Chapter 3

3.1 INTRODUCTION

One of the major tasks proposed in this thesis is to use ICA to find the separating matrix which will be used to estimate the independent components. The separating matrix cannot be written as some function of received samples, whose value could be directly solved, instead the solution is based on cost functions also known as objective functions or contrast functions. The solutions of the separating matrix are found at the maxima or minima of these functions.

Optimization theory includes the study of the maximal and/or minimal values of a function and the conditions for the existence of a unique maximal or minimal value. An example of an optimization problem is that of finding the minimum of a scalar function or objective function and where the minimum is located. Assuming that this objective function say \( g(x) \) is differentiable the conditions for finding a local and global minimum are:

\[
\begin{align*}
\frac{dg(x)}{dx} &= 0 \quad (3.1) \\
\frac{d^2g(x)}{dx^2} &> 0 \quad (3.2)
\end{align*}
\]

It’s important to note that these conditions must be fulfilled in order to define a stationary point as local minima, because without the second condition, fulfilment of the first condition could be either a maxima, a minima or even an inflection point. Stationary point refers to solutions of objective function that satisfied the first equation, but to find the Global minimum, that is, value or values where the function achieves its smallest value. The objective function must be evaluated at each stationary point and the smallest is selected.

In a case where the objective function \( g(x) \) can be shown to be strictly convex there is at most one solution to the equation (3.1) and this solution corresponds to the global minimum of \( g(x) \). A function \( g(x) \) is strictly convex over a closed interval \([a, b]\) if, for any two points \( x_1 \) and \( x_2 \) in \([a, b]\) and for any scalar \( \alpha \) such that \( 0 \leq \alpha \leq 1 \) then
\[ f(\alpha x_1 + (1 - \alpha)x_2) < \alpha f(x_1) + (1 - \alpha)f(x_2) \]  

(3.3)

When the objective function depends on a complex variable say \( z \) and it is differentiable (i.e. analytic) then finding the stationary points of \( g(x) \) is the same as in the real case. Find the derivatives, set it to zero and then solve for the locations of the extrema (i.e. maximals and minimals). The particular interest is when the function is not differentiable, unfortunately in contrast to functions of real variable, many of the functions that are to be minimized are not differentiable. A very good and common example is the simple function \( g(z) = |z|^2 \). Although it is clear that \( g(z) \) has a unique minimum that occurs at \( z = 0 \), this function is not differentiable \([2]\). The problem is that \( g(z) = |z|^2 = zz^* \) is a function of \( z \) and \( z^*(\text{conjugate}) \) and any function that depends on the complex conjugate \( z^* \) is not differentiable with respect to \( z \). The complication can be resolved with either of two methods - the first is to express the objective function in terms of the real and imaginary parts \( z = x + jy \) and minimize \( g(x, y) \) with respect to those two variables.

This approach is unnecessarily tedious but will yield the solution. A more elegant solution is to treat \( z \) and \( z^* \) as independent variables and minimize \( g(z, z^*) \) with respect to both \( z \) and \( z^* \). For example, if we write \( g(z) = |z|^2 \) as \( g(z, z^*) = zz^* \) and treat \( g(z, z^*) \) as a function of \( z \) and keep \( z^* \) constant then,

\[ \frac{dg(z)}{dz} = z^* \]  

(3.4)

Where as if we treat \( g(z, z^*) \) as a function of \( z^* \) with \( z \) constant, then

\[ \frac{dg(z)}{dz^*} = z \]  

(3.5)

Setting both derivatives equal to zero and solving this pair of equations we see that the solution is \( z = 0 \). \([2]\)
3.1.1 Vector Gradients

Another situation is when the objective function depends on a vector–valued quantity \( \mathbf{x} \) the evaluation of the function’s stationary points is a simple extension of the scalar-variable case. For a scalar function of \( m \) real variables \( f(\mathbf{x}) = (x_1, x_2, \ldots, x_m) \), assuming that the function is differentiable this involves computing the gradient, which is the vector of partial derivatives defined by

\[
\nabla f = \begin{bmatrix}
    \frac{\partial f(x)}{\partial x_1} \\
    \vdots \\
    \frac{\partial f(x)}{\partial x_m}
\end{bmatrix}
\tag{3.6}
\]

The gradient point is the direction of the maximum rate of change of \( f(\mathbf{x}) \) and it is equal to zero at the stationary points of \( f(\mathbf{x}) \). Hence the aforementioned condition for a point \( \mathbf{x} \) to be a stationary point of \( f(\mathbf{x}) \) must be satisfied, which is

\[
\frac{df(\mathbf{x})}{d\mathbf{x}} = 0
\tag{3.7}
\]

For this stationary point to be the minimum the Hessian matrix \( H_x \) must be positive definite i.e.

\[
\mathbf{X}^T H_x \mathbf{X} > 0 \quad \text{for all } \mathbf{x}
\]

The Hessian matrix is the second-order gradient of a function, in other words second-order derivatives with \((i,j)\)th element as

\[
\frac{\partial^2 f}{\partial x_i \partial x_j} = \begin{bmatrix}
    \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_m} \\
    \vdots & \ddots & \vdots \\
    \frac{\partial^2 f}{\partial x_m \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_m^2}
\end{bmatrix}
\tag{3.8}
\]

If \( f(\mathbf{x}) \) is strictly convex, then the solution is unique and is equal to the global minimum.
It’s easy to see that the Hessian matrix is always symmetric. The Jacobian matrix of $f(x)$ with respect to $x$ which we will later use is given by:

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$  \hspace{1cm} (3.9)

Thus the $i^{th}$ column of the Jacobian matrix is the gradient vector of $f_i(x)$ with respect to $x$.

### 3.1.2 Matrix gradient

Assuming a scalar valued function $g$ of the elements of an $m \times n$ matrix $W$ in analogy with the vector gradient, then the matrix gradient means a matrix of same size $m \times n$ as the matrix $W$, whose $ij^{th}$ element is the partial derivative of $g$ can be written as:

$$\frac{\partial g}{\partial W} = \begin{bmatrix} \frac{\partial g}{\partial W_{11}} & \cdots & \frac{\partial g}{\partial W_{1m}} \\ \vdots & \ddots & \vdots \\ \frac{\partial g}{\partial W_{n1}} & \cdots & \frac{\partial g}{\partial W_{nm}} \end{bmatrix}$$ \hspace{1cm} (3.10)

### 3.2 LEARNING RULES FOR OPTIMIZATION

Let us look at the rules (or algorithms) of finding the extrema points of a cost function. Although the vector that minimizes the cost function may be found by setting the derivatives of the cost function equal to zero as earlier said, another approach is to search for the solution. There are many search method and we are going to point out few.

