The covariance matrix is the second-order cumulant tensor, and the fourth order tensor is defined by the fourth-order cumulants. The whitening of the data means that we transform the data so that second-order correlations are zero. As a generalization of this principle, the fourth-order cumulant tensor can be used to make the fourth-order cumulants zero, or at least as small as possible. This kind of (approximative) higher-order decorrelation gives one class of methods for ICA estimation.

Joint approximate diagonalization of eigenmatrices (JADE) firstly described by Cardoso & Souloumiac (1993) and uses equal eigenvalues of cumulant tensors. A method closely related to JADE is given by the eigenvalue decomposition of the weighted correlation matrix. Also SHIBBS algorithm uses the same cumulant based batch algorithm approach as in JADE. Both algorithms are described by Cardoso (1999). JADE requires no parameter tuning, but SHIBBS needs some tuning in applications. SHIBBS needs less memory when it is compared to JADE algorithm. Both algorithms use algebraic ideas to optimize a 4th-order measure of independence.

2.3.5 Methods using time structure

In the approaches explained so far, the time structure has not been considered. However, most of the mixed signals are time signals. If the ICs are time signals, they may contain much more structure than simple random variables. For example, the autocovariances of the ICs are good candidates for statistic properties. Additional statistic information can be used to improve the estimation of the model. This additional information can make the model has more performance in cases where the basic ICA methods cannot be used for estimation. If the ICs are gaussian and correlated in time, this is helpful.

For example, Algorithm for Multiple Unknown Source Extraction (AMUSE) algorithm which is similar to PCA algorithm can be used. AMUSE applies two cascade PCA algorithms: first PCA is used for whitening and the second PCA is applied to time delayed covariance matrix of the pre-whitened data. Here, PCA is a good choice for separating the sources because there is no independency requirement like in ICA, but the sources must have temporal structure. AMUSE was first proposed by Tong, Soon, Huang, & Liu (1991).

2.4 Tests on Independent Component Analysis

Now, ICA test has been done for following more complex signals as another synthetic test using Fast ICA, ML ICA, Jade ICA, Shibbs ICA results and their spectrograms. Four test signals are generated to see normal ICA algorithms performance and other four signals set are generated to see how ICA algorithms fail. Following sub sections show signals, spectrograms of the signals, mixtures, and the results of FastICA algorithms and its spectrogram and the results of other algorithms.

2.4.1 ICA algorithms on a normal case

First test uses four different signals such as a signal with two different frequency sinusoids added, a low frequency sinusoid, a square wave signal, and gaussian random signal. The test signals and the MATLAB generation codes are given below:

```
N=11025;
t=linspace(0,10.2*pi,N); k1=sin(t)'+sin(3*t+0*pi/18)';     % sin(t)+sin(3t)
t=linspace(0,45.1*pi,N); k2=sin(0.07*t+0*pi/18)';           %sin(0.07t)
t=linspace(0,65.3*pi,N); k3=0.9*square(1.8*t)';             %square_wave(1.8t)
t=linspace(0,77.5*pi,N); k4=randgauss(1,5,N)';              % gaussian random
```

Figure 2.10 shows the test signals and spectrograms of the test signals are shown in Figure 2.11.

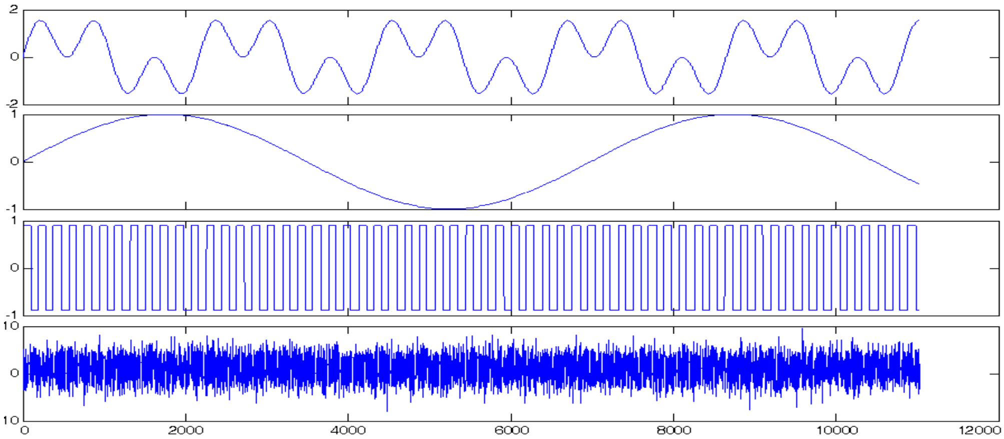


Figure 2.10 Four test signals. y axes show the signal amplitudes and x axes show the samples.

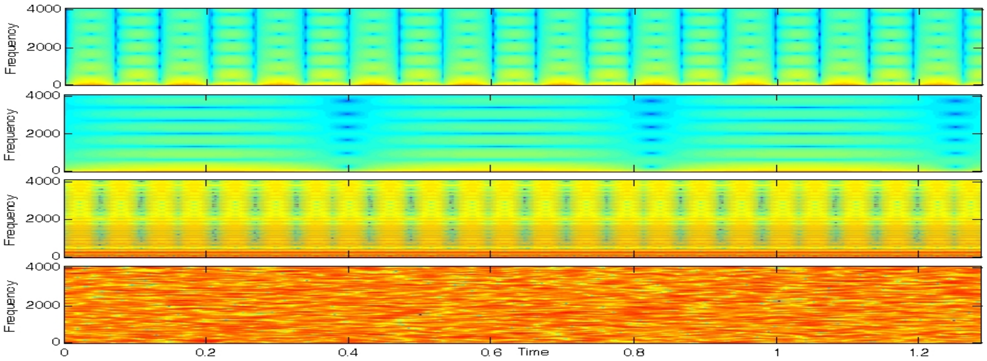


Figure 2.11 Spectogram of four test signals.

The test signals were mixture with each other. The mixture signals are shown in Figure 2.12.

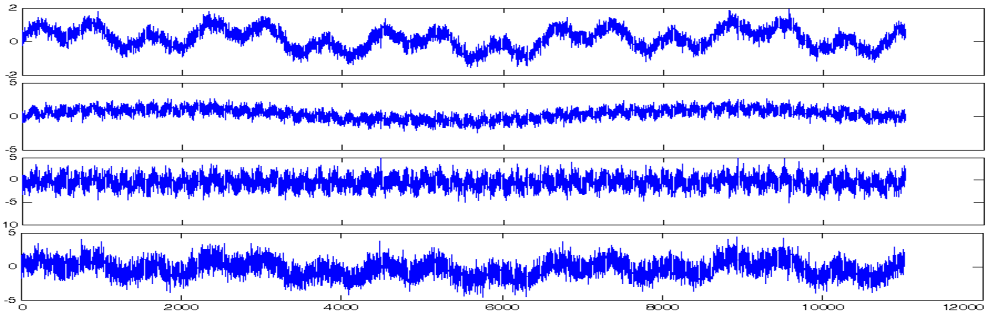


Figure 2.12 Four test signals mixtures. y axes show the signal amplitudes and x axes show the samples

Fast ICA results were given in Figure 2.13. Note that Fast ICA recovers the signals not in a correct order which has given first. Fast ICA is successful, but low frequency sinusoidal signal is not recovered well. Spectrograms of the ICA result were given in Figure 2.14. Also spectrograms show that recovered signals are not exactly the same with the original signals.

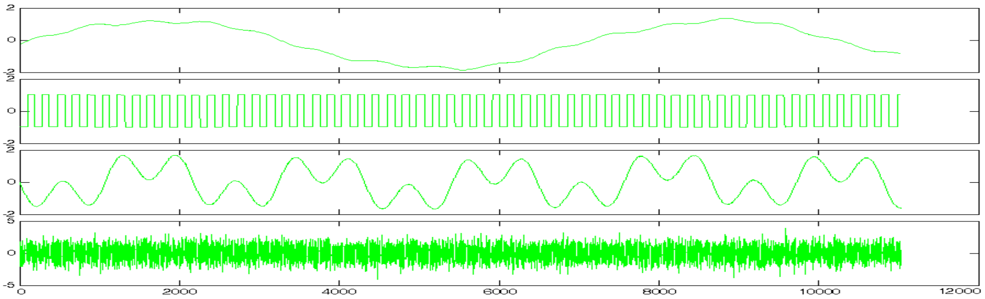


Figure 2.13 Fast ICA results. y axes show the signal amplitudes and x axes the samples.

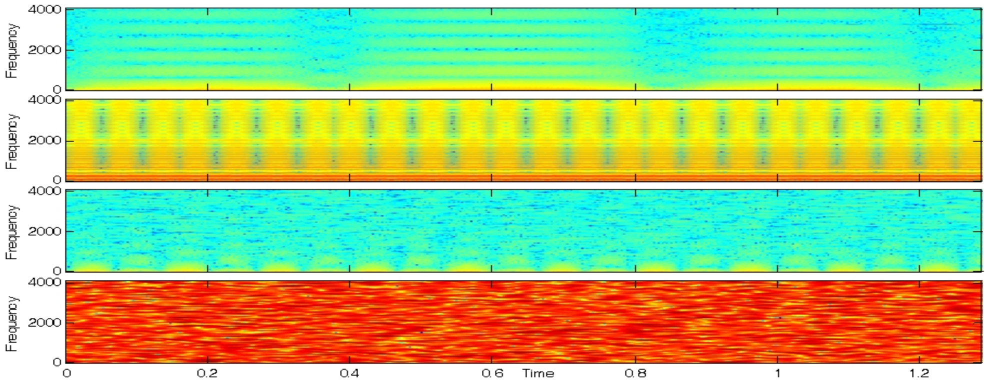


Figure 2.14 Spectogram of Fast ICA results.

ML ICA results were given in Figure 2.15. ML ICA fails. Jade ICA results were given in Figure 2.16. Jade ICA results are better than Fast ICA.

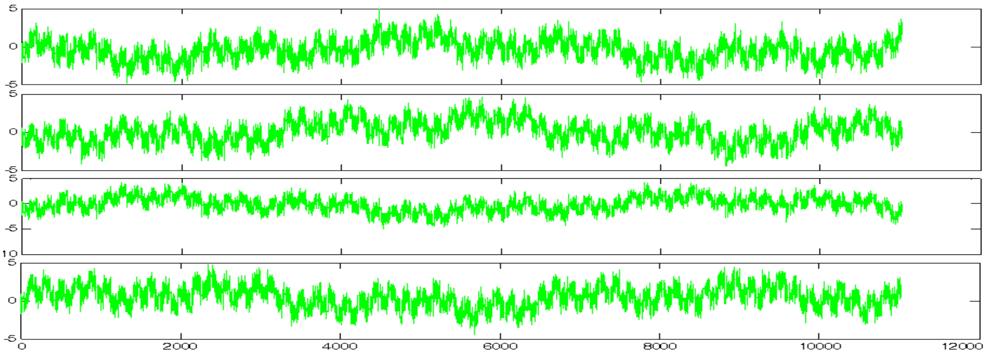


Figure 2.15 ML ICA results. y axes show the signal amplitudes and x axes the samples.

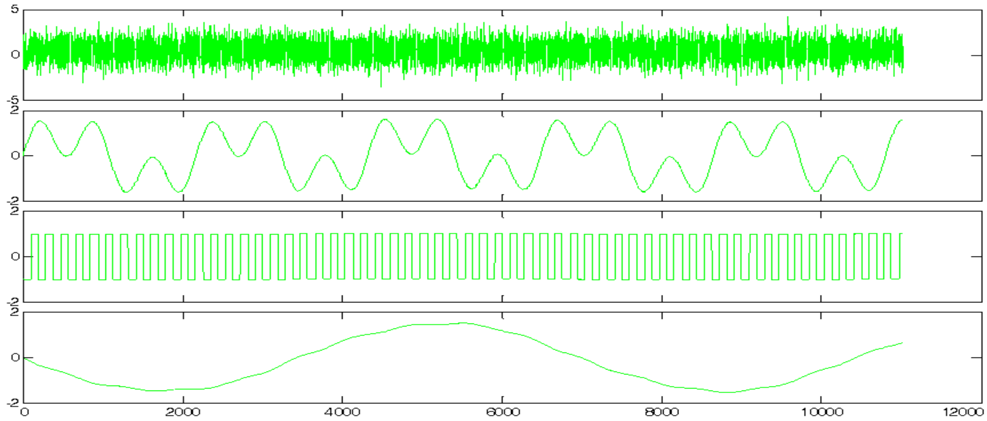


Figure 2.16 Jade ICA results. y axes show the signal amplitudes and x axes the samples.

Shibbs ICA results were given in Figure 2.17. Note that Shibbs ICA results are similar to the Jader ICA results.

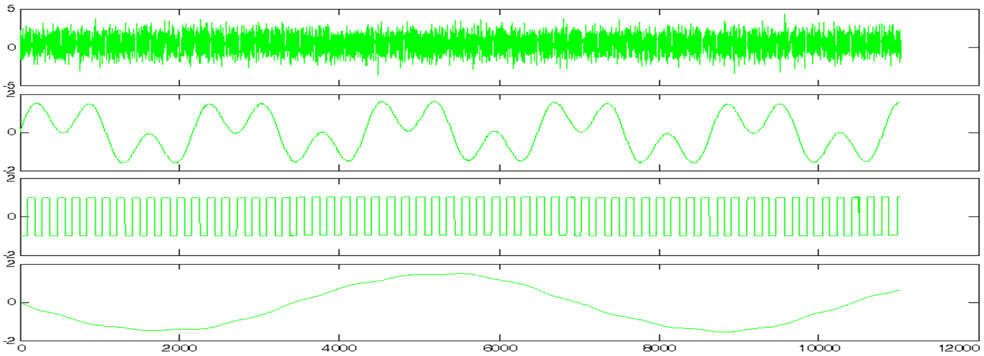


Figure 2.17 Shibbs ICA results. y axes show the signal amplitudes and x axes the samples.

2.4.2 ICA algorithms on an abnormal case

A complex problem is here. Following sinusoidal and square wave signals were generated. The first sinusoid frequency was doubled in the second signal. Also the same thing happened in the second square wave signal (Figure 2.18).

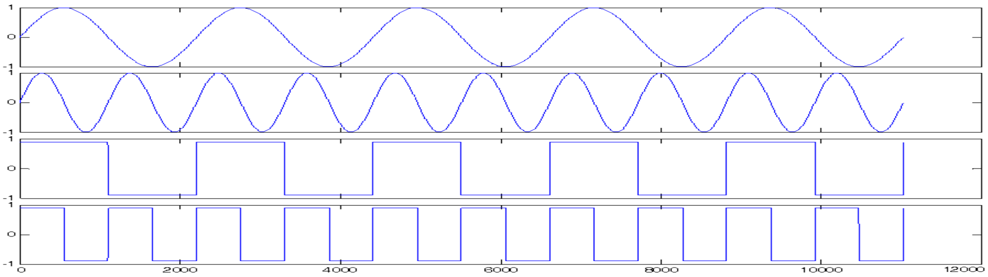


Figure 2.18 Test Signals. y axes show the signal amplitudes and x axes the samples.

Spectrograms of the test signals are shown in Figure 2.19.

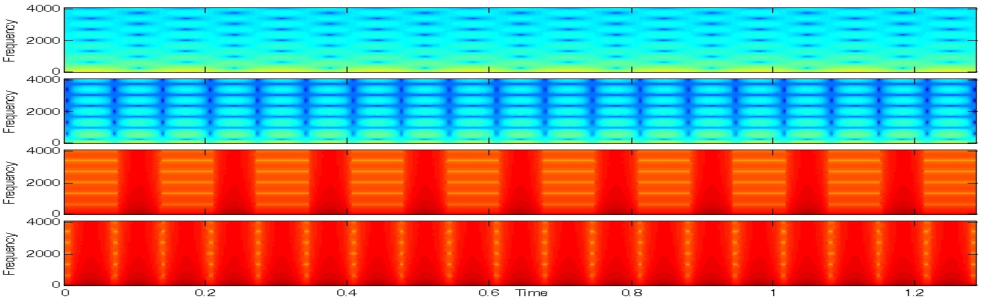


Figure 2.19 Spectrogram of the test signals.

The test signals were mixture with each other. The mixture signals are shown in Figure 2.20.

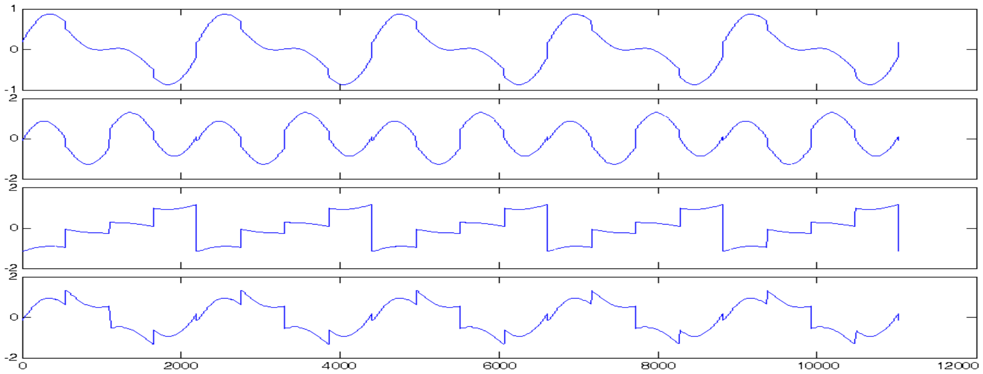


Figure 2.20 Mixture of Test Signals. y axes show the signal amplitudes and x axes the samples.

Fast ICA results were given in Figure 2.21. Note that Fast ICA recovers only the square waves. Spectrograms of the ICA result were given in Figure 2.22.

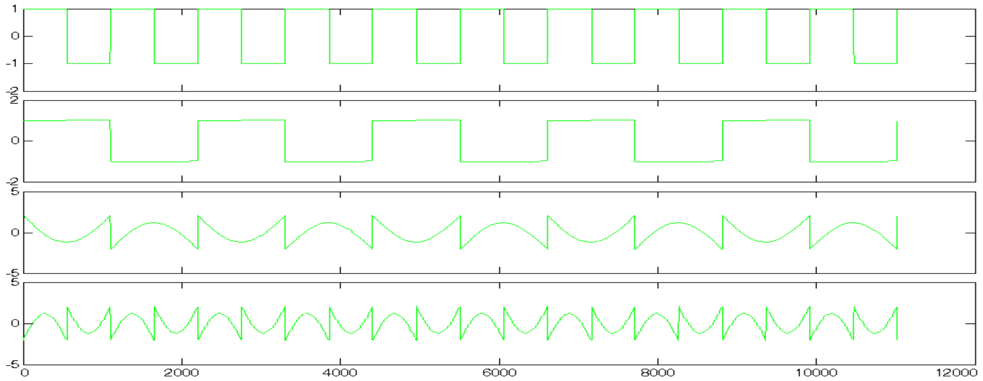


Figure 2.21 Fast ICA results. y axes show the signal amplitudes and x axes the samples.

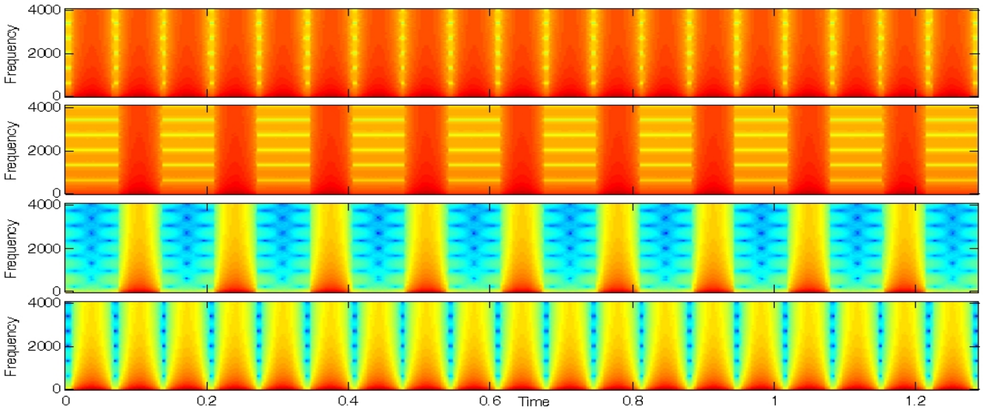


Figure 2.22 Spectrogram of Fast ICA results.

ML ICA results were given in Figure 2.23. Note that ML ICA results are not successful.

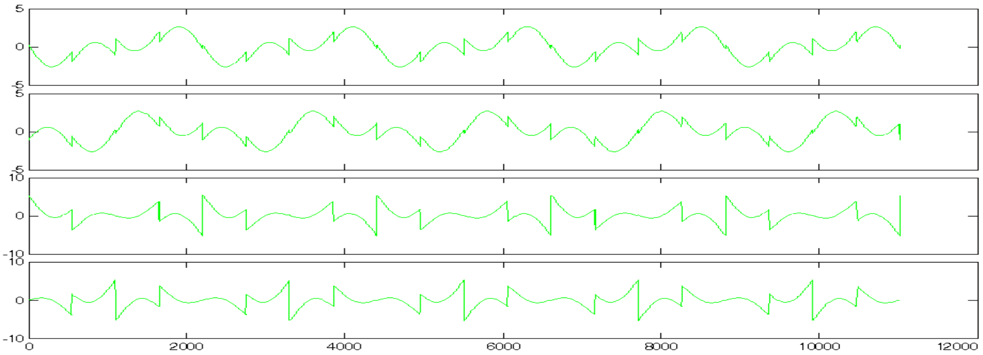


Figure 2.23 ML ICA Results. y axes show the signal amplitudes and x axes the samples.

Jade ICA results were given in Figure 2.24. Note that Jade ICA results are not successful. Shibbs ICA results were given in Figure 2.25. Note that Shibbs ICA results are not successful.

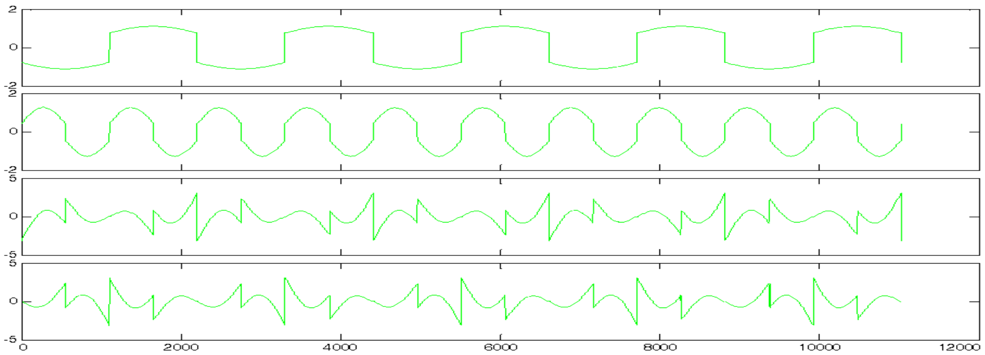


Figure 2.24 Jade ICA Results. y axes show the signal amplitudes and x axes the samples.

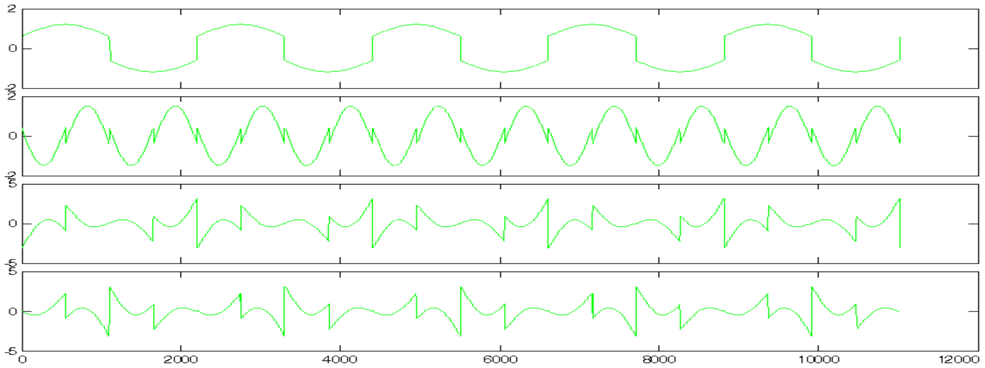


Figure 2.25 Shibbs ICA Results. y axes show the signal amplitudes and x axes the samples.

Here, we see that separation is not possible. Because, square waves fundamental frequencies are the same with the related sinusoidal waves. This means the signals are the same.

CHAPTER THREE
ICA REALIZATION PROBLEMS

Independent Component Analysis algorithms are usually very efficient on a simulation environment. For example, if noise added to a sound signal on a MATLAB environment, Fast ICA easily finds out the original sound component.

Although instantaneous mixing models are extensively studied, where many algorithms are proposed with promising results for separation, it fails to model in real life situations, for instance recording in a real room. In such a situation the microphones in the environment picks up not only the signals of the sources, but also the delayed and attenuated versions of the same signals due to reverberation. Hence, this can be viewed as microphones receiving a filtered version of different signals, and can be modeled as follows. Lee, Bell, and OrglMeister (1997a) have a study on real world signals blind source separation.

3.1 Phase Delay Problems

Instantaneous ICA cannot be applied to real-world problems. Noise affects the performance of blind source separation method. Also, sources are not mixed simultaneously, because the propagation of the signals through the medium is not instantaneous. There will be arrival time difference between the sources in the mixtures. Instantaneous separation methods simply cannot cope with these delays. Delay is a problem, because the probability distribution function (pdf) of the source data does not change but the pdf of the mixtures changes. So the separation process using ICA fails.

Also, another problem is the media that sound is emitted, if reverb effect occurs in that media the mixtures become convolutive mixtures. Blind separation of convolutive mixtures considers the combined blind de-convolution and instantaneous blind source separation problem. In this problem, there are several source (input) signals and several observed (output) signals just like in the instantaneous ICA problem. However, the source signals have different time delays in each observed signal due to the finite propagation speed in the medium. Each observed signal may also contain time-delayed versions of the same source due to multi-path propagations caused typically by reverberations in an acoustic environment.

Lee, Ziehe, Orglmeister, & Sejnowski (1998) show phase delay effect on blind source separation. For instant mixtures, separation is usually very good and straightforward. However, in the case of audio signals in a room, mixing process is will be more complex. Because of the delays of the recorded signals with respect to each others, delayed mixtures are the cases we need to investigate. In that kind of problems, information maximization approach can be used. Here entropy maximization is applied in the separated signals.

On information maximization or maximizing the entropy, separation is achieved by minimizing the mutual information of components of $y = g(u)$. g is a nonlinear function of cumulative density function of the independent signal s .

In a model of mixtures with delay, the signal components $s_1(t)$ and $s_2(t)$ in the mixtures are delayed with respect to one another. This was illustrated in Figure 3.1. The signals were created using MATLAB.

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t - d_{12}) \tag{3.1}$$

$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t - d_{21}) \tag{3.2}$$

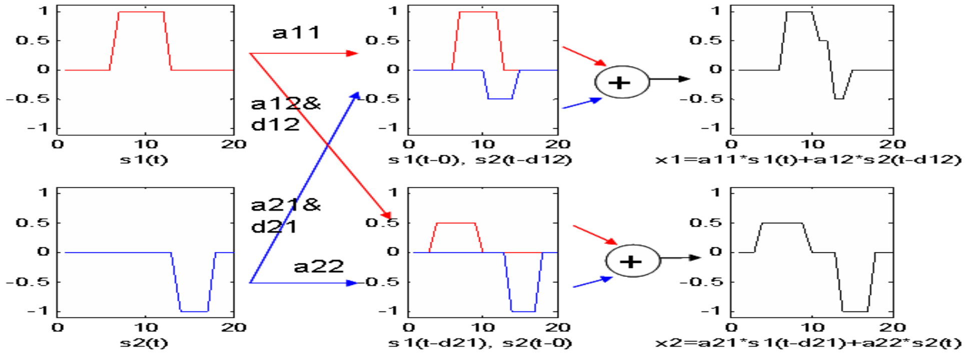


Figure 3.1 Two sources mixed with weights and delays.

Separation neural network can be designed like that for the delayed mixtures in Figure 3.2 (Torkkala, 1996).

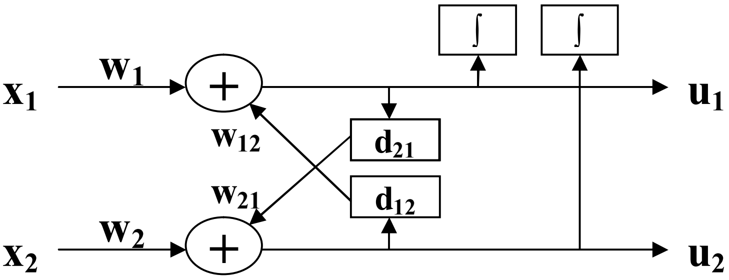


Figure 3.2 Block Scheme of the separation neural network

Network equations are shown in Equations 3.3 and 3.4:

$$u_1(n)=w_1x_1(n)+w_{12}u_2(n-d_{12}), y_1(n)=g(u_1(n)) \tag{3.3}$$

$$u_2(n)=w_2x_2(n)+w_{21}u_1(n-d_{21}), y_2(n)=g(u_2(n)) \tag{3.4}$$

The determinant for the Jacobian and the entropy of the network are:

$$\det(J)=\frac{\partial y_1}{\partial x_1}\frac{\partial y_2}{\partial x_2}-\frac{\partial y_1}{\partial x_2}\frac{\partial y_2}{\partial x_1}=y_1'y_2'D \tag{3.5}$$

$$\log(\det(J))=\log(y_1')+\log(y_2')+\log(D) \tag{3.6}$$

where

$$D=\left(\frac{\partial u_1}{\partial x_1}\frac{\partial u_2}{\partial x_2}-\frac{\partial u_1}{\partial x_2}\frac{\partial u_2}{\partial x_1}\right)=w_1w_2\text{, } y_1'=\frac{\partial y_1}{\partial u_1} \text{ and } y_2'=\frac{\partial y_2}{\partial u_2} \tag{3.7}$$

The adaptation rule for each parameter is derived by computing the gradient of log(det(J)) with respect to the parameter. For w_1 we obtain

$$\Delta w_1 \alpha \frac{\partial \log(\det(J))}{\partial w_1} = \frac{1}{y_1'}\frac{\partial y_1'}{\partial w_1}+\frac{1}{y_2'}\frac{\partial y_2'}{\partial w_1}+\frac{1}{D}\frac{\partial D}{\partial w_1} \tag{3.8}$$

Partial derivatives can be written as:

$$\frac{\partial y_1'}{\partial w_1}=\frac{\partial y_1'}{\partial y_1}\frac{\partial y_1}{u_1}\frac{\partial u_1}{\partial w_1}=\hat{y}_1y_1'x_1 \text{ and } \frac{\partial y_2'}{\partial w_1}=\frac{\partial y_2'}{\partial y_2}\frac{\partial y_2}{u_2}\frac{\partial u_2}{\partial w_1}=\hat{y}_2y_2' \tag{3.9}$$

$$\frac{\partial D}{\partial w_1}=\frac{\partial(w_1w_2)}{\partial w_1}=w_2 \tag{3.10}$$

where $y_i'=\frac{\partial y_i'}{\partial y_i}$ which depends on the cdf used.

So, the adaptation rule for w_1 becomes:

$$\Delta w_1 \alpha(\hat{y}_1x_1+1/w_2) \tag{3.11}$$

Using similar calculations, other adaptation rules can be calculated as in equations below:

$$\Delta w_2 \alpha (\widehat{y}_2 x_2 + 1/w_1) \tag{3.12}$$

$$\Delta w_{12} \alpha (\widehat{y}_1 u_2 (n - d_{12})), \qquad \Delta w_{21} \alpha (\widehat{y}_2 u_1 (n - d_{21})) \tag{3.13}$$

$$\Delta d_{12} \alpha (-\widehat{y}_1 w_{12} \dot{u}_2 (n - d_{12})), \qquad \Delta d_{21} \alpha (-\widehat{y}_2 w_{21} \dot{u}_1 (n - d_{21})) \tag{3.14}$$

Now, how the delayed mixtures separated to independent components, above method can be used. For example two signal mixed with A and d_{12} and d_{21}. Two signals are created using mixing_delayed.m (Appendix A.6). First one is sinusoidal signal and the other is sawtooth signal. First we have no delays. Original signals, mixture with no delay between original signals and FastICA results are in Figure 3.3. Figure 3.4 shows probability density & joint density functions of original signals, mixtures and FastICA results.

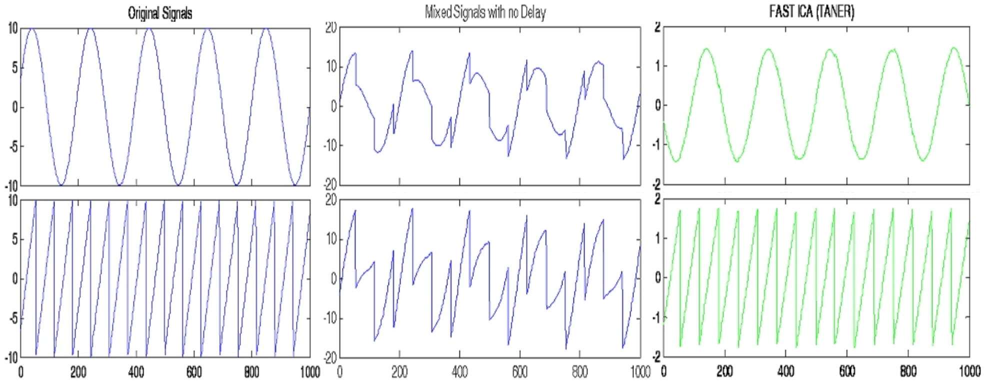


Figure 3.3 a) original Signals b) mixtures without delays c) Fast ICA results

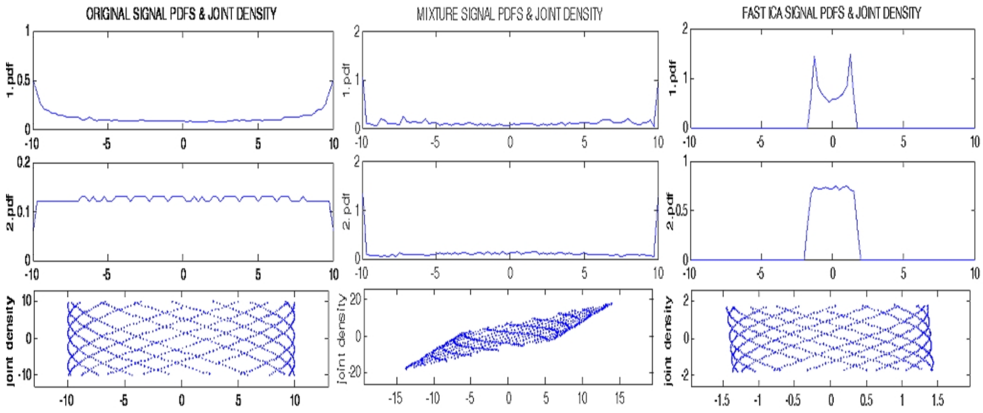


Figure 3.4 Pdfs and joint density graphics of a) original signals b) mixture signals c) Fast ICA results.

The second test was done with delays of $d_{12}=d_{21}=10$ in Figure 3.5. The third test was done with delays of $d_{12}=d_{21}=30$ in Figure 3.6. As seen below Figure 3.5b and Figure 3.6b, ICA can't recover the sources from delayed mixtures. We have to discover a method for solving this problem.

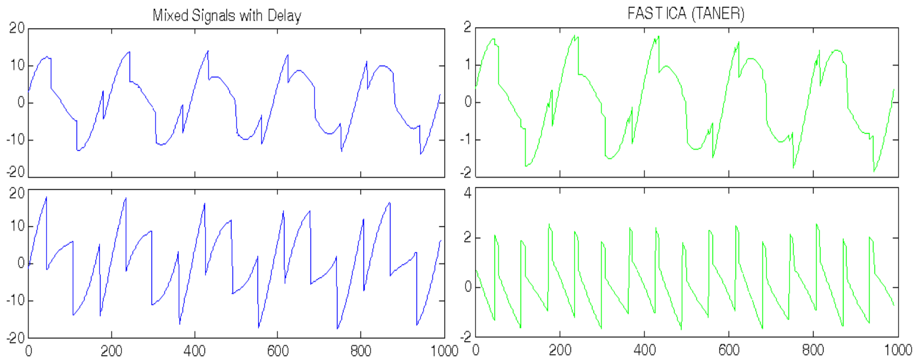


Figure 3.5 a) Ten Samples Delayed Mixtures b) Fast ICA results

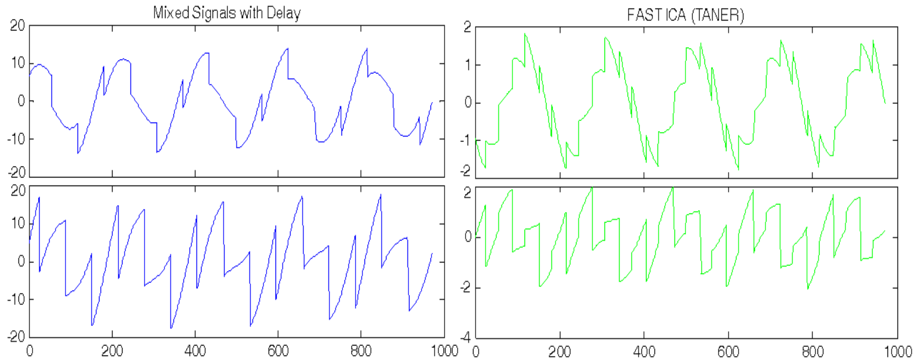


Figure 3.6 a) Thirty Samples Delayed Mixtures b) Fast ICA results

Now, another test can be done stating the problem in a different way. A Matlab code generates two signals with or without delays and mix them to find normal or delayed mixtures, then we find how Fast ICA performs? Here, test inputs are chirp and train sound. Chirp signal sampling frequency is 8192 Hz and delayed 20 samples that corresponds 20/8192=0.00244sec=2.44msec delay. Train signal sampling frequency is 8192 Hz and delayed 30 samples (30/8192 =0.00366sec=3.66msec) delay. Signals are emitted in air and accepting the sound speed in air is approximately 300m/s. The delays show that receive sensors (microphones) are placed 3e-3s*300m/s=0.9m from each others. Test results are displayed below (Figure 3.7- 3.9).

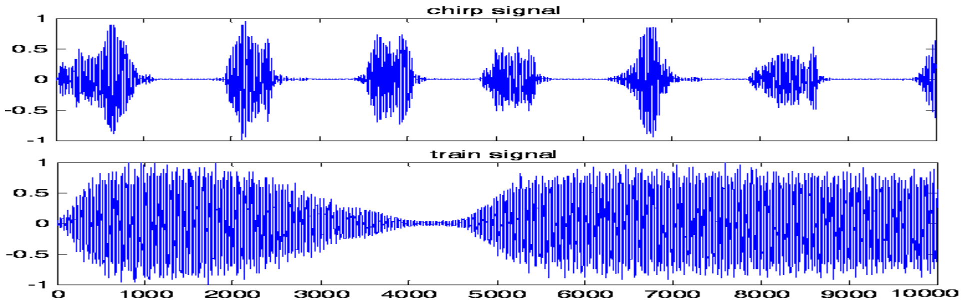


Figure 3.7 The Original signals.

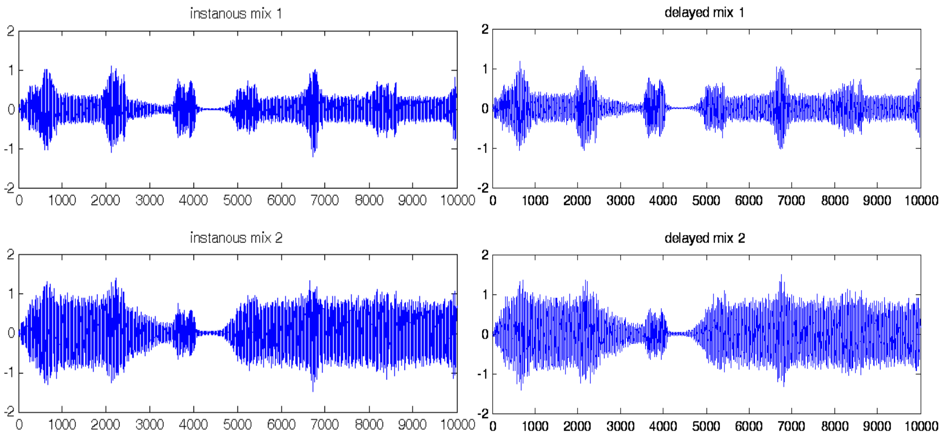


Figure 3.8 a) Instantaneous Mixtures b) Delayed Mixtures

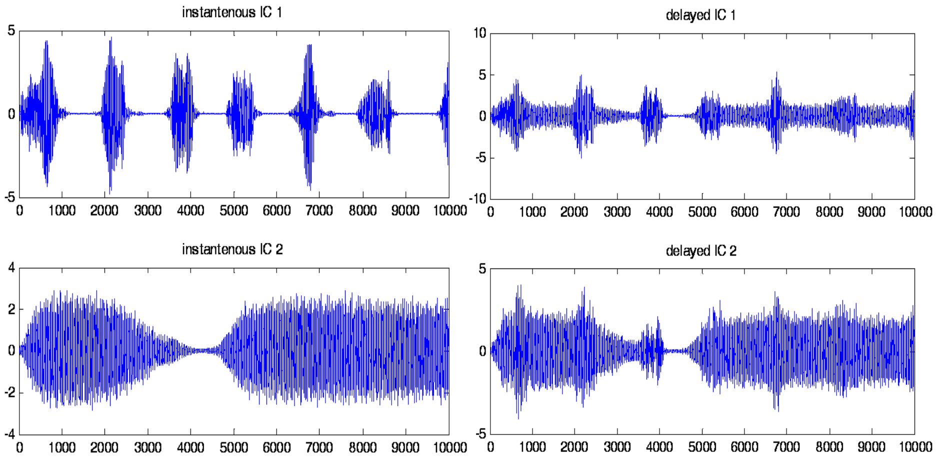


Figure 3.9 a) Instantaneous ICs by FastICA b) Delayed ICs by FastICA

FastICA performance is not so well with delays; even small delays can affect the performance. A more suitable algorithm can be found on the studies (Lee, Bell & Lambert, 1997b; Smaragdis, 1997; Parra & Spence, 2000; Pedersen, Larsen, Kjems, & Parra, 2007).

Another idea may be shifting one mixture respect to the other mixture and we search the maximum correlation value for all time band. Block scheme can be like this as in Figure 3.10.

