

FREQUENCY DOMAIN ANALYSIS OF BIOMEDICAL SIGNALS

Viktor A. Nedialkov

Technical University - Sofia, Faculty of Telecommunication,
8, "Kliment Ohridsky" str., 1000 Sofia, Bulgaria

Abstract

The paper illustrates signal processing techniques used in the analysis of biomedical signals such as EMG, EEG and Heart Rate Variability, in the frequency domain. Normalization of the signals, FFT transform, windowing and quantitative variable calculations are demonstrated. Using modern programming languages new algorithms are developed for each part of the analysis. The result of this study is a synthesized toolset with algorithms and equations ready to be used in various biomedical signal frequency domain analysis.

1. INTRODUCTION

Frequency domain analysis is very important in the biomedical signal processing. With the rapid improvement of the processing power it is becoming much more usable in various signal analysis even in real time signal processing.

In order to produce the expected results the signal transformations must be performed in very specific sequence and the transformations themselves must be performed accurately.

The most common biomedical signal analysis to date are of the Heart Rate Variability (HRV) for calculating parameters describing the low frequency changes in the HR. These parameters are useful for predicting the risk of developing myocardial infarction and sudden cardiac death. Other areas of frequency analysis are the EEG (Electroencephalogram) where calculating the power spectrum of the signals correlates to the function of specific brain functions and are very important for brain diagnostics. EMG is another area of use of the frequency domain analysis. The calculated spectrum density shows how much each frequency contributes to the muscle contraction, which can be very useful for neurological and muscle disorders. This type of analysis combined with time domain algorithms can improve significantly the reliability of an EMG driven active prosthesis.

In the bottom of every frequency domain analysis lies the Discrete Fourier Transform (DFT) and its computational version Fast Fourier Transform (FFT).

But you can't just put a time series into an FFT and get a frequency domain data. There are many signal and algorithm specific procedures to make in

order to have an accurate and predictable result. This includes: interpolation – not all biomedical time series are uniformly discretized signals. Windowing: this is to minimize the effect of spectral spread due to the DFT expecting the input sequence to be periodic. Then we have the DFT itself in the form of FFT algorithm. Then we have to normalize the result of the FFT which is a complex vector, into a real value representing the frequency spectrum. And at last we may need to average the result of several transformations to reduce the variance error.

All these transformations are described in the following sections.

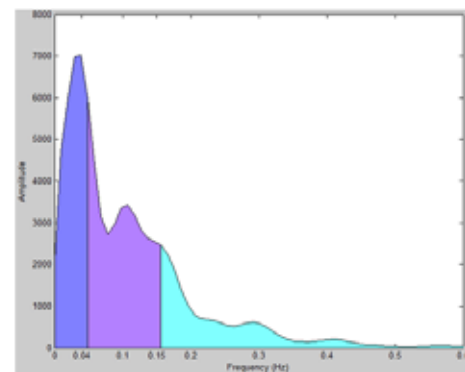


Fig. 1. HRV PSD Distribution

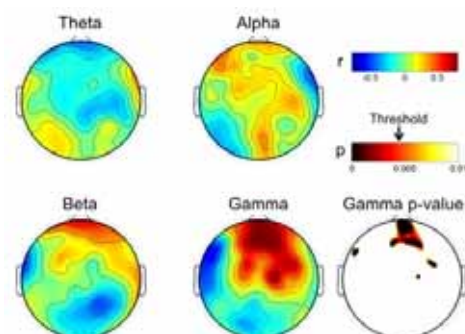


Fig. 2. EEG Frequency mapping

2. INTERPOLATION OF THE SIGNAL

Not all biomedical signals are uniformly sampled series. For example an RR interval time series is a series representing the intervals between two adjacent QRS complexes. In order to obtain a uniform sample frequency we need to resample or interpolate the time series.