#### 3.2.1 Gradient descent

This is an iterative procedure that has been used to find the extrema values of functions long since before the time of Newton. Let us consider in more detail the case when the solution is a vector $w$; the matrix case goes through in the same way.

The basic idea behind this method is to minimize the cost function say $\zeta(w)$ by starting from some initial point value $w(0)$, then computing the gradient of $\zeta(w)$ at this point and moving
in the direction of the negative gradient or the steepest descent by a suitable distance. This procedure is repeated at the new point, this is continued until it converges, which in practice happens when the Euclidean distance between two consecutive solutions, that is \( \|w(t) - w(t-1)\| \) goes below some small tolerance level.

Thus the learning rule update is

\[
w(t) = w(t-1) - \alpha(t) \frac{\partial \zeta(w)}{\partial w}
\]  

(3.11)

With the gradient taken at the point \( w(t-1) \), while the parameter \( \alpha(t) \) often refers to the step size, or the learning rate, gives the length of the step in the steepest descent which always point in the negative gradient.

Geographically it can be viewed as a graph of the function \( \zeta(w) \) which is equivalent of mountain terrain, so the gradient descent learning rule means that we are always going down the hill in the steepest direction.

The major disadvantage of this method is that it always leads to the closest local minimum instead of a global minimum, unless the function \( \zeta(w) \) is strictly convex. That is when one local minimum exists which is also the global minimum. Non-quadratic functions may have many local minimal values therefore good initial values are important in initializing the algorithm [5]. It’s important to also note that the choice of an appropriate learning rate is essential, because too small a value will lead to slow convergence while too large a value will lead to overshooting and instability which prevents convergence altogether.
For better understanding of the idea, let us assume that the contour plot of the cost function $\zeta(w)$ is as shown in Fig. 3.1

![Contour map of assumed cost function.](image)

**Fig.3.1: Contour map of assumed cost function.**

From the illustration from fig 3.1, there are three extrema, two local minima and one global minimal. By choosing the initial values in the direction of the gradient vector, the result is one local minimum, which is not the optimal solution. Based on this it’s important to be very careful when choosing initial values say $w(0)$ as earlier mentioned.
3.2.2 Natural Gradient

From the disadvantage of Gradient descent unless the cost function is quadratic it may not lead to the global minimum in non-quadratic function which may have more than one extrema. As earlier said, it points in the opposite direction of the gradient in a Euclidean orthogonal coordinate system. Orthogonal mean that the coordinates are perpendicular to each other.

Amari reports in [13,14], that parameter space is not always Euclidean but also has Riemannian metric structure. In this case the Gradient descent does not give the steepest direction of the target function; instead the steepest direction is given by the Natural gradient. Since ICA learning rules will constrain to the case of nonsingular matrix also know as invertible matrix (i.e. matrices with existing inverse) which are very important to ICA. Amari also added that these matrices (i.e. nonsingular) have a Riemannian structure with a convenient computable natural gradient.

Let us assume that the function is $\zeta$ and $\partial W$ be a small deviation or minimization of a matrix from $W$ to $W + \partial W$, under the constraint that the square norm $\|\partial W\|^2$ is a constant.

One of the major requirements is that at any step in gradient algorithm for minimization of a function, there must be a direction of the step and the length of the step. Keeping this length constant the optimal direction is search for [8].

Introducing an inner product at $W$ by defining the square norm of $\partial W$ as:

$$\|\partial W\|^2 = \langle \partial W, \partial W \rangle_W$$

(3.12)

And by multiplying $W^{-1}$ from right, we mapped $W$ to $WW^{-1} = I$ i.e. identity matrix and $W + \partial W$ is mapped to

$$(W + \partial W)W^{-1} = I + \partial W W^{-1} = I + \partial X.$$  

(3.13)

where $\partial X = \partial W W^{-1}$.

It means that a deviation $\partial W$ at $W$ is equivalent to a deviation $\partial X$ at $I$.  

34 | P a g e
Amari argues that due to the Riemannian structure of the matrix space, it requires that metric is kept invariant, that is, the inner product of $\partial W$ at $W$ is equal to the inner product of $\partial WZ$ at $WZ$ for any $Z$.

$$\langle \partial W, \partial W \rangle_w = \langle \partial WZ, \partial WZ \rangle_{wz} \quad (3.14)$$

Putting $Z = W^{-1}$ then $WZ = I$ and if the inner product at $I$ is defined as

$$\langle \partial W, \partial W \rangle_I = \sum_{i,j} (\partial W_{i,j})^2 = trace(\partial W^T \partial W) \quad (3.15)$$

then equation (3.14) can be deduced to

$$\langle \partial W, \partial W \rangle_w = \langle \partial WW^{-1}, \partial WW^{-1} \rangle_I = trace( (W^T)^{-1} \partial W^T \partial WW^{-1} ) \quad (3.16)$$

Amari finally showed that keeping this inner product constant, the largest increment for $\zeta(W + \partial W)$ is obtained in the direction of the natural gradient which is

$$\frac{\partial \zeta}{\partial W_{nat}} = \frac{\partial \zeta}{\partial W} W^T W \quad (3.17)$$

It then means that the usual Gradient descent at point $W$ must be multiplied from right by the matrix $W^T W$ and this gives the Natural Gradient learning rule update at point $W(t)$ as follows:

$$W(t) = W(t-1) - \alpha(t) \frac{\partial \zeta}{\partial W} W^T W \quad (3.18)$$

Amari pointed out that Natural Gradient learning rule is Fisher efficient, implying that it has asymptotically the same performance as the optimal batch estimation of parameters. Natural Gradient can be used in many applications like statistical estimation of probability density function, multilayer neural network, blind image deconvolution, blind image separation, etc and it is the algorithm that is implemented in this thesis.
3.3 BLIND SOURCE SEPARATION

Blind source separation (BSS), is a technique for estimating individual source components from their mixtures at sensors. This is called blind because the estimation is done without prior information on the sources, that is their spatial location and time activity distribution; and on the mixing function, i.e. information about the mixing process. The problem has become increasingly important in the area of image and image processing due to their prospective application in image recognition, teleconferencing, hearing aids, telecommunications and medical image processing, etc. In these applications, images are mixed in a convolutive manner, at times with reverberation otherwise literary called echo. This makes blind source separation a very difficult problem. Very long Finite Impulse Response (FIR) filters of several thousand taps will be needed to separate acoustic images mixed in such manner. [1]

Many scholars and researchers have been studying the problem of Blind Image Separation and numerous ways have been proposed to solve the problem. Recently attention has been drawn to Independent Component Analysis (ICA), which is a very important statistical tool for solving the BSS problem.