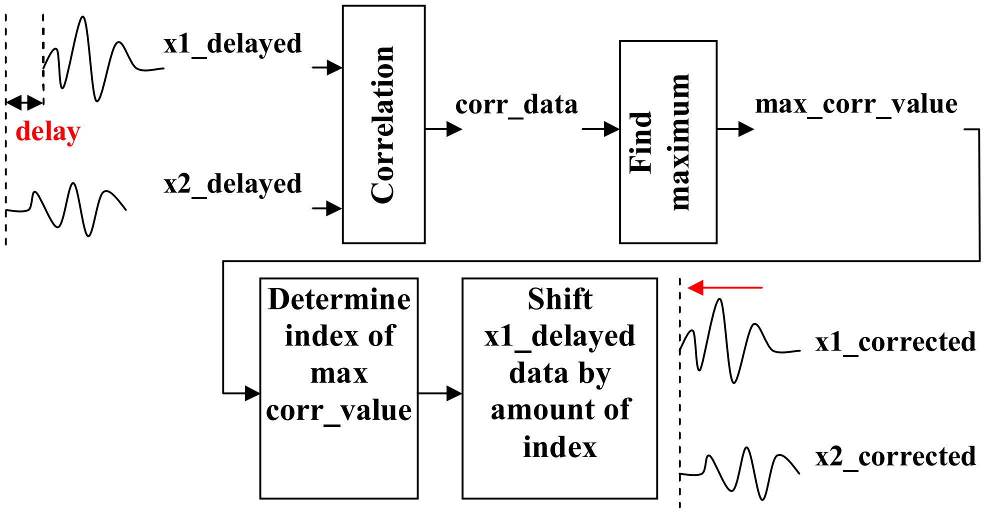


Figure 3.10 Block scheme of delay correction

For example, we have three source signals and two mixtures. Here, original signals are a laughter sound, a chirp signal, and a white noise (Figure 3.11). These original signals are mixed with a mixing matrix and zero delay matrix. We get only two mixtures from the three source signals without delays (Figure 3.12)

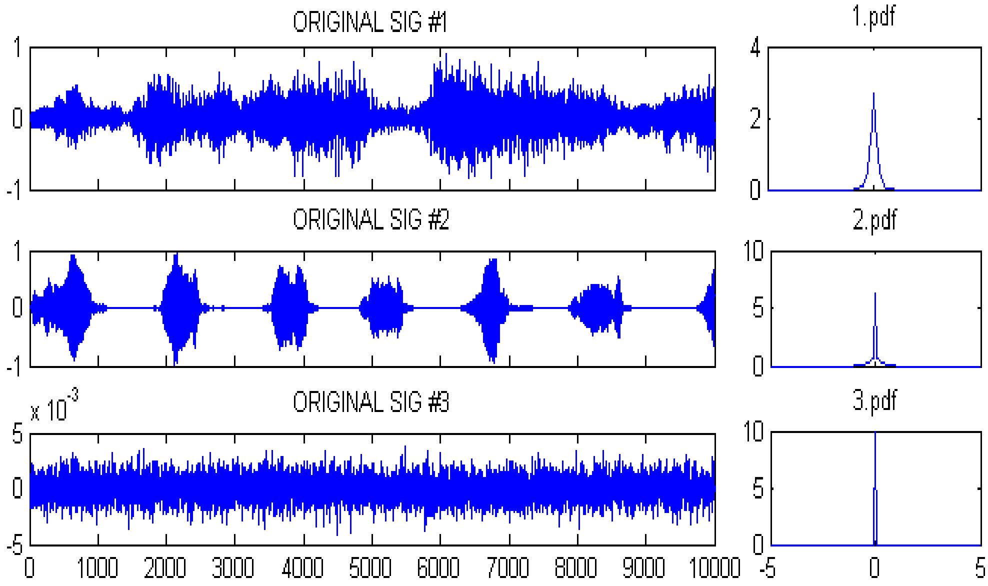


Figure 3.11 Original Signals

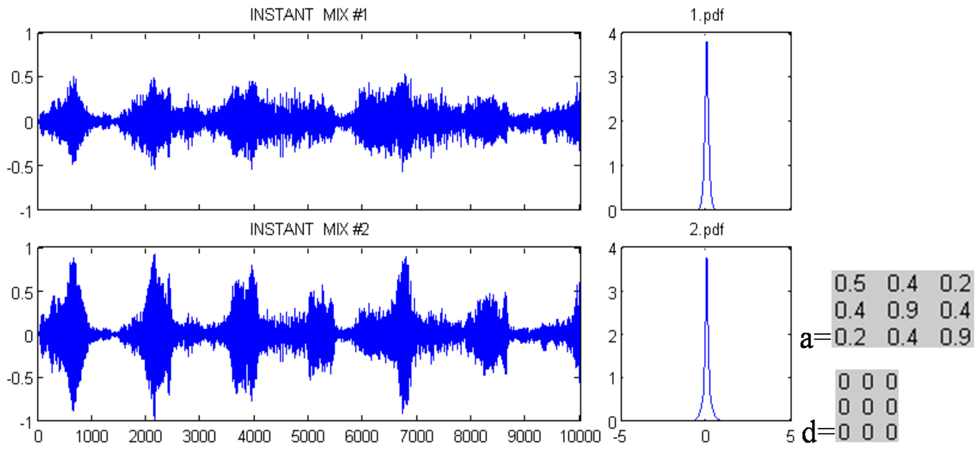


Figure 3.12 Mixtures without delays

These original signals are mixed with the same mixing matrix and delay matrix. We get only two mixtures from the three source signals with delays about 5 samples (Figure 3.13).

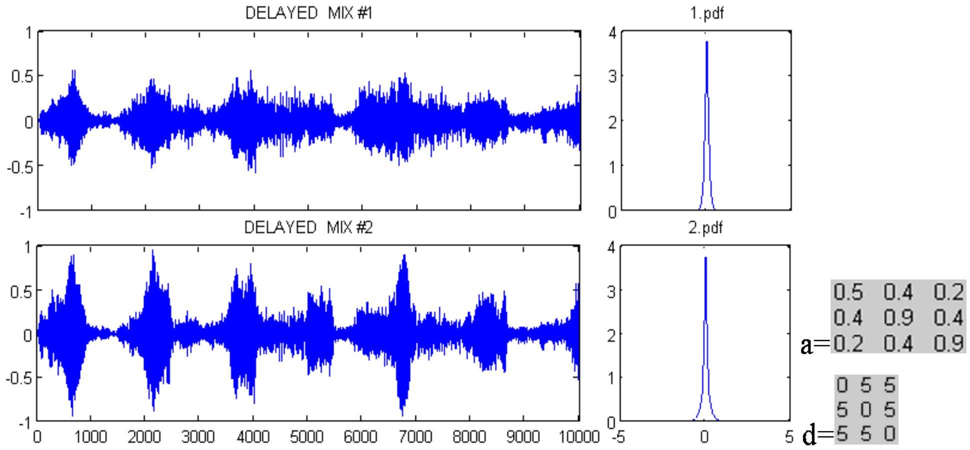


Figure 3.13 Mixtures with delays

Now we find the shift value for correcting one of the mixture time shifts. Correlation applied to these two mixtures and we found that delay is about 6 samples (Figure 3.14). Delayed mixture is shifted 6 samples and we get corrected mixtures (Figure 3.15). If we apply ICA to the non-delayed mixtures we get the following results with the unmixing matrix is in Figure 3.16. If we apply ICA to the delayed mixtures (non-corrected with the correlation algorithm) we get the following independent components with the de-mixing matrix is in Figure 3.17.

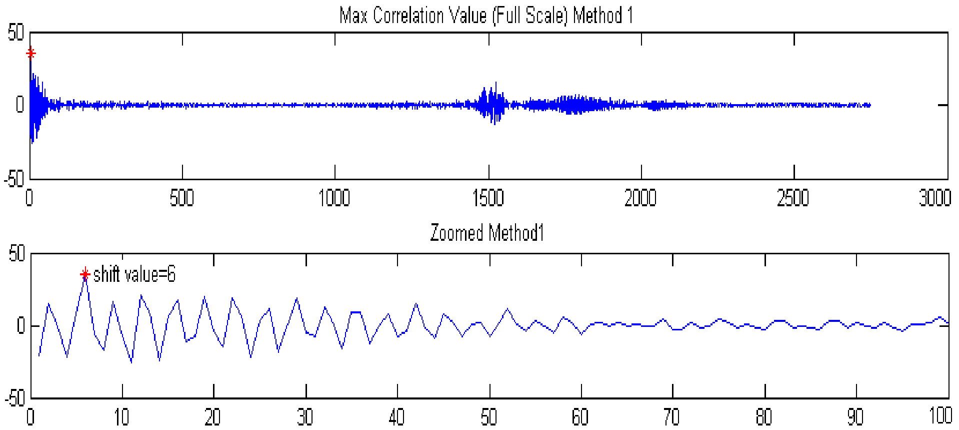


Figure 3.14 Finding delay correction value. Upper diagram denotes the maximum correlation value. Vertical axes denotes amplitude, horizontal provides the sample number. The lower chart provides a "zoom"ed version of the same chart. The shift delay value has been found to be 6 (asterix).

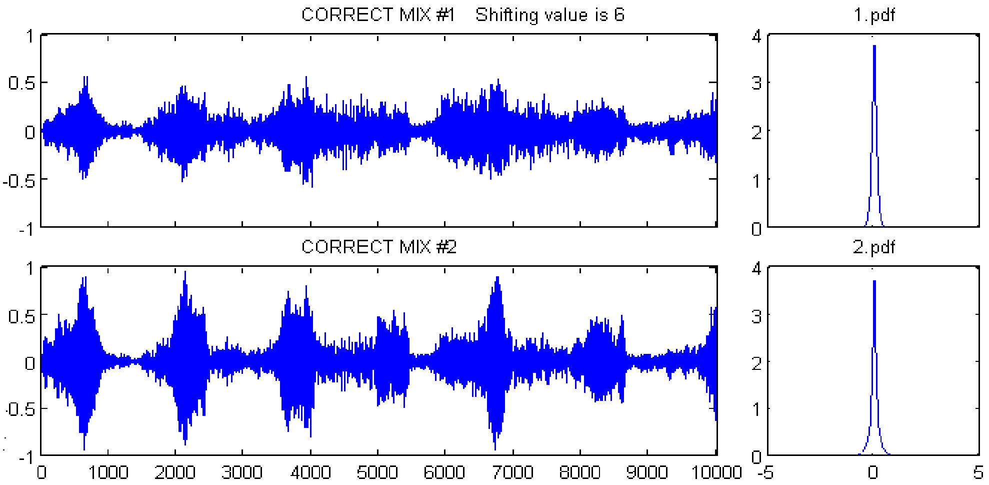


Figure 3.15 Corrected mixtures with shifting one mixture using delay correction.

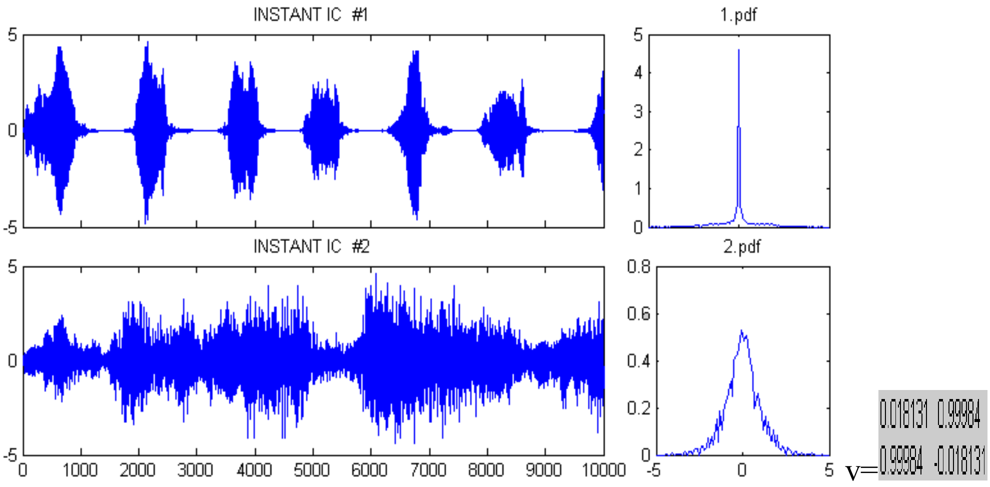


Figure 3.16 ICA results using mixtures without delays. (pdf=probability distrubion function)

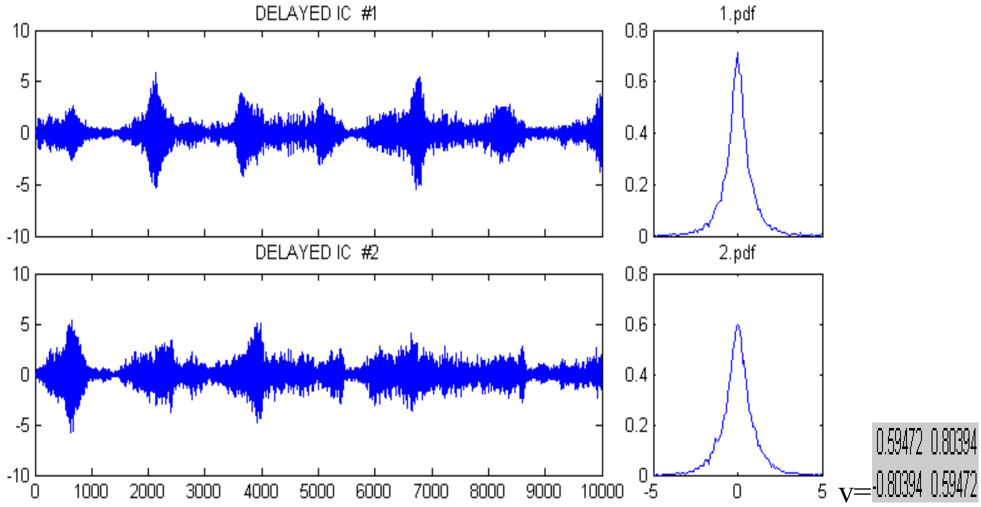


Figure 3.17 ICA results using mixtures with delays.

ICA results from corrected mixtures with the unmixing matrix are in Figure 3.18. We get better performance from corrected mixtures as seen below.

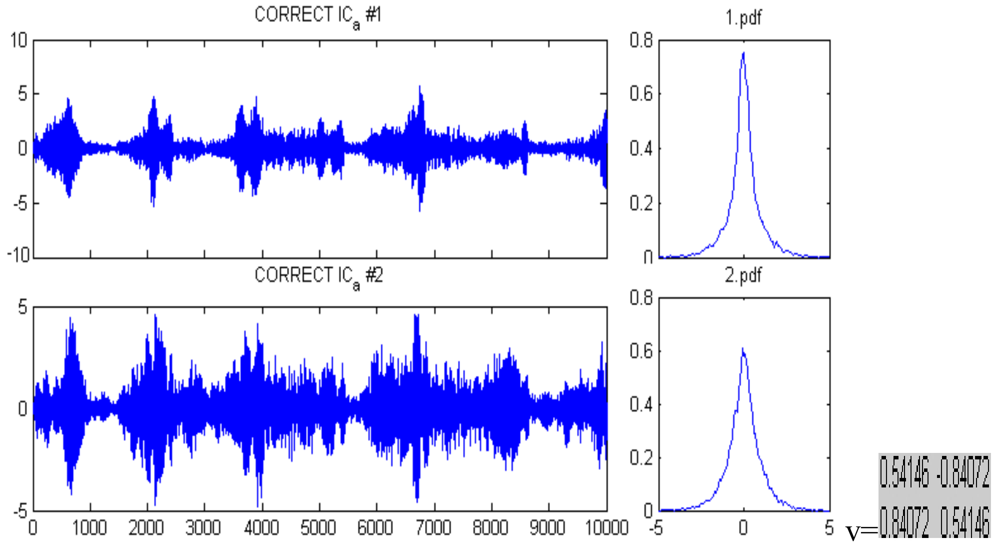


Figure 3.18 ICA results using mixtures after delay correction

3.2 Sound separation using less sensors

Normally, separation is not possible when number of mixtures is greater than number of sensors using ICA (i.e. N mixtures for two sensors). But, if the mixtures can be divided subspaces and one of the subspaces holds the information which we want to intend to get, separation is more possible.

One pre-processing method is enhancing desired source signal and suppressing the other sources. When an amplitude dominant signal is mixed with the other signals, ICA method tends to find and separate this signal which is close to the receiving sensor. The enhancement of the desired source signal is done by using a filter mask which is applied on spectrogram domain. Our aim is using this single channel separation method to emphasize our desired signal in the mixture and ease the work of ICA and increasing the performance of the ICA algorithms. Also, using this method we can really separate the signals using fewer sensors.

Only using one sensor, still speaker separation is possible (Pedersen, Wang, Larsen, & Kjems, 2005; Reddy & Raj, 2004). More generally, sound separation can be made easier using the same concept. Extraction of a desired sound signal from a sound mixture can be done also using masking algorithms. To do that, spectrogram of the mixture signal is obtained and multiplied with a known mask spectrogram which is derived from the desired signal. Then the result spectrogram is inversed to time domain. If the mask is perfect, only we get the desired signal, and the other signals in the mixture are ignored. This is a filtering application in fact, but the difference is not only noise filtering we can also filter other signals. Figure 3.19 shows flow scheme of masking.

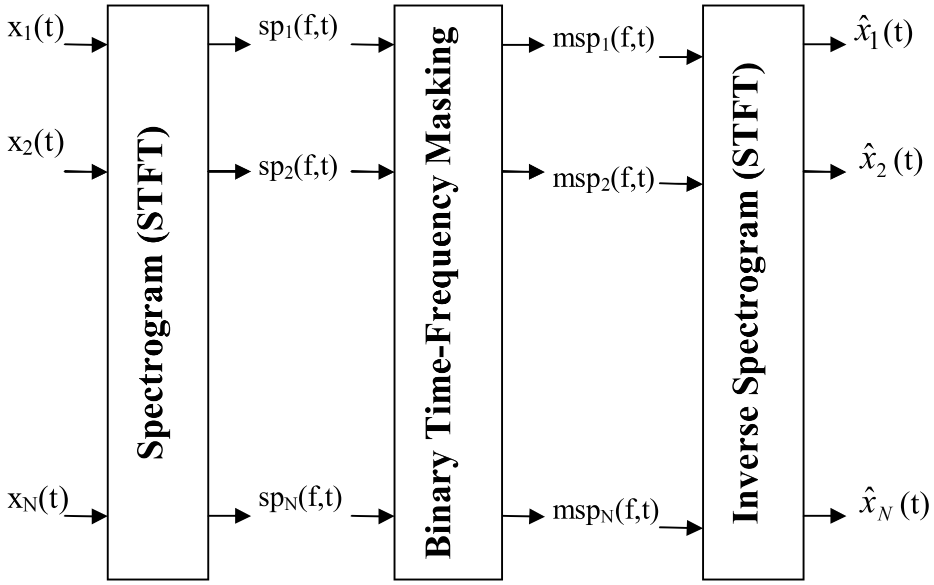


Figure 3.1 Flow of Time-Frequency Masking Method

3.2.1 Time Frequency Transformations

Time-Frequency Transformations will be explained and reviewed. Advantages and disadvantages are given for transformation types. The type of transformation will be the most suitable for our research will be investigated. Booshash (2003) gives details about time-frequency signal analysis.

In time domain, we can't see some features about the signal. Figure 3.20 explains this. Here, the independent variable is time. But as we consider the mixture signal, three different sinusoidal signals with different frequencies are hidden. A transform is needed for obtaining further information. This transform must give us this signal frequency structure. The most popular transform is Fourier transform. The other types are Hilbert Transform, Short-time Fourier Transform, Wigner Distributions, the Radon Transform, the Wavelet Transform, and so on. So, a frequency analysis is needed and it shows which frequencies exist in the signal.

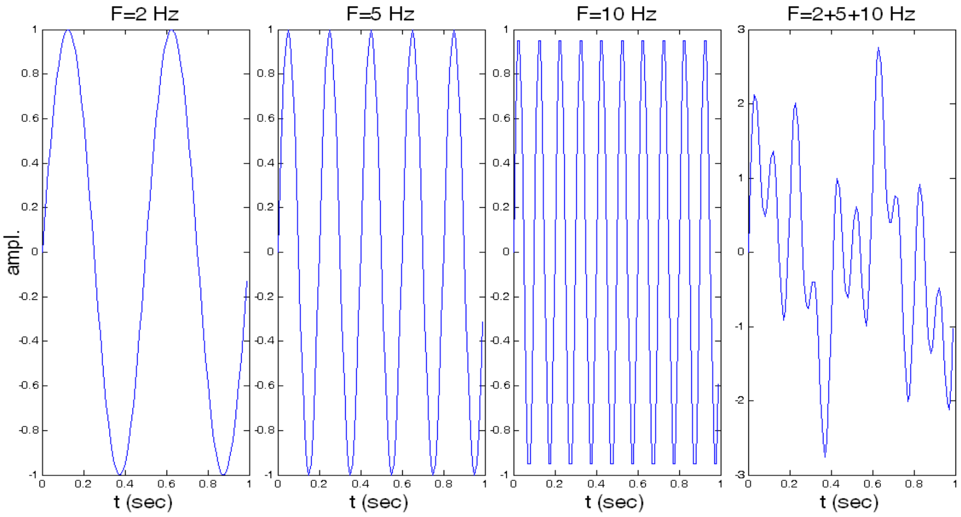


Figure 3.20 A sinusoidal wave in time domain.

3.2.1.1. Fourier Transform

Fourier Transform (F.T.) is defined for the continuous or discrete signals as follows:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-2j\pi ft}dt \, , \quad X(t) = \int_{-\infty}^{\infty} X(f)e^{2j\pi ft}df \qquad (3.15)$$

$$X(k+1)=\int_{-\infty}^{\infty} x(n+1)W_N^{kn}\ ,\ \ x(n+1)=\frac{1}{N}\int_{k=0}^{N-1} X(k+1)W_N^{-kn} \tag{3.16}$$

where $W_N = e^{-j\left(\frac{2\pi}{N}\right)}$.

The signals can be categorized into two classes: stationary and non-stationary signals. Frequency content does not change in time and all frequency components occur at all times in frequency domain for stationary signals. Frequency changes in time and do not appear all time in frequency domain for non-stationary signals (Figure 3.21).

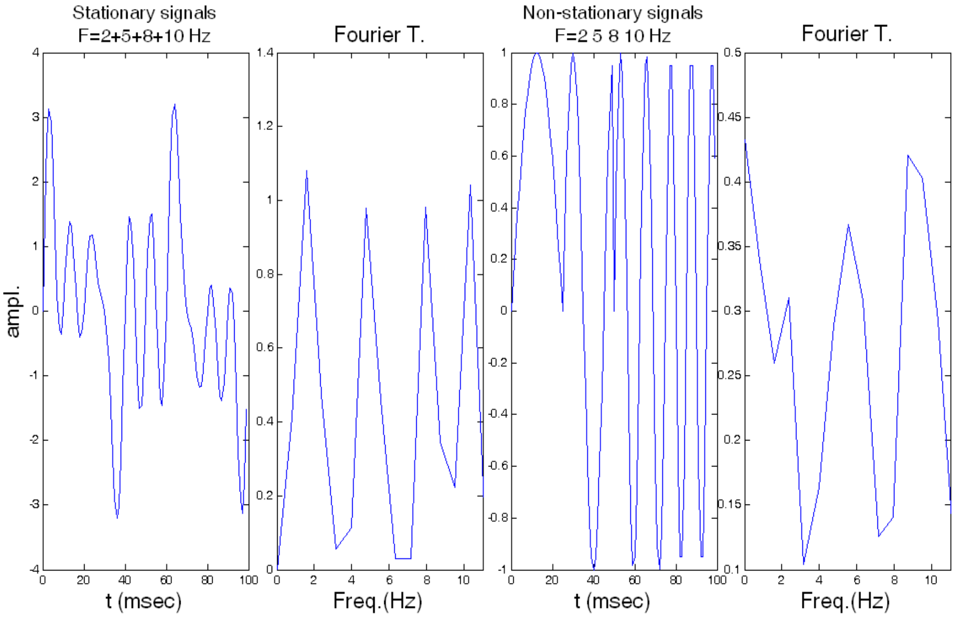


Figure 3.21 a) Stationary Signal and its F.T. b) Non-stationary Signal and its F.T.

As an example chirp signal is a sinusoidal waveform that its frequency changes linearly. In the Figure 3.22 first chirp signal frequency sweep is from 2 Hz to 20 Hz and the second one is from 20 Hz to 2 Hz. They are different signals in time domain, but in frequency domain they are the same. That’s why Fourier Transform cannot show the frequency component occurring times. So, the disadvantages of Fourier Transforms are; FT only tells the frequency components existence in the signal and the time and frequency information cannot be seen at the same time.

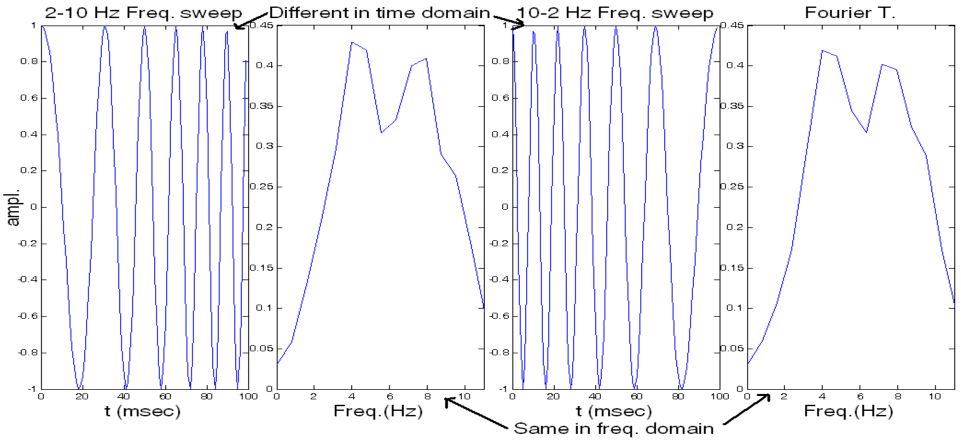


Figure 3.22 Fourier Transformation of Chirp Signal.

3.2.1.2 Short Time Fourier Transformation

Better solution is to use Short Time Fourier Transform (STFT) which is first used by Dennis Gabor. STFT analyzes only a small section of the signal at a time via using a window segment which is assumed as stationary (Figure 3.23).

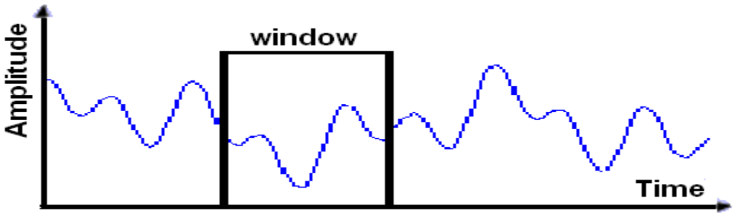


Figure 3.23 Short Time Fourier Transform Windowing

STFT is a function of time and frequency and formula is given below.

$$STFT_x^{(w)}(t',f) = \int_t \left[x(t) \bullet w^*(t-t') \right] \bullet e^{-j2\pi ft} dt \tag{3.17}$$

where w(t) is the windowing function.

Some drawbacks of STFT are; window size is constant, narrow window selection yields poor frequency resolution, and wide window selection means poor time resolution (Polikar, 1996). Figure 3.24 shows this comparison. Also, there is no knowledge about which time intervals have which frequency components (Heisenberg uncertainty principle).

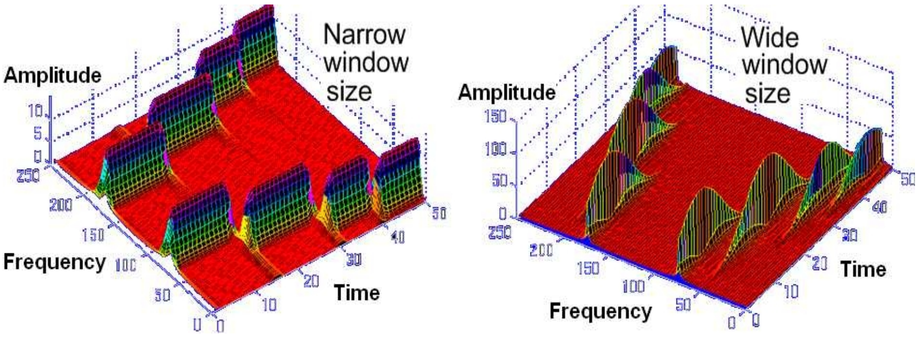


Figure 3.24 Narrow and wide window size comparison (Polikar, 1996).

3.2.1.3 Spectrogram

Spectrogram is the square of the STFT, a sort of running spectrum.

$$SP(t,w) = \left|STFT(t,w)\right|^2 \tag{3.18}$$

For example, the spectrogram of a chirp signal with linear instantaneous frequency deviation can be seen in Figure 3.25. Chirp signal sample frequency is 1 kHz, and starts from 0 Hz and crosses 100 Hz at 1 sec.

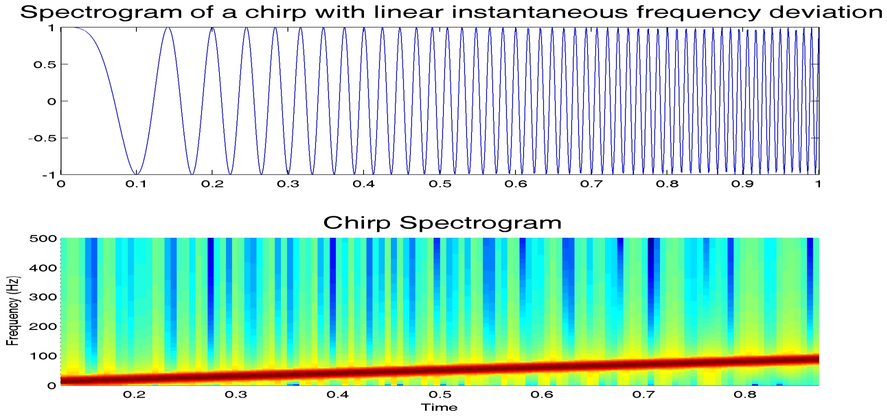


Figure 3.25 Example of spectrogram obtained using Matlab.

3.2.1.4. Gabor expansion

The Gabor expansion is defined as

$$s(t) = \sum_m \sum_n c_{m,n} h(t - mT) e^{jn\Omega t} \tag{3.19}$$

$$STFT(t,w) = \int s(\tau)\gamma(\tau - t)e^{-jw\tau}d\tau \qquad (3.20)$$

$$s(t) = \sum_{m}\sum_{n} c_{m,n}h(t - mT)e^{jn\Omega t} \qquad (3.21)$$

The Gabor expansion is actually a sampled STFT (Figure 3.26). If $n\Omega = 2\pi$, critical sampling is selected. If $n\Omega < 2\pi$, this is called redundant sampling.

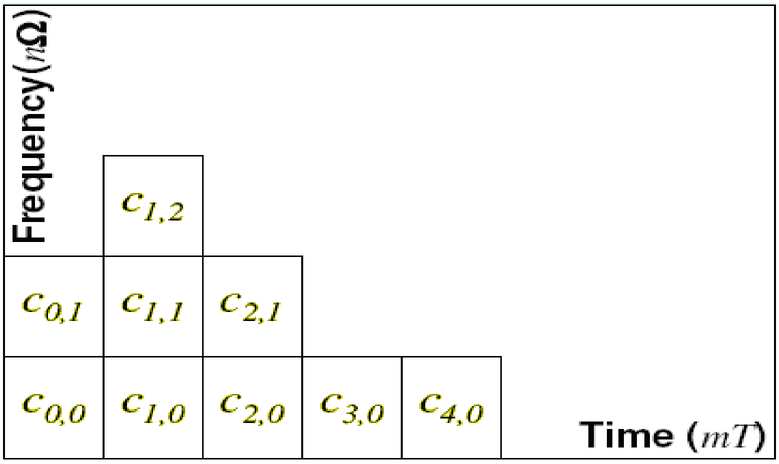


Figure 3.26 Gabor expansion sampling.

Gabor coefficients can be found using Equation 3.22:

$$c_{m,n} = \int s(t)\gamma^{*}(t - mT)e^{-jn\Omega t}dt = STFT(mT, n\Omega) \qquad (3.22)$$

So, the dual function γ^{*} needs to be calculated.

3.2.1.5 Wavelet Transform

Solving the resolution problem of STFT is possible with wavelet transform. It is possible to analyze the signal at different frequencies with different resolutions. Time resolution is good and frequency resolution is poor at high frequencies, and the reverse at low frequencies. Wavelet transform is more suitable for short duration of higher frequency; and longer duration of lower frequency components. Wavelets have some advantages over STFT; window width changes for every spectral component during transformation, and different resolutions occur.

Wavelet means the finite length window function. Wavelet transform separates the signal into signal components which represented by different frequency bands and which frequency bands exist at which time intervals can be seen. Continuous Wavelet transform (CWT) is defined as;

$$CWT_x^{\psi}(\tau,s)=\Psi_x^{\psi}(\tau,s)=\frac{1}{\sqrt{|s|}}\int x(t)\bullet\psi^*\left(\frac{t-\tau}{s}\right)dt \qquad (3.23)$$

where τ is translation and defines the location of the window, s is the scale parameter. $\psi^*\left(\frac{t-\tau}{s}\right)$ is the mother wavelet. Mother wavelet is a prototype for generating the other window functions and all the used windows are its dilated or compressed and shifted versions. Scale parameter dilates (s>1) or compresses (s<1) the signal. If high scale is selected, entire signal is spanned but not in details. If low scale is selected, there is a detailed view but in a short time of duration.

CWT algorithm steps are:

1) Wavelet window is at the position of the signal beginning.
2) Set the scale parameter equal to 1 (s=1).
3) Integrate the multiplication of the signal with the wavelet function.
4) Shift the wavelet to $t=\tau$, and get the transform value there.
5) Go to step 3 till the end of the signal.
6) Increase s with a small value; repeat 3 to 5 for all s.
7) CWT is calculated.

Decomposing the signal into the sub bands is shown in Figure 3.27. Also, inverse transform is needed for getting the signal back without loss of information. The wavelet coefficients are used for reconstruction. Wavelet analysis involves filtering and downsampling, and reconstruction process consists of upsampling and filtering.

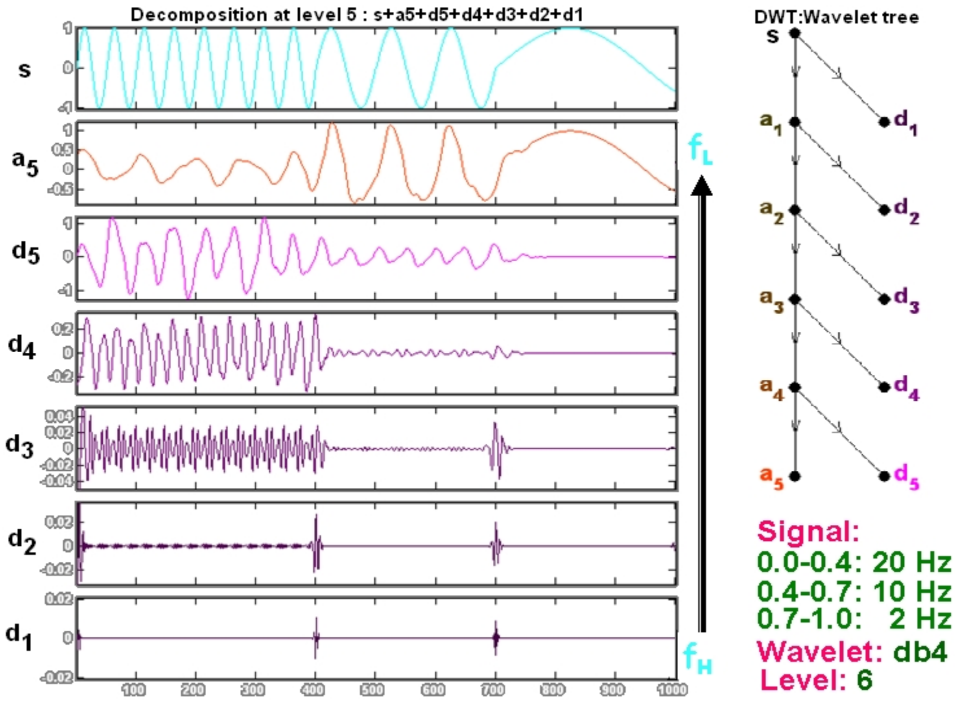


Figure 3.27 Decomposing of Non-Stationary Signal.

3.2.2 Comparison of Time Frequency Conversion Methods

Before making a comparison, Heisenberg uncertainty principle must be known. The uncertainty relationship states that one cannot measure a frequency, and the time at which the frequency occurs, simultaneously with infinite accuracy. The product of time-resolution and bandwidth is a constant close to one, therefore a proper consistency is required between them (Figure 3.28).

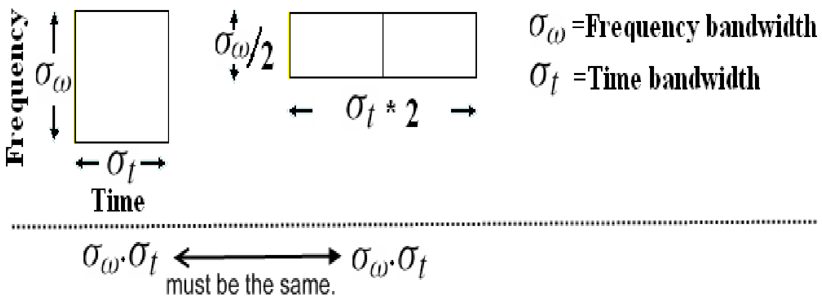

Frequency
σ_ω
σ_t
Time
$\sigma_\omega/2$
$\sigma_t * 2$
σ_ω =Frequency bandwidth
σ_t =Time bandwidth
$\sigma_\omega \cdot \sigma_t$
must be the same.
$\sigma_\omega \cdot \sigma_t$

Figure 3.28 Energy Bandwith Products must be the same

Resolution of time and frequency is shown in Figure 3.29. Every box area is equal. Resolution can be selected for different sizes. Note that, resolution in STFT is selected at the beginning and constant. Here the horizontally large and vertically narrow box means frequency resolution is good, but time resolution is poor. In contrary, horizontally narrow and vertically large box means frequency resolution is poor, but time resolution is good.

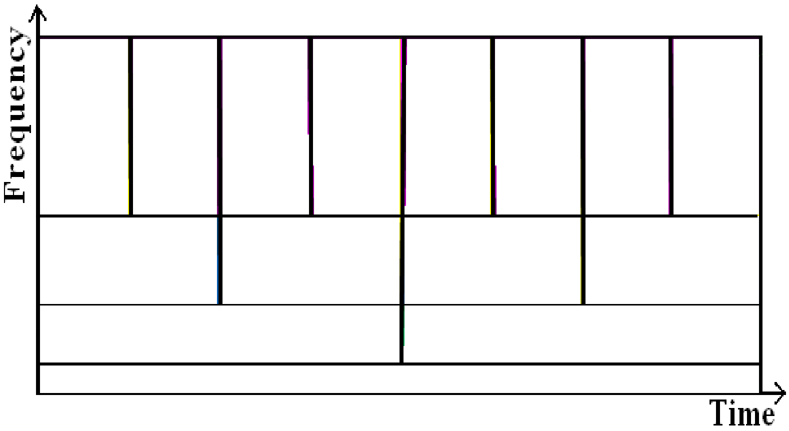


Figure 3.29 Time and Frequency resolution

σ_t and σ_w can be calculated for a function *s*(*t*) using equations 3.24.

$$\sigma_t^2 = \int (t-\mu_t)^2 |s(t)|^2 dt \qquad \sigma_w^2 = \int (w-\mu_w)^2 |s(w)|^2 dw \tag{3.24}$$

since σ determines the spreading of the energy, we want a small σ_t as well as a small σ_w. But, according to Heisenberg's uncertainty principles we will always have Equation 3.25 for any function:

$$\sigma_t \sigma_w \geq \frac{1}{2} \tag{3.25}$$

and only gaussian functions achieve equality.

The comparison of transformations is in Figure 3.30a and the performance of the methods is shown in Figure 3.30b. Here, precision or accuracy means how details of the signal in time converted into time-frequency domain successfully. Security means how successful inverse transforming into time domain again. Window size of the wavelet is automatically adjusted; therefore it is a better method than STFT.

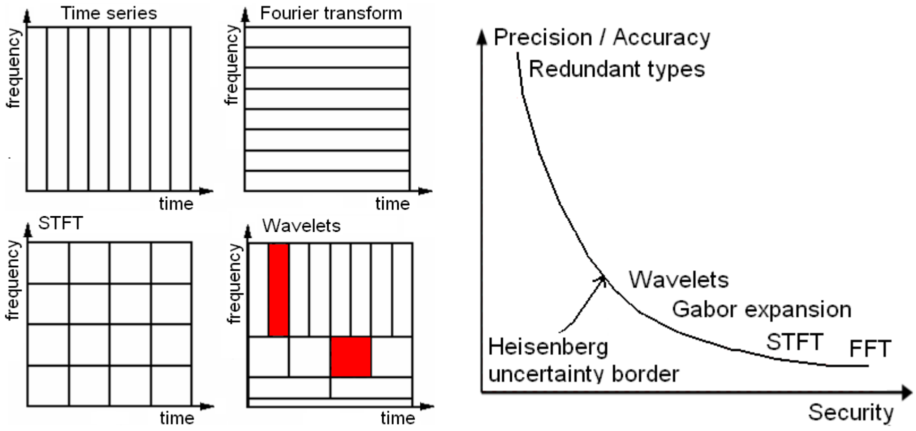


Figure 3.30 a) Transformation comparison b) Precision versus security graph

3.2.3 Time Frequency Masking

In another approach, time frequency masking gets two inputs, one of them is directly from the output of short time Fourier transformation and the other is using ICA after short time Fourier transformation and then basis vector clustering. The method is based on a two-stage process where independent component analysis (ICA) is first employed in each frequency bin and then time-frequency masking is used to improve the performance further (Sawada, Araki, Mukai & Makina, 2006). The flow of this method is in Figure 3.31.

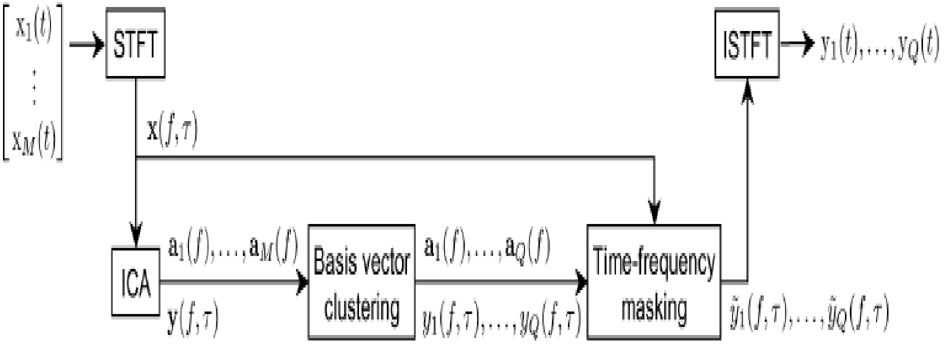


Figure 3.31 Flow of another Time-Frequency Masking approach (Sawada, Araki, Mukai & Makina, 2006).

In spectrogram graphics (Figure 3.32), we can see that the desired source can be recovered by certain success, but combination of ICA and T-F masking, some frequency components can be clearly recovered (from 250 Hz to 450 Hz portion).

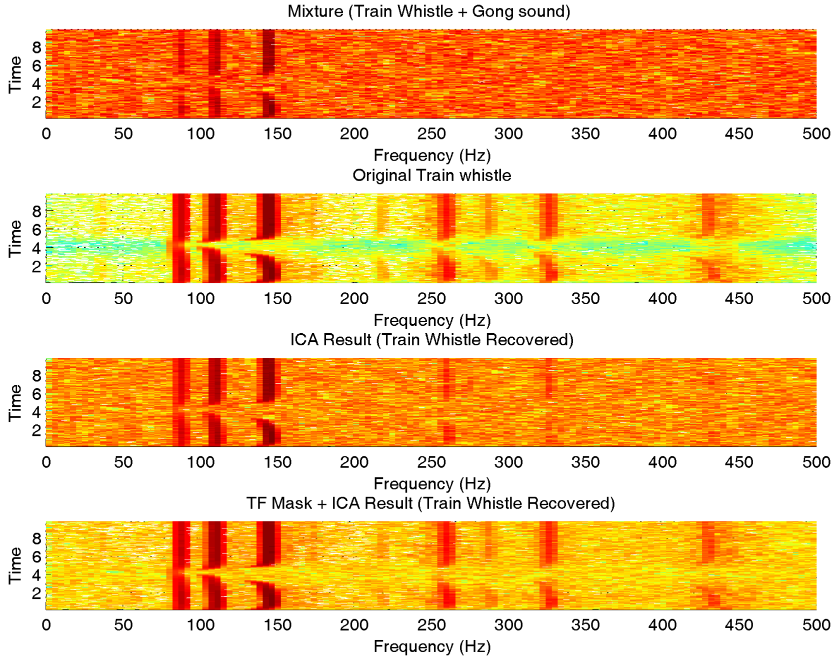


Figure 3.32 ICA performance enhancement using TF-masks.

