

A windows computer application for long term data acquisition, storage and presentation of biological signals

Samo Ribarič *, Mihovil Šantić

Medical Faculty, Institute of Pathophysiology, Zaloška 4, SI-1000 Ljubljana, Slovenia

Received 14 July 2000; received in revised form 20 September 2000; accepted 25 September 2000

Abstract

We developed a windows application dedicated to recording, displaying and storing large amounts of biological signals on a standard PC. The application works in Matlab and uses an A/D-converter interface that enables a user to run the same software for signal acquisition, display and processing with different A/D-converter boards. The software program consists of three integrated modules, for setting the protocol for signal acquisition; for commands that can be accessed during data acquisition and for software routines that enable off-line viewing, pre-processing and processing of recorded data. It adapts to different computer hardware configurations by enabling the user to define file size, data resolution and file storage on local and network hard disks during signal acquisition. © 2001 Elsevier Science Ireland Ltd. All rights reserved.

Keywords: Long-term data acquisition; Computer network; Biological signals; Software trigger; Matlab; Real time toolbox

1. Introduction

Computer aided acquisition of biological signals is a well-established method of recording experiments in biomedicine. Although there are many, specialised computer programs there seems to be a lack of applications dedicated to handling large amounts of data, or long term data acquisition on a standard PC. A probable reason for this situation is that modern PCs

with reliable operating systems, large memory and high capacity hard disks have been made available only recently.

Long term data acquisition is a feature in the study of electromyographic (EMG) [1,2] and electroencephalographic (EEG) activity in animal models [3,4] where one experiment can last up to several weeks. Therefore, it would be useful to have a computer program capable of handling large amounts of data over long time intervals. To solve these problems, we developed a windows application dedicated to recording, displaying and storing large amounts of biological signals on a standard PC. However, the program's design is more general and can be used for any kind of analogue signals.

* Corresponding author. Tel.: + 386-1-5437053; fax: + 386-1-5437021.

E-mail address: ribaric@ibmi.mf.uni-lj.si (S. Ribarič).

2. Program description

The application runs in Matlab 5.3 (The Mathworks Inc., USA, <http://www.mathworks.com>) and uses Matlab library functions where convenient (e.g. software filters). Humusoft (Humusoft, Czech Republic, <http://www.humusoft.cz>) wrote the Matlab-A/D-converter interface (i.e. real time toolbox (RTT) 3.0). Humusoft's web site provides a list of drivers that are supported by the RTT 3.0. Requests for drivers, not listed at this site, are considered by the company on a case by case basis. This interface enables conversion of analogue data to a format accessible by Matlab functions and is compatible with many A/D-converter boards that are currently available on the market.

Graphic user interfaces and software modules

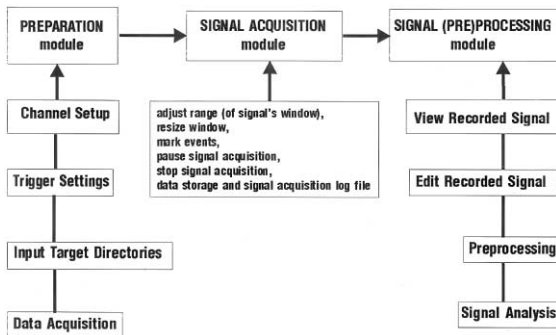


Fig. 1. A block diagram of the application lists the options that can be accessed in each program module. Preparation module. (1) Channel set-up, set number of channels, unit and channel name, external gain, input range, offset. (2) Trigger settings, set upper, middle or lower range, record before and after trigger, set pulse tolerance. (3) Input target directories, select local or network directories, allocate storage space. (4) Data acquisition, set output file size, set precision of stored data (8, 16, 32 or 64 bit), set continuous or scheduled recording. Signal (Pre)processing module. (1) View signal, play forward or reverse, fast forward or reverse, go to start or end of file, select next or previous file, overview and overlap view, adjust range of signal's window, resize window, print window or entire signal, select, zoom in/out, file info, convert between ASCII and Matlab file format. (2) Edit signal, cut, delete, paste, paste in new file. (3) Pre-processing, infinite impulse response (IIR) and finite impulse response (FIR) filters, running average, averaging signals. (4) Signal analysis, time to peak, 0.5 or 0.75 peak amplitude; time of peak to fall to 0.75 or 0.25 amplitude; time between peaks; Fourier transform and power spectrum.

dedicated to long term data acquisition, storage and display were written in Matlab programming language. A block diagram of our application is presented in Fig. 1. The program consists of three integrated modules, (a) a module that defines the protocol for signal acquisition, (b) commands that can be accessed during data acquisition and (c) software routines that enable off-line viewing, pre-processing and processing of recorded data. Most of the software commands require only the basic Matlab 5.3 program for processing and displaying data files. A list of required software for each module is presented in Table 1.

