**INTRODUCTION**

**1.1 BACKGROUND**

In modern technologies like movie making, satellite communications, mass photography, medical applications and so on, captured signals represent a mixture of two or more original signals. When a person is talking in a phone beside of a railway track while train is moving then the other person will get the noise i.e. train sound while receiving the data? Such kind of mixed signals will disturb the human perception and sometimes received data may be improper.

The separation of signals is generally very challenging and has received much attention in various fields in the past few years. The separation of mixtures usually belongs to the scope of Blind Source Separation (BSS), also known as blind source separation, is the separation of a set of signals from a set of mixed signals [4]. The separation of set of signals can be achieved without the aid of information or with very little information about the source signals or the mixing process and a number of BSS based approaches have been proposed. Some of these methods introduce Independent Component Analysis (ICA), Auto De-correlation Filtering (ADF) and beam forming.

In this paper, the problem considered is the enhancement and separation of speech signals corrupted by environmental acoustic noise, interferences and other speakers using array of microphones containing at least two microphones. This work presents the implementation of the blind source separation using ICA (Independent Component Analysis). The ICA algorithm that uses wavelets is used to exploit the structure in the signals of interest and thus learn the source separation more efficiently [1]. We propose a new algorithm for blind source separation (BSS), in which frequency-domain ICA and time-domain ICA are combined to achieve a superior source-separation performances.Thus ICA can be seen as an extension to Principal Component Analysis(PCA) and Factor Analysis. ICA is a much richer technique, however, capable of finding the sources when these classical methods fail completely.

A wavelet is, as the name might suggest, a little piece of a wave. A wavelet exists only within a finite domain, and is zero-valued elsewhere. Wavelets transform of signals in the time domain (rather, assumed to be in the time domain) to a joint time-frequency domain.

The main weakness that was found in Fourier transforms was their lack of localized support, which made them susceptible to Heisenberg's Uncertainty principle. In short, this means that we could get information about the frequencies present in a signal, but not where and when the frequencies occurred. Wavelets, on the other hand, are not anywhere as subject to it.

**1.2 AIM OF THE PROJECT:**

To achieve the separation of a set of signals from a mixed signal by using wavelets with Independent Component Analysis (ICA).

**1.3 METHODOLOGY:**

As all the traditional approaches like Independent Component Analysis, auto decorrelation filtering, principle component analysis, algorithms to separate set of signals from mixed signals will result in various drawback, either the signal was segmented or the partially noise was removed with the existing ICA analysis, appreciable results were not found by implementing only ICA method for signal segmentation. So in this project we are implementing the ICA by using wavelets, in which the signal was segmented completely and also the noise was removed completely.

* 1. **SIGNIFICANCE OF THIS WORK**

**1.4.1 IMPORTANCE:**

Blind source separation using wavelets is extended to handle mixtures with not only knowing the mixing coefficients but also unknown signal sources. Such types of mixtures are more general and applicable than the previous methods. The methods used for the separation of mixed signals has a number of desirable properties like:

1. Noise will be removed efficiently.
2. Appreciable results will be found.

**1.4.2 APPLICATIONS:**

* Audio signal processing find a wide range of application in communication fields, signal analysis.
* Application in areas which include storage, level compression, data compression.
* And also in transmission equalisation ,noise cancellation, echo reverberation.
* Can be extended to speech therapy for physically challenged in medical fields.
  1. **OUTLINE OF THIS REPORT:**

This section presents brief overview of main requirements and project report organization of this work. Chapter2provides the literature review of the work. Chapter 3 provides the background theory. Chapter 4 provides the Analysis of the methodologies. Chapter 5 provides the simulation results. Chapter 6 provides the advantages .Chapter 7 provides the applications. Chapter 8 provides the conclusion and future scope. Appendix A provides the MATLAB introduction. Appendix B provides the basics of MATLAB functions. Appendix C provides the code implementation.

* + 1. **REQUIREMENTS OF THE PROJECT**

**SOFTWARE REQUIREMENT**

* MATLAB 2012b
* Math Type7.14

**LITERATURE SURVEY**

[1]. M. Akay**, “Time Frequency and Wavelets in Biomedical Signal Processing**(Book style). **party processing**”, Proc. of the 9th Int.Conference on Digital Audio Effects (DAFx-06), Montreal, Canada, September 18-20, 2006.

In signal processing and biomedical engineering, Time Frequency and Wavelets in Biomedical Signal Processing introduces time-frequency, time-scale, wavelet transform methods, and their applications in biomedical signal processing. This incorporates the most recent developments in the field to illustrate thoroughly how the use of these time-frequency methods is currently improving the quality of medical diagnosis, including technologies for assessing pulmonary and respiratory conditions, EEGs, hearing aids, MRIs, mammograms, X rays, evoked potential signals analysis, neural networks applications, among other topics.

[2]Kenneth E.Hild and David Pinto,”**Convoluted Blind Source Separation by minimizing Mutual information between segments of signals**”,IEEE transactions on Circuits and systems, regular papers, vol.52, No.10, October 2005.

A method to perform convoluted blind source separation of super-Gaussian sources by minimizing the mutual information between segments of output signals is presented. The proposed approach is essentially an implementation of an idea previously proposed by Pham. The formulation of mutual information in the proposed criterion makes use of a nonparametric estimator of Renyi's α-entropy, which becomes Shannon's entropy in the limit as α approaches 1. Since α can be any number greater than 0, this produces a family of criteria having an infinite number of members. Interestingly, it appears that Shannon's entropy cannot be used for convoluted source separation with this type of estimator. In fact, only one value of α appears to be appropriate, namely α=2, which corresponds to Renyi's quadratic entropy. Four experiments are included to show the efficacy of the proposed criterion.

[3] OzgurYilmaz and Scott Rickard,”**Blind Separation of Speech Mixtures via TimeFrequency Masking**”, IEEE transactions on signal processing, vol.52,No.7,July 2004.

Binary time-frequency masks are powerful tools for the separation of sources from a single mixture. Perfect de-mixing via binary time-frequency masks is possible provided the time-frequency representations of the sources do not overlap: a condition we call W-disjoint orthogonally. We introduce here the concept of approximate W-disjoint orthogonally and present experimental results demonstrating the level of approximate W-disjoint orthogonally of speech in mixtures of various orders. The results demonstrate that there exist ideal binary time-frequency masks that can separate several speech signals from one mixture. While determining these masks blindly from just one mixture is an open problem, we show that we can approximate the ideal masks in the case where two anechoic mixtures are provided. Motivated by the maximum likelihood mixing parameter estimators, we define a power weighted two-dimensional (2-D) histogram constructed from the ratio of the time-frequency representations of the mixtures that is shown to have one peak for each source with peak location corresponding to the relative attenuation and delay mixing parameters. The histogram is used to create time-frequency masks that partition one of the mixtures into the original sources. Experimental results on speech mixtures verify the technique.

[4] E. Visser and T. W. Lee, “**Speech enhancement using blindsource separation and two channel energy based speakerdetection**,”IEEE Int.Conf. onAcoustic, Speech, and Signal

Process. vol. 1, pp. 884–887, April 2003.

A speech enhancement scheme is presented integrating spatial and temporal signal processing methods for blind de-noising in non-stationary noise environments. In a first stage, spatially localized point sources are separated from noisy speech signals recorded by two microphones using a Blind Source Separation (BSS) algorithm assuming no a priori knowledge about the sources involved. Spatially distributed background noise is removed in a second processing step. Here, the BSS output channel containing the desired speaker is filtered with a time-varying Wiener filter. Noise power estimates for the filter coefficients are computed from desired speaker absent time-intervals identified by comparing only signal energy of separated source signals from the BSS stage. The scheme's performance is illustrated by speech recognition experiments on real recordings corrupted by babble noise and compared to conventional beam forming and single channel de-noising techniques.

[5] Nedelko, GrbicXiao-Jiao Tao, SxenNordholm, IngrarClaesson,”**Blind signal separation using over complete sub-band representation**”, IEEE Transactions on speech and audio processing, Vol.9, no.5 123–135.

In this we discusses a multirate filterbank-based extended info-max algorithm for real-world signal separation, i.e., convolved mixtures separation. Since convolution in the time domain corresponds to instantaneous mixing in the frequency domain, polyphase sub-band projection naturally becomes an efficient alternative to the Fourier transform based frequency domain approach. The online implementation proposed is featured by a simultaneous inverse channel identification in the frequency domain and signal filtering in the time domain. It is shown that an over-representation structure reduces aliasing between different bands and results in more accurateinverse channel estimates. Therefore, it provides better performance than the Fourier transform based structure in the measures of both separation and distortion. The performance limitation of the method is also evaluated in terms of the Wiener solution.

[6] Carl Taswell, Computational Tool smiths, “**What, How, and Why of Wavelet Shrinkage De-noising**”, Stanford, CA 94309-9925.

Wavelet shrinkage denoising is not smoothing (despite the use by someauthors of the term smoothingas a synonym for the term denoising). Whereas smoothing removes high frequencies and retains low frequencies, denoising attempts to remove whatever noise is present and retain whatever signal is present regardless of the frequency content of the signal. Wavelet shrinkage denoising has been theoretically proven to be nearly optimal from the following perspectives: spatial adaptation, estimation when local smoothness is unknown, and estimation when global smoothness is unknown. In effect, no alternative procedure can perform better without knowing *a* priorithe smoothness class of the signal. It is unlikely that one particular wavelet shrinkage denoising procedure will be suitable, noless optimal, for all practical problems. However, it is likely that there will be many practical problems, for which after appropriate experimentation, wavelet-based denoising with either hard or soft thresholding proves to be the most effective procedure. Estimation of the power spectrum by wavelet-based denoising of the log-periodogram may prove to be one such important application with great promises for further development.

[7] David L. Donoho, “**De-noising via soft thresholding”**. IEEE Transactions on Information Theory, 41:613-627, May 1995.

Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. Algorithms process data at different scales or resolutions. If we look at a signal (or a function) through a large “window,” we would notice gross features. Similarly, if we gaze at a signal through a small “window,” we would notice small features. The result in wavelet analysis is to see both the forest and the trees together. This makes study of wavelets interesting and useful.The data sets of many scientific experiments are corrupted with noise, either because of the data acquisition process, or because of environmental effects. A first pre-processing step in analyzing such datasets is De-noising, that is, estimating the unknown signal of interest from the available noisy data. There are several different approaches to De-noise signals and images. Despite similar visual effects, there are subtle differences between De-noising, de-blurring, smoothing and restoration.

Generally smoothing removes high frequency and retains low frequency (with blurring). De-blurring increases the sharpness signal features by boosting the high frequencies, whereas de-noising tries to remove whatever noise is present regardless of the spectral content of a noisy signal .Restoration is a kind of de-noising that tries to retrieve the original signal with optimal balancing of de-blurring and smoothing Wavelet transforms enable us to represent signals with a high degree of sparsity. This is the principle behind a non-linear wavelet based signal estimation technique known as wavelet de-noising. Wavelet de-noising attempts to remove the noise present in the signal while preserving the signal characteristics, regardless of its frequency content. Wavelet de-noising must not be confused with smoothing. As already mentioned smoothing only removes the high frequencies and retains the lower ones.

[8] William Addison and Stephen Roberts, “**Blind Source Separation with Non Stationary Mixing Using Wavelets**, “Pattern Analysis Research Group, The University ofLiverpool,2006.

The problem of blind source separation in the situation where the mixing process is dynamic. We first present a new ICA algorithm for the static mixing problem that exploits a wavelet representation of the signals. This outperforms standard ICA in our experiments thus allowing the unmixing to be estimated from a smaller number of samples. We use this to create a sliding window based algorithm that is capable of tracking the dynamics a non-stationary mixing process in the blind source separation problem. An effective initialization for each new window is calculated based on a smoothed estimate of the unmixing process learnt in previous windows. This reduces the computation required for updating each window and reduces the chance of the algorithm falling into undesirable local minima of the cost function. The efficacy of the algorithm is demonstrated on some simulated data using artificially mixed audio sources.

[9] RobiPolikar, “**The Engineer’s Ultimate Guide to Wavelet Analysis**,” Hosted by Rowan University, College of Engineering Web Servers, Last major updates January 2001.

Wavelet transform provides the time-frequency representation. (There are other transforms which gives this information too, such as short time Fourier transform, Wigner distributions, etc.). Often times a particular spectral component occurring at any instant can be of particular interest. In these cases it may be very beneficial to know the time intervals these particular spectral components occur.

Wavelet transform is capable of providing the time frequency information simultaneously, hence giving a time-frequency representation of the signal. How wavelet transform works is completely a different fun story, and should be explained after **SHORT TIME FOURIER TRANSFORM (STFT).** The WT was developed as an alternative to the STFT. WT developed to overcome some resolution related problems of the STFT.

[10] George Tzanetakis, Georg Essl, Perry Cook, “**Audio Analysis using the Discrete Wavelet Transform**,” Computer Science Department also Music Department, Princeton University.

The Discrete Wavelet Transform (DWT) is a transformation that can be used to analyze the temporal and spectral properties of non-stationary signals like audio. In this paper we describe some applications of the DWT to the problem of extracting information from non-speech audio. More specifically automatic classification of various types of audio using the DWT is described and compared with other traditional feature extractors proposed in the literature. In addition, a technique for detecting the beat attributes of music is presented. Both synthetic and real world stimuli were used to evaluate the performance of the beat detection algorithm.

[11] AkshayDayal, John Stein Bauer, Angela Qian and Mark Eastaway, “**Blind source separation via ICA: Math behind Method**, “version 1.1: Dec 19, 2007 8:41 PM US/Central.

The independent component analysis algorithm allows two source signals to be separated from two mixed signals using statistical principles of independence and non-Gaussianity. The ICA assumes that the value of each source at any given time is a random variable. It also assumes that each source is statistically independent, meaning that the values of one source cannot be correlated to values in any other sources. With these assumptions ICA allows us to separate source signals from mixtures of these source signals. The algorithm requires that there be as many as input signals. ICA can only handle linear mixtures that can be represented in the form X=As. The algorithm cannot accurately guess the independent sources if the sources are out of phase in the mixtures or if the mixtures have other nonlinear features.

**BACKGROUND THEORY**

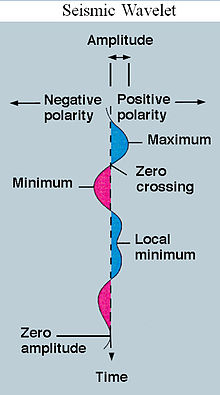
**3.1 WAVELETS**

**3.1.1 INTRODUCTION**

Wavelets are one or a few functions whose integer translations and dilations can generate a basis for a Hilbert space. The concept of wavelets was introduced in the 1980's and has since been generalized and extended in many directions. The theory and applications have been continuously developed. One of its significant features is that it provides a systematical approach for designing various filters and filter banks for signal and image processing. Another feature is that wavelets leads to the theory of multi-resolution approximation (MRA). Wavelets and MRA have found many applications in most areas of science and technology.

EXAMPLE: Astronomy, electric engineering, fuzzy logic, geo science, medical imaging, physics, and statistics. Wavelets have become an important subject in applied mathematics, approximation theory, numerical analysis and harmonic analysis.

**Wavelet** is a [wave](http://en.wikipedia.org/wiki/Wave)-like [oscillation](http://en.wikipedia.org/wiki/Oscillation) with an [amplitude](http://en.wikipedia.org/wiki/Amplitude) that begins at zero, increases, and then decreases back to zero. It can typically be visualized as a "brief oscillation" like one might see recorded by a [seismograph](http://en.wikipedia.org/wiki/Seismograph) or [heart monitor](http://en.wikipedia.org/wiki/Heart_monitor). Generally, wavelets are purposefully crafted to have specific properties that make them useful for [signal processing](http://en.wikipedia.org/wiki/Signal_processing).