Binary like masks are sometimes called hard masks. The mask has only two levels: "1" for accepting, "0" for removing. Some features of a signal are lost because the decision is so strict: pass or do not pass. The solution is soft mask that has more levels. One method is finding a ratio mask based on SNR ratio. Another method is finding a soft-mask based on an estimation of the weights for the frequency sub-bands of mixture.

Masking can be used for signal separation and it is not like ICA unless the mask is generated from the mixture signal without knowing about the desired signal. If we derive the mask using the desired signal this can be only used for filtering.

For the first method (Srinivasan, Roman, & Wang, 2004), we want to find a signal partly suppressed by other signals. There is no need to have the knowledge of complete spectrum for signal recognizing and this is called as missing data approach. But in our case, recovering the signal is important.

There are two steps for missing data approach; Identifying reliable evidences and recognizing the signal using this incomplete evidence.

For the first step, a mask is placed on the spectral data and is defined using local SNR. A speech and a noisy speech spectrogram is an example. Speech mask is obtained from the speech and the SNR Mask is calculated from the SNR ratio. Here, Auditory Scene Analysis (ASA) and SNR estimation can be done together to find better SNR mask (Figure 3.33). ASA is a pre-processing stage for robust automatic speech recognition and requires no noise model. ASA separates out the speech evidences from other sound sources, but fails getting all speech evidences especially if speech spectro-temporal regions are contaminated with other sounds.

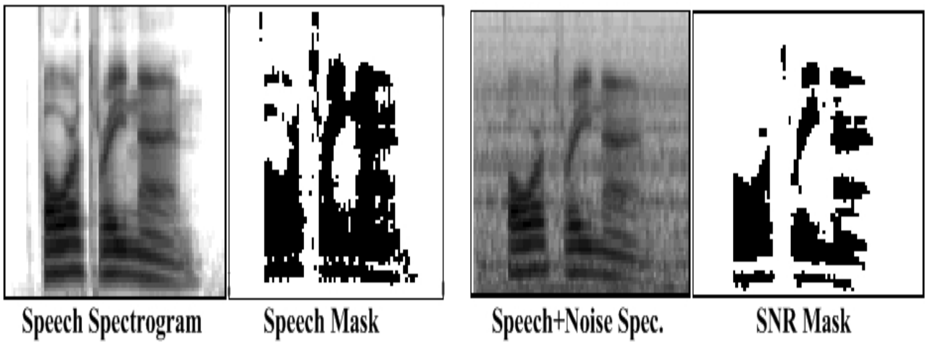


Figure 3.33 Speech and SNR mask example (Srinivasan, Roman, & Wang, 2004).

For the second step, there are two approaches. One can either estimate the missing values and then proceed as normal (missing data imputation) or use the distribution of the remaining values alone (marginalization). Marginalization generally outperforms imputation. Both techniques can be improved by the additional use of counter-evidence: even if we don't know the true speech value for some time-frequency pixel we can put a bound on it: the speech energy cannot be greater than the energy in the mixture. Thus speech sounds which require more energy than the total available can be rejected.

For example, two re-scaled output signals, $\hat{y}_1(n)$ and $\hat{y}_2(n)$, are transformed into the frequency domain e.g. using the Short-Time Fourier Transform STFT so that two spectrograms are obtained as in Equation 3.26:

$$\hat{y}_1 \rightarrow Y_1(\omega, t), \;\; \hat{y}_2 \rightarrow Y_2(\omega, t) \tag{3.26}$$

where ω denotes the frequency and t is the time frame index. The binary masks are then determined for each time-frequency unit by comparing the amplitudes of the two spectrograms as in Equation 3.27:

$$\begin{aligned} \mathrm{BM}_1(\omega, t) &= \tau\,|Y_1(\omega, t)| > |Y_2(\omega, t)| \\ \mathrm{BM}_2(\omega, t) &= \tau\,|Y_2(\omega, t)| > |Y_1(\omega, t)| \end{aligned} \tag{3.27}$$

where τ is a threshold. Next, each of the two binary masks is applied to the original mixtures in the T-F domain, and by this non-linear processing; some of the speech signals are removed by one of the masks while other speakers are removed by the other mask. After the masks have been applied to the signals, they are reconstructed in the time domain by the inverse STFT. If there is only a single signal left in the masked output, a selection criteria is needed (Bell, & Sejnowski, 1997; Abdallah, & Plumbley, 2001). Figure 3.34 shows the block scheme of the system.

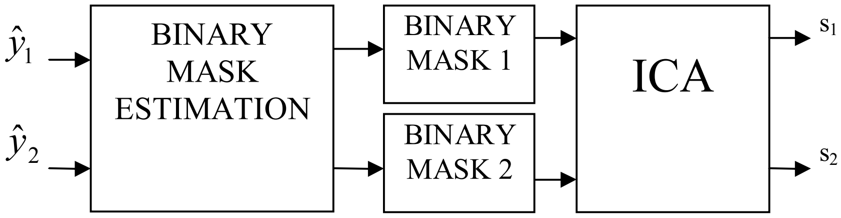


Figure 3.34 Masking the noisy original source using inverse spectrogram

3.2.4 Time Frequency Enhanced ICA Method

A block diagram of the mixing process is in Figure 3.35.

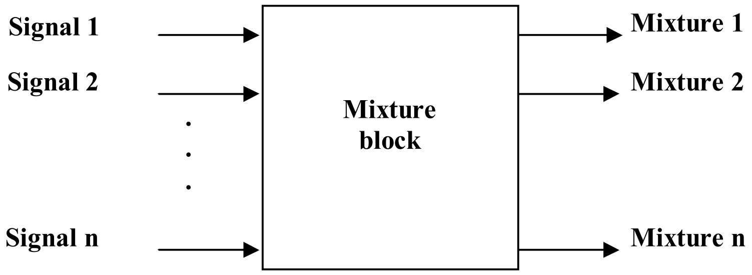


Figure 3.35 A block diagram of Mixing Sound Signals.

In order to introduce time-dependency in the Fourier transform, a simple and intuitive solution consists in pre-windowing the signal x(u) around a particular time t, calculating its Fourier transform, and doing that for each time instant t. The resulting transform is called the Short-Time Fourier Transform (STFT). A block diagram of creating masks is in Figure 3.36.

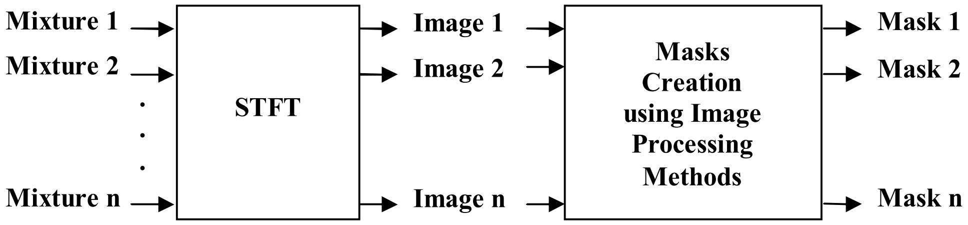


Figure 3.36 A block diagram of finding Masks

A block diagram of applying masks to the STFT images is in Figure 3.37.

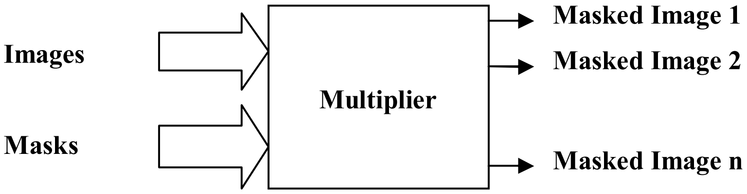


Figure 3.37 A block diagram of applying masks.

A block diagram of getting recovered sound mixtures back from T-F domain into time domain is in Figure 3.38.

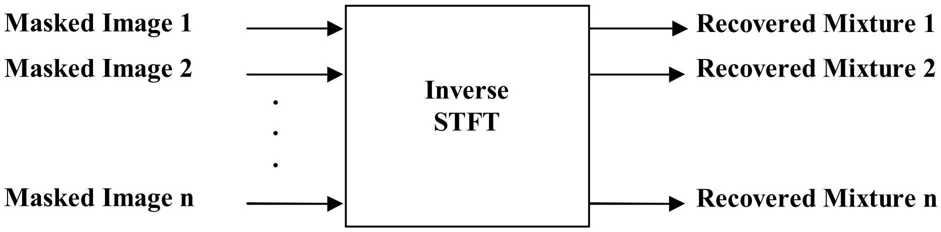


Figure 3.38 A block diagram of getting mixtures back.

Now, Independent Component Analysis can be applied to these enhanced mixtures to get the original signals back more successfully (Figure 3.39).

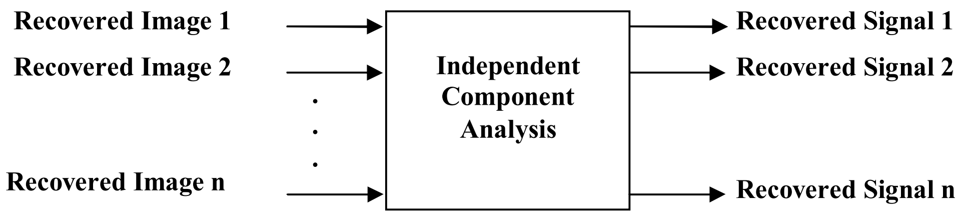


Figure 3.39 A block diagram of Independent Component Analysis.

3.2.5 Application of Time-Frequency Enhanced ICA method

Considering sound mixtures, sound analysis in time domain cannot show the details of the sound signal or it is not clear to understand the structure of sound mixtures and additive noise. Independent Component Analysis can be used in the time domain to separate the sound mixture into its components. But to increase the performance of Independent Component Analysis, a pre-processing method can be used. A pre-processing method can be, first transforming the sound mixture into the time-frequency domain, second applying the image processing techniques in this domain, third inverse transforming back to the time domain. After pre-processing stage Independent Component Analysis can concentrate on a desired component of sound mixture or can be more selective after pre-processing stage.

Time-frequency transforming methods have two important parameters: precision and security. Precision or accuracy means how details of the signal in time converted into time-frequency domain successfully. This is important to obtain all the signal components in a sound mixture in detail in the resulting image for our research. Security means how successful inverse transforming into time domain again. Here, in our research image must be converted into sound signal with less loss.

For transforming sound into time-frequency domain STFT or spectrogram is used. But, also Wavelet transformation is also used for comparison which is better choice. STFT is a very secure method, but resolution is not so high. Wavelet resolution is better than STFT, but security is not better.

Detecting noisy segments on an image and creating a mask which removes those segments from this image is a very popular image processing filtering application. In our research the same approach can be used either noise or unwanted signal components in a signal mixture. Noise removal also can be used but the main aim is to remove or weaken the weight of some sound components using proper masking. If this masking can be possible in time-frequency domain, we get a new mixture to use ICA method more effectively in time domain.

Here, STFT and Wavelet Analysis applications were searched to understand the filtering and ICA enhancement and the mechanism of sound processing using image processing in MATLAB.

This example generates a sawtooth and sinusoidal test signal corrupted by a noise source. First corrupted signals transformed into time-frequency domain individually using Short Time Frequency Transform (STFT). Then a hard mask is calculated and applied to the Time Frequency image to obtain noiseless image version of the signals. At the last inverse STFT is applied to obtain original signals back without noise. Matlab figures show these steps below. Figure 3.40 shows original signals, noise signal and their Power Spectral Density Estimations.

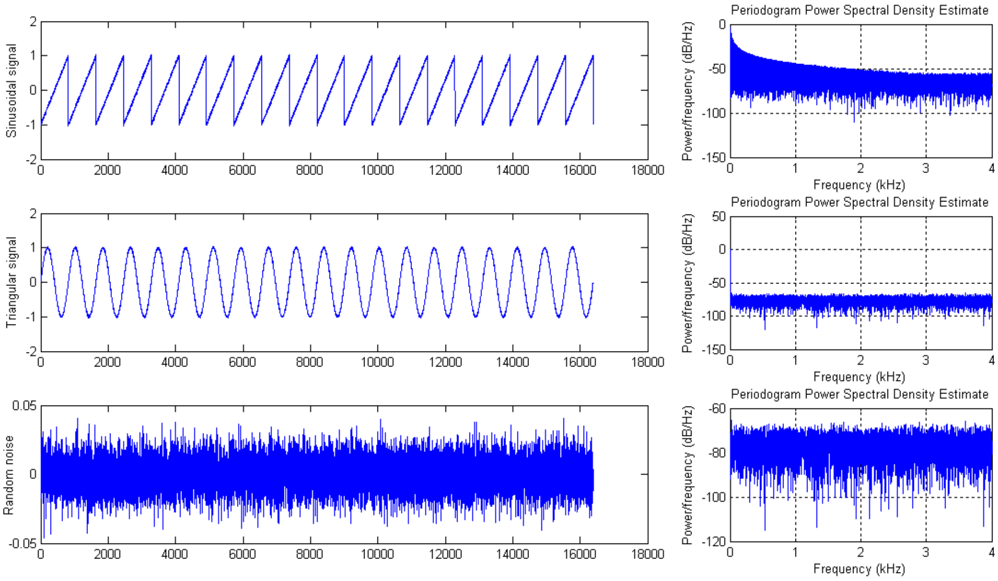


Figure 3.40 Original signals, noise signal and their Power Spectral Density Estimations

Figure 3.41 shows STFT image of corrupted saw tooth and sinusoidal signals on the first two graphics. The following two graphics show calculated hard masks. The last two graphics show the masked versions of STFT images. The Figure 3.42 shows the original signals that contain noise and the recovered signals that have reduced noise.

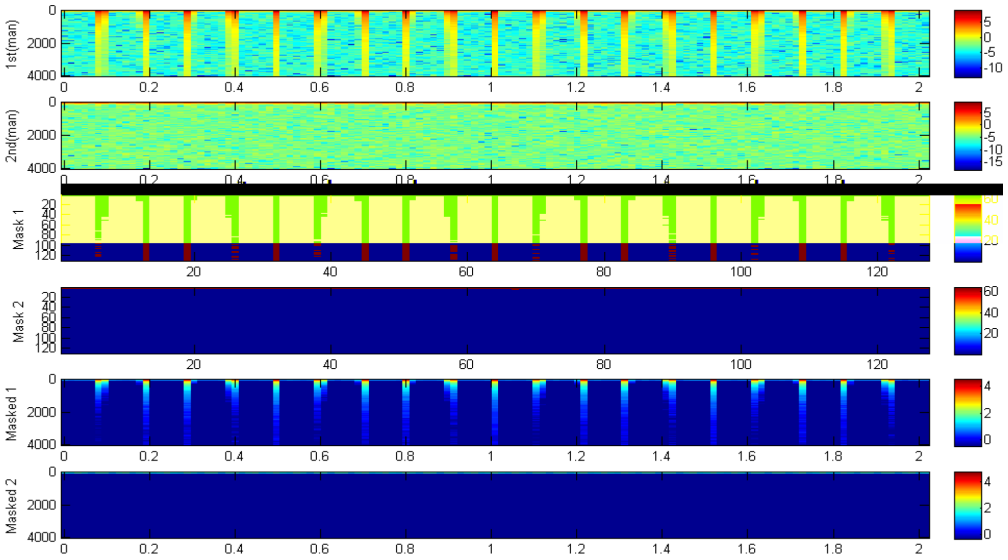


Figure 3.41 Creation of masked STFT images

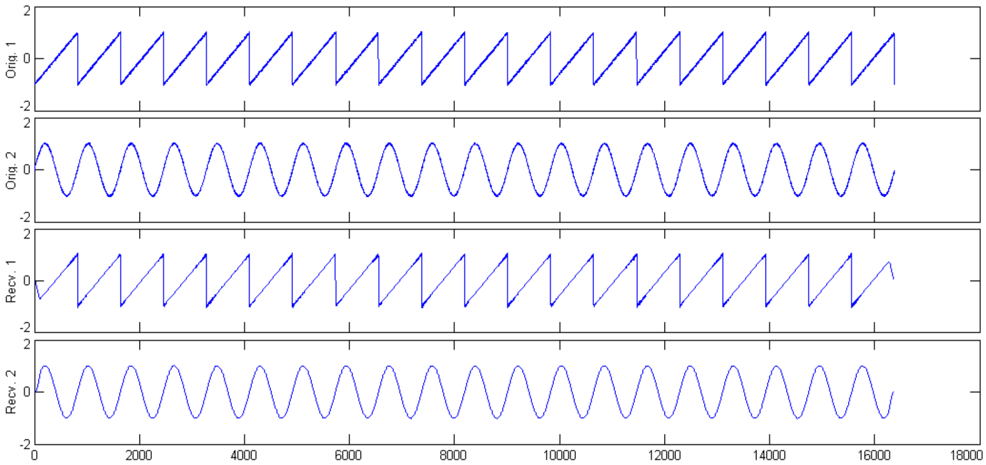


Figure 3.42 Original noisy test signals and the recovered signals with noise reduction.

In MATLAB, a ratio mask is created quantizing a signal into levels using their energy thresholds. Here MATLAB quantizer command is used to create mask using the original signal. Figure 3.43 shows the mask and the signal we get using inverse spectrogram.

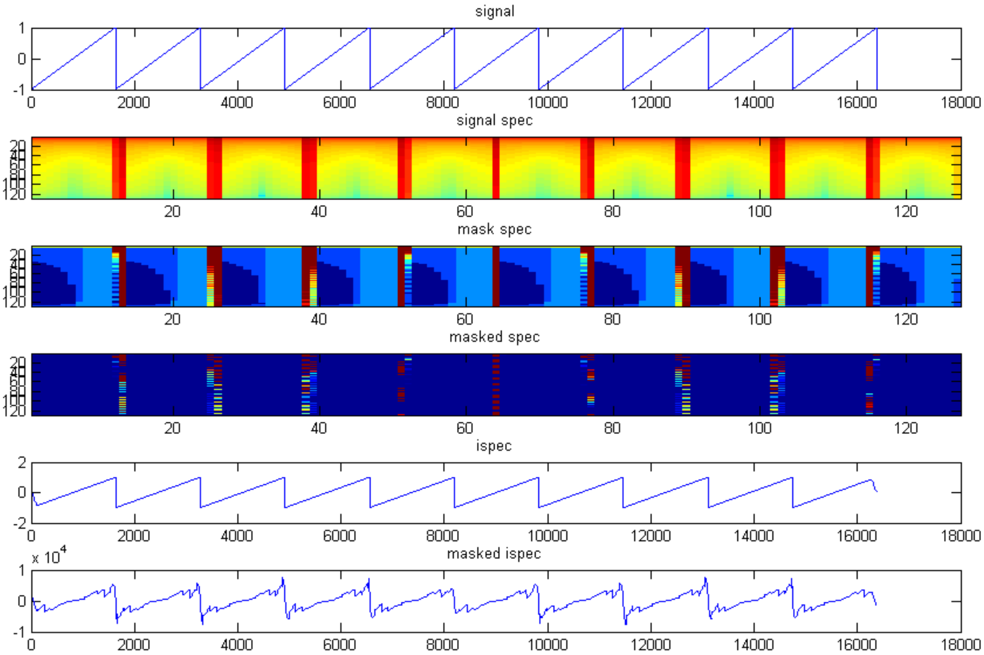


Figure 3.43 Masking of the spectrogram and getting the original source using inverse spectrogram

If we mix the original sawtooth signal with a random noise, still we get the original using the filtering option of the masking. The filtered result is shown in Figure 3.44.

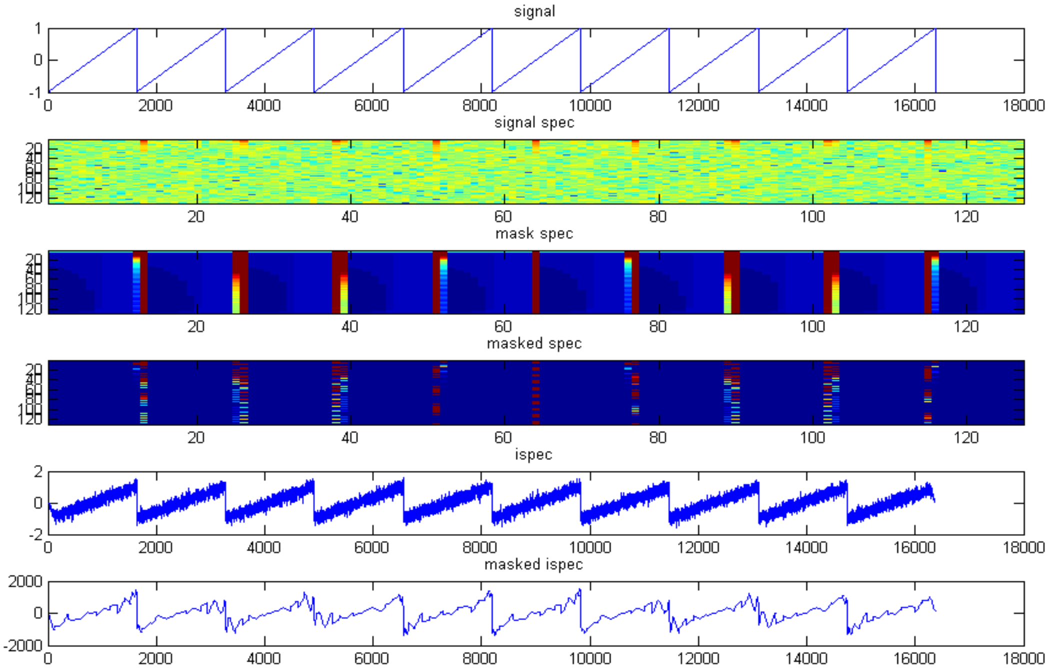
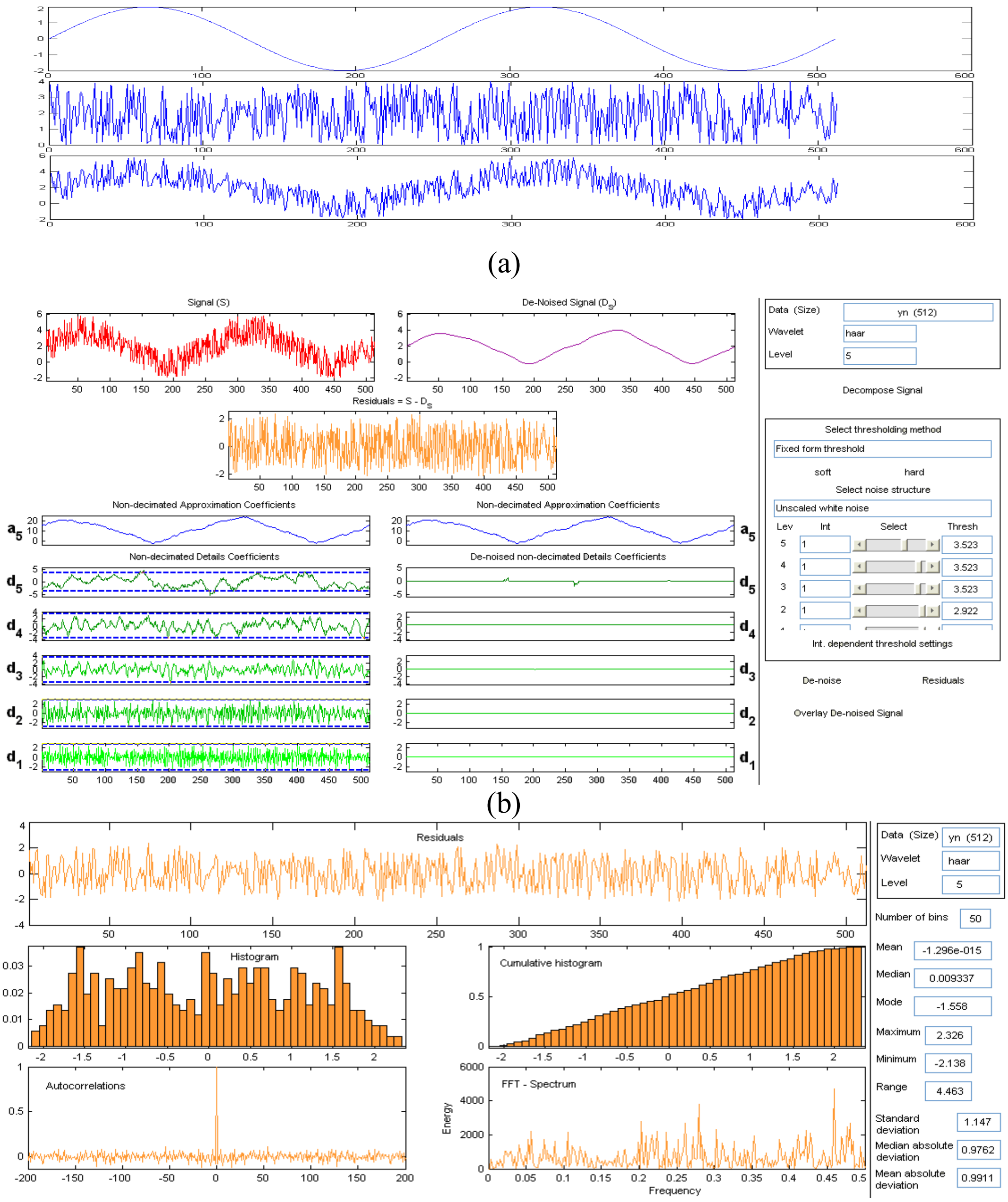


Figure 3.44 Masking the noisy original source using inverse spectrogram

3.2.6 Applications of Wavelet Signal Denoising

Wavelet denoising applications were presented to understand wavelet filtering (Figure 3.45 - 3.51). All the graphics were created using the Wavelet tools of MATLAB.

3.2.6.1 Wavelet sinusoidal signal denoising



(a)

Data (Size): yn (512)
Wavelet: haar
Level: 5

Select thresholding method: Fixed form threshold

soft hard

Select noise structure: Unscaled white noise

Lev	Int	Select	Thresh
5	1		3.523
4	1		3.523
3	1		3.523
2	1		2.922

(b)

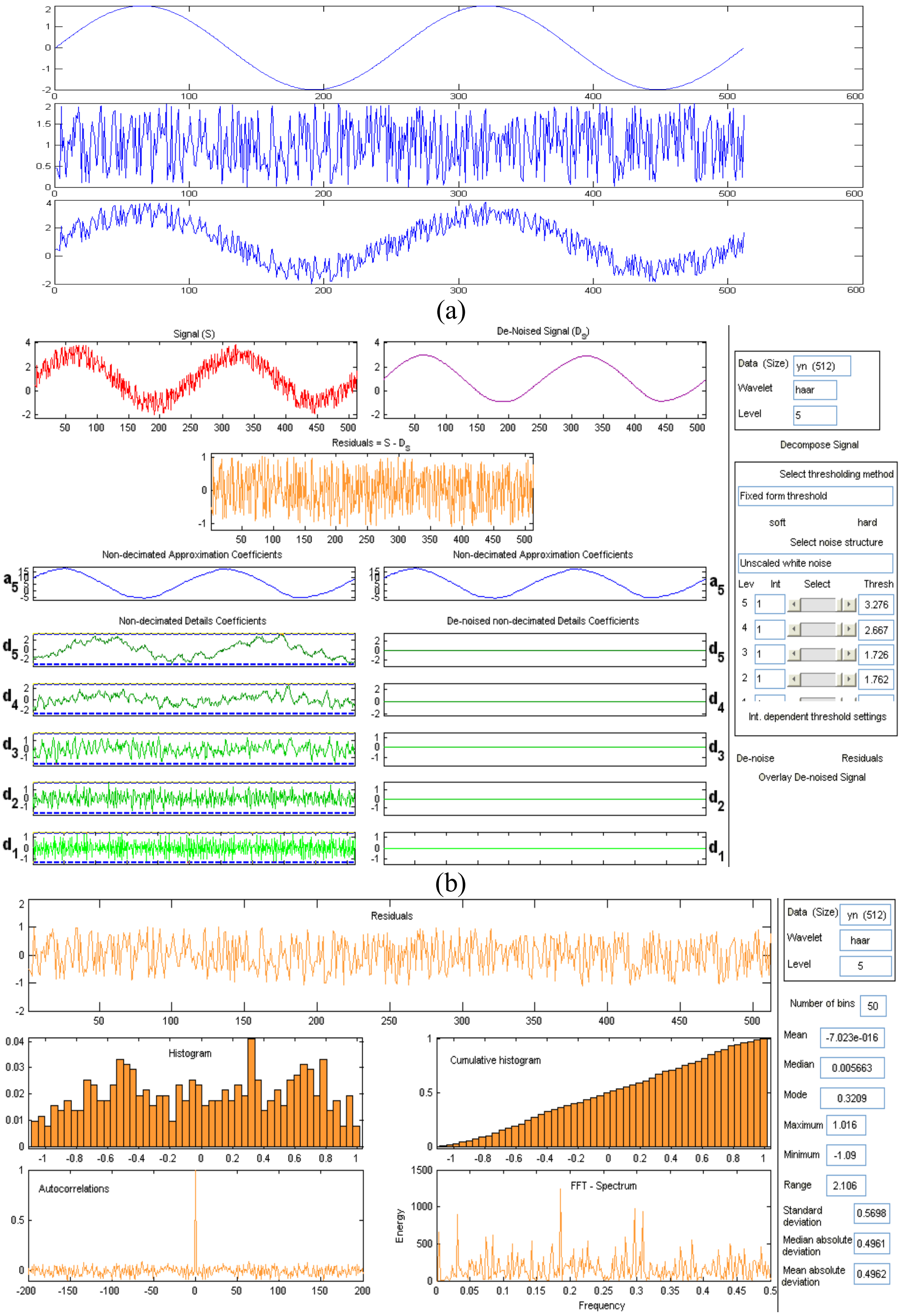
Data (Size): yn (512)
Wavelet: haar
Level: 5
Number of bins: 50
Mean: -1.296e-015
Median: 0.009337
Mode: -1.558
Maximum: 2.326
Minimum: -2.138
Range: 4.463
Standard deviation: 1.147
Median absolute deviation: 0.9762
Mean absolute deviation: 0.9911

(c)

Figure 3.45 a) Signal generation: Sinusoidal signal is corrupted by random noise. b) Denoising with Wavelet Transform toolbox. c) Residual display

3.2.6.2 Sinusoidal signal denoising with low noise level

Here denoising is more efficient compared to application 2.



Data (Size) yn (512)
Wavelet haar
Level 5

Select thresholding method: Fixed form threshold
Select noise structure: Unscaled white noise

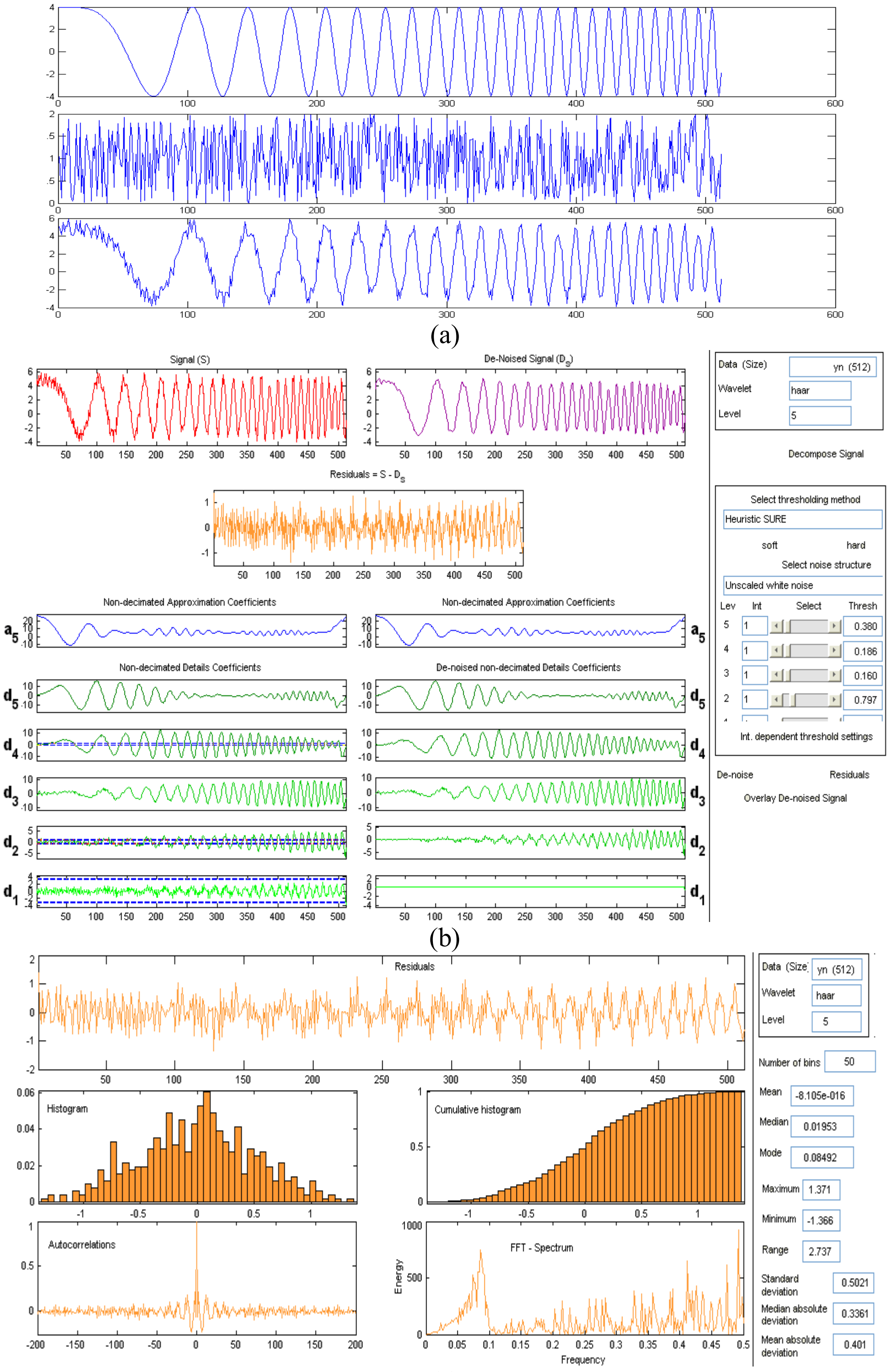
Lev	Int	Thresh
5	1	3.276
4	1	2.667
3	1	1.726
2	1	1.762

Data (Size) yn (512)
Wavelet haar
Level 5

Number of bins	50
Mean	-7.023e-016
Median	0.005663
Mode	0.3209
Maximum	1.016
Minimum	-1.09
Range	2.106
Standard deviation	0.5698
Median absolute deviation	0.4961
Mean absolute deviation	0.4962

Figure 3.46 a) Signal generation: Sinusoidal signal denoising with decreased noise level. b) Denoising with Wavelet Transform toolbox. c) Residual display

3.2.6.3. Denoising of noise corrupted chirp signal



(a)

(b)

Data (Size) yn (512)
Wavelet haar
Level 5

Select thresholding method: Heuristic SURE

soft hard

Select noise structure: Unscaled white noise

Lev	Int	Thresh
5	1	0.380
4	1	0.186
3	1	0.160
2	1	0.797

Int. dependent threshold settings

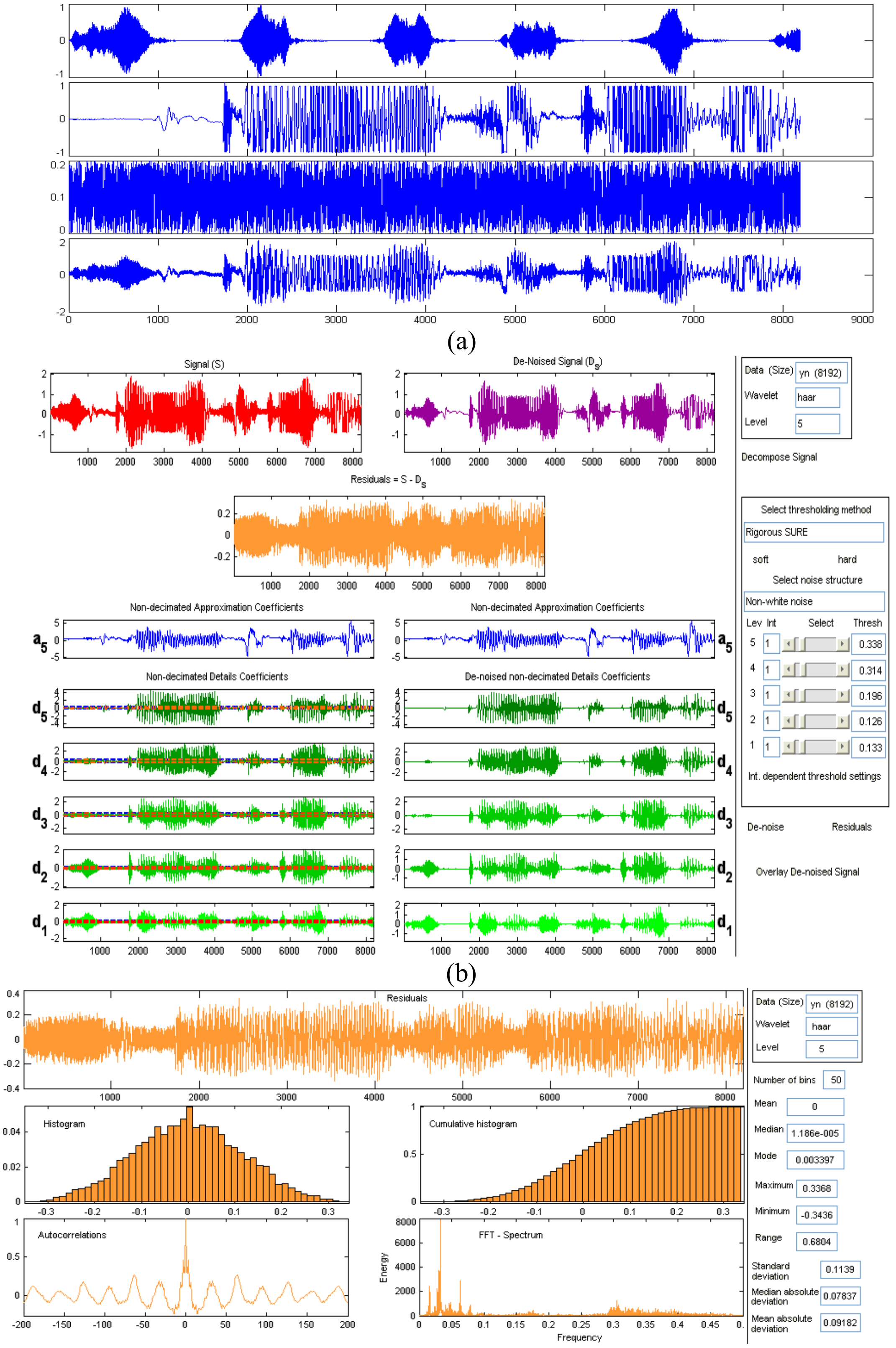
(c)

Data (Size) yn (512)
Wavelet haar
Level 5

Number of bins	50
Mean	-8.105e-016
Median	0.01953
Mode	0.08492
Maximum	1.371
Minimum	-1.366
Range	2.737
Standard deviation	0.5021
Median absolute deviation	0.3361
Mean absolute deviation	0.401

Figure 3.47 a) Signal generation: Chirp signal corrupted with noise b) Denoising with Wavelet Transform toolbox. c) Residual display

3.2.6.4 De-noising of speech, sound and noise mixture application



(a)

Data (Size) yn (8192)
Wavelet haar
Level 5

Select thresholding method: Rigorous SURE
soft hard
Select noise structure: Non-white noise

Lev	Int	Thresh
5	1	0.338
4	1	0.314
3	1	0.196
2	1	0.126
1	1	0.133

Int. dependent threshold settings

(b)

Data (Size) yn (8192)
Wavelet haar
Level 5

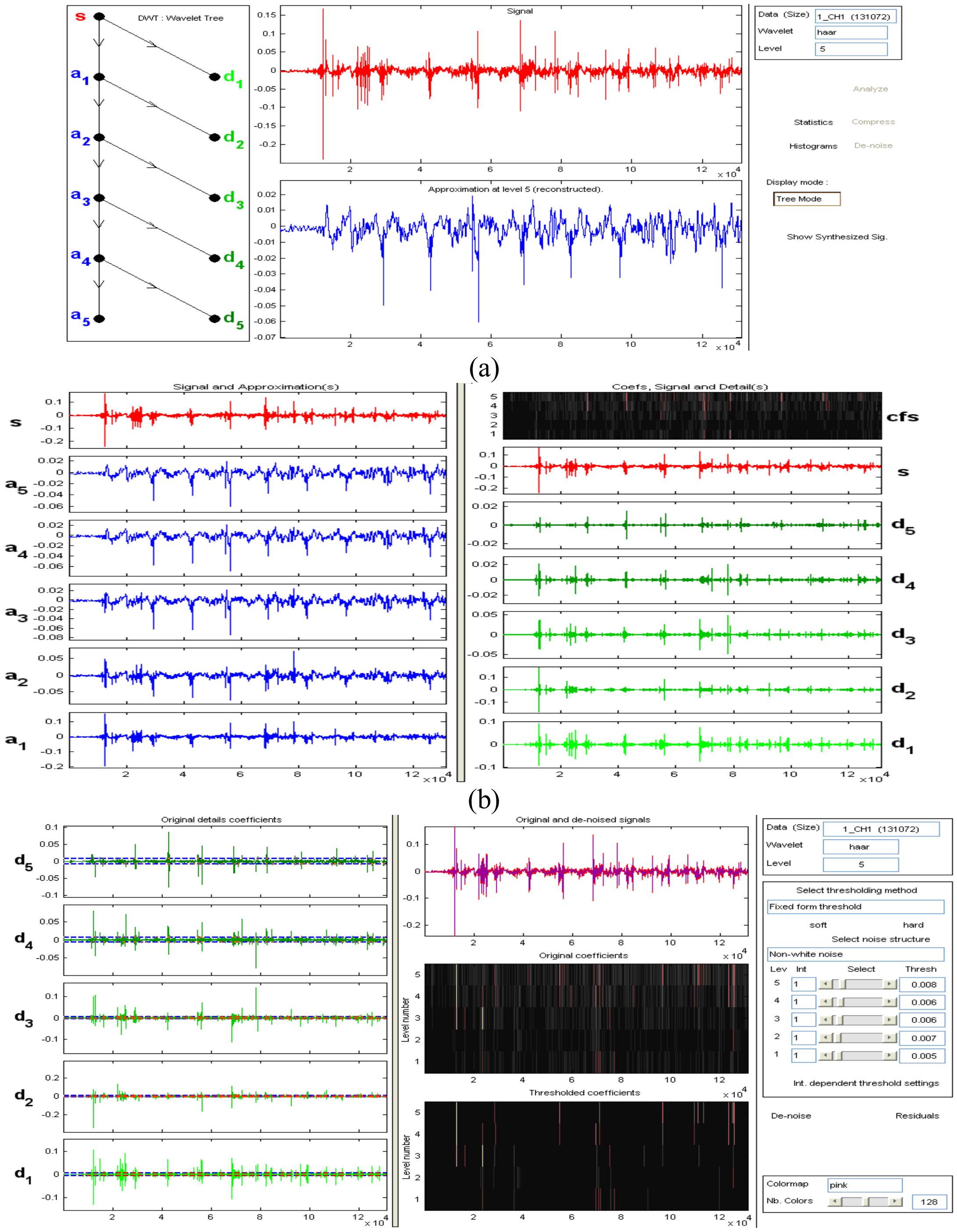
Number of bins	50
Mean	0
Median	1.186e-005
Mode	0.003397
Maximum	0.3368
Minimum	-0.3436
Range	0.6804
Standard deviation	0.1139
Median absolute deviation	0.07837
Mean absolute deviation	0.09182

(c)

Figure 3.48 a) Signal generation: Speech, bird sound and noise mixture b) Denoising with Wavelet Transform toolbox. c) Residual display

3.2.6.5 Signal decomposition of knee signal

Level 5 Haar Wavelet was used. Figure 3.46 and 3.47 show the decomposition steps. Here knee signal was logged using sound data logging system with stethoscope heads attached.