The software was tested on IBM compatible PCs, with a single Pentium processor and a Windows 95, 98 or NT 4.0 operating system. The minimum system requirements are Pentium II 200 MHz processor, 32 MB RAM, and 4 MB (excluding Matlab) of available disk space. These specifications may vary depending on acquisition parameters (sampling frequency and number of channels acquired simultaneously).

2.1. Data acquisition

The software was designed to adapt to a computer's processing power, available random access memory (RAM) and hard disk storage space. Therefore, in addition to selecting the number of data acquisition channels and sampling rate per channel, the user can also allocate the amount of RAM used by the software, set data file size (Fig. 2) and select target directories on local or network hard disks (Fig. 3).

The amount of computer memory used by the software is related to the resolution of stored data (i.e. 8, 16, 32 or 64 bit) and file size (Fig. 2b). This is because Matlab uses a double precision (64 bit) representation of data. For example, if a file size is 4 MB and resolution of stored data 16 bit then the software needs 16 MB of RAM. The size of required RAM increases to 64 MB for a 4 MB file with 8-bit data resolution. The file size option enables the breakdown of large signal records into more manageable blocks of data. These data blocks (files) have a common root file name and a unique sequence number (order of storage on a hard disk). An example is given in Fig. 4. Such

Table 1
The necessary software for individual modules^a

Module	Commands	Required software		
		Basic Matlab 5.3	Matlab signal processing Toolbox	Real time Toolbox 3.0
Preparation module	Channel setup	●		
	Trigger settings	●		
	Input target directories	●		
	Data acquisition	●		
Signal acquisition module	Signal acquisition module commands	●		●
Signal (Pre)processing module	View recorded signal	●		
	Edit recorded signal	●		
	Preprocessing commands	●	● (See table legend)	
	Signal analysis commands	●		

^a Most modules need only the basic Matlab 5.3 software. Exceptions are the signal acquisition module and IIR and FIR filter commands. The signal acquisition module also needs the Real Time Toolbox 3.0 software; the IIR filter and FIR filter commands also need the Matlab Signal Processing Toolbox. Channel set-up, trigger settings, target directories and data acquisition parameters can be defined and stored in a file for later use. Symbols — filled circles indicate the required software for specific commands.

Channel Settings **Trigger Settings** **Schedule Settings** **Directory Settings**

New File Name:

Description:

Sampling Rate: Hz Displayed Interval (Max. 25166): s

Continuous Sampling:

Sampling Time: s

Input Channels: 1 2 3 4 5 6 7 8 EVENTS

9 10 11 12 13 14 15 16

a

b

Available RAM: MB Saved data resolution:

Max. output file size: MB (RAM allocated for signal: 16 MB)

Fig. 2. Settings for data acquisition (a) and allocation of computer RAM and file size of recorded data (b). The user inputs the amount of available computer RAM (available RAM), the desired data file size (maximum output file size) and data resolution (saved data resolution). The software calculates the necessary RAM for the selected file size and data resolution (RAM allocated for signal), compares it with the available RAM and will not continue with the execution of data acquisition, if the RAM size allocated for signal is more then the available RAM.

data file management enables data acquisition of virtually unlimited duration.

Data can be stored on several local or network hard disks (Figs. 3 and 4). If for some reason (e.g. a breakdown of computer network, no available space on a selected hard disk), it is not possible to store data on a defined drive (i.e. persisting error from file system), the software moves stepwise down the list of selected drives until it locates a hard disk that will accept recorded data. The heart of this loop-fail-safe option is a temporary data buffer, with a default value capable of storing 90 s of recorded data, the size of which can be adjusted by the user. As a last resort, if data storage fails on all listed drives, files are stored on the program's (default) directory. During data acquisition, a log file is generated that records storage of data on hard disks.

The software's recommended, maximum sampling frequency is 41 kilo samples per second (KSPS) for a single channel and 3 KSPS for 16 channels. The meaning of a recommended, maximum sampling frequency is explained in Figs. 5 and 6.

