

## Visualização e Armazenamento de Sinais Biológicos

Filipe Trancho, Ricardo Conceição, Osvaldo Rocha Pacheco, Ana Maria Tomé

**Resumo** - Neste artigo descreve-se uma aplicação para visualização de sinais biológicos, armazenados digitalmente, para ambientes de Neurofisiologia. Os requisitos da aplicação tiveram por base as características apresentadas no registo poligráfico. A aplicação foi desenvolvida utilizando técnicas de programação por objectos, tendo como suporte a livreria *ObjectWindows* e o *Resource Workshop* da *Borland*.

**Abstract**- In this paper, we describe an application to visualize biological events, particularly suited to neurophysiology environments. The application requirements are based on the polygraphic record. The application was developed using Object-Oriented techniques supported by *Borland C++* compiler and its *ObjectWindows* library and *Resource Workshop*.

### I. INTRODUÇÃO \*

O Electroencefalograma, habitualmente conhecido por EEG, consiste no registo dos potenciais eléctricos que resultam da actividade electrofisiológica cerebral, ao longo do tempo, obtido por intermédio de eléctrodos estrategicamente colocados no escalpe humano. Este é um importante meio de investigação e de diagnóstico de patologias cerebrais.

Definindo uma onda como uma qualquer modificação de duração limitada do potencial eléctrico registado entre dois eléctrodos, qualquer que seja a sua forma, chama-se actividade a uma sequência de ondas com determinadas características consistentes. Estas actividades, situam-se por sua vez em bandas de frequência bem definidas, designadas por banda delta, teta, alfa e beta.

Os eléctrodos de recolha de sinal poderão ser de vários tipos (eléctrodos de agulha, eléctrodos de disco com várias dimensões, etc.), cobrindo a superfície do escalpe com intervalos igualmente espaçados, seguindo modelos pré-estabelecidos, de que o *standard* 10-20 [1] é um bom exemplo.

A máquina de EEG recebe os sinais eléctricos de entrada directamente dos eléctrodos colocados no escalpe. Posteriormente, o sinal passa por um andar de amplificação, constituído por um amplificador diferencial, que se destina a rejeitar o sinal de modo comum. São de seguida aplicados filtros de alta e baixa frequência que se

destinam a reduzir variações de potencial muito rápidas e muito lentas e para realçar actividades importantes do ponto de vista clínico. Pode ainda ser introduzido um filtro rejeita-banda que se destina a eliminar os 50 Hz, introduzidos pela rede de alimentação. Os potenciais eléctricos amplificados são usados para provocar deflecções numa agulha, cujos movimentos ficam registados em papel contínuo que se desloca a uma velocidade constante, que poderá ser de 10, 15, 20 ou 30 mm/s. A este registo dá-se o nome de registo poligráfico.

O registo poligráfico, de grande importância clínica terá que satisfazer um certo número de requisitos técnicos para disponibilizar todas as suas potencialidades. Os requisitos para estes registos diferem da análise pretendida (registos de noites de sono, observação de pacientes com suspeitas de lesões cerebrais, etc.), e podem influenciar a escolha da velocidade do papel, o ajuste do ganho dos amplificadores da máquina de EEG, o número de canais, etc.

Algumas das desvantagens desta forma convencional de armazenamento de sinal, são:

- para o registo de algumas horas de EEG de sono, são necessários várias centenas de metros de papel, sendo por isso difícil o seu posterior armazenamento, nos processos clínicos dos pacientes nos hospitais;
- impossibilidade de realizar uma nova visualização, com outra resolução do sinal, já que o registo original é definitivo;
- incapacidade de nesta forma de registo efectuar qualquer tipo de processamento sobre o sinal.

Com o desenvolvimento de novas tecnologias surgiram alternativas aos métodos convencionais de registo do EEG, assumindo especial relevância o uso de computadores e de registos magnéticos.