[](http://en.wikipedia.org/wiki/File:Seismic_Wavelet.jpg)

**Fig.3.1 Seismic Wavelet**

Wavelets can be combined, using a "reverse, shift, multiply and integrate" technique called [convolution](http://en.wikipedia.org/wiki/Convolution), with portions of a known signal to extract information from the unknown signal. As a mathematical tool, wavelets can be used to extract information from many different kinds of data, including but certainly not limited to audio signals and images. Sets of wavelets are generally needed to analyze data fully. A set of "complementary" wavelets will decompose data without gaps or overlap so that the decomposition process is mathematically reversible. Thus, sets of complementary wavelets are useful in [wavelet based compression](http://en.wikipedia.org/wiki/Wavelet_compression)/decompression algorithms where it is desirable to recover the original information with minimal loss.

In formal terms, this representation is a [wavelet series](http://en.wikipedia.org/wiki/Wavelet_series) representation of a [square-integrable function](http://en.wikipedia.org/wiki/Square-integrable_function) with respect to either a [complete](http://en.wikipedia.org/wiki/Complete_orthogonal_system#Incomplete_orthogonal_sets), [orthonormal](http://en.wikipedia.org/wiki/Orthonormal) set of [basic functions](http://en.wikipedia.org/wiki/Basis_function), or an [over complete](http://en.wikipedia.org/wiki/Complete_orthogonal_system#Incomplete_orthogonal_sets) set or [frame of a vector space](http://en.wikipedia.org/wiki/Frame_of_a_vector_space), for the [Hilbert space](http://en.wikipedia.org/wiki/Hilbert_space) of square integrable functions.

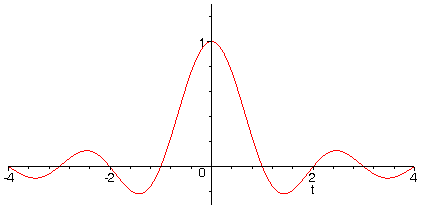
**3.1.2 WAVELET THEORY:**

Wavelet theory is applicable to several subjects. All wavelet transforms may be considered forms of [time-frequency representation](http://en.wikipedia.org/wiki/Time-frequency_representation) for [continuous-time](http://en.wikipedia.org/wiki/Continuous-time) (analog) signals and so are related to [harmonic analysis](http://en.wikipedia.org/wiki/Harmonic_analysis). Almost all practically useful discrete wavelet transforms use [discrete-time](http://en.wikipedia.org/wiki/Discrete-time) [filter banks](http://en.wikipedia.org/wiki/Filterbank). These filter banks are called the wavelet and scaling coefficients in wavelets nomenclature. These filter banks may contain either [finite impulse response](http://en.wikipedia.org/wiki/Finite_impulse_response) (FIR) or [infinite impulse response](http://en.wikipedia.org/wiki/Infinite_impulse_response) (IIR) filters. The wavelets forming a [continuous wavelet transform](http://en.wikipedia.org/wiki/Continuous_wavelet_transform) (CWT) are subject to the [uncertainty principle](http://en.wikipedia.org/wiki/Fourier_uncertainty_principle) of Fourier analysis respective sampling theory: Given a signal with some event in it, one cannot assign simultaneously an exact time and frequency response scale to that event. The product of the uncertainties of time and frequency response scale has a lower bound. Thus, in the [scale-o-gram](http://en.wikipedia.org/wiki/Scaleogram) of a continuous wavelet transform of this signal, such an event marks an entire region in the time-scale plane, instead of just one point. Also, discrete wavelet bases may be considered in the context of other forms of the uncertainty principle.

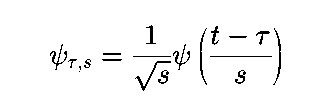
**3.1.2.1 DEFINITION:**

The basic idea of wavelet transform is to represent any arbitrary function as a superposition of a set of such wavelets or basic functions. These basis functions or baby wavelets

are obtained from a single prototype wavelet called the mother wavelet, by dilation or contraction (scaling) and translation (shifts) [2]. In wavelet transform the basis function are wavelets. Wavelets tend to be irregular and symmetric. All wavelet functions are derived from a single mother wavelet, w(t). This wavelet is a small wave or pulse as shown in figure below.



**Fig.3.2 Mother Wavelet**



Where ψ*T,S* is called the mother wavelet, *T* is the shifting parameter and *s* is the scaling parameter.

The wavelets are called orthogonal when their inner products are zero. The smaller the scaling factor is, the wider the wavelet is. Wide wavelets are comparable to low frequency sinusoids and narrow wavelets are comparable to high frequency sinusoids [2].

At the beginning of this paper, background information such as denoising using wavelets and separation of set of sources from a mixed signals are given. Techniques related to signal separation are also briefly discussed. The mid-portion of the paper focuses on the wavelet transform and wavelet denoising and 1-level decomposition of a mixed signal at transformation.

|  |
| --- |
|  |

### 3.1.2.2 SCALING FILTER:

An orthogonal wavelet is entirely defined by the scaling filter a lowpass [finite impulse response](http://en.wikipedia.org/wiki/Finite_impulse_response) (FIR) filter of length 2Nand sum 1. In [bi-orthogonal](http://en.wikipedia.org/wiki/Biorthogonal_system) wavelets, separate decomposition and reconstruction filters are defined. For analysis with orthogonal wavelets the high pass filter is calculated as the [quadrature mirror filter](http://en.wikipedia.org/wiki/Quadrature_mirror_filter) of the low pass, and reconstruction filters are the time reverse of the decomposition filters.

Daubechies and Symlet wavelets can be defined by the scaling filter.

### 3.1.2.3 SCALING FUNCTION:

Wavelets are defined by the wavelet function ψ(*t*) (i.e. the mother wavelet) and scaling function φ(*t*) (also called father wavelet) in the time domain.

The wavelet function is in effect a band-pass filter and scaling it for each level halves its bandwidth. This creates the problem that in order to cover the entire spectrum, an infinite number of levels would be required. The scaling function filters the lowest level of the transform and ensures all the spectrum is covered.

For a wavelet with compact support, φ(*t*) can be considered finite in length and is equivalent to the scaling filter *g*. Meyer wavelets can be defined by scaling functions.

### 3.1.3WAVELET FUNCTION:

The wavelet only has a time domain representation as the wavelet function ψ(*t*).Wavelets are a class of a functions used to localize a given function in both space and scaling. A family of wavelets can be constructed from a function ψ(x), sometimes known as a "mother wavelet," which is confined in a finite interval. "Daughter wavelets" ψa,b(x) are then formed by translation (b) and contraction (a). Wavelets are especially useful for compressing image data, since a [wavelet transform](http://mathworld.wolfram.com/WaveletTransform.html) has properties which are in some ways superior to a conventional [Fourier transform](http://mathworld.wolfram.com/FourierTransform.html). An individual wavelet can be defined by

|  |  |
| --- | --- |
| psi^(a,b)(x)=|a|^(-1/2)psi((x-b)/a). | (1) |

Then

|  |  |
| --- | --- |
| W_psi(f)(a,b)=1/(sqrt(a))int_(-infty)^inftyf(t)psi((t-b)/a)dt, | (2) |

and [Calderón's formula](http://mathworld.wolfram.com/CalderonsFormula.html) gives

|  |  |
| --- | --- |
| f(x)=C_psiint_(-infty)^inftyint_(-infty)^infty<f,psi^(a,b)>psi^(a,b)(x)a^(-2)dadb. | (3) |

A common type of wavelet is defined using [Haar functions](http://mathworld.wolfram.com/HaarFunction.html).

**3.1.4 WAVELET FAMILIES:**

There are different types of wavelet families whose qualities vary according to several criteria.The main criteria are: The supportive of http://radio.feld.cvut.cz/matlab/toolbox/wavelet/psi.gif, http://radio.feld.cvut.cz/matlab/toolbox/wavelet/ch06_a51.gif,(and http://radio.feld.cvut.cz/matlab/toolbox/wavelet/phi.gif, http://radio.feld.cvut.cz/matlab/toolbox/wavelet/ch06_a59.gif) the speed of the convergence to 0 of these functions (http://radio.feld.cvut.cz/matlab/toolbox/wavelet/ch06_16a.gifor http://radio.feld.cvut.cz/matlab/toolbox/wavelet/ch06_26a.gif), when the time t or the frequency ω goes to infinity, which qualifies both time and frequency localizations.

1. The symmetry, which is useful in avoiding de-phasing in image processing.
2. The number of vanishing moments for http://radio.feld.cvut.cz/matlab/toolbox/wavelet/psi.gif or for http://radio.feld.cvut.cz/matlab/toolbox/wavelet/phi.gif (if it exists), which is useful for compression purposes.
3. The regularity, which is useful for getting nice features, like smoothness of the reconstructed signal or image, and for the estimated function in nonlinear regression analysis.

These are associated with two properties that allow fast algorithm and space-saving coding:

1. The existence of a scaling function http://radio.feld.cvut.cz/matlab/toolbox/wavelet/phi.gif
2. The orthogonality or the bi-orthogonality of the resulting analysis

They may also be associated with these less important properties:

1. The existence of an explicit expression
2. The ease of tabulating
3. The familiarity with use.

Typing waveinfo in command line mode displays a survey of the main properties of all wavelet families available in the toolbox.

Note that thehttp://radio.feld.cvut.cz/matlab/toolbox/wavelet/phi.gifand http://radio.feld.cvut.cz/matlab/toolbox/wavelet/psi.gif functions can be computed using wave function, the filters are generated using w filters. We provide definition equations for several wavelets. Some

are given explicitly by their time definition, others by their frequency definition, and still others by their filter. The table below outlines the wavelet families included in the toolbox.

|  |  |
| --- | --- |
| **Wavelet Family Short Name** | **Wavelet Family Name** |
| 'haar' | Haar wavelet. |
| 'db' | Daubechies wavelets. |
| 'sym' | Symlets. |
| 'coif' | Coiflets. |
| 'bior' | Bi-orthogonal wavelets. |
| 'rbio' | Reverse bi-orthogonal wavelets. |
| 'meyr' | Meyer wavelet. |
| 'dmey' | Discrete approximation of Meyer wavelet. |
| 'gaus' | Gaussian wavelets. |
| 'mexh' | Mexican hat wavelet. |
| 'morl' | Morlet wavelet. |
| 'cgau' | Complex Gaussian wavelets. |
| 'shan' | Shannon wavelets. |
| 'fbsp' | Frequency B-Spline wavelets. |
| 'cmor' | Complex Morlet wavelets. |

**Table 3.1Wavelet Families**

**3.1.5 SOME APPLICATION OF WAVELETS:**

Wavelets are a powerful statistical tool which can be used for a wide range of applications, namely

1. Signal processing

2. Data compression

3. Smoothing and image de-noising

4. Fingerprint verification

Biology for cell membrane recognition, to distinguish the normal from the pathological membranes

1. DNA analysis, protein analysis
2. Blood-pressure, heart-rate and ECG analyses
3. Finance (which is more surprising), for detecting the properties of quick variation of values
4. In Internet traffic description, for designing the services size
5. Industrial supervision of gear-wheel
6. Speech recognition
7. Computer graphics and multi fractal analysis
8. Many areas of physics have seen this paradigm shift, including molecular dynamics, astrophysics, optics, turbulence and quantum mechanics. Wavelets have been used successfully in other areas of geophysical study.

Orthonormal wavelets, for instance, have been applied to the study of atmospheric layer turbulence. In one study by J.F. Howell and L. Mahrt, turbulence measurements were taken over a nine-hour period and analyzed using wavelet decomposition. In another study by Brunet and Collinear, turbulence data recorded over a corn crop was analyzed using the wavelet transform.

Wavelets have also been used to analyze seafloor bathymetry or the topography of the ocean floor. In one study by Sarah Little, the use of wavelet analysis revealed patterns, trends, and structures that may be overlooked in raw data. Also, the use of methods like local oracles allowed for separation of data in regions of interest.

Several other geophysical applications such as analysis of marine seismic data and characterization of hydraulic conductivity distributions have also been used. The usefulness of wavelets in data analysis is clear, particularly in the field of geophysics, where large and cumbersome data sets abound. Studies such as the atmospheric layer turbulence and corn crop turbulence have further shown the proficiency of wavelets in the analysis of time-dependent data sets Christensen, Debnath, Meyer.

**3.2 WAVELET DENOISING:**

Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. Algorithms process data at different scales or resolutions. The data sets of many scientific experiments are corrupted with noise, either because of the data acquisition process, or because of environmental effects. A first pre-processing step in analyzing such datasets is denoising, that is, estimating the unknown signal of interest from the available noisy data. There are several different approaches to denoise signals. Generally smoothing removes high frequency and retains low frequency (with blurring). De-blurring increases the sharpness signal features by boosting the high frequencies, whereas denoising tries to remove whatever noise is present regardless of the spectral content of a noisy signal.

Wavelet shrinkage denoising is not smoothing (despite the use by some authors of the term smoothing as a synonym for the term denoising). Whereas smoothing removes high frequencies and retains low frequencies, denoising attempts to remove whatever noise is present and retain whatever signal is present regardless of the frequency content of the signal[6].

When we denoise music corrupted by noise, we would like to preserve both the treble and the bass. Second, it is denoising by shrinking (i.e., nonlinear soft thresholding) in the wavelet transform domain. Third, it consists of three steps:

* + - * 1. a linear forward wavelet transform,
        2. a non-linear Shrinkage denoising, and
        3. a linear inverse wavelet transform.

Because of the nonlinear shrinking of coefficients in the transform domain, this procedure is distinct from those denoising methods that are entirely linear [11].

Finally, wavelet shrinkage denoising is considered a non-parametric method. Thus, it is distinct from parametric methods in which parameters must be estimated for a particular model that must be assumed *a* priori. (For example, the most commonly cited parametric method is that of using least squares to estimate the parameters a and b in the model y=ax+b.).

Restoration is a kind of denoising that tries to retrieve the original signal with optimal balancing of de-blurring and smoothing Wavelet transforms enable us to represent signals with a high degree of sparsity. This is the principle behind a non-linear wavelet based signal estimation technique known as wavelet denoising. Wavelet denoising attempts to remove the noise present in the signal while preserving the signal characteristics, regardless of its frequency content. Wavelet denoising must not be confused with smoothing. As already mentioned smoothing only removes the high frequencies and retains the lower ones. The principle work on denoising is based on thresholding the DWT of the signal. The method relies on the fact that noise commonly manifests itself as fine-grained structure in the signal, and WT provides a scale-based decomposition. Thus, most of the noise tends to be represented by the wavelet coefficients at finer scales. Discarding these coefficients would result in a natural filtering out of noise on the basis of scale [11].

Because the coefficients at such scale also tend to be the primary carriers of edge information, this method thresholds the wavelet coefficients to zero if their values are below a threshold. These coefficients are mostly those corresponding to the noise. The edge related coefficients of the signal on the other hand, are usually above the threshold. An alternative approach to hard thresholding is the soft thresholding which leads to less severe distortion of the signal of interest. Several approaches have been suggested for setting the threshold for each band of the wavelet decomposition. A common approach is to compute the sample variance of the coefficients in a band and set the threshold to some multiple of the deviation.

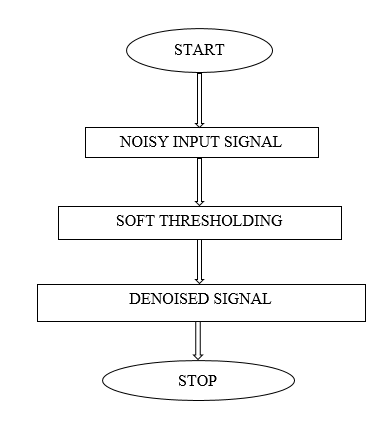
An alternative approach to hard thresholding is the soft thresholding, which leads to less severe distortion of the signal of interest. Several approaches have been suggested for setting the threshold for each band of the wavelet decomposition. A common approach is to compute the sample variance of the coefficients in a band and set the threshold to some multiple of the deviation. Wavelet denoising has wide range of application in signal processing as well as other fields. The signals may be one-dimensional, two-dimensional and three-dimensional. They carry useful information. Denoising (noise reduction) is the first step in many applications.

Hard threshold is a “keep or kill” procedure and is more intuitively appealing. The alternative, soft thresholding, shrinks coefficients above the threshold in absolute value. While at first sight hard thresholding may seem to be natural, the continuity of soft thresholding has some advantages. It makes algorithms mathematically more tractable. Sometimes, pure noise coefficients may pass the hard threshold and appear as annoying ‟blips’’ in the output. Soft thresholding shrinks these false structures. The choice of threshold is a fundamental issue. A very large threshold cuts too many coefficients, resulting in an over smoothing. Conversely, a too small threshold value allows many coefficients to be included in reconstruction, giving a wiggly, under smoothed estimate.