2.2. Continuous and scheduled recording

The software enables the user to make a continuous or scheduled recording. A recording sched-

ule (Fig. 7) and a software trigger that does not require any dedicated hardware modules on an acquisition board (Fig. 8) enable the program to perform intermittent recordings of signals. Software trigger settings are grouped in three windows — level settings define the trigger voltage or voltages, record settings determine the recording time before or after a trigger event and pulse tolerance. Pulse tolerance, an alternative to the after trigger event option, extends the recording time interval for a specific trigger when the signal voltage stops to activate it for a short time interval (e.g. 0.01–1 s). This is especially important when recording irregular events, common in biological systems.

2.3. Viewing stored data

Our software application stores and displays biological signals as channels (one-dimensional arrays of signal values). All of the relevant signal information (e.g. sampling rate, channel gain and units) is stored in a file header. The software can display simultaneously, on the computer monitor, up to 16 channels of stored data (e.g. 16 files with one data channel or one file with 16 channels of data). Therefore, several data channels from the same or different files can be displayed, analysed or compared.

A sample record of a rat's four-channel EEG trace is presented in Fig. 9. The experiment was performed on an adult, male Wistar rats weighing 225 g. The animal was anaesthetised with a single intraperitoneal injection of ketaminhydrochlorid (45 mg/kg b.w.) and xylazinhydrochlorid (7 mg/kg b.w.) and placed on a thermostated operating table with an stereotaxic apparatus. The rat's skull, over the frontal and parietal brain, was exposed and four pairs of holes were drilled through the animal's skull into which recording electrodes were inserted for differential recording of EEG signals over the left and right parietal and frontal cortex. A ninth, reference electrode was attached to the nasal bone. The recording electrodes were fabricated out of platinum wire (diameter 0.2 mm) and secured to the animal's skull by Harvard Cement. (Richter and Hoffman Harvard Dental-GmbH, Germany). The electrodes

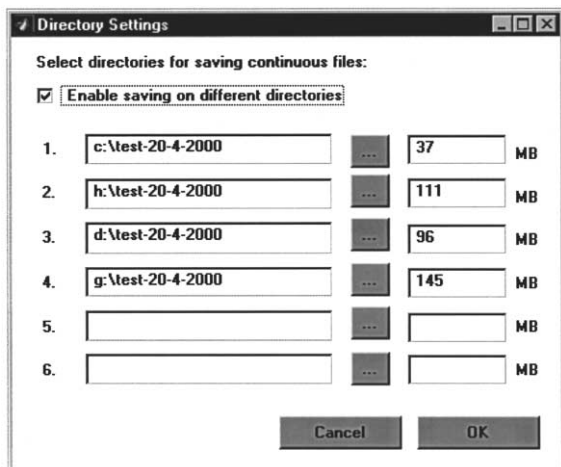


Fig. 3. Directory settings. Data can be stored on local (c:\ and d:\) or network hard disk drives (h:\ and g:\) depending on available hard disk space (see also Fig. 4).