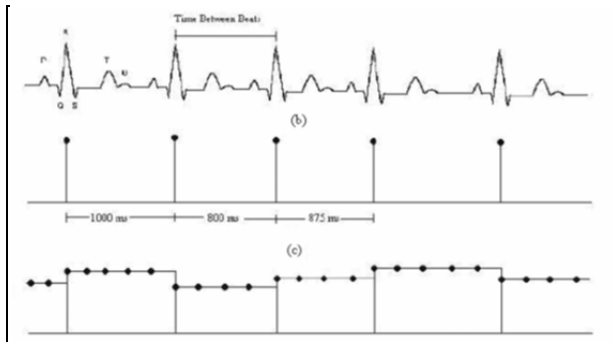


Fig. 3. Interpolation of the RR interval time series

There are two main methods for interpolation used – Linear interpolation and Cubic spline interpolation. The first one is more straight-forward and is suitable for signals which are not with sinusoidal origin and the cubic spline interpolate more accurately the sinusoidal signals.

Resampling is needed also when we need higher bandwidth analysis then the actual sampling frequency in order to meet the Nyquist criteria.

3. WINDOWING

If we simply take a stretch of length N out of a time series containing a sinusoidal signal and perform a DFT, we will most likely find that the sinusoidal signal which we might naively expect to result in a sharp peak in only one frequency bin, will instead show up as something ugly like shown in Figure 4. The reason is that the DFT implicitly assumes that the signal is periodic, i.e. that the time series of length N repeats itself infinitely in a cyclic manner. If the frequency of the sinusoidal input signal is not an exact multiple of the frequency resolution f_{res} , i.e. does not fall in the exact center of a frequency bin, this assumption is not true, and the DFT will 'see' a discontinuity between the last sample and the first sample due to the cyclic continuation. That discontinuity spreads power all across the spectrum.

The remedy is to multiply the time series with a 'window function' in the time domain before applying the DFT. This window function starts near or at

zero, then increases to a maximum at the center of the time series and decreases again. Thus the discontinuity is removed. Many window functions have been defined and given names. They usually involve some compromise between the width of the resulting peak in the frequency domain, the amplitude accuracy and the rate of decrease of the spectral leakage into other frequency bins.

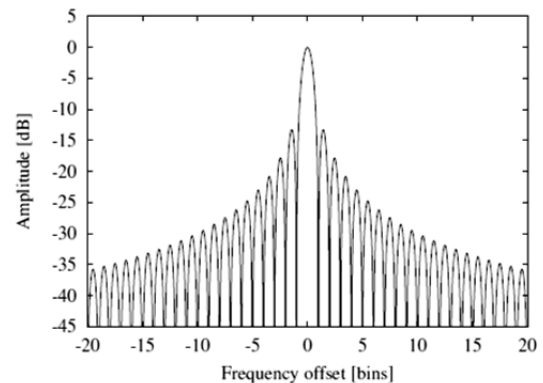


Fig. 4. Frequency response to a rectangular window

For our purpose we are using the simple but useful Hanning window.

A window function to be used with a DFT of length N is defined by a vector of real numbers $\{w_j\}$, $j = 0 \dots N-1$. It is used by multiplying the time series x_j with the window before performing the DFT, i.e. using $x_{1j} = x_j * w_j$ as input to the DFT.

The Hanning window is defined as follows:

$$w_j = \frac{1}{2} \left[1 - \cos\left(\frac{2\pi \cdot j}{N}\right) \right]; \quad j = 0 \dots N-1,$$

We define the following two sums for normalization purposes:

$$S_1 = \sum_{j=0}^{N-1} w_j,$$

$$S_2 = \sum_{j=0}^{N-1} w_j^2.$$

We will use them in the normalization of the complex result of the DFT.

Because we will use S_1 and S_2 in the normalization of our final results, we can multiply the window values w_j with any convenient constant factor.

4. FFT

After we have calculated the windowed time sequence we are ready to compute the DFT with the FFT algorithm. We are using a Radix2 FFT in our study.

We have developed a function

Radix2(XREtmp, XIMtmp, Nf);

It takes as parameters the data arrays for the real and imaginary part of the result and the length of the time series.