With regard to wavelet shrinkage denoising, the theoretical justifications and arguments in its favor remain highly compelling. The procedure does not require any assumptions about the nature of the signal, permits discontinuities and spatial variation in the signal, and exploits the spatially adaptive multi resolution features essential to the wavelet transform. Furthermore, the procedure exploits the fact that the wavelet transform maps white noise in the signal domain to white noise in the transform domain. Thus, while signal energy becomes more concentrated into fewer coefficients in the transform domain, noise energy does not. It is this important principle that enables the separation of signal from noise.

Wavelet shrinkage denoising has been theoretically proven to be nearly optimal from the following perspectives: spatial adaptation, estimation when local smoothness is unknown, and estimation when global smoothness is unknown. In effect, no alternative procedure can perform better without knowing a priori the smoothness class of the signal.

It is unlikely that one particular wavelet shrinkage denoising procedure will be suitable, no less optimal, for all practical problems. However, it is likely that there will be many practical problems, for which after appropriate experimentation, wavelet-based denoising with either hard or soft thresholding proves to be the most effective procedure. Estimation of the power spectrum by wavelet-based denoising of the log-periodogram may prove to be one such important application with great promise for further development.



**Fig.3.3Denoising Procedure**

**3.3 WAVELET TRANSFORM:**

Wavelet transforms have become increasingly important in image compression since wavelets allow both time and frequency analysis simultaneously. This paper investigates the fundamental concept behind the wavelet transform and provides an overview of some improved algorithms on the wavelet transform. The latter part of this paper emphasize on lifting scheme which is an improved based on the wavelet transform.

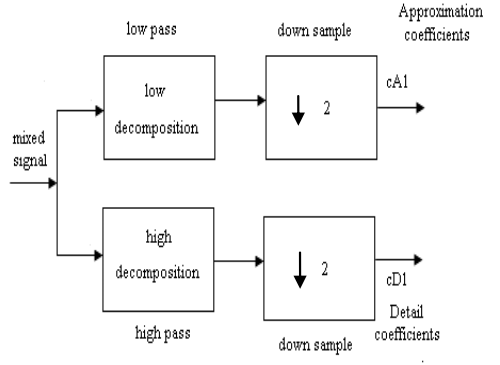
The wavelet transform plays an extremely crucial role in image compression. For image compression applications, wavelet transform is a more suitable technique compared to the Fourier transform. Fourier transform is not practical for computing spectral information because it requires all previous and future information about the signal over the entire time domain and it cannot observe frequencies varying with time because the resulting function after Fourier transform is a function independent of time. On the other hand, wavelet transforms are based on wavelets which are varying frequency in limited duration. Due to the practicality of the wavelet transforms, this research paper is written to investigate the properties and the improvements that can be made to enhance the performance of the wavelet transforms.

**3.3.1DEFINITION**

Wavelets are mathematical functions defined over a finite interval and having an average value of zero that transform data into different frequency components, representing each resolution matched to its scale.

**3.3.2 DISCRETE WAVELET TRANSFORM (DWT)**

In [numerical analysis](http://en.wikipedia.org/wiki/Numerical_analysis) and [functional analysis](http://en.wikipedia.org/wiki/Functional_analysis), a **discrete wavelet transform** (DWT) is any [wavelet transform](http://en.wikipedia.org/wiki/Wavelet_transform) for which the [wavelets](http://en.wikipedia.org/wiki/Wavelet) are discretely sampled. As with other wavelet transforms, a key advantage it has over [Fourier transforms](http://en.wikipedia.org/wiki/Fourier_transform) is temporal resolution, it captures both frequency and location information (location in time).Dwt is well-organized algorithm for extracting original information and spectral properties of non-stationary signals.

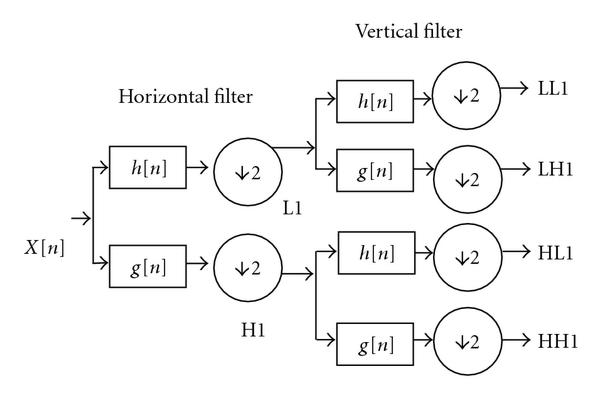


**Fig 3.4 Block Diagram of Discrete Wavelet Transform**

Where **↓2** denotes down sampling and keeps even indexed terms.

The first DWT was invented by the Hungarian mathematician [Alfred Haar](http://en.wikipedia.org/wiki/Alfr%C3%A9d_Haar). For an input represented by a list of 2nnumbers, the [Haar wavelet](http://en.wikipedia.org/wiki/Haar_wavelet) transform may be considered to simply pair up input values, storing the difference and passing the sum. This process is repeated recursively, pairing up the sums to provide the next scale, finally resulting in 2n-1 differences and one final sum.

In the discrete wavelet transform, an image can be analyzed by passing it through an analysis filter bank followed by decimation operation. The analysis filter consists of a low pass and high pass filter at each decomposition stage. When the image signal passes through these filters, it splits into two bands. The low-pass filter, which corresponds to an averaging operation, extracts the coarse information of the signal. The high-pass filter, which corresponds to a differencing operation, extracts the detail information of the signal. The output of the filtering operation is then decimated by two. A two-dimensional transform is accomplished by performing two separate one-dimensional transforms. First, the image is filtered along the row and decimated by two. It is then followed by filtering the sub-image along the column and decimated by two. This operation splits the image into four bands, namely LL, LH, HL and HH respectively [8].

****

## **Fig.3.5 Discrete Wavelet Decomposition**

## **3.3.3 THE HAAR WAVELET:**

## Haar (1910) and others were seeking functional expansions that would converge to explain other functions that were not the sine and cosine series of Fourier(1807). Haar transform is given as

фi*j*= ф (2jx -1)

where i=0,1,…..,2j-1

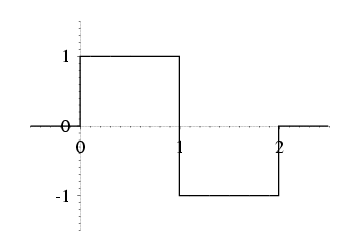
Haar began with the initial function

1; when 0<t<1/2

Ψ(t) = -1; when ½< t<1

0; otherwise

The Haar transform is based on a class of orthogonal matrices whose elements are either1, -1 or 0 multiplied by power of √2. The Haar transform is a computationally efficient transform as a transform of an N-point vector requires only 2(N-1) additions and N multiplication [1].



**Fig 3.6 Haar Wavelet**

From the fig. 3.6 it is obvious that the Haar wavelet is a real function, anti-symmetric with respect to t=1/2. The Haar wavelet is discontinuous in time. The Haar wavelet is localised in time and frequency domains.

**3.3.4 PROPERTIES:**

The Haar DWT illustrates the desirable properties of wavelets in general. First, it can be performed in O(n) operations second, it captures not only a notion of the frequency content of the input, by examining it at different scales, but also temporal content, i.e. the times at which these frequencies occur. Combined, these two properties make the [Fast wavelet transform](http://en.wikipedia.org/wiki/Fast_wavelet_transform) (FWT) an alternative to the conventional [Fast Fourier Transform](http://en.wikipedia.org/wiki/Fast_Fourier_Transform) (FFT).

**3.4 BLIND SOURCE SEPARATION:**

Blind signal separation, also known as blind source separation, is the separation of a set of signals from a set of mixed signals. The separation of set of signals can be achieved without the aid of information or with very little information about the source signals or the mixing process.

Blind source separation (BSS) is a technique for estimating individual source components from their mixtures at multiple sensors. It is called blind because we don't use any other information besides the mixtures.

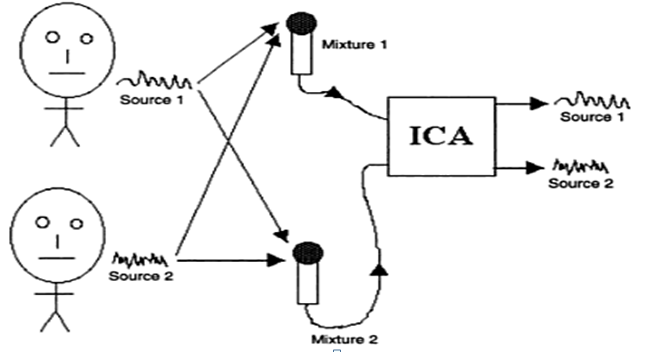
For example, imagine a room with a number of persons present and a number of microphones for recording. When one or more persons are speaking at the same time, each microphone registers a different mixture of individual speaker's audio signals. It is the task of BSS to untangle these mixtures into their sources, i.e. the individual speaker's audio signals. In general, this is a difficult problem because of several complicating factors [2].

1. Different locations of speakers and microphones in the room: the individual speaker's audio signals do not reach all microphones at the same time.
2. Room acoustics: the signal that reaches a microphone is composed of the signal that directly travels to the microphone and parts that come from room reverberations and echoes.
3. Varying distances to microphones: one or more speakers might be moving. This makes the mixing time dependent.

If the number of sensors is larger than the number of sources we speak of an over determined problem. If the number of sensors and the number of sources are equal we speak of a determined problem. The more difficult problem is the underdetermined one where the number of sensors is less than the number of sources.

**3.5 INDEPENDENT COMPONENT ANALYSYS:**

In signal processing, independent component analysis(ICA) is a computational method for separating a multivariate signal into additive subcomponents. This is done by assuming that the subcomponents are non-Gaussian signals and that they are statistically independent from each other. ICA is a special case of blind source separation. A common example application is the "cocktail party problem" of listening in on one person’s speech in a noisy room.



**Fig 3.7 Independent Component Analysis**

Imagine that you are in a room where two people are speaking simultaneously. You have two microphones, which you hold in different locations. The microphones give you two recorded time signals, which we could denote by x1(t) and x2(t), with x1 and x2 the amplitudes, and tthe time index. Each of these recorded signals is a weighted sum of the speech signals emitted by the two speakers, which we denote by s1(t) and s2(t). We could express this as a linear equation:

x1(t) = a11s1 +a12s2 (1)

x2(t) = a21s1 +a22s2 (2)

Where a11,a12,a21, anda22 are some parameters that depend on the distances of the microphones from the speakers. It would be very useful if you could now estimate the two original speech signals s1(t) and s2(t), using only the recorded signals x1(t) and x2(t). This is called the cocktail-party problem. For the time being, we omit any time delays or other extra factors from our simplified mixing model [1].

If we knew the parameters aij, we could solve the linear equation in (1) by classical methods. The point is, however, that if you don’t know the aij, the problem is considerably more difficult. One approach to solving this problem would be to use some information on the statistical properties of the signals si(t) to estimate the aii. Actually, and perhaps surprisingly, it turns out that it is enough to assume that s1(t)and s2(t), at each time instant t, are statistically independent. Since the recent increase of interest in ICA, it has become clear that this principle has a lot of other interesting applications as well. It would be most useful to estimate the linear transformation from the data itself, in which case the transform could be ideally adapted to the kind of data that is being processed. This is a very general-purpose method of signal processing and data analysis.

To rigorously define ICA, we can use a statistical “latent variables” model. Assume that we observe n linear mixtures x1, ...,xn of n independent components

xj= aj1s1+aj2s2+...+ajnsn, for all j. (3)

We have now dropped the time index t, in the ICA model, we assume that each mixture xjas well as each independent component skis a random variable, instead of a proper time signal. Without loss of generality, we can assume that both the mixture variables and the independent components have zero mean. If this is not true, then the observable variables xican always be centered by subtracting the sample mean, which makes the model zero-mean.

It is convenient to use vector-matrix notation instead of the sums like in the previous equation. Let us denote by **x** the random vector whose elements are the mixtures x1, ...,xn, and likewise by **s** the random vector with elements*s*1, ..., *s*n. Let us denote by **A** the matrix with elements *aij*. Generally, bold lower case letters indicate vectors and bold upper-case letters denote matrices. All vectors are understood as column vectors; thus xT, or the transpose of x, is a row vector. Using this vector-matrix notation, the above mixing model is written as

x = As (4)

Sometimes we need the columns of matrix **A**, denoting them by **a**j the model can also be written as

X= aisi  (5)

The statistical model in Eq. 4 is called independent component analysis, or ICA model. The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components si. The independent components are latent variables, meaning that they cannot be directly observed. Also the mixing matrix is assumed to be unknown. All we observe is the random vector x, and we must estimate both Aand **s** using it. This must be done under as general assumptions as possible.

The starting point for ICA is the very simple assumption that the components si are statistically independent. It will be seen below that we must also assume that the independent component must have non-Gaussian distributions. However, in the basic model we do not assume these distributions known (if they are known, the problem is considerably simplified.) For simplicity, we are also assuming that the unknown mixing matrix is square, but this assumption can be sometimes relaxed. Then, after estimating the matrix A, we can compute its inverse, say W, and obtain the independent component simply by:

s =Wx (6)

**3.5.1 INDEPENDENCE:**

To define the concept of independence, consider two scalar-valued random variables y1and y2. Basically, the variables y1 and y2 are said to be independent if information on the value of y1 does not give any information on the value of y2, and vice versa. Above, we noted

that this is the case with the variables s1, s2but not with the mixture variables x1,x2.Technically, independence can be defined by the probability densities. Let us denote by p(y1,y2)the joint probability density function (pdf) of y1 and y2. Let us further denote byp1(y1) the marginal pdf of y1, i.e. the pdf of y1 when it is considered alone:

P1(y1) = ∫p(y1,y2)dy2 (7)

and similarly for y2 Then we define that y1 and y2 are independent if and only if the joint pdf is factorable in the following way:

p(y1,y2) = p1(y2)p2(y2). (8)

This definition extends naturally for any number n of random variables, in which case the joint density must be a product of *n* terms. The definition can be used to derive a most important property of independent random variables. Given two functions, h1 and h2, we always have

E{h1(y1)h2(y2)} = E{h1(y1)}E{h2(y2)} (9)

This can be proven as follows:

E{h1(y1)h2(y2)} = ∫ ∫ h1(y1)h2(y2)p(y1,y2) dy1 dy2

= ∫ ∫ h1(y1)p1(y1)h2(y2)p2(y2)dy1dy2

= ∫ h1(y1)p1(y1)dy1

=E{h1(y1)}E{h2(y2)} (10)

**3.5.2 UNCORRELATED VARIABLES ARE ONLY PARTLY**

**INDEPENDENT:**

A weaker form of independence is uncorrelatedness. Two random variables **y**1 and y2 are said to be uncorrelated, if their covariance is zero:

E{y1y2}−E{y1}E{y2} =0 (11)

If the variables are independent, they are uncorrelated, which follows directly from Eq. (9), taking h1(y1)=y1and h2(y2)=y2.On the other hand, uncorrelatedness does *not* imply independence. For example, assume that (y1,y2)are discrete valued and follow such a distribution that the pair are with probability 1/4 equal to any of the following values: (0,1), (0,−1), (1,0), (−1,0). Then y1and y2 are uncorrelated, as can be simply calculated. On the other hand,

E{y21,y22} = E{y21}E{y22}. (12)

so the condition in Eq. (9) is violated, and the variables cannot be independent. Since independence implies uncorrelatedness, many ICA methods constrain the estimation procedure so that it always gives uncorrelated estimates of the independent components. This reduces the number of free parameters, and simplifies the problem.

