

## USER INTERFACE FOR ANALYSIS OF POLYGRAPHIC RECORDINGS OF PHYSIOLOGICAL SIGNALS IN EUROPEAN DATA FORMAT

Tarmo LIPPING and Andres ANIER

Biomedical Engineering Centre, Tallinn Technical University, Ehitajate tee 5, 19086 Tallinn, Estonia; tl@pori.tut.fi

Received 15 January 1999, in revised form 14 July 1999

**Abstract.** In this paper we present a user interface for analysing recordings of various physiological signals (electroencephalogram, electrocardiogram, blood pressure, etc.). The recordings are in European Data Format. We first introduce the data format and then describe the analysis environment, which runs on MATLAB software, version 5, but can also be compiled into executable code for PC. The user interface is open in the sense that it can easily be modified to support different data formats and to include new application tools. Finally, we give an example how the analysis environment can be used for wavelet analysis.

**Key words:** polygraphic recordings, physiological signal processing, open analysis environment, wavelet analysis.

### 1. INTRODUCTION

With rapidly developing technology, the amount of patient data measured in hospitals is growing all the time. Most of the data are used for patient monitoring during, e.g., clinical operation or the patient's stay in intensive care unit and can be forgotten as soon as the situation is over. However, sometimes we want to save the data to compare the patient's state after some treatment to his/her state before the treatment to decide on the results. Also, comparing the measurements taken from patients in similar clinical state can give us information on how analogous situations can be treated better.

Another current trend in physiological signal interpretation is multivariate analysis. This means that instead of looking at individual signals, information is gathered from different measurements and decisions are made after joint analysis

of the measurements. For example, it has been noticed that monitoring blood pressure signals alone in some cases does not give enough information to prevent brain deterioration during heart surgery. Electroencephalogram (EEG) signal can serve as an additional source of information. However, it is very difficult to extract the needed information from EEG as the changes in the signal due to anaesthesia are similar to those due to deterioration in brain activity.

The above-mentioned trends suggest the need for a well-determined data format for physiological measurements and a collection of tools to deal with the recorded data. In this paper we consider the widely used European Data Format (EDF) and present a MATLAB-based open environment for analysing the data and for developing and testing new analysis methods before they are used in medical equipment. The data format can be thought of as a "black box" for clinical use, where patient information is saved and which can later be opened to see what actually happened during a critical treatment and which parameters may be useful to predict the changes in patient's state.

## 2. EUROPEAN DATA FORMAT

European Data Format was developed in 1991 by European engineers who were occupied in medical establishments [1]. The work was coordinated by the Task Group on Signal Analysis, which operated within the European Concerted Action "Methodology for the Analysis of the Sleep-Wakefulness Continuum".

We decided to use EDF for the following reasons:

- EDF is specially designed for recording polygraphic physiological measurements;
- large number of European companies and research groups are using EDF and/or their software is compatible with EDF;
- there exist large data banks with physiological data stored in EDF.

EDF consists of a header part and data records. The variable-length header identifies the patient and specifies the technical characteristics of the recorded signals. It includes the following general fields: the version number of the format, patient identification information, recording specification information, the date and time of the recording, number of bytes in the header, number of data records, length of data records in seconds, and number of signals. After the general fields come the following signal specific fields: signal label (e.g., containing information about the EEG lead), the transducer type, the physical dimension of the signal, calibration information (the lowest and highest physiological and digital values), prefiltering information, and the number of samples in each data record. Using the last field and the length of one data record in seconds, the sampling frequency for each signal can be calculated.

The header is followed by data records. One data record contains the same segment (in time) of all the signals. Each signal value is represented by a 16-bit



integer. The amount of bytes of a certain signal in one data record depends on the sampling frequency of the signal. For example, if the recording consists of two signals, EEG sampled at 100 Hz and electrocardiogram (ECG) sampled at 50 Hz and if the length of the data record is ten seconds, then each data record consists of 2000 bytes of EEG values followed by 1000 bytes of ECG values.

### 3. ANALYSIS SYSTEM

The purpose of the analysis environment is to provide a user-friendly graphical front end for MATLAB. All necessary tools for file and data handling, data visualization and analysis is provided by MATLAB. However, it is inconvenient to read data from EDF into MATLAB, run the analysis functions from MATLAB command prompt, and compare the results of different analysis methods. The developed analysis environment enables the user to visualize acquired data, select the area of interest, and then apply a provided or self-made analysis tool, all through one convenient user interface.