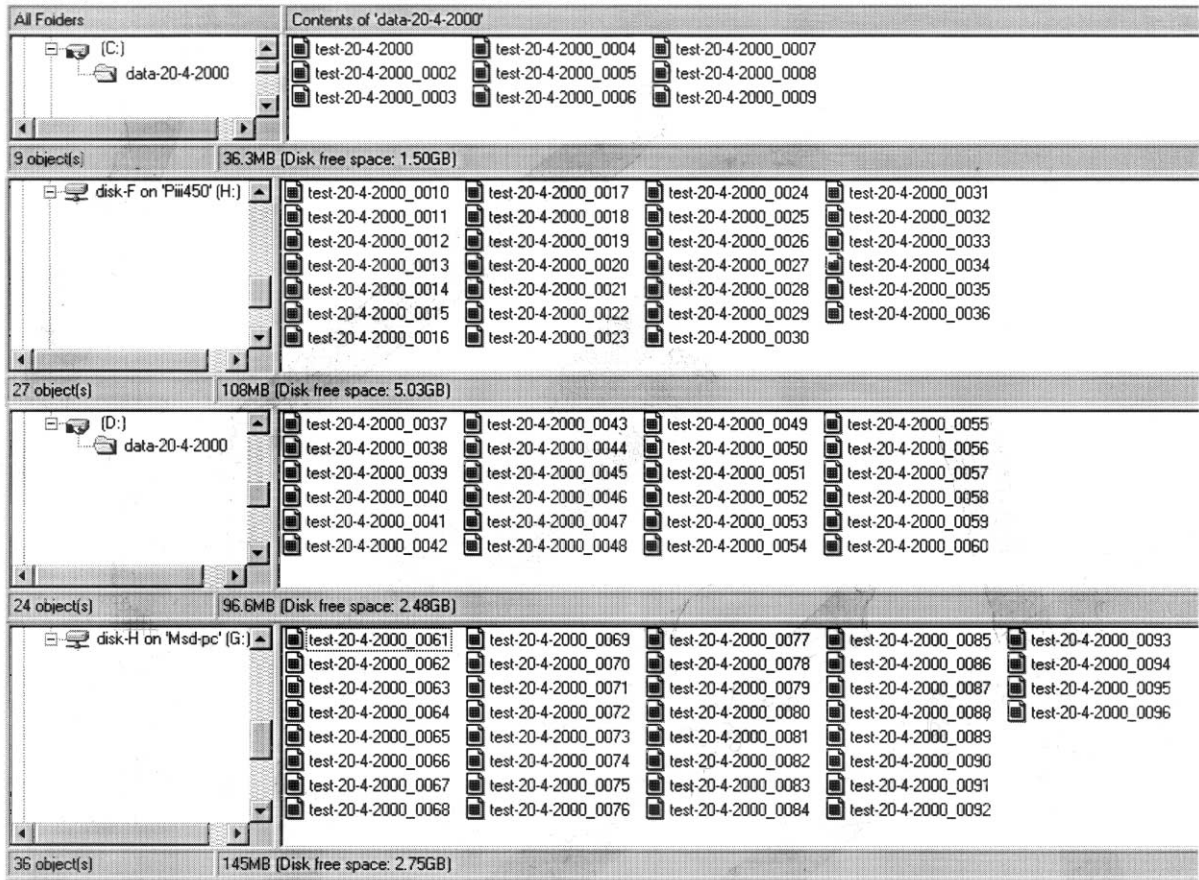


Fig. 4. Storage of recorded data on local and network hard disk drives. Each data file name has a common root (e.g. test-20-4-2000) and a unique sequence number (e.g. _0092). For this example, the amount of stored data on each drive was defined in Fig. 3.

were connected to a custom made four-channel amplifier (amplification $\times 4000$ per channel) and the signals were feed into a computer via a termination panel of an AT-MIO-16E-1 acquisition board (National Instruments, USA). EEG signals were recorded continuously, with a 500-Hz sampling rate for four channels, until the rat regained consciousness. The resolution of stored data was 16 bit, file size was set to 4 MB. Raw EEG data were filtered with a software, Butterworth, low pass filter (passband frequency 35 Hz, stopband frequency 40 Hz). A power spectrum (0–30 Hz) was calculated for selected sections of EEG records with the fast Fourier transform algorithm. The experiment was part of an ongoing study that aims to link specific

EEG patterns with the depth of general anaesthesia.

The results of editing, pre-processing or processing of a data channel are also stored as a channel, either overwriting the source channel or as an additional, new channel. To increase the resolution of an observed data channel on a computer monitor, individual channels can be switched off. While viewing data channels, the speed of the play and fast play option can be adjusted depending on the complexity of a specific signal. If an experiment is stored on several files, the software displays them seamlessly and in their proper sequence by the load automatically next/previous file option. A signal channel can display three types of events, trigger events

— mark the start and end time of a trigger activating voltage, schedule events — mark the start and end time of a scheduled recording and user activated events — events marked by the user with any symbol on the keyboard. Also, user events can be added to signal channels offline, while viewing a signal record.