Due to filtering imposed on sources by their environments difference between the sensors and propagation delays in real world applications like the aforementioned applications are convolved image. There is a need to extend the Blind Source Separation using Independent Components Analysis technique so that it will be applicable to the convolutive mixtures. There are three major approaches of solving the convolutive mixtures using ICA/BSS as enumerated in [1] by Makino et al, which all have some advantages and disadvantages.

First is the Time domain BSS, where ICA is directly applied to the convolutive mixtures. This achieves a good result once the algorithm converges, but it is computationally expensive because we are dealing with convolution operations.

The second is the Frequency domain BSS, where the convolutive mixtures are first converted to Frequency domain, then ICA is applied to each frequency bin, which is seen now as instantaneous mixtures, since convolution in time domain is equal to multiplication in frequency domain. This method is simple but the problem of permutation and scaling is even greater than in
the Time domain BSS since different frequency bands may have different permutation and scaling.

The third approach uses both the time domain and frequency domain. Here the filter coefficients are updated in the frequency domain but the nonlinear functions for evaluating independence are applied in the time domain. This approach does not have the permutation problem, because the independence of the separated images are evaluated in time domain, but the time spent in switching between the time domain and the frequency domain is non negligible. [19]

From these three approaches, the second is better in terms of computational demand, but the problem of permutation and scaling has to be resolve as earlier mentioned.

Fig.3.2: Convulsive Mixture of two sources and two sensors represented as Images

Consider there are two people in a room speaking simultaneously, let us denote the image emitted from Speaker1 with \( S_1(t) \) and Speaker2 with \( S_2(t) \) correspondingly, and there are two microphones placed at different locations in the same room. These microphones will produce two time images which we can called \( X_1(t) \) and \( X_2(t) \), where \( t \) is the time index. Each of these recorded images is a sum of the images from the two speakers because each microphone is “hearing” the two speakers at the same time. It can be expressed as a linear equation as shown below

\[
X_1(t) = a_{11} \otimes S_1 + a_{12} \otimes S_2 \\
X_2(t) = a_{21} \otimes S_1 + a_{22} \otimes S_2
\]

(3.19)
Where $\otimes$ represents convolution, $a_{ij}$ are some parameters that depend on the distances of the microphones from the speakers and on the room properties, these parameters are referred to as the room impulse response.

This scenario is normally refer to as the cocktail-party problem, it would be very useful if the original image images $S_1(t)$ and $S_2(t)$ could be estimated. What it then means is that if we knew the values of $a_{11}$, $a_{12}$, $a_{21}$ and $a_{22}$ (i.e. the impulse response) we could easily solve the linear equation above with any of the classical methods available, but not knowing these values is the main problem and even make it a difficult one. Independent Component Analysis (ICA) can be used to estimate these values and allow us to separate the two original images $S_1(t)$ and $S_2(t)$ from their mixtures $X_1(t)$ and $X_2(t)$.

The above diagram shows how the blind source separation for convolutive mixtures can be formulated. Assuming with $N$ source images $s_i(t)$ that are mixed and observed at $M$ sensors as $x_j(t)$, mathematically can be represented as:

$$x_j(t) = \sum_{i=1}^{N} h_{ji}(l)s_i(t-1) \quad j = 1, \ldots, M \quad (3.20)$$

where $j,i$ represent the impulse response from source $i$ to sensor $j$.

It is always assumed that the number of sources $N$ is known or can be estimated and the number of sensors $M$ is equal to or greater than $N$; $M \geq N$. The mixed image $X$ is passed through the separation system which consists of finite impulse response (FIR) filters $w_{ij}(l)$ of length $l$ to produce $N$ separated images $y_j(t)$ as the output. Thus;
\[ y_i(t) = \sum_{j=1}^{M} \sum_{l=0}^{L-1} w_{ij}(l)x_j(t-l) \quad i = 1, \ldots, N \quad (3.21) \]

The separation filters, \( w_{ij}(l) \) should be estimated blindly, i.e., without any knowledge of the source image \( s_i(t) \) or \( h_{ji}(t) \) the mixing function.

### 3.4 INDEPENDENT COMPONENT ANALYSIS

Independent Component Analysis is a powerful higher order statistical technique used to separate independent sources that were linearly mixed together through a medium and received at several sensors.

Let us assume that there are \( N \) linear mixtures \( X_1(t), X_2(t), \ldots, X_N(t) \) of \( M \) independent components.

\[
X_j = a_{j1}s_1 + a_{j2}s_2 + \ldots + a_{jN}s_N \quad \text{for } j = 1, 2, \ldots, M \quad (3.22)
\]

It is to be noted that the number of sensors is usually greater than or equal to the number of sources \((M \geq N)\).

In the general ICA model the time index is not used because each of the mixtures \( x_j \) as well as independent components \( S_k \) are assumed as random variables instead of proper images [5]. It is convenient to use vector-matrix notation instead of the sums as shown above in 3.22. Let \( \mathbf{x} \) denote the random vector whose elements are the mixtures \( x_1, \ldots, x_M \) and let \( \mathbf{s} \) denote the random vector with elements \( s_1, \ldots, s_N \). Let \( \mathbf{A} \) denote the matrix with elements \( a_{ij} \). The mixture model using the vector-matrix is written as \( \mathbf{x} = \mathbf{A} \mathbf{s} \) or graphically as

\[
\begin{bmatrix}
X_1 \\
\vdots \\
X_M
\end{bmatrix} =
\begin{bmatrix}
a_{11} & \cdots & a_{1N} \\
\vdots & \ddots & \vdots \\
a_{M1} & \cdots & a_{MN}
\end{bmatrix}
\begin{bmatrix}
s_1 \\
\vdots \\
s_N
\end{bmatrix}
\quad (3.23)
\]

This ICA model is a generative model because it describes how the observed data are generated by a process of mixing the components \( s_j \). It’s important to recall that all we observe
and know is the random vector $x$ and we estimate the inverse $A$ and $s$ using it alone. This can’t be achieved without some general and fundamental assumptions which are:

i) Assume that the components $s_j$ (i.e. the sources) are statistically independent between each other

ii) And that the Independent components must have non Gaussian distribution (At most one source may have Gaussian distribution).

