

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

**Monitorização da Evolução Temporal da Resposta à  
Estimulação Somato-Sensitiva usando a Magnitude  
Quadrática da Coerência**

Autor:

---

Dayana Sant' Anna Lole

Orientador:

---

Prof. Marcio Nogueira de Souza, D. Sc.

Co-orientador:

---

Prof. Antonio Fernando Catelli Infantosi, Ph. D.

Co-orientador:

---

Danilo Barbosa Melges, D. Sc.

Examinador:

---

Prof. Antonio Maurício Ferreira Leite Miranda de Sá, D. Sc.

Examinador:

---

Prof. Luiz Wagner Pereira Biscaíno, D. Sc.

Examinador:

---

Prof. Eduardo Antonio Barros da Silva, Ph. D.

DEL

Outubro de 2010

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica – Departamento de Eletrônica e de Computação

Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária

Rio de Janeiro – RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade da autora e dos orientadores.

## **DEDICATÓRIA**

*"...aprender não é um ato findo. Aprender é um exercício constante de renovação..."*

Paulo Freire

A Deus, minha família, meu noivo, meus professores e amigos.

## **AGRADECIMENTO**

A Deus, em primeiro lugar, pela sua benignidade em delinear cada um dos meus dias e me prover forças para a realização de cada tarefa.

À minha mãe, Marília, que além de ter sido minha primeira professora, foi amiga e referência ao longo de toda a minha vida.

Ao meu pai, José Carlos, por todo o amor, amizade e apoio demonstrados no cuidado, muitas vezes além de seu alcance, para provisão de condições para o prosseguimento de meus estudos.

Aos meus irmãos e avós, que me fizeram sentir bem quista a cada novo dia, sem esquecerem-se de mim em suas orações.

Ao meu noivo, Marlon, pelo amor e afeto evidenciados em cada momento que se esforçou para estar ao meu lado, procurando me ajudar e estimular em todo novo passo.

Ao meu co-orientador, Danilo, pela dedicação, amizade e ensino providos durante a minha formação, tendo feito o papel de um “irmão mais velho” acadêmico e que muito me estimulou por seu apreço no desenvolvimento das atividades científicas.

Ao professor Infantsi, por compartilhar sua experiência e conhecimentos, além da paciência e carinho durante o ensino que demonstraram sua humildade e atuação genuína enquanto educador e pesquisador.

Ao professor Márcio, pela confiança e compreensão na orientação deste projeto.

Aos professores, cada um a sua maneira, que me ajudaram a compor o caminho até o momento trilhado.

Aos meus amigos, Clark, Carlos, Renan e Pedro, que foram meus companheiros durante o curso e, através de sua amizade, tornaram sua carga menos pesada.

## RESUMO

A resposta cortical à estimulação elétrica (Potencial Evocado Somato-Sensitivo – PESS) tem sido considerada importante ferramenta no diagnóstico clínico de neuropatias e na monitorização durante cirurgias. Na prática médica, a análise desta resposta se baseia na identificação, por inspeção visual, de amplitudes e latências características da forma de onda do PESS. Esta análise é subjetiva e, portanto, influenciada pela experiência do observador, pela qualidade do registro do eletroencefalograma (EEG) e pelo regime de anestesia. Para contornar tal limitação, são adotadas técnicas estatísticas de detecção objetiva da resposta (ORD, de *Objective Response Detection*). Dentre essas, a Magnitude Quadrática da Coerência (MSC – *Magnitude-Squared Coherence*), técnica no domínio da frequência, tem apresentado resultados promissores na detecção de resposta ao estímulo sensorial somático. O monitoramento da presença/ausência de resposta à estimulação ao longo do exame também pode ser realizado por meio da evolução temporal da MSC em frequências localizadas na banda de máxima resposta (30-60 Hz). Complementando esta monitorização, a evolução temporal de características do PESS também pode ser realizada para se avaliar a resposta no domínio do tempo. O presente trabalho consistiu no desenvolvimento de um sistema que permite realizar desde a análise tradicional morfológica do PESS até a monitorização da resposta cortical por meio de técnicas ORD. Para validar a ferramenta desenvolvida foram utilizados sinais EEG (Sistema Internacional 10-20) durante a estimulação do nervo tibial direito de 40 voluntários adultos normais. Esses sinais foram filtrados de 0,1 a 100 Hz e digitalizados a 600 Hz (resolução 16 bits). A estimulação foi aplicada por meio de pulsos de duração de 200  $\mu$ s, frequências de 2, 5, 7 e 9 Hz e intensidade igual ao limiar motor (menor intensidade capaz de produzir oscilações do hálux). O sistema foi desenvolvido utilizando-se a linguagem C/C++ para implementação das técnicas de processamento de sinais e a biblioteca gráfica wxWidgets para a interface com o usuário e traçado dos gráficos. Este sistema foi concebido para ser uma ferramenta de baixo custo e funcionar nas plataformas Linux e Windows. A ferramenta se mostrou adequada ao estudo de atividades corticais em resposta à estimulação somato-sensitiva, sendo extensível a outros protocolos de estimulação.

*Palavras-Chave:* Eletroencefalograma (EEG), *software* multiplataforma, potencial evocado somato-sensitivo (PESS), monitorização.

## ABSTRACT

The cortical response elicited by electrical stimulation (Somatosensory Evoked Potential – SEP) has been considered an important tool for clinical diagnosis of neuropathies and surgery monitoring. Usually, the analysis of this response is based on identifying, by visual inspection, amplitudes and latencies which are the most relevant characteristics of the SEP waveform. Such analysis is subjective and hence influenced by the observer experience, the electroencephalogram (EEG) quality and the anesthesia regimen. In order to overcome this limitation, objective response detection (ORD) statistical techniques have been adopted. Among them, the Magnitude-Squared Coherence (MSC), an ORD frequency-domain technique, has been presenting promising results for detecting response to the somatosensory stimulation. The monitoring of the presence/absence of response to the stimulation may be performed through the temporal evolution of MSC in those frequencies located within the maximum response band. The temporal evolution of SEP characteristics can also be monitored with such purpose. Therefore, this work aimed at developing a system to perform both the traditional morphological analysis of SEP and cortical response monitoring by an ORD approach. In order to evaluate the developed tool, EEG signals were recorded (International 10-20 System) during right posterior tibial nerve stimulation of 40 normal adult volunteers. These signals were filtered from 0.1 to 100 Hz and digitized at 600 Hz (16 bit resolution). The stimulation was applied using pulses with 200  $\mu$ s duration, at frequencies of 2,5,7 and 9 Hz and the motor threshold intensity level (the lowest intensity capable of produce hallux oscillations). This system was developed using the C/C++ programming languages for implementation of signal processing techniques and the wxWidgets graphical library for user interface layout and graphic plotting. Furthermore, it was designed to be a low-cost tool and run both on Linux and Windows platforms. The tool showed to be adequate for studying the cortical activities in response to the somatosensory stimulation and is extendible to other stimulation protocols.