Data (Size) 1_CH1 (131072)
Wavelet haar
Level 5
Display mode : Tree Mode

(a)

(b)

Data (Size) 1_CH1 (131072)
Wavelet haar
Level 5

Select thresholding method: Fixed form threshold
soft hard
Select noise structure: Non-white noise

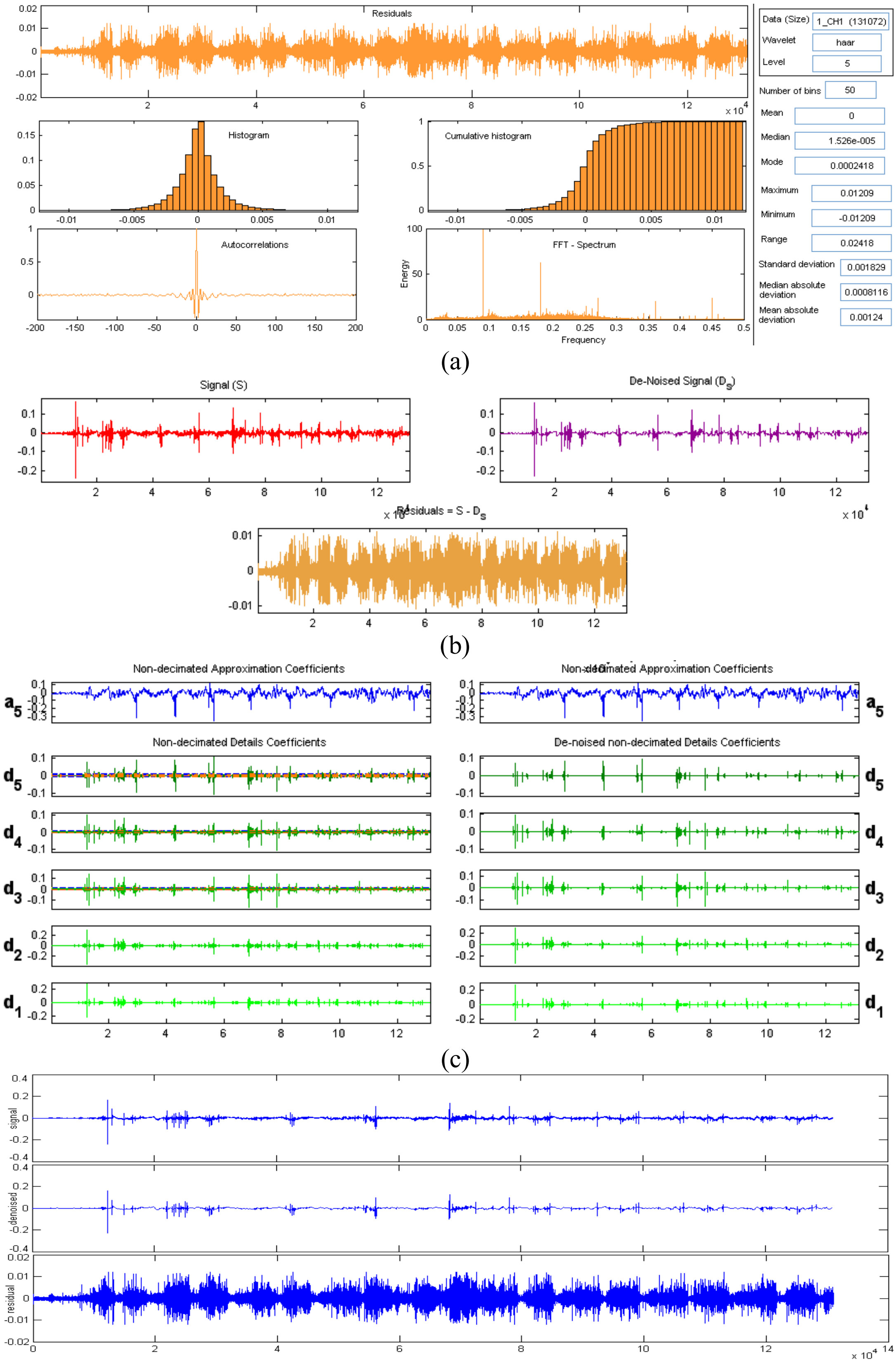
Lev	Int	Thresh
5	1	0.008
4	1	0.006
3	1	0.006
2	1	0.007
1	1	0.005

Colormap pink
Nb. Colors 128

Original details coefficients
Original and de-noised signals
Original coefficients
Thresholded coefficients
Level number
Select
Int. dependent threshold settings
De-noise
Residuals
x 10^4

(c)

Figure 3.49 a) Knee Signal decomposition tree mode display b) Separate mode display. Decomposition at level 5: s=a5+d5+d4+d3+d2+d1 c) Denoising with Wavelet Transform toolbox.



Data (Size) 1_CH1 (131072)
Wavelet haar
Level 5
Number of bins 50
Mean 0
Median 1.526e-005
Mode 0.0002418
Maximum 0.01209
Minimum -0.01209
Range 0.02418
Standard deviation 0.001829
Median absolute deviation 0.0008116
Mean absolute deviation 0.00124

(a)

(b)

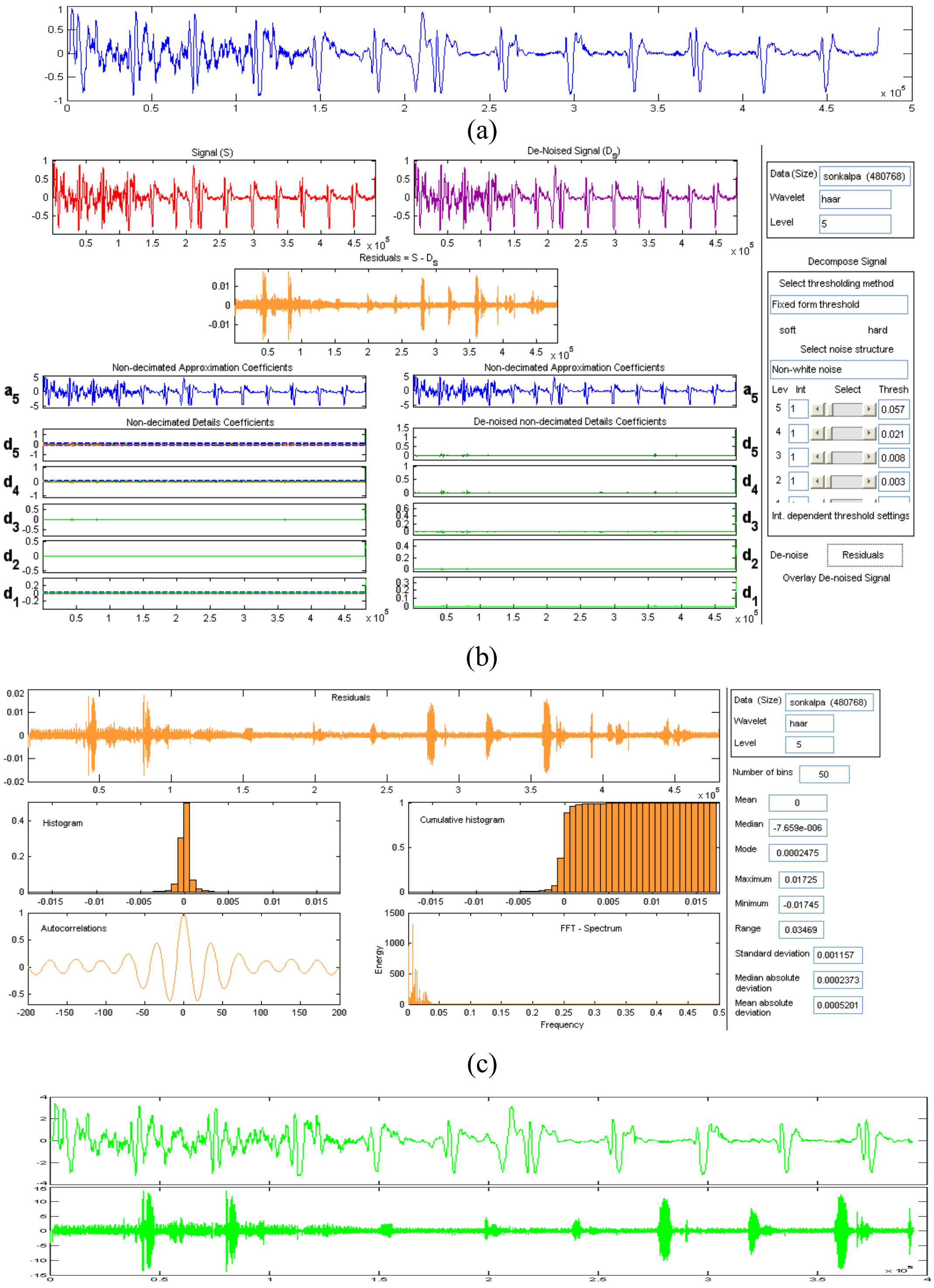
(c)

(d)

Figure 3.50 a) Residual display of the signal after denoising. b) Signal and de-noised signal and their coefficients. c) Using wavelet denoising, de-noised and residual version of knee signal. d) Using wavelet denoising, de-noised and residual version of knee signal.

3.2.6.6 Heart sound

Heart Sound data (sonkalpa.wav) recorded by sound data logging system.



(a)

Data (Size) sonkalpa (480768)
Wavelet haar
Level 5

Decompose Signal

Select thresholding method
Fixed form threshold
soft hard
Select noise structure
Non-white noise

Lev Int Select Thresh
5 1 0.057
4 1 0.021
3 1 0.008
2 1 0.003

Int. dependent threshold settings

De-noise Residuals
Overlay De-noised Signal

(b)

Data (Size) sonkalpa (480768)
Wavelet haar
Level 5
Number of bins 50
Mean 0
Median -7.659e-006
Mode 0.0002475
Maximum 0.01725
Minimum -0.01745
Range 0.03469
Standard deviation 0.001157
Median absolute deviation 0.0002373
Mean absolute deviation 0.0005201

(c)

(d)

Figure 3.51 a) Heart Sound data. b) De-noising. c) Residual. d) ICA components (icayap.m)

Sound Analysis in time domain cannot show the details of the sound signal or it is not clear to understand the structure of sound mixtures and additive noise.

In future, ICA will be applied to a mixture without proposed pre-processing method. Also, another ICA results will be obtained to a mixture after pre-processing stage. These two results will be compared to show the success of proposed method. Also this research will show the comparison of the time-frequency methods. Mask estimation and missing components reconstruction will be also analyzed.

CHAPTER FOUR
DATA LOGGING SYSTEMS

During the current thesis work, two data logging hardware have been designed and developed for sound and biomedical applications especially designed to use independent component analysis. For sound applications a multi-channel USB sound card has been designed and used, for kinematic signal processing such as knee and jaw movement detection an accelerometer data logging system. The following sections describe the hardware and the software of these systems.

4.1 Multi Channel USB Sound Data Logging System

The sound processing applications usually require powerful and easily controlled hardware. Generally channel switched analog – digital converters are used on that hardware and they don't achieve parallel data recordings. But, multi channel simultaneous recording is needed for some sound processing applications. These applications are sound separation into components, speech filtering from background noise or music, speech enhancement, etc. Here, Independent Component Analysis (ICA) method is a very important pre-processing stage for sound signal processing applications.

A low cost software controlled multi channel sound capturing device was designed and was operated with the aid of user-friendly PC software. It has universal serial bus (USB) interface because of it's widely use. Therefore it can be used in both desktop and notebook computers. System consists of parallel USB sound cards for every recording channel. For example, in our application four recording channels are built and their inputs are gathered in a standard connector to connect microphones from one cable. The hardware is not so complicated, but the software has to start and stop sound cards synchronously, and record the sounds simultaneously which is required ICA applications.

Here, the system was introduced and some ICA applications were done using the developed sound data logging system.

4.1.1 Sound Data Logging System Hardware

Initially, a three channel capture device was built using one microphone input of a sound card with the aid of the analog switch mechanism. Channel selection was made by using the serial port of the PC. The major drawback of this device is the non-simultaneous recording. For this reason ICA analysis could not be applied properly. Also, because of the need for the RS232 serial channel control, notebooks PCs are not suitable. Notebook PCs don't have serial ports.

For the new version of pre-developed hardware four channels simultaneous sound capturing device was designed using most popular USB 2.0 interface. This system and four channel microphone connection was shown in the Figure 4.1. Here, four USB sound cards were connected in a parallel manner without a switching mechanism over a multiplexer and time divison for one sound card. After USB connection to a PC, every sound card gets the same sequentially following device id everytime. Thus, the sound recording channels are exactly at the same position.

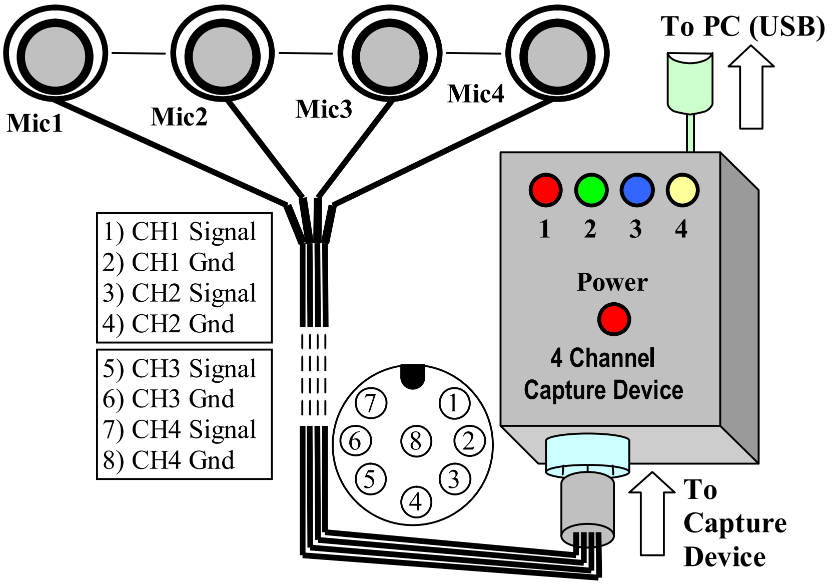


Figure 4.1 Four Channels Sound Capture Device.

The block schematic of the sound data logging system is in Figure 4.2. The USB expander is required many USB sound cards connection from one USB port to a PC or mobile computer. Figure 4.3 shows the photos of the system. Figure 4.4, shows photo of the microphone apparatus.

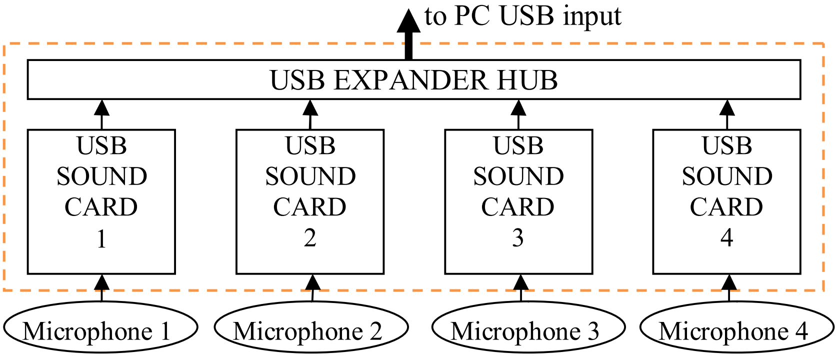


Figure 4.2 Sound data logging system block schematic.

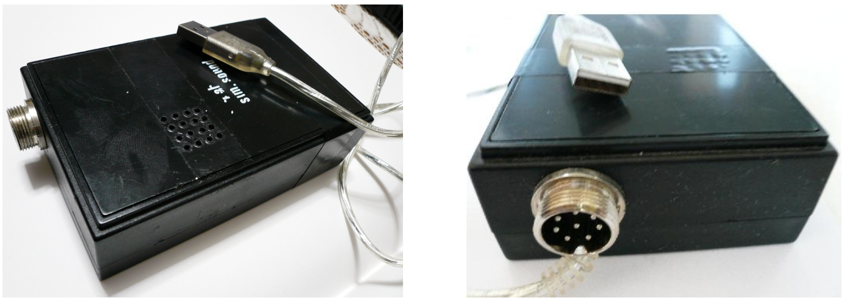


Figure 4.3 Picture of the Sound Data Logging System developed for the current thesis work.

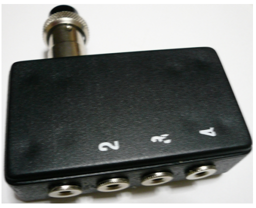


Figure 4.4 Photo of Four Channel microphone apparatus.

A stethoscope head and microphone combination was used instead of capacitive microphones only. The drum on the stethoscope head gets the vibration from the knee cap or lateral positions. Vibration is converted to sound through the air channel and at the end of the tube a microphone is placed to record a sound signal. Figure 4.5 shows one channel stethoscope apparatus.

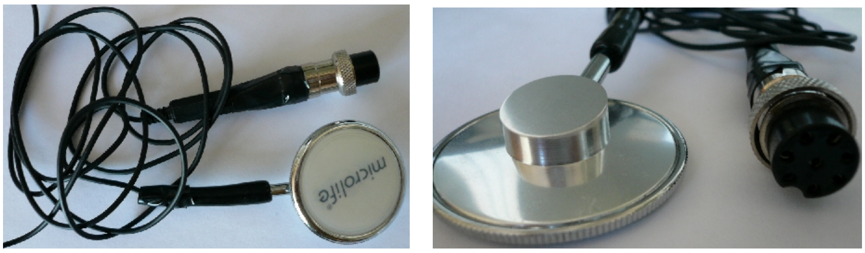


Figure 4.5 One channel stethoscope apparatus.

4.1.2 Sound Recording Software

As seen before the hardware is very basic, but software especially has to run the sound cards in parallel. For sound capturing, recording software was designed and written in Borland C++ Builder. The main screen of the recording software is in Figure 4.6. Four channel sounds are captured to the system RAM for guaranteeing synchronizing speed. After capturing has finished, RAM buffers are transferred to the hard disk. Software detects every sound card USB identities every time at the right order. Recording sound files first being saved in a sufficient RAM buffer area, after the recording operation completed transferred from RAM buffers to the harddisk automatically. The purpose of using RAM buffering is to guarantee the speed of multiple channel synchronous recording. Any sound analysis like ICA is not operated in this program. The analysis software is in MATLAB.

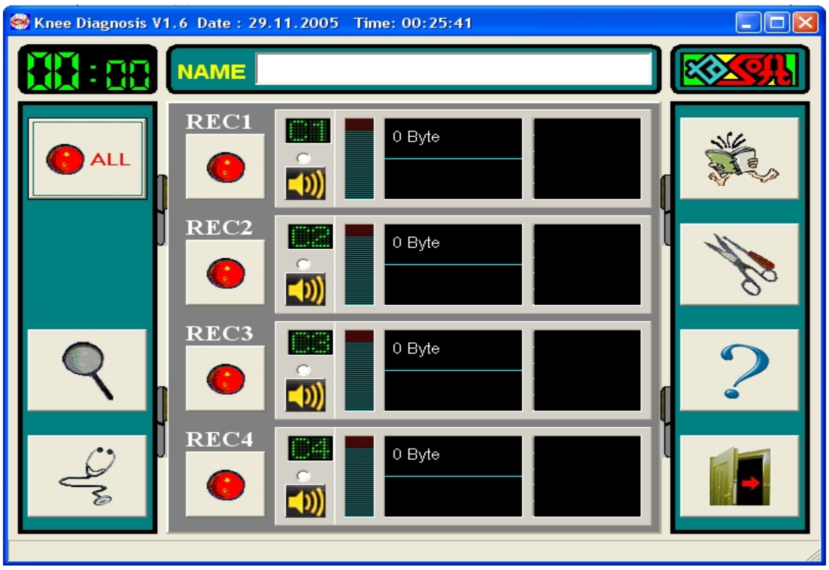
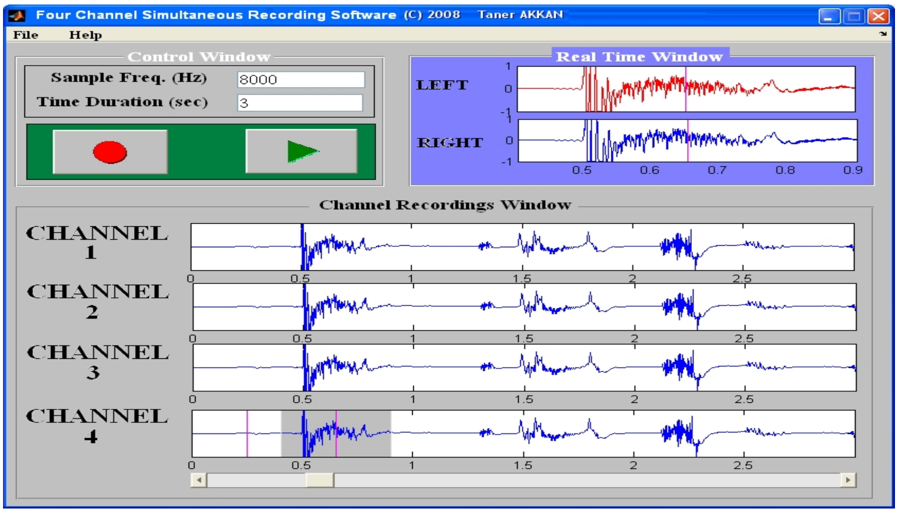


Figure 4.6 The main screen of the recording software.

Recording software is in C and the analysis software in MATLAB, but they can be operated in MATLAB software. Matlab Data Capture Library is used in Graphical User Interface (GUI) code. Four analog objects simultaneous triggering function of the library was used to achieve simultaneous data capturing.

The main screen of the recording software in MATLAB is shown in Figure 4.7. Both C and MATLAB code successfully operates sound data logging hardware. Here, the input of the sound cards also can be other signals instead of sounds to record. For example, knee joint movement sounds can be recorded and analyzed with the aid of stethescope apparatus. Also, the heart pulses and breathing can be recorded using multi channel stethescope microphones.



Sample Freq. (Hz) 8000
Time Duration (sec) 3

Figure 4.7 Main screen of MATLAB recording software

4.1.3 Sound Data Recording System Tests

A recording test with three microphones in a room with MATLAB software was done with speaking beside microphone 1. All the microphones are in a line distant 1 cm each others. Channel four is empty. The system guarantees the simultaneous recording criteria. Recording results for three channels are shown in Figure 4.8.

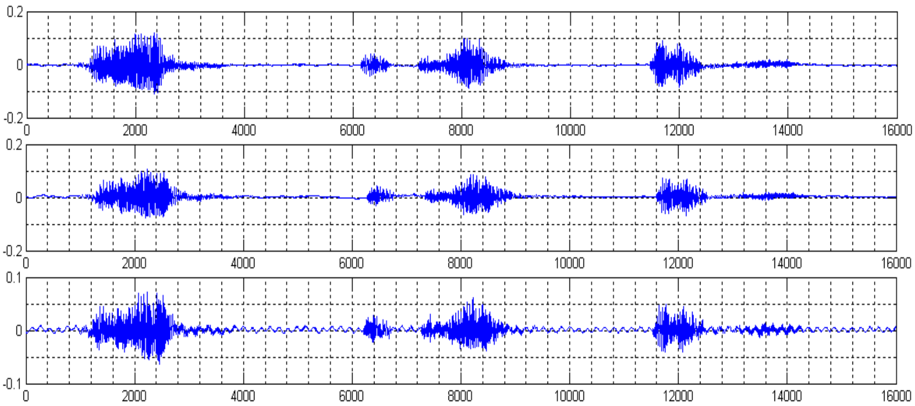


Figure 4.8 Three channels simultaneous sound recording result. (1 cm distance)

When the distance between microphones increased to 20 cm and spoken beside the microphone 1, the three simultaneous recording results are in Figure 4.9. Amplitude and noise difference between microphone 1 and 3 can be seen, and also signal delay difference was existed. The delay difference was not sourced from the sound data logging system; it was the result of the experiment.

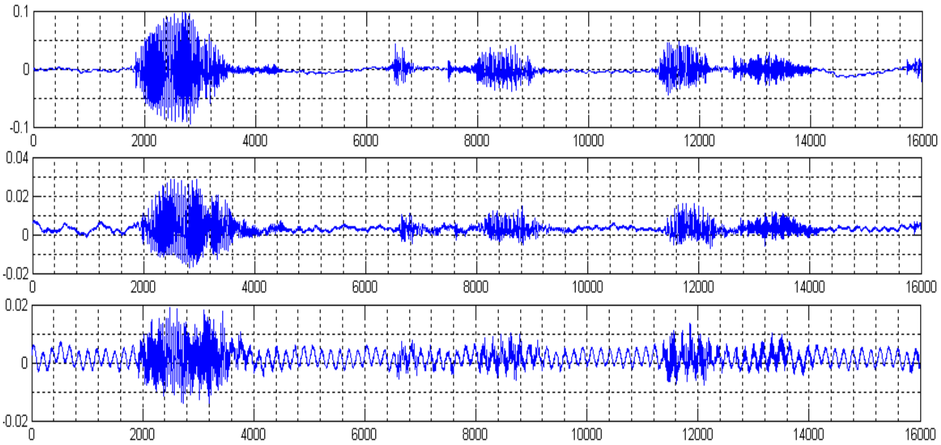


Figure 4.9 Three channels simultaneous sound recording result (20cm distance).

Placing four microphones separated with equal distances from each others on a straight line causes phase differences in the recorded data at different recording channels. Because of the distance differences of microphones to the sources, we have phase delays on the recorded signals. Four test signals such as sinusoidal wave, square wave, sawtooth wave, and chirp sound was created to test ICA performance.

In the Figure 4.10, a sample microphone placement is shown.

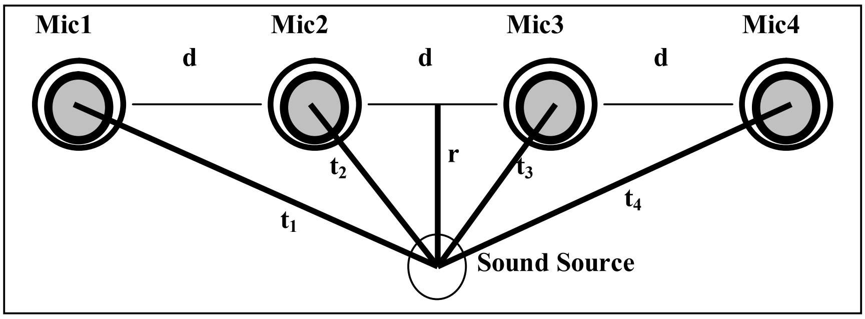


Figure 4.10 Microphone placement and the distances from the sound source.

Several tests have been realized using MATLAB software to understand how ICA algorithms perform and which are the best algorithms for knee problem.

4.1.3.1 Sound Data Recording System Test 1

Original signals (Fig. 4.11), mixtures (Fig 4.12), some ICA methods results (Fig 4.13 to Fig. 4.16) with no phase shift (using dortkanal.m).

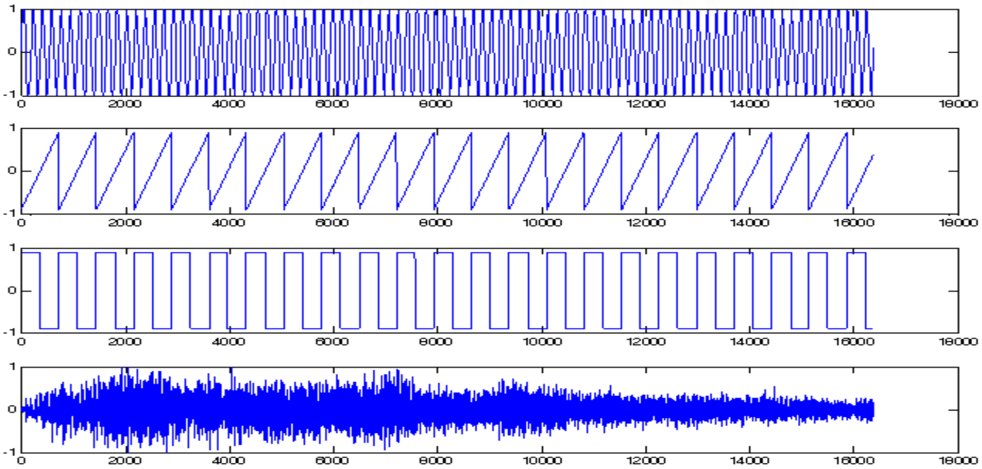


Figure 4.11 Original Sources

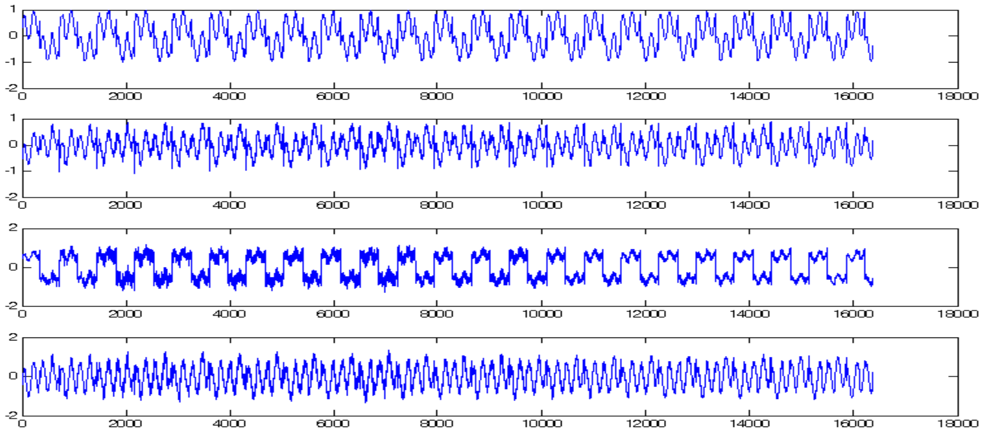


Figure 4.12 Mixed Signals

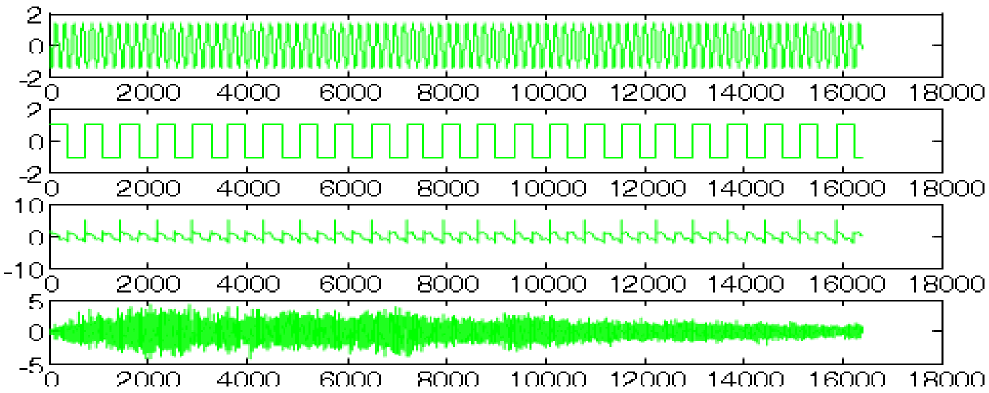


Figure 4.13 Fast ICA Algorithm Results

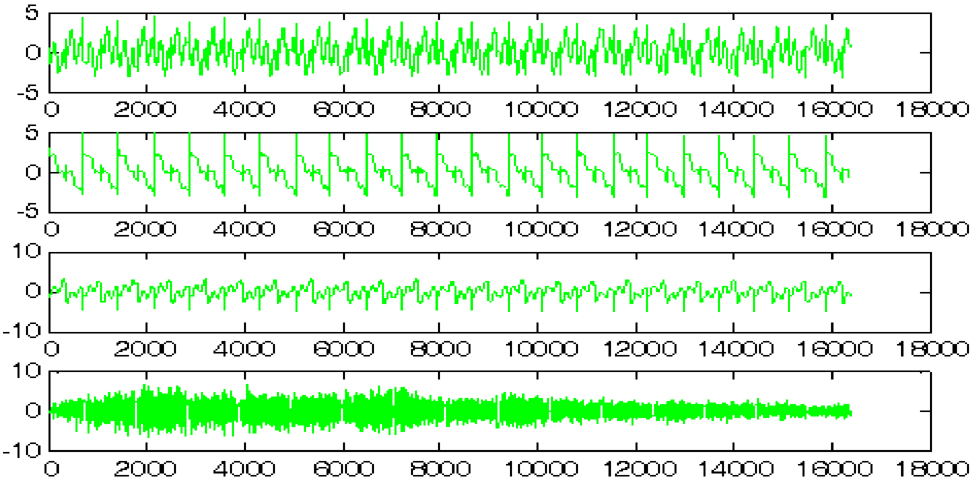


Figure 4.14 ML ICA Algorithm Results

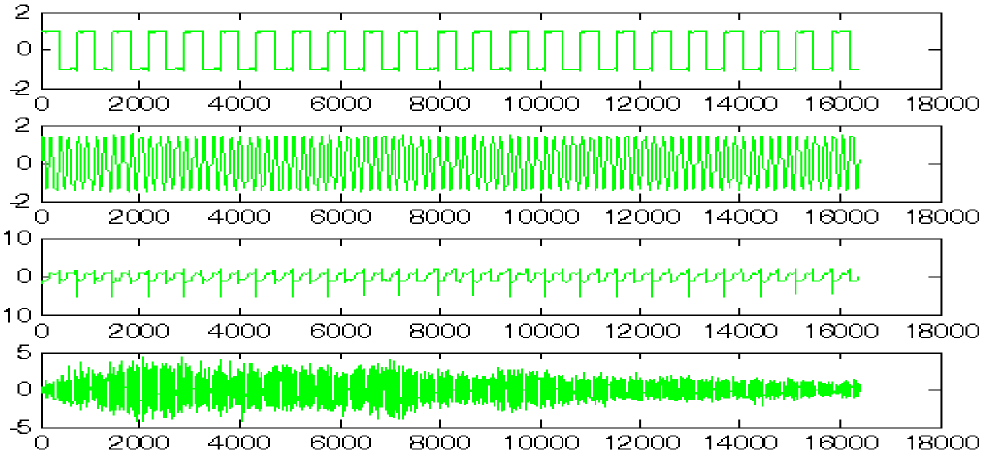


Figure 4.15 Jade Algorithm Results

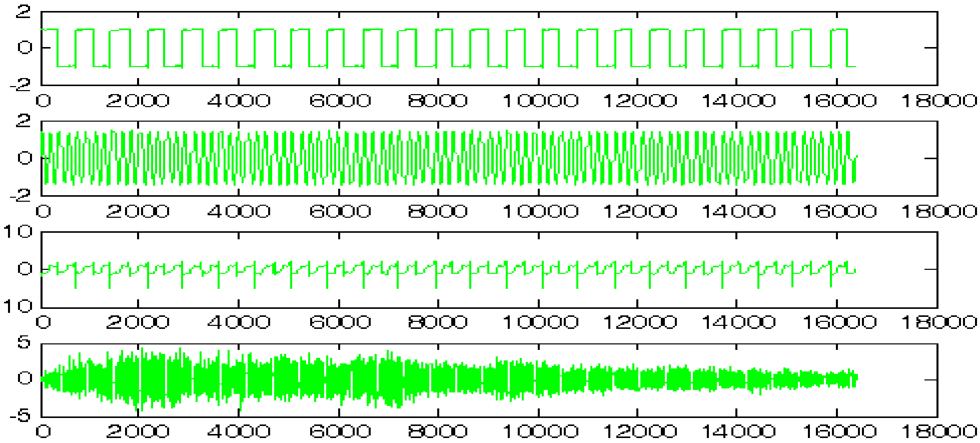


Figure 4.16 Shibbs Algorithm Results

Here as a result, Fast ICA, Jader and Shibbs algorithms perform better, but ML ICA fails. Only the band-limited chirp sound separated successfully.

4.1.3.2 Sound Data Recording System Test 2

To prove, there are phase delays between microphones, on a glass surface four microphones recorded the test signal which is emitted from a speaker. (Figure 4.17) Test signal here is a sinusoidal wave which has the frequency of 1 kHz.

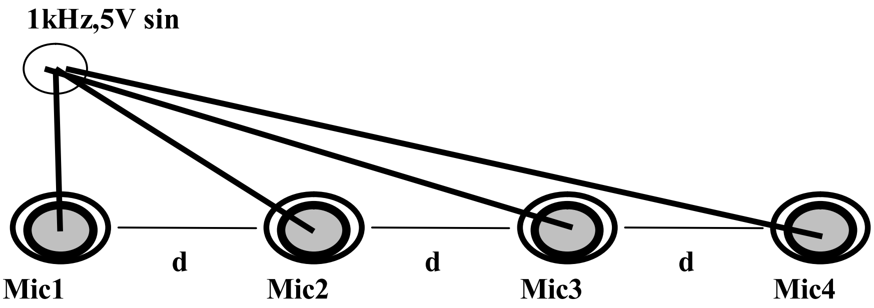


Figure 4.17 Phase Delay Test System.

Below, the Figure 4.18 shows the phase delays between the recordings in general, and Figure 4.19 shows the same delays in detail.

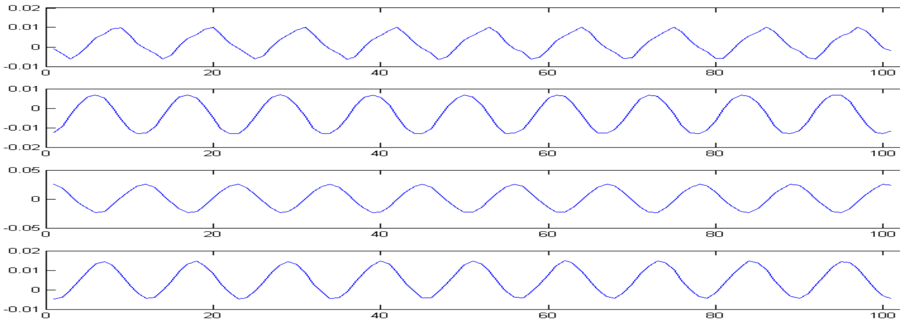


Figure 4.18 Phase Delays between channels (general view)

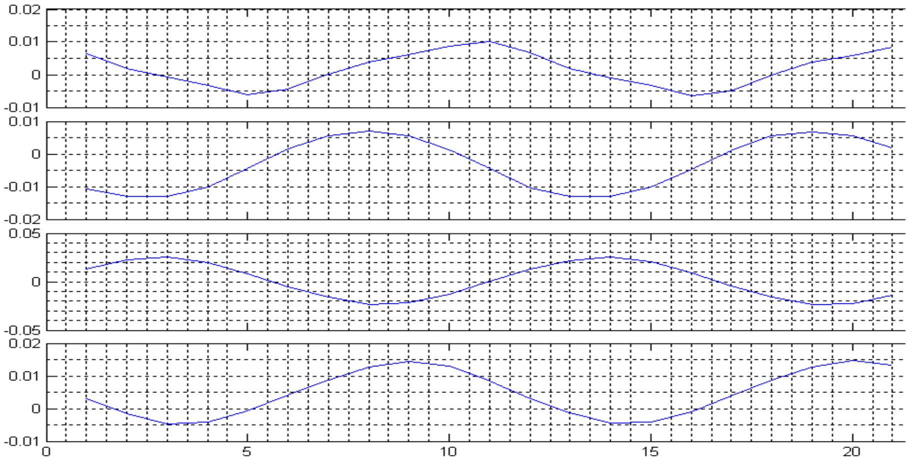


Figure 4.19 Phase Delays between channels (detailed view)

4.1.3.3 Sound Data Recording System Test 3

A real-time ICA test system created to separate three sounds on a glass surface (Figure 4.20). The sinusoidal signal amplitude is 5V and frequency is 1 kHz. Gong and Child Laughter sound waveforms are shown below in Figure 4.21.

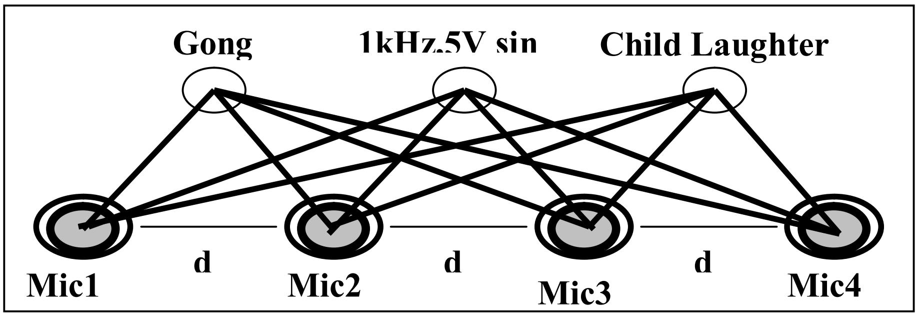


Figure 4.20 Real Time ICA Test System.

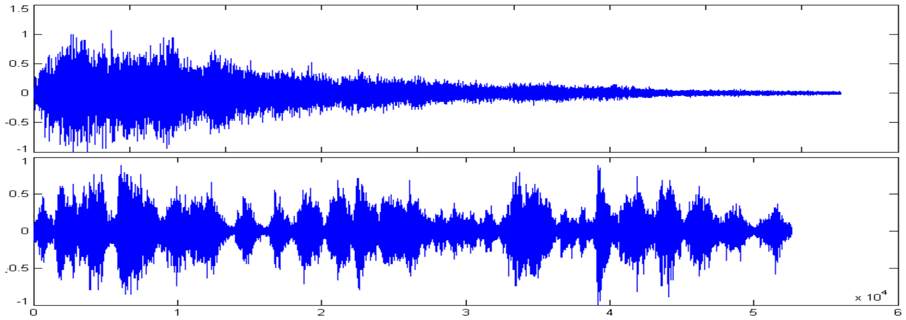


Figure 4.21 Gong and Laughter Sound.

Mixtures recorded by the microphones are shown in Figure 4.22.

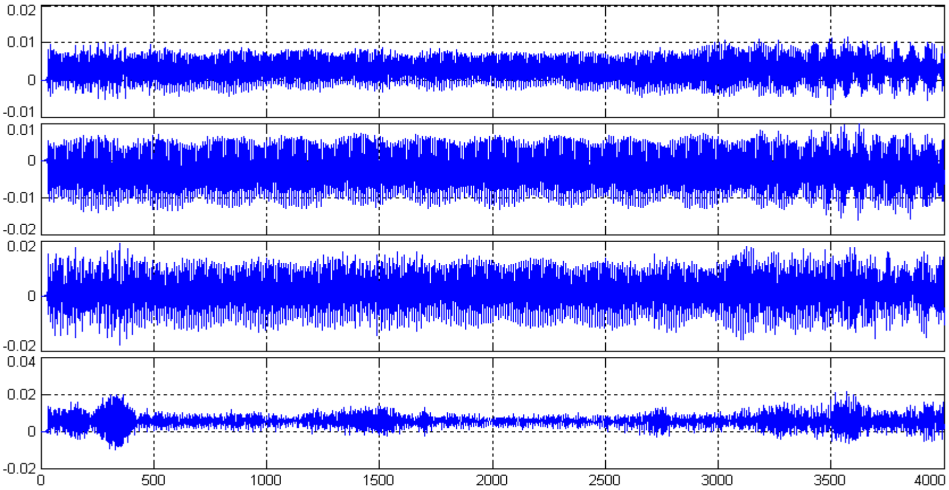


Figure 4.22 Mixtures

Fast ICA, ML ICA, Jader algorithm results are shown in the following figures (Figure 4.23 to Figure 4.25).

