ATLANTA
120th ASEE Annual Conference & Exposition
FRANKLY, WE DO GIVE A D*MN
June 23-26, 2013

Paper ID #6013

Matching Pursuits in EEG Analysis

Dr. Mohammad Rafiq Muqri, DeVry University, Pomona
Mr. Furqan Muqri, UT Southwestern Medical School
Prof. Shih Ek Chng, DeVry University

Matching Pursuits in EEG Analysis :
Development of a diagnostic tool

An Electroencephalogram (EEG) signal is the recording of the electrical activity (voltage fluctuations) along the scalp due to the currents that flow during synaptic excitations of the dendrites of many pyramidal neurons in the cerebral cortex. When neurons are activated, the synaptic currents are produced within the dendrites. This current produces the magnetic field giving rise to mapping of cortical activity measurable by magneto-encephalography (MEG) machines and a secondary electrical field over the scalp measurable by EEG systems. Mastery and expertise in clinical EEG interpretation is one of the most desirable diagnostic clinical skills in interpreting seizures, epilepsy, sleep disorders, and other neurocognitive studies.

In most cases EEG activity is described in terms of frequency, amplitude, distribution or location, symmetry, synchrony, reactivity, morphology, rhythmicity and regulation. The dynamic nature of epileptic phenomena causes EEG signals to exhibit stochastic and non-stationary behavior. The time frequency distributions are potentially very useful for detecting and analyzing non-stationary epileptic EEGs. Although visual analysis of raw EEG traces is still the major clinical tool and the point of reference for other methods, we can relate visual analysis to mathematics with a time-frequency description. The EEG signal analysis is often complemented with MEG and functional magnetic resonance imaging (fMRI) to correlate specific EEG findings with pathology of the brain and selectively demonstrate the diagnosis of certain neuronal disease processes, and assessment parameters.

This diagnostic tool is in no way to be taken as a final word to replace physician's clinical consultation and opinion, rather it is intended to be an early monitoring and warning tool which will aid in diagnosis of certain aspects of critical care and emergency medicine. This interactive teaching module will be highly beneficial since it will facilitate progressive learning of students by enhancing their understanding of clinical EEG parameters and their relationship with differential diagnosis of the patients.

Introduction

The advances in computing have given a new meaning to the term applied signal processing. Enhanced methods are becoming available to users with little mathematical background. Adaptive time frequency approximations of signals with known algorithms and implementation of the matching pursuits is computer intensive. This inhibited their everyday practical applications before the last decade, but today they can run on standard laptops, tablets and even smart phones.

Acquiring data from the heart's electrical activity as an electrocardiogram (EKG), from muscles as an electromyogram (EMG), from the eyes as an electrooculogram (EOG), from the brain as an electroencephalogram (EEG) or as a magnetoencephalogram (MEG),[12] has become vital for early diagnosis of a variety of diseases. A wide array of medical and neurological disorders can cause changes in the EEG patterns. These changes are not always pathognomonic, and they may tend to be highly variable, nonspecific and sometimes even limit their diagnostic importance. Unlike traditional books on EEG, this paper will attempt to concentrate on pattern recognition, identification of rhythms and waveforms. It also explains the laboratory setup of a conventional 10-20 electrode based EEG monitoring station using modern data acquisition tool and software for EEG feature extraction. Students will begin their analysis by looking at frequency, amplitude, distribution or location, symmetry, synchrony, reactivity, morphology, rhythmicity, and regulation and correlate the characteristic appearance on the EEG with existing conditions, certain pathology, and drug or electrolyte effects.

A diagnostic tool using Java programming has been developed. The graphical user interface will be used in conjunction with EEG monitoring system to correlate specific visual EEG findings with pathology of the brain and selectively demonstrate polysomnography, neurocognitive studies, and neurological assessment parameters.

EEG Lab Instrument Setup

To ensure that EEG recordings are reproducible and consistent from one recording to next, a standard for electrode placement was developed in the 1950s by Dr. Herbert Jasper at the Montreal neurological Institute. This widely accepted scheme is known as the International 10-20 System of Electrode Placement. The underlying principle is that accurate measurements of the skull using specific identifiable landmarks can be subdivided into smaller distances based on 10% or 20% increments of the total distance. The three primary measurements are in the sagittal, coronal, and horizontal planes. The sagittal measurement is from nasion (notch at the top of the nose) to the inion (midline prominence at the occipital pole). The coronal measurement is from just anterior to the tragus of the ear in front of the auditory canal to the midpoint of the saggital measurement to the opposite tragus. The intersection of these two lines is defined as the vertex. The horizontal measurement is made connecting points that are 10 % up from nasion, inion, and tragus points on the sagittal and coronal lines. This circumference measurement defines the horizontal plane and is based on an iterative subdivision of arcs on the scalp starting from the craniometric reference points: Nasion, Inion, Left (PAL) and Right (PAR) pre-auricular points. The intersection of the longitudinal (Ns-In) and lateral (PAL-PAR) is named the Vertex.

The original 10-20 system included only 19 electrodes (see panel B of the figure below). Later on, extensions were proposed so that now you can place over 70 electrodes in standard positions (see panel C of the figure). This extension also renamed four electrodes (marked in black in the figure); the original names were: T3, T5, T4, and T6 for T7, P7, T8, and P8, respectively. Odd numbers indicate the left hemisphere, even numbers indicate the right hemisphere.

By convention, electrodes are named using capital letters to designate the underlying brain lobe or region. (F = Frontal, T = Temporal, C = Central, O = Occipital, P = Parietal)

The letters “p” and “z” denote the frontopolar and midline regions.