*Key-words:* Electroencephalogram (EEG), somatosensory evoked potential (SEP) multiplatform software, monitoring.

## SIGLAS

API – *Application Programming Interface*  
CBEB – Congresso Brasileiro de Engenharia Biomédica  
CEP – Comitê de Ética em Pesquisa  
COPPE – Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa em Engenharia  
DEL – Departamento de Engenharia Eletrônica  
DFT – *Discrete Fourier Transform*  
EEG – Eletroencefalograma  
FFT – *Fast Fourier Transform*  
HUCFF – Hospital Universitário Clementino Fraga Filho  
JICAC – Jornada Giulio Massarani de Iniciação Científica, Artística e Cultural  
LAPIS – Laboratório de Processamento de Imagens e Sinais  
MSC – *Magnitude-Squared Coherence*  
ORD – *Objective Response Detection*  
PE – Potencial Evocado  
PEA – Potencial Evocado Auditivo  
PEB – Programa de Engenharia Biomédica  
PESS – Potencial Evocado Somato-Sensitivo  
PEV – Potencial Evocado Visual  
RSR – Razão Sinal-Ruído  
UFRJ – Universidade Federal do Rio de Janeiro

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
	1.1 - Tema .....	1
	1.2 - Delimitação .....	1
	1.3 - Justificativa .....	2
	1.4 - Objetivos .....	4
	1.5 - Metodologia .....	4
	1.6 - Descrição .....	5
<b>2</b>	<b>EEG Espontâneo e durante a Estimulação Somato-Sensitiva</b>	<b>6</b>
	2.1 - O Sistema Sensorial Somático .....	6
	2.2 - O Eletroencefalograma (EEG) e o Sistema Internacional 10-20	9
	2.3 - Potencial Evocado Somato-Sensitivo .....	11
	2.4 - Artefato .....	15
<b>3</b>	<b>Técnicas para Análise do PESS e Detecção da Resposta</b>	<b>17</b>
	3.1 - Análise Morfológica: Média Coerente .....	18
	3.2 - Evolução Temporal do PESS .....	20
	3.3 - Técnicas de Detecção Objetiva da Resposta Evocada (ORD) ..	21
	3.3.1 - Magnitude Quadrática da Coerência (MSC) ...	22
	3.3.2 - Considerações sobre a FFT ( <i>Fast Fourier Transform</i> ) ..	25
	3.4 - Evolução Temporal da MSC .....	26



<b>4</b>	<b>Materiais e Métodos</b>	<b>28</b>
	4.1 - Sinais Simulados .....	28
	4.2 - Sinais EEG .....	30
	4.3 - Pré-processamento .....	32
	4.3.1 - Janela Tukey .....	33
	4.3.2 - Rejeição Automática de Artefatos .....	34
<b>5</b>	<b>Avaliação do Sistema Multiplataforma para Análise do PESS</b>	<b>37</b>
	5.1 - Média coerente .....	37
	5.2 - Rejeição de Artefatos .....	41
	5.3 - Remoção do Valor Médio .....	43
	5.4 - Magnitude Quadrática da Coerência .....	45
	5.5 - Janela Tukey .....	49
	5.6 - Evolução Temporal do PESS .....	51
	5.7 - Evolução Temporal da MSC .....	54
<b>6</b>	<b>Interface com o usuário</b>	<b>57</b>
	6.1 - Leitura de Informações do Exame .....	57
	6.2 - Estimção do PESS .....	59
<b>7</b>	<b>Discussão e Conclusão</b>	<b>64</b>
<b>8</b>	<b>Trabalhos Futuros</b>	<b>66</b>
<b>9</b>	<b>Trabalhos Apresentados e Prêmio</b>	<b>67</b>

<b>Bibliografia</b>	<b>69</b>
<b>Apêndice A Resultado da aplicação do Sistema para diversas frequências de estimulação</b>	<b>74</b>
A.1 – Frequência de Estimulação de 2 Hz .....	74
A.2 – Frequência de Estimulação de 7 Hz .....	75
A.3 – Frequência de Estimulação de 9 Hz .....	76
<b>Apêndice B Guia de Instalação de wxWidgets</b>	<b>78</b>
B.1 - Linux .....	78
B.2 - Windows .....	78
<b>Apêndice C Makefile para compilação</b>	<b>79</b>
C.1 - Linux .....	79
C.2 - Windows .....	81
<b>Apêndice D Rotinas de Processamento</b>	<b>84</b>
D.1 - Leitura de cabeçalho de arquivo *.peb .....	84
D.2 - Separação do sinal em épocas .....	87
D.3 - Remoção da forma normalizada .....	92
D.4 - Pré-processamento .....	94
D.5 - Rejeição automática de artefatos .....	96
D.6 - Janela Tukey .....	97
D.7 - Média Coerente .....	98
D.8 - Remoção de nível médio .....	100

D.9 - FFT .....	101
D.10 - MSC .....	104
D.11 - Evolução temporal da MSC .....	108
D.12 - Evolução temporal do PESS .....	110
D.13 - Definição da principal classe do sistema .....	112

# Lista de Figuras

2.1 – (a) Córtex somato-sensitivo (b) Representação das diferentes regiões corporais no córtex .....	7
2.2 – Via lemniscal .....	9
2.3 – Sistema Internacional 10-20 .....	11
2.4 – Vias utilizadas de acordo com a estimulação: (a) Visual (b) Auditiva (c) Somato-sensitiva .....	12
2.5 – Amplitudes e Latências do PE .....	13
2.6 – Amplitudes e Latências características do PESS .....	14
2.7 – Exemplo de artefato de estímulo presente na morfologia do PESS .....	16
3.1 – Média Coerente .....	19
3.2 – Modelo Linear: $y[n]$ representa o estímulo (trem de impulsos); $h[n]$ , a função de transferência; $s[n]$ , a resposta evocada; $r[n]$ , o EEG espontâneo; $x[n]$ , o EEG coletado .....	19
3.3 – Procedimento para avaliação da evolução temporal da forma de onda do PESS para $N = 4$ estimações: a cada estimação são extraídas as amplitudes e latências do P37 e N45 após realização da Média Coerente de $M$ épocas .....	20
3.4 – Transformada de Fourier de $Y_i(f)$ , representada pelos pontos, e a soma dos $Y_i(f)$ , representada pela seta, no plano de Argand-Gauss: (a) Presença de resposta explícita pelo sincronismo de fase (b) Ausência de resposta pela falta de sincronismo de fase .....	24
3.5 – Identificação da banda de máxima resposta (30-60 Hz) .....	27
4.1 – Parte do sinal lido pela rotina de leitura desenvolvida em C++: Em azul está a estimulação; em verde, uma senóide com $RSR = 0,0152$ ; em vermelho, uma senóide amortecida com $RSR = 4,0$ .....	29
4.2 – Posicionamento de eletrodos .....	31
4.3 – Aplicação de estímulo .....	31
4.4 – (a) Derivação C3, C4 e Cz do Sistema Internacional 10-20 localizadas sobre o (b) córtex somato-sensitivo .....	32
4.5 – Janela Tukey .....	34
4.6 – Cálculo do desvio padrão ( $\delta$ ) por meio de <i>software</i> desenvolvido em LabVIEW .....	35