O SIGIF (*SIGnal Interchange Format*) [2] é um formato de armazenamento digital, desenvolvido pelo INESC/Aveiro. O ficheiro que suporta este formato, tem a informação organizada em duas partes distintas: o cabeçalho e os dados. O cabeçalho, que se situa sempre no início do ficheiro, contém informação geral necessária sobre a organização dos dados no mesmo ficheiro, incluindo a data de aquisição do sinal e da última alteração do ficheiro, número de canais, frequência de amostragem, calibração dos sinais, entre outros. A secção destinada aos dados, que se situa a seguir ao cabeçalho, é constituído por um ou vários segmentos, sendo cada um, de uma forma genérica, constituído pelo tempo referente ao início do mesmo e pelos dados propriamente ditos.

\* Trabalho realizado no âmbito da disciplina de projecto.

## II. OBJECTIVOS

Os requisitos fundamentais para elaborar uma interface que permita armazenar, visualizar e processar os sinais em formato digital são:

- *Fundo a imitar o papel* - Permite aos clínicos uma melhor identificação dos fenómenos biológicos presentes no sinal, mantendo-se assim uma certa coerência com o aspecto visual observado no papel. Pode-se assim, garantir ao clínico uma certa continuidade de hábitos, em oposição à ruptura que poderia ser suficiente para uma maior adversidade ao uso do sistema.
- *Amplitude em valor absoluto* - Na linha de continuidade do ponto anterior, é importante que na visualização do sinal, cada canal mantenha a mesma escala que teria no papel. Caso esta condição não se verifique, existe a possibilidade de ocorrerem erros na identificação de eventos e na análise do sinal;
- *Combinação de canais* - Permite ao clínico a visualização do número de canais que pretende, pela ordem que deseja. Isto é relevante quando se dispõe de um número elevado de canais e se pretende apenas visualizar alguns deles;
- *Variação dos bits de resolução dos canais* - Variação do número de *pixels* para cada sinal que está a ser visualizado, através da variação dos *bits* de resolução de cada canal, por um factor escolhido;
- *Alteração do tempo por página* - Possibilidade de variar a duração do segmento do sinal apresentado por página. Isto permite incorporar maior quantidade de informação numa página, possibilitando assim uma análise de contexto para o segmento em estudo;
- *Escolha de segmentos de interesse* - Possibilidade de selecção de segmentos com interesse clínico, de forma a tornar possível uma análise mais detalhada dos mesmos;
- *Informação "on line"* - Aceder de uma forma fácil às informações relativas ao canal, amplitude e tempo respectivo;
- *Gravação de ficheiros com o formato MATLAB* - Selecção e armazenamento de sequências de sinal, para facilitar o desenvolvimento de novos algoritmos de processamento de sinal;
- *Gravação de canais seleccionados, com formato SIGIF* - Permite a gravação em ficheiro, de determinados canais, com interesse do ponto de vista do clínico, ignorando os restantes;
- *Multi-segmentos* - Gravação sequencial de segmentos com interesse, de forma a criar ficheiros de dimensões bastante mais reduzidas, que contudo contenham toda a informação relevante para a análise. Para este método poder ser totalmente eficaz, tem que se garantir a permanência das referências temporais dos vários segmentos gravados, relativamente ao sinal original.

## III. MÉTODOS

### *O ambiente Windows*

O *Windows* é um ambiente multi-tarefa com janelas baseadas em gráficos que permite que programas escritos especialmente para o *Windows*, tenham uma aparência e uma estrutura de comando consistente.

O principal meio usado para trocar informações no ambiente multi-tarefa é o sistema de mensagens do *Windows*. Do ponto de vista do programa, uma mensagem pode ser vista como uma notificação de que algum evento ocorreu, podendo esta ser ou não associada a uma acção específica.

### *A livreria OWL (ObjectWindows Library)*

A *ObjectWindows Library* contém um conjunto de classes criadas pela *Borland* [3, 4, 5] para o desenvolvimento de aplicativos para plataformas *Windows* (*Windows 3.1*, *Win32S* e *Windows NT*).

