Investigating Low-complexity Architectural Issues under UBSS

P Sreenivasa Reddy

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

Department of Electrical Engineering

June 2014

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

P. Sreenivasa Reddy

(Signature)

(P Sreenivasa Reddy)

EE12P1007

(Roll No.)

Approval Sheet

This Thesis entitled "Investigating Low complexity Architectural Issues under UBSS" by P.SREENIVASA REDDY is approved for the degree of Master of Technology from IIT Hyderabad.

(Dr. Ch. Sobhan Babu) Member
Dept. of Computer Science and Engineering
Indian Institute of Technology Hyderabad

(Dr. Ashudeb Dutta) Member
Dept. of Electrical Engineering
Indian Institute of Technology Hyderabad

(Dr. Amit Acharyya) Adviser
Dept. of Electrical Engineering
Indian Institute of Technology Hyderabad

(Dr. Ketan P. Detroja) Chairman
Dept. of Electrical Engineering
Indian Institute of Technology Hyderabad

Photocopy received.
30/6/14

Acknowledgements

I would like to acknowledge my guide Dr. Amit Acharyya for his guidance in my M.tech thesis. Also I would like to acknowledge my faculties Dr. Asudeb Dutta, Dr. Shiv Govind and Dr. Shiva Rama Krishna who taught us M.Tech courses so that I got the knowledge to accomplice my M.Tech Project.

Also I would like to thank the DEITY India for partly supporting our work, under the "IOT for Smarter Healthcare" under Grant No: 13(7)/2012-CC&BT, Dated 25th Feb, 2013.

Dedication

To

Indian Institute of Technology Hyderabad

Abstract

Our Project aim is to develop a real time chip to process the sensor signals and separating the source signals, which is used in Health care like Autism. Autism is a disease which affects the child mental behavior. So If we analyze the signals form the brain so we can observe the how effectively the disease is cured. So to analyze the Autism we need EEG signals from almost 128 Leads from the scalp of child, which is difficult to do so. Thus we have to reduce the number of Leads used and at the same time we should get the all information as in the case of 128-Leads. Thus solving our problem is to solve Underdetermined Blind Source Separation (UBSS).

And in some other cases we may have only one mixture signal (M=1), which is extreme case of UBSS, from which we have to extract the unknown sources, which is called Single channel Independent Component Analysis also called SCICA. In SCICA if we have N source signals then it is called ND-SCICA.

In real time UBSS or SCICA problem we require a Digital chip which will separate the sources in real time case. So we require a chip which is High speed so that it will be suitable for real time applications and also it should be Reconfigurable so that it can work for different type of applications where the frame length of signals vary.

So first we investigated the architectural issues of Reconfigurable Discrete Hilbert Transform for UBSS where M is greater than one. Thus we proposed a high-speed and reconfigurable Discrete Hilbert Transform architecture design methodology targeting the real-time applications including Cyber-Physical systems, Internet of Things or Remote Health-Monitoring where the same chip-set needs to be used for various purposes under real-time scenario. By using this architecture we are able to get Discrete Hilbert Transform for any given M-point by re-using N-point Discrete Hilbert Transform as a kernel. Here N and M are multiple of 4 and N respectively. Subsequently we provide the architecture design details and compare the proposed architecture with the conventional state-of-the-art architecture. Thorough theoretical analysis and ex-

perimental comparison results show that the proposed design is twice as fast and reconfigurability is also achieved simultaneously.

After DHT, we proposed a new algorithm for ND-FastICA which is used for extreme case of UBSS where the number of mixture/sensor signals are only one. In this algorithm we used CORDIC based ND-FastICA which is reconfigurable so that the same chip can be used for different dimensioned FastICA.

Contents

Chapter 1

Introduction

Now a days in communication engineering and Biomedical Engineering we are facing problems in collecting information from a mixture of data. In other words we want to separate the Unknown sources (N) from the known mixture signals(M). Thus the problem is called Blind Source Separation Problem. But in real life scenario we have very less no of mixture/sensor signals from which we have to separate the unknown sources. This type of problem is called Underdetermined Blind Source Separation Problem also called UBSS.

In UBSS algorithm we have to use Discrete Hilbert Transform(DHT) to get analytical Signal. In addition that the DHT should be recongfigurable and high speed so that it will suitable for real time problems.

Discrete Hilbert Transform (DHT) has significant applications in Signal processing and Digital Communications especially where Analytical signals have to be derived from the input signals as follows,

$$a(n) = x(n) + jH\{x(n)\}$$

where $x(n), H\{x(n)\}, a(n)$ are Input signal, Hilbert transform of $x(n)$ and Analytical signal respectively. For example to compute Wigner Ville Distribution [1] for solv-

ing Underdetermined Blind Source Separation (UBSS) [3] problem, firstly, Analytical Signals has to be derived using DHT on the input signals. Similarly in Healthcare Systems for example Ultrasound image extraction uses Hilbert Transform for envelop detection [4]. In the field of Geophysics Hilbert transform plays a major role in the direct detection of hydrocarbons (oil/gas) [5]. Even in the field of Engineering Structures, Hilbert transform is used to find the Envelop detection for the detection of Damages in Building Structures [6]. In [12] DHT is used as the Minimum phase type filter for the forecasting and characterization of wind speed. In addition that in the emerging fields including cyber physical systems, internet of things, remote health monitoring applications, there is a need of separation of signals from the composite in such a way that it meets real time requirements without putting significant burden on available resources. Therefore it is important to design high speed DHT under the real-time scenario. At the same time these applications demand the multipurpose operations of the same chip set there by creating a need of Reconfigurable architecture design. There exists various DHT architectures in the transformation domain based on DFT [9] and FHT [10] which require more resources to convert from frequency domain to time domain. In time domain also there exists FIR filter [11] based DHT however it is based on causality of input signals which will have less accuracy. However these methodologies are not suitable for on-chip reconfigurable applications. Recently a systolic array based reconfigurable architecture was proposed in [8] ,but it is achieved at the cost of high processing time there by making it unsuitable for real time applications. This motivates us to propose a high-speed and reconfigurable DHT architecture design methodology targeted mainly at the real-time applications where the same chip-set can be used for various purposes depending upon different applications. Hence in this thesis we are proposing a methodology for high speed and Reconfigurable M-point DHT Architecture.