### 3.4.1 Statistical Independence

The concept of statistical independence can easily be explained with an example. Let us assume that $y_1$ and $y_2$ are scalar valued random variables $y_1$ and $y_2$ are said to be independent if the information of the value of $y_1$ does not give any information of the values of $y_2$ and likewise information of the values of $y_2$ does not give any information of $y_1$. It is important to note that the sources ($s_i$) are referred alone and not the mixtures ($x_i$), which generally are highly dependent.

In probability theory independence can be defined by the probability densities. Let $P_1(y_1)$ denote the Marginal Probability Density Function (i.e. the probability density function when $y_1$ is considered alone) and let $P(y_1, y_2)$ denote the Joint Probability Function (i.e. considering $y_1$ and $y_2$ together).

Then

$$P_1(y_1) = \int P(y_1, y_2) \, dy_2$$  \hspace{1cm} (3.24)

Likewise

$$P_2(y_2) = \int P(y_1, y_2) \, dy_1$$  \hspace{1cm} (3.25)

Here, $y_1$ and $y_2$ are independent if and only if the Joint Probability Density Function can be factorised in the following way:

$$P(y_1, y_2) = P_1(y_1)P_2(y_2)$$  \hspace{1cm} (3.26)
In other words two events are statistically independent if the probability of their occurring jointly equals the product of their respective probabilities.

3.4.2 NonGaussian Distribution

Another fundamental restriction or assumption in ICA is that the independent component must be nonGaussian or at most may have one Gaussian distribution, for Independent Component Analysis to be possible[5,7,8]; this is because of the joint probability densities of Gaussian random variables are completely symmetric.

Moreover for Gaussian random variables mere uncorrelatedness implies independencies and thus any decorrelating representation would give independent components. Nevertheless if not more than one of the components are Gaussian it is still possible to identify the nonGaussian independent components as well as the corresponding columns of the mixing matrix. In other words without nonGaussianity, estimation of the ICA model is not possible at all.

Fig.3.4: Plot of Probability Density Function of some Gaussian distributions. [10], \( \mu \) is the mean and \( \sigma^2 \) is the variance.
3.5 MEASURES OF NONGAUSSIANITY

NonGaussianity is important in estimation of ICA models thus there is need for a quantitative measure. Let us look at some measures of nonGaussianity their advantages and disadvantages.

3.5.1 Kurtosis

Kurtosis is a parameter that describes the shape of a random variables’ probability distribution function. It can also be used to measure nonGaussianity of a random variable, because a Gaussian distribution (which sometimes is refer to as normal distribution) has a normalized kurtosis equal to zero, in other words Mesokurtic.

Since Gaussian distribution normalizes kurtosis equal to zero, it can be used as a reference point to know distributions that are below Gaussian distribution (subGaussian, they have negative kurtosis) in other words Platykurtic and those that are above Gaussian distribution(superGaussian, these have positive kurtosis) otherwise called Leptokurtic. A leptokurtic distribution has a more acute “spiky” shape around zero this means a higher probability than a Gaussian distribution near the mean and a “long tail”; this entails a higher probability than a Gaussian distribution at the extreme values. A good example of a leptokurtic distribution is the Laplace distribution.

A platykurtic distribution has a “flatter peak” around mean, this implies a lower probability than Gaussian distribution near the mean and “small tail” (that is lower probability than a Gaussian distribution at the extreme values. A typical example of a platykurtic distribution is the Uniform Distribution.
Using statistical terms the Normalized Kurtosis of a standard variable $y$ is defined by:

$$kurt(y) = \frac{E\{y^4\}}{E\{y^2\}^2} - 3$$  \hfill (3.27)

This shows that Kurtosis is simply a normalized version of the fourth moment $E\{y^4\}$.

Some properties of kurtosis are

If $x_1$ and $x_2$ are two independent random variables, then

$$kurt(x_1 + x_2) = kurt(x_1) + kurt(x_2)$$  \hfill (3.28)

and

$$kurt(\alpha x_i) = \alpha^2 kurt(x_i)$$  \hfill (3.29)

Fig.3.5: Different well known distribution together with their kurtosis measure, D= Laplace Distribution, S=hyperbolic Secant Distribution, L= logistic Distribution, N= Normal Distribution, C= Raised Cosine Distribution, W=Wigner Semicircle Distribution, and U= Uniform Distribution [10].

The simplicity both in computation and theory of these properties makes kurtosis widely used to measure non-Gaussianity in ICA but it has some drawbacks in practice when its value has
to be estimated from a measured sample. Its major hindrance is that kurtosis can be very sensitive to “far away” data where its value may depend on only a few observations in the “tail” of a distribution which may be erroneous or irrelevant data. This makes kurtosis not a robust measure of nonGaussianity [5].

3.5.2 Negentropy

Another important measure of nonGaussianity is Negentropy. It is a short word for “Negative Entropy” which from its name is based on Entropy. To continue with negentropy it is important to understand the meaning of Entropy which is a basic concept in information theory.

Entropy of a random variable as defined by [5] can be interpreted as the degree of information that an observation of a random variable gives. It is a measure of the uncertainty associated with random variables, in other words the more random, unpredictable and unstructured the variable is, the larger its entropy.

The entropy of a discrete random variable $x$ is defined as:

$$H(x) = -\sum_{a_i} P(x = a_i) \log_2 P(y = a_i) = E_x (I(x))$$  \hfill (3.30)

where $a_i$ is the possible values of $x$, $I(x)$ is the information content or self – information of $x$ which is itself a random variable, $P(x = a_i), P(y = a_i)$ is the probability density function.