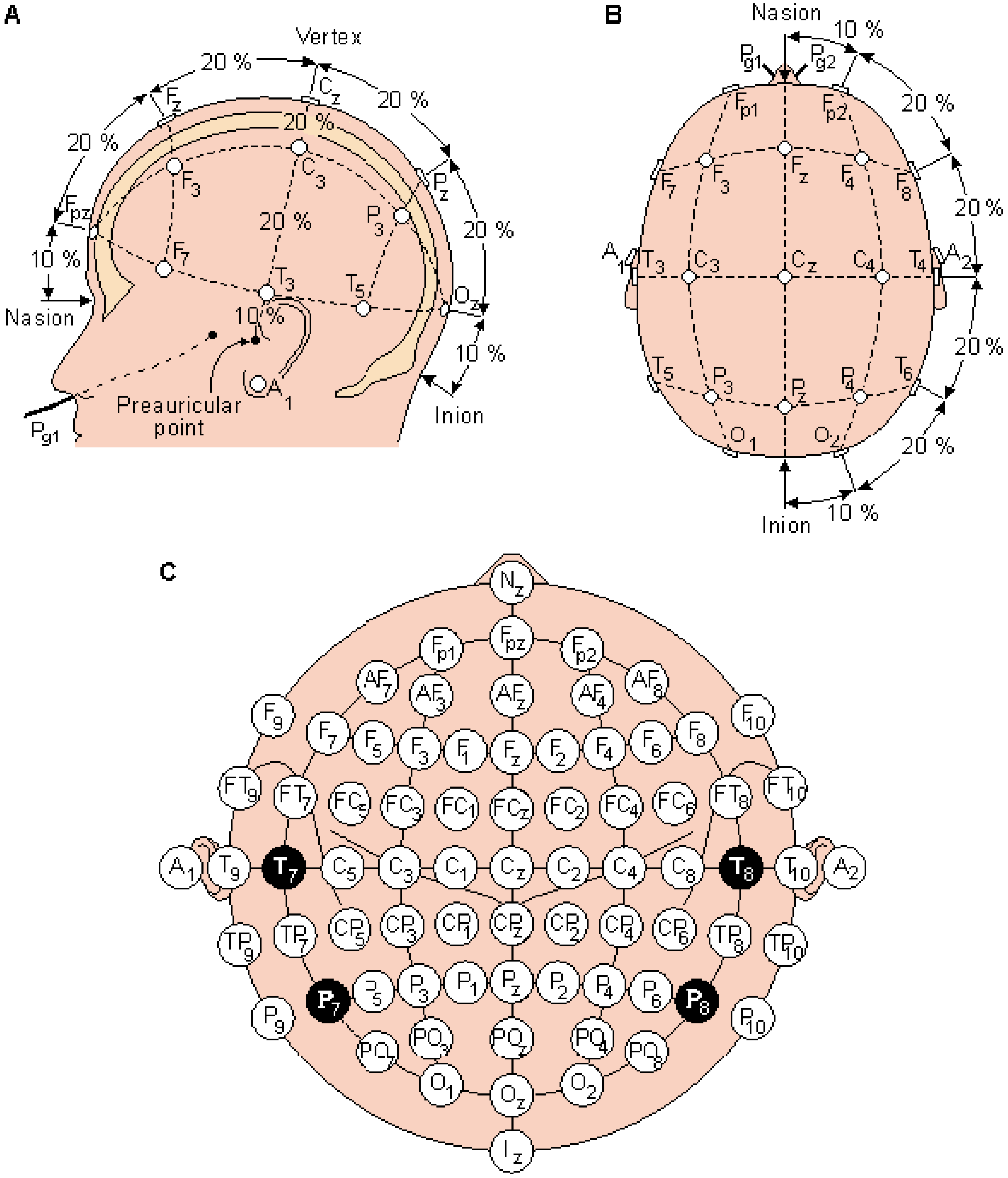
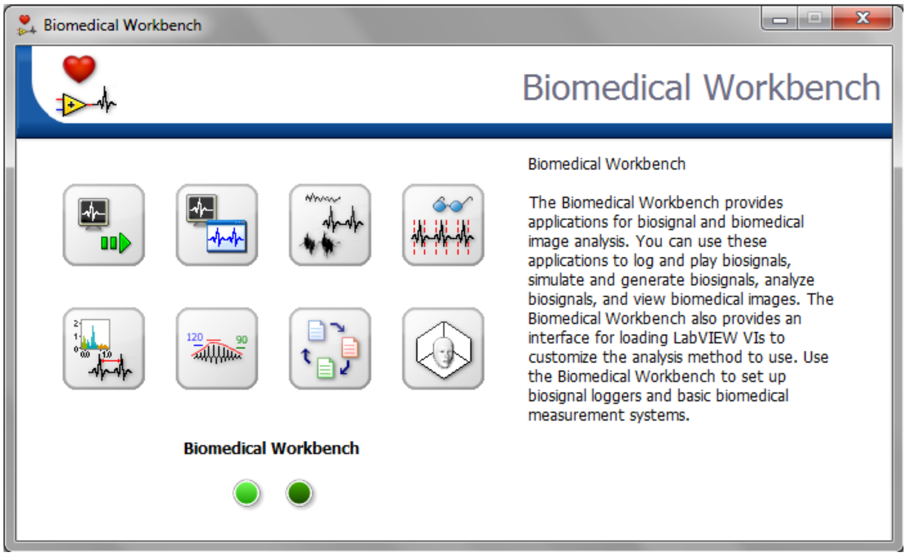

A
Vertex
20 %
C_z
20 %
F_z
20 %
20 %
F_3
C_3
20 %
F_{pz}
P_3
P_z
20 %
10 %
F_7
T_3
T_5
O_z
Nasion
10 %
10 %
A_1
Preauricular point
Inion
P_{g1}
B
Nasion
10 %
P_{g1}
P_{g2}
F_{p1}
F_{p2}
20 %
F_7
F_3
F_z
F_4
F_8
20 %
A_1
T_3
C_3
C_z
C_4
T_4
A_2
T_5
P_3
P_z
P_4
T_6
20 %
O_1
O_2
20 %
Inion
10 %
C
N_z
F_{p1}
F_{pz}
F_{p2}
AF_7
AF_3
AF_z
AF_4
AF_8
F_9
F_7
F_5
F_3
F_1
F_z
F_2
F_4
F_6
F_8
F_{10}
FT_9
FT_7
FC_5
FC_3
FC_1
FC_z
FC_2
FC_4
FC_6
FT_8
FT_{10}
A_1
T_9
T_7
C_5
C_3
C_1
C_z
C_2
C_4
C_6
T_8
T_{10}
A_2
TP_9
TP_7
CP_5
CP_3
CP_1
CP_z
CP_2
CP_4
CP_6
TP_8
TP_{10}
P_9
P_7
P_5
P_3
P_1
P_z
P_2
P_4
P_6
P_8
P_{10}
PO_7
PO_3
PO_z
PO_4
PO_8
O_1
O_z
O_2
I_z