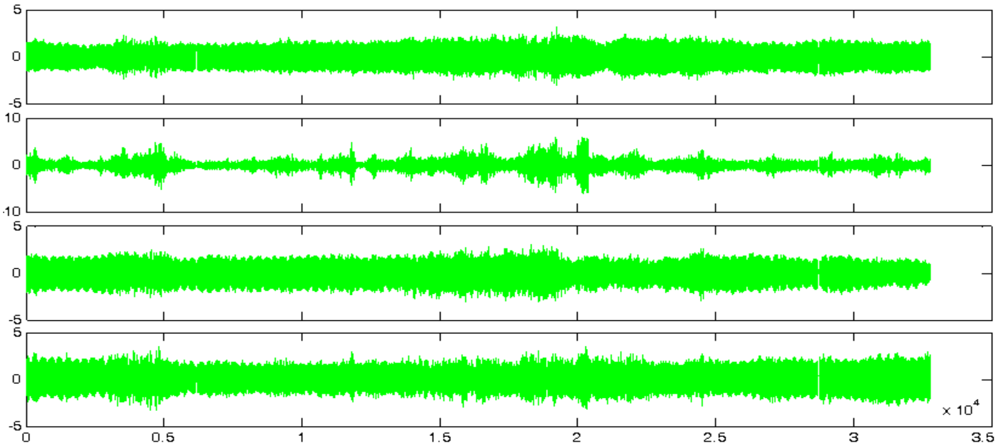


Figure 4.23 Fast ICA Results

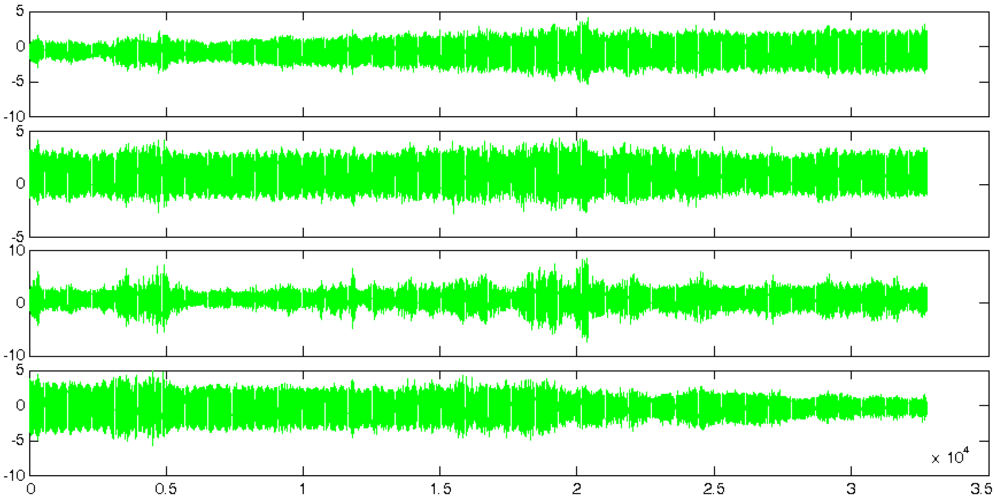


Figure 4.24 ML ICA Algorithm Results

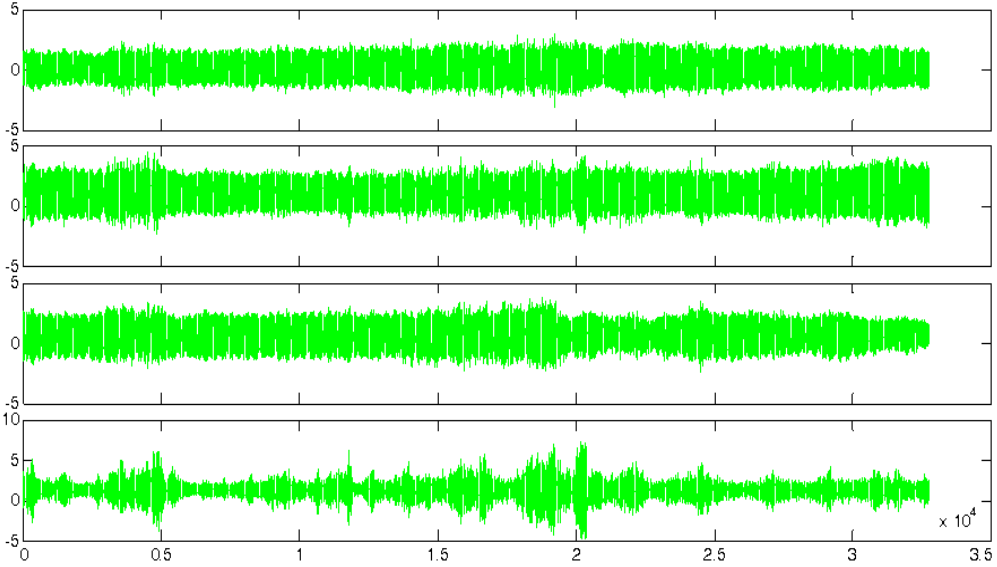


Figure 4.25 Jade Algorithm Results

4.2 Multi Point Multi Dimensional Accelerometer Data Logging System

For knee and EEG related studies a Multi Point Multi Dimensional (MPMD) Accelerometer Data Logging System (ADLS) has been built on a PIC microcontroller.

4.2.1 Accelerometer Data Logging System hardware

The MPMD ADLS hardware consists of two main parts; i) accelerometers, and ii) a controller board. Three accelerometers are placed on an elastic band to collect data from knee cap and two lateral positions. The same elastic band fits also to the jaw locations. The microprocessor based controller unit is located on a printed circuit board and mounted on a plastic case. On the controller board there is an LCD display and a LED to illustrate the currently collected data and working status of the controller, respectively. USB cable from PC is connected to the USB port of the system for data transfer and power supply. The designed system photo is given in Figure 4.26.

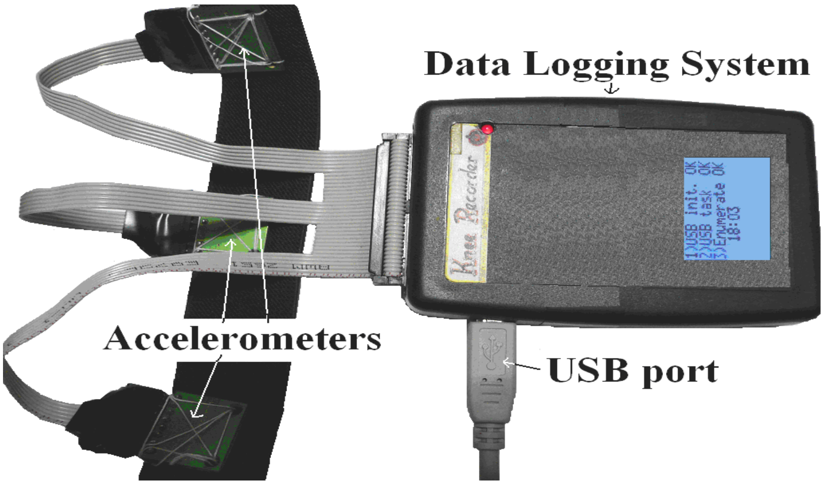


Figure 4.26 Picture of Accelerometer Data Logging System developed for the current thesis work.

The designed MPMD ADLS system block scheme is shown in Figure 4.27. The heart of the MPMD ADLS is based on a USB PIC microcontroller (Microchip, 2009). Microcontroller is needed to collect three accelerometers data simultaneously and store or send them to a desktop or mobile personal computer. Also, the ADLS system circuit board has 3.3V voltage regulator, 3.3V-5V level translators, current buffers, real time clock circuit, and SD card interface circuit. A microcontroller based real-time operated system has been developed to achieve high speed USB data transfer to a mobile computer. This software has been developed in C programming platform. USB 2.0 guarantees enough data bandwidth for real time operation and mobile operability without using a RS232 serial port.

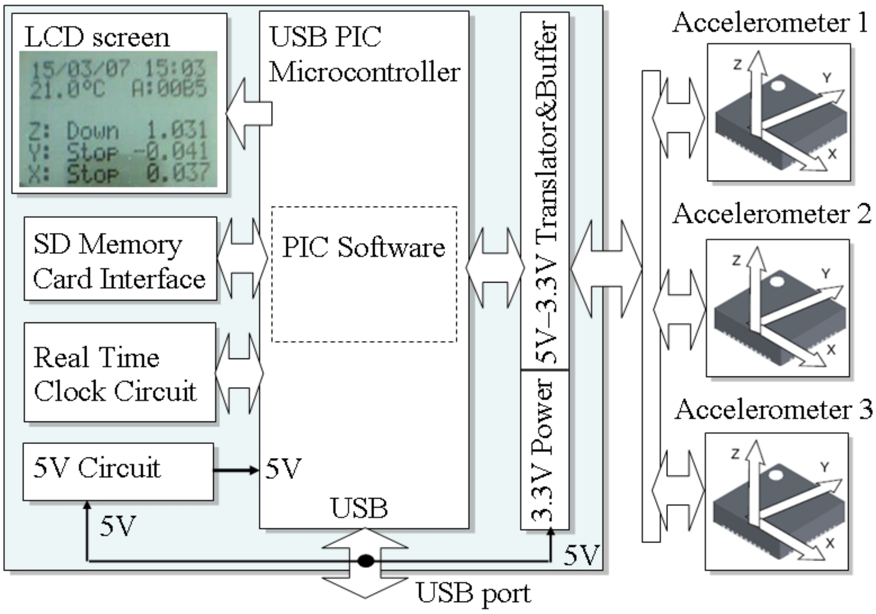


Figure 4.27 Accelerometer Data Logging System Schematic

Mobile computer software has been developed on MATLAB platform to achieve robust data logging and analyzing on the same program code. MATLAB software communicates with ADLS via USB and shows the collected data on the screen and stores them to the RAM memory or hard disk at real time. Moreover, data analysis with the developed software can be processed at real time or any time after data collection has finished.

In the developed hardware system, three axes accelerometer module LIS3LV02DQ (ST Microelectronics, 2009) was used to collect accelerometer data. This accelerometer simultaneously converts three axes acceleration data and it has a user selectable full scale conversion of ±2g, ±6g. It has the capability of measuring acceleration over a bandwidth of 640 Hz for all axes. The communication protocol can be I^{2}C or SPI for the microcontroller interface. The possible applications are: free fall detection, motion activated functions, inertial navigation, virtual reality input devices and vibration monitoring and compensation. Accelerometer module and the development prototype board is shown in Figure 4.28.

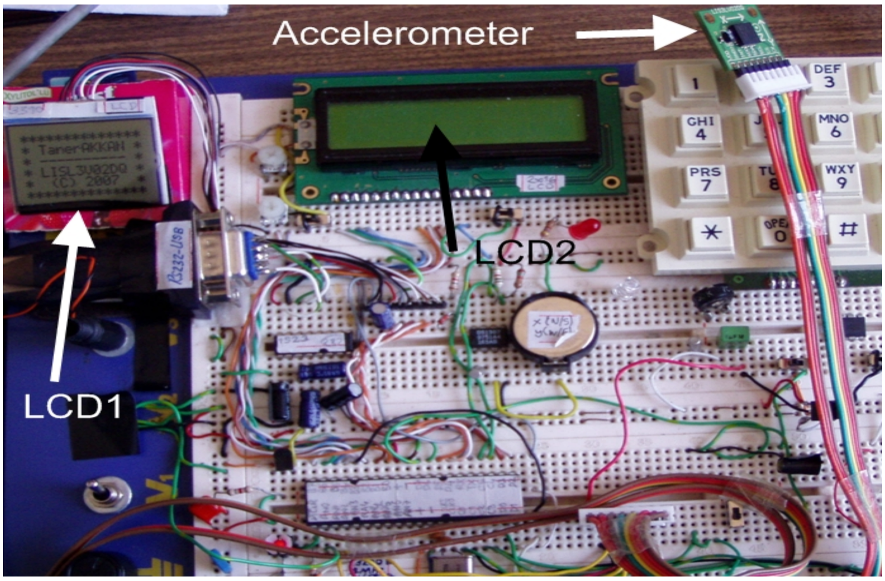


Figure 4.28 Development prototype board and the accelerometer module

The currently developed MPMD ADLS supports up to four accelerometers simultaneously. However the number of supported sensors can be easily increased by upgrading the buffer hardware. PIC microcontroller collects data from accelerometers and transfers them to a mobile computer with USB 2.0 interface. Three axes data from each three accelerometers means nine channels of data need to be acquired. This size of data collection is better achieved by USB interface rather than available RS-232. The LCD screen shows the accelerometer results, processing steps, and time stamps. The collected accelerometer data were transferred to the PIC microcontroller memory via serial peripheral interface (SPI) bus.

4.2.2 Mobile Computer and ADLS software

The software of the MPMD ADLS system is based on two platforms; mobile computer and USB PIC microcontroller. The software schematic is shown in Figure 4.29.

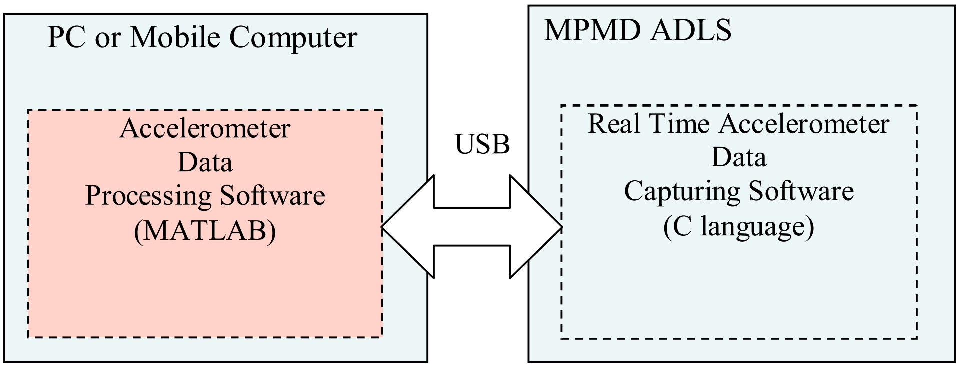

PC or Mobile Computer
Accelerometer Data Processing Software (MATLAB)
USB
MPMD ADLS
Real Time Accelerometer Data Capturing Software (C language)

Figure 4.29 The software schematic of the complete MPMD ADLS system.

On the MPMD ADLS part, the microcontroller software collects three accelerometer data simultaneously, forms a data packet in a defined protocol format and sends them to a mobile computer using on-board USB interface. On-board USB unit provides an advantage of speed and space over many currently available systems with RS232-USB converters.

The most important specification of the software is to send a convert command to all accelerometers at the same time and reading the accelerometer data simultaneously. This is one of the main requirements for the success of ICA technique. A data packet consists of 15 bytes of accelerometer data is transferred to a mobile computer via USB interface without making any processing on them. Data sending and receiving control are performed to guarantee synchronous data transfer between microcontroller and the mobile computer.

The flow chart representation of the microcontroller software algorithm is shown in Figure 4.30. START command from computer initiates data receive from data logging system and STOP command from computer cancels data transfer and puts data logging system into wait state.

MATLAB graphical user interface (GUI) was used to obtain data from data logging system and store on computer disk. The developed GUI program provides real time or offline data visualization and analyzing option. Figure 4.31 shows MATLAB software main graphical user interface screen.

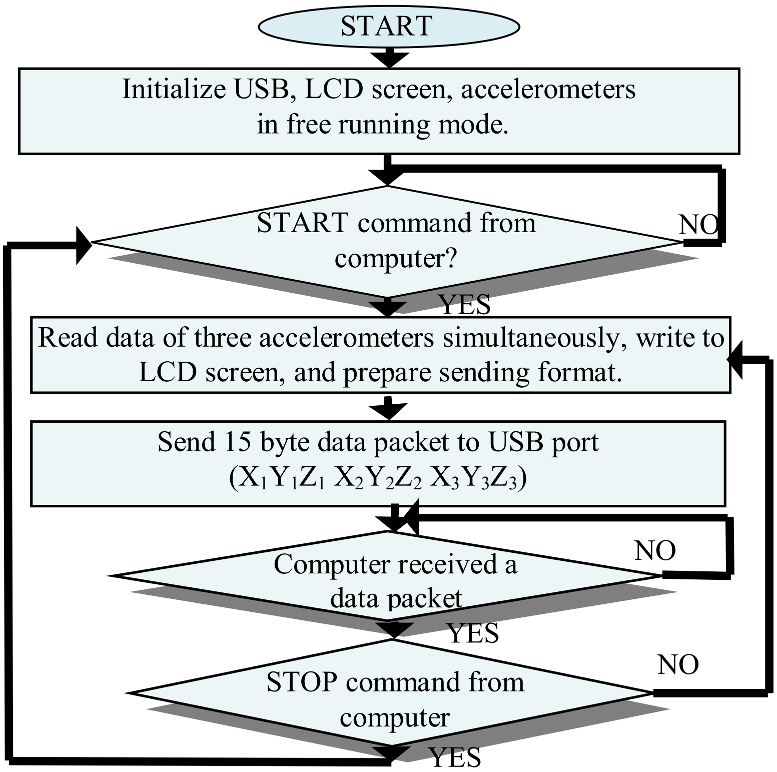
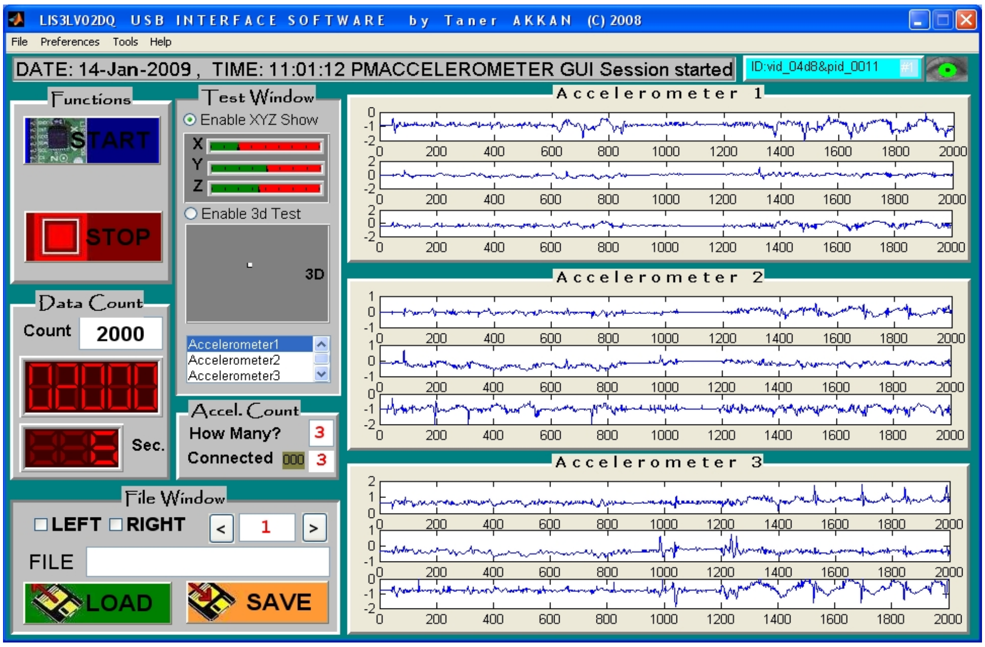

START
Initialize USB, LCD screen, accelerometers in free running mode.
START command from computer?
NO
YES
Read data of three accelerometers simultaneously, write to LCD screen, and prepare sending format.
Send 15 byte data packet to USB port $(X_1Y_1Z_1\ X_2Y_2Z_2\ X_3Y_3Z_3)$
Computer received a data packet
NO
YES
STOP command from computer
NO
YES

Figure 4.30 Algorithm of the microcontroller software


LIS3LV02DQ USB INTERFACE SOFTWARE by Taner AKKAN (C) 2008

DATE: 14-Jan-2009 , TIME: 11:01:12 PMACCELEROMETER GUI Session started

ID:vid_04d8&pid_0011

Count 2000

How Many? 3

Connected 000 3

Figure 4.31 MATLAB software developed for the current thesis work.

On the main screen, there are many buttons, option or list selections, and displays to provide a user-friendly interface. Start and Stop buttons initialize and finalize data logging process. If data count is defined different from zero, then the number of samples is collected according to provided count number. The number of accelerometers to be used can be selected from “How Many?” edit box and physically connected number of accelerometers is shown on “Connected” edit box. Previously collected data loading from memory and saving the currently collected or edited data to memory is controlled from the file window. For the test purposes, all accelerometers data can be visualized in two or three dimensional illustration. Accelerometer time trends are shown in real-time on the main screen. Figure 4.32 shows nine channels of logged data clearly when the accelerometers moved in each direction randomly.

Available MATLAB graphical components such as numeric displays or bar and three dimensional graphics are slow processes and affect the overall real-time software performance. However, graphical user interface is important for visualization. ActiveX visual components were used here for speeding up the graphical update instead of readily available MATLAB components.

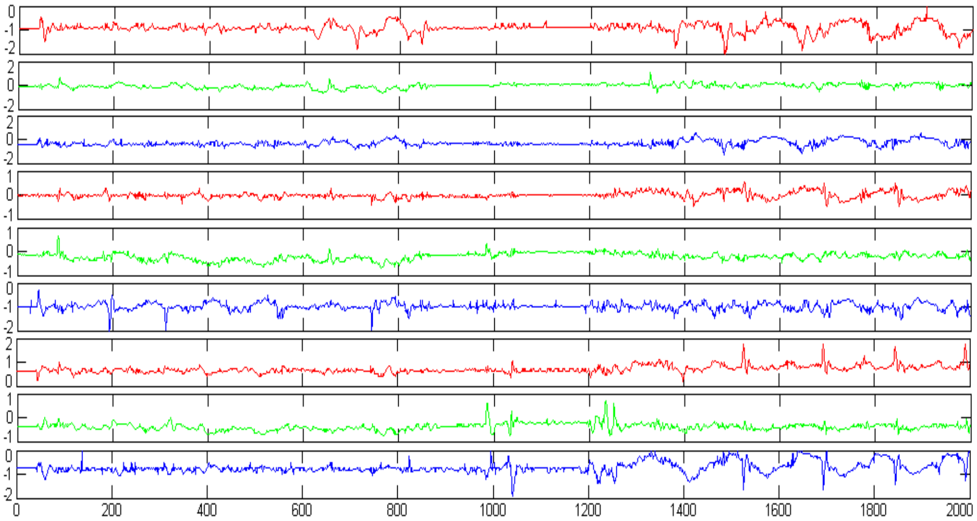


Figure 4.32 Nine channels of example logged data.

The MATLAB software opens two USB communication tunnel for reading and writing data and examines whether USB data logging system connected or not. If the system is connected, then the system USB identity is verified and the software runs online, if not software enters simulation mode for testing or analyzing pre-recorded data.

CHAPTER FIVE
ICA BIOMEDICAL APPLICATIONS

Applying Blind Source Separation (BSS) techniques is very helpful for cleaning and analyzing Electroencephalogram (EEG) data. Because of the improved separation nature, ICA is a preferred method rather than formerly used PCA by EEG researchers. The same EEG potential is recorded from more than one electrode, for that reason these signals are supposed to be highly correlated. ICA is very successful in order to get the most important information contained in EEG signals. That is important both in diagnosis and research because the amount of data to be processed is reduced.

Independent Component Analysis methods were analyzed and applied to a problem to discover the advantages and performance of it. As application areas, Artifact Cleaning on EEG Data and diagnosing the problems of human knee is selected.

5.1 EEG Study

EEG measures potential distribution as a sum of large number of neurons potentials by placing electrodes on scalp. EEG measures spontaneous activity (continuous recording) and evoked potentials (triggered by a stimulus, e.g. auditory or visual). EEG is a superior method to explore the brain activity because of its milliseconds time resolution compared to seconds and minutes resolution of other methods. EEG directly measures the electrical activity in the brain, but other methods measure blood flow or other metabolisms. Newer research typically combines EEG or MEG with MRI or PET to get high temporal and spatial resolution.

Contamination of EEG data can occur at many points during the recording process. Externally generated artifacts, such as line noise, can be removed, but biological artifact signals must be removed after the recording process using enhanced methods such as filtering and ICA algorithms. Figure 5.1 shows waveforms of some of the most common EEG artifacts.

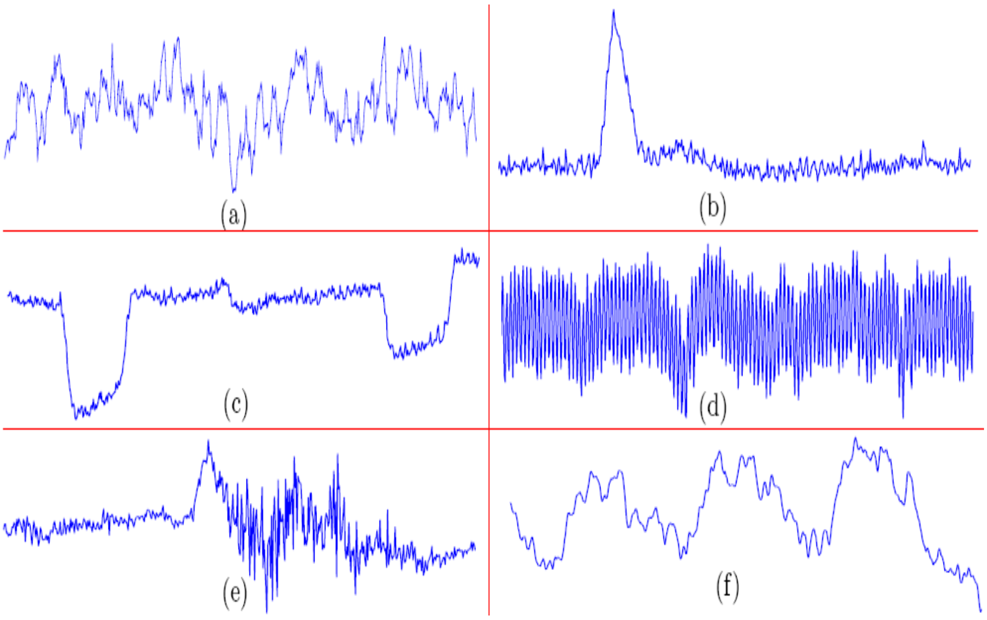


Figure 5.1 EEG artifacts. a) Normal EEG b) Eye Blink c) Eye Movement d) 50 Hz Line Noise e) Muscle Activity f) Pulse.

Applying Blind Source Separation (BSS) techniques is very helpful for cleaning and analyzing EEG data. Because of improved separation nature, ICA is a preferred method rather than formerly used PCA by EEG researchers. ICA Schematic Diagram for EEG is in Figure 5.2 (Enghoff, 1999).

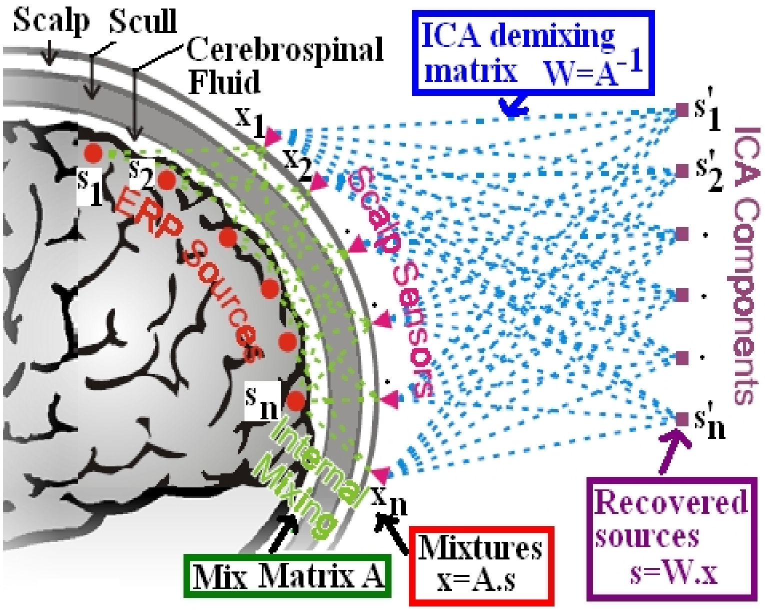


Figure 5.2 ICA Schematic diagram for EEG.

Because the same EEG potential is recorded from more than one electrode, these signals are supposed to be highly correlated. ICA is very successful in order to get the valuable information contained in EEG signals. That is important both in diagnosis and research because the amount of data to be processed is reduced. Functional relations between neocortical regions can be measured by covariance and coherence of EEG data. Evoked or event-related potentials are the subject of the study (Nunez et al., 1997, 1999). Also, there is a very popular software for ICA in EEG is MATLAB based ICALAB software (Delorme & Makeig, 2004).

The EEG signal has typical amplitude of 2-100 microvolts and a frequency spectrum from 0.1 to 60 Hz. Most activity occurs within the following frequency bands; delta (0.5 - 4 Hz), theta (4-8 Hz), alpha (8-13 Hz), beta (13-22 Hz) and gamma (30-40 Hz). Here, delta waves occur in sleep stages three and four, theta waves are seen in light sleep of before falling sleep, alpha waves occur in relaxed condition and usually relates to consciousness, sensorimotor (SMR) waves are related to physical activities, beta waves are seen in active concentration, and gamma waves occur in higher brain activities such as calculation or thinking. The frequency bands of one second EEG signal are shown in Figure 5.3. These frequency bands activities are often related to particular cognitive states (Teplan, Krakovska, & Stolc, 2006). Brain Computer Interface (BCI) related studies such as Wolpaw (2007) states that EEG signals can be used for computer interfaces. For example, detected alpha band signal electrodes over the visual cortex show visual relaxation (University of Oxford Robotic Research Group, 2009).

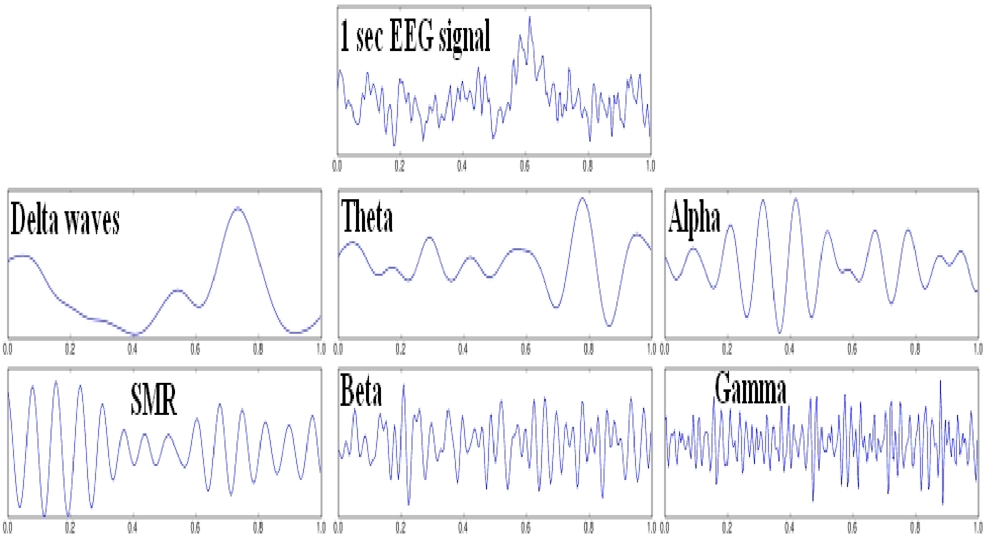


Figure 5.3 EEG frequency bands

EEG signals have high temporal resolution, but resolution is lost due to non-stationary nature. Advanced extraction techniques can cope with the non-stationarity problem. At least 0.5 sec data length is required to apply these techniques.

Cleaning of artifacts and EEG source localization are two consecutive steps in this study. Rejecting artifact contaminated EEG data is possible, but it causes information loss. Regression can be applied for removing electrooculogram (EOG) artifact using simultaneously recorded EOG signals, but also this removes valuable EEG portions. Here, ICA is a superior method for removing any kinds of artifacts from recorded EEG channels. PCA is also can be used for this purpose (Jung et al., 2000; Romero, Mailanas, Clos, Gimenez, & Barbanoj, 2003; Xue, Li, J., Li, S., & Wan, 2006).

Determining which regions of the brain are active is very challenging problem. The EEG data is first decomposed into signal and noise subspaces using PCA decomposition. After discarding the noise subspace, signal subspace contain less noisy and lower dimension data. Then ICA is applied to this signal space for separating multi-channel EEG data into activation maps due to temporally independent stationary sources. For each activation map, EEG source localization is performed by looking only for a single dipole per map. Source localization is completed by localizing multiple dipoles independently.

Here, heart pulses contaminated EEG data were analyzed using various ICA algorithms and heart pulses extracted from EEG data successfully. Also, the sweating artifact identified and cleaned using ICA. Our further levels in this paradigm will be to design custom made experiments with certain artifacts. Thus we will have a clear assesment of our methods in real life problems.

5.1.1 ICA on EEG Heart Pulses Artifact

The theoretical appeal of ICA applications makes it a powerful tool for EEG analysis in real life. So in this study, first level approach of our group is shown on this issue.

The real EEG signal was from a healthy subject at an age of 30, provided from Dokuz Eylul University Biophysics Department. The EEG was truly contaminated with heart pulses. Even the eye observation could prove the presence of approximately 70 beat/s heart rate. The contamination was so strong that the peaks of the heart pulses were in the most of the sweeps higher than the average EEG baseline rhythm. Ten channels of spontaneous EEG data (Cz, O1, O2, F3, F4, P3, P4, T3, T4, EOG) are shown in Figure 5.4.

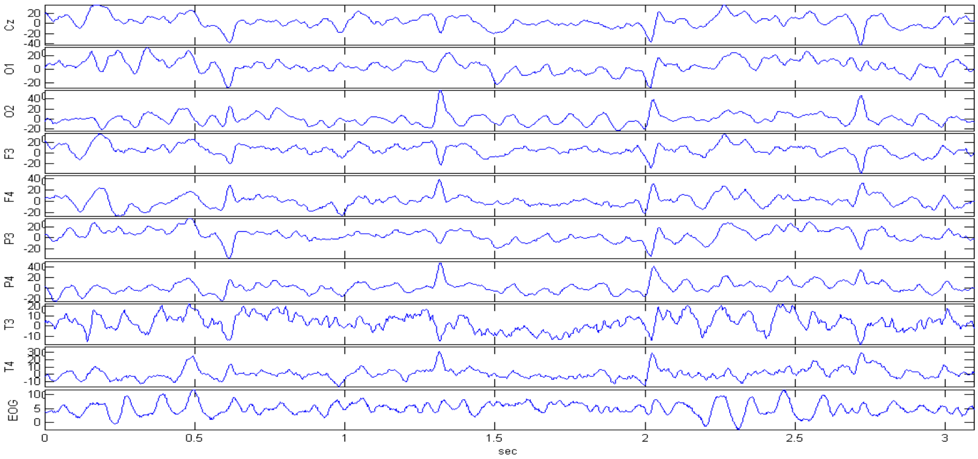


Figure 5.4 10 channel EEG data (obtained from DEU Biophysics Dept.).

2048 samples (4.096 seconds) of contaminated EEG data were processed by using Fast ICA, Efficient ICA and Maximum Likelihood (ML) ICA algorithms. Ideal approximation of an isolated contamination signal was found by ICA extraction. The results are shown from Figure 5.5 to Figure 5.8.

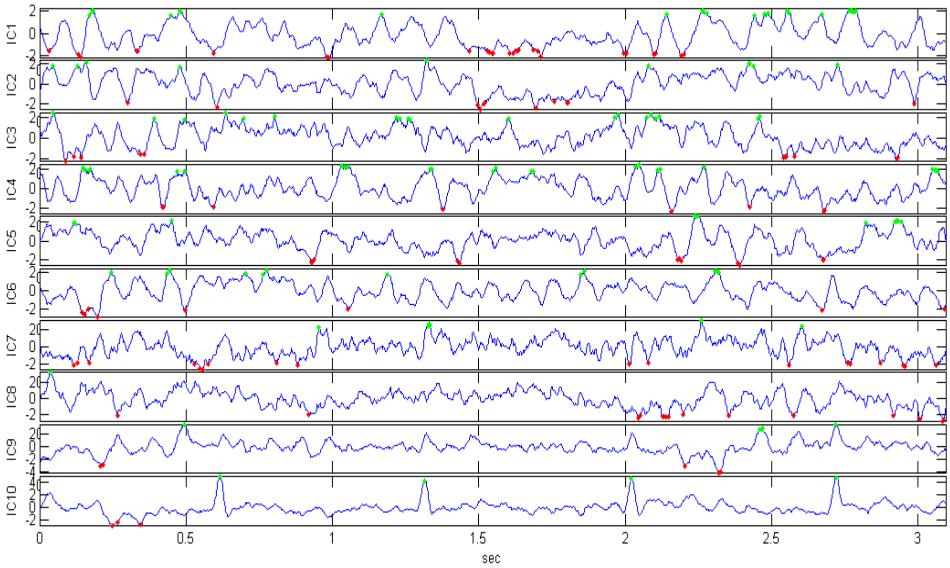


Figure 5.5 Components found by Fast ICA.

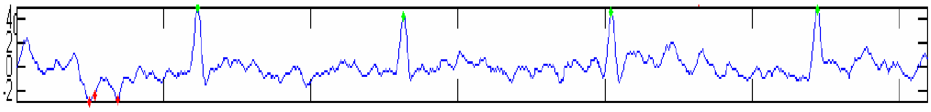


Figure 5.6 Heart Pulses Found by FastICA

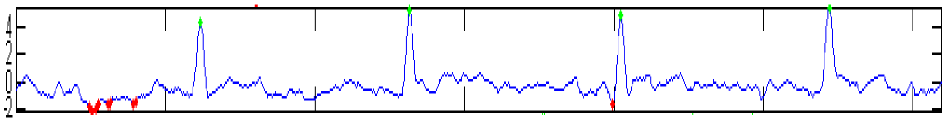


Figure 5.7 Heart Pulses found by EFICA.

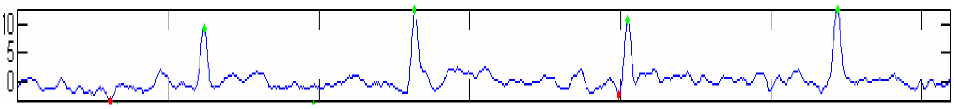


Figure 5.8 Heart Pulses found by MLICA.

Heart pulses were easily detected and extracted from EEG recordings using different type of ICA algorithms. Fast ICA algorithm doesn't converge well, but using hyperbolic tangent nonlinearity a satisfactory result was obtained. On the other hand, Efficient ICA (EFICA) and Maximum Likelihood ICA (MLICA) algorithms worked very well. Due to different statistical properties of artifacts, different ICA methods have been used. For example, extended ICA converges fast and can be used for both super and sub-gaussian signals.

EEG recordings from scalp are linear mixtures of temporally independent cerebral and artifactual sources that occur from brain parts, scalp and body. Because of the non-gaussian nature of EEG signals, ICA is very successful for extracting and removing the artifacts.

5.1.2 ICA on EEG Sweating Artifact

The real EEG signal was taken from two healthy subjects with sweating problem during EEG recording. The EEG signals were contaminated with sweating artifact as a slowly changing sinusoidal baseline signal which has approximately 10 to 20 seconds periodic-like behavior. EEG data with sweating contamination are shown in Figure 5.9. About 0.05 Hz sinusoidal artifact can be easily seen in all EEG channels. Also, at the right end of every channel plot, histogram of this channel is given for every EEG channel. This histogram gives an idea about the gaussianity of the signals.

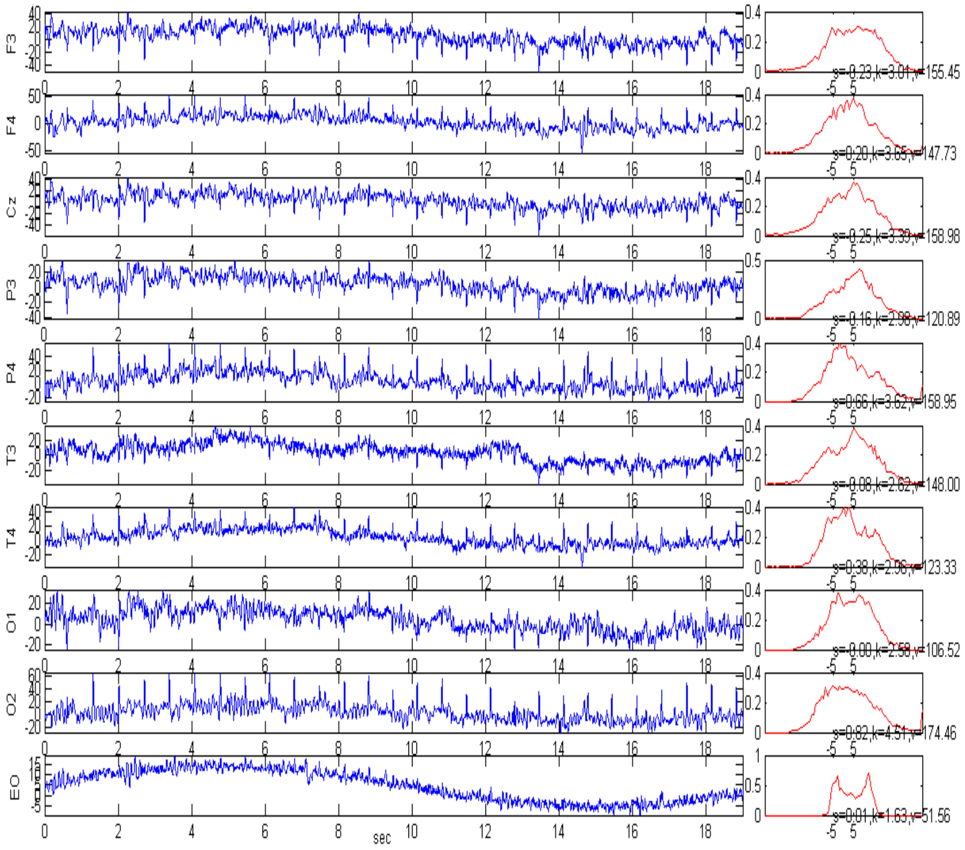


Figure 5.9 EEG channels and their histograms (obtained from DEU Biophysics Dept.)..

Sweating artifact is filtered using 0.05 Hz notch filter design. Filter Magnitude and phase responses are given in Figure 5.10. The cleaned EEG channels after filtering are shown in Figure 5.11.

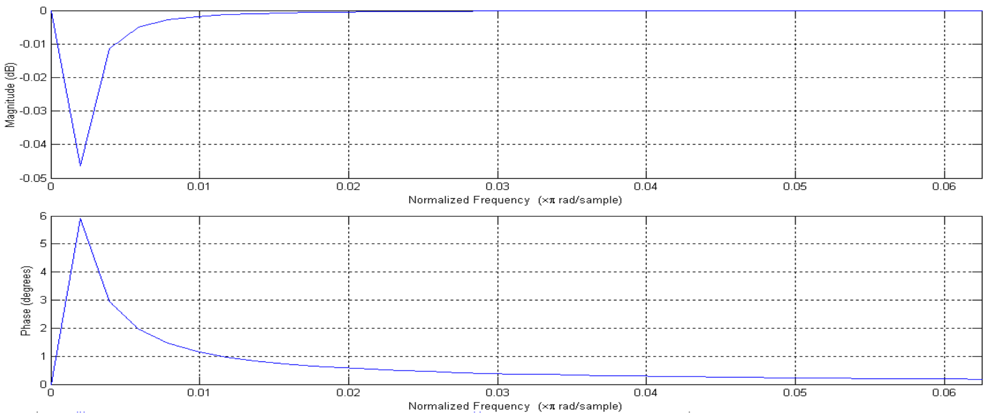


Figure 5.10 Notch Filter Characteristics.