5.1 – Senóide de 20 Hz com aplicação da Média Coerente (em verde, estimulação; em azul, sinal promediado): (a) $M = 10$ épocas (b) $M = 100$ épocas (c) $M = 200$ épocas .....	39
5.2 – Senóide amortecida de 20 Hz com aplicação da Média Coerente (em verde, estimulação; em azul, sinal promediado): (a) $M = 10$ épocas (b) $M = 100$ épocas (c) $M = 200$ épocas .....	40
5.3 – Estimativa do PESS por Média Coerente ( $M = 500$ épocas) para o Voluntário #1 estimulado a 11 mA com frequência de estimulação nominal de 5 Hz: (a) Derivação C3 (b) Derivação C4 (c) Derivação Cz .....	41
5.4 – Efeito da Rejeição Automática de Artefatos na Média Coerente do PESS para derivação C3 do Voluntário #2 estimulado a 11 mA com frequência de estimulação nominal de 5 Hz ( $M = 1000$ épocas): (a) Antes e (b) depois da aplicação da rejeição de artefatos .....	42
5.5 – Remoção de nível médio de senóide de 1 Hz: (a) com nível DC (b) sem nível DC .....	43
5.6 – Remoção de nível médio da média coerente: (a) com nível DC (b) sem nível DC .....	44
5.7 – FFT obtida através de rotina desenvolvida: (a) Senóide de 10 Hz (b) Senóides de 1 e 20 Hz somadas (c) Senóide de 20 Hz com ruído aditivo .....	46
5.8 – MSC da derivação (a) C3 (b) C4 (c) Cz do Voluntário #3 estimulado a 10 mA com frequência de estimulação nominal de 5 Hz ( $M = 50$ épocas e $\alpha = 0,05$ : $\hat{\kappa}_{crit} = 0,06$ ) .....	47
5.9 – MSC da derivação (a) C3 (b) C4 (c) Cz do Voluntário #3 estimulado a 10 mA com frequência de estimulação nominal de 5 Hz ( $M = 500$ épocas e $\alpha = 0,05$ : $\hat{\kappa}_{crit} = 0,006$ ) .....	48
5.10 – (a) PESS e (b) MSC da derivação C3 para o Voluntário #4 estimulado a 10 mA com frequência de estimulação nominal de 5 Hz ( $M = 500$ épocas e $\alpha = 0,05$ : $\hat{\kappa}_{crit} = 0,006$ ) .....	49
5.11 – (a) PESS e (b) MSC da derivação C3 para o Voluntário #4 estimulado a 10 mA com frequência de estimulação nominal de 5 Hz ( $M = 500$ épocas e $\alpha = 0,05$ : $\hat{\kappa}_{crit} = 0,006$ ) .....	50
5.12 – Evolução Temporal do PESS da derivação Cz para o Voluntário #5 estimulado a 11 mA com frequência de estimulação nominal de 5 Hz ( $M = 50$ épocas por estimativa): (a) Latência e (b) Amplitude do P37 (c) Latência e (d) Amplitude do N45 .....	52
5.13 – Evolução Temporal do PESS da derivação Cz para o Voluntário #5 estimulado a 11 mA com frequência de estimulação nominal de 5 Hz ( $M = 500$ épocas por estimativa): (a) Latência e (b) Amplitude do P37 (c) Latência e (d) Amplitude do N45 .....	53

5.14 – Evolução Temporal da MSC da derivação Cz para o Voluntário #6 estimulado a 11 mA com frequência de estimulação nominal de 5Hz ( $\alpha=0,05$ ): (a) $M = 50$ épocas (1 estimativa = 10 s de exame) (b) $M = 100$ épocas (1 estimativa = 20 s de exame) (c) $M = 200$ épocas (1 estimativa = 40 s de exame) .	55
5.15 – MSC da derivação Cz para o Voluntário #6 estimulado a 11 mA com frequência de estimulação nominal de 5Hz ( $M = 500$ épocas e $\alpha=0,05$ : $\hat{\kappa}_{crit} = 0,006$ )	56
6.1 – <i>Menu</i> principal do sistema: Seleção de funcionalidade de leitura do cabeçalho do exame	58
6.2 – Interface de leitura de cabeçalho de arquivo *.peb nas plataformas (a) Windows e (b) Linux	59
6.3 – Interface para inserção de parâmetros para processamento do exame nas plataformas (a) Windows e (b) Linux	60
6.4 – Seleção de derivações desejadas nas plataformas (a) Windows e (b) Linux .	62
6.5 – Apresentação gráfica de resultados nas plataformas (a) Windows e (b) Linux	62
6.6 – Opções de melhor visualização gráfica e impressão	63
A.1 – Estimativa do PESS (derivação Cz) por Média Coerente ( $M = 500$ épocas) para o Voluntário #7 estimulado a 11 mA com frequência de estimulação nominal de 2Hz	74
A.2 - MSC do PESS (derivação Cz) para o Voluntário #7 estimulado a 11 mA com frequência de estimulação nominal de 2Hz ( $M = 500$ épocas e $\alpha = 0,05$ : $\hat{\kappa}_{crit} = 0,006$ )	75
A.3 – Estimativa do PESS (derivação Cz) por Média Coerente ( $M = 500$ épocas) para o Voluntário #8 estimulado a 11 mA com frequência de estimulação nominal de 7Hz	75
A.4 – MSC do PESS (derivação Cz) para o Voluntário #8 estimulado a 11 mA com frequência de estimulação nominal de 7Hz ( $M = 500$ épocas e $\alpha = 0,05$ : $\hat{\kappa}_{crit} = 0,006$ )	76
A.5 – Estimativa do PESS (derivação Cz) por Média Coerente ( $M = 500$ épocas) para o Voluntário #9 estimulado a 11 mA com frequência de estimulação nominal de 9Hz	76
A.6 – MSC do PESS (derivação Cz) para o Voluntário #9 estimulado a 11 mA com frequência de estimulação nominal de 9Hz ( $M = 500$ épocas e $\alpha = 0,05$ : $\hat{\kappa}_{crit} = 0,006$ )	77