**3.5.3 PRINCIPLES OF ICA ESTIMATION:**

The key to estimating the ICA model is non-Gaussianity, without non-Gaussianity the estimation is not possible at all. In most of classical statistical theory, random variables are assumed to have Gaussian distributions, thus precluding any methods related to ICA. The Central Limit Theorem, a classical result in probability theory, tells that the distribution of a sum of independent random variables tends toward a Gaussian distribution, under certain conditions. Thus, a sum of two independent random variables usually has a distribution that is closer to Gaussian than any of the two original random variables. Let us now assume that the data vector x is distributed according to the ICA data model in Eq. 4, i.e. it is a mixture of independent components. For simplicity, let us assume in this section that all the independent components have identical distributions. To estimate one of the independent components, we consider a linear combination of the xi, let us denote this by y=wTx=åiwixi, where **w** is a vector to be determined. If w were one of the rows of the inverse of A, this linear combination would actually equal one of the independent components. Here we use the Central Limit Theorem to determine w so that it would equal one of the rows of the inverse of A**,** we cannot determine such a w exactly, because we have no knowledge of matrix A, but we can find an estimator that gives a good approximation.

To see how this leads to the basic principle of ICA estimation, let us make a change of variables, defining z =ATw. Then we have y= wTx= wTAs= zTs. y is thus a linear combination of si, with weights given by zi. Since a sum of even two independent random variables is more Gaussian than the original variables, zTs is more Gaussian than any of the si and becomes least Gaussian when it in fact equals one of the si. In this case, obviously only one of the elements zi of z is nonzero. (Note that the si were here assumed to have identical distributions.) Therefore, we could take as w a vector that maximizes the non-Gaussianity of wTx. Such a vector would necessarily correspond (in the transformed coordinate system) to a **z** which has only one nonzero component. This means that wTx=zTs equals one of the independent components.

Maximizing the non-Gaussianity of wTx thus gives us one of the independent components. In fact, the optimization landscape for non-Gaussianity in the n-dimensional space of vectors w has 2n local maxima, two for each independent component, corresponding to si and −si(recall that the independent components can be estimated only up to a multiplicative sign). To find several independent components, we need to find all these local maxima. This is not difficult, because the different independent components are uncorrelated: We can always constrain the search to the space that gives estimates uncorrelated with the previous ones. This corresponds to orthogonalization in a suitably transformed (i.e. whitened) space. To use non-Gaussianity in ICA estimation, we must have a quantitative measure of non-Gaussianity of a random variable, say y. To simplify things, let us assume that y is centered (zero-mean) and has variance equal to one [10].

**3.5.4 PREPROCESSING FOR ICA:**

The practical algorithms based on the statistical principles are used in the ICA. However, before applying an ICA algorithm on the data, it is usually very useful to do some preprocessing. In this section, we discuss some preprocessing techniques that make the problem of ICA estimation simpler and better conditioned. They are

**3.5.4.1 CENTERING:**

The most basic and necessary preprocessing is to center x, i.e. subtract its mean vector m= *E*{**x**} so as to make x a zero-mean variable. This implies that **s** is zero-mean as well, as can be seen by taking expectations on both sides of Eq. (4).

This preprocessing is made solely to simplify the ICA algorithms. It does not mean that the mean could not be estimated. After estimating the mixing matrix A with centered data, we can

complete the estimation by adding the mean vector of **s** back to the centered estimates of s. The mean vector of **s** is given by A−1m, where m is the mean that was subtracted in the preprocessing.

**3.5.4.2 WHITENING:**

Another useful preprocessing strategy in ICA is to first whiten the observed variables. This means that before the application of the ICA algorithm (and after centering), we transform the observed vector x linearly so that we obtain a new vector ˜x which is white, i.e. its components are uncorrelated and their variances equal unity. In other words, the covariance matrix of ˜x equals the identity matrix.

The whitening transformation is always possible. One popular method for whitening is to use the eigen-value decomposition (EVD) of the covariance matrix *E*{xxT}=EDET, where E is the orthogonal matrix of eigen vectors of *E*{xxT} and D is the diagonal matrix of its eigen values, D=diag(d1, ...,dn). Note that *E*{xxT}can be estimated in a standard way from the available sample x(1), ...,x(T). Whitening can now be done by

˜x = ED−1/2ETx (13)

Where the matrix D−1/2is computed by a simple component-wise operation as

D−1/2=diag(d−1/21 , ...,d−1/2n). Itis easy to check that now *E{˜x˜xT} =* I.

Whitening transforms the mixing matrix into a new one˜**A**. We have from (4) and (13):

˜x = ED−1/2E*T*As = ˜As(14)

The utility of whitening resides in the fact that the new mixing matrix ˜A is orthogonal. This can be seen from

E{˜x˜xT} = ˜AE{ssT}˜AT = ˜A˜AT= I.(15)

Here we see that whitening reduces the number of parameters to be estimated. Instead of having to estimate the n2 parameters that are the elements of the original matrix A, we only need to estimate the new, orthogonal mixing matrix ˜A. An orthogonal matrix contains n(n−1)/2 degrees of freedom. For example, in two dimensions, an orthogonal transformation is determined by a single angle parameter. In larger dimensions, an orthogonal matrix contains only about half of the number of parameters of an arbitrary matrix. Thus one can say that whitening solves half of the problem of ICA. Because whitening is a very simple and standard procedure, much simpler than any ICA algorithms, it is a good idea to reduce the complexity of the problem this way.

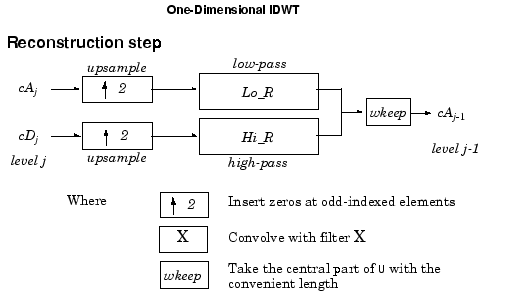
**3.5.5 PROPERTIES OF ICA ALGORITHM:**

The ICA algorithm and the underlying contrast functions have a number of desirable properties when compared with existing methods for ICA.

* The convergence is cubic (or at least quadratic), under the assumption of the ICA data model. This is in contrast to ordinary ICA algorithms based on (stochastic) gradient descent methods, where the convergence is only linear. This means a very fast convergence, as has been confirmed by simulations and experiments on real data.
* Contrary to gradient-based algorithms, there are no step size parameters to choose. This means that the algorithm is easy to use.
* The algorithm finds directly independent components of (practically) any non-Gaussian distribution using any nonlinearity. This is in contrast to many algorithms, where some estimate of the probability distribution function has to be first available, and the nonlinearity must be chosen accordingly.
* The performance of the method can be optimized by choosing a suitable nonlinearity *g*. In particular, one can obtain algorithms that are robust and/or of minimum variance. In fact, the two nonlinearities have some optimal properties.
* The ICA has most of the advantages of neural algorithms: It is parallel, distributed, computationally simple, and requires little memory space. Stochastic gradient methods seem to be preferable only if fast adaptivity in a changing environment is required.

**3.6 INVERSE DISCRETE WAVELET TRANSFORMATION:**

Once we arrive at our discrete wavelet coefficients, we need a way to reconstruct them back into the original signal (or a modified original signal if we played around with the coefficients). In order to do this, we utilize the process known as inverse discrete waveform transform. Much like the DWT can be explained by using filter bank theory, so can the reconstruction of the IDWT [11]. The process is simply reversed. The DWT coefficients are first up sampled by placing zeros in between every coefficient, effectively doubling the length of each. These are then convolved with the reconstruction scaling filter for approximation coefficients and the reconstruction wavelet filter for the detail coefficients. These results are then added together to arrive at the original signal.



**Fig 3.8 Block Diagram of Inverse Discrete Wavelet Transform**

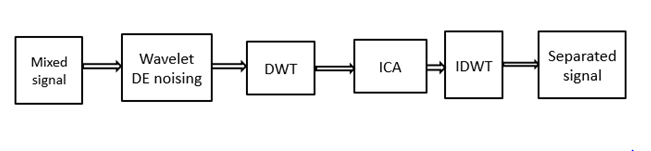
Starting from the approximation and detail coefficients at level j, cAj and cDj*,* the inverse discrete wavelet transform reconstructs cAj-1, inverting the decomposition step by inserting zeros and convolving the results with reconstruction filters.

**ANALYSIS OF THE METHODOLOGIES**

**4.1 INTRODUCTION:**

Blind signal separation is also known as blind source separation. It is the separation of set of signals from an unknown mixed linear combinations. A digital signal processing system can be developed to extract the required voice from the rest of the speakers. Here we are implementing the source separation by independent component analysis (ICA). The independent component analysis algorithm allows two source signals to be separated from two mixed signals using statistical principles of independence and Non-Gaussianity. In The following subsection we have described the source separation process using the flowchart.

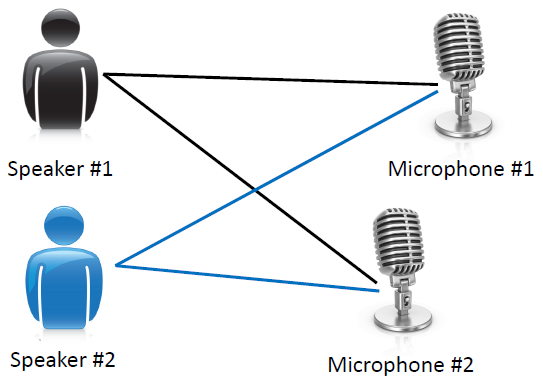
A brief working of individual blocks are discussed below.



**Fig 4.1 Block Diagram of Blind Source Separation.**

**4.2 MIXED SIGNAL:**

1. Record too many person’s voice as an input which will consists of environmental noise, which is known as the mixed signal through the microphone or array of microphones.



**Fig 4.2 Mixed Signal**

1. Now the recorded audio file will be in the **.WMA** file.
2. To read the input signal to **MATLAB**, we have to convert the **.WMA file** into **.WAV file.**
3. By using the **Total Video Converter** software we can convert the **.WMA file** into a desired audio extension i.e**.WAV file.**
4. The total video player extension is also used along with the total video converter which helps to play the extended.

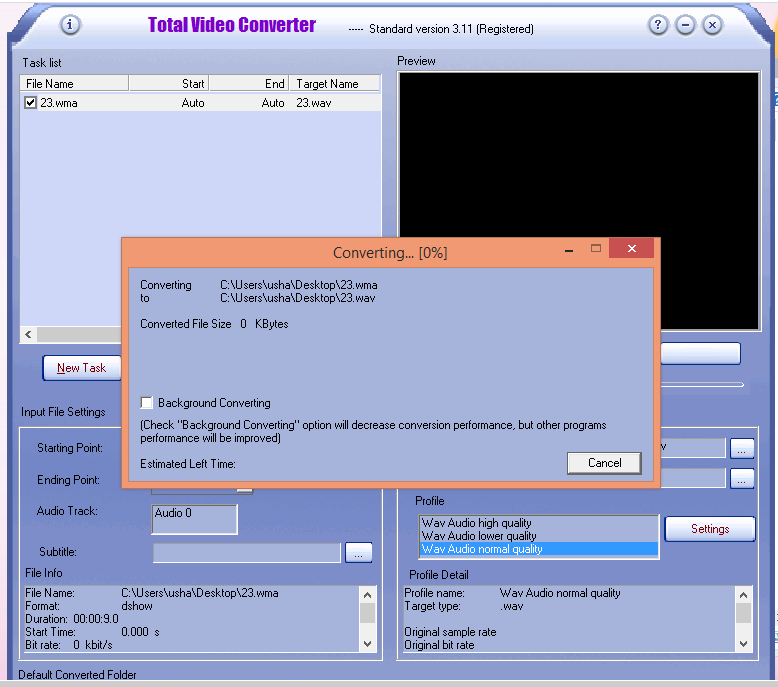
**CONVERTING .WMA FILE INTO .WAV FILE:**

Open total video converter and select new task

* 1. In cd/wav audio select Ms wav audio



After selecting ms wav/audio we need to select the location to save .wav file and then click convert. A new dialogue box will open which shows the conversion.

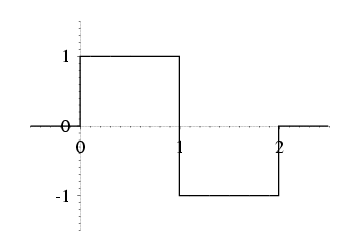


Thus the .wma file will be converted into .wav file.

The converted audio file is then applied to the wavelet denoising to remove the noise from the given input mixed signal.

## **4.3 THEHAARWAVELET:**

The Haar transform is based on a class of orthogonal matrices whose elements are either1, -1 or 0 multiplied by power of √2. The Haar transform is a computationally efficient transform as a transform of an N-point vector requires only 2(N-1) additions and N multiplication [1].



**Fig 4.3 Haar Wavelet**

**4.4 WAVELET DENOISING:**

Wavelets are suited to the denoising of signals with sharp transients. A threshold is used to delete the wavelet coefficients where the signal is smooth (thus leaving the denoising to the low pass cascade) and preserve these coefficients when they are large. Wavelet thresholding methods for noise removal, in which the wavelet coefficients are thresholded in order to remove their noisy part. Here we will remove the noise from the given signal by applying soft thresholding

Wavelet shrinkage denoising is not smoothing (despite the use by some authors of the term smoothing as a synonym for the term denoising). Whereas smoothing removes high frequencies and retains low frequencies, denoising attempts to remove whatever noise is present and retain whatever signal is present regardless of the frequency content of the signal.

When we denoise music corrupted by noise, we would like to preserve both the treble and the bass. Second, it is denoising by shrinking (i.e*.,* nonlinear soft thresholding) in the wavelet transform domain.

Third, it consists of three steps:

1) a linear forward wavelet transform,

2) a nonlinear Shrinkage denoising, and

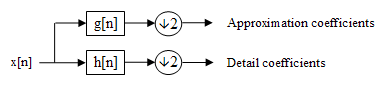
3) a linear inverse wavelet transform.

Because of the nonlinear shrinking of coefficients in the transform domain, this procedure is distinct from those denoising methods that are entirely linear. Finally, wavelet shrinkage denoising is considered a non-parametric method. Thus, it is distinct from parametric methods in which parameters must be estimated for a particular model that must be assumed *a* priori.

Restoration is a kind of denoising that tries to retrieve the original signal with optimal balancing of de-blurring and smoothing Wavelet transforms enable us to represent signals with a high degree of sparsity. This is the principle behind a non-linear wavelet based signal estimation technique known as wavelet denoising. Wavelet denoising attempts to remove the noise present in the signal while preserving the signal characteristics, regardless of its frequency content. Thus, most of the noise tends to be represented by the wavelet coefficients at finer scales. Discarding these coefficients would result in a natural filtering out of noise on the basis of scale.

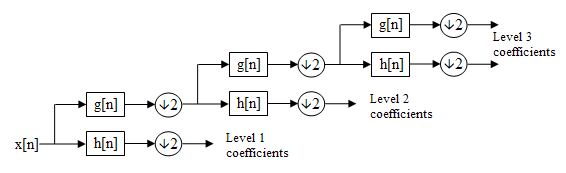
**4.5DISCRETE WAVELET TRANSFORM:**

1. The wavelet Transform is a technique for analyzing signals. It was developed as an alternative to the short time Fourier Transform (STFT) to overcome the problems related to its frequency and time resolution properties.
2. The discrete Wavelet Transform (DWT) is a special case of the WT that provides a compact representation of a signal in time and frequency that can be computed efficiently.
3. Discrete wavelet transform can be used for easy and fast denoising of a noisy signal.
4. Firstly, Dwt separates the input signal information into :
   1. Approximation coefficients
   2. Detail coefficients



**Fig.4.4Block Diagram of Filter Analysis**

1. Most of the data which we require is present at approximation coefficients. The data present at detail coefficients are considered as noise.
2. This decomposition is repeated to further increase the frequency resolution and the approximation coefficients decomposed with high and low pass filters and then down-sampled. This is represented as a binary tree with nodes representing a sub-space with a different time-frequency localization. The tree is known as a [filter bank](http://en.wikipedia.org/wiki/Filter_bank).



**Fig.4.5 A 3-Level Filter Bank**

1. Secondly, We are applying **Discrete Wavelet Transformation** to **Haar Wavelet**.
2. Finally, we performed down sampling on approximation coefficients.
3. A down sampling technique decreases sampling rate by integer factor.