This definition can also be extended to the continuous random variable. In this case it is often referred as Differential Entropy. Assuming a continuous random variable $y$, the differential entropy is written as:

$$H(y) = -\int f(y) \log_2 f(y) dy$$ \hfill (3.31)

where $f(y)$ is the probability density function of $y$.

One of the fundamental rule of thumb in information theory is that a Gaussian Variable has very large entropy among all random variables of equal variance. This means that Gaussian
distribution is very random, disorganised and unstructured distribution and this implies that entropy can be used to measure nonGaussianity.

To measure a nonGaussianity of a variable that will give us zero for a Gaussian variable and non negative for nonGaussian variable, we use Negentropy which can be defined for a standard variable $y$ as:

$$J(y) = H(y_{gauss}) - H(y)$$  \hspace{1cm} (3.32)$$

where $y_{gauss}$ is a Gaussian variable of the same covariance matrix as $y$.

Negentropy is a very good measure of nonGaussianity but not with a setback which is difficulty in computation. Estimating negentropy requires an estimate of the probability density function so simple approximations of negentropy are very helpful. A well known computational and simple approximation of entropy of a standardised (i.e. zero mean and unit variance) random variable is

$$J(y) = \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} kurt(y^2)$$  \hspace{1cm} (3.33)$$

Assumed that the random variable is standardised that is zero mean and unit variance then the first term on the right hand side of the above equation is equal to zero in 3.33 when the random variable has a symmetric distribution which is often the case. Then this approximation will be equal to square of the Kurtosis, thereby putting us in the same problem as mention in the last subsection, which is non-robustness of measure of nonGaussianity.

Another approach described in [5] is based on the maximum – entropy principle, here the higher order approximation is replaced with the expectation of general non-quadratic functions or “non-polynomial moments”. The polynomial function $y^3$ and $y^2$ can be replaced by any other functions $G_i$ (where i is an index not exponent), the method then gives a simple way of approximating the negentropy based on the expectation $E\{G_i\}$ [5]. So, the new approximation the becomes

$$J(y) \approx [E\{G(y)\} - E\{G(y)\}]^2$$  \hspace{1cm} (3.34)$$

for one quadratic function $G$. 

45 | P a g e
where \( v \) is a Gaussian variable of zero mean and unit variance.

Now, in choosing the function \( G \) one has to be careful so as to:

1. get an approximation better than (3.33)
2. not to get a kurtosis based approximation.

Therefore choosing \( G \) we need to be sure that practical estimation of \( \mathbb{E}\{G(y)\} \) should not be practically difficult and should not be too sensitive to the outliers (outliers is a statistical term for extreme values of data). Secondly \( G(y) \) must not grow faster than the quadratic function of \( y \) i.e. \( |y|^2 \).

The choices of \( G \) that have proved very useful according to [5,7,8] are:

- \( G_1(y) = \frac{1}{a_i} \log \cosh(a_i y) \)  
  \( (3.34) \)

- \( G_2(y) = -\exp\left(-\frac{y^2}{2}\right) \)  
  \( (3.35) \)

Where \( 1 \leq a \leq 2 \) is a suitable constant, often taken equal to one, this gives a very good compromise between kurtosis and Negentropy [5,7,8].

### 3.6 MAXIMUM LIKELIHOOD METHOD

In order to understand ICA by Maximum likelihood (ML), it is important to understand some basic terms in Estimation Theory which maximum likelihood belongs to. Estimation theory aims at extracting from noise corrupted observations, information or data of interest. Assuming that there are \( t \) scalar measurements say \( x(1), x(2), \ldots, x(t) \) containing information about \( n \) quantities that we which to estimate say \( u_1, u_2, \ldots, u_n \). These quantities \( u_i \) are called parameters, they can be represented as a vector \( \mathbf{u} = [u_1, u_2, \ldots, u_n]^T \), so also the measurements \( \mathbf{x} = [x(1), x(2), \ldots, x(t)]^T \), where \( T \) means transpose of the vector.

Generally, the estimator of the parameter vector, represented as \( \hat{\mathbf{u}} \), is a vector function by which the parameters can be estimated from the measurements, so mathematically we can write it as:

\[
\hat{\mathbf{u}} = f(x_n) = f(x(1), x(2), \ldots, x(t)) 
\]

Or individually can be written as from 3.35:
\[ \hat{u}_i = f_i(x_i) \]  

(3.36)

The numerical value of an estimator \( \hat{u}_i \) is called the estimate of \( u_i \).

There are numerous examples of estimators like method of moments, least squares, Bayes, maximum likelihood, to name but a few but here we are dealing with maximum likelihood.

Maximum likelihood (ML) estimator assumes that the priori information or density of the distribution is known or assumed known [8]. It has some asymptotic (asymptotic in the sense that the more the variables the better the result) optimal properties (which are its consistency and efficiency) that makes it a robust and desirable choice of estimation. The maximum likelihood estimate of a parameter \( U \) (represented as \( \hat{U}_{ML} \)) is the value of the estimate that maximizes the likelihood function of the measurement just as its name implies.

The likelihood function, which can be likened to the joint probability function of the measurements assuming that the measurements are independent, is given as:

\[
L(x(1), x(2), \ldots, x(t) \mid u_1, u_2, \ldots, u_n) = L = \prod_{i=1}^{t} P(x(i); u_1, u_2, \ldots, u_n) 
\]  

(3.37)

Taking the logarithm of the above equation we have

\[
\ln L = \sum_{i=1}^{t} \ln P(x_i; u_1, u_2, \ldots, u_n) 
\]  

(3.38)

Maximum likelihood estimates are obtained by maximizing \( L \) or \( \ln L \), by maximizing \( \ln L \) which is much easier than \( L \), the maximum likelihood estimate of \( U \) are the simultaneous solutions of \( n \) equations such that:

\[
\frac{\partial \ln L}{\partial u_j} = 0 \quad \text{for } j = 1, 2, \ldots, n 
\]  

(3.39)

Now, that a little background knowledge of maximum likelihood estimates is known in order to derive the ICA algorithm using maximum likelihood estimator there is a need to have a review of the probability density caused by a linear transform to a standard variable.
3.6.1 Probability Density Function of a Transformed Variable