In extreme case of UBSS where the number of sources are only one which is called

SCICA problem. So we used CORDIC to solve the SCICA problem. The SCICA is algorithm can be used in Biomedical applications like Protein analysis [14]. In ND-SCICA we need to use reconfigurable FastICA so that for different case we may need to use different dimensioned FastICA. In our literature study we found the algorithm for SCICA which is proposed by C.J. James in [13]. And recently an architecture was proposed for 3D-SCICA [15] in 2013. But the 3D-SCICA is not not useful for ND-SCICA which is useful in real time application. And also the static (N is fixed for a chip) ND-FastICA is proposed in [18], but it is not suitable for ND-SCIA where we need to use reconfigurable (N can be varied) FastICA.So we proposes the ND-SCICA in which we proposed an architecture for Reconfigurable ND-FastICA which is used for different number of signals for different cases.

Chapter 2

UBSS

Underdetermined Blind Source Separation is one of the case in the Blind Source Separation Problem where the number of sensors/mixture signals (M) are less than the number of source signals(N). To solve the UBSS problem, two algorithms were proposed in 2012 [3].

The typical UBSS architecture is shown in 2.1.

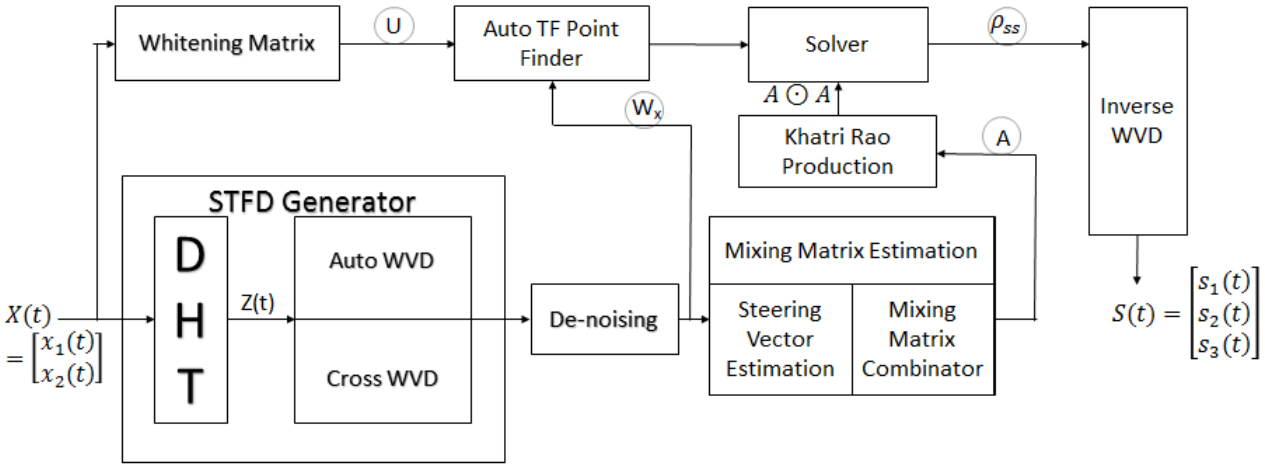

Figure 2.1: Typical UBSS Architecture

As shown in 2.1 we will give Mixture signals(M) as inputs and we will get the source signals (N M). To solve UBSS Boualem Boashash proposed some algorithms

in the text book "Time Frequency Signal Analysis and Processing" [2].

The basic equation for inputs and outputs is shown as,

$$X(n) = A \times S(n) \tag{2.1}$$

Where $X(n)$ is a Mixture matrix of order $[M \times L]$ i.e. each row represents one mixture signal of frame length L . Similarly $S(n)$ is a Source matrix of order $[N \times L]$ where each row represents one source signal. And A is Mixing matrix of order $[M \times N]$.

For UBSS M is always less than N.

As shown in 2.1, after getting M mixture signals we have to find find the analytical signal of each mixture signal so that we will get WVD of those signals as all real valued. The equation for analytical signal is shown as follows,

$$Z(n) = X(n) + jH\{X(n)\} \tag{2.2}$$

Here $H\{X(n)\}$ represents Discrete Hilbert Transform of $X(n)$. After getting Analytical signal we can solve the UBSS as shown in 2.1.

But the main challenge here is to design an architecture for DHT which should be Reconfigurable for different frame lengths and also should be high speed so that it can be useful for real time applications like biomedical signal processing. So we proposed an architecture of High speed DHT [16] which is explained in the next chapter.

Chapter 3

DHT

3.1 Theoretical Background

As we know, there are several different definitions for Hilbert transform in continuous case, which relate to different space of functions(signals), The most popular one is defined on the real line with singular kernel(relating to the theory of Hardy space on the upper half plane). In some sense, it can be proved they are equivalent. But in discrete case, the equivalence is not obvious, However as mentioned in Section-I the targeted application is on-chip real time signal processing. Therefore our focus is on discrete case.

The formulas for Discrete Analytical Signal having M (M is even) samples were given in [7] as follows, For n is even,

$$a(n) = x(n) + j\frac{2}{M}\sum_{p=0}^{M/2-1} x(2p+1)cot(\pi(n-(2p+1))/M) \tag{3.1}$$

For n is odd,

$$a(n) = x(n) + j\frac{2}{M}\sum_{p=0}^{M/2-1} x(2p)cot(\pi(n-2p)/M) \tag{3.2}$$

By observing Discrete Analytical Signal for various M points like 4,6,8... by using

(3.1) and (3.2) we can formulate the generalized formula for M-point (even number of samples/points) as follows,

$$\begin{aligned} a(n) =& x(n) + j\frac{2}{M} \sum_{p=0}^{floor(\frac{M}{4})-1} \{x\left(mod(n+M-2p-1, M)\right) \\ & - x\left(mod(n+1+2p, M)\right)\} cot\left[\frac{\pi}{M}\left(2p+1\right)\right] \end{aligned} \tag{3.3}$$

Where $M = 4, 6, 8, 10, \ldots, etc$ and $n = 0, 1, 2, \ldots, M-1$.