**SYNTAX:**

Y= downsample (x,n)

**DESCRIPTION:**

Y=down sample (x,n) decreases the sampling rate of x by keeping every nth sample starting with first sample. x can be a vector or matrix, if x is a matrix each column is considered a separate sequence.

**EXAMPLE:**

Decrease the sampling rate of a sequence by 3

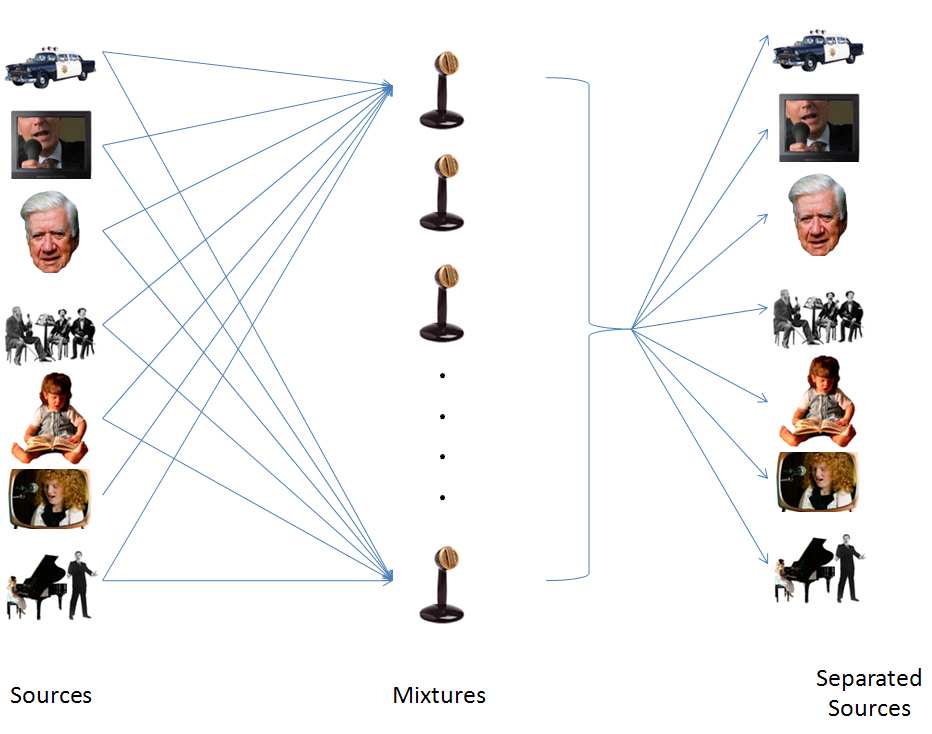
x = [1 2 3 4 5 6 7 8 9 10];

y = downsample (x,3)

y =[1 4 7 10]

**4.6 INDEPENDENT COMPONENT ANALYSYS:**

1. ICA belongs to a class of Blind Source Separation (BSS) methods for separating data in to underlying informational components from multidimensional statistical data.
2. The term “blind” is intended to imply that such methods can separate data in to source signals even if very little is known about the nature of those source signals.
3. The independent component analysis algorithm allows two source signals to be separated from two mixed signals using statistical principles of independence and Non-Gaussianity.
4. ICA assumes that the value of each source at any given time is a random variable. It also assumes that each source is statistically independent, meaning that the values of one source cannot be correlated to values in any of these other sources.
5. With these assumptions, ICA allows us to separate source signals from mixtures of the source signals.

  
 **Fig.4.6 Independent Component Analysis.**

For this purpose we are taking weights for our signal

To calculate the **ICA** we are using formula mentioned below:

Y=w\*x

Here w=weight of signal

**Weights** are updated by a formula

New weight=old weight+2\*step size\*error

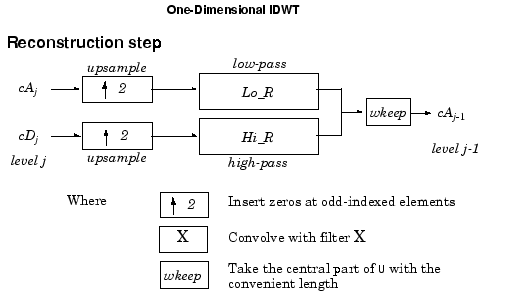
Hence, ICA

Y=New Weight \*x

Where x= our signal.

**4.7 INVERSE DISCRETE WAVELET TRANSFORMATION:**

1. Once we arrive at our discrete wavelet coefficients, we need a way to reconstruct them back into the original signal (or a modified original signal if we played around with the coefficients).
2. In order to do this, we utilize the process known as inverse discrete waveform transform.
3. This process is simply reverse of **DWT**.



**Fig.4.7 Block Diagram of Inverse Discrete Wavelet Transform.**

Where

**cA** – Approximation coefficients.

**cD** – Detail coefficients.

1. Starting from the approximation and detail coefficients at level *j,cAj*and c*Dj,* the inverse discrete wavelet transform reconstructs cAj-1, inverting the decomposition step by inserting zeros and convolving the results with reconstruction filters.
2. Here we are performing upsampling.
   1. Upsampling increases sampling rate by integer factor.

**SYNTAX:**

Y=upsample(x,n)

**DESCRIPTION:**

y = upsample (x,n) increases the sampling rate of x by inserting n-1zeros between samples. x can be a vector or a matrix. If x is a matrix, each column is considered a separate a sequence. The up sampled y has x\*n samples.

**EXAMPLE:**

Increase the sampling rate of a sequence by 3:

x = [1 2 3 4];

y = upsample (x,2);

x,y

x = [1 2 3 4]

y = 1 0 0 2 0 0 3 0 0 4 0 0

**4.8 ALGORITHM:**

Step 1: Record the audio file

Step 2: Convert it into **.W AV** file

Step 3: Place the .WAV audio file in the required **MATLAB** folder

Step 4: Read the audio file which is the input signal.

Step 5: To remove the noise apply input to the **Wavelet Denoising**, as the given input consists of noise.

Step 6: Then apply it to the **Discrete Wavelet Transform**

Step 7: The output of the discrete wavelet transform is given to the **ICA**

Step 8: Output signal.

**4.9 FLOW CHART:**

Record the audio file

Convert it into the .WAV file

Yes No

Noise

Output

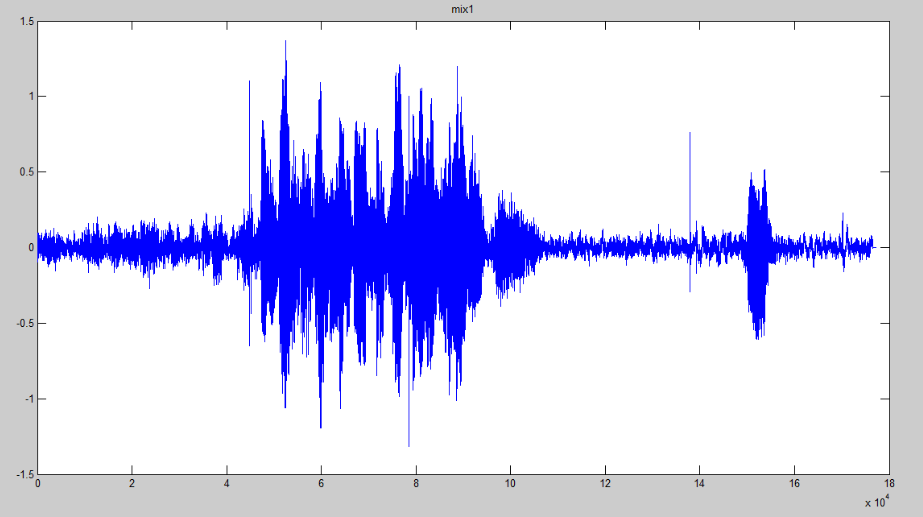
Independent Component Analysis

DWT

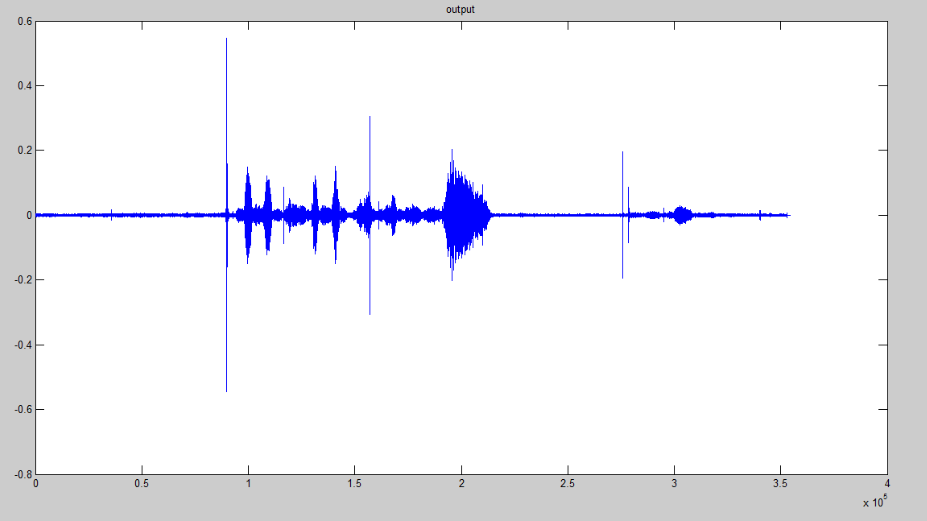
Wavelet Denoising

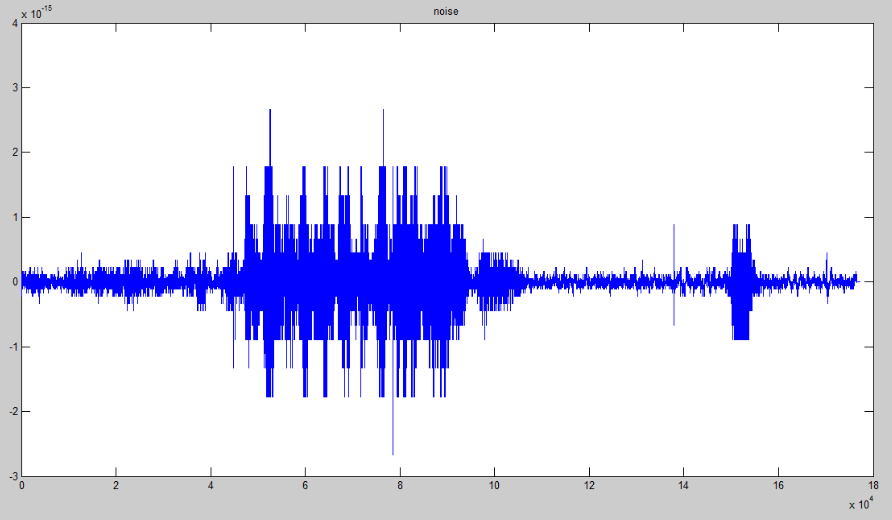
**5.SIMULATION RESULTS**

It was the plotted figure for one of the different inputs. The below is the input which is a mixed signal

****

The below figure is the output which was a separated signal

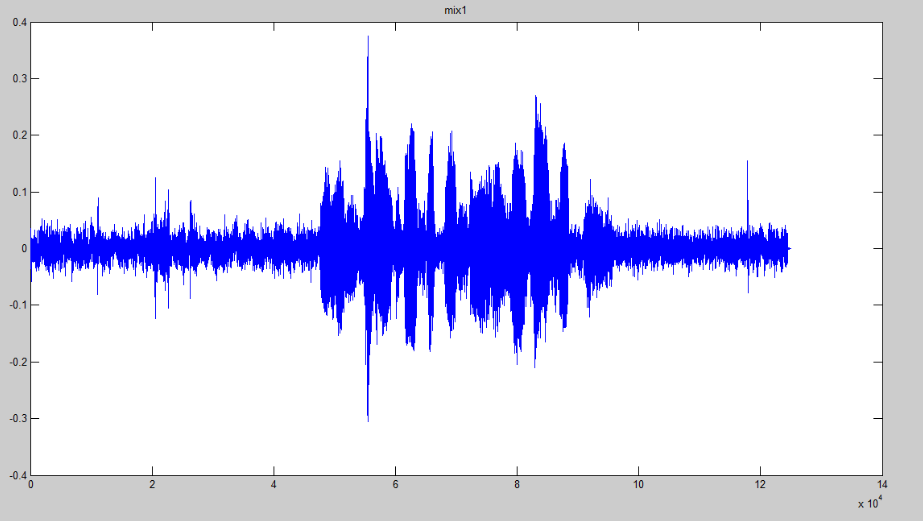
****

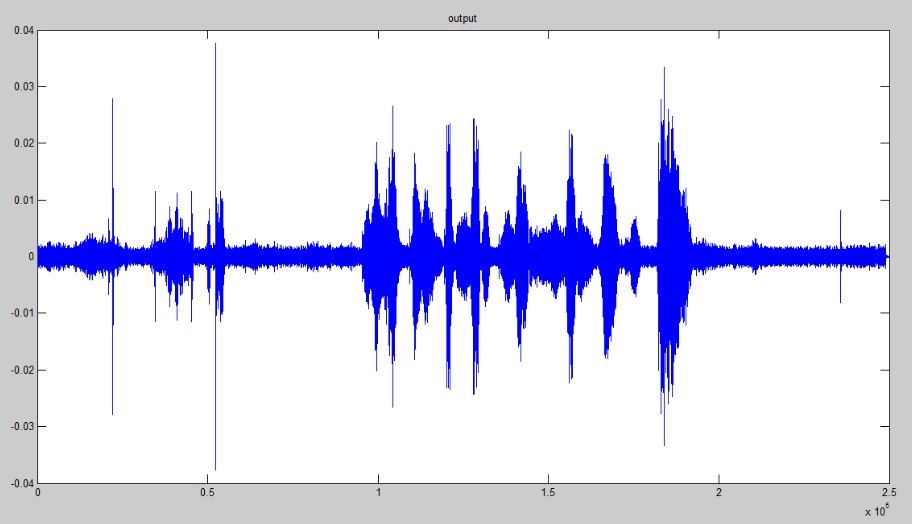
****

The above figure was the noise signal which was been separated from the input mixed signal

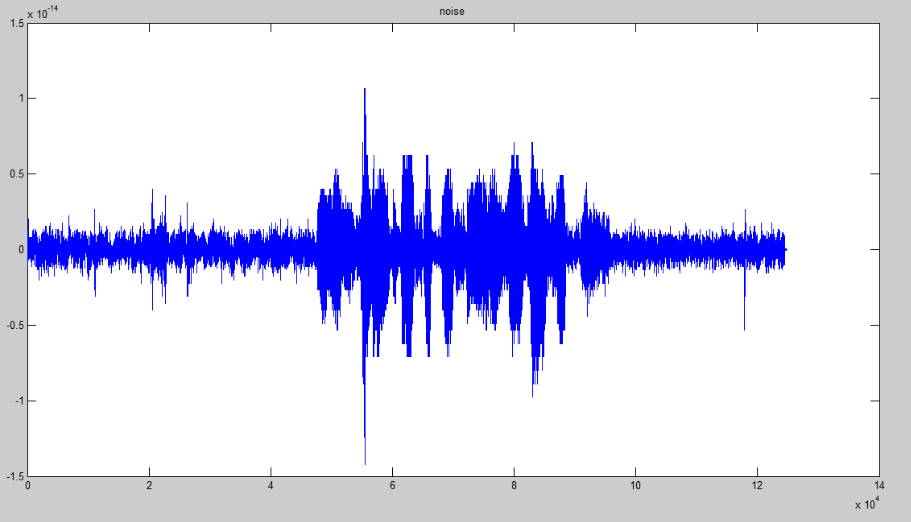
This above results are the experimental run out waveforms for the input mixed signal which was taken by us.

This below figures are the results for the second input which we have taken i.e., v1 .And this figure was the input mixed signal for the input.





This above and below figures are the separated noise and voice signals.