Figure 1 shows the main user interface of the analysis environment. When the environment is started (by typing **main** command in MATLAB command prompt while in the directory where the environment files are situated), an empty array of signal windows appears. The array can be arbitrarily large, limited by the resolution and size of the monitor (in Fig. 1 we have presented a  $2 \times 2$  array of signal windows). There are three main menus in the upper left corner of the window: 1) **File** for opening EDF files or exiting from the environment, 2) **Sync: On/Off** for selecting scrolling of signals for all the signal windows synchronously or separately, and 3) **Help**.

When an EDF file is opened either from the **File** menu or by right clicking the mouse in one of the signal windows, a signal can be selected for each of the signal windows. It is possible to open several recordings and choose signals from different recordings. The signal label and the name of the EDF file appear in the upper left corner of the signal window. The physical dimension appears on the left of the window and the horizontal and vertical axis show absolute time from the beginning of the recording and the signal value, respectively. The signals can be scrolled either synchronously or separately using three pairs of buttons at the bottom of the signal window. Scrolling may be done window by window or in some user defined steps. The user can also go to the end or to the beginning of the recording.

When a signal is chosen for a signal window, a pop-up menu allows three further selections: **Analyse**, **Properties** and **Clear**. The last one simply frees the signal window and deletes the previously chosen signal. The **Properties** menu allows the user to choose values for four parameters: the time scale of the signal window, the amount of window lengths by which the signal can be scrolled using the middle button pair at the bottom of the window, and the lowest and the

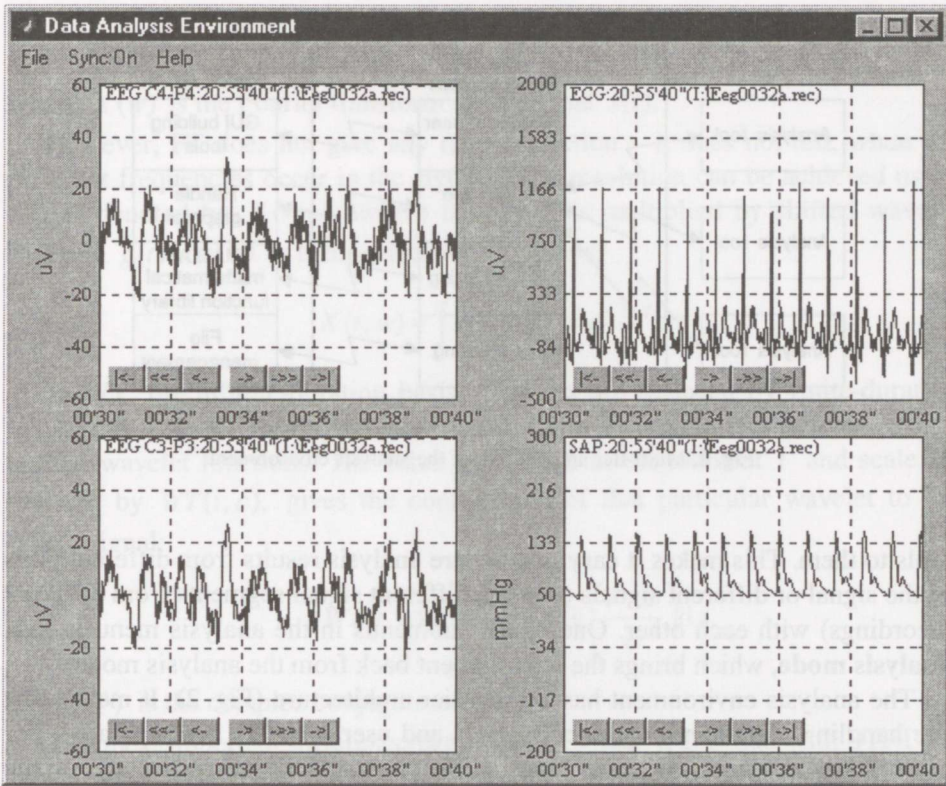


Fig. 1. Main user interface of the analysis system.

highest values in the window (enables vertical scaling). The **Analysis** menu switches the environment into analysis mode.

In the analysis mode the user can draw a box, holding down the left mouse button, around the signal segment the user wants to apply the analysis tool to. When the box is drawn, the right mouse button opens an analysis pop-up menu where the analysis tools are listed. The analysis part of the environment is totally independent in the sense that it does not need to know where the data comes from. All the files containing analysis tools are situated in a separate directory. When the environment is set up, the directory is scanned so that the environment knows the list of the analysis tools. New tools can be added by just adding files to the analysis directory. The files may be based on ready MATLAB functions or they can be totally user-written. After a tool is applied to the signal segment in the box, the result is presented in a manner specified in the analysis tool file, e.g., in case of short time Fourier transform (STFT) the result can be the time-frequency plot of the STFT of the signal segment. One of the analysis tools enables also to save the signal segment in a separate file. The user can select as many boxes from different signal windows as he/she likes and apply arbitrary