Figure 1

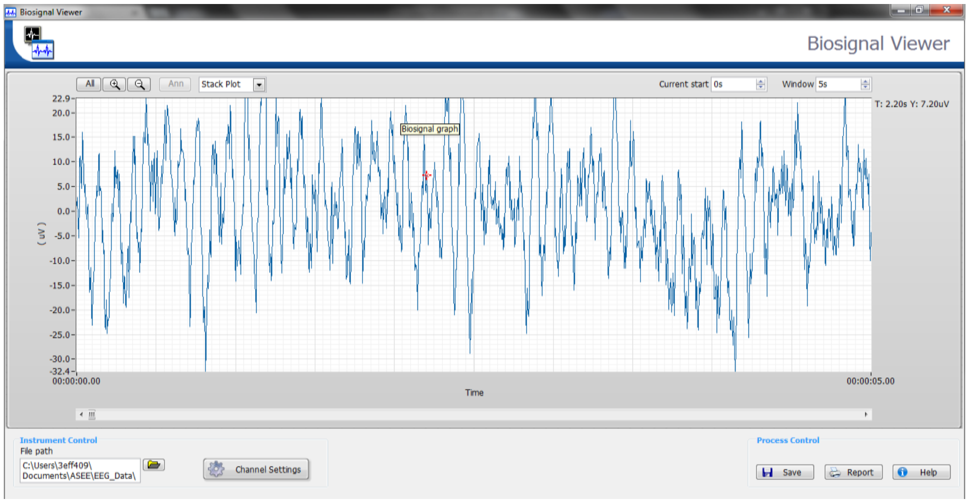
The diagram in figure 2 shows the new Biomedical Tool Kit from National Instrument. This application software can be used for real-time acquisition or generation of simulated EEG signal. Figure 3 shows an example of a five-second duration of the time-domain EEG signal display using the Biosignal Viewer. Every five-second duration of the recorded EEG data is processed individually, and this procedure continues until the very last segment of time. Since EEG signal contains multiple channels of data corresponding to different electrode position on the brain, similar processing of the EEG signal is carried out for each channel.



Biomedical Workbench

The Biomedical Workbench provides applications for biosignal and biomedical image analysis. You can use these applications to log and play biosignals, simulate and generate biosignals, analyze biosignals, and view biomedical images. The Biomedical Workbench also provides an interface for loading LabVIEW VIs to customize the analysis method to use. Use the Biomedical Workbench to set up biosignal loggers and basic biomedical measurement systems.

Figure 2



Stack Plot

Current start 0s

Window 5s

T: 2.20s Y: 7.20uV

File path
C:\Users\3eff409\
Documents\ASEE\EEG_Data\

Figure 3

Biosignal Viewer Report

File Name: C:\Users\3eff409\Documents\ASEE\EEG_Data\EEG_Data_1.tdms
Report Time: 3/14/2013 11:47:59 AM
User Name: 3eff409

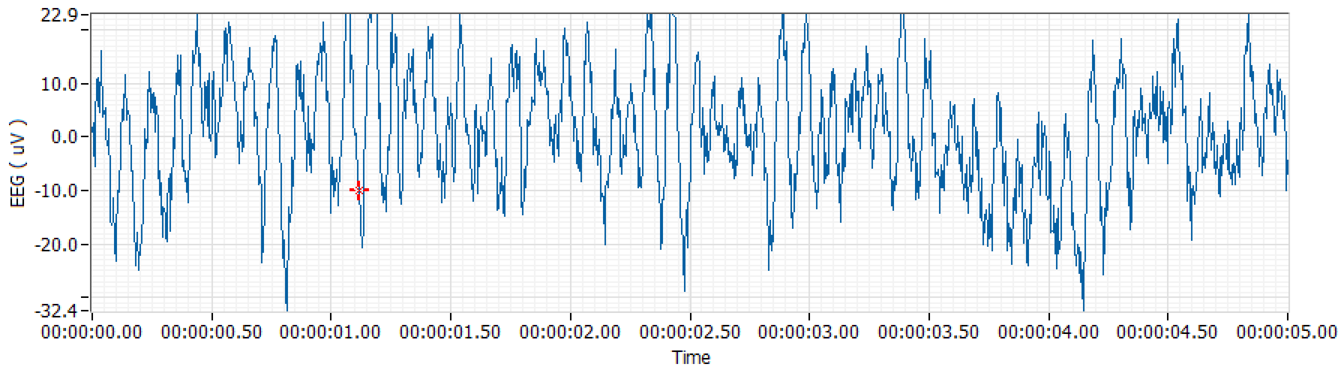
General

Channel Name	**Sampling Rate (Hz)**	**Unit**
EEG	256	uV

Measurements Result

Channel Name	**Cursor Position**
EEG	(00:00:01.11 , -10.11uV)

Plots



Cursor position: (00:00:01.11 , -10.11uV)

Figure 4

Abnormal spike activity that is associated with epilepsy can be measured using the cursor position as shown in Figure 4. Unlike the ECG signal which has specific feature that can be extracted using the ECG Feature Extractor in Biomedical Workbench, the EEG signal is more complex and thus meaningful analysis can only be done by converting into frequency domain using FFT algorithm.

The EEG signals of interest in recording are composed of low amplitude voltage oscillations. While it would be possible to measure each signal relative to electrical ground, most of the important biological information is contained in the small fluctuating voltages associated with synaptic events rather than absolute potential distance from ground. To allow sufficient amplification to see these fluctuating potentials without magnifying the absolute potentials on which they ride, the logical choice is to use a low-frequency filter to eliminate the direct current potentials. With low frequency filters, one can select Low Frequency Filter (LFF) cutoff frequencies of 0.1, 0.3, 1, 3, or 10 Hz, corresponding to time constants of 1.6, 5.3, 0.16, 0.05, or 0.016 second. Even with modern digital equipment, the same LFF values work well with standard scalp EEG frequencies. The most common setting is 1 Hz ($\tau = 0.16$ s), which nicely preserves most low frequency information but largely eliminates drift of the baseline due to dc potentials.

As mentioned earlier with modern digital EEG systems, high and low frequency filtering are usually performed using signal processing software after the digitized signal is recorded, making it possible for the reader to change these values when the study is read and view the effects of several different filter settings on the same EEG trace.

Invasive EEG refers to the surgical intervention [12] that is required to obtain intracranial EEG recordings, and may be obtained either acutely or chronically. Invasive methods provide a focused look at targeted areas of the brain through surgically placed or implanted electrodes directed at sites that have been preselected based upon noninvasive localization. Direct recording of the EEG from the brain is known as electrocorticography (ECoG) and may be acquired either in or outside of the operating room. Chronically implanted or indwelling electrodes are typically used for seizure monitoring prior to respective epilepsy surgery. Invasive EEG records the same electrocerebral potentials as those recorded by scalp EEG. In certain types of epilepsy, removal of the epileptogenic zone within the brain tissue is the only possible therapy due to drug-resistance. However it is still very difficult to delineate the exact localization of the epileptogenic zone without invasive EEG registration.