3.2 Proposed Methodology

In this thesis we propose a Reconfigurable DHT for M points which are multiples of N (But in systolic based Reconfigurable DHT [8] N=4). Since M is multiple of N and N is multiple of 4, without any loss of generality (3.3) can be written for DHT as follows,

$$\begin{aligned} h(n) =& \frac{2}{M} \sum_{p=0}^{\frac{M}{4}-1} \{x\left(mod(n+M-2p-1, M)\right) \\ & - x\left(mod(n+1+2p, M)\right)\} cot\left[\frac{\pi}{M}\left(2p+1\right)\right] \end{aligned} \tag{3.4}$$

Where $M = 4, 8, 12, \ldots, etc$ and $n = 0, 1, 2, \ldots, M-1$. The above equation can be written in matrix form as follows,

$$\begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(M-1) \end{bmatrix} = K \times \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(M-1) \end{bmatrix} \tag{3.5}$$

Where

$$K = \begin{bmatrix} 0 & -k_1 & 0 & -k_2 & ... & -k_{M/4} & 0 & k_{M/4} & ... & 0 & k_2 & 0 & k_1 \\ k_1 & 0 & -k_1 & 0 & ... & -k_{M/4-1} & -k_{M/4} & 0 & ... & k_3 & 0 & k_2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -k_1 & 0 & -k_2 & 0 & ... & 0 & k_{M/4} & 0 & ... & k_2 & 0 & k_1 & 0 \end{bmatrix}$$

Which is essentially a diagonal-constant matrix(Toeplitz Matrix).

$$k_i = \frac{2}{M} \times cot\left[\frac{\pi}{M}(2i-1)\right], \quad i = 1, 2, \ldots, M/4$$

Reconfigurable DHT is defined as, getting DHT for given any M-point by reusing N-point kernel for multiple times. Since in our proposed methodology, (3.4) is considered as the kernel, N is multiple of 4 and M is multiple of N. In other words,the physical interpretation would be, N (which is multiple of 4 as shown in (3.4) and will be discussed in detail in Section-IV) is chip parameter known as kernel, which designer can set while designing the chip. On the other hand, M can vary depending upon different applications but can be realized using the same chip with fixed N achieving reconfigurability and high speed as per our proposed methodology. For example considering N=8-point kernel (multiple of 4), which is fixed on a chip, that can be used to implement M=512 points UBSS system for Speech Processing application, can also be used for M=4096-point UBSS for medical applications using the same chip.

From (3.5) it is apparent that every row(except the first one), in matrix K, is one-element circular right shift of previous row. So (3.5) can be written as sub-matrices

form as follows,

$$\begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{M/N} \end{bmatrix} = \begin{bmatrix} K_1 & K_2 & \dots & K_{M/N} \\ K_{M/N} & K_1 & \dots & K_{M/N-1} \\ \vdots & \vdots & \vdots & \vdots \\ K_2 & K_3 & \dots & K_1 \end{bmatrix} \times \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{M/N} \end{bmatrix}$$
$$= \begin{bmatrix} K_1 \cdot X_1 + K_2 \cdot X_2 + \cdot + K_{M/N} \cdot X_{M/N} \\ K_{M/N} \cdot X_1 + K_1 \cdot X_2 + \cdot + K_{M/N-1} \cdot X_{M/N} \\ \vdots \\ K_2 \cdot X_1 + K_3 \cdot X_2 + \cdots + K_1 \cdot X_{M/N} \end{bmatrix} \tag{3.6}$$

Here H_i, K_i and X_i are sub-matrices of orders $N \times 1, N \times N$ and $N \times 1$ respectively drawn from (3.5). Where $i = 1, 2, \dots, M/N$ We can generalize (3.6) as,

$$H_i = \sum_{j=1}^{M/N} K_{mod(\frac{M}{N}-i+j+1, \frac{M}{N})} \times X_i$$
$$= \sum_{j=1}^{M/N} kernel(i, j) \tag{3.7}$$

Where $kernel(i, j) = K_{mod(\frac{M}{N}-i+j+1, \frac{M}{N})} \times X_i$ and $i = 1, 2, \dots, M/N$. It can be noted that the kernel is multiplication of two matrices of order $N \times N$ and $N \times 1$ which gives matrix of order $N \times 1$. So from (3.7) we can conclude that, to calculate DHT for given M samples by using a fixed kernel which does multiplication of two matrices of order $N \times N$ and $N \times 1$ and gives a matrix of order $N \times 1$ for $\left(\frac{M}{N}\right)^2$ times. It means for any given M samples we can re-use same kernel for $\left(\frac{M}{N}\right)^2$ times which brings the reconfigurability property in the proposed DHT architecture. In reconfigurable DHT we will have random samples for DHT i.e, M=N,2N,3N,4N,. . . ,etc which are multiples of N. So for given M-samples we have to re-use our only resource N-sampled kernel accordingly. So we have to select kernel inputs for given M-samples. From (3.7) we

have to define all elements for sub-matrices from (3.5) i.e, for H_i, K_i and X_i. where $i = 1, 2, \ldots, M/N$. The Sub-Matrices of H_i and X_i can be written, from (3.5), as,

$$H_i = \begin{bmatrix} h(N \times (i-1) + 0) \\ h(N \times (i-1) + 1) \\ \vdots \\ h(N \times (i-1) + N - 1) \end{bmatrix} X_i = \begin{bmatrix} x(N \times (i-1) + 0) \\ x(N \times (i-1) + 1) \\ \vdots \\ x(N \times (i-1) + N - 1) \end{bmatrix} \tag{3.8}$$

Where $i = 1, 2, \ldots, M/N$.