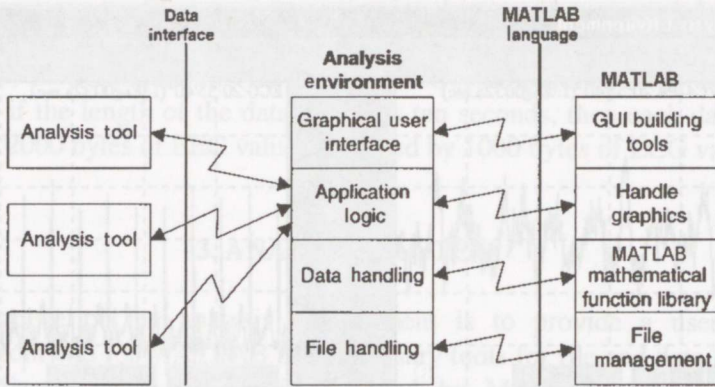


Fig. 2. Multi-tier architecture of the analysis environment.

tools to them. This makes it easy to compare analysis results from different parts of the signal or different signals (or even different signal segments from different recordings) with each other. One of the submenus in the analysis menu is **Exit analysis mode**, which brings the environment back from the analysis mode.

The analysis environment has a multi-tier architecture (Fig. 2). It means that file handling, data handling, data analysis and user interface are separated. For example, if the user needs to change file format, it is necessary to rewrite functions that are related to extracting data from files and there is no need to modify any other functions (for example, signal presentation).

The whole environment has several behavioural parameters that the user may want to modify. These parameters are defined in a MATLAB function called **loadconf**. The user can edit the file to give suitable values to the parameters. Such parameters are, e.g., the number and pattern of signal windows ( $2 \times 1$ ,  $1 \times 2$ ,  $2 \times 2$ , etc.), font and size of the text items, colours of the items of the environment, the location (the directory name) of the analysis tools, etc.

#### 4. EXAMPLE: WAVELET ANALYSIS

In this section we give an example of application of the analysis environment. Although different kinds of applications can be added to the system, a typical application can be comparison of the signal segments in some transform domain. In this example we use the wavelet transform (WT) [2,3].

The WT can be described as an extension of the Fourier transform (FT). Fourier transform reveals the frequency content of the signal, i.e., the amplitude and phase of sinusoids of different frequency the signal consists of. In other words, FT expands the signal on the basis of complex exponentials:

$$X(w) = \int x(t)e^{-j\omega t} dt, \quad (1)$$

where  $X(w)$  is the Fourier transform of the signal  $x(t)$ .

However, FT does not give any time resolution – it does not tell, when the different frequencies occur in the signal. Time resolution can be achieved using STFT, denoted by  $X(t, w)$ , where the signal is multiplied by shifted wavelet function  $g(t)$  and FT is taken window by window:

$$X(t, w) = \int x(\tau)g(\tau - t)e^{-j\omega\tau} d\tau. \quad (2)$$

In WT, the infinite-duration basis functions are replaced by finite-duration wavelet functions. The signal can be expressed as a sum of compressed and shifted wavelet functions. The value of the WT at the moment  $t$  and scale  $a$ , denoted by  $WT(t, a)$ , gives the contribution of that particular wavelet to the whole signal:

$$WT(t, a) = \frac{1}{\sqrt{a}} \int x(\tau) \bar{g}\left(\frac{t - \tau}{a}\right) d\tau. \quad (3)$$

Here  $\bar{g}(\cdot)$  is the complex conjugate of  $g(\cdot)$ .