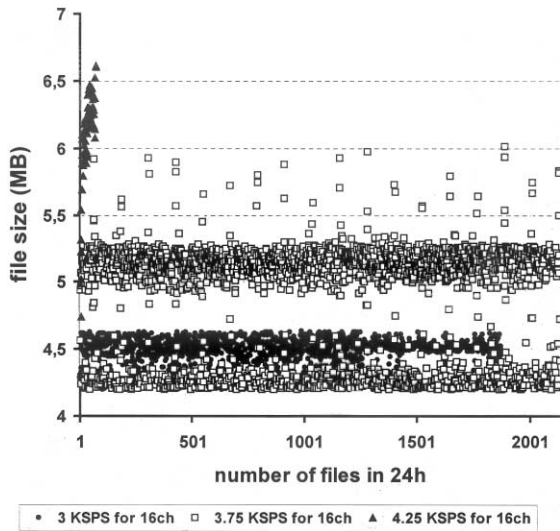


Fig. 5. The recommended, maximum sampling rate for 1 (data not shown) and 16 channels was determined in 24 h sessions of continuous recording. Data acquisition was performed on an IBM compatible PC with 256 MB RAM, a 333 MHz Pentium II processor and a 6 GB IBM hard disk. During data acquisition the software transmitted data files over a local network and stored them on a second IBM PC with 128 MB RAM, a 550 Mz Pentium III processor and a 20 GB IBM hard disk. The following software was installed on both computers, Windows NT 4.0 with service pack 6, Norton Utilities for Windows NT 4.0 (Symantec, USA), Virus Scan NT 4 (Network Associates Inc., UK) set to scan all incoming and outgoing files and a fire wall software set to block all local and internet servers (Zone Alarm 2, Zone Labs Inc., USA). File size was 4 MB and resolution of stored data was 16 bit for all sampling rate tests. The graph shows the variations in file size of three sampling rates for 16 channels, 3 KSPS per channel (3KSPS for 16ch), 3.75 KSPS per channel (3.75 KSPS for 16ch) and 4.25 KSPS per channel (4.25 KSPS for 16ch) during a 24 h recording session. Increasing the sampling rate over 3 KSPS per channel had two consequences. First, increased variations of file size (3.75 KSPS for 16ch) and then a premature termination of data acquisition (4.25 KSPS for 16ch). Termination of data acquisition was a preventive measure initiated by the software when it predicted that data overflow of the temporary buffer was imminent.

3. Discussion

The software application was written specifically to facilitate acquisition, storage and processing of large amounts of biological signals. Application-specific, signal-processing routines (e.g. spectral analysis) can be selected from Matlab Signal Processing Toolboxes. The software program stores data in Matlab *.mat files and reads ASCII text files that have channels stored as columns of numbers. This enables the user to analyse data with Matlab libraries or custom made routines. Some ASCII text files include data acquisition information (e.g. acquisition rate, channel offset and gain) in a file header that cannot be read by our software. To solve this problem, a user defines the number of lines to be ignored from the beginning of a file and inputs the acquisition rate. In addition, a Matlab data file can be exported to an ASCII text file for analysis with other software tools.

All of the program routines are hardware-independent — settings for different acquisition boards are available directly from the program menu. A/D-converter board drivers are available as part of the RTT 3.0 module. The number of input channels that can be recorded by our software was set to 16 and can be extended to match the number supported by an A/D-converter.

The software application was written in Matlab because it combines an adequate computing environment for high performance numeric computation with a large library of routines for visual display of data. Preferably it should run under Windows NT 4.0 to assure a stable network communication between computers that share recorded data. The program code was optimised for speed by using Matlab matrix functions whenever possible. This approach has been found useful in the design of data acquisition software with Matlab [5].

The loop-fail-safe option reduces the chances of loss of data during acquisition (e.g. network failure, no available free space on a hard disk). The program stores recorded data in a temporary data buffer until it locates a drive that will accept recorded files. This recording process is fully documented. The software creates a log file with a list