**APPLICATIONS**

* Audio signal processing find a wide range of application in communication fields, signal analysis.
* Application in areas which include storage, level compression, data compression.
* And also in transmission equalisation, noise cancellation, echo reverberation.
* Can be extended to speech therapy for physically challenged in medical fields.
* In many engineering fields like aerospace, radar and telemetry tracking, and at signalling to aeroplanes at air domes.
* It can helps to our military men, these thesis can be extended for human identification purposes this work helps in tracking the enemies at border works.

**ADVANTAGES**

* Noise will be removed completely.
* We can separate the signals from the given mixed signals without any loss or compression.
* No clumsy eruptions rises in between the process.
* No domination or unnecessary suppression of data is possible through this analysis.
* A clear audio can be presented by this output.
* Can be extended to human speech frequencies.

**CONCLUSION AND FUTURE SCOPE**

**CONCLUSION:**

We conclude that the implementation of source separation using the ICA algorithm that uses wavelets led to some appreciable results when compared to direct ICA application on the signal. The signals were first separated by applying ICA algorithm directly on the mixed signal. This produced quite impressive results as both the signals were partially separated and were generated as the output.

**FUTURE SCOPE:**

In this project, we have presented a blind source separation using wavelets and discrete wavelet transformation switch was been applied to our mixed signal input. We have performed a separation of a mixed signal by applying wavelet de-noising on the mixed signal and on to that a discrete wavelet transformation and on to that our main techniques ICA was been applied and finally, separation was achieved without any loss of information.

* A super personal human identification can be extended to our work.
* A smart human identification and speech recognition can be papered by our work.
* Our work can further helps for the security purposes.

**APPENDIX -A**

**INTRODUCTION TO MATLAB**

MATLAB (Matrix Laboratory) is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB, we can solve technical computing problems faster than with traditional programming languages, such as C, C++, and FORTRAN.

MATLAB is used in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modelling and analysis, and computational biology.

We can integrate our MATLAB code with other languages and applications, and distribute our MATLAB algorithms and applications.

**How to open the MATLAB:**

* On desktop, start matlab by **double clicking** the **matlab shortcut icon**on your **windows desktop**.
* After that it appears as follows:



After that a special window called the **MATLAB desktop** appears. The desktop is a window that contains other windows. The major tools within or accessible from the desktop are:

* The Command Window
* The Command History
* The Workspace
* The Current Directory
* The Help Browser
* The Start button

The **Command Window** is the window on the middle of the screen.This window is used to both enter commands for MATLAB to execute, and toview the results of these commands.

The **Command History** window, in the lower right side of the screen, displays the commands that have been recently entered into the Command Window.

In the upper right hand side of the screen there is a window that can contain two different windows with tabs to select between them. The first window is the **Current Directory**, which tells the user which M-files are currently in use.

The second window is the **Workspace** window, which displays which variables are currently being used and how big they are.

If programmer writes code that do not want to reappear in the MATLAB Command Window, you must place a semi colon after the line of code. If there is no semi colon, then the code will print in the command window

>>x=1; y=1;

The x and y values are suppressed that means it will not appear on the command window.

>>x=1

>>y=1

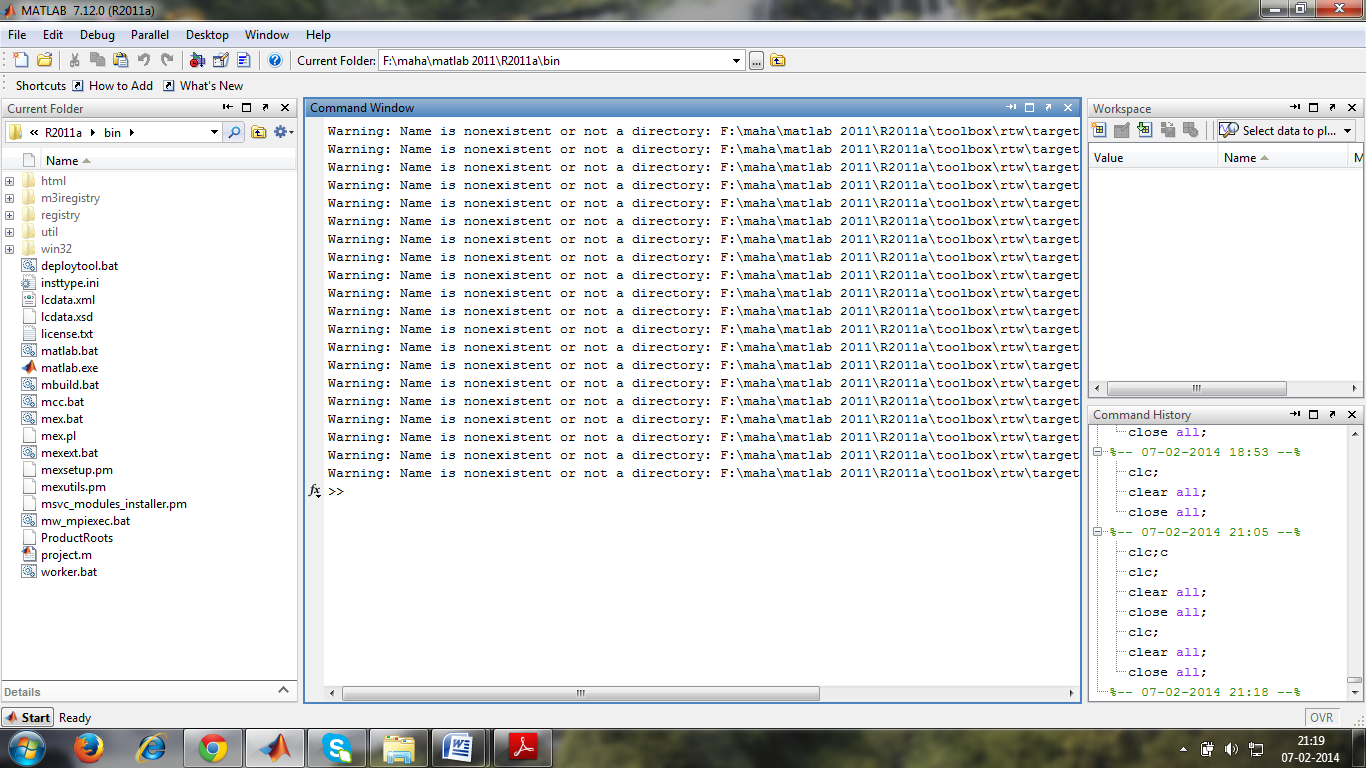
The values of x and y are appeared on command window as follows

>>x

=1

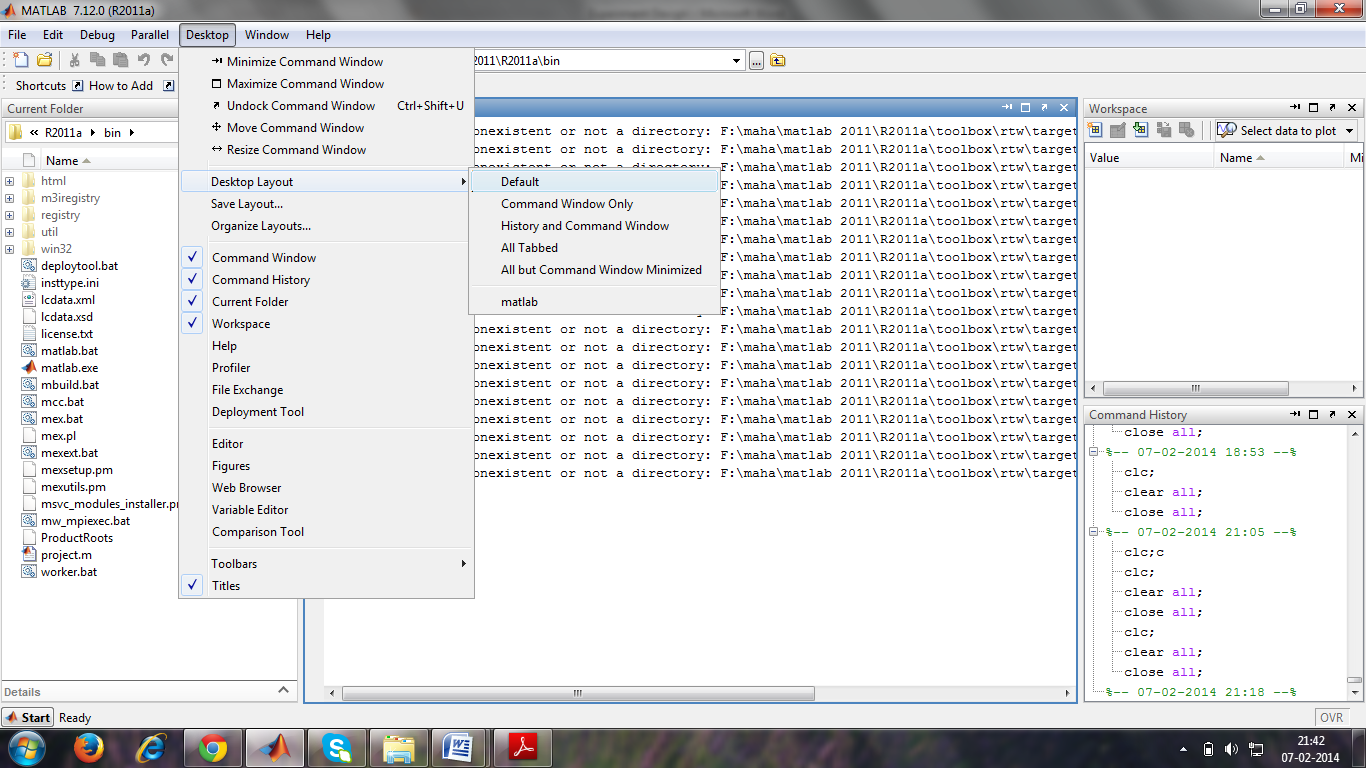
>>y

=1



In some cases mat lab desktop does not appear like above figure at that time follow the below path to get mat lab desktop

Click on **Desktop** ---🡪 In that choose **Desktop Layout** -🡪 In that click on **Default**



**How to clear the command window and workspace and close all the previously executed figures:**

After that in command window type as

**clc;**%for clear the command window

**clc** clears all input and output from the Command Window display, giving you a clean screen. After using **clc**, you cannot use the scroll bar to see the history of functions, but you still can use the up arrow to recall statements from the command history

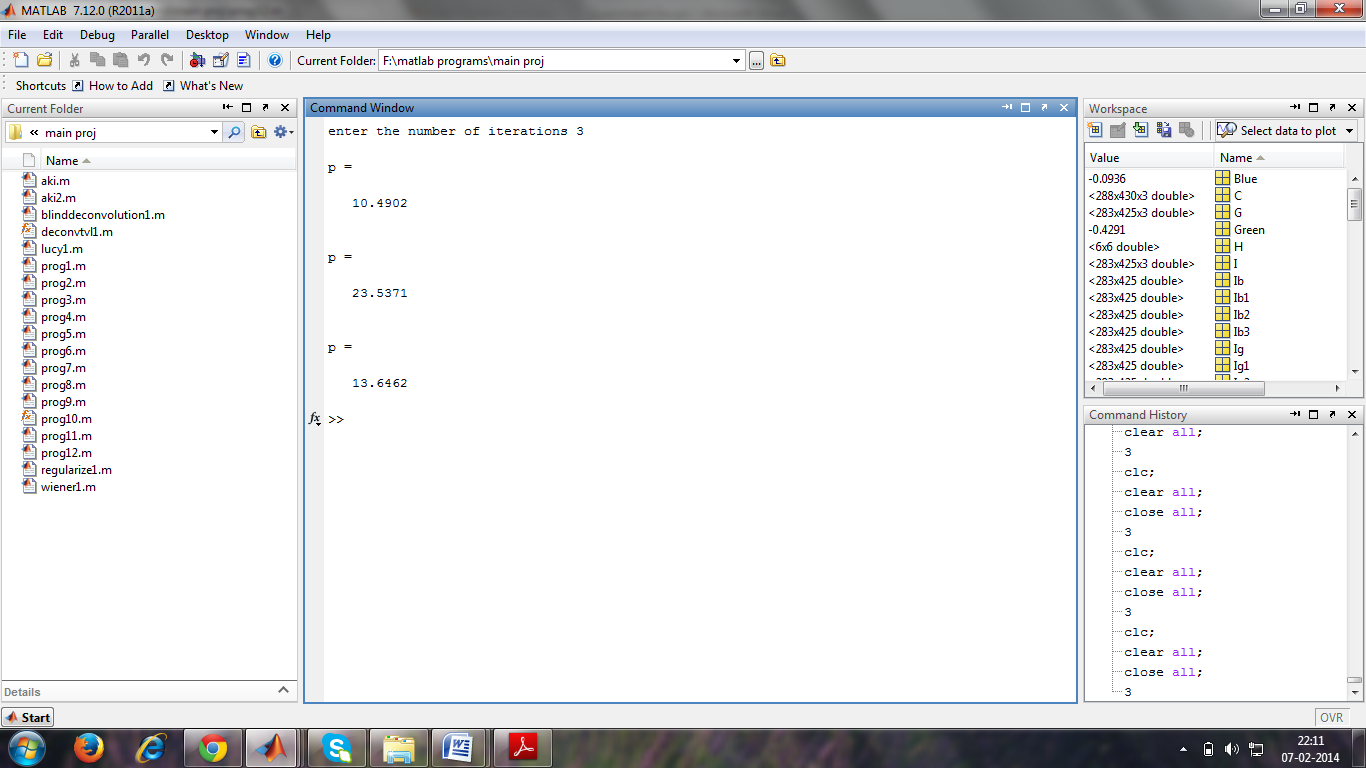
**Clear all**; %for clear the work space

Removes all variables, functions, and MEX-files from memory, leaving the workspace empty. Using clear all removes debugging breakpoints in M-files and reinitializes persistent variables, since the breakpoints for a function and persistent variables are cleared whenever the M-file is changed or cleared. When issued from the Command Window prompt.

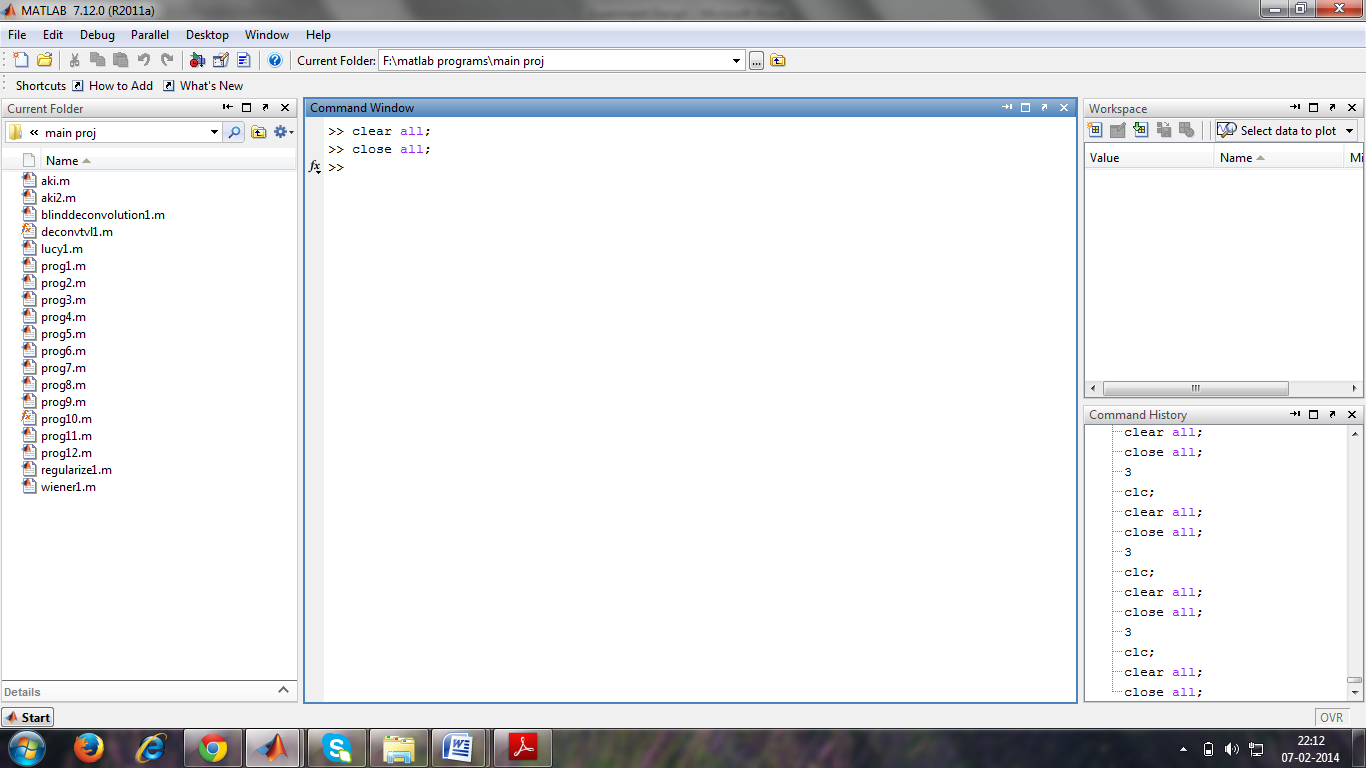
**Close all;** %for closing all the figures

‘Clear all’ deletes all figures whose handles are not hidden

Consider the window before applying above operations



After applying above operations we can get window as possible.