The code of the Radix2 implementation is not shown here but it is a classic Radix2 algorithm shown in many articles around the internet.

4. SCALING AND NORMALIZATION

The normalized equivalent noise bandwidth NENBW of the window, expressed in frequency bins, is given by

$$\text{NENBW} = N \frac{S_2}{(S_1)^2}.$$

The effective noise bandwidth ENBW is given by

$$\text{ENBW} = \text{NENBW} \cdot f_{\text{res}} = \text{NENBW} \cdot \frac{f_s}{N} = f_s \frac{S_2}{(S_1)^2},$$

where f_s is the sampling frequency and f_{res} the width of one frequency bin. For the Hanning window we have $\text{NENBW} = 1.5$ bins. This equivalent noise bandwidth is required when the resulting spectrum is to be expressed as spectral density (such as for noise measurements). It can be understood by considering white noise as input to our algorithm. Due to the width of the window in the frequency domain, each frequency bin collects not only the noise in that frequency bin, but also from adjacent bins. Dividing the result by the effective noise bandwidth corrects for this phenomenon.

Now we come back to the problem of normalizing the results of the FFT. Assume we have an input time series x_j of length N . After multiplication with the chosen window function, it is subjected to a real-to-complex FFT as defined in the previous section. We will also need the window sums S_1 and S_2 defined in section 3.

The result of the FFT is a complex vector y_m of length $N = 2 + 1$. We interpret it as a power spectrum, expressed as V^2_{rms} , as follows:

$$PS_{\text{rms}}(f_m = m \cdot f_{\text{res}}) = \frac{2|y_m|^2}{S_1^2};$$

$$m = 0 \dots N/2,$$

The factor S_1 takes the role of N . It takes into account both the length N of the DFT and the gain of the window function, plus any constant factor that was used in the computation of the window values w_j .

The factor 2 originates from the fact that we use an efficient FFT algorithm that does not compute the redundant results for negative frequencies.

We now return to the scaling of the “normal” results. If the desired result is a power spectral density (PSD) expressed in V^2/Hz , it is obtained by dividing the power spectrum (PS) by the effective noise-equivalent bandwidth ENBW:

$$\begin{aligned} \text{PSD}_{\text{rms}}(f_m = m \cdot f_{\text{res}}) &= \\ &= \frac{PS_{\text{rms}}(f_m)}{\text{ENBW}} = \frac{2|y_m|^2}{f_s \cdot S_2}; \quad m = 0 \dots N/2, \end{aligned}$$

Since the ratio between the peak and the noise plateau in the FFT output depends on N , we need to distinguish carefully between spectra and spectral densities. The magic number to convert them into each other is the effective noise bandwidth ENBW, which should hence always be recorded when a spectrum or spectral density is computed, such that the result can be converted to the other form at a later stage, when the information about the frequency resolution f_{res} and the window that was used is normally not easily available any more.

Further processing of the output is straightforward: If the desired result is a linear spectrum (expressed in V) or a linear spectral density (expressed in V/pHz), simply take the square root of the corresponding power spectrum or spectral density:

$$\text{LSD} = \sqrt{\text{PSD}},$$

$$\text{LS} = \sqrt{\text{PS}}.$$

Finally the result can be converted into other units (such as V_{pk} or dB).

5. AVERAGING AND OVERLAP

If we compute one estimate of a spectrum with the methods described so far (i.e. multiplying one segment of the time series with a suitable window function, performing a DFT and scaling the results), we will typically find the result to be rather 'noisy'. The theory confirms this practical observation: The standard deviation of the spectrum estimate in one frequency bin is equal to the estimate itself, i.e. 100%, if the signal in that bin is stochastic. It does not help the increase the length N of the DFT; that only reduces the width of one frequency bin without improving the variance.