A primer on FFT, DFT, STFT, Wavelet Transform, and Gabor functions

Proper description of the EEG often requires simultaneous localization in both time and frequency domains. The classical Fourier Transform (FT) analysis is able to achieve infinite frequency resolution, but it does not provide temporal localization information. In case of multichannel EEGs, where the geometrical position of the electrodes reflect the spatial dimension, a space-time-frequency (STF) analysis through multiway processing methods has also become popular. The short-time Fourier transform (STFT) is defined as the discrete-time Fourier transform evaluated over a sliding window.

The wavelet transform (WT) is another alternative for a time-frequency analysis. Unlike the short-time Fourier transform, the time-frequency kernel for the wavelet transform based methods can better localize the signal components in time-frequency space and as such can efficiently exploit the dependency between time and frequency components.

Introduction to Matching Pursuit Algorithm

One of the first applications of matching pursuit (MP) in EEG analysis was detection and parameterization of sleep spindles--structures present in sleep EEG recordings. The matching pursuit (MP) algorithm[3] is often used instead of popular time-frequency (TF) domain approaches such as short-time Fourier transform and the wavelet transform because of its higher temporal-spatial resolution in the TF space. Some preliminary applications of the matching pursuit method have appeared, analyzing routine EEG, ictal scalp EEG, sleep EEG, and evoked potentials. In this study, it is shown that the suitability of the MP analysis for entire seizures, but specifically examine the periods after seizure onset, when the dynamics of the signal are typically rapidly changing.

Given a set of functions (dictionary) D = $\{g_1, g_2, \ldots, g_n\}$such that $\| g_i \| = 1$, we can define an optimal M-approximation as an expansion, minimizing the error ε of an approximation of signal f(t) by M waveforms:

$$\epsilon = \left\| f(t) - \sum_{i=1}^{M} w_i g_{\gamma_i}(t) \right\| \tag{1}$$

Where $\{\gamma_i\}_{i=1..M}$ represents the indices of the chosen functions $g_{\gamma i}$. Finding such an optimal approximation is not practical. A suboptimal expansion can be found by means of an iterative procedure, such as the MP algorithm. In the first step of MP, the waveform $g_{\gamma 0}$ which best matches the signal f(t) is chosen. In each of the consecutive steps, the waveform $g_{\gamma n}$ is matched to the signal R^n f, which is the residual left after subtracting results of previous iterations:

$$\begin{cases} R^0 f = f \\ R^n f = < R^n f, g_{\gamma_n} > g_{\gamma_n} + R^{n+1} f \\ g_{\gamma_n} = \arg\max_{g_{\gamma_i} \in D} | < R^n f, g_{\gamma_i} > | \end{cases} \tag{2}$$

Orthogonality of R^{n+1} f and $g_{\gamma n}$ in each step implies energy conservation:

$$||f||^2 = \sum_{n=0}^{m-1} | < R^n f,\ g_{\gamma_n} > |^2 + ||R^m f||^2 \tag{3}$$

For a complete dictionary the procedure converges to f :

$$f = \sum_{n=0}^{\infty} < R^n f,\ g_{\gamma_n} > g_{\gamma_n} \tag{4}$$

From this equation we can derive a time-frequency distribution of the signal's energy. The (5) is shown below, that is free of cross-terms, by adding Wigner distributions of selected functions:

$$Ef(t,\omega) = \sum_{n=0}^{M} | < R^n f,\ g_{\gamma_n} > |^2 \ Wg_{\gamma_n}(t,\omega)$$

Discrete Gabor Dictionary

Other method is the Gabor representation [5] that does not assume that the signal is known at arbitrary time and frequency points, but at a lattice points: t = *n*T, w = *k*F;
where *n, k* ∈ *Z; T* is the sampling interval in the time domain; and F is the sampling interval in the frequency domain. Figure 5 demonstrates examples of different shapes of Gabor functions, which can be included in the dictionary used for MP decomposition.

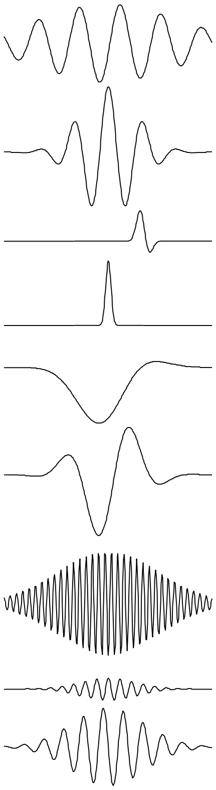


Figure 5

Gabor functions are essentially sine-modulated Gaussian functions that provide optimal joint time-frequency localization. A real Gabor function can be expressed as:

$$g_\gamma(t) = K(\gamma)e^{-\pi\left(\frac{t-u}{s}\right)^2} \sin\left(2\pi\frac{\omega}{N}(t-u)+\phi\right) \qquad (6)$$

N is the size of the signal for which the dictionary is constructed and $K_{(\gamma)}$ is such that $\| g_i \| = 1$. $\gamma = \{\mu, \omega, s, \phi\}$, denoting parameters of the dictionary's functions (time-frequency atoms). The length of signal (N points) suggests ranges for the parameters of Gabor functions which can be reasonably fitted to such an epoch. However, within these ranges no particular sampling is a priory defined, and we are confronted with a three-dimensional continuous space; this results in an infinite dictionary size. Therefore, in practical implementations, we use subsets of the possible dictionary functions.

In the dictionary implemented originally by Mallat and Zhang, [3] the parameters of the atoms are chosen from dyadic sequences of integers. Their sampling is governed by an extra parameter—octave j (integer). Scale s, which corresponds to an atom's width in time, is derived from the dyadic sequence $s = 2^j$, $0 <= j <= L$ (signal size $N = 2^L$). Parameters μ and ω, which correspond to an atom's position in time and frequency, are sampled for each octave j with interval $s = 2^j$, or, if oversampling by l is introduced, with interval $s = 2^{j-l}$.

There is considerable interest in identifying patterns of seizure discharges that would characterize complex partial seizures originating from different brain regions. Modern efforts have dealt with visual identification of seizure patterns and rhythms, either from scalp, sphenoidal, or intracranial recordings. The utilization of intracranial recording electrodes, either subdural strips, grids or depth electrode arrays, reduces artifact and facilitates recording and localization of seizures in many instances. Visual interpretations of the recordings, however, are still severely restricted by the fact that EEG signals, even from a single channels, rather than being a continuous periodic waveform, more typically have components of multiple frequencies and may be rapidly evolving.