# Lista de Tabelas

5.1 – Parâmetros da janela Tukey utilizada .....	50
6.1 – Informações do cabeçalho de um arquivo *.peb .....	58
6.2 – Parâmetros para estimação do PESS .....	61

# Capítulo 1

## Introdução

### 1.1 – Tema

O presente trabalho se insere na área de Processamento de Sinais Biológicos, uma das disciplinas de concentração da Engenharia Biomédica. Nesse contexto, este projeto consiste no desenvolvimento de um sistema computacional multiplataforma (execução em Windows e Linux) que incorpore métodos desta área de conhecimento visando ao processamento de sinais eletroencefalográficos (EEG) adquiridos em protocolo de estimulação elétrica do nervo tibial. Assim, para obtenção do Potencial Evocado Somato-sensitivo (PESS), resposta cortical eliciada pela estimulação, utiliza-se a Média Coerente, técnica no domínio do tempo. Para a obtenção desta resposta no domínio da frequência, emprega-se a Magnitude Quadrática da Coerência (MSC), técnica estatística de Detecção Objetiva da Resposta Evocada (ORD, de *Objective Response Detection*), a qual permite estabelecer *a priori* uma taxa máxima de falsos-positivos.

Além disso, o sistema permite o monitoramento da presença/ausência de resposta à estimulação utilizando para tal a evolução temporal de parâmetros extraídos do PESS e/ou obtidos a partir da MSC.

### 1.2 – Delimitação

Este estudo trata de procedimentos de processamento digital de sinais aplicados ao eletroencefalograma (EEG) durante estimulação elétrica do nervo tibial. Dessa forma, o sistema desenvolvido permite a análise e o monitoramento de sinais previamente coletados tanto por meio da estimação do potencial evocado somato-sensitivo (PESS) como da aplicação de técnicas de detecção objetiva da resposta. Estas



últimas, surgidas no contexto de sinais EEG na década de 1980, contrapõem-se à inspeção visual da forma de onda do PESS no domínio do tempo, usualmente realizada na prática médica. Isso se deve ao estabelecimento de critérios estatísticos para detecção da resposta sensorial, promovido por tais técnicas.

Além disso, o sistema, implementado através de ferramentas não-proprietárias, pode ser empregado tanto na plataforma Linux como Windows, o que lhe confere abrangência e versatilidade, especialmente no âmbito científico.

### **1.3 – Justificativa**

O eletroencefalograma (EEG) consiste no registro da atividade elétrica cerebral por meio de eletrodos posicionados no escalpo. Trata-se, portanto, de um exame não-invasivo e de baixo custo, comumente empregado no diagnóstico clínico de disfunções do Sistema Nervoso.

A resposta cortical eliciada por estímulo sensorial, denominada Potencial Evocado (PE), possui amplitudes e latências características da normalidade, o que justifica sua utilização no diagnóstico clínico de neuropatias e monitorização intra-operatória e pós-cirúrgica. Todavia, sua forma de onda possui reduzida magnitude ( $5 \mu\text{V}$ ) quando comparada à do EEG espontâneo ( $100 \mu\text{V}$ ), sendo mascarada por este. Logo, para obter sua estimação, realiza-se a média de um determinado número de épocas (trechos de sinal) de sinal EEG, sincronizadas com a estimulação, a qual é denominada Média Coerente.

Em geral, a análise do PE é realizada por inspeção visual de sua forma de onda, apresentando, portanto, elevado caráter subjetivo e dependência da experiência do profissional que a realiza. Um tipo de PE amplamente utilizado, especialmente durante cirurgias de coluna e vasculares, é o Potencial Evocado Somato-Sensitivo (PESS), o qual usualmente é eliciado pela aplicação de estímulos elétricos de baixa intensidade.

O PESS tem sido amplamente utilizado na prática clínica, bem como na monitorização de cirurgia de ressecção da aorta torácica e abdominal [14] e no prognóstico de hipertensão intracraniana [15].

Assim, com o intuito de aperfeiçoar e ampliar a utilização do PESS, inúmeros estudos têm investigado a aplicação de técnicas de processamento de sinais ao EEG durante estimulação somato-sensitiva, tanto no domínio da frequência quanto no domínio do tempo. Nesse âmbito, destacam-se as técnicas de Detecção Objetiva da Resposta (ORD – *Objective Response Detection*), que empregam testes estatísticos para verificar a presença ou ausência de resposta cortical. Dentre tais técnicas, aquela conhecida como Magnitude Quadrática da Coerência (MSC – *Magnitude Squared Coherence*) tem apresentado resultados relevantes mesmo em casos considerados duvidosos, contribuindo também para o treinamento profissional. Por se tratar de uma técnica no domínio da frequência, permite levar em consideração o sincronismo entre o estímulo e a resposta cortical e, conseqüentemente, melhor compreender as atividades cerebrais envolvidas no processamento da informação sensorial.

Ademais, o acompanhamento da atividade cortical ao longo do exame através da evolução temporal da MSC nas frequências da banda de máxima resposta permite verificar a presença/ ausência de resposta à estimulação. Esta técnica, complementada pela evolução temporal de suas amplitudes e latências características, no domínio do tempo, oferece mais informações para uma monitorização cirúrgica mais completa.

Com base neste cenário, o desenvolvimento de um *software* que permita a análise morfológica do PESS e sua investigação por meio de técnicas objetivas é relevante no auxílio ao diagnóstico e à monitorização cirúrgica. Além disso, é requisito para essa ferramenta a possibilidade de emprego em Windows e Linux, a fim de atender a um maior número de usuários. Adicionalmente, seu desenvolvimento deve basear-se em ferramentas livres e gratuitas, levando à construção de um sistema computacional auditável e com baixo custo de desenvolvimento, características fundamentais para a aplicação em pesquisas científicas.