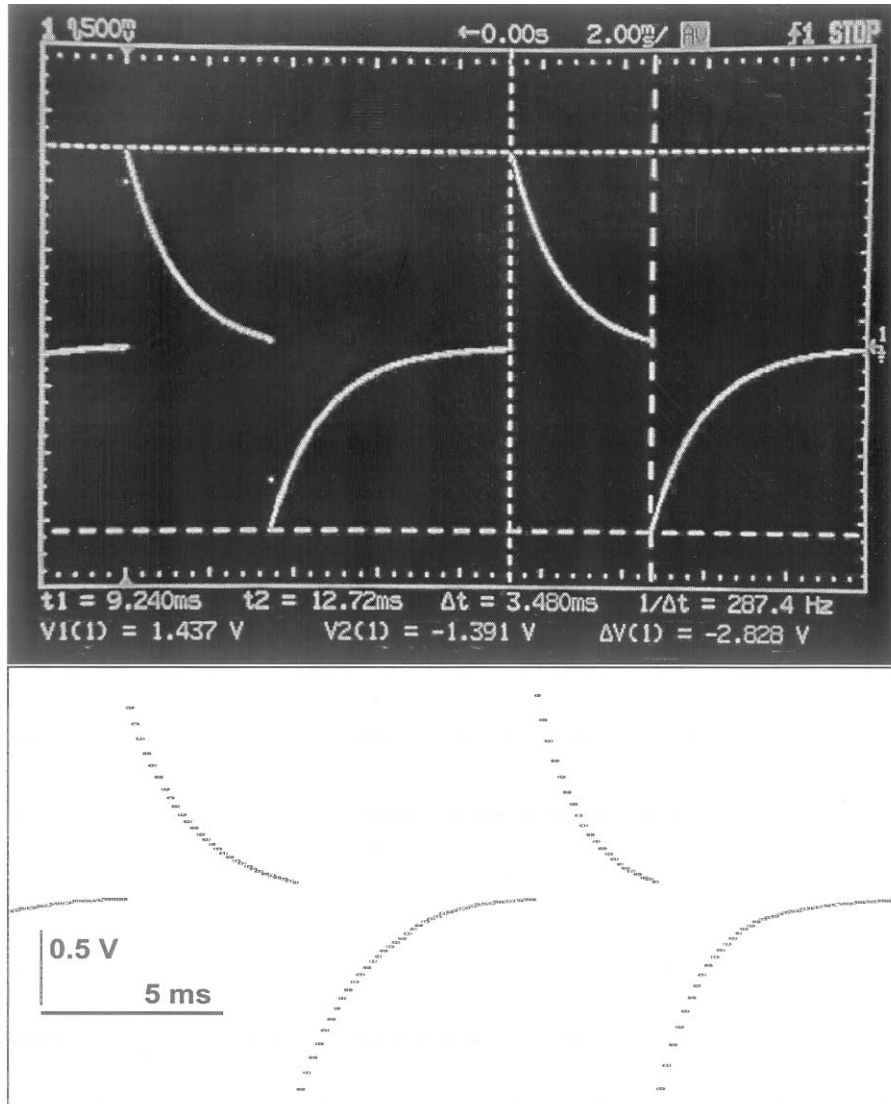


Fig. 6. The fidelity of signal acquisition was tested by recording repetitive, biphasic voltage pulses (287.4 Hz, pulse width 3.48 ms, peak to peak voltage 2.828 V) at different sampling rates for 1 (data not shown) and 16 channels. These records were compared with simultaneous measurements displayed on a 100 MHz digital oscilloscope (HP 54601A, Hewlett-Packard, USA). Fidelity of data acquisition was not compromised even at extreme sampling frequencies when the software terminated data acquisition to prevent data overflow of the temporary data buffer. An oscilloscope, screen shot of two, successive test voltage pulses is displayed in the top half of Fig. 6. The bottom half of Fig. 6 displays two, successive biphasic voltage pulses recorded by the software when the sampling rate was set to 6 KSPS per channel for 16 channels. The resolution of stored data was 16 bit.

of file paths of data files created during signal acquisition. During continuous data acquisition the software moves along the list of selected drives (Fig. 3) in an endless loop enabling the transfer of data to other locations and thus creating addi-

tional free space on target drives. The software will exceed the size of allocated hard disk space by less than 1 MB, therefore it tends to store a number of files that is equal or less than the specified available hard disk space (Figs. 3 and 4).