The ability to further analyze seizure activity is often limited by methods that require stationary epochs of data. Fourier spectral analysis measures frequency distributions for selected epochs but it is not optimal for analysis of rapidly changing data. Application of the Fourier transform assumes relative stationarity of the signal or treats it as such for the analysis window. With less stationary signals, Fourier analyses can be performed with shorter time analysis windows (Short Time Fourier transform, STFT), but Fourier analysis is still best suited to investigations of background rhythms and not dynamic events such as seizures.

The aim of time-frequency analysis is to offer the ability to analyze relatively long continuous segments of seizure activity even when the dynamics of the signal are rapidly changing. One approach to accurately analyze non-stationary signals such as the ictal EEG is to represent the signal as a sum of waveforms with well-defined time-frequency properties. In Fourier analysis these waveforms are cosines and sines which provide localization in frequency but not in time.

The wavelet transform methods use base waveforms with localization both in time and frequency. Wavelet transforms do not require the fixed data window needed for Fourier analysis and are applicable to analysis of non-stationary signals. However, even wavelet bases are not well suited to exact simultaneous representation of time and frequency domains of signal components whose localizations in time and frequency vary widely. Wavelet analyses are best suited for analysis of random spikes or phenomena such as evoked potentials. Similarly the Wigner distribution (WD), while allowing for excellent decomposition of non-stationary signals over time, is most appropriately applied to signals of a single component; multicomponent signals produce interference due to cross-terms.

Video-EEG Monitoring and Epilepsy Surgery

Long-term video-EEG monitoring with scalp electrodes is a well-established tool that permits time-locked correlation of paroxysmal clinical events with electrographic data. Video-EEG monitoring enables the accurate classification of seizures for optimal choice of medical therapy and, for patients with medically intractable partial epilepsy, assists in localization of the epileptogenic focus for possible surgical resection. Video-EEG monitoring is also helpful in monitoring the response to treatment of epilepsy. Seizure frequency documentation in epilepsies characterized by typical and atypical absence seizures and for clinical manifestations and monitoring of epileptiform activity in sleep studies can be accomplished as well.

The monitoring of sleep is complex and requires a distinct skill set including a detailed knowledge of EEG, respiratory monitoring, and EKG. Expertise in only one of these areas does not confer the ability to accurately interpret the polysomnogram. Sleep is not homogeneous and is characterized by sleep stages based on EEG, EOG, and EMG. Since reading, analyzing and interpreting noninvasive and invasive EEG for epilepsy, polysomnography and neurocognitive studies are huge topics, the focus will be mainly on EEG interpretation by leveraging [2] the power of Java graphical user interface (GUI) and its data structures. So as to not be overly verbose in our descriptions of abnormal and normal EEGs and evoked potentials, only a success story will be told.

Given below is an example of an instructor lead program which the student edited, compiled and displayed. The purpose of this program is to show how general Java workhorse discrete Fourier Transform and other EEG analysis methods [7] can be introduced at an early stage to engineering technology students with tools and concepts that will be further reinforced in EEG analysis and future DSP courses.

```
public class Fourier {
    public static double[] discreteFT(double[]fdata, int N, boolean fwd){
        double X[] = new double[2*N];
        double omega;
        int k, ki, kr, n;
        if (fwd){
            omega = 2.0*Math.PI/N;
        } else {
            omega = -2.0*Math.PI/N;
        }
    for(k=0; k<N; k++) {
        kr = 2*k;
        ki = 2*k + 1;
        X[kr] = 0.0;
        X[ki] = 0.0;
        for(n=0; n<N; ++n) {
```

```
                        X[kr] += fdata[2*n]*Math.cos(omega*n*k) +
fdata[2*n+1]*Math.sin(omega*n*k);
                        X[ki] += -fdata[2*n]*Math.sin(omega*n*k) +
fdata[2*n+1]*Math.cos(omega*n*k);
                         }
            }
        if ( fwd ) {
                for(k=0; k<N; ++k) {
                        X[2*k] /= N;
                        X[2*k + 1] /= N;
                }
        }
        return X;
        }
        }
```

The TestDFT application class given below uses class Fourier and invokes its methods.

```
//TestDFT Program

public class TestDFT {
 public static void main(String args[]) {
            int N = 64;
            double T = 2.0;
            double tn, fk;
            double fdata[] = new double[2*N];
            for(int i=0; i<N; ++i) {
                    fdata[2*i] = Math.cos(4.0*Math.PI*i*T/N);
                    fdata[2*i+1] = 0.0;
            }
 double X[] = Fourier.discreteFT(fdata, N, true);
 for (int k=0; k<N; ++k) {
   fk = k/T;
   System.out.println("f["+k+"] = "+fk+"Xr["+k+"] = "+X[2*k]+ "  Xi["+k+"] = "+X[2*k + 1]) ;
}
 for (int i=0; i<N; ++i) {
            fdata[2*i] = 0.0;
            fdata[2*i+1] = 0.0;
            if (i == 4 || i == N-4 ) {
                    fdata[2*i] = 0.5;
            }
 }
 double x[] = Fourier.discreteFT(fdata, N, false);
 System.out.println();
 for (int n=0; n<N; ++n) {
   tn = n*T/N;
    System.out.println("t["+n+"] = "+tn+"xr["+n+"] = "+x[2*n]+"  xi["+n+"]  = "+x[2*n + 1]);
    }
   }
}
```

Here is a sample of TestDFT results:

```
f[0] = 0.0Xr[0] = -1.1275702593849246E-16  Xi[0] = 0.0
f[1] = 0.5Xr[1] = -4.5102810375396984E-17  Xi[1] = -6.505213034913027E-17
f[2] = 1.0Xr[2] = -5.898059818321144E-17  Xi[2] = -3.426078865054194E-17
f[3] = 1.5Xr[3] = 3.469446951953614E-17  Xi[3] = -1.5872719805187785E-16
```

Highly efficient algorithms for computing the DFT were first developed in the 1960s. Collectively known as Fast Fourier Transforms (FFTs), they all rely upon the fact that the standard DFT involves redundant calculation. Strictly speaking, there is no such thing as ‘the FFT’ [3]. Rather, there is a collection of algorithms with different features, advantages, and limitations. An algorithm which is suitable for programming in a high level-language on a general purpose computer may not be the best for special purpose DSP hardware. What the different algorithms have in common is their general approach – the decomposition of the DFT into a number of successively shorter, and simpler, DFTs.