## 1.4 – Objetivos

O presente trabalho visa desenvolver um sistema livre e multiplataforma capaz de sistematizar a análise de sinais EEG durante estimulação elétrica. Seus objetivos específicos compreendem:

- Apresentação gráfica do sinal coletado;
- Estimação e apresentação do PESS no domínio do tempo, através da implementação da média coerente;
- Implementação e exibição da MSC, permitindo a detecção objetiva da resposta evocada;
- Implementação de etapas de pré-processamento com vistas à mitigação de efeitos relativos à presença de artefatos;
- Acompanhamento da evolução temporal da MSC, no domínio da frequência, e das amplitudes e latências do PESS, no domínio do tempo.

## 1.5 – Metodologia

Diversas técnicas de pré-processamento e processamento foram implementadas e aplicadas a sinais simulados e sinais EEG reais coletados de voluntários adultos normais submetidos à estimulação somato-sensitiva (elétrica).

O sistema desenvolvido incorpora técnicas de pré-processamento para redução de efeitos de artefatos, tais como o janelamento Tukey (artefato de estímulo) e um algoritmo de rejeição semi-automática de épocas com elevado nível de ruído (artefatos fisiológicos, externos ou instrumentais). Para a obtenção do PESS utilizou-se a técnica da Média Coerente, a qual consiste na realização de uma média de  $M$  janelas (épocas) de sinal sincronizadas com os instantes de estimulação.

Com vistas à detecção objetiva da resposta evocada, foi utilizada a MSC, definida pela razão entre o espectro cruzado do sinal pelo produto das raízes quadradas dos auto-espectros. Esta técnica foi implementada com o auxílio da FFT (*Fast Fourier Transform*) aplicada às janelas de sinal EEG. A identificação da resposta acontece através da comparação da estimativa realizada com um limiar, estabelecido a partir do nível de significância  $\alpha$  adotado (taxa máxima de falsos-positivos na condição de não-estimulação).

## **1.6 – Descrição**

No Capítulo 2, é realizada a introdução a conceitos relativos ao sistema nervoso humano e, particularmente, ao EEG durante a estimulação somato-sensitiva. No capítulo seguinte, são expostos conceitos relativos ao processamento do EEG através de técnicas no domínio do tempo e da frequência. O Capítulo 4, então, apresenta as ferramentas de desenvolvimento, teste e aprimoramento da análise da resposta evocada, justificando sua utilização. O quinto capítulo tem o objetivo de demonstrar e discutir alguns testes e resultados obtidos através do sistema desenvolvido. O Capítulo 6 ilustra o uso do sistema através de sua interface, esclarecendo aspectos sobre seus requisitos de usabilidade. No sétimo capítulo são realizadas breve discussão e conclusão. Finalmente, os Capítulos 8 e 9 apresentam, respectivamente, perspectivas de trabalhos futuros e Trabalhos publicados e Prêmios. Dos apêndices constam alguns exemplos de utilização do sistema para frequências de estimulação diferentes daquelas apresentadas ao longo do texto (Apêndice A), um guia para instalação da biblioteca gráfica (Apêndice B), o *makefile* para Windows e Linux (Apêndice C) e as listagens de código relativas à implementação do processamento (Apêndice D).

## **Capítulo 2**

# **EEG Espontâneo e durante a Estimulação Somato-Sensitiva**

O EEG tem sido considerado uma ferramenta útil no diagnóstico clínico de disfunções do Sistema Nervoso e na monitorização intra e pós-operatória. Em especial, quando da aplicação de estimulação sensorial, o EEG contém a resposta cortical, que pode ser estimada pelo Potencial Evocado, o qual se constitui em um importante método na avaliação da integridade das vias sensoriais e motoras.

### **2.1 – O Sistema Sensorial Somático**

O sistema sensorial somático (ou somato-sensorial) é responsável por fornecer informação sobre o ambiente externo e sua integração com o interno. Os receptores deste sistema encontram-se distribuídos por todo o corpo, apresentando modalidades perceptuais relacionadas ao tato e vibração (mecanorreceptores), aos deslocamentos mecânicos de músculos e articulações (proprioceptores), à sensação térmica (termorreceptores) e à dor (nociceptores). Podem ser encontrados na pele, músculos, órgãos viscerais e articulações. A Figura 2.1 ilustra o mapeamento sensorial de regiões do corpo no córtex cerebral. Este mapeamento é conhecido como somatotopia.

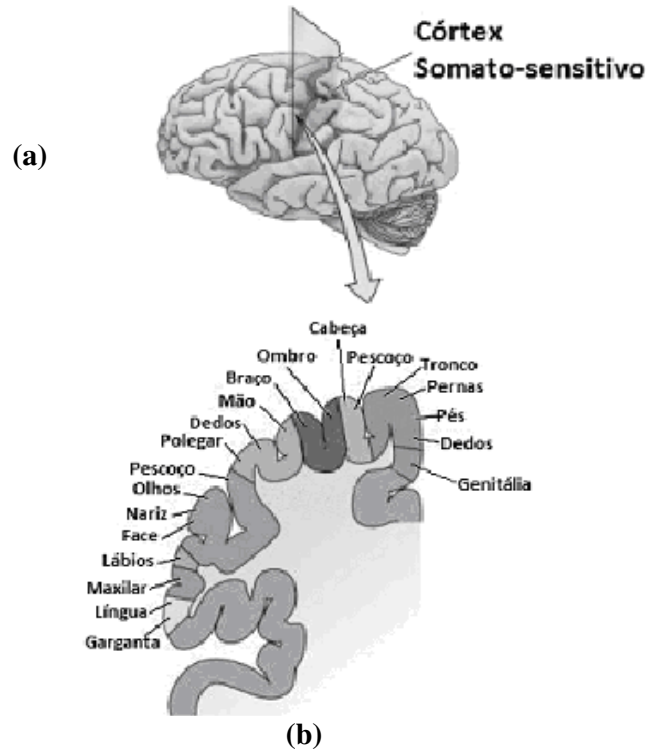


Figura 2.1 – (a) Córtex somato-sensitivo (b) Representação das diferentes regiões corporais no córtex

Fonte: Adaptado de [3]

Os receptores sensoriais apresentam resposta relacionada à intensidade do estímulo. Dessa forma, para que um receptor seja ativado, a magnitude do estímulo deve ultrapassar um dado limiar. A maioria dos receptores responde exclusivamente a certos estímulos, sendo denominados receptores específicos. Existem, ainda, aqueles que se relacionam com mais de um tipo de estímulo, conhecidos, portanto, como multimodais. Estes últimos, todavia, apresentam limiar de resposta mais baixo (maior sensibilidade) para determinadas excitações em detrimento de outras (menor sensibilidade em função de limiar mais elevado).

Adicionalmente, a resposta sensorial somática a excitações variantes ou abruptas é mais intensa. No sentido contrário, essa resposta tem seu potencial de ação reduzido quando há persistência do estímulo.