Assume two random variables \( X \) and \( S \) of \( n \) dimension, that are related as follows:

\[
X = AS \quad \text{(3.40)}
\]

for which the inverse

\[
S = A^{-1}X \quad \text{(3.41)}
\]

exist and is unique, then the probability density \( P_x(X) \) of \( X \) can be obtained from the Probability Density Function (PDF) \( P_s(S) \) of \( S \) as shown below:

\[
P_x(X) = \frac{1}{\det J_g(A^{-1}(X))} P_s(A^{-1}(X)) \quad \text{(3.42)}
\]

where \( J_g \) is the Jacobian matrix, that is:

\[
J_g(X) = \begin{bmatrix}
\frac{\partial g_1(X)}{\partial x_1} & \cdots & \frac{\partial g_a(X)}{\partial x_1} \\
\vdots & \ddots & \vdots \\
\frac{\partial g_1(X)}{\partial x_n} & \cdots & \frac{\partial g_a(X)}{\partial x_n}
\end{bmatrix}
\quad \text{(3.42a)}
\]

Since the transformation \( X = AS \) is linear and nonsingular, then the equation (3.42a) can be reduced to:

\[
P_x(X) = \frac{1}{|\det A|} P_s(A^{-1}(X)) \quad \text{(3.43)}
\]

3.6.2 The Maximum Likelihood of ICA Model

Based on equation (3.43) above which is the PDF of a linear transformation and being aware of our ICA generative model [9]

\[
x = As
\]

The PDF of the mixture vector \( x \), \( P_x \) can be written as:
\[ P_s(x) = |\text{det } C| P_s(s) \] (3.44)

Where \( C = A^{-1} \) since \( A \) is nonsingular, \( P_i \) denotes the densities of the independent components. Recalling that \( s = A^{-1}x = Cx \), we can write equation (4.29) as follows:

\[ P_i(x) = |\text{det } C| \prod \hat{P}_i(c_i^T x) \] (3.45)

Assuming \( x \) are \( N \) observations which of course are statistically independent of each other then the likelihood denoted by \( L \) (please note that at this juncture there is a need to find the likelihood on which the calculation of the Maximum likelihood is based) can be obtained as the product of these densities evaluated at the \( N \) points. This is evaluated as a function of \( C \)

\[ \ln L(C) = \prod_{j=1}^{N} \prod_{i} P_i(c_i^T x(j))|\text{det } C| \] (3.46)

As earlier said it is easier and more practical to use the logarithm of the likelihood based on this 3.46 which can be written as

\[ \ln L(C) = \sum_{j=1}^{N} \sum_{i} \ln P_i(c_i^T x(j)) + N \ln|\text{det } C| \] (3.47)

It is important to note that taking logarithm of likelihood does not make any difference in the location of the optimum value since the maximum of logarithm \( \ln L(C) \) is obtained at the same point as the maximum of the \( L(C) \). The base of the logarithm does not make any difference although natural logarithm is preferred.

Equation (3.47) can be further reduced as follows

\[ \frac{1}{N} L(C) = M \left[ \sum_{i} \ln P_i(c_i^T x(j)) \right] + \ln|\text{det } C| \] (3.48)

where \( M \) is an average computed from the observed sample i.e. mean over \( N \) samples.

Performing maximum likelihood estimation, in practice an algorithm is needed to perform the numerical maximization of the likelihood function such that the gradient of the log likelihood function in equation (3.48) can be written as [10]:
\[
\frac{1}{T} \frac{\partial \ln L}{\partial \mathbf{C}} = \left[(\mathbf{C}^T)^{-1}\right] + g(\mathbf{C}x)\mathbf{x}^T \tag{3.49}
\]

Or in the gradient descent learning rule form:

\[
\mathbf{W}(t) = \mathbf{W}(t-1) - \alpha \left( (\mathbf{C}^T)^{-1} + g(\mathbf{C}x)\mathbf{x}^T \right) \mathbf{C} \tag{3.50}
\]

The Natural gradient method simplifies the maximization of the likelihood and it makes it better conditioned. To derive the natural gradient ICA algorithm multiply the second part of the above equation (3.50) by the matrix \(\mathbf{C}^T \mathbf{C}\) from right just as earlier shown in 3.13

\[
\mathbf{w}(t) = \mathbf{w}(t-1) - \left( \alpha (\mathbf{C}^T)^{-1} + g(\mathbf{C}x)\mathbf{x}^T \right) \mathbf{C} \mathbf{C}^T \tag{3.51}
\]

and this reduces to

\[
\mathbf{W}(t) = \mathbf{W}(t-1) - \left( \alpha (\mathbf{I} + g(\mathbf{s})\mathbf{s}^T) \right) \mathbf{C} \tag{3.52}
\]

But the problem is that the density distribution of the independent components which we don’t know is assumed as a model. The images have a distribution close to Laplacian distribution and that the derivative of tanh function is close to it and can be represented as [11]:

\[
g(s) = \tanh(s) \tag{3.53}
\]

If we write that

\[
g(s) = -\frac{\partial P_s(s)}{\partial s} \tag{3.54}
\]

where

\[
P = \frac{\partial g}{\partial s} \tag{3.55}
\]

then

\[
P_s(s) = \frac{\partial \tanh(s)}{\partial s} = 1 - \tanh^2(s) \tag{3.56}
\]
\[
\frac{\partial P_t(s)}{\partial s} = \frac{\partial^2 \tanh(s)}{\partial s^2} = -2 \tanh(s)(1 - \tanh(s)^2) \quad (3.57)
\]

\[
g(s) = \frac{-2 \tanh(1 - \tanh(s)^2)}{(1 - \tanh(s)^2)} = 2 \tanh(s) \quad (3.58)
\]

At the optimum \( C = W \) and we use \( W(t-1) \) as \( C \), so ICA Natural gradient learning rule update is shown below as

\[
W = W(t-1) - \alpha[I - 2 \tanh(s)s^T]W(t-1) \quad (3.59)
\]

Please note that the constant from 3.58 may be removed when using the update rule. Thus the resulting algorithm is recapitulated as [12]

ICA NATURAL ALGORITHM

Given M mixed images \( x(1), x(2), \ldots, x(m) \) of length L

- Step 1: Initiate \( W \) to the identity matrix (m x m)
  
  set \( m = 1 \)

- Step 2: Calculate the output \( s(m) \) that is the independent components,
  
  \( s(m) = Wx(m) \)

- Step 3: Update \( W \) as \( W = W(t-1) - \alpha[I - 2 \tanh(s)s^T]W(t-1) \) until convergence.