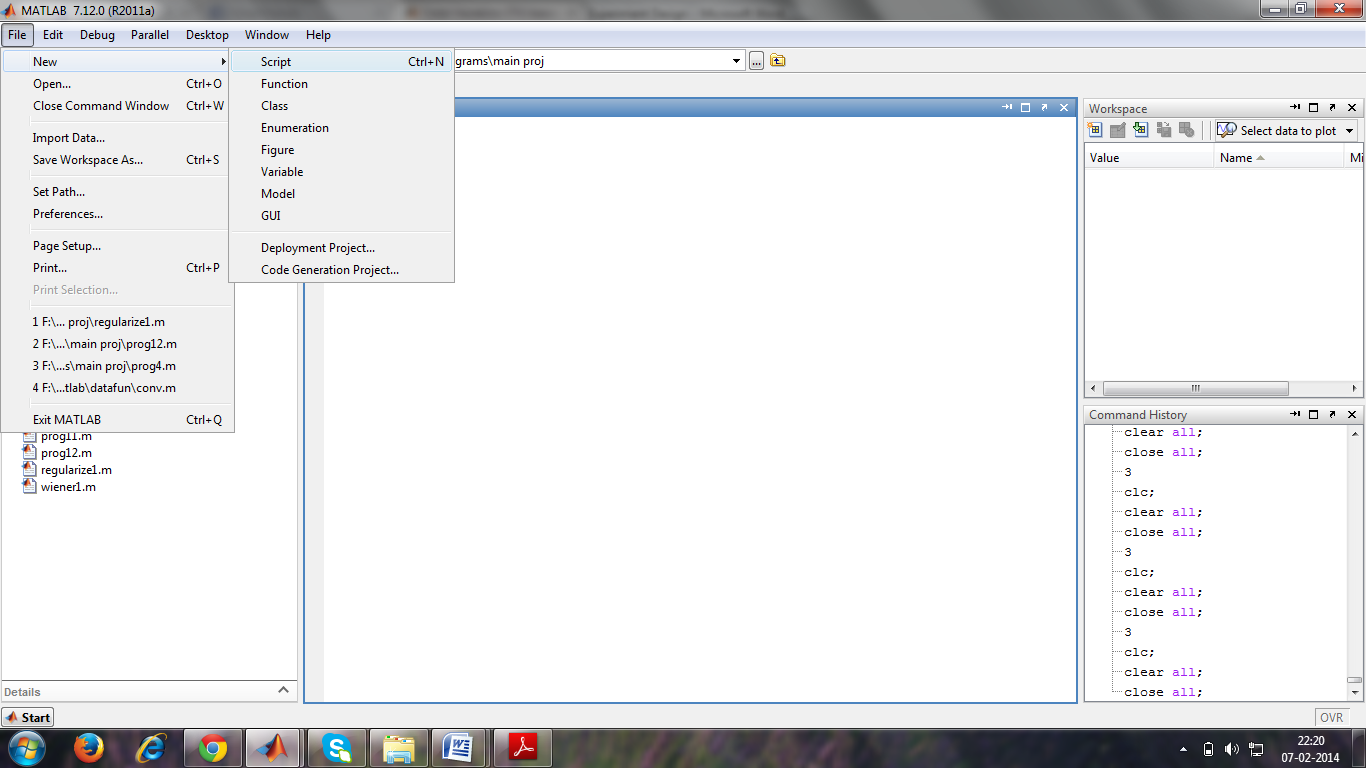
**How to written code in editor window:**

Program is written in editor .so to get the .m file for writing the program follow the below path.

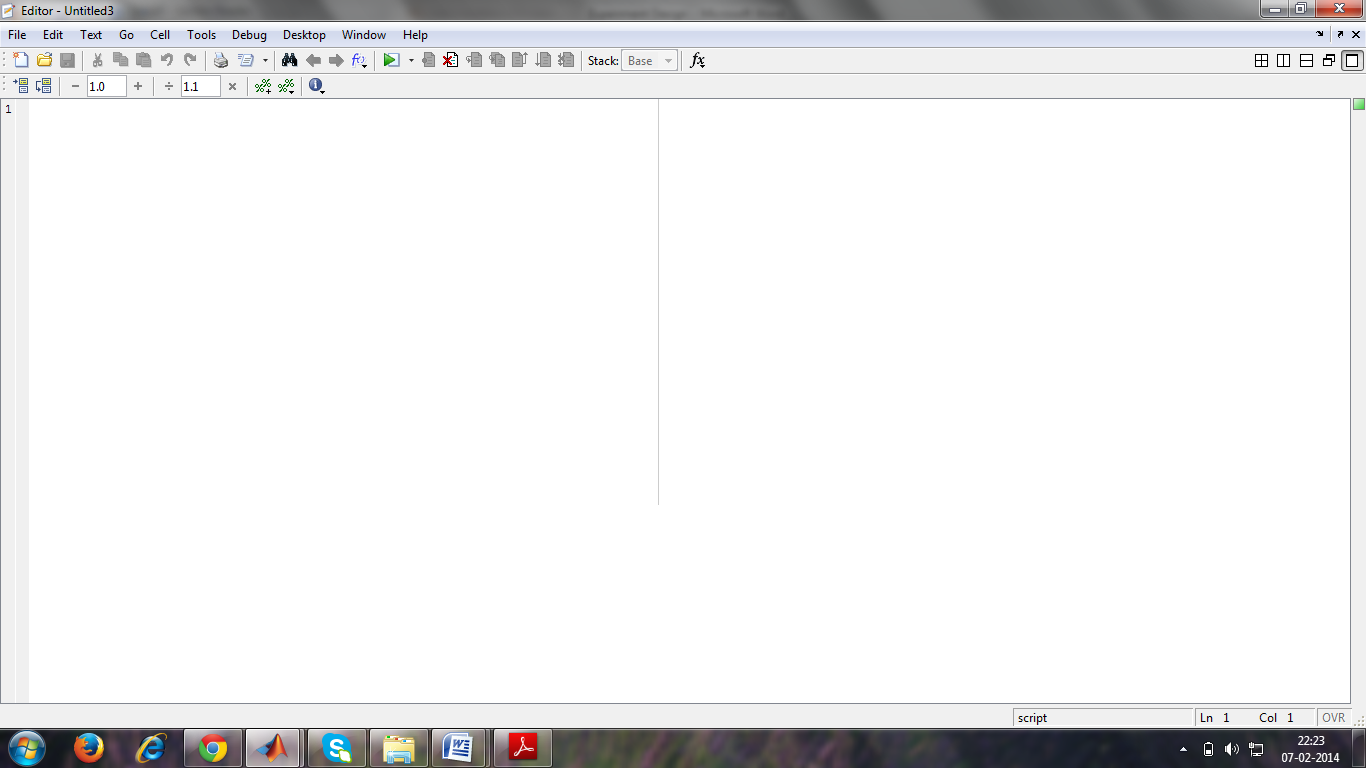
**M-FILE:**

An M-file is a MATLAB document the user creates to store the code they write for their specific application. **Creating an M-file is highly recommended**, although not entirely necessary. An M-file is useful because it saves the code the user has written for their application. It can be manipulated and tested until it meets the user’s specifications. The advantage of using an M-file is that the user, after modifying their code, must only tell MATLAB to run the M-file, rather than renter each line of code individually.

**Creating an M-file**: Click on the **File menu--🡪**In that choose **New**--🡪In that choose and click on **Script OR m-file**



After that the editor window is appeared as follows.



After that we can write the program in the above window as follows. Let us consider a small program as follows.

**SAVING:**

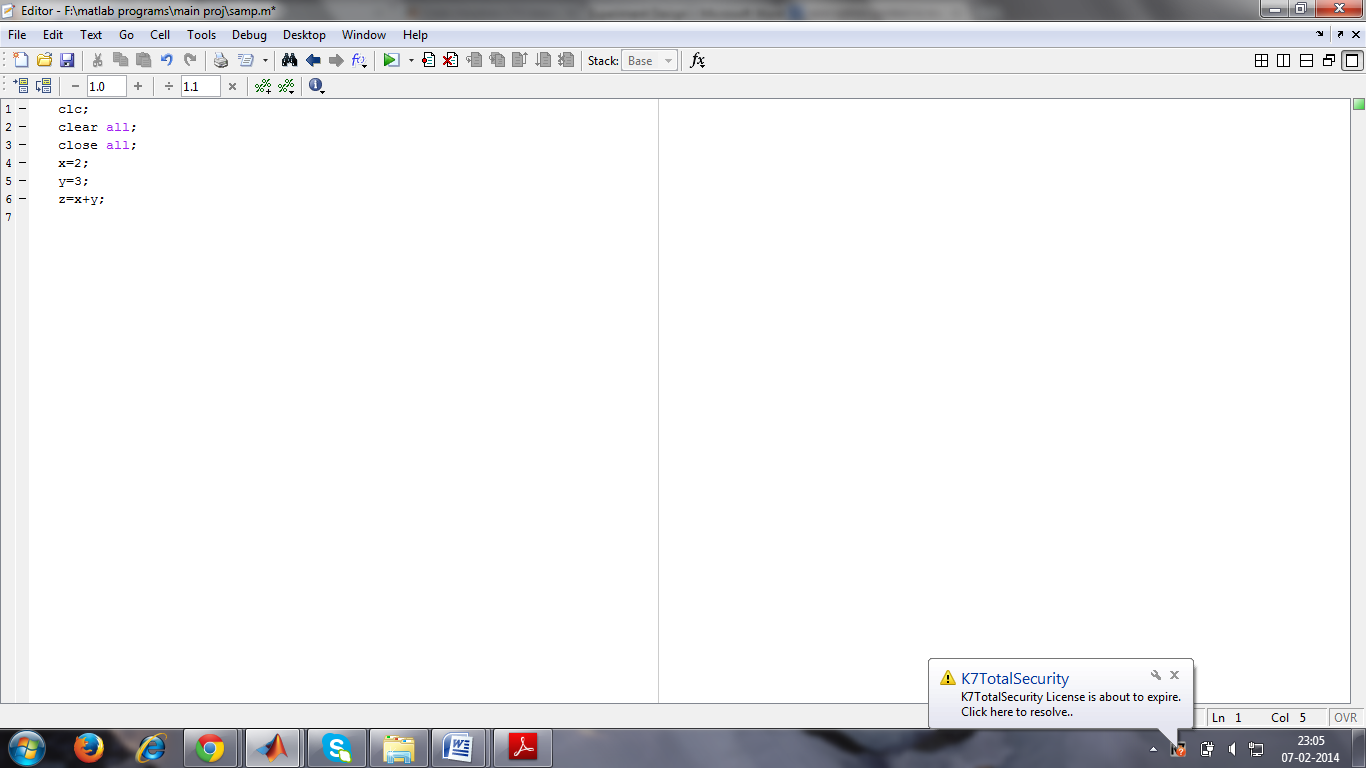
The next step is to save the newly created M-file. In the M-filewindow, select **File\Save as…** Choose a location that suits your needs, such asa disk, the hard drive or the U drive. It is not recommended that you work fromyour disk or from the U drive, so before editing and testing your M-file you maywant to move your file to the hard drive

**OPENING AN M-FILE:**

To open up a previously designed M-file, simply openMATLAB in the same manner as described before. Then, open the M-file bygoing to **File\Open…**, and selecting your file. Then, in order for MATLAB torecognize where your M-file is stored, you must go to **File\Set Path…** Thiswill open up a window that will enable you to tell MATLAB where your M-fileis stored. Click the **Add Folder…** button, then browse to find the folder thatyour M-file is located in, and press **OK**. Then in the Set Path window, select**Save**, and then **Close**. If you do not set the path, MATLAB may open awindow saying your file is not in the current directory. In order to get by this,select the “**Add directory to the top of the MATLAB path**” button, and hit**OK**. This is essentially the same as setting the path, as described above.

**WRITING CODE:**

After creating and saving the M-file, the next step is to begin writing code. First move is to begin by writing comments at the top of the M-file with a description of what the code is for, who designed it, when itwas created, and when it was last modified if needed. Comments are declared by placinga % symbol before them. Comments appear in green in the M-file window.

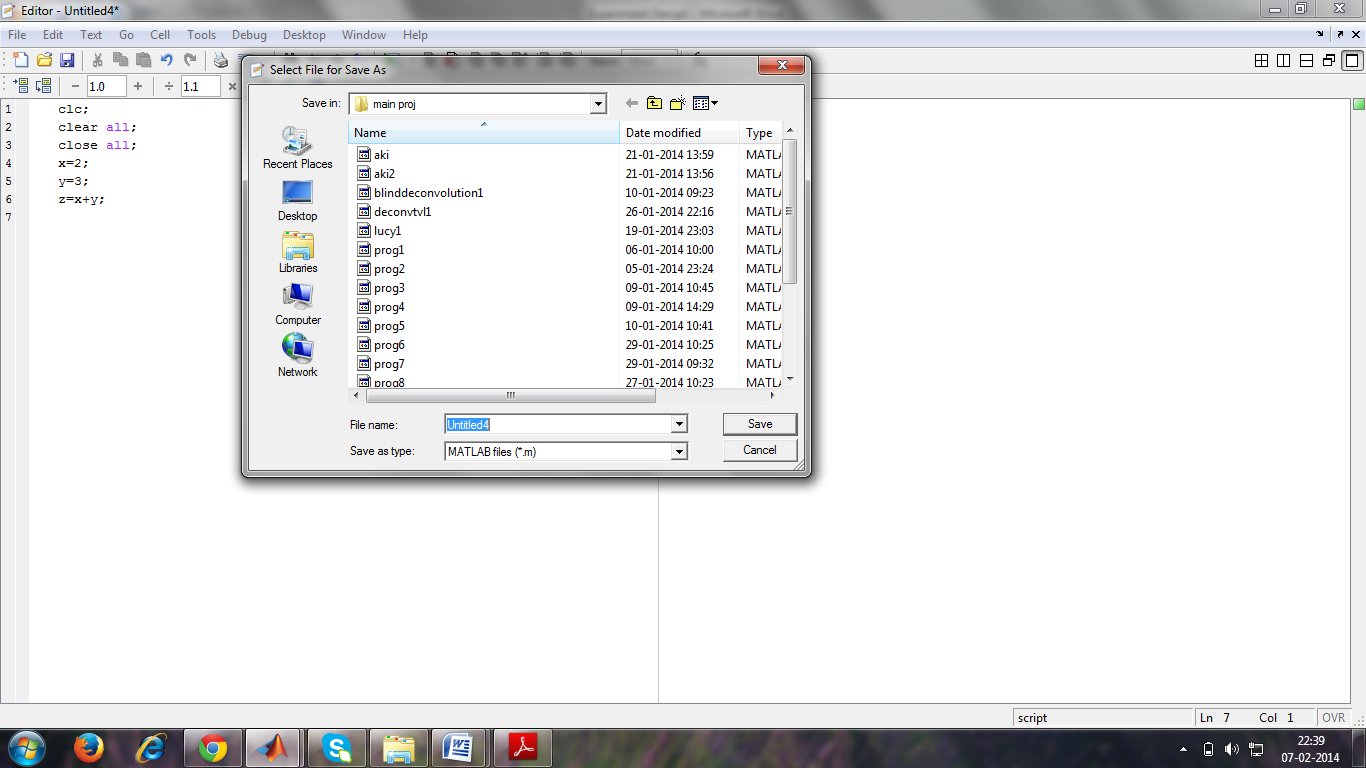


**RESAVING:**

After writing code, you must save your work before you can run it.Save your code by going to **File\Save**.