A sensação traduzida pelos receptores é transmitida pelos neurônios através de impulsos (potenciais de ação). Os axônios dos receptores são designados aferentes

sensoriais (ou vias sensoriais), por realizarem o transporte dessa informação pela medula espinhal ou pelo tronco cerebral até o córtex (área sensorial central), onde é interpretada. Os axônios provenientes dos mecanorreceptores adentram a via lemniscal (ou lemnisco medial – Figura 2.2) através da raiz dorsal e sobem pela coluna dorsal através dos fascículos grácil e cuneiforme, onde as fibras mielinizadas e calibrosas fazem sinapses com os neurônios de segunda ordem<sup>1</sup> no bulbo. Os axônios dos neurônios de segunda ordem sofrem decussação<sup>2</sup> e, através da via lemnisco-medial, levam informação ao tálamo. Nos núcleos ventrais postero-laterais do tálamo ocorrem as sinapses com os neurônios de terceira ordem, que se ligam ao giro pós-central do córtex cerebral, região correspondente ao córtex somato-sensitivo primário.

Descobertos em 1906, pelo fisiologista Sir Charles Sherrington, os nocirreceptores são terminações nervosas que atuam na possibilidade de injúria física<sup>3</sup> ou química ao organismo. Analogamente ao funcionamento dos mecanorreceptores, a trajetória do estímulo de dor tem início pela transmissão da informação por meio de sinapses dos neurônios aferentes com a raiz dorsal da medula espinhal, que se apresenta separada em lâminas. As sinapses são formadas entre as lâminas e os neurônios aferentes de acordo com o tipo destes últimos. Após a primeira sinapse, os neurônios sofrem decussação e seguem pelo trato espinotalâmico por duas vias distintas: sistema lemnisco-medial, em caso de dor regular, e sistema ântero-lateral, para dores intensas. Por fim, atinge-se o núcleo ventral posterior do tálamo seguido do córtex somato-sensitivo primário, onde a consciência de dor é formada [21]. Esta percepção pode levar à liberação de substâncias químicas pelo hipotálamo para inibição ou redução da dor.

---

<sup>1</sup> Neurônios aferentes são denominados de segunda ordem após a primeira sinapse na via lemniscal.

<sup>2</sup> Decussação é o cruzamento oblíquo no plano mediano (em forma de X) dos feixes nervosos do Sistema Nervoso Central.

<sup>3</sup> As injúrias físicas podem ter origem mecânica ou térmica.

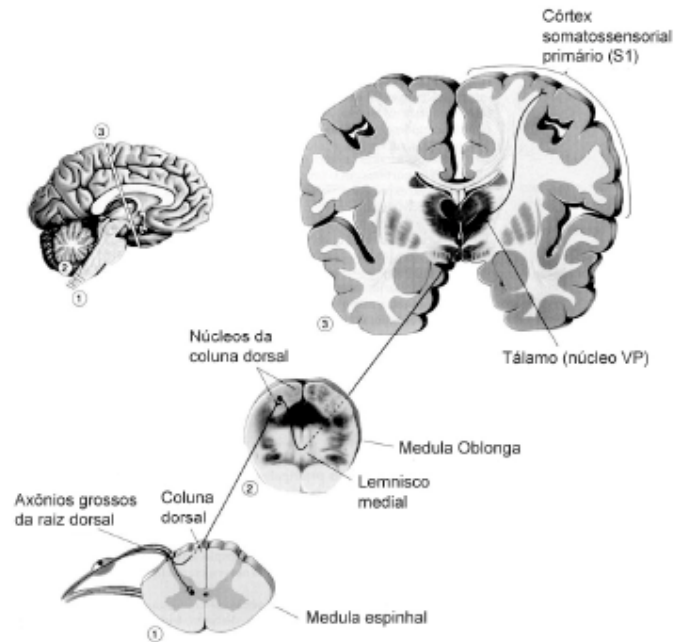


Figura 2.2 – Via lemniscal  
 Fonte: Adaptado de [4]

## 2.2 – O Eletroencefalograma (EEG) e o Sistema Internacional 10-20

O Eletroencefalograma (EEG) é o registro da atividade elétrica cerebral, gerado a partir de correntes iônicas de origem neuronal. Mais precisamente, o EEG é o resultado da somação espaço-temporal dos potenciais pós-sinápticos inibitórios e excitatórios. O exame do EEG tem sido utilizado, por exemplo, no diagnóstico de epilepsia, tumores cerebrais, acidente vascular encefálico e traumatismos cranianos, bem como na investigação de períodos de inconsciência e distúrbios no sono [19].

Quando este registro é realizado na ausência de estímulo, é denominado EEG de fundo ou espontâneo. Neste contexto, a atividade cerebral apresenta registros da ordem de 10 a 100  $\mu\text{V}$  e banda de 0,1 a 100 Hz [27]. A atividade rítmica do EEG é dividida em bandas [5,6,11,18, 25,26, 28,32]:

- Delta (0,1 - 4 Hz): Caracterizada por elevadas amplitudes e presente em bebês, no sono profundo de adultos e em atividades que exigem atenção contínua. É



utilizada para o diagnóstico de lesões difusas e subcorticais, hidrocefalia metabólica e lesões profundas na linha média;

- Teta (4 - 7 Hz): Encontrada em crianças, na sonolência ou ociosidade de jovens e adultos e na inibição de respostas eliciadas (quando o paciente procura reprimir ativamente uma resposta). Empregada no diagnóstico de encefalopatia metabólica, lesões subcorticais focais, entre outras;
- Alfa (8 – 12 Hz): Associada ao controle de inibição, fechamento de olhos e estado de relaxamento, e aproveitada no acompanhamento de coma;
- Beta (12 – 30 Hz): Em geral, apresenta-se no estado de alerta (atenção) e é atenuada na atividade motora intensa. Pode ser empregada no diagnóstico de neuropatias decorrentes da utilização de drogas;
- Gama (acima de 30 Hz): Apresenta-se durante o processamento de informação sensorial (e.g. visão, audição e tato) e na memória de curto prazo através do reconhecimento de objetos, sons e sensações. A redução de atividade nesta banda pode estar associada à diminuição de capacidade cognitiva.