Now to generate the elements for K_i we have to observe the K matrix in (3.5). In the K matrix, in (3.5),because of the every row(except the first one) is one element right-circular shift of previous row, as shown in Fig.1, all the elements along axes which are parallel to the principal diagonal are same and alternative axes elements along the diagonal axis are zeros. Hence the full K matrix in (3.5) can be formed

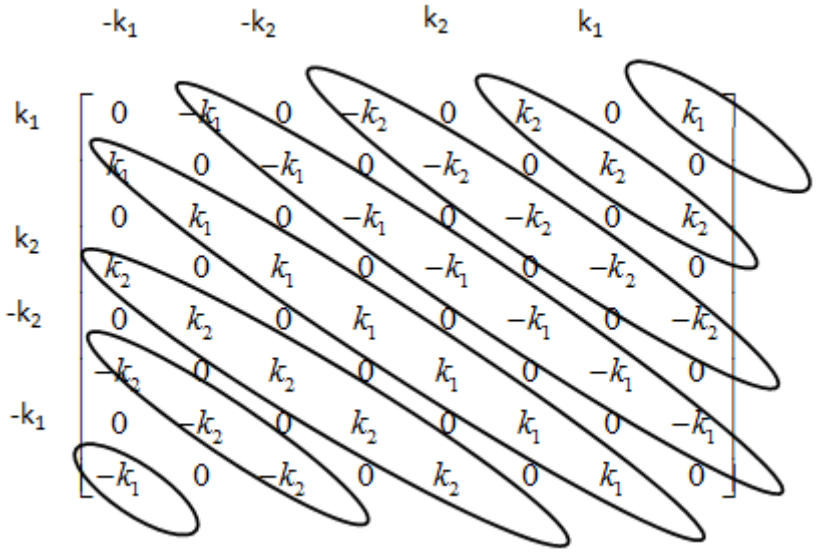

Figure 3.1: Order of elements in K-Matrix for M=8

with the M elements which are first elements of the axes (except zeros as elements of alternate axes) which are parallel to principal diagonal, instead of using all $M \times M$ elements. So we can write all the M elements starting form top right side of matrix

K to the bottom left side as shown in Fig.1 , as a set,

$$Kset = \{k_1, k_2, \ldots, k_{M/4}, -k_{M/4}, -k_{M/4-1}, \ldots, -k_1, \\ k_1, k_2, \ldots, k_{M/4}, -k_{M/4}, -k_{M/4-1}, \ldots, -k_1\} \tag{3.9}$$

The set contains total of $(M/4) \times 4 = M$ elements which are all first elements of alternate axes which are parallel to principal diagonal, starting from the top right side of matrix K to the bottom left side. Similarly we can generate elements for sub matrices in (3.6) , which will have N elements in each sub-matrix K_i of (M/N) matrices. So for M-sample DHT by using N-sample kernel we have to generate all elements for sub-matrices K_i in (3.6) from M/4 constants, as

$$Kset_i = Kset\left(\frac{M - N \times i + 2}{2} : \frac{M - N \times i + 2 \times N}{2}\right) \tag{3.10}$$

Where $i = 1, 2, \ldots, M/N$. We can generate the matrices K_i by using $Kset_i$, as $Kset_i$ is set of elements which are the first elements of all axes parallel to principal diagonal except alternative axes having zeros as the elements.

For example, to generate parameters for M=16 and N=8. Then, matrix K in (3.5) can be written as follows,

$$K = \begin{bmatrix} 0 & -k_1 & 0 & \ldots & -k_4 & 0 & k_4 & 0 & \ldots & k_1 \\ k_1 & 0 & -k_1 & \ldots & 0 & -k_4 & 0 & k_4 & \ldots & 0 \\ 0 & k_1 & 0 & \ldots & -k_3 & 0 & -k_4 & 0 & \ldots & k_2 \\ k_2 & 0 & k_1 & \ldots & 0 & -k_3 & 0 & -k_4 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -k_1 & 0 & -k_2 & \ldots & 0 & k_4 & 0 & k_3 & \ldots & 0 \end{bmatrix} \tag{3.11}$$

From (3.6) the sub-matrix can be written for M=16 and N=8 as,

$$\begin{bmatrix} H_1 \\ H_2 \end{bmatrix} = \begin{bmatrix} K_1 & K_2 \\ K_2 & K_1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{3.12}$$

Here H_1, H_2, X_1 and X_2 can be written by using (3.8). Now for K_1 and K_2 we have to find the set of elements, Kset as in (11), which are first elements of alternative axes which are parallel to principal diagonal in matrix K in (3.11). So Kset can be written as,

$$\begin{aligned} Kset = \{&k_1, k_2, k_3, k_4, -k_4, -k_3, -k_2, -k_1, \\ &k_1, k_2, k_3, k_4, -k_4, -k_3, -k_2, -k_1\} \end{aligned} \tag{3.13}$$

Now as in (3.10) $Kset_i$ can be written from above equation as,

$$\begin{aligned} Kset_1 &= Kset(5:12) = \{-k_4, -k_3, -k_2, -k_1, k_1, k_2, k_3, k_4\} \\ Kset_2 &= Kset(1:8) = \{k_1, k_2, k_3, k_4, -k_4, -k_3, -k_2, -k_1\} \end{aligned} \tag{3.14}$$

Now from above sets, the sub-matrices K_1 and K_2 in (3.12) can be written as,

$$K_1 = \begin{bmatrix} 0 & -k_1 & 0 & -k_2 & 0 & -k_3 & 0 & -k_4 \\ k_1 & 0 & -k_1 & 0 & -k_2 & 0 & -k_3 & 0 \\ 0 & k_1 & 0 & -k_1 & 0 & -k_2 & 0 & -k_3 \\ k_2 & 0 & k_1 & 0 & -k_1 & 0 & -k_2 & 0 \\ 0 & k_2 & 0 & k_1 & 0 & -k_1 & 0 & -k_2 \\ k_3 & 0 & k_2 & 0 & k_1 & 0 & -k_1 & 0 \\ 0 & k_3 & 0 & k_2 & 0 & k_1 & 0 & -k_1 \\ k_4 & 0 & k_3 & 0 & k_2 & 0 & k_1 & 0 \\ 0 & k_4 & 0 & k_3 & 0 & k_2 & 0 & k_1 \end{bmatrix} \tag{3.15}$$

and

$$K_2 = \begin{bmatrix} 0 & k_4 & 0 & k_3 & 0 & k_2 & 0 & k_1 \\ -k_4 & 0 & k_4 & 0 & k_3 & 0 & k_2 & 0 \\ 0 & -k_4 & 0 & k_4 & 0 & k_3 & 0 & k_2 \\ -k_3 & 0 & -k_4 & 0 & k_4 & 0 & k_3 & 0 \\ 0 & -k_3 & 0 & -k_4 & 0 & k_4 & 0 & k_3 \\ -k_2 & 0 & -k_3 & 0 & -k_4 & 0 & k_4 & 0 \\ 0 & -k_2 & 0 & -k_3 & 0 & -k_4 & 0 & k_4 \\ -k_1 & 0 & -k_2 & 0 & -k_3 & 0 & -k_4 & 0 \\ 0 & -k_1 & 0 & -k_2 & 0 & -k_3 & 0 & -k_4 \end{bmatrix} \tag{3.16}$$

We can observe $Kset_1, Kset_2$ in (3.14) as the set of elements which are first elements of the alternative axes parallel to principal diagonal axis, form top right side to bottom left side, of the matrices K_1, K_2 as in (3.15) and (3.16). It is to be noted that in this thesis our thrust is on the On-chip Reconfigurable High-speed DHT Architecture Design Methodology for Real time Signal Processing, therefore the computations related to the inverse DHT and the corresponding inverse k matrix are out of the

scope of this thesis.