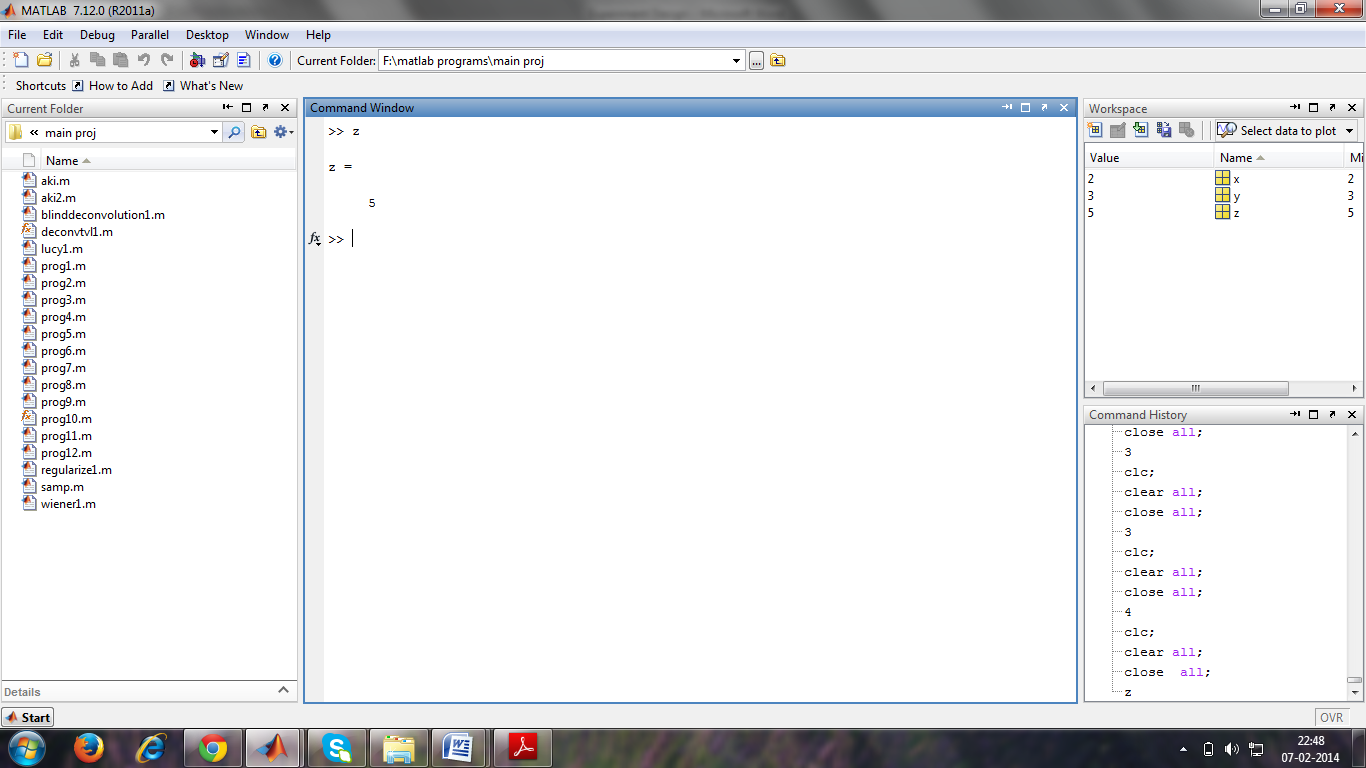
**RUNNING CODE:**

To run code, simply go to the main MATLAB window andtype the name of your M-file after the **>>**prompt. Other ways to run the M-fileare to press **F5** while the M-file window is open, select **Debug\Run**, or pressthe **Run** button in the M-file window toolbar.



To resave that program we have to give any name at the **File name** and then click on**save.**

Suppose if we want to perform any operation to that program go to command window and do that operation. After that the window is appeared as follows.



**APPENDIX-B**

**BASICS OF MATLAB**

**BASIC COMMANDS USED IN MATLAB:**

The help from matlab is obtained by typing some commands on command window. They are

* **Help**
* **lookfor**
* **helpwin**
* **helpdesk**
* **demos**

**MATRICES IN MATLAB:**

Matrix is a main MATLAB’s data type

Build a matrix in matlab

**A = [1 2 3; 4 5 6; 7 8 9];**

Creates matrix A with size 3x3.

Special matrices:

* **zeros(n,m)**

Creates matrix of zeroswith size nxm.

* **ones(n,m)**

Creates matrix of oneswith size nxm.

* **eye (n,m)**
* Creates matrix of identity with size nxm.

**MATLAB OPERATORS:**

**B.1 ARITHMETIC OPERATORS:**

Expressions use familiar arithmetic operators and precedence rules.

|  |  |
| --- | --- |
| **Operator** | **Operation** |
| **+** | **Addition** |
| **-** | **Subtraction** |
| **\*** | **Multiplication** |
| **/** | **division** |
| **\** | **Left** |
| **‘** | **complex conjugate transpose** |
| **( )** | **specify evaluation order** |
| **^** | **Power** |

**B.2 LOGICAL OPERATORS:**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Operation** | **Priority** |
| **~** | **NOT** | **Highest** |
| **&** | **Element wise AND** |  |
| **|** | **Element wise OR** |  |
| **&&** | **Short-circuit AND** |  |
| **||** | **Short-circuit OR** | **Lowest** |

**B.3 ELEMENT WISE OPERATORS:**

|  |  |
| --- | --- |
| **.\*** | **Element wisemultiplication** |
| **./** | **Elementwise division** |
| **.^** | **Elementwise raising to a scalar power** |

**B.4 LOGICALCONDITIONS:**

|  |  |
| --- | --- |
| **Operator** | **Operation** |
| **==** | **Equal to** |
| **<** | **Less than** |
| **>** | **Greater than** |
| **~=** | **Not equal to** |
| **~** | **Not** |

**FLOW CONTROL STATEMENTS:**

Matlab has 5 flow control statements. They are

* if statements
* switch statements
* for loops
* while loops
* break statements

**IF STATEMENT:**

**If**statement condition:

The general form of the ifstatement is

Ifexpression

statements

elseifexpression

statements

else

statements

end

**SWITCH STATEMENT:**

**Switch** - Switch among several cases based on expression.

The general form of the switch statement is:

switchswitch\_expr

casecase\_expr,

statement, ..., statement

case {case\_expr1, case\_expr2, case\_expr3,...}

statement, ..., statement

...

otherwise,

statement,..., statement

end

**FOR STATEMENT:**

**For-** Repeat statements a specific number of times.

The general form of a for statement is:

for variable = expr,

statement, ...,

end

(or)

for variable1=expr,

for variable2=expr,

statement, ...,

end

end

**WHILE STATEMENT:**

**While -**Repeat statements an indefinitenumber of times.

The general form of a whilestatement is:

while expression

statements

end

**VISUALIZATION AND GRAPHICS:**

• **plot(x,y), plot(x,sin(x))** *-* plot 1-D function

• **figure , figure(k)** *-* open a new figure

• **hold on, hold off** *-* refreshing

• **mesh(x\_ax,y\_ax,z\_mat)** *-* view surface

• **subplot(3,1,2)** *-* locate several plots in figure

• **axis ([xminxmaxyminymax]) *-*** change axes

• **title(‘figure title’)** *-* add title to figure.

**B.5 MATLAB SIGNAL PROCESSING COMMANDS:**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **COMMAND** | **FUNCTION** |
| 1 | wavread | Load an input sound signal |
| 2 | Rand | Return each different value you use |
| 3 | Mdwtdec | Computes wavelet decomposition |
| 4 | Remmean | Removes old mean and returns new vectors |
| 5 | haar | Type of wavelet to be performed |
| 6 | Cov | Normalization is performed |
| 7 | Eig | Calculation of eigen valued vectors |
| 8 | Whitenv | Places the vectored values in rows |
| 9 | Filter | Helps to decompose the signal |
| 10 | dwt 2 | Computes 2D wavelets transform |
| 11 | idwt2 | Computes the inverse wavelet transform |
| 12 | dwt | Single-stage discrete one-dimensional wavelet decomposition |
| 13 | idwt | Single-stage discrete one-dimensional wavelet reconstruction |
| 14 | Clc | Clears the screen |
| 15 | close all | Closes all the figure windows created by MATLAB programs |

**APPENDIX C**

**CODE IMPLIMENTATION**

**SUB CODES:**

**TO REMOVE THE MEAN FROM THE VECTORS:**

function [newVectors, meanValue] = remmean(vectors)

%REMMEAN - remove the mean from vectors

%

% [newVectors, meanValue] = remmean(vectors);

%

% Removes the mean of row vectors.

% Returns the new vectors and the mean.

%

% This function is needed by FASTICA and FASTICAG

% @(#)$Id: remmean.m,v 1.2 2003/04/05 14:23:58 jarmoExp $

newVectors = zeros (size (vectors));

meanValue = mean(vectors')';

newVectors = vectors – meanValue \* ones (1,size (vectors, 2));

**FOR WHITENING:**

function [newVectors, whiteningMatrix, dewhiteningMatrix] = whitenv ...

(vectors, E, D, s\_verbose)

%WHITENV - Whitenv vectors.

%

% [newVectors, whiteningMatrix, dewhiteningMatrix] = ...

% whitenv(vectors, E, D, verbose);

%

% Whitens the data (row vectors) and reduces dimension. Returns

% the whitened vectors (row vectors), whitening and dewhitening matrices.

%

% ARGUMENTS

%

% vectors Data in row vectors.

% E Eigenvector matrix from function 'pcamat'

% D Diagonal eigenvalue matrix from function 'pcamat'

% verbose Optional. Default is 'on'

%

% EXAMPLE

% [E, D] = pcamat(vectors);

% [nv, wm, dwm] = whitenv(vectors, E, D);

%

%

% This function is needed by FASTICA and FASTICAG

%

% See also PCAMAT

% @(#)$Id: whitenv.m,v 1.3 2003/10/12 09:04:43 jarmoExp $

% ========================================================

% Default value for 'verbose'

ifnargin< 4, s\_verbose = 'on'; end

% Check the optional parameter verbose;

switch lower(s\_verbose)

case 'on'

b\_verbose = 1;

case 'off'

b\_verbose = 0;

otherwise

error(printf('Illegal value [ %s ] for parameter: ''verbose''\n', s\_verbose));

end

% ========================================================

% In some cases, rounding errors in Matlab cause negative

% eigenvalues (elements in the diagonal of D). Since it

% is difficult to know when this happens, it is difficult

% to correct it automatically. Therefore an error is

% signalled and the correction is left to the user.

if any (diag (D) < 0),

error (printf (['[ %d ] negative eigenvalues computed from the' ...

' covariance matrix.\nThese are due to rounding' ...

' errors in Matlab (the correct eigenvalues are\n' ...

'probably very small).\nTo correct the situation,' ...

' please reduce the number of dimensions in the' ...

' data\nby using the ''lastEig'' argument in' ...

' function FASTICA, or ''Reduce dim.'' button\nin' ...

' the graphical user interface.'], ...

sum (diag (D) < 0)));

end

% ========================================================

% Calculate the whitening and dewhitening matrices (these handle

% dimensionality simultaneously).

WhiteningMatrix = E'/(sqrt (D));

DewhiteningMatrix = E \* sqrt (D);

% Project to the eigenvectors of the covariance matrix.

% Whiten the samples and reduce dimension simultaneously.

ifb\_verbose, fprintf ('Whitening...\n'); end

newVectors = whiteningMatrix \* vectors;

% ========================================================

% Just some security...

if ~isreal(newVectors)

error ('Whitened vectors have imaginary values.');

end

% Print some information to user

ifb\_verbose

fprintf ('Check: covariance differs from identity by [ %g ].\n', ...

max (max (abs (cov (newVectors', 1) - eye (size (newVectors, 1))))));

end

**MAIN CODE:**

%x=wavread('boy.wav');

%y=5\*wavread('liu.wav');

h=ones(1,1);

%y=y(1:size(x));

[y,fs] = wavread('v1.wav');

length(y);

%y = y(1:100000,1);

% y = mdwtdec('r',y,6,'db2');

[z1,h11] = dwt(y,'haar');

%[x,h111] = dwt(x,'db1');

z=h\*(z1');

z=remmean(z);%Remove mean

v=cov(z',1);

[E,D]=eig(v);

[x1 A]=whitenv(z,E,D);%Whiten

m=length(z);

%Value w

w=ones(1,1);

w=w/norm(w);%Normalized weighted Matrix

epslon=1e-6;n=0;

%check convenge performance

% filter\_current = filter\_current + 2\*step\_size\*error\*input\_vector;

while (abs(w(1))>=epslon )

y1=w'\*x1;

w=sum(x1\*tanh(y1)',2)/m-sum(1-tanh(y1).^2)/m\*w;

n=n+1;

end;

%display Interation Number;

%plot

w=w\*1e2;

y2=w'\*x1;

y3=x1(1,:)-y2\*((x1(1,:)\*y2')/(y2\*y2'));

size(y2)

%size(h111)

size(h11)

y2 = idwt(y2,h11','haar');

%subplot(3,2,1);plot(y);title('sig1')

%subplot(3,2,2);plot(y);title('sig2')

figure

subplot(1,1,1);

plot(z(1,:));

title('mixed signal');

figure

subplot(1,1,1);

plot(y2);

title('output');

figure

subplot(1,1,1);

plot(y3);

title('noise signal');

soundsc(100\*y2.\*1e30,fs);

soundsc(y3,fs/2);

**BIBLIOGRAPHY**

[1] AapoHyvarinen and ErkkiOja, “**Independent Component Analysis**, ”Neural Networks,13(4 5):411-430,2000.

[2] Kenneth E.Hild and David Pinto,”**Convolutive Blind Source Separation by minimizing Mutual information between segments of signals**”,IEEE transactions on Circuits and systems, regular papers, vol.52, No.10, October 2005.

[3] Ozgur Yilmaz and Scott Rickard,”**Blind Separation of Speech Mixtures via TimeFrequency Masking**”,IEEEtransactions on signal processing, vol.52,No.7,July 2004.

[4] E. Visser and T. W. Lee, “**Speech enhancement using blind source separation and two channel energy based speaker detection**”, IEEE Int.Conf. on Acoust., Speech, and Signal Process., vol. 1, pp. 884–887, April 2003.

[5] NedelkoGrbic;Xiao-JiaoTao;SxenNordholmIngrarClaesson,”**Blind signal separation using overcomplete subband representation**”,IEEE Transactions on speech and audio processing,Vol.9,no.5 123–135.

[6] David L. Donoho, “**Denoising via soft thresholding**”. IEEETransactions on Information Theory, 41:613-627, May 1995.

[7] William Addison and Stephen Roberts, “**Blind Source Separation with Non Stationary Mixing Using Wavelets**, “Pattern Analysis Reasearch Group, The University of Liverpool,2006.

[8] RobiPolikar, “**The Engineer’s Ultimate Guide To Wavelet Analysis**,” Hosted by Rowan University, College of Engineering Web Servers, Last major updates January 2001.

[9] George Tzanetakis, Georg Essl, Perry Cook, “AudioAnalysis using the Discrete Wavelet Transform, ”Computer Science Department also Music Department, Princeton University.

[10] AkshayDayal, John Steinbauer, Angela Qian and MarkEastaway, “**Blind source separation via ICA:Math Behind Method**, ”version 1.1:Dec 19,2007 8:41 PM US/Central.

[11] Carl Taswell, Computational Toolsmiths, “**What, Hoe, and Why of Wavelet Shrinkage Denoising**”, Stanford, CA 94309-9925.