There are various ways of explaining FFT decomposition. We can show that a DFT can be expressed in terms of shorter, simpler DFTs by dividing the signal x[n] into subsequences. This method which is widely used in DSP literature is also referred to as conventional decomposition. There is also an alternative approach known as index-mapping. It should be clear in our mind that conventional decomposition and index mapping are just two ways of looking at the same problem and there is no essential difference between them.

Suppose there is a signal with N sample values, where N is an integer power of 2. We first separate x[n] into two subsequences, each with N/2 samples. The first subsequence consists of even number points in x[n], and the second consists of odd number points- Writing n = 2k, when n is even, and n = 2k + 1 when n is odd. This can be expressed as the original N-point DSP in terms of two N/2 point DFTs. The decomposition can be taken further, by breaking each N/2-point subsequence down into two shorter, N/4-point subsequences. The process can continue until there is only a series of 2-point subsequences, each of which requiring a very simple 2-pointDFT. A complete decomposition of this type gives rise to one of the commonly used radix-2, decimation in time, FFT algorithms.

The students can implement FFT as a Java method. It is called the fastFFT( ) method and is also defined in the Fourier class.

```
public class Fourier {
public static void fastFFT(double[ ] fdata, int N, boolean fwd) {
                    double omega, tempr, tempi, fscale;
                    double xtemp, cosine, sine, xr, xi;
                    int i, j, k, n, m, M;
```

```
    j=0;
    for(i=0; i<N-1; i++) {
        if (i<j) {
            tempr = fdata[2*i];
            tempi = fdata[2*i + 1];
            fdata[2*i] = fdata[2*j];
            fdata[2*i + 1] = fdata[2*j + 1];
            fdata[2*j] = tempr;
            fdata[2*j + 1] = tempi;        }
    k = N/2;
    while (k <= j) {
        j  -=  k;
        k >>= 1; }
    j  += k;
    }
if (fwd)
    fscale = 1.0;
else
    fscale = -1.0;
M = 2;
while( M < 2*N ) {
    omega = fscale*2.0*Math.PI/M;
    sin = Math.sin(omega);
    cos = Math.cos(omega) - 1.0;
    xr = 1.0;
    xi = 0.0;
    for (m=0; m<M-1; m+=2) {
        for (i=m; i<2*N; i+=M*2) {
    j = i + m ;
            tempr = xr*fdata[j] - xi*fdata[j+1];
            tempi = xr*fdata[j+1] + xi*fdata[j];
            fdata[j] = fdata[i] - tempr;
            fdata[j+1] = fdata[i+1] - tempi;
            fdata[i] += tempr;
            fdata[i+1] += tempi; }
    xtemp = xr;
    xr = xr + xr*cos - xi*sin;
    xi = xi + xtemp*sin +xi*cos;
    }
M *=2;
}
if( fwd ) {
    for (k=0; k<N; k++) {
        fdata[2*k] /= N;
        fdata[2*k + 1] /= N;
    }
```

```
        }
    }
```

In the next step, the students apply this method to compute the FFT on a 2Hz cosine wave, whereby 64 data samples were taken over a 2-second sample period. The program first computes the FFT to obtain the frequency spectrum for a 2Hz cosine wave. Then the program below was used by students to perform an inverse Fourier transform that reconstructs the 2Hz cosine wave from its frequency spectrum. Next, they implemented the FFT as another Java method. It is also called the fastFFT( ). They tested the fastFFT( ) by applying it to the composite cosine signal that was processed earlier. The amplitude time history for a signal containing three different frequency components is generated and sent to the fastFFT () method. The TestFFT class source code is shown below:

```
public class TestFFT {
public static void main(String args[ ]) {
        int N = 64;
        double T = 1.0;
        double tn, fk;
        double fdata[ ] = new double[2*N];

        for(int i=0; i<N; ++i) {
                fdata[2*i] = Math.cos(8.0*Math.PI*i*T/N) +
Math.cos(14.0*Math.PI*i*T/N) +
   Math.cos(32.0*Math.PI*i*T/N);
                fdata[2*i+1] = 0.0;
                }
Fourier.fastFT(fdata, N, true);
System.out.println();
for(int k=0; k<N; ++k) {
        fk = k/T;
System.out.println(  "f["+k+"] = " + fk + " Xr["+k+"] = " + fdata[2*k] + " Xi["+k+"]
                                                                  = " + fdata[2*k+1]) ;
         }
      }
}
```

Here are the sample partial TestFFT results:

```
f[0] = 0.0 Xr[0] = 14.118483415068333 Xi[0] = 8.299076200364345
f[1] = 1.0 Xr[1] = -4.320681943456142 Xi[1] = -11.70913283509337
f[2] = 11.5931285252644 Xi[2] = 11.382484516157067
f[3] = 3.0 Xr[3] = 0.20410293867180673 Xi[3] = -9.83647671361791
…

f[60] = 60.0 Xr[60] = 0.012868248913439052 Xi[60] = 0.058867631665321246
f[61] = 61.0 Xr[61] = -0.20682636511072414 Xi[61] = 0.025288953333891355
f[62] = 62.0 Xr[62] = -0.03895429084677146 Xi[62] = -0.007474788441969488
```

```
f[63] = 63.0 Xr[63] = 0.02651390602928162 Xi[63] = 0.0
```

On most computers, only some of the total computation time of an FFT is spent performing the FFT butterfly computations- determining indices, loading and storing data, computing loop parameters and other operations consume the majority of cycles. Careful programming that allows the compiler to generate efficient code can make a several-fold improvement in the run-time of an FFT. The best choice of radix in terms of program speed may depend more on characteristics of the hardware (such as the number of CPU registers) or compiler than on the exact number of computations.

As stated earlier Gabor functions are the functions which best represent signals in the time-frequency domain in the sense that they possess the smallest product of effective duration by effective frequency width. The MP algorithm efficiently decomposes a signal into waveforms that belong to a dictionary of functions which include the Gabor functions known as time-frequency atoms.

MP is an iterative procedure which can be best described as follows:

1. Find (in the dictionary) the first function that best fits the signal.
2. Subtract its contribution from the signal.
3. Repeat these steps on the remaining residuals, until the representation of the signal in terms of chosen functions is satisfactory.

The onset and propagation of partial complex seizures are evolving and dynamic processes. Complex partial seizures[10] frequently originate from mesial temporal lobe structures, but can also originate from the lateral temporal neocortex as well as extratemporal regions. Accurate localization of complex partial seizures is of particular importance in the treatment of individuals with complex partial seizures, particularly when surgical resection is being considered since the surgical approach to a patient with mesial temporal lobe epilepsy is much different than that for the patient with seizures originating from neocortical regions, either temporal or extratemporal. Because of the patterns of seizure onset, evolution and propagation may differ depending upon the region.