3.3 Results and Discussions

To design the digital circuit which has N-sample kernel and can be used for any number(M) of samples/points DHT upto a maximum value M_{max} without changing the hardware of the design. The Digital Architecture for above methodology is shown in the Fig.2 as block diagram. The controller, in Fig.2(a), controls all the blocks

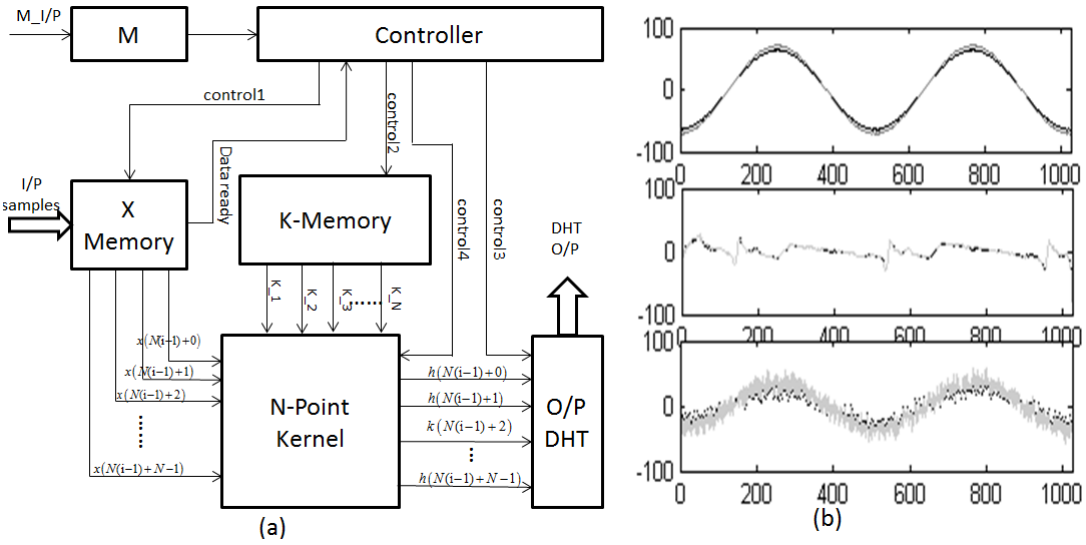

Figure 3.2: (a)proposed Architecture, (b) comparison of the conventional algorithm (black) and proposed architecture's outputs (gray).

so that they act as the reconfigurable DHT as shown in (3.7). First we have to give the input value M, so that the block works for M-point DHT. Then X_Memory block temporarily stores the M input samples/points. But in the the K_Memory, all the constants k_i which will be used for all M points (which are multiples of 4) upto M_{max}permanently. Now we have to design controller so that for each usage of kernel, kernel should get all the inputs from X_Memory block and K_Memory block as given in (3.8) and (3.10). In this way kernel should get inputs for $\left(\frac{M}{N}\right)^2$ times. The proposed architecture is also compared with the conventional(MATLAB) DHT's

output as shown in Fig. 2(b).

We synthesized the proposed architecture for N=4 and $M_{max} = 1024$, using Cadence RTL compiler UMC 90nm technology at 1MHz frequency for illustration purpose. However it can be noted that the same architecture can be synthesized under different technology libraries with different frequencies on any hardware or embedded platform. The power values, for various points M, computed using Synopsys' PrimeTime, are plotted in Fig.3(b). Here the power consumption increases as M increases, because the number operations, $\left(\frac{M}{N}\right)^2$, increases with M. Please note that the proposed architecture as shown in Fig.2(a) is not a systolic architecture.

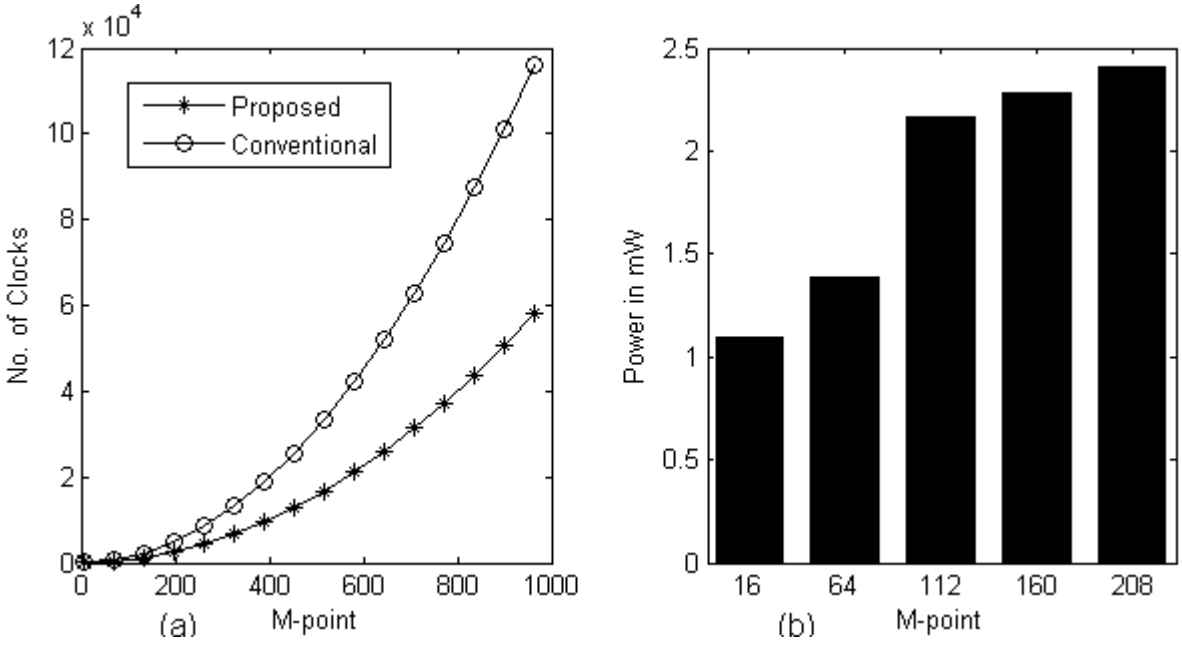

Figure 3.3: (a)Comparison of processing speed of the proposed architecture with the state-of-the art architecture [8]. (b) Power Report for Various Points DHT.