O EEG é obtido por meio do posicionamento de eletrodos de superfície sobre o escalpo do paciente. Devido à sua baixa amplitude, o EEG deve ser acondicionado utilizando-se amplificadores biológicos e filtros. Comumente, o posicionamento dos eletrodos é realizado com base no Sistema Internacional 10-20 (Figura 2.3). Este padrão, criado em 1958 por um comitê da *International Federation of Societies for Electroencephalography and Clinical Neurophysiology*, propõe o distanciamento mútuo dos eletrodos a 10% e 20% de distância entre marcas anatômicas bem definidas. O ponto médio da sutura nasofrontal (násio) e o vértice da protuberância occipital externa (ínion) são as marcas anatômicas em que se baseia este posicionamento.

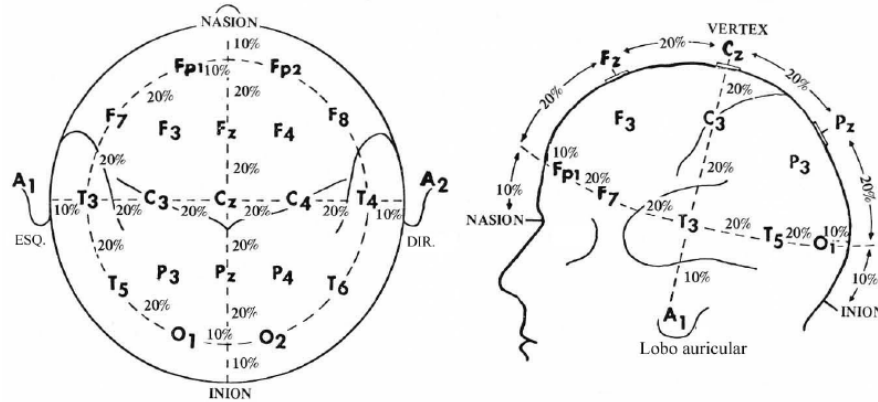


Figura 2.3 – Sistema Internacional 10-20

Fonte: Adaptado de [17].

A nomenclatura das posições é dada segundo a região anatômica e o hemisfério em que estão localizadas. Assim, as regiões anatômicas são identificadas pelas letras: Fp (frontopolar), F (frontal), C (central), T (temporal), P (parietal), O (occipital) e A (auricular). Eletrodos no hemisfério direito recebem numeração par; no esquerdo, ímpar; aqueles que se encontram sobre a linha média recebem a letra “z” [17].

## 2.3 – Potencial Evocado Somato-Sensitivo

O Potencial Evocado (PE) é a resposta do Sistema Nervoso a estímulos motores ou sensoriais [25]. Esse potencial pode ser classificado, conforme o tipo de estimulação aplicada, como:

- Visual (PEV): Estimulação por meio de padrões visuais (e.g.: tabuleiro de xadrez alternante) apresentados, em geral, por meio de uma tela (Figura 2.4a);
- Auditivo (PEA): Estimulação por *clicks*, tons ou tons modulados (Figura 2.4b);
- Somato-sensitivo (PESS): Estimulação através de, por exemplo, mudanças de temperatura, movimentação de juntas, dor, aplicação de corrente elétrica (Figura

2.4c). Esta última é a mais utilizada por ser facilmente controlada e produzir potenciais melhor definidos e de amplitudes mais elevadas [7].

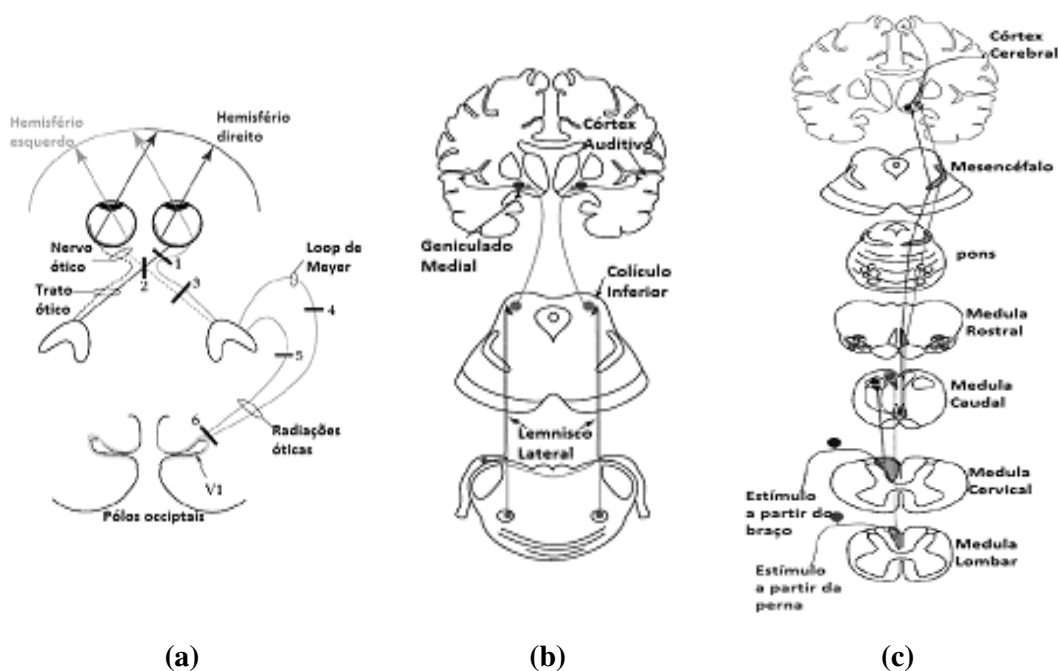


Figura 2.4 – Vias utilizadas de acordo com a estimulação: (a) Visual (b) Auditiva (c) Somato-sensitiva

Fonte: Adaptado de [32].

O PESS pode ser classificado como sendo de curta, média ou longa latência. As respostas evocadas somato-sensitivas de curta latência acontecem até 50 ms [28] e apresentam menor variabilidade quando comparadas às de média e longa latência, o que justifica sua aplicabilidade clínica e intra-operatória. Além disso, as amplitudes relacionadas ao PESS são, pelo menos, cerca de 10 vezes menores do que aquelas apresentadas pelo EEG de fundo [16], sendo, portanto, mascaradas por ele.

A neuromonitorização realizada por meio do PE é baseada na inspeção visual de sua forma de onda através de amplitudes, latências e polaridades características [25], conforme a Figura 2.5. Aumentos na latência ou diminuições nas amplitudes, em geral, são indicativos de disfunções neurológicas observadas na avaliação médica. Embora apresente variações em função de dosagem anestésica, o PESS é o PE com menor

sensibilidade à anestesia [13], o que o torna mais interessante na monitorização durante cirurgias.