A utilização da *OWL* simplifica significativamente a criação e o desenvolvimento de vários elementos da aplicação. Estes elementos incluem a criação das próprias janelas, caixas de dialogo, vários tipos de controlos (*status bars*, *scroll bars*, botões, etc.), menus, *icons*, *bitmaps*, etc.

### *Menus e Caixas de Dialogo*

Os menus e as caixas de dialogo foram construídos utilizando uma das aplicações do compilador *Borland C++ 4.5*, denominada *Resource Workshop*, em alternativa ao método descrito no ponto anterior. Esta aplicação permite-nos criar vários tipos de recursos de uma forma rápida e perceptiva. Com esta aplicação, associa-se um identificador a cada um dos recursos criados (*radio buttons*, *check boxes*, *push buttons*, etc.), sendo depois possível a sua evocação no programa através destes identificadores.

### *Estrutura do Programa*

A aplicação foi desenvolvida com base no compilador *Borland C++ 4.5*. As razões da escolha deste compilador baseiam-se no aproveitamento das vantagens e características da programação orientada por objectos *C++*.

Este modo de programação permite a definição de classes, sua derivação, criação de funções virtuais e um maior encapsulamento de dados. Com a utilização de classes é possível criar um novo tipo de dados, definido pelo próprio programador de *C++*, tendo as vantagens de uma estrutura e a possibilidade de limitar o acesso aos dados específicos das funções que também são membros da classe. Tendo em consideração as características referidas anteriormente, construímos a seguinte hierarquia de classes (Fig. 1).

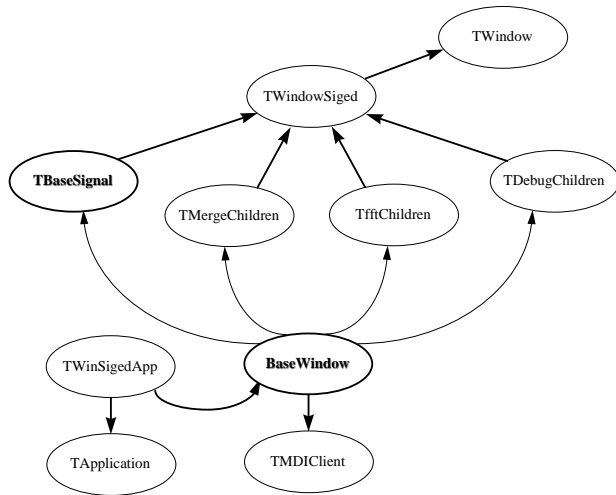


Fig. 1 - Hierarquia das classes principais.

Para se criar uma janela para a aplicação (utilizando a OWL), é necessário inicialmente efectuar os seguintes procedimentos:

- Definir um objecto que possa aceder à parte pública da classe *TApplication* da OWL;
- Evocar a função-membro da classe *TApplication*. *Run()*;

Na nossa aplicação, a classe que acede à parte pública da classe *TApplication* é a classe *TWinSingedApp*, que é definida da forma:

```
class TWinSingedApp : public TApplication
{ ... }
```

A nossa classe principal *BaseWindow* é uma classe derivada da classe *TMDIClient* da OWL, sendo esta definida da forma:

```
class BaseWindow : public TMDIClient
{ ... }
```

A classe *TMDIClient* pertence ao MDI (*Multiple Document Interface*), que tal como o nome indica, permite criar um número distinto de *child windows* (janelas filhas), possibilitando assim a associação de cada uma delas a diferentes tarefas. Pode-se definir o MDI como tendo uma janela principal, à qual se pode associar diferentes tipos de objectos, tais como, *status bars*, *control bars*, etc., mas sobre a qual não se associa nenhuma tarefa em especial e as janelas filhas às quais se associa uma tarefa específica. No caso do nosso programa, associámos uma *status bar* à janela principal, da seguinte forma:

```
void TWinSingedApp::InitMainWindow()
{ Frame = new TDecoratedMDIFrame ("WINDOWS SIGED",
"Menu_2", *new BaseWindow);
sb = new TStatusBar (0, TGadget::Recessed);
Frame->Insert (*sb, TDecoratedFrame::Bottom); }
```