We also compared the speed in terms of the number of clocks with [8] and we are attaining double the speed of the state-of-the-art systolic array based architecture which is better than or comparable to its contemporary techniques as mentioned in Section-I, requiring kernel for $2 \times \left(\frac{M}{N}\right)^2$ times as shown in the Fig.3(a). Please note that the number of clocks shown in Fig.3(a) denotes the time taken to complete the

M-point DHT computation, where M varies from 4 to 1024 points. Since different architectures may have different numbers of computations per clock cycle, we therefore considered the number of clocks to compute M-point DHT computation instead of an individual computation needed in the DHT process.

3.4 Conclusion for DHT

Here a high-speed and reconfigurable DHT architecture design methodology is proposed using N-point kernel. This architecture is capable of calculating DHT for any number of points M. In the proposed architecture N and M are considered to be the multiple of 4 and N respectively. Our proposed architecture has been shown to have double the speed of the state-of-the art systolic array based DHT [8], thereby making it suitable for the real-time applications targeted for emerging cyber-physical systems, internet-of-things and remote healthcare applications where the same chip-set are planned to be used for various purposes.

Chapter 4

ND-SCICA

ND-SCICA is extreme case of UBSS where the number of mixture signals are only one to separate or find the N number of sources. In the real time applications like Protein spectral analysis we need a digital chip which works in real time scenario. So we proposed an architecture based on the algorithm proposed in [13].

In our proposed architecture of ND-SCICA we require ND-FastICA block which can work for different dimensioned FastICA i.e dynamically ND-FastICA can be reconfigurable accordingly for different number of signals.

4.1 Algorithm for ND-SCICA

The typical algorithm based on [13] and [14] is shown in 4.1. In the architectural design of the ND-SCICA problem the main challenge we face in the design of fpica block. Because in the process of ND-SCICA we will get different number of signals as input to the fastica block. So FastICA block should be able to reconfigure according to the number of input signals.

So we are proposing the Reconfigurable ND-FastICA block based on COrdinate Rotation DIgital Computer (CORDIC) using the idea of static ND-FastICA proposed by Amit Acharyya et. al. [18].

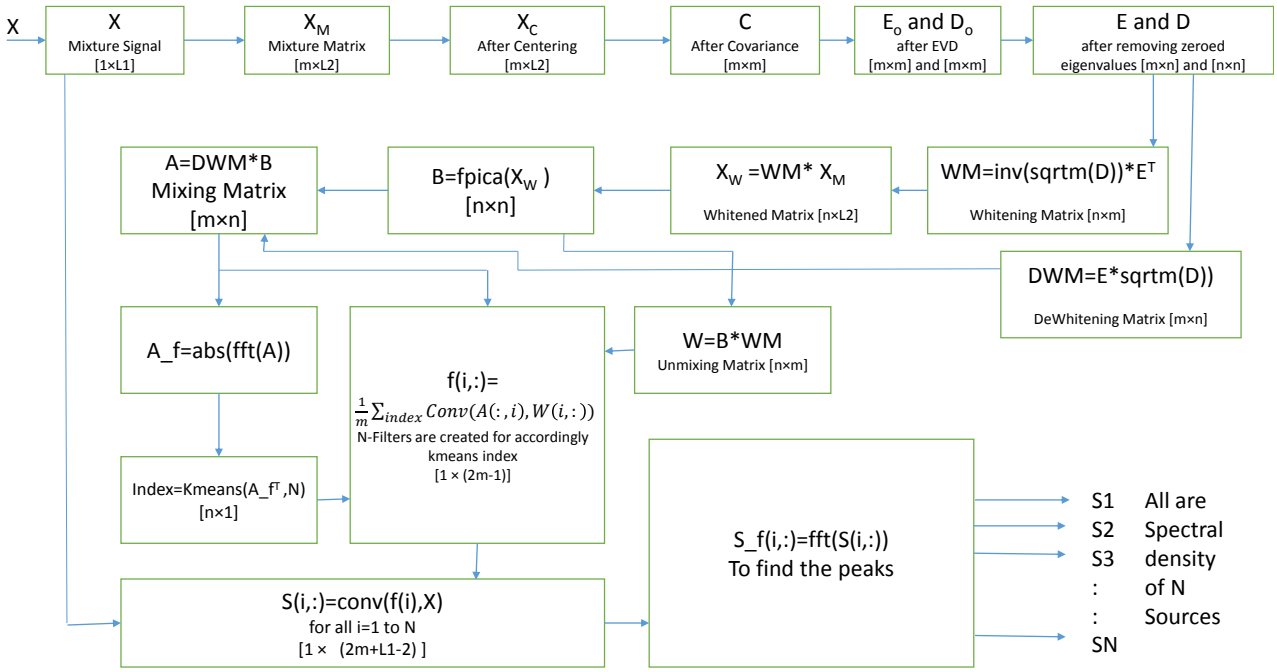

Figure 4.1: Flowchart to solve ND-SCICA

4.2 ND-FastICA

The main objective of FastICA is to find the N-Estimator vectors of length N by processing N signals. The algorithm to find the estimator vectors is proposed by Aapo Hyvrinen in [20].