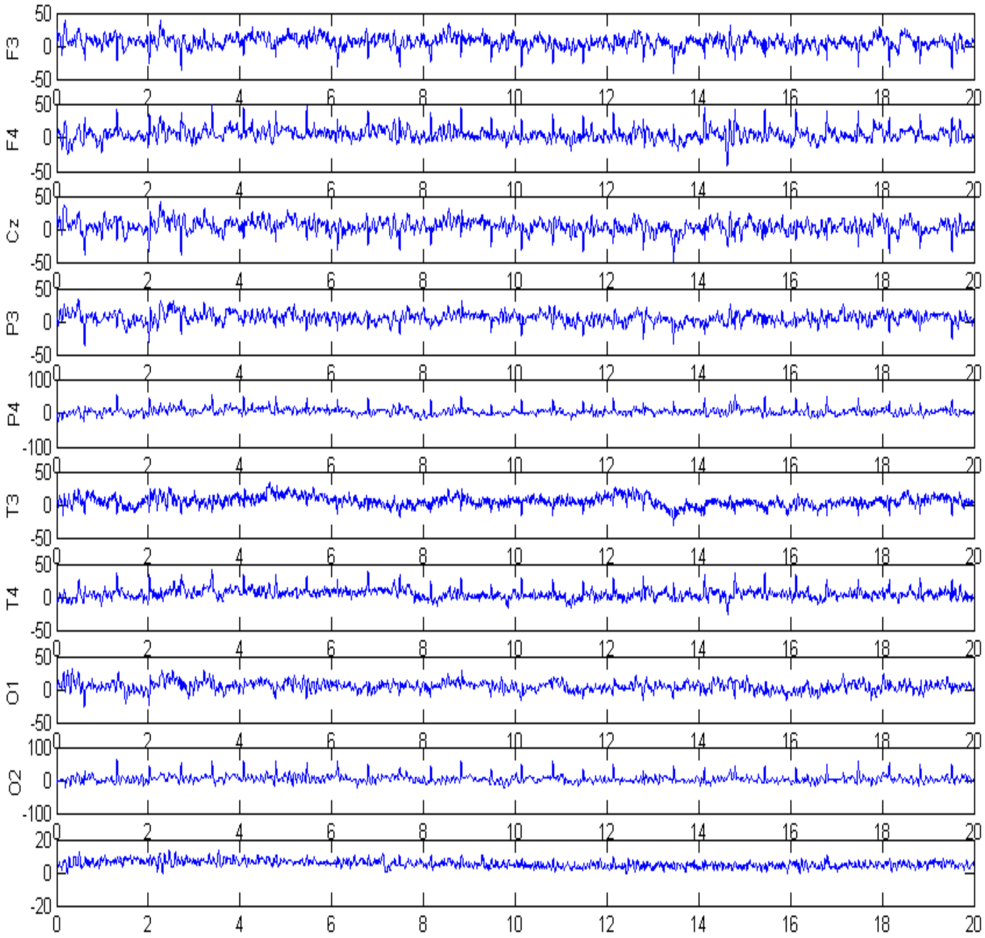


Figure 5.11 Sweating artifact removed from EEG (obtained from DEU Biophysics Dept.) channels.

Now another EEG data set will be analyzed for muscle contamination. After cleaning these muscle artifacts, a custom experiment will be designed for EEG data recording. This experiment includes every artifact contamination scenarios.

EEG recordings contain desired brain activities data and undesired artifacts. These artifacts that caused from muscles, line noise, sweating and electrode reference problems must be removed using various filtering methods or statistical analysis.

Sweating artifact is caused by skin impedance change on the electrodes. Sodium chloride and lactic acid from sweating reacting with metals of the electrodes may produce slow baseline sways at high amplitude. Frontal electrodes generally contains sweating artifact. Removing this artifact can be done with a high-pass filter. Designing this very low frequency filter is not easy, but still practically applicable.

A band-pass filter was designed to find the sweating artifact. Also moving average can be used for this purpose. Possible slow oscillations due to sweating artifact can be detected using autocorrelation function. Independent component analysis (ICA) is also used to find this sweating artifact component In this study spontaneous EEG recordings were obtained from subjects with sweating problems. These kind of sweating signal frequencies are in the range of 0.05 Hz to 0.5 Hz. This requires 2-20 seconds of recordings to analyze for one complete period (Siddiqui, Osuna, Walters, & Chokroverty, 2006). Sweating artifact was identified and extracted from ten channel recordings by using ICA techniques.

Figure 5.12 shows ten channels of spontaneous EEG data (Cz, O1, O2, F3, F4, P3, P4, T3, T4, EOG) and their autocorrelation graphics. Possible periodic-like signal channels can be detected by investigating periodicity of autocorrelation results.

A 0.05 Hz band-pass filter was designed to find this sweating artifact. Figure 5.13 shows possible sweat artifact oscillations after filtering.

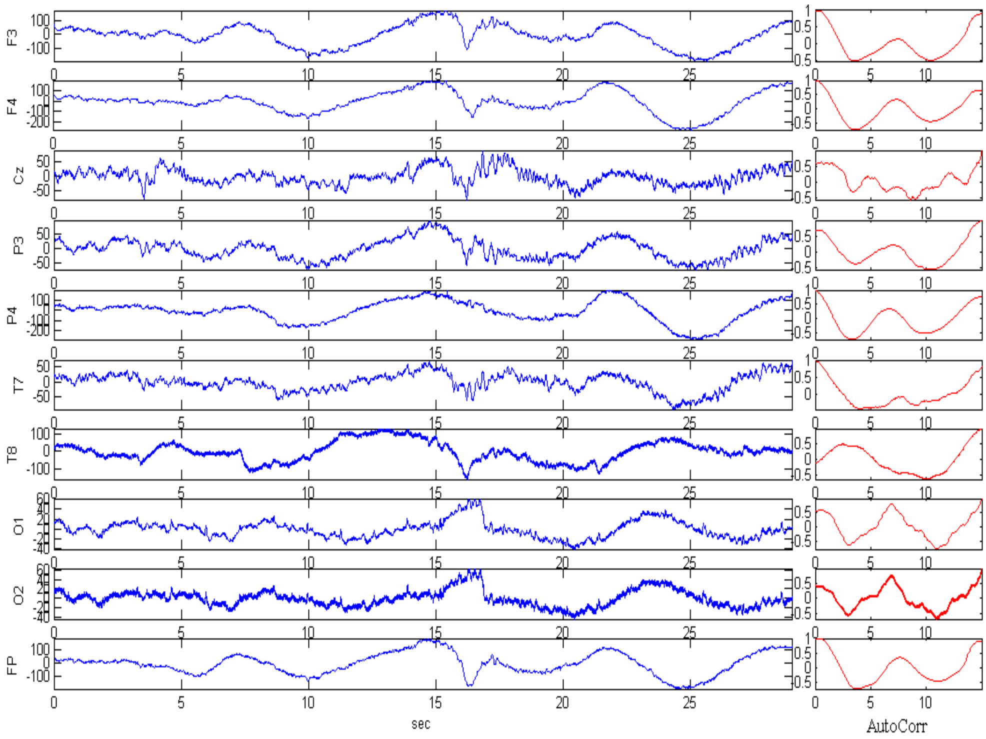


Figure 5.12 Ten channels of sweating contaminated spontaneous EEG data (obtained from DEU Biophysics Dept.).

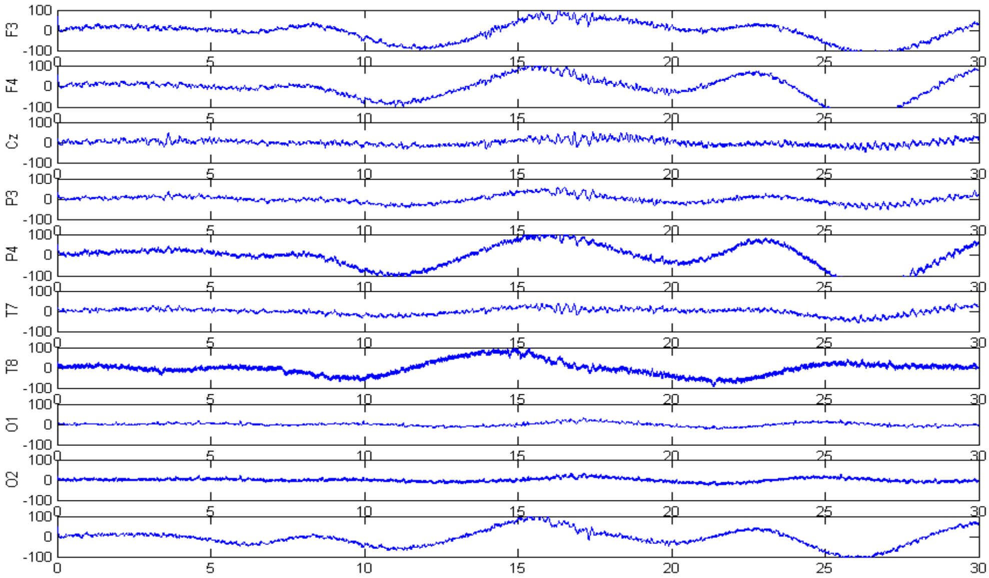


Figure 5.13 Band-pass filter results.

Contaminated EEG data were processed by using Fast ICA, Efficient ICA and Maximum Likelihood (ML) ICA algorithms. Although they gave similar results, here only Fast ICA results are presented in Figure 5.14. Approximate sweating signal is on the last channel of ICA results that ordered by their kurtosis values.

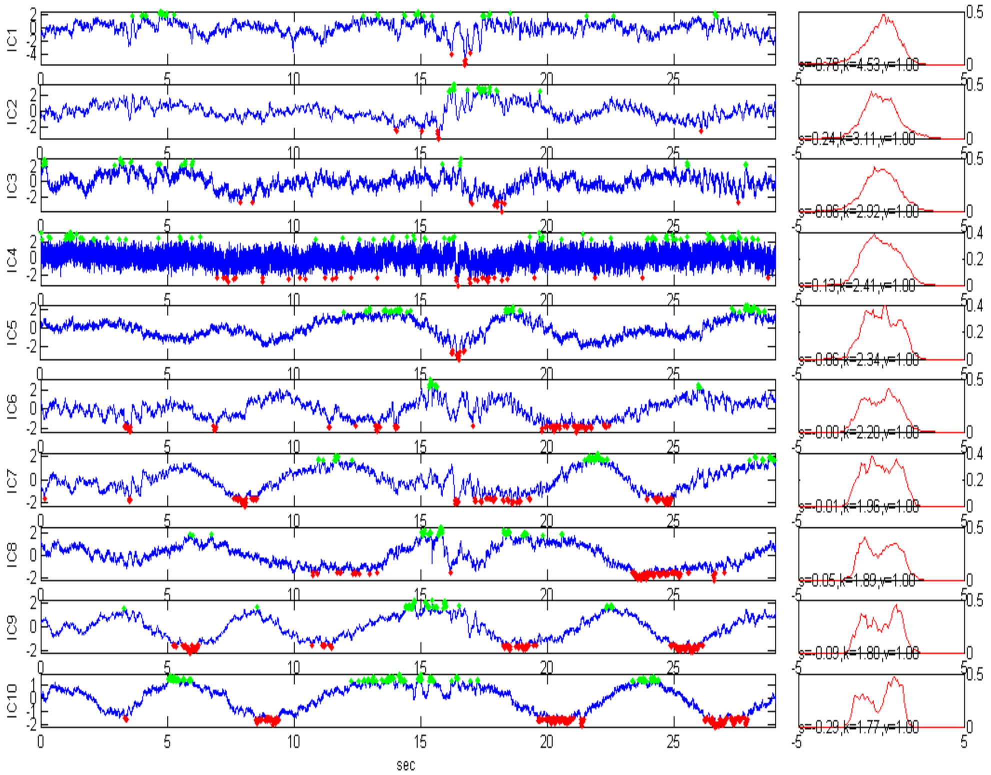


Figure 5.14 FAST ICA results and their kurtosis values.

The result can be seen more clearly by applying moving average method in Figure 5.15. The default span for the moving average is 5. The same analysis results for another subject are shown in Figure 5.16, 5.17 and 5.18.

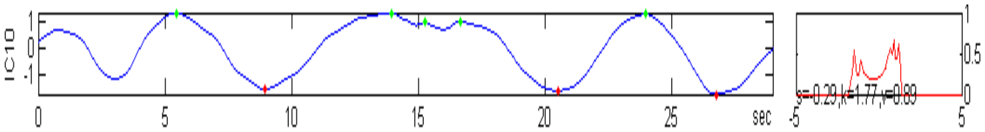


Figure 5.15 Moving average result of Last IC

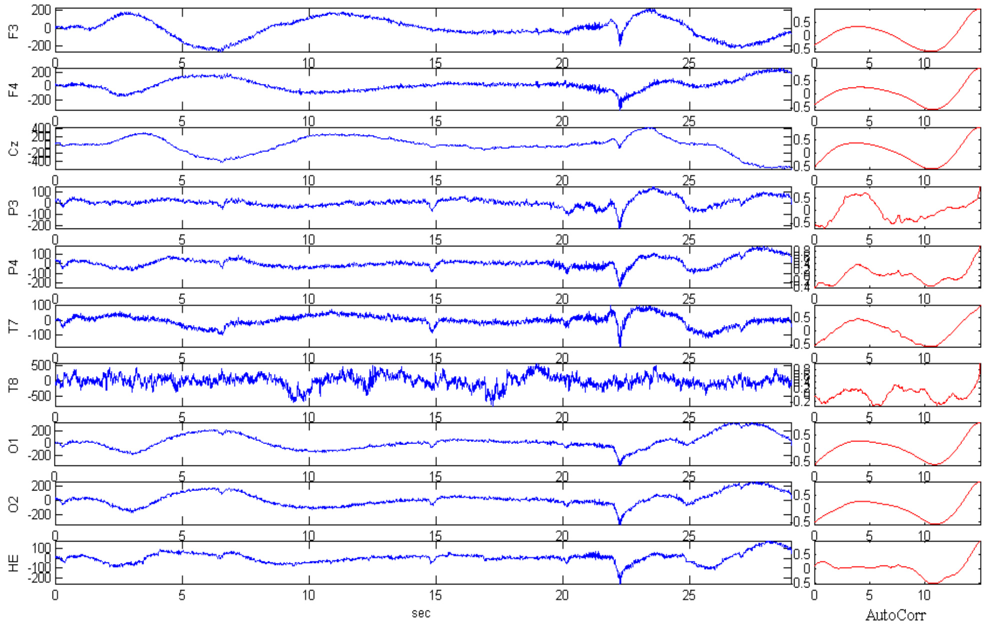


Figure 5.16 10 channel of EEG data and their autocorelation results.

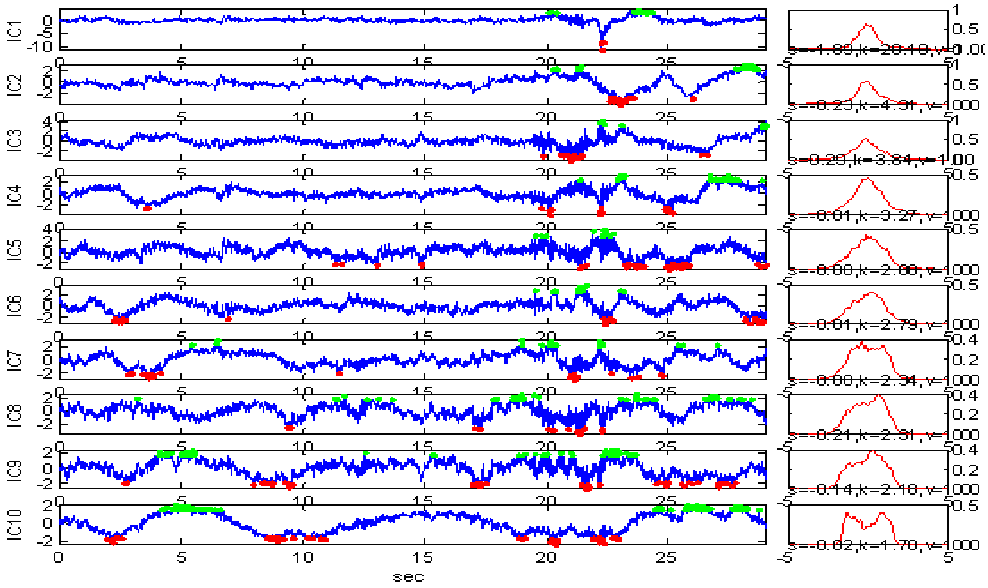


Figure 5.17 FAST ICA results and their kurtosis values.

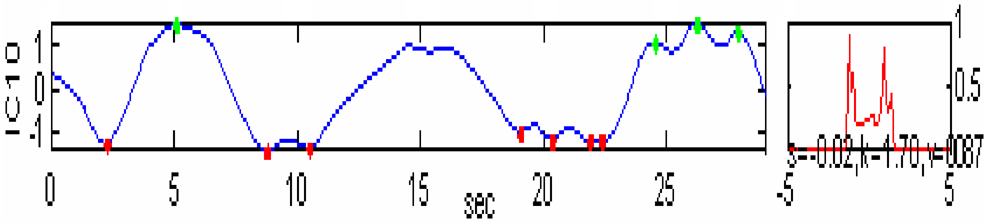


Figure 5.18 Moving average result of Last IC.

Sweating artifact was detected on the last channel of ICA results. Results using Fast ICA algorithm is quite satisfactory, on the other hand, Efficient ICA (EFICA) and Maximum Likelihood ICA (MLICA) algorithms also give the similar results.

The other artifacts, if they exist, especially muscle artifacts could be seen on the last channel of independent components which has the smallest kurtosis value. This method needs to be improved to obtain pure sweating artifact, because only kurtosis criteria does not enough to classify it. Instead of sweating artifact removing by high-pass filtering, using ICA method can be a solution for not losing the very low frequency EEG information.

5.2 Knee Study

Knee pain is an extremely common complaint, and there are many common causes. It is important to make an accurate diagnosis of the cause of your symptoms so that appropriate treatment can be directed at the cause. Some causes are meniscus tears, tendonities, deformity of the knee joint with the age, pilica syndrome, etc. Usually to diagnose them, X-Rays, MRIs, or arthroscopy methods are used. Purpose of this study is developing a non-invasive method that supports diagnosing. The first studies about knee problems diagnosing were done by Tavatha, et.al (1992) and Zhang, Rolston, Rangayyan, Frank, & Bell (1992).

After using stethescope, it is found that knee researches are using accelerometer instead of microphones or stethescope-microphone combination. But this stethescope-microphone combination can be also searched whatever can be used or not for knee recording in future. For example a new research was about stethoscope based knee diagnosis system by Kim, Seo, Kang, & Song (2009).

An easy way to diagnose knee problems from the exterior of the knee using microphones with stethescope heads attached to the kneecap, and lateral knee positions. Using stethoscope head provides us much gain than the one in microphone only, so the gain must be reduced. The most important problem is to get the clean sounds from the knee bones, cartilages, ligaments. Moreover, friction of the microphone surface on the skin of the knee will cause noise. Independent component analysis is used to separate the sound sources in the knee and also remove the noise.

Microphones are replaced by accelerometers for better achievement. Accelerometers are well suited for the signals that are vibrations like signals. Again ICA is needed to extract the knee signal from the other signal sources. Here, also friction of the sensor to the surface of skin is not problem as in the microphone type. These signals recorded from knee are called as vibroartographic (VAG) signals.

Studies (Krishnan & Rangayyan, 1999, 2000a; Krishnan, Rangayyan, Bell, & Frank, 2000b, 2001) show us the proper selection for knee recordings is using accelerometers. For VAG analysis, also another study by Eskandari, et. al (2003) has been done with analog accelerometer. Accelerometer is used for detecting vibrations on the knee cap when leg is on the movement. Microphones are not capable to detect the very low frequency vibrations, but accelerometers are. Therefore, new accelerometer interface has been developed and beside all mentioned knee studies, a digital accelerometer was chosen in this thesis study.

The main aim of the designed data logging system is to develop a non-invasive diagnosing technique in order to identify knee related problems (especially the degeneration of articular cartilage surfaces). The undesired signals caused by movement of knee, sensor-skin surface interactions, and muscle contractions can be removed from knee recordings and the desired VAG signals for diagnosing of knee problems can be obtained by using Time-Frequency analysis. Especially VAG signals are non-stationary. Therefore time-frequency analysis is very powerful on that kind of analysis. Nevertheless, so far there have been no applications about using ICA for the filtering and analysis of VAG signals. We assume that, with the proposed MPMD ADLS, it will be possible to utilize ICA, especially, to extract the signal of interest from the accelerometer sensor array.

Diagnosing knee problems with the vibration analysis on the knee cap and lateral locations would be possible with the data logging system. The experimental protocol was approved by the Clinical and Laboratory Research Ethics Board of the Dokuz Eylul University.

The positioning of the three accelerometers to a knee cap and to lateral positions is given in Figure 5.19a. Here, while the patient is moving the leg from the ground to a certain upward position, vibrations inside the knee and also the angle position of the leg in respect to the initial resting position are collected and transferred to a mobile computer for analysis. Figure 5.19b shows the position of the accelerometers for the right leg and also illustrates the leg movement. The VAG signal analysis cannot be performed by the currently designed MPMD ADLS. However, when a useful algorithm is developed to identify normal and abnormal VAG signals, it would be possible to use more powerful microprocessor or processors to realize on-board real-time analysis by MPMD ADLS.

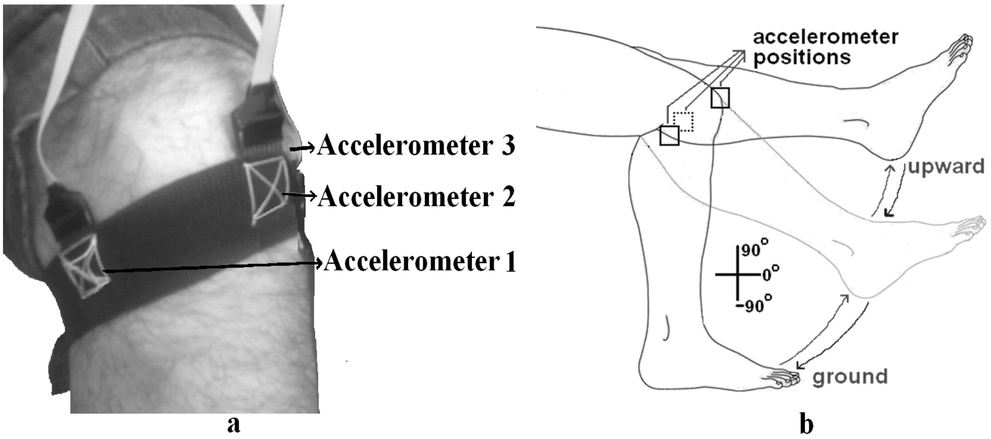


Figure 5.19 (a) Photo of the three accelerometers connected to the knee.
(b) The movement of the leg during recording

Analyzing the structure of the knee joint is important to define the source of vibration signals. The frictionless system of knee contains the synovial fluid, articular cartilage and supporting bones (Buckwalter, J.A, Einhorn, T.A, Simon SR

(Eds): 2000). Synovial fluid is important for lubrication and it also provides a medium for the propagation of vibration signals. As a result, vibration signals obtained from normal knee will be exempt from the additive vibration signals caused by abnormalities. Figure 5.20 shows the schematic of the knee joint and the synovial fluid.

Analysis of the knee joint movement is also very important to understand the nature of the vibration signals. The main muscles responsible for the knee joint movement are the quadriceps and hamstring muscles. The quadriceps is attached to the patella, and the patellar tendon connects this muscle to the front of the tibia. When the quadriceps muscles contract the knee extends. In contrast, when the hamstring muscles contract, they pull the knee into flexion. It is clear that these mentioned muscles contraction will take part within the VAG signal.

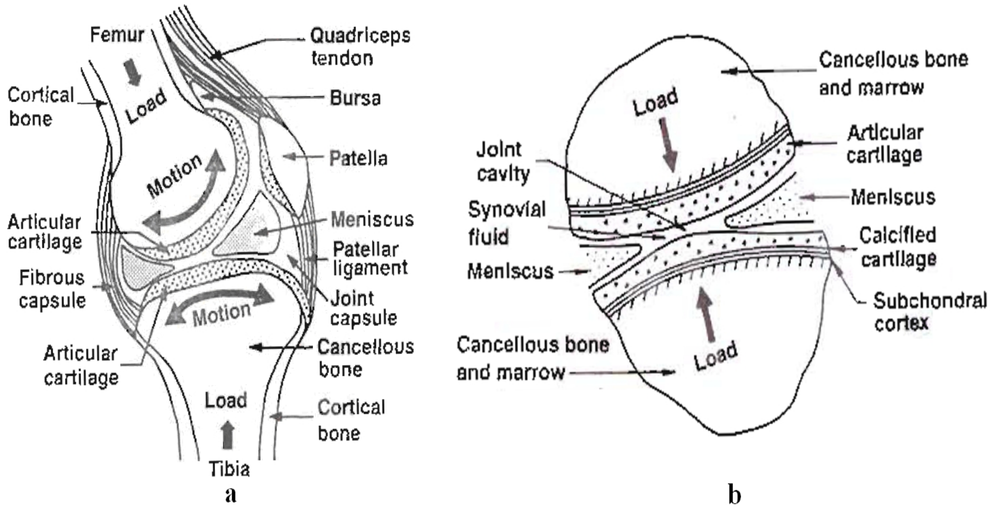


Figure 5.20 (a) Schematic of the knee joint.(b) Synovial fluid and joint cavity. (Reprinted with permission from Buckwalter JA, Einhorn TA, Simon SR (eds): *Orthopaedic Basic Science*, 2nd edition. Rosemont, IL, American Academy of Orthopaedic Surgeons 2000)

The repetitive movement of the leg can be achieved in two steps:

a) The movement starts from a position of the foot resting on the ground, and continues rising the foot upwards. The movement ends at an angle smaller than 90° because of the limitation of the knee joint.
b) The reverse movement is achieved from top to the ground.

Of course, the speed of the movement is important. More smooth movements can be obtained at high speeds, but the resolution of the vibration recordings is reduced. The resolution can be made higher, but more discontinued movements can be obtained at slow speeds. Here, a mechanical system can be built for supporting the knee movements in an adjustable knee movement angle without forcing the limits of the knee and adjustable knee movement speed.

Two accelerometers can be used for knee recording; one for vibration recording and one for y axis tilt angle reading. But for the ICA, it is more appropriate to use the three accelerometers for vibration recording. Moreover, the angle information is obtained from the accelerometer located on the knee cap.

Using the mentioned experimental setup, the three channel vibration data was obtained with an MPMD ADLS for the duration of 8,000 sample counts (400 samples per second). Nine channels of logged raw data are obtained from knee movements. Each consecutive three signals representing vibrations on the x, y, and z axes directions belong to a related accelerometer. Thus, there are a total of nine signals. The signal amplitudes show the g-force in the range of -2g to 2g for the recorded time interval of 20 seconds. Each signal shows a periodic-like behavior representing the knee movements, in the direction down to up and up to down. The crest corresponds to the extended leg position, whereas the trough corresponds to the flexion position.

To have a better understanding, a small part of the x axes data from each accelerometer is illustrated in Figure 5.21a. The actual vibration data are superimposed on the tilting data and the low frequency component shows the leg movement upwards and downwards. This low frequency component can be found using a running average algorithm. The tilting information was filtered out using a running average algorithm to obtain the vibration data alone. The vibration information is shown in Figure 5.21b. Other filtering techniques are also available to obtain better results for getting the most appropriate knee vibration data.

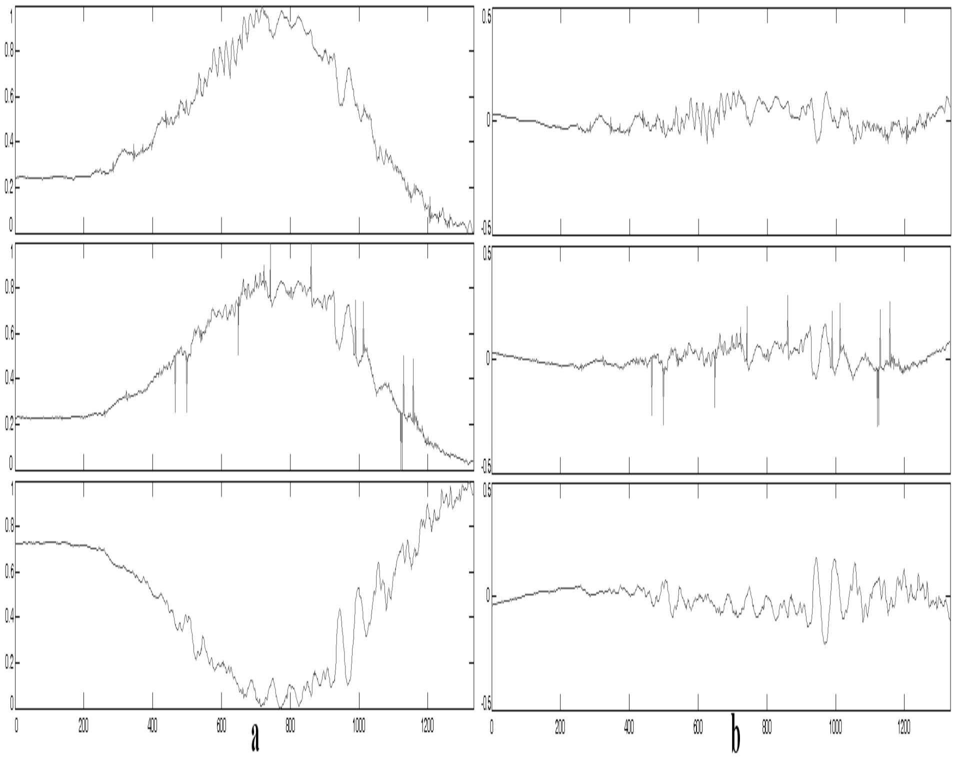


Figure 5.21: (a) Three channels knee raw data. (b) Three channels knee vibration data.

5.2.1 Vibration Data Filtering Performance

For getting the vibration data, the method is to find the baseline signal which is tilt data and subtracting from the original data. Two types of approach used in filtering the baseline. Transform types are finite-duration impulse (FIR) filter, infinite-duration impulse (IIR) filter, fast Fourier transform (FFT), and Wavelet Daubechies Filter. Smoothing types are Running Average Filter and Hodrick-Prescott Filter. Their filtering performances are evaluated in terms of their variances, skewness, kurtosis, root mean square error (RMSE) parameters. Also, their histograms and spectrograms are used to find the similarities and the differences between the information signal and filtered signal.

The knee accelerometer data has maximum 2 Hz baseline. The synthetic signal has assumed to be a 2Hz sinusoidal signal and the real vibration data added to a certain time periods of the baseline signal as a Gaussian random signal. The Figure 5.22 shows 1 Hz baseline, vibration info and the mixed signal.

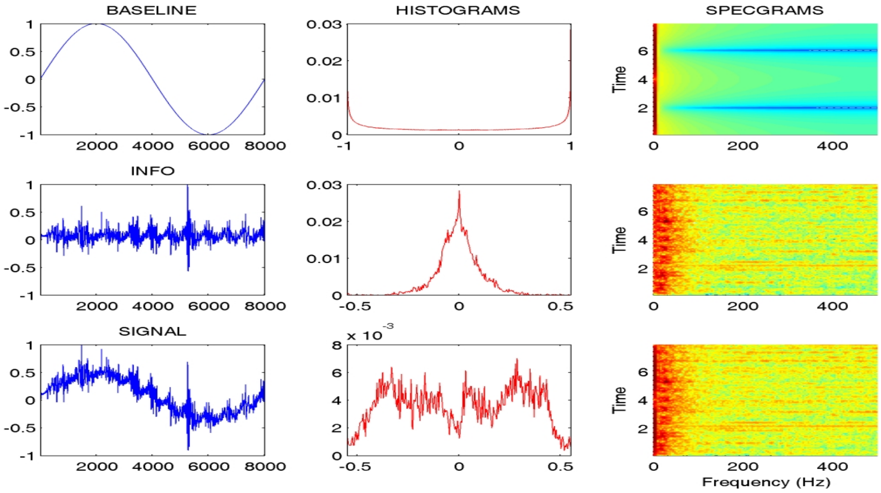


Figure 5.22 Baseline, Vibration information, mixed signal, their histograms and spectrograms.

The results of the filter errors are shown in Figure 5.23, signals, their histograms and spectrograms respectively. The first signal simulates the accelerometer data; the second one is the information signal we want to get it without any loss. Also variance, skewness, kurtosis values are printed following the name of the signals.

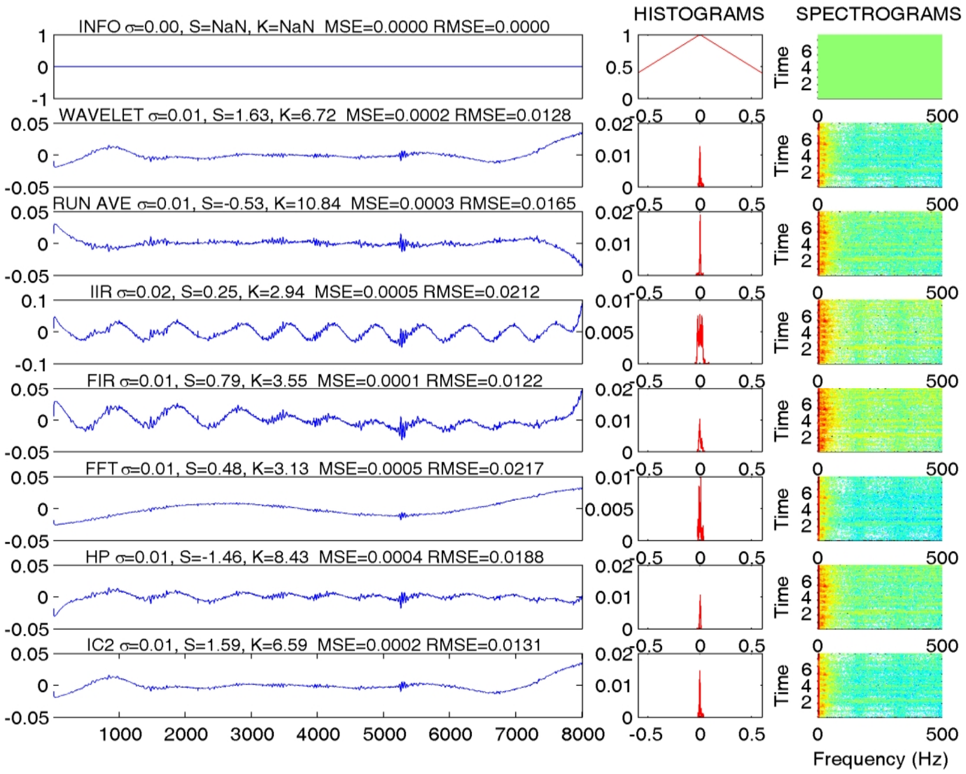


Figure 5.23 Filter Errors

If the performance parameters of the filters evaluated with the following graphics in Figure 5.24, most appropriate filter is the Wavelet Filter and the second good filter is FIR Filter. Also, accept kurtosis value is high; Running Average Filter is good choice.

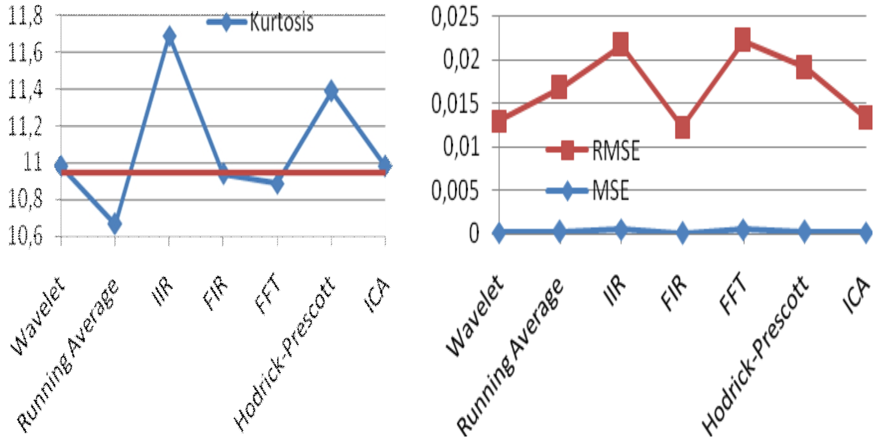


Figure 5.24 Filter Comparisons

5.2.2 Knee Data Analysis

Normal and abnormal knee VAG data are shown in Figure 5.25a and 5.25b respectively. It is observed that a normal knee VAG data has added information at a certain place of the rising part of the signal. This can also be felt physically as a vibration when you place your hand on the knee cap while moving the leg from ground upwards. However, there is no such information if the leg is bent downwards to the ground direction. Also, some oscillatory signals with small amplitudes and some spikes with small amplitudes appear other times.

As can be seen, an abnormal knee VAG signal looks similar to a normal VAG signal form, but has many and significant impulsive amplitudes. At every cycle of the abnormal knee signal, fluctuations do repeat themselves. Thereof, it is consistent with the leg movement sequences. When the normal and abnormal VAG signals are compared, the additional signal fluctuations on the abnormal signal can be interpreted as the information occurring as a result of knee related problems.

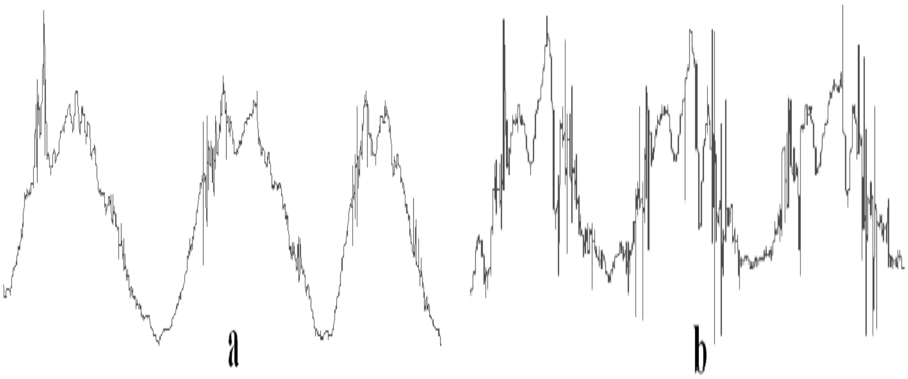


Figure 5.25 (a) A normal knee VAG data. (b) An abnormal knee VAG data.

In total, left and right knee data from ten cases were recorded using MPMD ADLS device and pre-analyzed to classify which cases are normal or abnormal. Eight cases are verified with the X-ray and MR tools with the expert diagnosis, two cases have no tools to be verified. Abnormal and normal cases were detected with the pre-analysis of the accelerometer data, but they are not classified for their related knee problems.

The analysis route map has some steps:

- Signals are recorded from healthy knees to define the signal features by the help of medical experts.

- Signals are recorded from healthy and non-healthy persons using our four channel capture system and a signal database will be created. The results of the diagnosing methods such as MRIs, X-Rays, or arthroscopy for the same patient are used for the performance analysis of the proposed system.

- After creating the signal database, ICA recovers the desired signals to extract features for determining the problem of the knee.

- A neural network system also are designed and tested to diagnose the problem of the knee. Neural Network system outputs and the pre-recorded diagnosing method outputs are compared to determine the success of the methods.

Vibration data from left and right knee has been collected for three times. We have eight cases with expert diagnosis shown in Table 5.1. Expert diagnosis method is also described in the table. The most reliable technique in the diagnosis phase is the arthroscopy which can see the inner view of the knee joint by a camera, but of course this is a invasive method. Magnetic Resonance Imaging (MRI) is also reliable technique and it is non-invasive method. XRAY is sometimes a fast, cheap and descriptive method usually medical doctors use.

Table 5.1: Cases that we have expert diagnosis.

Case Name	Diagnosis	Method
AG	Left early grade arthrosis.	XRAY
ET	Left medial meniscus rear horn degeneration. Right medial meniscus rear horn degeneration.	MRI
HK	Left high-grade arthrosis Right high-grade arthrosis.	XRAY
AE	NORMAL	XRAY
FC	Left high-grade arthrosis* Right early-grade arthrosis	XRAY
MK	Left early grade 1 arthrosis. Right early grade 1 arthrosis.	XRAY
İM	Left high-grade arthrosis Right normal-grade arthrosis	XRAY
DA	Left normal Right normal	XRAY

(*arthrosis is a common condition of degeneration of the knee acromioclavicular joint),

There are two cases that have no expert review in Table 5.2.

Table 5.2 Cases that have no expert diagnosis.

Case Name	Complaint (Case said his/her problem)	Method	Diagnosis
Aİ	Right knee problem. Left knee inner meniscus tear?	-	-
SE	Left and right knee pain	-	-

Analyses were done with the MATLAB software ciz_vektor.m. Here, two conditions are: Normal knee data and abnormal knee data.

Data with no knee problem (da_normal.dat) is shown in Figure 5.26 and knee data in detail is shown in Figure 5.27.

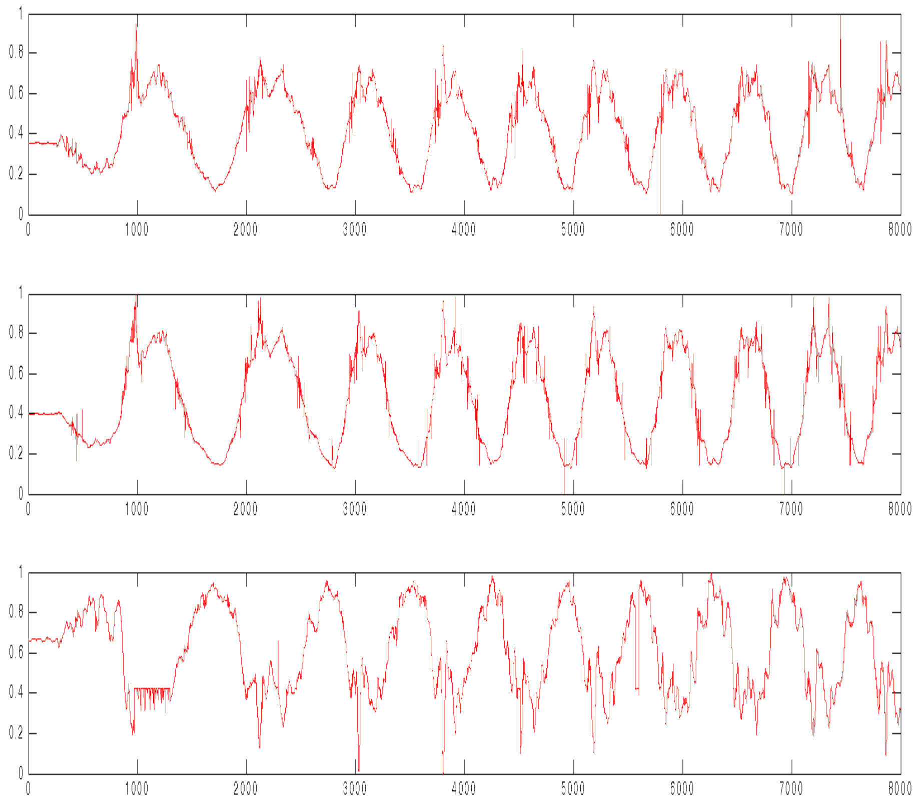


Figure 5.26 Normal knee data

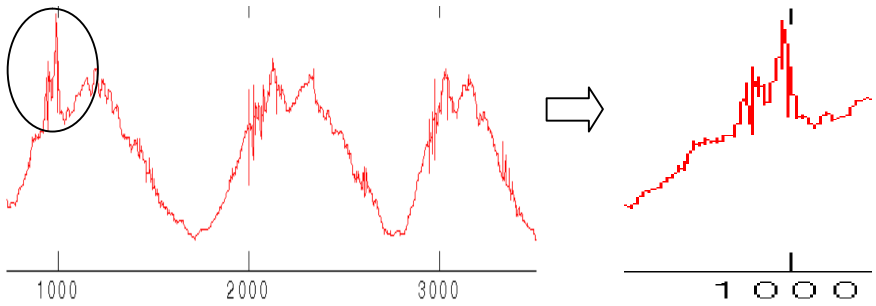


Figure 5.27 Normal knee data in details.