A review of the literature reveals that a good number of researchers have applied the MP method in analyzing routine EEG, ictal scalp EEG, sleep EEG, and evoked potentials. The application of MP analysis to continuous segments of seizure activity recorded from intracranial depth electrodes in patients with seizures originating from mesial temporal structures is discussed here. This case study was selected for our students because it demonstrates the suitability of the MP analysis for entire seizures and because the study specifically examines the periods after seizure onset, when the dynamics of the signal are typically rapidly changing.

Methods of Data Acquisition

Data from patients monitored[7] prior to seizure surgery for intractable complex partial seizures were analyzed. All patients had intracranial EEG (icEEG).

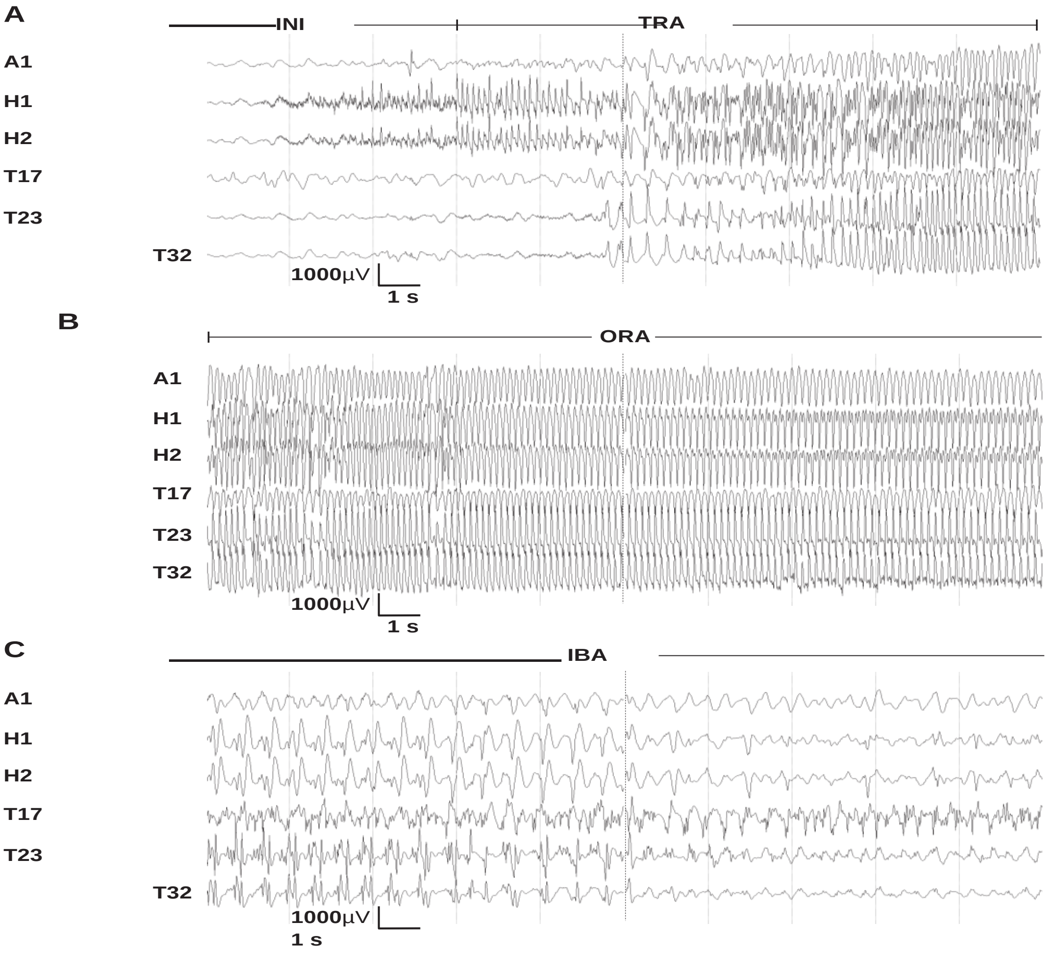


Figure 6

Experimental Setup and EEG Signal Analysis

A typical experimental setup for EEG signal analysis [7] simulated in the laboratory is described next. Figure 6 was selected from the case study to illustrate the selected segments of a complex partial seizure originating from the mesial temporal structures and recorded with combined depth electrodes and a 32-contact subdural grid array are illustrated (patient 1). Two 6-contact depth electrode arrays pass through the grid and with the deepest contacts Al and Hl H2 residing in the amygdala and hippocampus and the most superficial contacts (e.g. A6 H6) at the lateral temporal neocortex.

Note that the seizure begins nearest the deepest hippocampal contacts (Hl H2) and subsequently spreads to regionally involve the temporal lobe. To facilitate illustration only three depth electrode contacts (those nearest the seizure onset) and three representative subdural grid contacts (two Tl7 T23) from the anterior region one (T32) from the posterior region of the 32-contact grid) are shown. Panel A shows 20 seconds of intracranial EEG (ICEEG) that includes the later portions of the period of seizure initiation (INI; the low voltage fast activity was preceded by some periodic spiking not shown) and the period of transitional rhythmic activity (TRA) l4 seconds. Panel B continuous with A shows 20 seconds of ICEEG showing the initial 20 seconds of organized rhythmic activity (ORA) Panel C is 20 seconds of ICEEG late in the seizure beginning 40 seconds after panel B and illustrates the pattern of intermittent bursting activity (IBA) frequently seen at the conclusion of a seizure.

Decisions to perform intracranial monitoring were based on needs for functional mapping of eloquent cortex for e.g. language mapping of the dominant temporal lobe and seizure localization. Only seizures from patients having good mesial temporal location of depth electrodes as confirmed by MRI were included. All nine patients selected for this case study had seizures with mesial temporal onset as determined by visual inspection of the combined recordings in Fig 6.

For this case study, [7] a 64 channel video-EEG recording system (64-channel Telefactor MODAC) was used to digitize and store the EEG signals at a rate of 2 samples per second. For these MP analysis recordings from the depth electrode contact showing the earliest seizure onset were selected. The entire seizure was analyzed in each instance. A low pass digital filter with a 50 Hz cutoff frequency was employed. For display purposes the MP decompositions were typically computed until 9% of the energy was represented.