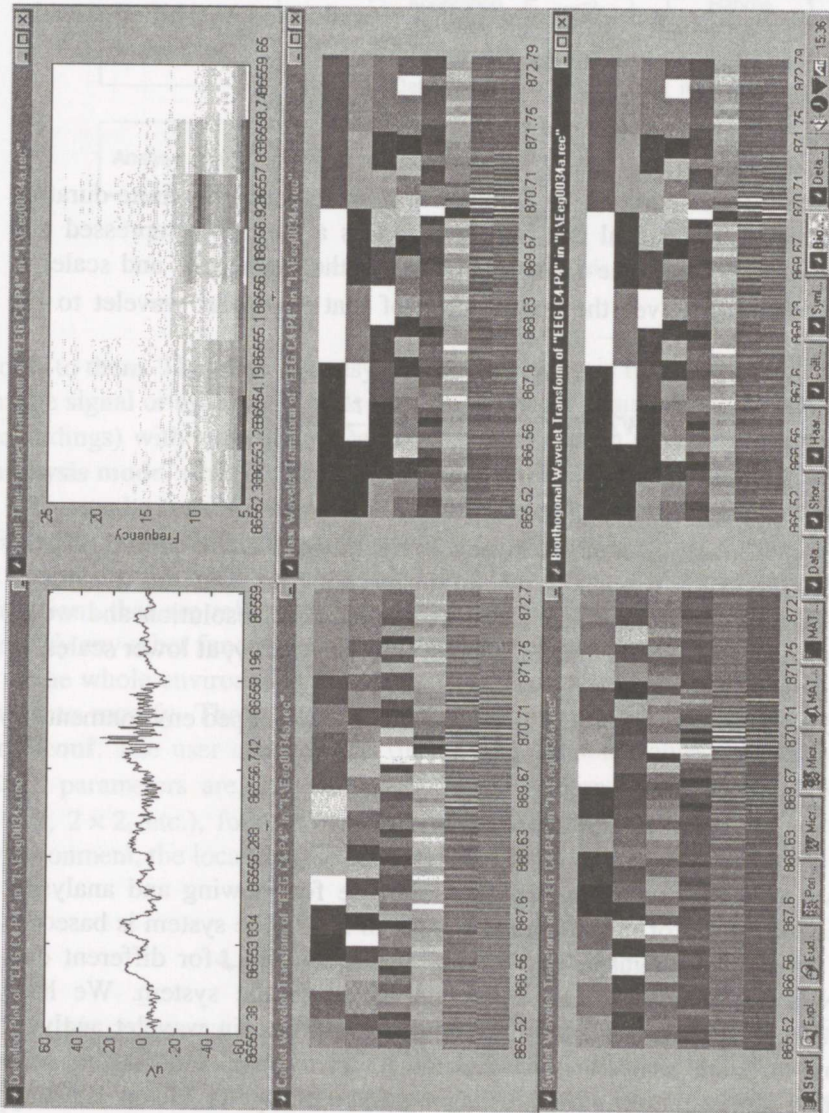
The most important difference between STFT and WT is that with STFT the time resolution and the frequency resolution are constant over the total time-frequency plane while with WT we can use better time resolution and worse frequency resolution at higher frequencies (or, more correctly, at lower scales, as basis functions are not sinusoids).

In Fig. 3 we present a set of results obtained in the developed environment.

## 5. CONCLUSIONS

We have presented a convenient user interface for viewing and analysing polygraphic recordings of physiological signals in EDF. The system is based on MATLAB high level language and is easy to modify, e.g., for different data formats. New analysis tools can easily be added to the system. We have presented an example of application of the user interface in wavelet analysis. The purpose of such analysis is to determine which wavelet basis functions reveal certain signal patterns best. The programs and a test file in EDF are available at <http://www.pori.tut.fi/~tl/EDF/environment.html>. In the near future, we will develop the system further to include new kinds of applications.





**Fig. 3.** An example of possible usage of the system for wavelet analysis: (a) analysed signal segment; (b) STFT with segment length 2 s; (c) Coiflet1 wavelet; (d) STFT with segment length 0.5 s; (e) haar wavelet; (f) Daubechies' wavelet.

## REFERENCES

1. Kemp, B., Värri, A., Rosa, A. C., Nielsen, K., and Gade, J. A simple format for exchange of digitized polygraphic recordings. *Electroencephalogr. Clin. Neurophysiol.*, 1992, **82**, 5, 391-393.
2. *Wavelets: A Tutorial in Theory and Applications* (Chui, C. K., ed.). Academic Press, New York, 1992.
3. Strang, G. and Nguyen, T. *Wavelets and Filter Banks*. Wellesley-Cambridge, Wellesley, 1996.

## KASUTAJALIIDES EDF-FORMAADIS SALVESTATUD FÜSIOLOOGILISTE SIGNAALIDE ANALÜÜSIKS

Tarmo LIPPING ja Andres ANIER

On esitatud autorite poolt välja töötatud kasutajaliides EDF-formaadis olevate füsioloogiliste signaalide salvestuste analüüsimiseks. Kasutajaliides põhineb MATLAB-keel ja kujutab endast avatud süsteemi, mida on lihtne muuta nii erinevate andmeformaadide kui ka uute analüüsivahendite tarvis. Kasutajaliides võimaldab ekraanil vaadelda erinevate salvestuste signaale, valida signaalidest huvipakkuvaid lõike ning rakendada neile analüüsivahendeid. Näitena on toodud ühe signaalilõigu analüüs Fourier' teisenduse ja mitmesuguste lainekeste teisen-  
duste abil.