Normal knee data has added oscillatory amplitudes at a certain place of the rising signal. This can be felt as a vibration by pressing your hands on the knee cap when you moving your leg from floor to upwards. But you can’t feel the same when the knee is moving upwards to the floor direction.

Data with knee problem (et_abnormal.dat) is shown in Figure 5.28 and knee data in detail is shown in Figure 5.29.

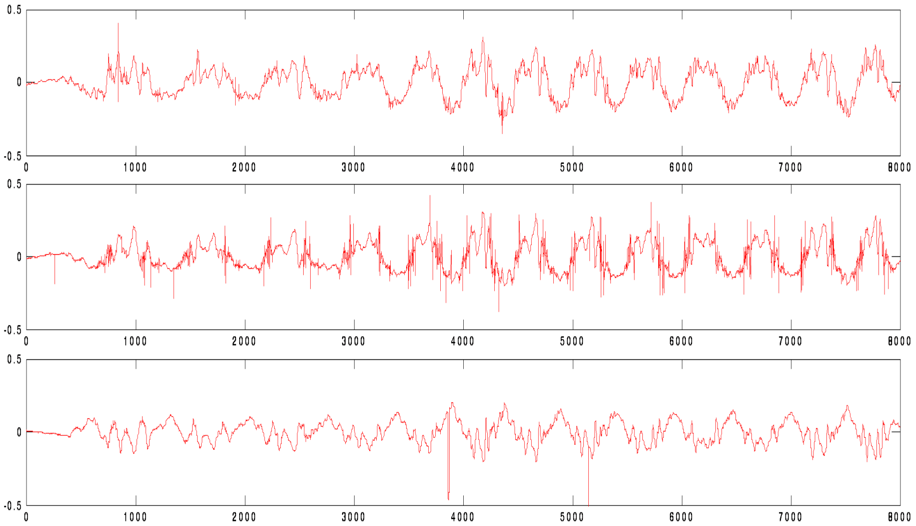


Figure 5.28 Abnormal knee data

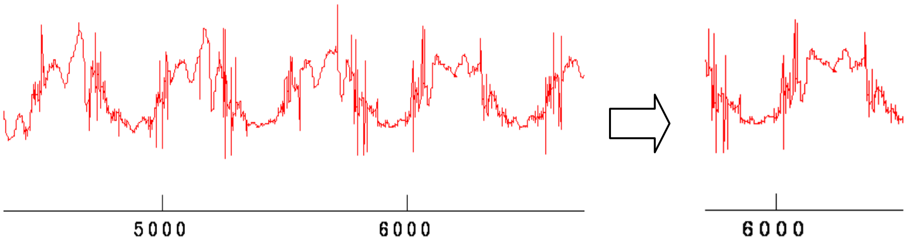


Figure 5.29 Abnormal knee data in details.

Initial phase of rising and falling signals have noisy components. This means, the examined knee has a problem and probably meniscus tear with using the diagnosis.

5.3. Filtering Jaw Related Artifacts from EEG Recordings

Various filtering and blind separation methods are used to clean EEG artifacts. Some of these artifacts are muscle activity, eye movement, eye blink, electrocardiogram, respiration, skin, EEG electrode, salt bridge, and power line 50 Hz (Jung, et.al, 2000; Selim & Diego, 2008). In order to clean such artifacts easily, EEG signals are recorded together with some extra signals carrying information related with the artifact occurrences. These extra signals are used to identify the location of artifact formations on the recorded EEG signals. For this purpose, a proper sensor must be used for each specific artifact recording in addition to EEG electrodes. For example, electro-oculogram recordings are used to remove eye-blink artifact. Jaw movements are accepted as artifact, and therefore must be removed from EEG recordings. The accelerometer is a proper sensor selection to identify jaw related EEG artifacts. Here, we propose MPMD ADLS to be used to record jaw movements simultaneously with EEG recording electrodes. Figure 5.30 shows the accelerometer placements on the subject's face attached to the jaw and chin.

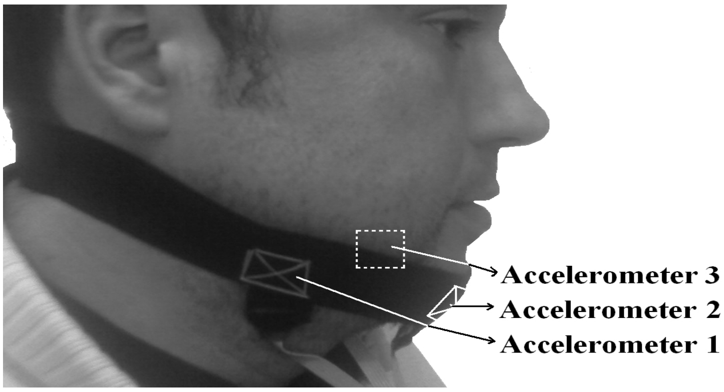


Figure 5.30 Connection of three accelerometers to the jaw.

Using the above mentioned experimental setup, three channel vibration data was obtained for the duration of 20 seconds. Nine channels of logged raw data from jaw movements are obtained. Here, during recording the subject made various jaw movements. Figure 5.28 shows the selected axes (x axis from accelerometer one, z axis from accelerometer two, and x axis from accelerometer three) data from each accelerometer for the 8,000 sample recordings. First and third accelerometers are placed on the temporal areas of the jaw, and the second accelerometer is placed on the chin. Here, 0-3,000 samples are recorded while the subject moved his/her head up and down, between 3,000-5,000 samples the head turned left and right, and between 5,000-8,000 samples the subject spoke out the number digits from one to nine in English.

The last signal from Figure 5.31a was filtered with a running average algorithm to obtain head movements. The up and down vertical movements of the head can be seen more clearly between 0 to 3,000 samples. However, between 3,000 to 5,000 samples, the head moved only from left to right and vice versa. Therefore, a constant signal level is expected at this interval. However, very small signal fluctuations are seen in this interval probably as a result of moving the head slightly up and down simultaneously. After 5,000 samples, the number digits were spoken out without any head movement.

For each x axis of each accelerometer a running average algorithm is run, and the obtained head movement related data is extracted from the corresponding channel and given in Figure 5.31b. This information is now related with jaw movements and independent from head movements.

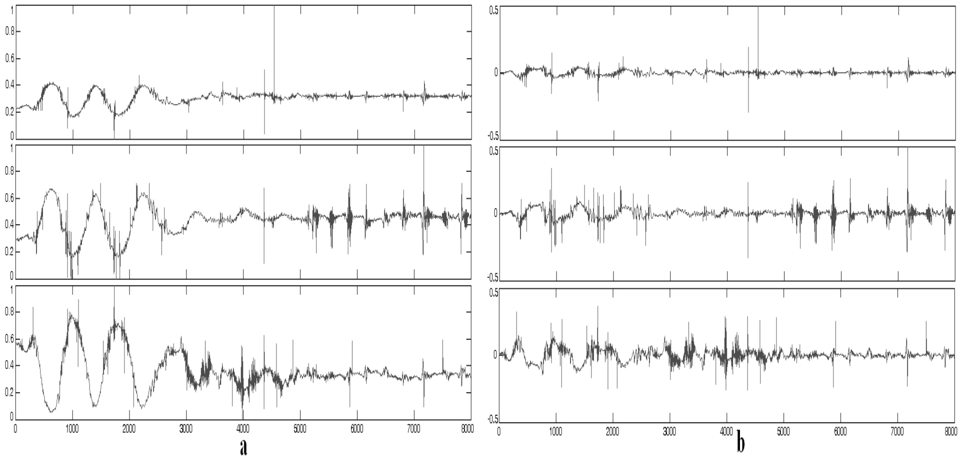


Figure 5.31 (a): Three channels vibration raw data from jaw. (b) Jaw vibration data.

Data from jaw using three accelerometers simultaneously recorded with EEG is important to identify and cleaning jaw artifacts from EEG data. Figure 5.32 shows accelerometer connection to the jaw and the EEG cap.

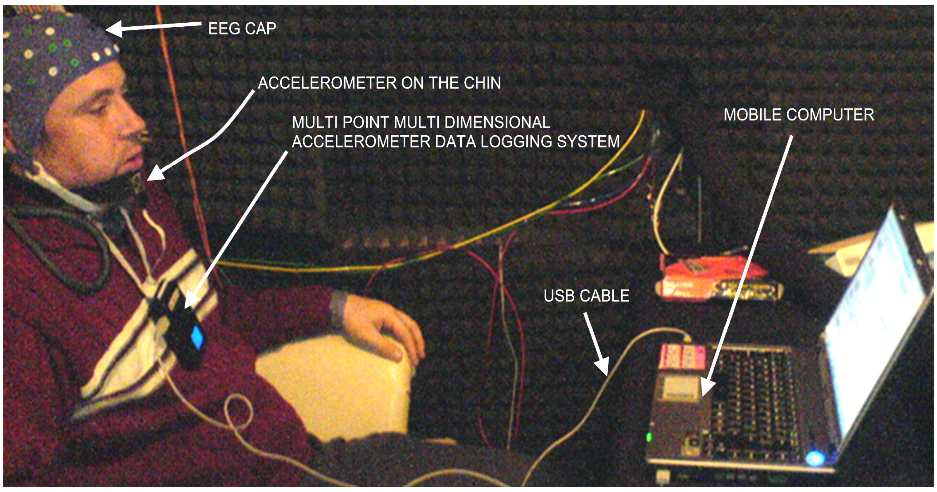

EEG CAP
ACCELEROMETER ON THE CHIN
MULTI POINT MULTI DIMENSIONAL
ACCELEROMETER DATA LOGGING SYSTEM
MOBILE COMPUTER
USB CABLE

Figure 5.32 Three accelerometers connection to the jaw

EEG cap electrodes placements are shown in Figure 5.33. The jaw movements are expected to effect frontal and near ear areas, so FT7 and FT8 electrodes are selected to show with corresponding three channels three axes accelerometer data in Figure 5.34.

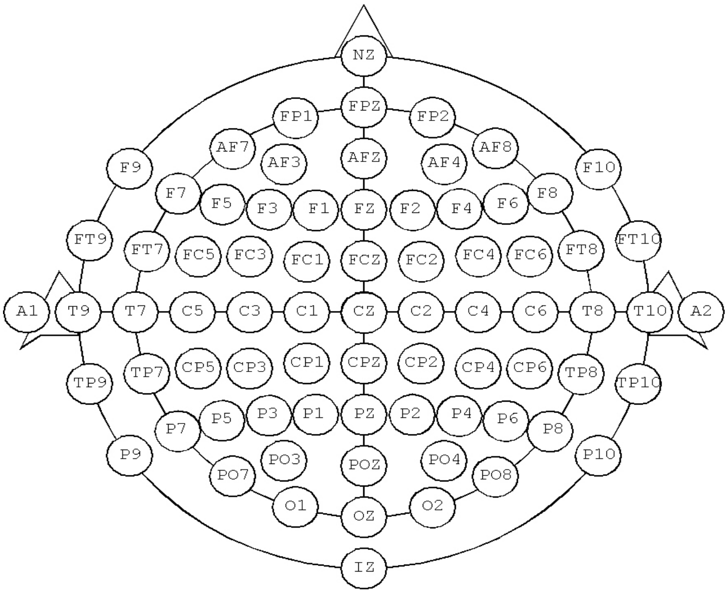


Figure 5.33 EEG electrodes placement on the scalp.

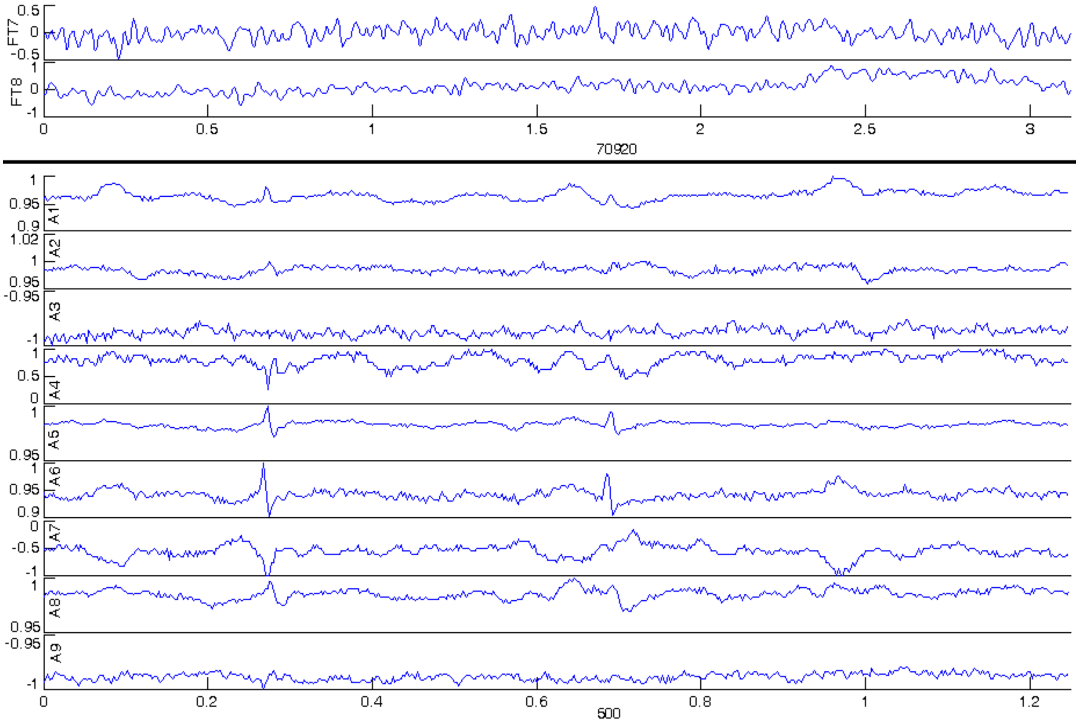


Figure 5.34 EEG FT7&FT8 channels and corresponding nine channels accelerometer data (obtained from DEU Biophysics Dept.).

CHAPTER SIX
CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

The current study has focused on Independent Component Analysis (ICA) with applications on biomedical signal processing. The current ICA approach was not only for finding components in a mixture of signals, but also used for signal filtering. Additionally, the study has been resulted in development of data logging systems.

In this thesis, mainly two related subjects based on ICA were studied; i) custom design of EEG artifact rejection, and ii) the design of sound and accelerometer data logging systems to diagnose knee related problems and jaw artifact rejection in EEG. EEG artifact rejection studies were concentrated on specific artifact problems as heart pulses and sweating artifacts filtering using ICA. Accelerometer data logging system is a multi point multi dimensional vibration data capturing device.

For the EEG studies, heart pulses were successfully detected and extracted from the EEG recordings using kurtosis ordering of ICA results. Because of the non-gaussian nature of EEG signals, ICA is very successful for extracting and removing the artifacts. Fast ICA algorithm doesn’t converge well, but using hyperbolic tangent nonlinearity a satisfactory result was obtained. Also, Efficient ICA (EFICA) and Maximum Likelihood ICA (ML ICA) algorithms run very well. Sweating artifact was cleaned using kurtosis ordering of ICA results.

Custom experiments for different conditions with various artifact contaminations can be designed and related data can be collected from subjects. These conditions can be related to different sweating, body movements, eye blinking and etc. These kinds of scenarios would give us a chance to understand the nature of artifacts. Thus, ICA algorithms can be tested and enhanced, and this may have a clear assesment of mentioned filtering or ICA methods in real life problems.

In this thesis work, the multi point multi dimensional accelerometer data logging system (MPMD ADLS) with its complete hardware and software units has been successfully realized for mainly biomedical applications. Originally, the designed system was developed for the needs of an expert system which supports medical diagnosis of knee joint problems non-invasively. However, the system can also be used for other human body joints in addition to the knee application. We also proposed a technique for recording of the jaw movements during the EEG recording to eliminate the jaw artifact more effectively. This part of study is still in progress and therefore preliminary EEG and jaw data were logged synchronously using EEG recording and MPMD ADLS systems. Moreover, the designed MPMD ADLS system is available for industrial applications where vibration is existent as part of movements.

For flexibility and mobility, the system was designed as a standalone modular system which has reduced chip count, low energy consumption, and high speed computer connection. The communication between MPMD ADLS and computer is performed either in real-time or offline by using the USB 2.0 connection standard. This provides high speed data transfer, system power supply and allows the realization of real-time operation. The designed data logging system has a flash SD card memory to store data up to 1GB capacity and performs data logging operations independently from the connected computer. This size of memory is large enough to hold up to one million persons' vibration data. Thus, MPMD ADLS device can hold data with their recording date and time stamps and anytime recorded vibration data can be transferred to a computer offline via USB connection.

When an analysis and classification algorithm for identifying normal and abnormal VAG signals is developed for the MPMD ADLS, the system becomes a standalone diagnosing system. However, this requires using more powerful microprocessor unit or units. The MPMD ADLS system has been successfully applied to the human knee to identify knee related problems. It has been observed that the recorded VAG signals show different statistical behavior for normal and abnormal knees. There are numerous knee related problems and this requires creating a large data base from different cases to obtain data features for developing a convenient classification algorithm.

In addition to thesis study, also accelerometer vibration signal extraction from tilt data as a kinematic signal detrending application was examined by using the above given filtering techniques and their performances compared to each other. The most appropriate filtering technique was found as running average.

6.2 Limitations

The current device and the study design lack a broad clinical database (knee data etc.). Furthermore, it is only applied to a limited number of EEG recording cases. During the thesis work, it has been observed that the analytical tools have their own limitations upon the different data application needs. This can be problematic for instance in selecting ICA algorithm, filter etc. Therefore we have tried to provide comparative analyses of different filter effects (Section 5.2.1). However, every new case might need a new set of algorithms with different parameters. Accordingly we suggest a careful approach to the specific designs.

6.3 Future Works

This thesis study can be the inspiration of extended future works and related master or doctorate thesis. Some of the possible future works are listed below.

- The designed MPMD ADLS device can be more compact, power efficient, and standalone. Using surface mount chips and double layered printed circuit board makes the device size smaller. When the device becomes standalone, the data recording can be performed without an additional mobile computer and power. The use of more powerful microcontrollers and may be FPGA’s can help to realize onboard classification and therefore diagnosing.

- According to medical experts a total of at least hundred VAG signals from normal and abnormal cases are to be collected to make a successful expert diagnosing system. To make further classification between abnormal cases, there is a need to specify various abnormal cases depending on corresponding knee problem. This may require additional VAG signals collection.

- For a reliable classification, each collected VAG signal must be labeled by a medical specialist by using X-ray, MRI, and arthroscopy results. ICA can be used for filtering undesired signal components and may be used to recover important signal sources. For an expert classification system based on various classification algorithms based on neural networks and fuzzy can be used.

- To clean jaw related artifacts from EEG recordings, MPMD ADLS and EEG recording devices must be synchronized. The system performance can be enhanced by increasing the number of synchronized recordings from various cases.

- In this thesis study, only sweating and heart artifacts were removed from EEG recordings by using ICA. However, there are some other cases including excessive body movements, twitching, blinking etc.

- In addition to biomedical applications, some industrial applications where vibration and movements are available can be good candidates for the designed MPMD ADLS system. Therefore, the designed device can help to develop new research studies in various disciplines.

Acknowledgement

The current thesis made use of biological data obtained from Dokuz Eylul University (DEU) Faculty of Medicine. The knee data acquisition was performed at Othopeadics and Traumatology Department which was approved by the Clinical and Laboratory Research Ethics Board of the DEU. Knee data were collected by the help of Berivan Çeçen, Bora Uzun, and the Orthopeadics and Traumatology department

staff. The EEG data, heart artifacts, sweating and jaw recordings were obtained from DEU Biophysics Dept., Brain Biophysics Lab (2006.KB.SAG.017, 2008.KB.SAG.019 and TUBITAK 108S113 projects have been partially utilized). The EEG data has been collected by the help of Onur Bayazıt and the Biophysics staff.

REFERENCES

Abdallah, S. A., & Plumbley, M.D. (2001). If the independent components of natural images are edges, what are the independent components of natural sounds? *Proceedings of the International Workshop on Independent Component Analysis and Blind Separation (ICA'01),* 534-539.

Akkan, T., & Senol, Y. (2008a). A low-cost multi-channel simultaneous sound capturing system for independent component analysis. *IEEE 16th Signal Processing, Communication and Applications Conference SIU 2008.* 1-4.

Akkan, T., & Senol, Y. (2008b). Capturing and Analysis of Knee-Joint Signals Using Acceleremoters. *IEEE 16th Signal Processing, Communication and Applications Conference SIU 2008.* 1-4.

Akkan, T., & Senol, Y. (2009). Biometrical Source Identification in Complex EEG Signals. In: M. Özgören, A. Öniz (Eds.), *The Applied Biophysics-Uygulamali Beyin Biyofizigi ve Multidisipliner Yaklasim* (195-200). Izmir, Turkey: Dokuz Eylul Yayinlari, D.E.U. Rektorluk Matbaasi.

Amari, S., Cichocki, A., & Yang, H. (1996). *A new learning algorithm for blind source separation*. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing 8 (Proc. NIPS'95)*, (757-763). Cambridge, MA: MIT Press.

Amari, S. (1998). Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2), 251-276.

Anthony, J.B., & Terrence, J.S. (1997). The 'independent components' of natural scenes are edge filters, *Vision Research*, 37(23), 3327-3338.

Azzerboni, B., Ipsale, M., Foresta, F.L., Mammone, N., & Morabito, F.C. (2006). *A Comparison of ICA Algorithms in Biomedical Signal Processing*, in: B. Apolloni, M. Marinaro, R. Tagliaferri, *Biological and Artificial Intelligence Environments*, (313-320). Springer, Dordrecht (NLD).

Bell, A., & Sejnowski, T. (1995). An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6), 1129-1159.

Boashash, B. (Ed). (2003). *Time frequency signal analysis and processing: a comprehensive reference*. Boston; Elsevier.

Buckwalter, J.A, & Einhorn, T.A, Simon S.R (Eds.). (2000). *Orthopaedic Basic Science* (2nd ed.). Rosemont, IL; American Academy of Orthopaedic Surgeons, 168-180.

Cardoso, J.F., Souloumiac,A. (1993). *Blind beamforming for non Gaussian signals. IEEE Proceedings F*, 140(6), 362-370.

Cardoso, J.F. (1999). High-order contrasts for independent component analysis, *Neural Computation*, 11(1),157-192

Cardoso, J.F., *(n.d.). ICA algorithms available via ICA Central.* Retrieved September 7, 2009, http://www.tsi.enst.fr/icacentral/algos.html

Comon, P. (1994). Independent component analysis - a new concept? *Signal Processing*, 36, 287-314.

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis, *Journal of Neuroscience Methods*, 134(1), 9-21.

Enghoff, S. (1999). Moving ICA and Time-Frequency Analysis in Event-Related EEG Studies of Selective Attention, Thesis, Technical University of Denmark.

Eskandari, H., Shamsollahi, M.B., Rahimi, A., Behzad, M., Afkari, P., & Zamani, E.A. (2003). Feature extraction from optimal time-frequency and time-scale transforms for the classification of the knee joint vibroarthrographic signals, *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology*, 709-712.

Everitt, B.S. (2006). *The Cambridge Dictionary of Statistics.* (3rd ed.) (313–314). Cambridge: Cambridge University Press.

Friedman, J.H., & Tukey, J.W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. of Computers*, c-23(9), 881–890.

Friedman, J. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397), 249–266.

Huber, P. (1985). Projection pursuit. *The Annals of Statistics*, 13(2), 435–475.

Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10(3), 626-634.

Hyvärinen, A. (1999). Survey on independent component analysis. *Neural Computing Surveys*, 2, 94-128.

Hyvärinen, A., & Oja, E. (2000). Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 13(4-5), 411-430.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001a): *Independent Component Analysis*, New York; Wiley.

Hyvärinen, A., Patrik O. H, & Mika, I, (2001b). Topographic Independent Component Analysis, Neural Computation, 13(7), 1527-1558.

Jones, M., & Sibson, R. (1987). What is projection pursuit? *Journal of the Royal Statistical Society, Ser. A*, 150, 1–36.

Jung, T.P, Makeig, S., Humphries, C., Lee, T.W., McKeown, M.J., Iragui, V., et al. (2000). Removing Electroencephalographic Artifacts by Blind Source Separation, *Psychophysiology*, 37, 163-178.

Jutten, C., & Herault, J. (1991). Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24, 1-10.

Kim, K.S., Seo, J.H., Kang, J.U., & Song, C.G. (2009). An enhanced algorithm for knee joint sound classification using feature extraction based on time-frequency analysis. *Computer Methods and Programs in Biomedicine*, 94(2), 198-206.

Koldovsky, Z., Tichavsky, P., & Oja, E. (2006). Efficient Variant of Algorithm FastICA for Independent Component Analysis Attaining the Cramer-Rao Lower Bound. *IEEE Transactions on Neural Networks,* 17(6), 1265-1277.

Krishnan, S., & Rangayyan, R.M. (1999). Denoising Knee Joint Vibration Signals Using Adaptive Time-Frequency Representations, *IEEE Conference on Electrical and Computer Engineering*, 3, 1495-1500.

Krishnan, S., & Rangayyan, R.M. (2000a). Automatic de-noising of knee-joint vibration signals using adaptive time-frequency representations. *Medical & Biological Engineering & Computing*, 38(1). 2-8.

Krishnan, S., Rangayyan, R.M., Bell, G.D., & Frank, C.B. (2000b). Adaptive Time-Frequency Analysis of Knee Joint Vibroarthrographic Signals for Noninvasive Screening of Articular Cartilage Pathology. *IEEE Transactions on Biomedical Engineering*, 47(6), 773-783.

Krishnan, S., Rangayyan, R.M., Bell, G.D., & Frank, C.B. (2001). Auditory Display of knee-joint vibration signals. *Acoustical Society of America Journal*, 110(6), 3292-3304.

Lee, T.W., Bell, A.J., & Orglmeister, R. (1997a). Blind source separation of real world signals, *International Conference on Neural Networks*, 4, 2129-2134.

Lee, T.W., Bell, A., & Lambert, R. (1997b). Blind Separation of Delayed and Convolved Sources, In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), *Advances in Neural Information Processing Systems 9*, (758-764). Cambridge MA: MIT Press.

Lee T.W., Ziehe A., Orglmeister R,, & Sejnowski T. (1998). Combining time-Delay Decorrelation and ICA: Towards solving the cocktail party problem. *ICASSP-98*, 1249-1252.

Lee, T.W., Lewicki, M.S., Girolami, M., & Sejnowski, T.J., (1999a). Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Subgaussian and Supergaussian Sources, *Neural Computation*, 11 (2), 417-441.

Lee, T.W., Lewicki, M.S., Girolami, M., & Sejnowski, T.J., (1999b). Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Signal Processing Letters*. 6 (4), 87-90.

Lewicki, M.S., & Sejnowski, T.J. (1998). Learning nonlinear overcomplete representations for efficient coding. *Advances in Neural and Information Processing Systems*, 10,815-821.

Ludwig, A. (April 22, 2009). *Art Ludwig's Sound Page, Music and The Human Ear*. Retrieved September 12, 2009, from http://www.silcom.com/~aludwig/

Microchip Inc.,(2009). *Microchip PIC18F4550 Data Sheet*. Retrieved September 7, 2009, from ww1.microchip.com/downloads/en/ DeviceDoc/39632b.pdf

Nunez, P.L., Srinivasan, R., Westdorp, A.F., Wijesinghe, R.S., Tucker, D.M., & Silberstein R.B., et al. (1997). EEG coherency. I: Statistics, reference electrode, volume conduction, Laplacians, cortical imaging, and interpretation at multiple scales. *Electroencephalography and clinical neurophysiology*. 103(5), 499-515.

Nunez, P. L., Silberstein, R. B., Shi, Z., Carpenter, M. R., Srinivasan, R., & Tucker, D. M., et al. (1999). EEG coherency II: experimental comparisons of multiple measures. *Clinical Neurophysiology*, 110(3), 469-486.

Parra, L., & Spence, C. (2000). Convolutive blind source seperation of non stationary sources, *IEEE Transactions Speech and Audio Processing*, 8(3), 320-327.

Payne, R.S. (1962). How the Barn Owl Locates Prey by Hearing. The Living Bird, *First Annual of the Cornell Laboratory of Ornithology*, 151-159.

Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine* **2,** 6, 559–572.

Pedersen, M.S., Wang, D.L., Larsen, J., & Kjems, U. (2005). Overcomplete Blind Source Separation By Combining Time-Frequency Masking. *IEEE International workshop on Machine Learning for Signal Processing*. 15-20.

Pedersen M.S, Larsen,J., Kjems,U., & Parra L.C. (2007), A survey of Convolutive Blind Source Separation Methods. , J. Benesty, and A. Huang (Eds.), *Springer Handbook on Speech Processing and Speech Communication*. (1-34). Springer.

Polikar, R., (May 30, 2006). *The Wavelet tutorial part 2 – Short Time Fourier Transform.* Retrieved September 12, 2009, from
http://ccrma-www.stanford.edu/ ~unjung/mylec/WTpart2.html

Reddy, A.M., & Raj, B. (2004). Soft mask estimation for single channel speaker separation, *Workshop on Statistical and Perceptual Audio Processing SAPA-2004*, Paper 158.

Romero, S., Mailanas, M.A, Clos,S., Gimenez, S., & Barbanoj, M.J. (2003). Reduction of EEG Artifacts by ICA in Different Sleep Stages, *Proceedings of the 25th Annual International Conference of the IEEE*, (3), 2675-2678.

Sawada H.,Araki S., Mukai, R., & Makino, S. (2006). Blind Extraction of Dominant Target Sources Using ICA and Time-Frequency Masking. *IEEE Transactions on Audio, Speech & Language Processing,* 14(6), 2165-2173.

Selim, R.B., & Diego, R. (2008). EEG Artifacts. Electroencephalography Atlas, Retrieved September 12, 2009; from http://emedicine.medscape.com/article/ 1140247-overview

Siddiqui, F., Osuna, E., Walters, A.S., & Chokroverty, S. (2006). Sweat artifact and respiratory artifact occurring simultaneously in polysomnogram, *Sleep Medicine*, 7(22), 197-199.

Smaragdis, P. (1997). Efficient Blind Separation of Convolved Sound Mixtures, *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*. 19-22.

Srinivasan, S., Roman, N., & Wang, D.L. (2004). On Binary and Ratio Time-Frequency Masks for Robust Speech Recognition, *ICLSP*, 2541-2544.

ST Microelectronics, (2009). LIS3LV02DQ MEMS Inertial Sensor. Retrieved September 7, 2009; from www.st.com/stonline/products/literature/ds/11115/ lis3lv02dq.pdf

Tavatha, S., Rangayyan, R.M., Frank, C.B., Bell, G.D., Ladly, K.O, & Zhang, Y.T. (1992). Analysis of Knee Vibration Signals Using Linear Prediction, *IEEE Trans. Biomed. Eng*., 39, 959-970.

Teplan M., Krakovska A., & Stolc,S. (2006). EEG responses to long-term audio-visual stimulation, *International Journal of Psychophysiology*, 59(2), 81-90.

Tong, L., Soon, V., Huang, Y.F. & Liu, R. (1991), Indeterminacy and identifiability of blind identification, *IEEE Trans. CAS*, 38, 499-509.

Torkkola K. (1996). Blind separation of convolved sources based on information maximization, *Proc. IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing VI*, 423-432.

University of Oxford Robotic Research Group, (March 23, 2009). *Brain Computer Interfacing Project.* Retrieved September 4, 2009, from http://www.robots.ox.ac.uk/~parg/projects/bci/rev1.html

Wolpaw, J.R. (2007). Brain-computer interfaces as new brain output pathways. *Journal of Physiology*, 579, 613-619.

Xue, Z., Li, J., Li, S., & Wan, B. (2006). Using ICA to Remove Eye Blink and Power Line Artifacts in EEG, *Proceedings of the First International Conference on Innovative Computing, Information and Control*, 3,107-110.

Zhang, Y.T., Rolston, A., Rangayyan R.M, Frank, C.B., & Bell, G.D. (1992). Wavelet Transform Analysis of Vibroathrographic (VAG) signals obtained during dynamic knee movement", *Proc. IEEE-SP International Symposium on Time-frequency and Time-scale Analysis*, 235-238.

APPENDIX

Appendix A Important MATLAB Codes

A.1 pcaicademo4.m (Figure 2.4)

```
% Fig 2.1. PCA ICA comparison
clear all; close all; cizkal=2;boy=4000; kacbin=110; nbin=kacbin; fontsayz=11; bas=-1; son=1;
hbas=-1; hson=1; hara=0.05;
%-------------------------------------------- SIGNAL LOAD
load chirp; a=y'; a=a(1:boy)*1;  x(1,:)=a*1; load train; a=y'; a=a(1:boy)*1;  x(2,:)=a*1.5;

gau=random('Normal',bas,son,2,boy);
subgau=random('unif',bas,son,2,boy);
mu=0; sigma=1; m=2; n=boy; %[m, n]  : the dimension of y.
u = rand(m, n)-0.5; b = sigma / sqrt(2);
supgau= mu - b * sign(u).* log(1- 2* abs(u));  % Generate Laplacian random  mu:mean, sigma:std dev
%x(1,:)=supgau(1,:); %x(2,:)=gau(1,:);

mix(1,:)=x(1,:)*0.80+x(2,:)*0.2; mix(2,:) =x(1,:)*0.4+x(2,:)*0.6;

h=figure(2)
subplot(2,8,1:2); plot(x(1,:),x(2,:),'.'); title('Original sources');
[pc,score,latent,tsquare] = princomp(x');
aci1=atan2(pc(2,1),pc(1,1))*180/pi; aci2=atan2(pc(2,2),pc(1,2))*180/pi;
xmin=min(x(1,:)); xmax=max(x(1,:)); ymin=min(x(2,:)); ymax=max(x(2,:));  hold on;
  plot([xmin xmax]*pc(1,1),[ymin ymax]*pc(2,1),'r-','LineWidth',cizkal);
  plot([xmin xmax]*pc(1,2),[ymin ymax]*pc(2,2),'g-','LineWidth',cizkal);
  xlabel(['\theta1=' num2str(aci1,'%3.2f') ' \theta2=' num2str(aci2,'%3.2f')]);  hold off
%-------------------------------------
h2=subplot(2,8,3:4); plot(mix(1,:),mix(2,:),'.'); title('Mixtures');
 [pc,score,latent,tsquare] = princomp(mix');
aci1=atan2(pc(2,1),pc(1,1))*180/pi; aci2=atan2(pc(2,2),pc(1,2))*180/pi;
xmin=min(mix(1,:)); xmax=max(mix(1,:)); ymin=min(mix(2,:)); ymax=max(mix(2,:));
xlabel(['\theta1=' num2str(aci1,'%3.2f') ' \theta2=' num2str(aci2,'%3.2f')]);     ylim ([-1 1.1]);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PCA ALGOS
 [pc,score,latent,tsquare] = princomp(mix');z = (x'*pc)';  % Rotate the data to the PC's

subplot(2,8,5:6); plot(z(1,:),z(2,:),'.'); title('PCA result');
[pc,score,latent,tsquare] = princomp(z');
aci1=atan2(pc(2,1),pc(1,1))*180/pi; aci2=atan2(pc(2,2),pc(1,2))*180/pi;
xmin=min(z(1,:))*0.7; xmax=max(z(1,:))*0.7; ymin=min(z(2,:))*0.7; ymax=max(z(2,:))*0.7;
hold on;  plot([xmin xmax]*pc(1,1),[ymin ymax]*pc(2,1),'r-','LineWidth',cizkal);
  plot([xmin xmax]*pc(1,2),[ymin ymax]*pc(2,2),'g-','LineWidth',cizkal);
  xlabel(['\theta1=' num2str(aci1,'%3.2f') ' \theta2=' num2str(aci2,'%3.2f')]);hold off; xlim ([-1.4 1.4]);
 [ICs,W1] = icaMLson(mix,2);
subplot(2,8,7:8); plot(ICs(2,:),ICs(1,:),'.');

  [pc,score,latent,tsquare] = princomp(ICs');
aci1=atan2(pc(2,1),pc(1,1))*180/pi;aci2=atan2(pc(2,2),pc(1,2))*180/pi;
xmin=min(ICs(1,:))*1; xmax=max(ICs(1,:))*1;ymin=min(ICs(2,:))*0.7; ymax=max(ICs(2,:))*0.7;
  hold on;  plot([xmin xmax]*pc(1,1),[ymin ymax]*pc(2,1),'r-','LineWidth',cizkal);
  plot([xmin xmax]*pc(1,2),[ymin ymax]*pc(2,2),'g-','LineWidth',cizkal);
  xlabel(['\theta1=' num2str(aci1,'%3.2f') ' \theta2=' num2str(aci2,'%3.2f')]); hold off; xlim ([-5.2 5.2]);
title('ICA result');xlabel(['\theta1=' num2str(aci1,'%3.2f') ' \theta2=' num2str(aci2,'%3.2f')])
fno=['-f' num2str(h)];dizin='D:\taner kisisel\makaleler\tezim 2008\tezsekil\';  %fig write folder
fname=[dizin 'tfli.tiff']; print('-dtiffn',fno,'-r300',fname); crop( [dizin  'tfli.tiff']);
```

A.2 showgauss.m (Figure 2.3)

```
  boy=1000;kacbin=110;bas=-1;son=1;x=bas:2/boy:(son-2/boy);
gau=random('Normal',bas,son,boy,2);  subgau=random('unif',bas,son,boy,2);
% Generate Laplacian random values mu:mean, sigma: std dev.
mu=0; sigma=1; m=boy; n=2; %[m, n]  : the dimension of y.
u = rand(m, n)-0.5; b = sigma / sqrt(2);supgau= mu - b * sign(u).* log(1- 2* abs(u));

hh=figure(1), ss=subgau(:,1); subplot(2,3,1); [a1,b1]=hist(ss,kacbin); a1=[0 0 a1 0 0];
b1=[-2 -1 b1 1 2]; plot(b1,a1/sum(a1),'r');  S='Uniform';  h=title(S);  set(h,'fontsize',12) ;
subplot(2,3,4); plot(subgau(:,1),subgau(:,2),'.'); h=xlabel(' (a) SUB-GAUSSIAN'); set(h,'fontsize',12)

ss=gau(:,1);subplot(2,3,2); [a1,b1]=hist(ss,kacbin);  plot(b1,a1/sum(a1),'r');
S='Normal';  h=title(S);  set(h,'fontsize',12)
subplot(2,3,5); plot(gau(:,2),gau(:,1),'.'); h=xlabel(' (b) GAUSSIAN'); set(h,'fontsize',12)

ss=supgau(:,1); subplot(2,3,3); [a1,b1]=hist(supgau(:,1),kacbin);  plot(b1,a1/sum(a1),'r');
S='Laplacian';  h=title(S);  set(h,'fontsize',12) ;subplot(2,3,6); plot(supgau(:,1),supgau(:,2),'.');
h=xlabel(' (c) SUPER-GAUSSIAN'); set(h,'fontsize',12)
     fno=['-f' num2str(hh)]; dizin='D:\taner kisisel\makaleler\tezim 2008\tezsekil\';  %fig yazma dizini
     fname=[dizin 'fig2_6.tiff']; print('-dtiffn',fno,'-r300',fname); crop( [dizin  'fig2_6.tiff']);
```

A.3 FAST ICA code.

```
% only for two sources recovering    16-3-2005
function[y,W,V,iteration]=FICA(x,whpr); type=1; %pow3 ile
%%%%%%%%%%%%%%%%%%%%%%%%%%%% WHITENING increases the correlatedness.
if whpr==1, white_x=x'; expect_x=mean(white_x); % making mean zero
white_x(:,1)=white_x(:,1)-expect_x(1,1); white_x(:,2)=white_x(:,2)-expect_x(1,2);
size(white_x); Cx = cov(white_x,1);    % whitening
[E,D]=eig(Cx);  % eigenvalue decomposition of covariance matrix, finding eigenvectors of Cx
                % here E=matrix of eigenvectors, D=diagonal matrix of eigenvectors
D(1,1)=D(1,1)^(-0.5); D(2,2)=D(2,2)^(-0.5);   % Finding D^(-0.5)
V=E*D*E' ; z=white_x*V; % forming V and whitening data
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if whpr==2, z=x'; V=y; end % No whitening
  z=z';   eps=0.0001; m=2;p=1  %epsilon,  the number of ICS to recover, set counter =1 , p<=m

  W=zeros(2,2); temp=zeros(2,1);
   for p=1:m,     W(:,p)=rand(2,1); % Wp randomly initalized
     W(:,p)=W(:,p) / norm(W(:,p)); exit=0;  count=0;  iter=1; fprintf('Recovering number %i IC \n',p);
   while exit==0,    count=count+1; temp=W(:,p);
      switch (type)   % 1:pow , 2:tanh, 3:gauss
         case 1  W(1,p)=mean(z(1,:).*(((temp)'*z).^3)) - 3*temp(1,1);
                 W(2,p)=mean(z(2,:).*(((temp)'*z).^3)) - 3*temp(2,1);
         case 2  W(1,p)=mean(z(1,:).*(tanh((temp)'*z)))-(mean(1-(tanh((temp))'*z).^2)).*temp(1,1);
                 W(2,p)=mean(z(2,:).*(tanh((temp)'*z)))-(mean(1-(tanh((temp))'*z).^2)).*temp(2,1);
         case 3 %gauss
         otherwise  disp('Invalid nonlinearity');
      end
iteration(p)=iter;  sum=zeros(2,1);
for counter=1:p-1, sum=sum+(W(:,p)'*W(:,counter))*W(:,counter); end;
  W(:,p)=W(:,p)-sum;  W(:,p)=W(:,p)/norm(W(:,p));
 if(abs((dot(W(:,p),temp))) < 1+eps) & (abs((dot(W(:,p),temp))) > 1-eps),   exit=1; end;
iter=iter+1;end; end;  W=W'; y=W*z;  % the recoverd signal is y
```

A.4 dortkanal.m

```
% generate separate left channel and right channel sound data
clear all; type=0; boy=16384; clear y31; clear y32; clear y33;clear y34; clear y4; clear y5;

if (type==0), j1=-0.9; j2=-0.6; j3=-0.3; j4=0; f=11025; % f = out freq.

for i=1:boy, y31(i)=sin(2*(i+0)*pi/180);  y32(i)=sin(2*(i+30)*pi/180);
                 y33(i)=sin(2*(i+60)*pi/180); y34(i)=sin(2*(i+90)*pi/180);

if (j1<=0) t1=0.9; end;    if (j1>0) t1=-0.9; end ;  if (j2<=0) t2=0.9; end;    if (j2>0) t2=-0.9; end

if (j3<=0) t3=0.9; end;    if (j3>0) t3=-0.9; end;  if (j4<=0) t4=0.9; end;    if (j4>0) t4=-0.9; end
y51(i)=t1; y52(i)=t2; y53(i)=t3; y54(i)=t4;

if (j1>=0.9), j1=-0.9; end  if (j2>=0.9), j2=-0.9; end;

if (j3>=0.9), j3=-0.9; end  if (j4>=0.9), j4=-0.9; end

j1=j1+0.0025; j2=j2+0.0025; j3=j3+0.0025; j4=j4+0.0025;  y41(i)=j1; y42(i)=j2; y43(i)=j3; y44(i)=j4;
end; end

 load gong; y1=y(1:boy,:);
 if (type==1), load chirp;    y1=y;  load gong; y2=y;
a=max(size(y1)); b=max(size(y2)); c=min(a,b); y3=y1(1:c,:); y4=y2(1:c,:); end

z1=[y31;y41]';
figure(1); subplot(4,1,1); plot(y31,'r'); subplot(4,1,2); plot(y32,'r');
               subplot(4,1,3); plot(y33,'r'); subplot(4,1,4); plot(y34,'r');

figure(2); z=y31(1:1000);subplot(4,1,1);plot(z,'r'); z=y32(1:1000);subplot(4,1,2);plot(z,'r');
               z=y33(1:1000);subplot(4,1,3);plot(z,'r'); z=y34(1:1000);subplot(4,1,4);plot(z,'r');

figure(3); subplot(4,1,1); plot(y41,'r'); subplot(4,1,2); plot(y42,'r');
               subplot(4,1,3); plot(y43,'r'); subplot(4,1,4); plot(y44,'r');

figure(4); z=y41(1:1000);subplot(4,1,1);plot(z,'r'); z=y42(1:1000);subplot(4,1,2);plot(z,'r');
                z=y43(1:1000);subplot(4,1,3);plot(z,'r'); z=y44(1:1000);subplot(4,1,4);plot(z,'r');

figure(5);  subplot(4,1,1); plot(y51,'r'); subplot(4,1,2); plot(y52,'r');
                subplot(4,1,3); plot(y53,'r'); subplot(4,1,4); plot(y54,'r');

figure(6); z=y51(1:1000);subplot(4,1,1);plot(z,'r'); z=y52(1:1000);subplot(4,1,2);plot(z,'r');
               z=y53(1:1000);subplot(4,1,3);plot(z,'r'); z=y54(1:1000);subplot(4,1,4);plot(z,'r');

figure(7); subplot(1,1,1); plot(y1,'b');

wavwrite(z1,f,16,'dene1.wav'); wavwrite(y31,f,16,'sin1.wav'); wavwrite(y32,f,16,'sin2.wav');

wavwrite(y33,f,16,'sin3.wav'); wavwrite(y34,f,16,'sin4.wav'); wavwrite(y41,f,16,'tri1.wav');

wavwrite(y42,f,16,'tri2.wav'); wavwrite(y43,f,16,'tri3.wav'); wavwrite(y44,f,16,'tri4.wav');

wavwrite(y51,f,16,'squ1.wav'); wavwrite(y52,f,16,'squ2.wav'); wavwrite(y53,f,16,'squ3.wav');

wavwrite(y54,f,16,'squ4.wav'); wavwrite(y1,f,16,'chi1.wav');wavwrite(y2,f,16,'gon1.wav');
```