The estimator vector 'w' can be calculated from X_w by using the following equation based on [20],

$$w(:,i)^{p+1} = \left(X_w \times \left(\left(X_w^T \times \underline{w(:,i)}^p\right).\hat{}3\right)\right)/L - 3 \times \underline{w(:,i)}^p \tag{4.1}$$

$$\underline{w(:,i)} = w(:,i)/norm\left(w(:,i)\right) \tag{4.2}$$

Where w is Estimator matrix of order $[N \times N]$ and X_w is Whitened Matrix of order $[n \times L]$ i.e. Whitening matrix has N- Whitened signals of each frame length is

L. And $i = 1, 2, \ldots, N$.

We can write the 4.1 as follows,

$$\begin{bmatrix} w_{1,i}^{p+1} \\ w_{2,i}^{p+1} \\ \vdots \\ w_{N,i}^{p+1} \end{bmatrix} = \begin{bmatrix} E[z_{1,j}\{z_{1,j}\underline{w}_{1,i}^{p} + z_{2,j}\underline{w}_{2,i}^{p} + \ldots + z_{N,j}\underline{w}_{N,i}^{p}\}^3] \\ E[z_{2,j}\{z_{1,j}\underline{w}_{1,i}^{p} + z_{2,j}\underline{w}_{2,i}^{p} + \ldots + z_{N,j}\underline{w}_{N,i}^{p}\}^3] \\ \vdots \\ E[z_{N,j}\{z_{1,j}\underline{w}_{1,i}^{p} + z_{2,j}\underline{w}_{2,i}^{p} + \ldots + z_{N,j}\underline{w}_{N,i}^{p}\}^3] \end{bmatrix} - 3 \times \begin{bmatrix} \underline{w}_{1,i}^{p} \\ \underline{w}_{2,i}^{p} \\ \vdots \\ \underline{w}_{N,i}^{p} \end{bmatrix} \tag{4.3}$$

Where $j = 1, 2, \ldots, L$.

Now we can write 4.3 as follows,

$$\begin{bmatrix} w_{1,i}^{p+1} \\ w_{2,i}^{p+1} \\ \vdots \\ w_{N,i}^{p+1} \end{bmatrix} = \begin{bmatrix} E[z_{1,j}\{G_{ND}\}^3] \\ E[z_{2,j}\{G_{ND}\}^3] \\ \vdots \\ E[z_{N,j}\{G_{ND}\}^3] \end{bmatrix} - 3 \times \begin{bmatrix} \underline{w}_{1,i}^{p} \\ \underline{w}_{2,i}^{p} \\ \vdots \\ \underline{w}_{N,i}^{p} \end{bmatrix} \tag{4.4}$$

Where G_{ND} is column vector of length L. So

$$G_{ND}(j) = z_{1,j}\underline{w}_{1,i}^{p} + z_{2,j}\underline{w}_{2,i}^{p} + \ldots + z_{N,j}\underline{w}_{N,i}^{p} \tag{4.5}$$

For $j = 1, 2, \ldots, L$. And from the 4.2 we can write it as,

$$\underline{w_{i,k}} = \frac{w_{i,k}}{\sqrt{w_{1,k}^2 + w_{2,k}^2 + \ldots + w_{N,k}^2}} \tag{4.6}$$

For $k = 1, 2, \ldots, N$. So our challenge is to design an architecture which is reconfigurable so that for different values of N and L it can solve the equations 4.5 and 4.6. But in [18] Amit Acharyya et. al had proposed a static architecture based on CORDIC which is fixed for a chip, So we can not use it for reconfigurable applications. Hence we are proposing ND-FastICA based on CORDIC which is reconfigurable so

that we can use it in our ND-SCICA problem. So from [18] we can write the equations 4.5 using CORDIC as follows,

$$G_{ND}(j) = Rot_x^{N-1}(z_{N,j}, Rot_x^{N-2}(z_{N-1,j}, \ldots, Rot_x^1(z_{2,j}, z_{1,j}, \theta_1,) \ldots, \theta_{N-2}), \theta_{N-1}) \tag{4.7}$$

Here $j = 1, 2, \ldots, L$ where

$$\begin{aligned} \theta_1 &= Vec_\theta^1(w_{2,i}, w_{1,i}) \\ \theta_r &= Vec_\theta^r(w_{r+1,i}, Vec_x^{r-1}(w_{r,i}, Vec_x^{r-2}(w_{r-1,i}, \ldots, Vec_x^1(w_{2,i}, w_{1,i})))) \end{aligned} \tag{4.8}$$

Here $r = 2, 3, \ldots, N-1$ and For $i = 1, 2, \ldots, N$.

Similarly we can calculate 4.6 using cordic as follows,

$$\begin{aligned} \underline{w}_{1,k} &= Rot_x^{N-1}(0, Rot_x^{N-2}(0, \ldots, Rot_x^1(0, 1, \theta_{N-1}), \ldots, \theta_2), \theta_1) \\ \underline{w}_{m,k} &= Rot_y^{N-m+1}(0, Rot_x^{N-m}(0, \ldots, Rot_x^1(0, 1, \theta_{N-1}), \ldots, \theta_2), \theta_1) \end{aligned} \tag{4.9}$$

Here $m = 2, 3, \ldots, N$. And for θ terms we can get from 4.8.

From 4.7, 4.8 and 4.9 we can observe that to get the i^th estimate vector $w\{:, i\}$ we have to follow these steps:

Step-1. Take N random values for the vector $w(:, i)$.

Step-2. Find the $N-1$ θ terms for the vector taken in step-1 (for first iteration) or from the step-6 (for second iteration onwards). i.e. we have to use VectorMode Cordic for $N-1$ times.

Step-3. Find the Normalized vector $\underline{w(:, i)}$ by using θ terms from step-2. i.e we have to use RotationMode Cordic for $N-1$ times.

Step-4. Find the vector G_{ND} by using whitened matrix X_w of order $[N \times L]$ and the θ terms from step-2. i.e we have to use RotationMode for $N-1 \times L$.

Step-5. We have to use equation 4.4 using G_{ND} vector from step-4 and $\underline{w(:,i)}$ from step-3. Thus we will get maximum Kurtosis stimator vector $w(:,i)$.

Step-6. We have to check the estimator vector $w(:,i)$ in step-6 with the vector used in previous iteration. If both are in same direction, i.e. angle between them is zero, otherwise goto step-2.

So from above steps, we can conclude that, for a single iteration we have to use total $(N-1) \times (L+1)$ times RotationMode and $(N-1)$ times Vectormode Cordic. The Matlab files for the same are available in the chapter Appendix.