The usual remedy is to take the average of M estimates and hence reduce the standard deviation of the averaged result by a factor of $1/\sqrt{M}$. However, the properties of the signal must remain stationary during the averaging. Note that the averaging must be done with the power spectrum (PS) or the power spectral density (PSD), not with their square roots LS or LSD. If the square roots are desired as result, they must be computed at the end after the averaging is finished. In conjunction with the use of window functions, this method of averaging several spectra is also known under names such as "Welch's method of averaging modified periodograms", "Welch's overlapped Segmented average" etc.

If a long continuous data stream is simply split into several non-overlapping segments of length N and each segment is processed by a DFT with a window function, we have a situation as illustrated in Figure 5.

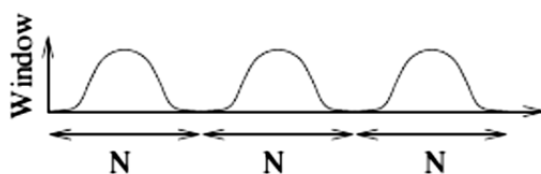


Fig. 5. Segmented data stream with window and without overlap

Due to the fact that the window function is typically very small or zero near its boundaries, a significant portion of the data stream is effectively ignored in the analysis. This is clearly not optimal in those situations where the data stream was produced at great expense, and maximal possible information is to be extracted from it. The situation can be im-

proved by letting the segments overlap, as illustrated in Figure 6.

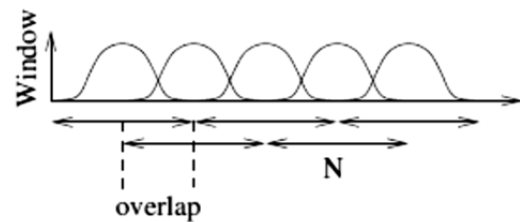


Fig. 6. Segmented data stream with windows and overlap

Figure 7 and Figure 8 shows the PSD calculated from a single and multiple averaged data segments. The averaged result is much smoother.

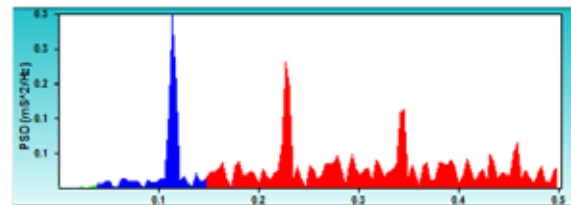


Fig. 7. Power spectrum density from a single data segment

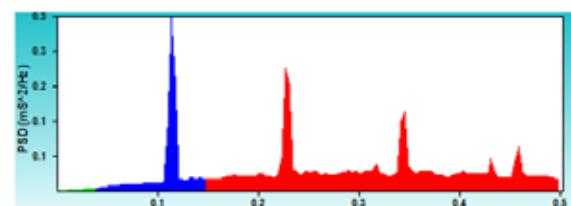


Fig. 8. Power spectrum density from several averaged data segments

6. CONCLUSIONS

The transformation of the signals from the time domain to the frequency domain must be performed very carefully and all of the above procedures must be performed in order to produce accurate results. Further studied must be performed on the different windows that can be used with the DFT. Also we did not remove the constant component of the signal which will appear in the bin 0 in the transformed signal.

Understanding the exact procedures for transforming time series into frequency domain gives us invaluable instrument for analysis of biomedical signals and effectively develop new algorithms for all kind of biomedical devices.

References

- [1] Rangaraj M. Rangayyan "Biomedical Signal Analysis" Wiley-Interscience.
- [2] Webster J, "Medical Instrumentation – application and design" John Wiley & Son.
- [3] R. Ivanov, "Digital processing of single dimensional signals", Almamater international.
- [4] G. Nenov, "Signals and systems", Novi Znanja.
- [5] R. Ivanov, "Digital processing of single dimensional signals", Almamater international
- [6] G. Heinzel, A. Ruudiger and R. Schilling "Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new at-top windows", Max-Planck-Institut fur Gravitationsphysik
- [7] Andrew J. Barbour and Robert L. Parker, "Normalization of Power Spectral Density estimates"
- [8] J. Schesser "HRV Analysis" BME 333 Biomedical Signals and Systems.