A.5 mixingfour.m

```
% dort farklı fazda sinus , kare ve üçgeni karıştırarak ica elde eden program
close all;  set=1; faz=1; sinusler=1;
autogen=0;   % 1 = generate signals here    % 0 = use signals generated using dortkanal.m

if (sinusler==1), N=11025; Fs=11025;
t=linspace(0,10.2*pi,N); k1=sin(t)'+sin(3*t+0*pi/18)'+0*randgauss(1,5,N)'; % sin1
t=linspace(0,45.1*pi,N); k2=sin(0.07*t+0*pi/18)'+0*randgauss(1,5,N)';  % sin2
t=linspace(0,65.3*pi,N); k3=0.9*square(1.8*t)';   % sin3
t=linspace(0,77.5*pi,N); k4=0*sin(t+0*pi/6)'+randgauss(1,5,N)';   % sin4

subplot(4,1,1); specgram(k1,512,Fs,kaiser(500,5),475);
subplot(4,1,2); specgram(k2,512,Fs,kaiser(500,5),475);
subplot(4,1,3); specgram(k3,512,Fs,kaiser(500,5),475);
subplot(4,1,4); specgram(k4,512,Fs,kaiser(500,5),475); k11=k1; k21=k2; k31=k3; k41=k4;
end

if (autogen==1) ,  delay=pi/90;  td=0.000001;   % sin, saw,square delay (phase) &chirp time delay
t=linspace(0,10,10000); amp=[8 8 8 8;  8 8 8 8;   8 8 8 8;  8 8 8 8];
freq  =[3 3 3 3;  5 5 5 5;   8 8 8 8; 11 11 11 11];

k11=amp(1,1)*sin(2*pi*freq(1,1)*(t+td*0)); k12=amp(1,2)*sin(2*pi*freq(1,2)*(t+td*1));
k13=amp(1,3)*sin(2*pi*freq(1,3)*(t+td*2)); k14=amp(1,4)*sin(2*pi*freq(1,4)*(t+td*3));
k21=amp(2,1)*sawtooth(2*pi*freq(2,1)*(t+td*0)); k22=amp(2,2)*sawtooth(2*pi*freq(2,2)*(t+td*1));
k23=amp(2,3)*sawtooth(2*pi*freq(2,3)*(t+td*2)); k24=amp(2,4)*sawtooth(2*pi*freq(2,4)*(t+td*3));
k31=amp(3,1)*square(2*pi*freq(3,1)*(t+td*0));  k32=amp(3,2)*square(2*pi*freq(3,2)*(t+td*1));
k33=amp(3,3)*square(2*pi*freq(3,3)*(t+td*2));  k34=amp(3,4)*square(2*pi*freq(3,4)*(t+td*3));

fo=25;f1=100;     tip='convex';          % Start at 25Hz, go up to 100Hz
k41=amp(4,1)*chirp(t+(td*0),fo,1,f1,'q',[],tip); k42=amp(4,2)*chirp(t+(td*1),fo,1,f1,'q',[], tip);
k43=amp(4,3)*chirp(t+(td*2),fo,1,f1,'q',[], tip); k44=amp(4,4)*chirp(t+(td*3),fo,1,f1,'q',[], tip);
end

if (autogen==0)
if (set==0),  boy=32768;  % data size
file1='ch1.wav'; file2='ch2.wav'; file3='ch3.wav'; file4='ch4.wav'; % microphones signals
Filesize_k1=wavread(file1,'size'); Filesize_k2=wavread(file2,'size');
Filesize_k3=wavread(file3,'size');  Filesize_k4=wavread(file4,'size');
[k1,Fsk1,bitsk11]=wavread(file1,boy); [k2,Fsk2,bitsk12]=wavread(file2,boy);
[k3,Fsk3,bitsk13]=wavread(file3,boy); [k4,Fsk4,bitsk14]=wavread(file4,boy);
end

if (set==1),  boy=16384; % sinus,square,saw,chirp sounds
[k11,Fsk11,bitsk11]=wavread('sin1.wav',boy); [k12,Fsk12,bitsk12]=wavread('sin2.wav',boy);
[k13,Fsk13,bitsk13]=wavread('sin3.wav',boy); [k14,Fsk14,bitsk14]=wavread('sin4.wav',boy);
[k21,Fsk21,bitsk21]=wavread('tri1.wav',boy); [k22,Fsk22,bitsk22]=wavread('tri2.wav',boy);
[k23,Fsk23,bitsk23]=wavread('tri3.wav',boy); [k24,Fsk24,bitsk24]=wavread('tri4.wav',boy);
[k31,Fsk31,bitsk31]=wavread('squ1.wav',boy); [k32,Fsk32,bitsk32]=wavread('squ2.wav',boy);
[k33,Fsk33,bitsk33]=wavread('squ3.wav',boy); [k34,Fsk34,bitsk34]=wavread('squ4.wav',boy);
[k41,Fsk41,bitsk41]=wavread('chi1.wav',boy); k42=k41; k43=k41; k44=k41;
end
end

a = figure('Name', 'ORIGINAL SIGNALS');
subplot(4,1,1); plot(k11); subplot(4,1,2); plot(k21);subplot(4,1,3); plot(k31); subplot(4,1,4); plot(k41);

if (set==0), z1=0.5*k1+0.3*k2+0.1*k3+0.1*k4; z2=0.1*k1+0.9*k2-0.3*k3+0.2*k4;
```

```
z3=0.2*k1+0.1*k2-0.8*k3-0.5*k4; z4=0.8*k1+0.5*k2+0.2*k3-0.4*k4; end
if (set==1),
if(faz==0),z1=0.5*k11+0.3*k21+0.6*k31+0.1*k41;z2=0.5*k11+0.9*k21+0.3*k31+0.2*k41;
z3=0.2*k11+0.3*k21+0.8*k31+0.5*k41; z4=0.8*k11+0.7*k21+0.2*k31+0.4*k41;end
if(faz==1),z1=0.5*k11+0.3*k21+0.6*k31+0.1*k41;z2=0.5*k12+0.9*k22+0.3*k32+0.2*k41;
z3=0.2*k13+0.3*k23+0.8*k33+0.5*k41; z4=0.8*k14+0.7*k24+0.2*k34+0.4*k41;end
end % autogen burada bitiyor.

if (set==1 & faz==1),
 a = figure('Name', 'k11 k12 k13 k14 delayed');  subplot(4,1,1); plot(k11);  subplot(4,1,2); plot(k12);
 subplot(4,1,3); plot(k13);  subplot(4,1,4); plot(k14);
 a = figure('Name', 'k21 k22 k23 k24 delayed'); subplot(4,1,1); plot(k21);  subplot(4,1,2); plot(k22);
      subplot(4,1,3); plot(k23);  subplot(4,1,4); plot(k24);
a = figure('Name', 'k31 k32 k33 k34 delayed');  subplot(4,1,1); plot(k31);  subplot(4,1,2); plot(k32);
     subplot(4,1,3); plot(k33);  subplot(4,1,4); plot(k34);
a = figure('Name', 'k41 k42 k43 k44 delayed'); subplot(4,1,1); plot(k41);  subplot(4,1,2); plot(k42);
     subplot(4,1,3); plot(k43);  subplot(4,1,4); plot(k44);
end
a = figure('Name', 'MIXED SIGNALS'); zz1=z1; zz2=z2; zz3=z3; zz4=z4;
subplot(4,1,1); plot(zz1); subplot(4,1,2); plot(zz2); subplot(4,1,3); plot(zz3);subplot(4,1,4); plot(zz4);

if (autogen==0), if (set==0),
fprintf('ch1 data sampling  rate = %i \n',Fsk1); fprintf('sampling rate of ch2 data = %i \n',Fsk2);
fprintf('sampling rate of ch3 data = %i \n',Fsk3); fprintf('sampling rate of ch4 data = %i \n',Fsk4);end
if (set==1),fprintf('sampling rate of ch1 data = %i \n',Fsk11);
fprintf('sampling rate of ch2 data = %i \n',Fsk21); fprintf('sampling rate of ch3 data = %i \n',Fsk31);
fprintf('sampling rate of ch4 data = %i \n',Fsk41);end
end

if (autogen==0), [mixed]= [z1';z2';z3';z4']; end;   if (autogen==1), [mixed]= [z1;z2;z3;z4]; end
nICA=4;  % number of  ICA components
[rec1]=myFastICA(mixed,nICA);[rec2,A]=icaMLson(mixed,nICA,2);
B=MatlabjadeR(mixed); rec4=B*mixed; B=MatlabshibbsR(mixed); rec5=B*mixed;
a = figure('Name', 'FASTICA (TANER)'); for p=1:nICA, subplot(nICA,1,p); plot(rec1(p,:),'g'); end
a = figure('Name', 'ICA ML SON'); for p=1:nICA, subplot(nICA,1,p); plot(rec2(p,:),'g'); end
a = figure('Name', 'MATLABJADER'); for p=1:nICA, subplot(nICA,1,p); plot(rec4(p,:),'g'); end
a = figure('Name', 'MATLABSHIBBSR'); for p=1:nICA, subplot(nICA,1,p); plot(rec5(p,:),'g'); end

for p=1:nICA, ica_filename=strcat('ICA',int2str(p),'.wav');
wavwrite(recovered1(p,:),Fsk11,16,ica_filename);end
```

A.6 mixing_delayed.m

```
clear all; close all;
set=1;

   if (set==1),t=linspace(1,100,1000);k1=10*sin(0.1*pi*t);k2=10*sawtooth(t);fsize=size(k1);
fsize=fsize(2); end
    if (set==2),file1='sin1.wav';file2='squ1.wav';boy=16384;
Filesize_k1=wavread(file1,'size');Filesize_k2=wavread(file2,'size');
fsize=Filesize_k1(1)/32; [k1,Fsk1,bitsk1]=wavread(file1,boy);[k2,Fsk2,bitsk2]=wavread(file2,boy);
fprintf('sampling rate of ch1 data = %i \n',Fsk1); fprintf('sampling rate of ch2 data = %i \n',Fsk2);
end
   Num_ICA=2; A=[1 0.4;   0.8 1]; d12=50;  d21=50;   % A matrix, sample delays
tot_delay=d12+d21;dif_delay=abs(d21-d12);
z1=A(1,1)*k1(d12:fsize-dif_delay)+A(1,2)*k2(1:fsize-d12-dif_delay+1);
z2=A(2,2)*k2(d21:fsize)+A(2,1)*k1(1:fsize-d12-dif_delay+1);
```

```
mixed= [z1 ; z2]; [recovered]=MyFastICA(mixed,2);

a=figure('Name', 'ORIGINAL SIGNALS');subplot(2,1,1); plot(k1); subplot(2,1,2); plot(k2);
a=figure('Name', 'MIXED SIGS - NO DELAY');subplot(2,1,1); plot(z1); subplot(2,1,2); plot(z2);
a=figure('Name', 'ORIGINAL SIG PDFS&JOINT DENSITY');
alt=-10; ust=10;ara=.25; [a1,b1]=hist(k1,alt:ara:ust);[a2,b2]=hist(k2,alt:ara:ust);
subplot(3,1,1); plot(b1,a1/.1/sum(a1)); ylabel('1.pdf');
subplot(3,1,2); plot(b2,a2/.1/sum(a2)); ylabel('2.pdf');
subplot(3,1,3); plotmatrix(k1',k2'); ylabel('joint density')
a=figure('Name', 'MIXTURE SIG PDFS & JOINT DENSITY');
alt=-10; ust=10;ara=.25;[a1,b1]=hist(z1,alt:ara:ust);[a2,b2]=hist(z2,alt:ara:ust);
subplot(3,1,1); plot(b1,a1/.1/sum(a1)); ylabel('1.pdf');
subplot(3,1,2); plot(b2,a2/.1/sum(a2)); ylabel('2.pdf')
subplot(3,1,3); plotmatrix(z1',z2'); ylabel('joint density')
a = figure('Name', 'FAST ICA (TANER)');
for p=1:Num_ICA, subplot(Num_ICA,1,p); plot(recovered(p,:),'g'); end

clear z1; clear z2; clear mixed;a=figure('Name', 'FAST ICA SIG PDFS&JOINT DENSITY');
rc1=recovered(1,:)';rc2=recovered(2,:)';
alt=-10; ust=10;ara=.25; [a1,b1]=hist(rc1,alt:ara:ust);[a2,b2]=hist(rc2,alt:ara:ust);
subplot(3,1,1); plot(b1,a1/.1/sum(a1)); ylabel('1.pdf');
subplot(3,1,2); plot(b2,a2/.1/sum(a2)); ylabel('2.pdf');
subplot(3,1,3); plotmatrix(rc1,rc2); ylabel('joint density')
```

A.7 delayexample.m

```
s1  = [0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0]; s1  = s1  *  1; % delay example
s1d = [0 0 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0]; s1d = s1d *  1;
 s2  = [0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0]; s2  = s2  * -1;
s2d = [0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0]; s2d = s2d * -1; miny=-1.1; maxy=1.1;
figure(1), title('delay example');subplot(2,8,1:2); plot(s1,'r');
axis([0 length(s1)  miny  maxy]); xlabel('s1(t)');
subplot(2,8,9:10); plot(s2,'b');axis([0 length(s1)  miny  maxy]);xlabel('s2(t)');
a=[ 1   0.5;   %a11=1 original signal 1 passes
    0.5 1];    %a22=1 original signal 2 passes

subplot(2,8,4:5); plot(a(1,1)*s1,'r'); hold; plot(a(1,2)*s2d,'b');
axis([0 length(s1)  miny  maxy]); xlabel('s1(t-0), s2(t-d12)');
 subplot(2,8,12:13); plot(a(2,1)*s1d,'r'); hold; plot(a(2,2)*s2,'b');
axis([0 length(s1)  miny  maxy]); xlabel('s1(t-d21), s2(t-0)');

x1=a(1,1)*s1+a(1,2)*s2d;x2=a(2,1)*s1d+a(2,2)*s2;
subplot(2,8,7:8); plot(x1,'k'); axis([0 length(s1)  miny  maxy]); xlabel('x1=a11*s1(t)+a12*s2(t-d12)');
subplot(2,8,15:16); plot(x2,'k');axis([0 length(s1)  miny  maxy]);
xlabel('x2=a21*s1(t-d21)+a22*s2(t)');
```

A.8 delaymix.m

```
clear all;  secim=0; boy=10000;

if(secim==0), load chirp; s1=y; Fs_s1=Fs;load train; s2=y; Fs_s2=Fs; end
if (secim==1), [s1,Fsk1,bitsk1]=wavread(file1,boy); rate=Fsk1;

[s2,Fsk2,bitsk2]=wavread(file2,boy);end

figure(1), subplot(2,1,1); plot(s1(1:boy));title('chirp signal');
subplot(2,1,2); plot(s2(1:boy)); title('train signal');
```

```
 a=[1 0.4;   1 0.8];        %mixing matrix
d=[0 20;     30 0];        % delay matrice

d12_time=d(1,2)/Fs_s1; sprintf('chirp delay time= %0.5f',d12_time)
d21_time=d(2,1)/Fs_s2; sprintf('train delay time= %0.5f',d21_time)

x1n(1:boy)=a(1,1)*s1(1:boy)+a(1,2)*s2(1:boy);
x2n(1:boy)=a(2,1)*s2(1:boy)+a(2,2)*s1(1:boy);
mixed_n=[x1n' x2n']'; Num_ICA_n=2;

x1d(d(1,2)+1:boy)=a(1,1)*s1(d(1,2)+1:boy)+a(1,2)*s2(1:boy-d(1,2));
x2d(d(2,1)+1:boy)=a(2,1)*s2(d(2,1)+1:boy)+a(2,2)*s1(1:boy-d(2,1));
mixed_d=[x1d' x2d']';Num_ICA_d=2;

figure(2),subplot(2,1,1); plot(x1n(1:boy));title('instanous mix 1');
subplot(2,1,2); plot(x2n(1:boy));title('instanous mix 2');

figure(3) subplot(2,1,1); plot(x1d(1:boy)); title('delayed mix 1');
subplot(2,1,2); plot(x2d(1:boy)); title('delayed mix 2');

[recovered_n,Wt_n]=myFastICA(mixed_n,Num_ICA_n);
[recovered_d,Wt_d]=myFastICA(mixed_d,Num_ICA_d);
 figure(4), subplot(2,1,1); plot(recovered_n(1,1:boy));
title('instantenous IC 1');  subplot(2,1,2); plot(recovered_n(2,1:boy));title('instantenous IC 2');
figure(5),subplot(2,1,1); plot(recovered_d(1,1:boy));title('delayed IC 1');
subplot(2,1,2); plot(recovered_d(2,1:boy));title('delayed IC 2');
```

Appendix B
KNEE SIGNALS

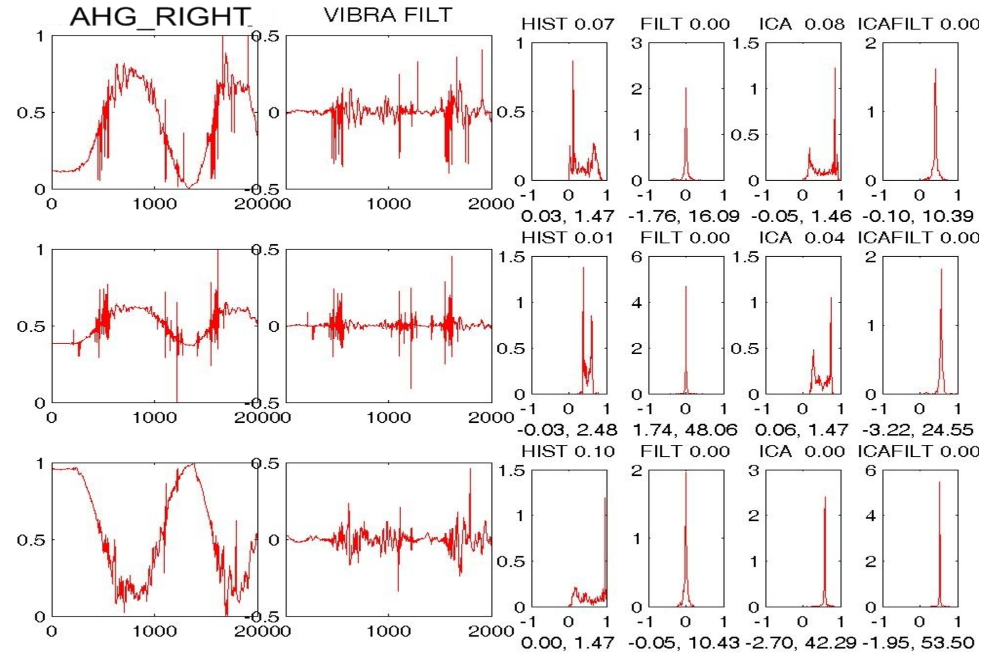
In this appendix some knee data were given to understand the nature of VAG signals. Because of the getting tilt and vibration information together from accelerometers, the running average filtering was used to separate the vibration information from the tilt data.

Running Average filtering is not always the best solution for getting the vibration information, also some other filtering methods like Finite Impulse Response (FIR), Infinite Impulse Response (IIR), Fast Fourier Transform (FFT), Wavelet, Hodrick-Prescott, etc. Their filtering performances are evaluated in terms of their variances, skewness, kurtosis, root mean square error (RMSE) parameters. But here, the running average filtering was ran long time to find the baseline tilt information correctly despite time time cost was too high.

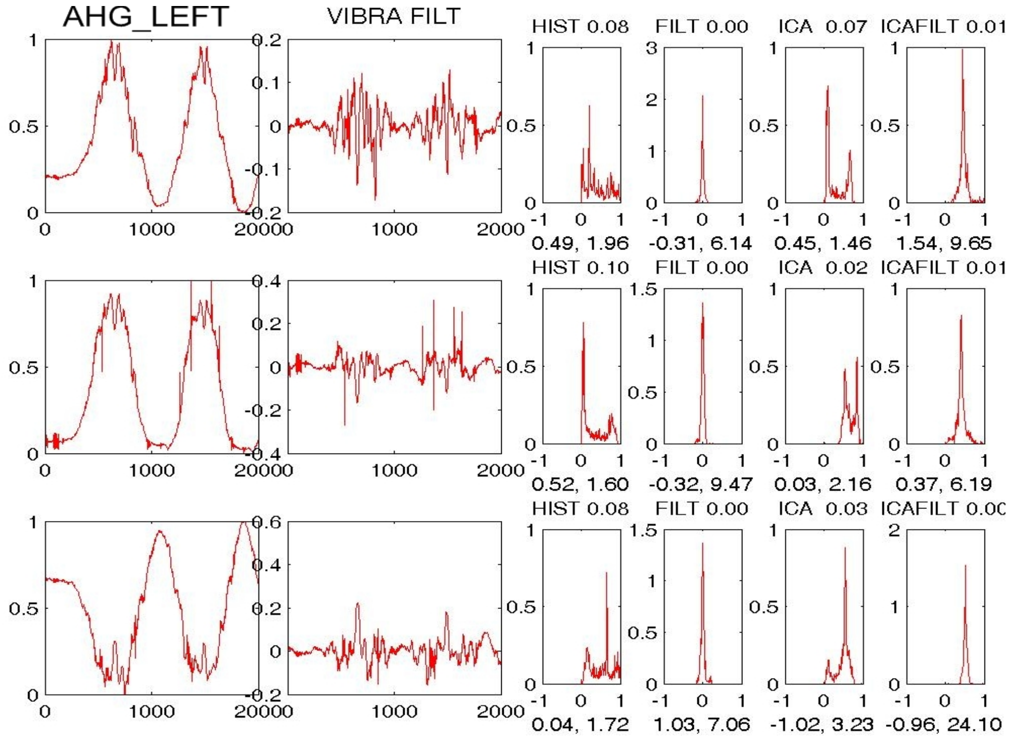
The aim is to show histograms for accelerometer data, vibration data, ICA result of accelerometer data, and ICA result of vibration data respectively in following figures for different cases. Here not only the histograms can show the data differences, also variance, skewness, and kurtosis parameters are important. Thus, below histogram graphics, variance, skewness, and kurtosis values are given. The value at the right of graphic titles are variance parameter, two values below the graphics are skewness and kurtosis parameters respectively.

B.1 Normal knee data 1

AHG right knee data. The trends show this data belongs to a normal case.

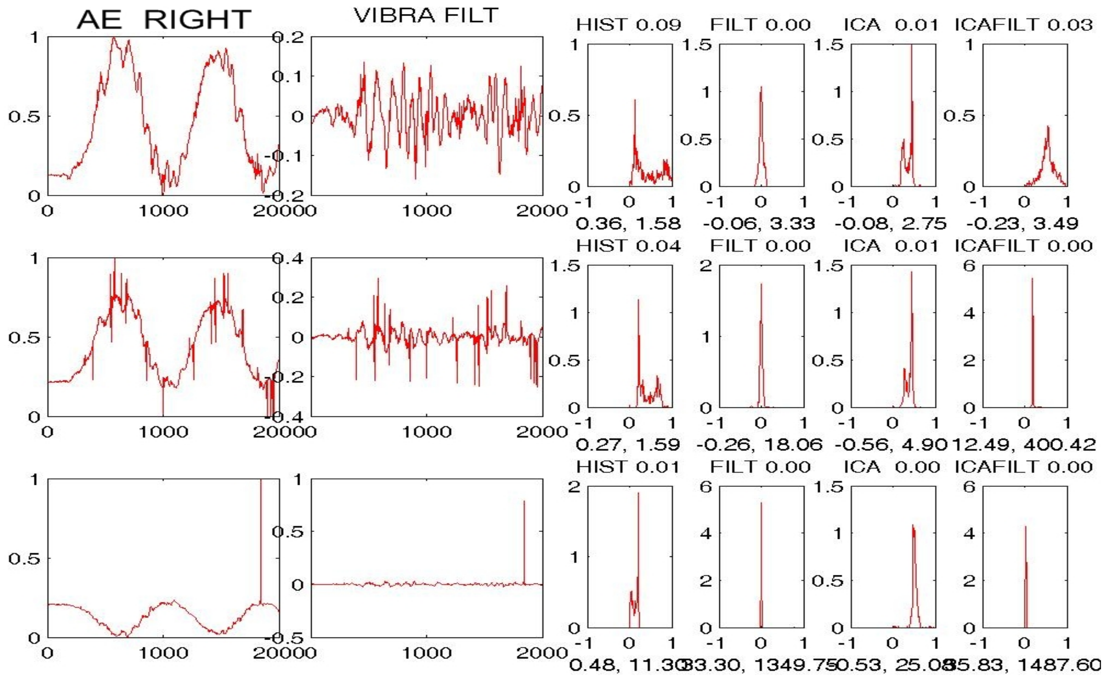


AHG left knee data. The trends show this data belongs to a normal case, but there is a little problematic case here; the knee problem was in a very beginning phase of the arthrosis.

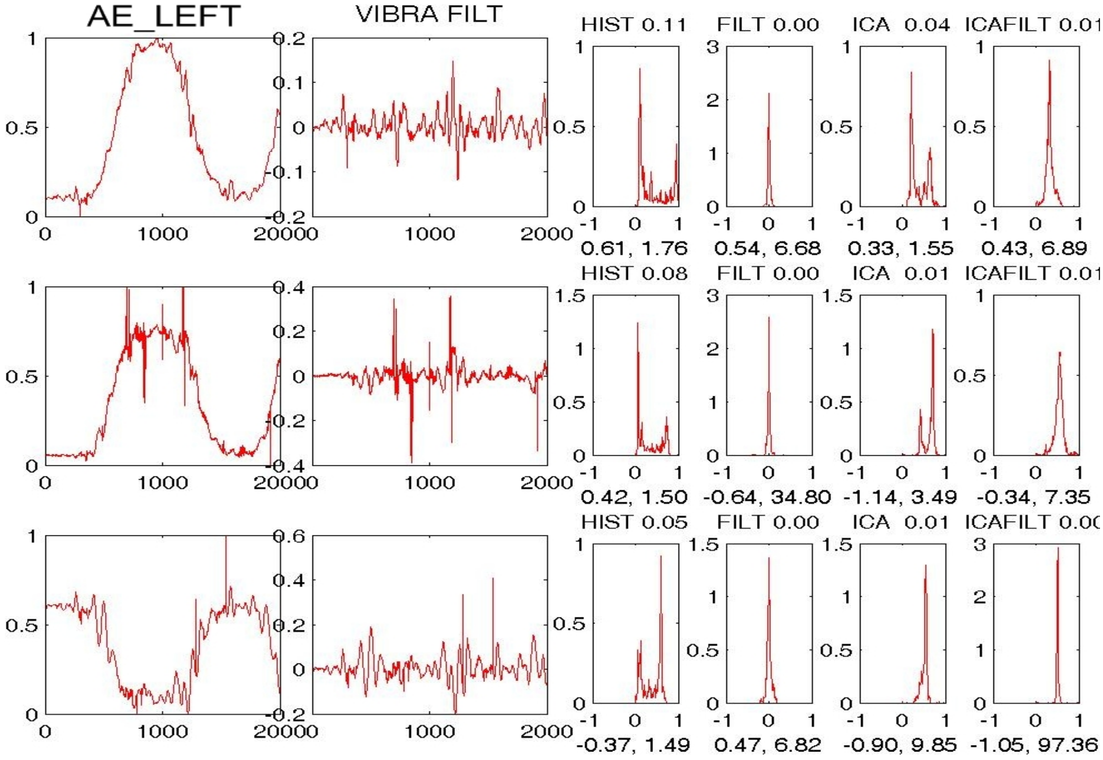


B.2 Normal knee data 2

AE right knee data. The trends show this data belongs to a normal case.

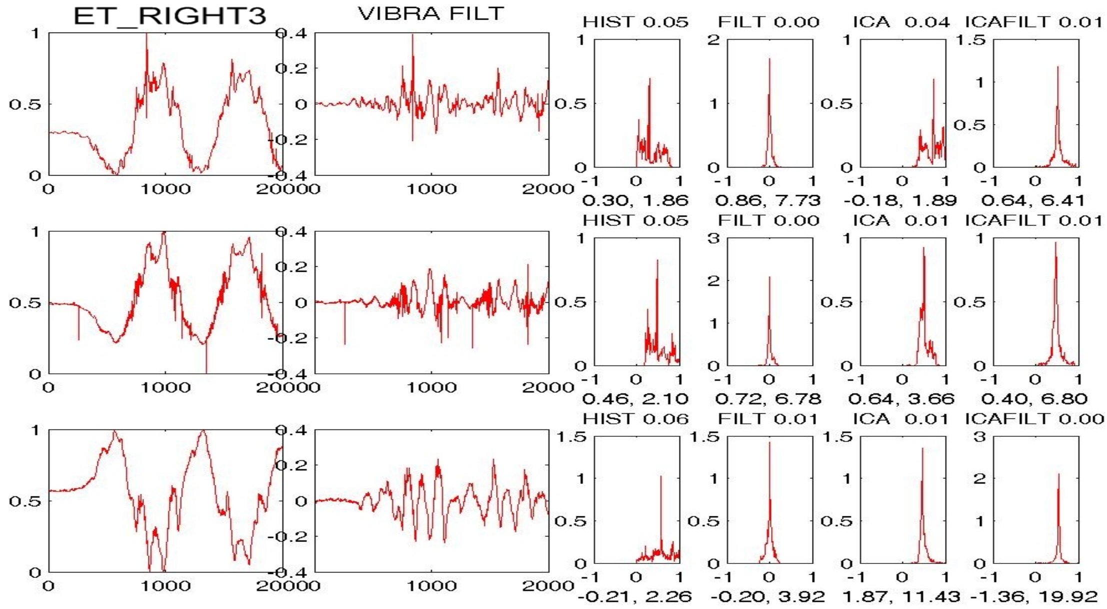


AE left knee data. The trends show this data belongs to a normal case.

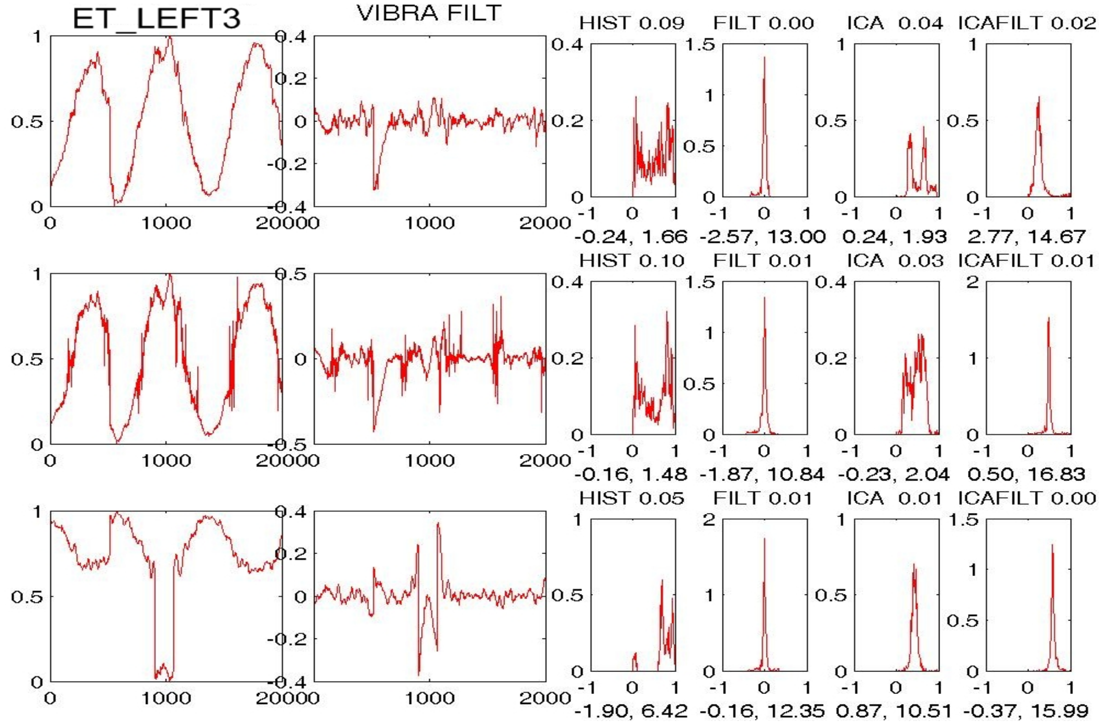


B.3 Abnormal knee data 1

ET right knee data. The trends show this data belongs to an abnormal case. Medial meniscus rear horn degeneration.

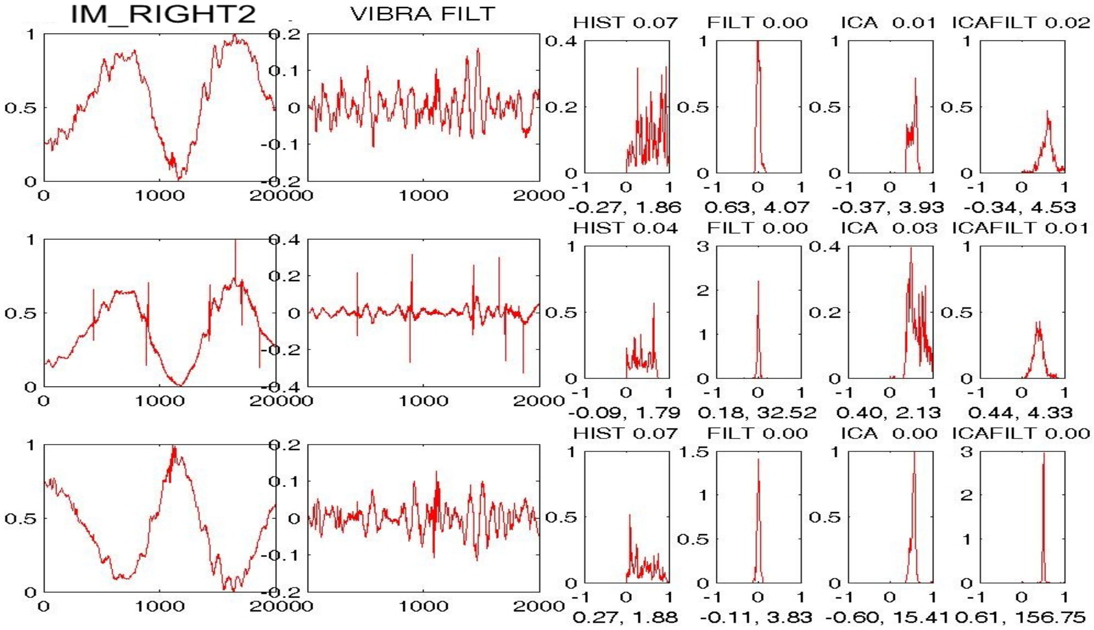


ET left knee data. The trends show this data belongs to an abnormal case. Medial meniscus rear horn degeneration.

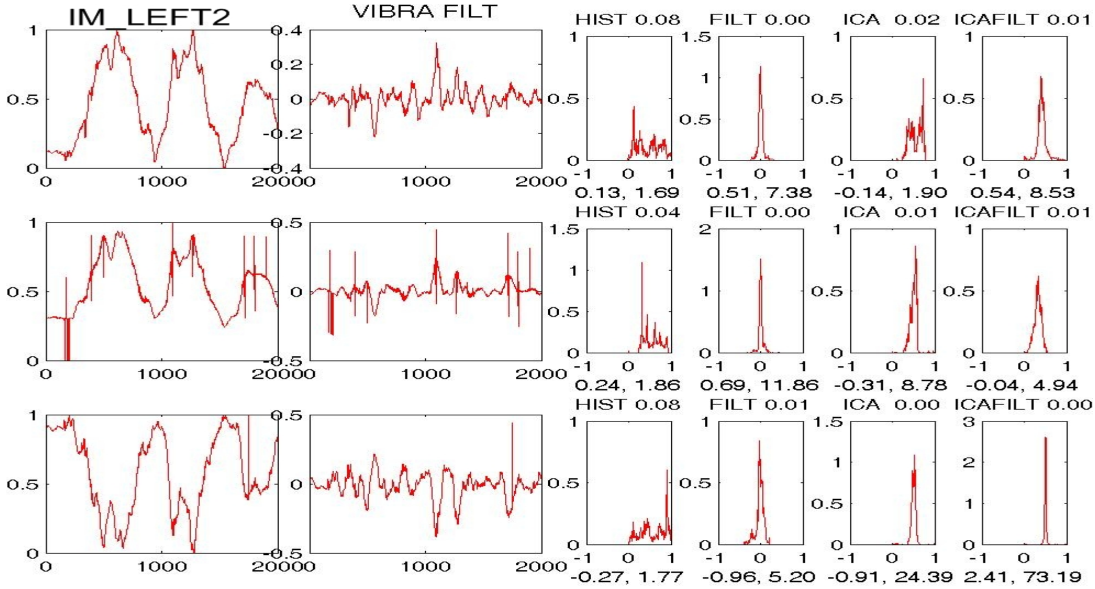


B.4 Abnormal knee data 2

IM right knee data. The trends show this data belongs to an abnormal case. Right normal-grade arthrosis.

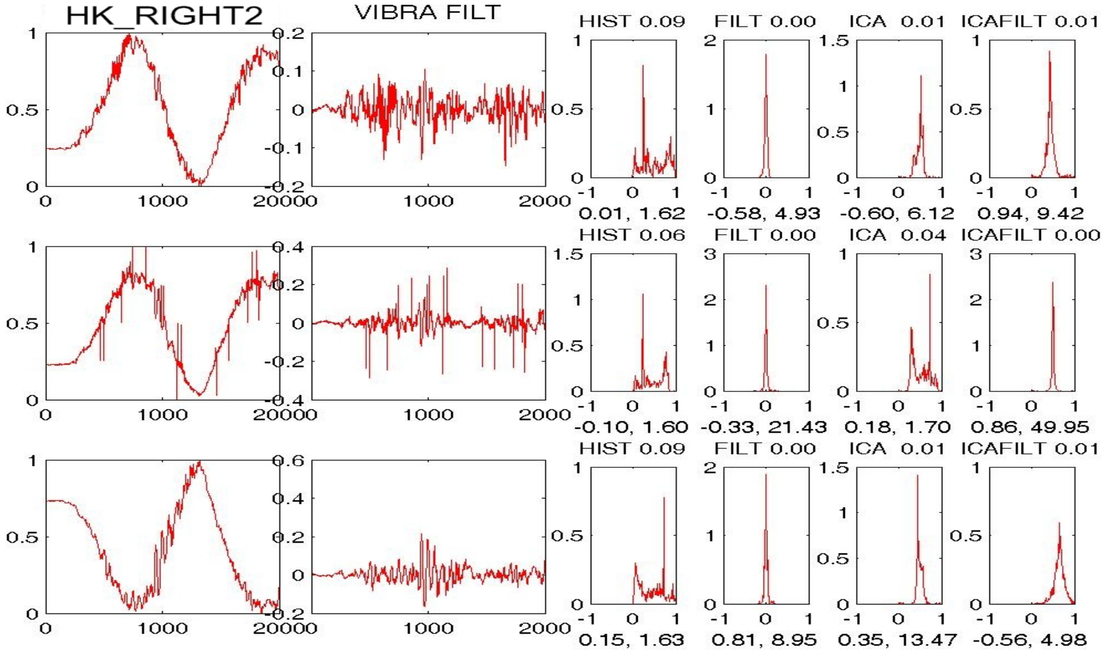


IM left knee data. The trends show this data belongs to an abnormal case. Left high-grade arthrosis.

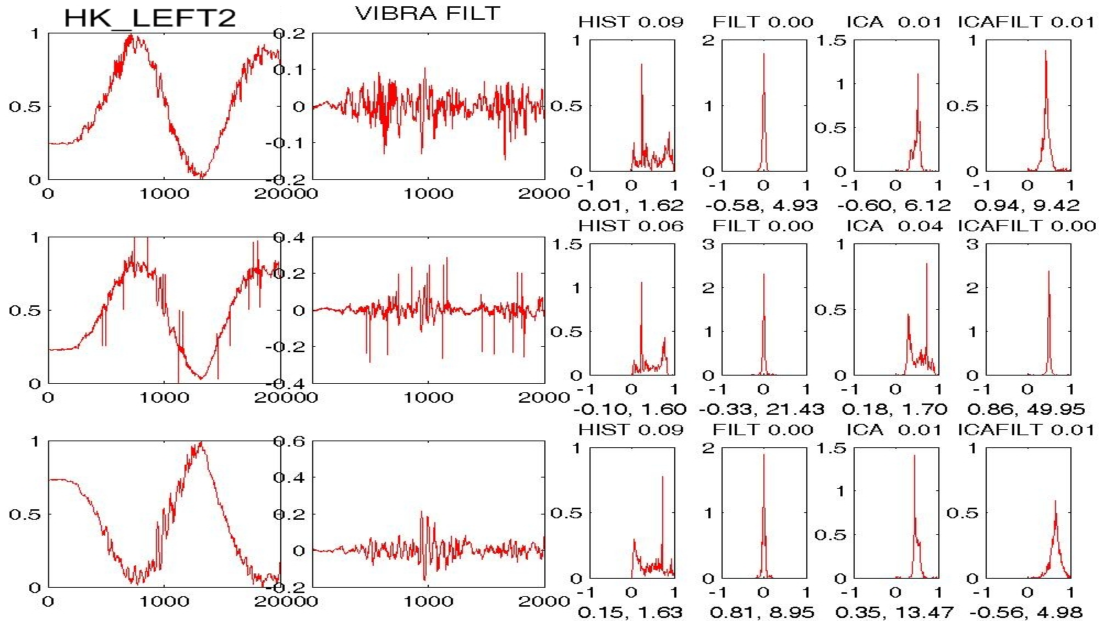


B.5 Abnormal knee data 3

HK right knee data. The trends show this data belongs to an abnormal case. Right high-grade arthrosis.



HK left knee data. The trends show this data belongs to an abnormal case. Left high-grade arthrosis.


HK_LEFT2
VIBRA FILT
HIST 0.09
FILT 0.00
ICA 0.01
ICAFILT 0.01
HIST 0.06
FILT 0.00
ICA 0.04
ICAFILT 0.00
HIST 0.09
FILT 0.00
ICA 0.01
ICAFILT 0.01