Chapter 5

Conclusion

Thus I have investigated two low-complex Architectural issues under UBSS problem. First one is High Speed Reconfigurable Discrete Hilbert Transform which is used in UBSS problem where number of sensors are less than number of sources. And the other one is Reconfigurable CORDIC based FastICA algorithm which is used in UBSS problem where number of sensor signals are only one which is also called SCICA problem.

Chapter 6

Appendix

6.1 Matlab for Normalization using CORDIC

```
function [wn vect]=norm_cordic(w)

N=length(w);

vecx=zeros(1,N-1);
vect=zeros(1,N-1);
for i=1:N-1
    if i==1
        [vecx(i),vect(i)]=VectorMode(w(i+1),w(i));
    else
        [vecx(i),vect(i)]=VectorMode(w(i+1),vecx(i-1));
    end
end

rotx=zeros(1,N-1);
roty=zeros(1,N-1);

for i=1:N-1
    if i==1
        [rotx(i) roty(i)]=VectorMode(0,1,vect(N-i));
    else
        [rotx(i) roty(i)]=VectorMode(0,rotx(i-1),vect(N-i));
    end
end
wn=[rotx(N-1) fliplr(roty(1:N-1))];
wn=wn';
```

Figure 6.1: Matlab for Normalization using CORDIC

6.2 Matlab for Iteration using CORDIC

```
function s=iter_cordic(Xw,vect_in)

L=size(Xw,2);
z=Xw;

vect=vect_in;
N=length(vect)+1;

rotx=zeros(1,N-1);
roty=zeros(1,N-1);

for i=1:L
    for j=1:N-1
        if j==1
            [rotx(j) roty(j)]=RotationMode(z(j+1,i),z(j,i),vect(j));
        else
            [rotx(j) roty(j)]=RotationMode(z(j+1,i),rotx(j-1),vect(j));
        end
    end
    s(i)=rotx(j);
end
s=s';
```

Figure 6.2: Matlab for Iteration using CORDIC

References

[1] Boashash, B.; Black, P., "An efficient real-time implementation of the Wigner-Ville distribution," Acoustics, Speech and Signal Processing, IEEE Transactions on , vol.35, no.11, pp.1611-1618, Nov 1987.

[2] Boualem Boashash, "Time Frequency Signal Analysis and Processing", Elsevier, ISBN: 978-0-08-044335-5 , 2003.

[3] Shengli Xie; Liu Yang; Jun-Mei Yang; Guoxu Zhou; Yong Xiang, "Time-Frequency Approach to Underdetermined Blind Source Separation," Neural Networks and Learning Systems, IEEE Transactions on , vol.23, no.2, pp.306-316, Feb 2012.

[4] Jin Chang; Yen, J.T.; Shung, K.K., "A Novel Envelope Detector for High-Frame Rate, High-Frequency Ultrasound Imaging," Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on , vol.54, no.9, pp.1792-1801, Sep 2007.

[5] N. Sundararajan; Y. Srinivas, "FourierHilbert versus HartleyHilbert transforms with some geophysical applications", Journal of Applied Geophysics, vol. 71, no.4, pp.157-161, Aug 2010.

[6] Ching-Tai Ng, "On the selection of advanced signal processing techniques for guided wave damage identification using a statistical approach", Engineering Structures, vol.67, pp.50-60, May 2014.

[7] Elfataoui, M.; Mirchandani, G., "Discrete-time analytic signals with improved shiftability," Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on , vol.2, pp.ii 477-80, May 2004.

[8] Li Liu; Yan Zhang, "Design and Implementation of Reconfigurable Discrete Hilbert Transform Based on Systolic-Arrays," Communications and Mobile Computing (CMC), 2010 International Conference on , vol.1, pp.245-249, April 2010.

[9] Cizek, V., "Discrete Hilbert transform," Audio and Electroacoustics, IEEE Transactions on , vol.18, no.4, pp.340-343, Dec 1970.

[10] Pei, Soo-chang; Sy-Been Jaw, "Computation of discrete Hilbert transform through fast Hartley transform," Circuits and Systems, IEEE Transactions on , vol.36, no.9, pp.1251-1252, Sep 1989.

[11] Kumar, Balbir; Dutta Roy, S. C., "Design of efficient FIR digital differentiators and hilbert transformers for midband frequency ranges", International Journal of Circuit Theory and Applications, vol.17, no.4, pp.483-488, 1989.

[12] Mukhopadhyay, S.; Bhattacharya, P.; Bhattacharjee, R.; Bose, P.K., "Discrete Hilbert Transform as Minimum Phase Type Filter for the forecasting and the characterization of wind speed," Communications, Devices and Intelligent Systems (CODIS), 2012 International Conference on , pp.333-336, Dec. 2012.

[13] M.E. Daviesa; C.J. James, "Source separation using single channel ICA", ELSEVIER Signal Processing, pp.1819-1832, 2007.

[14] Mavuduru Neehar, Amit Acharyya, "Fast and Robust Extraction of Reliable Protein Signal Profiles from Mass Spectrometry Data by Introducing the Con-

cept of Single Channel ICA with Statistical Offset Correction", IEEE EMBC, 2013.

[15] Jayesh B, Amit Acharyya, et.al, "Coordinate Rotation Based Low Complexity Architecture for 3D Single Channel Independent Component Analysis", IEEE EMBC, 2013.

[16] P Sreenivasa Reddy, Suresh Mopuri, Amit Acharyya, A Recongurable High Speed Architecture Design for Discrete Hilbert Transform, IEEE Signal Processing Letters, 2014.

[17] Suresh Mopuri, P Sreenivasa Reddy, Karthik Ch, Siva Prasad A, Amit Acharyya, Siva Ramakrishna V, Low complexity Underdetermined Blind Source Separation System Architecture for Emerging Remote Healthcare Applications, IEEE EMBC, 2014.

[18] Amit Acharyya, Koushik Maharatna, Coordinate Rotation Based Low Complexity N-D FastICA Algorithm and Architecture, IEEE Transactions On Signal Processing, 2011.

[19] Milos D. Ercegovac, Tomas Lang Redundant and On-Line CORDIC : Application to Matrix Triangularization and SVD, IEEE Transactions On Computers, 1990.

[20] Aapo Hyvrinen, "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis", IEEE Trans. on Neural Networks, 10(3):626-634, 1999.