Em relação às *child windows*, tivemos necessidade de criar quatro janelas com funções distintas, às quais associámos as classes *TBaseSignal*, *TfftChildren*, *TDebugChildren* e *TMergeChildren*, da seguinte forma:

- `TMDIChild* child = new TMDIChild (*this, "Base Window", new TBaseSignal);`
- `TMDIChild* childmerge = new TMDIChild (*this, "Time-Magnitude", new TMergeChildren);`
- `TMDIChild* childfft = new TMDIChild (*this, "FFT", new TfftChildren);`
- `TMDIChild* childdebug = new TMDIChild (*this, "List Segments", new TDebugChildren);`

A. Classe *TBaseSignal*

A classe *TBaseSignal* está associada à leitura, tratamento, visualização e gravação dos sinais pretendidos, que estejam presentes em ficheiro no disco. A janela gráfica associada a esta classe é apresentada na Figura 2.

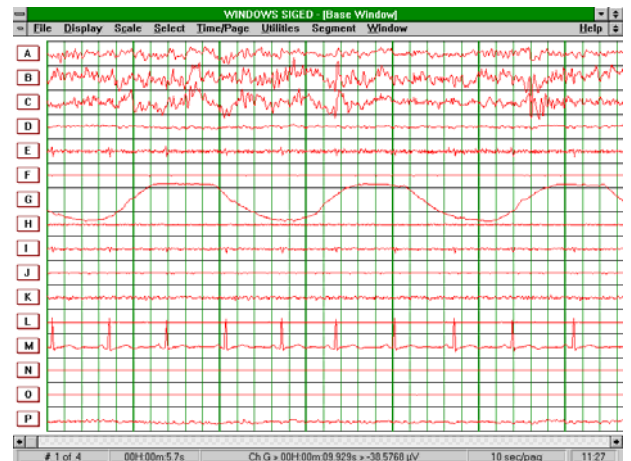


Fig. 2 - Janela gráfica associada à classe *TBaseSignal*.

Os principais objectivos inerentes a esta classe são:

- Leitura de dados e sinal gravados em ficheiro com o formato SIGIF. Através deste procedimento o cabeçalho do ficheiro é lido, sendo adquirida toda a informação que é vital ao funcionamento do programa, como por exemplo, o número de canais, valores de calibração dos diferentes canais e seus ganhos, frequência de amostragem, referências temporais, bem como parte ou a totalidade dos sinais contidos no ficheiro. Pode ser também necessário nesta fase, o programa percorrer a totalidade do ficheiro em busca de separadores de segmentos, já que devido às especificações do formato SIGIF, estes separadores podem-se encontrar em qualquer posição no ficheiro;
- Redimensionamento das variáveis consoante as características do sinal adquirido. Existem variáveis desta classe que necessitam de ser ajustadas sempre que se lê um novo ficheiro e se adquirem novos dados. Dentro destas variáveis encontram-se, por exemplo, as que são responsáveis pelo cálculo do *range* do sinal para a *scroll bar* - que terá que ser dimensionado conforme a duração do sinal no ficheiro;
- Tratamento do sinal adquirido, consoante os dados contidos no cabeçalho. O sinal recolhido na primeira fase vai ser processado de forma a poder ser utilizado por outras funções-membro da classe, principalmente pelas funções referentes à visualização do sinal;

- Visualização dos sinais e funções gráficas inerentes. Dentro destas funções englobam-se as que são responsáveis pela apresentação do sinal, das referencias temporais (grelha de fundo), mudança do número de canais e/ou da sua ordem, alteração do tempo por página, variação dos *bits* de resolução dos canais, etc.
- Selecção de segmentos para posterior processamento ou gravação;
- Leitura de multi-segmentos e activação das consequentes funcionalidades que permitam a selecção do segmento que se pretende visualizar;
- Possibilidade de efectuar saltos temporais no ficheiro de sinal, podendo estes ser absolutos ou relativos;
- Gravação de segmentos de sinal pré-seleccionados em ficheiros com formato SIGIF ou formato MATLAB, com possibilidades de escolher os canais pretendidos;
- Outras funcionalidades que tornam a interface gráfica mais interactiva, como por exemplo, a activação de uma régua auxiliar (sempre que se pretenda realizar medições na amplitude dos canais) e acesso a informação pontual sobre determinados parâmetros do sinal (nome do canal, referência temporal e amplitude).