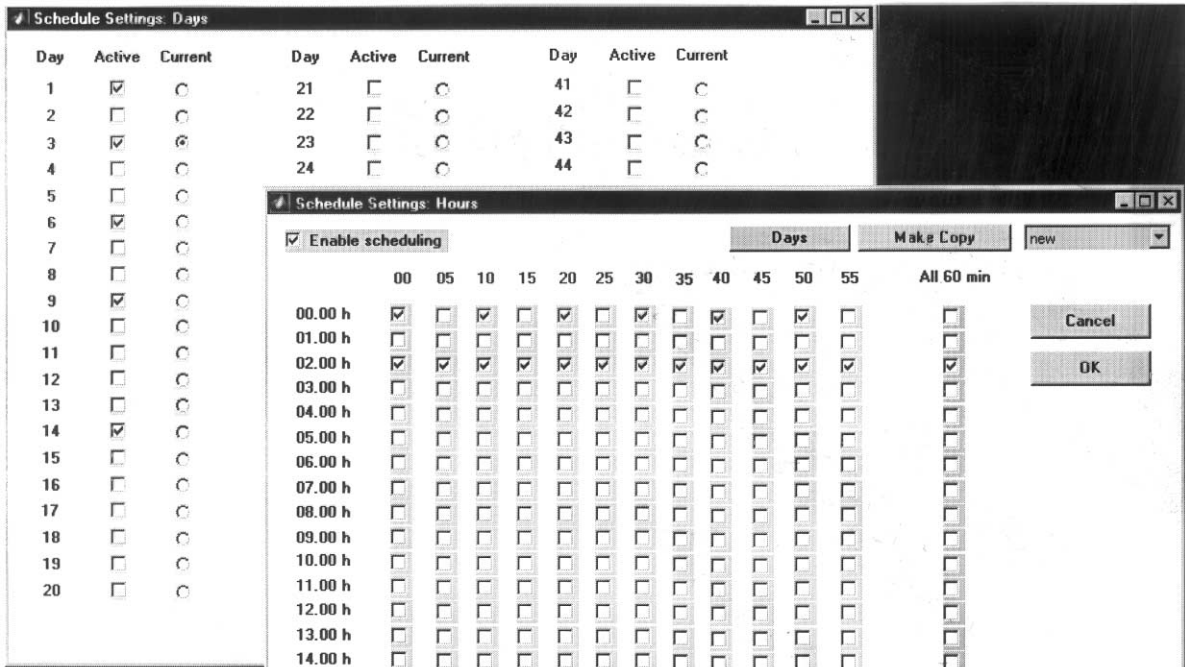


Fig. 7. Schedule settings. The user can construct a timetable (by ticking the appropriate box) that will suit his signal acquisition protocol. Recording intervals range from 5 min to 60 days and allow for a wide range of combinations. For example, the time schedule in Fig. 7 was set to record on days 1, 3, 6, 9 and 14 (schedule settings: days). The bottom, right half of the figure displays the time schedule for day 3 (schedule settings: hours).

For example, in Fig. 3 the amount of stored data was set to 96 MB for drive d, and the actual size of stored data on this drive was 96.6 MB (Fig. 4). Since a log file is generated during data acquisition, the user can spot any problems with data storage on local or network hard disks.

The software's modular structure facilitates integration of signal processing and analysis commands. Therefore, it can be easily adapted to incorporate new Matlab or custom written code. We estimate that the maximum sampling rate of our software can be increased by at least a factor of two. This will be done by optimising code for the temporary data buffer administration and display of recorded signals.

Our application adapts to different computer hardware configurations by varying the size of data files, data resolution and by distributing file storage on local and network hard disks thus enabling optimal signal acquisition and analysis. The software has been used in animal experiments

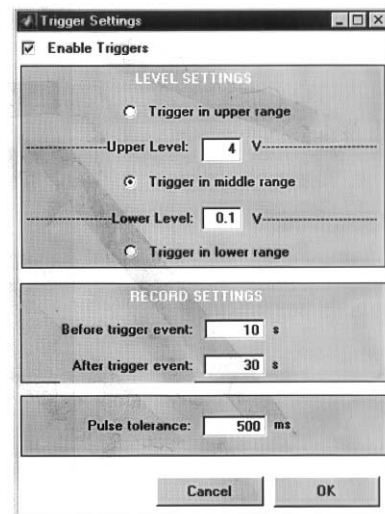


Fig. 8. Trigger settings.