3.7 FREQUENCY DOMAIN CONVOLUTIVE BSS/ICA

The proposed work in this thesis is summarized in the block diagram shown below, the convolved image (observed images) is first passed through the Short Time Fourier Transform (STFT), thereby converting it into frequency domain. The mixed frequency bins are then passed through the unmixing network to get the separated frequency bins; thereafter pass it through the permutation and scaling stage. Afterwards the Inverse Short Time Fourier Transform (ISTFT) is performed which converts the separated frequency bins in into separated time domain image [15].

---

51 | Page
3.7.1 STFT and ISTFT Stage

In section 3.1 as mentioned that frequency domain BSS/ICA is preferable when considering the computational demand. In first stage of the frequency domain BSS/ICA method; the time domain multivariate image $x_i(n)$, sampled at a frequency $f_s$ are converted into frequency domain $x_i(f, \tau)$ $L$–point STFT.

$$x_j(f, \tau) = \sum_{r=-L}^{L} x_j(\tau + r) \text{win}(r) e^{-j2\pi fr}$$

(3.60)

where $f \in \{0, \frac{1}{L}f_s, \ldots, \frac{L-1}{L}f_s\}$ is a frequency, $\text{win}(r)$ [13] is a window that tapers smoothly to zero at each end and $\tau$ is the new index representing time. Note that in the equation (3.35) the length $L$ is divided by 2, this is because in Fourier transformation half of the length of the transformation is repeated and flipped. So the result in the second half of $L$ is the mirror image of the first half. So, the problem is converted into the task of demixing an instantaneous mixture at each frequency bin,

$$x_j(f, \tau) = \sum_{i=1}^{N} hji(f) s_i(f, \tau)$$

(3.61)
Where $h_{ji}(f)$ is the frequency response from source $i$ to sensor $j$ and $s_i(f, \tau)$ is the frequency domain time series image of $s_i(t)$ obtained by using same $L$ point short time Fourier transform. The equation above can be rewritten using vector notation thus,

$$x(f, \tau) = \sum_{i=1}^{N} h_i(f)s_i(f, \tau) \quad (3.62)$$

Where $x = [x_1, x_2, \ldots, x_M]^T$ is the vector of mixed samples and $h_i = [h_{1i}, \ldots, h_{Mi}]^T$ is the vector of the frequency response from source $s_i$ to all $M$ sensors.

In this thesis, the recorded image available is taken as reference proposed by G Vijay Kumar and Dr GV Raju [26], where the representation of two images is sampled at 16kHz. The distance between the speaker, cassette player and the microphones is about 60cm in a square constellation [14].

Short time Fourier transform and inverse short time Fourier transform programs are written in MATLAB and to be sure that these programs work correctly the application is performed on a sinusoidal image with a frequency of 2kHz sampled at 10kHz frequency. The power spectrum of the image is estimated before short time Fourier transforms as show in the Fig. 3.7 below [21]:

![Fig.3.7: The Power Spectral Density of the sinusoidal image before STFT](image-url)
After STFT and consequently ISTFT the power spectrum of the output was also estimated as shown in Fig.3.8 below.

![PSD Image](image)

**Fig.3.8:** The Power Spectral Density of the sinusoidal image after ISTFT

These plots confirmed that STFT and ISTFT MATLAB programs working properly.

Alterning the length of the FFT (and IFFT) used in short time Fourier transform (and Inverse short time Fourier transform) it is found out that it does not affect the image after reconstruction and also the block length does not have any effect on the image after reconstruction, but a short length will give a better time resolution during the processing (i.e. after STFT) while a long one will give a better frequency resolution as mentioned [19].

### 3.7.2 ICA Stage

It can be discussed much on ICA and the derivation of the Natural gradient algorithm. The output of this stage is the unmixing network shown in the block diagram of the proposed work (see figure 3.6) [15]. By writing natural gradient ICA algorithm program in MATLAB before running the program on the frequency bins of the mixtures, it is tested on instantaneous mixtures of two sequences of Laplacian distribution with variance equal to two, zero mean and the length equal to ten thousand. The mixing matrix A is a 2-by-2 matrix with a full rank (i.e. the matrix is invertible).

By checking the performance of ICA separation by plotting the condition (the condition number of a matrix is a measure of stability or sensitivity of a matrix to numerical operations,
matrices with condition number near one is said to be well conditioned) of the performance matrix during training. The performance matrix is,

\[ \mathbf{P} = \mathbf{W} \times \mathbf{A} \]  

(3.63)

where \( \mathbf{W} \) is the separation matrix, should be equal to a possibly rescaled and permutated identity matrix. It then means that the condition number of the performance matrix should tend towards one during the training.

The plot of the condition number is show in Fig.3.9 below

![Plot of Condition of performance matrix](image)

**Fig.3.9:** The plot of the condition of a performance matrix.

By using the hyperbolic tangent as cost function in the ICA algorithm as earlier said in the previous chapter, by considering that the outputs of the STFT (that is the Fourier Transform) are complex variables, it is showed that due to the singularities at these points \( \left(k + \frac{1}{2}\right)\pi \), for \( k = 0, 1, \ldots \), the hyperbolic tangent is undefined. This will introduce numerical problems and seriously hinder convergence (See Fig.3.10 below) [16].
Fig. 3.10: The plot of hyperbolic tangent of complex number

He argued that it’s better to use hyperbolic tangent separately on the real and imaginary parts of the complex variables as [22]

\[ f(z) = \tanh(\text{Re}\{z\}) + \tanh(\text{Im}\{z\})i. \]  \hspace{1cm} (3.64)

Fig. 3.11: The plot of hyperbolic tangent of parts of complex number
In this thesis, it is decided to use \( f(z) = \tanh(\text{Re}(z)) + \tanh(\text{Im}(z)) \) as cost function for complex value ICA and apply it independently to each frequency bin obtained from a STFT of inputs. From which one complex valued unmixing matrix for every frequency bin is obtained [17].