Esta classe interage com outras classes, que suportam caixas de dialogo importantes na execução das tarefas descritas anteriormente. Esta interacção pode ser esquematicamente representada na Figura 3.

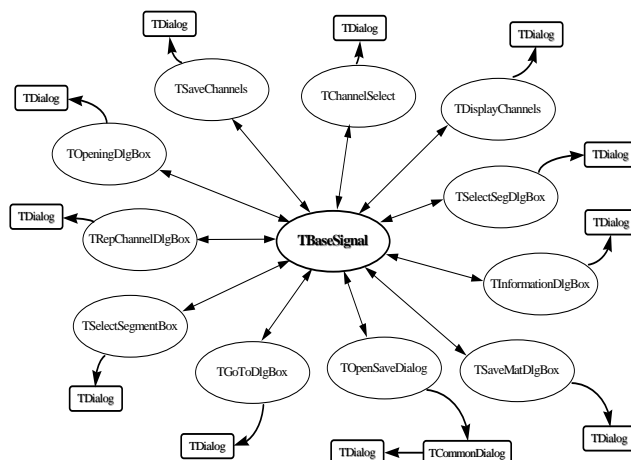


Figura 3 - Hierarquia das classes associada à classe TBaseSignal

**B. Classe TMergeChildren**

A classe *TMergeChildren*, serve de suporte à janela que permite a inspecção de um segmento de sinal previamente escolhido, do qual se pretende extrair informações mais detalhadas, como por exemplo, tempo inicial e final do segmento e amplitude máxima e mínima do sinal contido nesse mesmo segmento. Um exemplo da interacção desta classe, poderá ser observada na Figura 4.

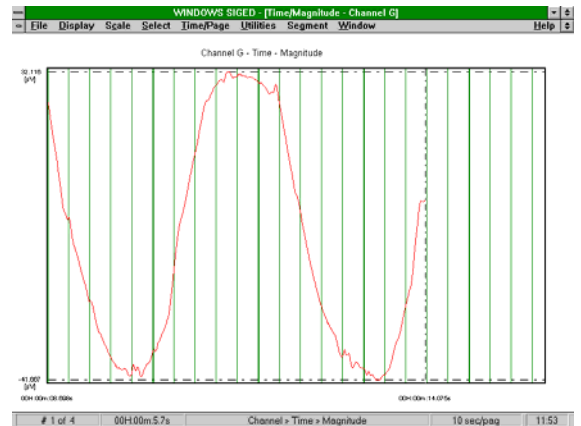


Figura 4 - Janela gráfica associada à classe TMergeChildren

**C. Classe TfftChildren**

A classe *TfftChildren* está relacionada com o cálculo e apresentação de resultados da aplicação da transformada de Fourier a um segmento de sinal seleccionado. Aqui são definidas as funções que realizam este processamento, ficando disponíveis dois tipos de gráficos, um representando a variação do espectro de potência do sinal com a frequência e o outro apresentando o histograma do espectro de potência para as bandas de interesse. Como exemplo do funcionamento desta classe e funções associadas, pode-se observar a Figura 5, representando o histograma do espectro de potência para as bandas Delta (1 e 2), Teta, Alfa, Sigma e Beta:

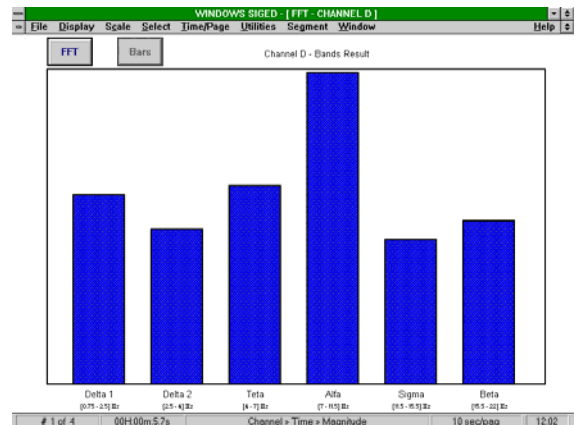


Figura 5 - Janela gráfica associada à classe TfftChildren

**D. Classe TDebugChildren**

A classe *TDebugChildren* contém todas as funções necessárias para a elaboração de uma listagem, com informação sobre o(s) segmento(s) previamente seleccionado(s). Esta informação engloba o tempo inicial e final dos segmentos e a sua dimensão em *Kbytes*.

Esta classe interage com a classe *TRejectSegDlgBox*, sendo esta última uma classe derivada da *TDialog*, conforme se mostra na Figura 6.

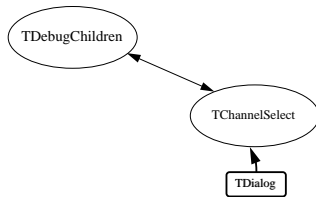


Figura 6 - Hierarquia das classes associada à classe *TDebugChildren*

A classe *TRejectSegDlgBox*, serve de suporte a uma caixa de diálogo, que permite seleccionar um segmento, já incluído na listagem anteriormente referida, que o utilizador pretenda rejeitar.

Pode-se observar a interacção destas duas classes, conforme descrevemos anteriormente, através da Figura 7.

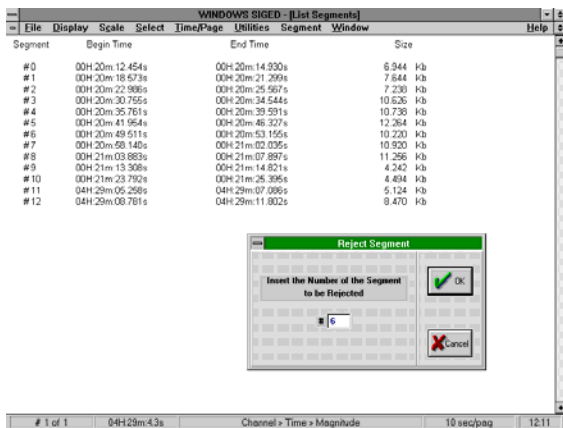


Figura 7 - Janela gráfica associada às classes *TDebugChildren* e *TRejectSegDlgBox*

IV. CONCLUSÕES

A aplicação desenvolvida disponibiliza um produto versátil e de fácil utilização, permitindo aos clínicos efectuar análises com um elevado índice de eficiência, devido à coerência com os hábitos de visualização adquiridos anteriormente (traçado poligráfico).

Este sistema, acompanhado de um manual de utilização, encontra-se presentemente a ser testado em rotina clínica no Hospital Geral de Sto. António. Até ao momento tem-se verificado, por parte dos clínicos da área de Neurofisiologia, uma boa aceitação e elevada motivação no seu uso.

Os desenvolvimentos futuros poderão passar pela inclusão de interfaces de comunicação bidireccional de dados, entre o sistema e aplicativos específicos de processamento de sinal e ainda pela introdução de algumas técnicas de compressão de sinal.

BIBLIOGRAFIA

[1] R. Spehlman, "EEG Primer", Elsevier - North Holland Biomedical Press Amsterdam.

[2] M. B. Cunha, J. P. Cunha, T. O. Silva, "Sigif: A Digital Signal Interchange Format for Biological Signals", Proc. Of 15<sup>th</sup> Annual Internacional Conference IEEE Engineering in Medicine and Biological Society, San Diego, USA, 1993.  
 [3] "Borland C++ 4.0 - User's Guide", Borland International, Inc.  
 [4] "Borland C++ 4.0 - Programmer's Guide", Borland International, Inc.  
 [5] "Borland C++ 4.0 - Reference Guide", Borland International, Inc.