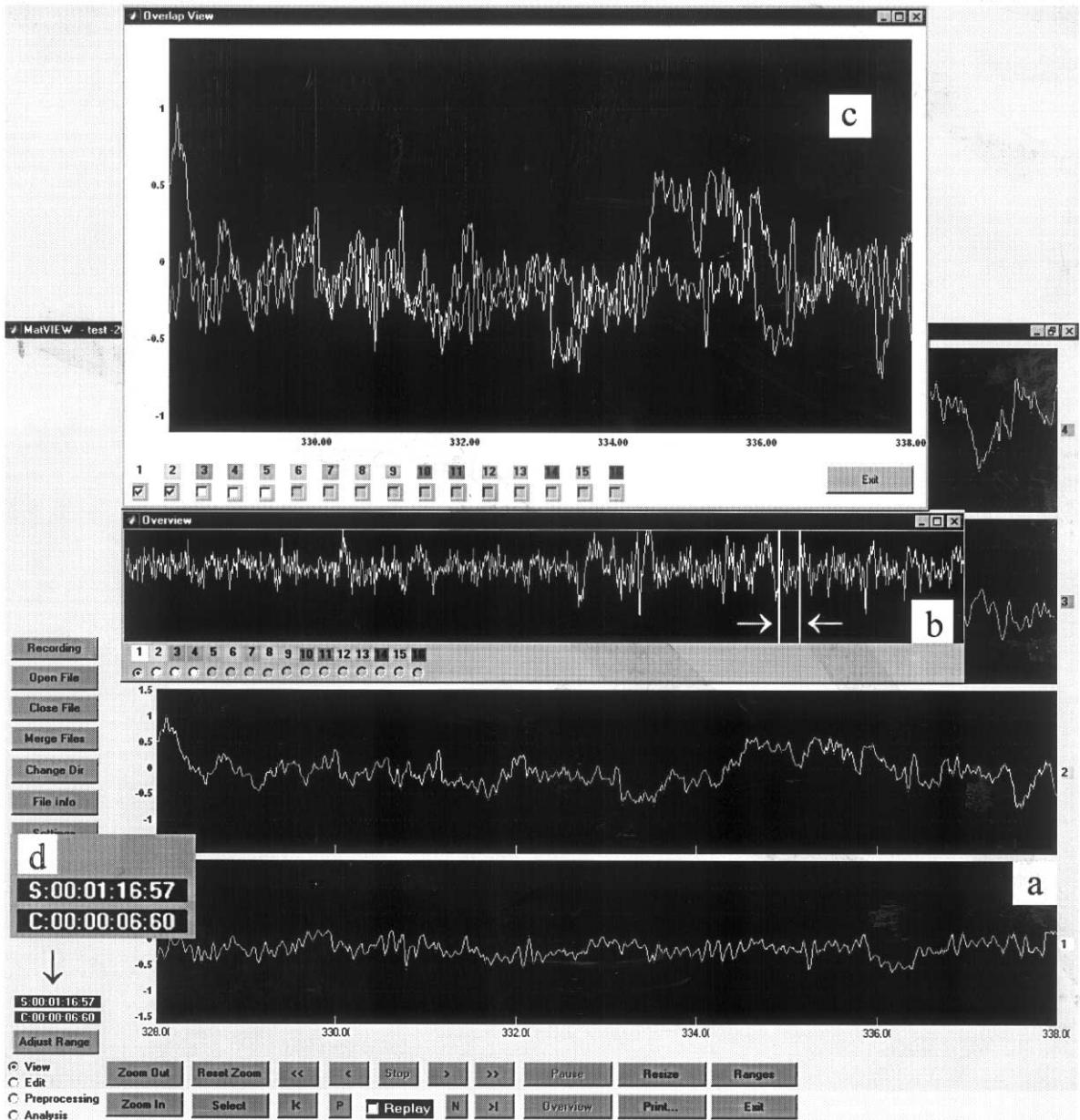


Fig. 9. Sample record of a rat's four-channel EEG trace. Recordings were made over the left and right frontal cortex (channels 1 and 2) and left and right parietal cortex (channels 3 and 4). Units on x -axis are seconds, units on y -axis are μV ($0.5 = 125 \mu\text{V}$). The position of a displayed segment on channel 1 (a) relative to the whole recorded file can be seen in inset b (the segment between arrows). For comparison, signals on individual channels can be superimposed on the same display (e.g. inset c displays channels 1 and 2). If a record is stored as a sequence of files, each file displays the time elapsed from the beginning of the experiment and its length (e.g. in inset d the time from the start of data acquisition is 1 h 16 min 57 s, the length of the current file is 6 min 60 s).

for acquisition or analysis of EEG, skin blood flow [6], and electroneurograms [7].

Acknowledgements

This work was funded by the Ministry of Science and Technology, Republic of Slovenia. The authors thank Dr Janez Rozman for the preparation of EEG recording electrodes.

References

- [1] E.K. Alford, R.R. Roy, J.A. Hodgson, V.R. Edgerton, Electromyography of rat soleus, medial gastrocnemius, and tibialis anterior during hind limb suspension, *Exp. Neurol.* 96 (1987) 635–649.
- [2] C. Blewett, G.C. Elder, Quantitative EMG analysis in soleus and plantaris during hindlimb suspension and recovery, *J. Appl. Physiol.* 74 (1993) 2057–2066.
- [3] Y. Mori, D. Kondziolka, J. Balzer, W. Fellows, J.C. Flickinger, L.D. Lunsford, K.R. Thulborn, Effects of stereotactic radiosurgery on an animal model of hippocampal epilepsy, *Neurosurgery* 46 (2000) 157–165.
- [4] D.L. Sherman, A.M. Brambrink, R.N. Ichord, V.K. Dasika, R.C. Koehler, R.J. Traystman, D.F. Hanley, N.V. Thakor, Quantitative EEG during early recovery from hypoxic-ischemic injury in immature piglets: burst occurrence and duration, *Clin. Electroencephalogr.* 30 (1999) 175–183.
- [5] R. Maestri, G.D. Pinna, POLYAN: a computer program for polyparametric analysis of cardio-respiratory variability signals, *Comput. Methods Programs Biomed.* 56 (1998) 37–48.
- [6] F. Bajrovic, M. Čenčur, M. Hožič, S. Ribarič, A. Stefanovska, The contribution of lumbar sympathetic neurones activity to rat's skin blood flow oscillations, *Eur. J. Physiol.* 439 (Suppl.) (2000) R158–R160.
- [7] J. Rozman, B. Zorko, A. Seliškar, M. Bunc, Selective recording of neuroelectric activity from the peripheral nerve, *Eur. J. Physiol.* 440 (Suppl.) (2000) R157–R159.