### 3.7.3 Permutation

By applying ICA algorithm for an instantaneous mixtures it is possible that the frequency components of the same source are recovered with arbitrary other. This result in permutation ambiguity and can lead us to a wrong reconstruction of the spectrum of the recovered sources.

There are many methods developed by researcher to mitigate this problem [19], but the fact is that they have some disadvantage. The methods and their set back as also enumerated are as follows:

- Using correlations between envelopes of band-passed image, but it’s setback is that it is not robust since misalignment at a frequency is propagated through consecutive frequencies.
- Direction of arrival (DOA) estimation or localization approach as termed by Sawada et al [25] is based on the basis vectors from the inverse of the separation matrix. This method lacks preciseness since the evaluation is based on approximation of the mixing system.
- Imposing constraints on demixing filter such as smoothing, gives a good result but only in a simple case, it cannot be used when mixing filter length is too long.
- Sawada et al [25] used both the correlation and localization method to form what he called Integrated method, but this increases the complexity and computational demand.
- Kim [23] uses a method of grouping the vectors of estimated frequency responses into clusters in such a way that each cluster contains frequency responses associated with the same source. He did this grouping, otherwise called clustering, by applying ICA on estimated frequency responses. Kim argued that this method works well and does not require any prior information such as the geometric configuration of microphones arrays or distances between the sources and microphones.
3.7.4 ICA Based Clustering

Consider the complex valued ICA algorithm converge on each of the frequency bin, outputting the unmixing matrix, thus by having \( y(f, \tau) = W(f) x(f, \tau) \). Calculating the pseudo inverse \( W^+ \) (which is reduced to the inverse \( W^{-1} \) if \( N = M \), just as in the case as stated) of \( W \)

\[
A = [a_1, a_2, \ldots, a_N] = W^{-1}
\]  

(3.65)

where \( a_i = [a_{i1}, a_{i2}, \ldots, a_{im}]^T \) are the columns of the inverse of the separating matrix \( W \) otherwise called the basis vectors. This inverse is also the estimated mixing matrix. Thus the sensor sampled vector \( x(\tau) \) can be represented by a linear combination of basis is vectors

\[
x(f, \tau) = \sum_{i=1}^{N} a_i(f) y_i(f, \tau)
\]

(3.66)

Construct a data matrix \( \tilde{X} \) from the basis matrix \( A \) of each frequency bin,

\[
\tilde{X} = \begin{bmatrix} A_1 & \cdots & \cdots & A_L \end{bmatrix}
\]

(3.67)

where \( A_k = A \left( \frac{(k-1)f_s}{L} \right) \).

The ICA algorithm is then applied on \( \tilde{X} \) to get the following decomposition

\[
\tilde{X} = \tilde{A} \tilde{S}
\]

(3.68)

Where \( \tilde{A} \) is a complex \( m \times n \) matrix denoting the ICA basis matrix and \( \tilde{S} \) is a complex \( n \times nL \) denoting the encoding variable matrix associated with \( \tilde{X} \) (Please note that this is not the sensor image). [20]

Clustering (that is grouping into cluster of the basis vector, so that each cluster will contain frequency response associated with the same source) is then done by considering absolute values of encoding variables that represent the contribution of basis vectors. Like in our case where \( N = 2 \), if we consider a point \( \tilde{x}_i \) we have two basis vectors \( \tilde{a}_1, \tilde{a}_2 \) and associated
encoding variables $\tilde{s}_i = [\tilde{s}_{1,i}, \tilde{s}_{2,i}]^T$. $\tilde{x}_i$ is assigned to cluster1, if $|\tilde{s}_{1,i}| > |\tilde{s}_{2,i}|$ and is assigned to cluster2 otherwise. Since $N = 2$, it means that two consecutive data points $\tilde{x}_i$ and $\tilde{x}_{i+1}$ cannot be in the same cluster, so if $\tilde{x}_i$ is in cluster one then $\tilde{x}_{i+1}$ should be in cluster two.

From these clusters, pick the columns based on the frequency, that is by assuming that the first frequency bin and pick the first columns form each of the cluster to form a matrix by inverting it to get back the unmixing matrix and then multiply with the first frequency bin of the mixture to get the separated source [21].

3.7.5 Scaling

Before the image is recovered (i.e. before the inverse Fourier transform), by trying to resolve the scaling problem using the method as proposed by Minje Kim and Seungjin Choi. Here Sawada et al [22] multiplied the diagonal values of the basis matrix with the separation matrix for each frequency bin. The reason behind this method is that the product of this multiplication will be a matrix with a normalized diagonal. And this will normalize the separated outputs for each frequency. Before applying this method, it is tested on the ICA outputs of instantaneous mixture, listening to the sound output, before and after applying this method, proving that this method actually works well. ISTFT is then applied to recover the image from frequency domain.

3.8 CONCLUSIONS

In this thesis the study and implementation of the Frequency domain BSS using ICA, which can be used in applications like teleconference, hearing-aids etc is performed. This is a very attractive method in solving convolutive mixture images which is the only form of mixtures applicable in real time applications. By looking at the formulation of the ICA algorithm based on higher order statistics which lead to talk about some statistical terms. The frequency domain ICA requires that the time domain mixture image be transformed into Frequency Domain and this is done with Short Time Fourier Transform (STFT). This gave us an edge to study the Short Time Fourier Transform (STFT), its methods, applications and benefits.
The study of Frequency Domain BBS/ICA didn’t just go without difficulties and problems, the most prominent one is the familiar problem with Frequency domain BSS/ICA and that is the permutation and Scaling alignment problem which is our main target in this work to resolve. Different methods proposed by some researchers have been used to mitigate the problem, and applied one of them in the proposed approach.

The results related to the methods proposed by G Vijay Kumar and Dr GV Raju [26] for permutation resolution works well on “offline” or artificial mixture. The scaling resolution method proposed by Makino et al also works well. It is finally recommended that more work should be done in the permutation problem (with more regards to speed and simplicity) and to increase the speed of convergence of the ICA algorithm, methods like the FastICA should be considered. The proposed BSS method is applied on the given set of fMRI images for analysis to determine the early detection of tumor.

3.9 REFERENCES


26. G Vijay Kumar, Dr GV Raju, Early Brain Cancer Detection implementing Neural Networks, ETICT – 08, December 19 – 20, 2008