The next program [8] demonstrates the power of java's advanced GUIs and Graphics. In this program JLPViewer.java, an array of strings is created consisting of the names of the EEG images. We also want the user to select only one EEG (item) image at a time from pJL (pictureJList), so we list pJL to single selection mode by using the method setSelectionMode together with the constant ListSelectionModel.SINGLE_SELECTION.

When the user selects an image from pJL, the program displays the corresponding EKG image using the JLabel object dPicJL (displayPictureJList). We use the class ImageIcon and create an array of images which include eeg1, eeg2, eeg3, eeg4, eeg5 etc.

There are five items in the JList. When the program executes, it displays only three of these image items in the list at a time. Therefore , we want to attach a vertical scroll bar to the JList object pJL, so that user can scroll to select an EEG not currently shown in the list. We set the pane layout to null and specify the size and location of the GUI components. These statements are provided in the complete program listing given next.

```
// JLPViewer.java
// JLPViewer application

// version 1.00 2013/01/04

import java.awt.* ;
import java.awt.event.* ;
import javax.swing.*;
import javax.swing.event.*;

public class JLPViewer extends JFrame implements ListSelectionListener {
  private String[] pNames = {"eeg 1", "eeg 2", "eeg 3", "eeg 4","eeg 5" } ;
  private ImageIcon[] pics = {new ImageIcon("eeg1.gif"),new ImageIcon("eeg2.gif"),new
ImageIcon("eeg3.gif"),new ImageIcon("eeg4.gif"),new ImageIcon("eeg5.gif")};
        private BorderLayout lBL ;
        private JList pJL ;
        private JScrollPane sJS ;
        private JLabel promptJL ;
        private JLabel dPicJL ;
        private JLabel infoJL ;
        public JLPViewer( )
        {
                super ("EEG Viewer");
                Container pane = getContentPane();
                pane.setLayout(null);
                promptJL = new JLabel(" Select an Electroencephalogram",
                SwingConstants.CENTER);
                promptJL.setSize(350,10);
                promptJL.setLocation(10,10);
                pane.add(promptJL);
                pJL = new JList(pNames);
                pJL.setVisibleRowCount(3);
                pJL.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
                pJL.addListSelectionListener(this);
                sJS = new JScrollPane(pJL);
                sJS.setSize(350,60);
                sJS.setLocation(10,20);
                pane.add(sJS);
                dPicJL = new JLabel(pics[4]) ;
                dPicJL.setSize(350, 350) ;
                dPicJL.setLocation(10,50) ;
                pane.add(dPicJL) ;
                infoJL = new JLabel(pNames[4],SwingConstants.CENTER);
                infoJL.setSize(350, 20) ;
                infoJL.setLocation(10, 380) ;
                pane.add(infoJL) ;
                setSize(480, 540) ;
                setVisible(true);
                setDefaultCloseOperation(EXIT_ON_CLOSE);
        }
        public static void main(String args[])
        {
                JLPViewer pV = new JLPViewer();
        }
        public void valueChanged(ListSelectionEvent e )
        {
                dPicJL.setIcon(
                        pics[pJL.getSelectedIndex() ]);
                infoJL.setText(
                        pNames[pJL.getSelectedIndex() ]);
                repaint() ;
        }
  }
```

Students were introduced early only to object based programming with Java, but also to basic concepts of discrete Fourier, fast Fourier Transforms, continuous and discrete time Wavelet Transforms, and MP algorithm. Having being exposed to this Java programming, DFT, FFT, Wavelet transform learning module has far reaching implications. Advances in technology and the widespread availability of powerful computing devices have given a new meaning to the term applied signal processing for real world applications as such recording and analysis of electrical activity of the human brain was the fascinating chosen context for this paper.

Conclusion

End of the course survey and diagnostic quizzes demonstrated the enhanced student understanding of DSP material which is again attributed to early exposure of Matlab scripting, Java Object Oriented Programming and the reinforcement of DFT, FFT, Wavelet transforms and EEG component analysis to which they had been exposed later on during their junior and senior years. The authors wish to stress that this paper is no attempt to challenge previous clinical or diagnostic knowledge. It is hoped that the concepts covered in this paper will instigate future research and development in the field. It will potentially enhance the significant clinically related previously applied methods for EEG analysis and thereby aid aspiring student researchers and physicians to optimize and ultimately provide more cost effective and efficient diagnostic tools.

Bibliography

1. Muqri, M., Shakib, J., *A Taste of Java-Discrete and Fast Fourier Transforms*, American Society for Engineering Education, AC 2011-451.
2. Shakib, J., Muqri, M., *Leveraging the Power of Java in the Enterprise*, American Society for Engineering Education, AC 2010-1701.
3. Mallat, S., Zhang, Z., *Matching pursuit with time-frequency dictionaries*, IEEE Trans. Signal Process., 41, 1993, 3397-3415.
4. Rangayyan, R., *Biomedical Signals analysis. A case-study approach,* IEEE Press on Biomedical Signals. Calgary, Alberta, Canada, 2002
5. Feichtinger, H., Strohmer, T., *Gabor Analysis and algorithms: Theory and applications*, Editors. Birkhauser, Boston, 1998.
6. Blinowska, K., Durka, P., *The application of wavelet transform and matching pursuit to the time-varying EEG signals,* in Intelligent Engineering Systems Through Artificial Neural Networks, Editors, Dagli & Fernandez, volume 4, pp. 535-540, ASME Press, New York, l994.
7. Franszczuk, P., Bergey, G., Durka, P., Eisenberg, H., *Time frequency analysis using the matching pursuit algorithm applied to seizures originating from the mesial temporal lobe*, Journal of Electroencephalography and Clinical Neurophysiology, volume 106, pp.513-521, Elsevier, 1998.
8. Deitel, H.M., Deitel, P.J., *Java How to program*, Prentice Hall, 2003.
9. Palmer G., *Technical Java - Developing Scientific and Engineering Applications*, Prentice Hall, 2003.
10. Sanei, S., Chambers, J., *EEG Signal Processing*, John Wiley & Sons, Ltd, West Sussex, England, 2007.
11. Greenfield, L., Geyer, J., Carney, P., *Reading EEGs*, Lippincott, 2010.
12. Schomer, D., Lopez da Silva, F., *Niedermeyer's Electroencephalography Basic Principles, Clinical, Applications, and Related Fields*, Lippincott, 2011.
13. Skill Modules, *Advanced Physical Diagnosis Learning and Teaching at the Bedside, Department of Medicine, University of Washington*, http://depts.washington.edu/physdx.html
14. ZhijunGu, NI Biomedical Start up Kit 3.0, July 2010, https://decibel.ni.com/content/docs/DOC-12646