Em especial, o PESS por estimulação do nervo tibial apresenta um pico e um vale característicos<sup>4</sup> de sua forma de onda (Figura 2.6), úteis no diagnóstico clínico. O vale ocorre, aproximadamente, aos 37 ms pós-estímulo, sendo, por isso, denominado P37. Analogamente, o N45 é um pico que acontece em torno de 45 ms pós-estímulo. Na análise do PESS, positividade é identificada por vales e negatividade por picos.

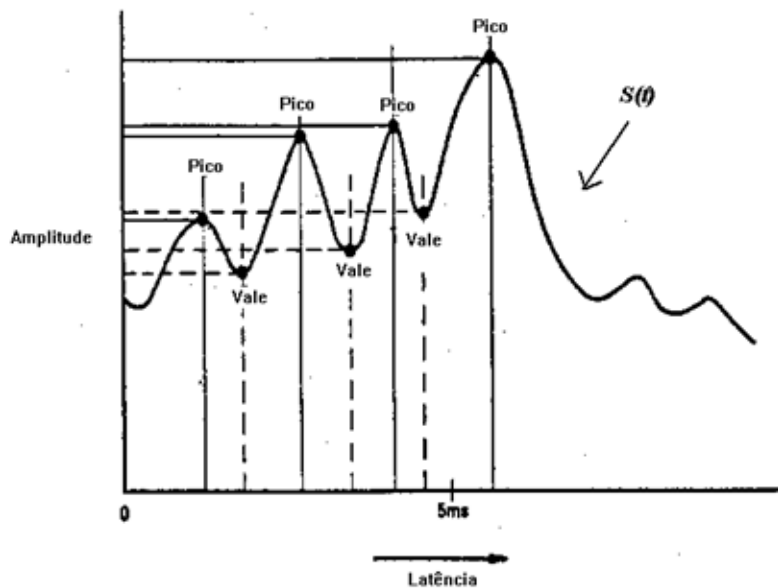


Figura 2.5 – Amplitudes e Latências do PE

Fonte: [17]

<sup>4</sup> Estes picos e vales característicos podem ser mais claramente visualizados ao se aplicar estimulação com intensidade igual à do limiar motor (menor intensidade capaz de gerar contração involuntária dos músculos intrínsecos do pé). Outras intensidades de estimulação podem causar variações nos valores de latência. Segundo [7], por exemplo, um aumento de latências acontece para intensidades abaixo do limiar motor.

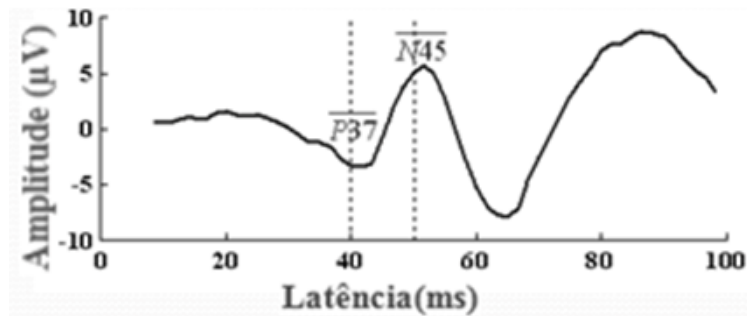


Figura 2.6 – Amplitudes e Latências características do PESS

Fonte: [22]

Mediante a facilidade na detecção de resposta evocada e extração de características que evidenciem a normalidade, a monitorização do PESS tem alcançado ampla aplicabilidade na monitorização cirúrgica (e.g: cirurgias vasculares e de coluna). Suas vantagens compreendem, entre outras, o baixo custo no acompanhamento da atividade cerebral através dos quatro membros, mitigando a possibilidade de complicações neurológicas transitórias ou permanentes. Mesmo em casos em que sequelas pós-operatórias não podem ser evitadas, o acompanhamento da atividade neural pelo PESS permite a prevenção de maiores danos. Além disso, o PESS tem apresentado resultados promissores no prognóstico de pacientes em coma em decorrência de hemorragia cerebral espontânea, além do reconhecimento de pacientes de alto risco e que podem requerer maiores medidas terapêuticas [11].

Dessa forma, o PESS é utilizado na monitorização de cirurgias de coluna e vasculares, em função de sua sensibilidade à hipotensão, estresse mecânico e isquemia cortical, permitindo a detecção de possibilidade de injúria ao tecido nervoso. Além disso, apresenta variações de acordo com a idade dos indivíduos e a temperatura dos membros utilizados no exame [21]. Este último fator permite o uso do PESS como meio mais adequado na identificação da temperatura ótima em cirurgias que necessitam de hipotermia profunda com vistas à redução da atividade cerebral.

O PESS também tem sido empregado no acompanhamento pós-operatório da atividade cortical, permitindo a identificação de sequelas neurológicas tardias. Isto se justifica por fazer uso de métodos não-invasivos e da confiabilidade conferida pela sua menor variabilidade mediante aplicação de sedativos.

Todavia, a qualidade do exame de PESS também sofre interferência da taxa de estimulação, que permite caracterizar o PESS como transiente ou em regime permanente. A obtenção do PESS em regime permanente se caracteriza pela elevada taxa de estimulação, que implica em um potencial com duração maior que o intervalo interestímulo – analogamente a um circuito ressonante em regime permanente [30]. Dessa forma, o PESS em regime permanente, permite sua descrição em termos de amplitude e fase para cada componente de frequência. Por outro lado, o PE transiente ocorre quando sua estimulação se dá dentro do intervalo interestímulo.

Ademais, a obtenção do PE e, portanto, do PESS é influenciada pela intensidade do estímulo aplicado. Uma vez estabelecido o limiar de estimulação (menor intensidade capaz de produzir resposta evocada [25]), intensidades baixas próximas a este têm como efeito a necessidade de um maior número de épocas para que se possa visualizar o PESS. A aplicação de elevadas intensidades de estimulação, por outro lado, causa uma menor latência dos picos, representando, ainda, risco de injúria ao tecido nervoso. A fim de prevenir tais riscos, a aplicação de estímulo em locais identificados como de menores limiares é aconselhável [28].

## **2.4 – Artefato**

Artefato é qualquer ruído presente no EEG ou PE, podendo comprometer sua análise. Há basicamente três tipos de artefatos [31]:

- Externos: em geral são provocados por ruídos elétricos ou eletromagnéticos provenientes de sistemas não envolvidos na aquisição ou estimulação. Podem-se citar, nesse caso, ruídos eletromagnéticos de 60 e 120 Hz, difíceis de eliminar (blindagem), sendo induzidos principalmente por ventiladores, condicionadores de ar e lâmpadas fluorescentes;
- Instrumentais: podem ser gerados por qualquer parte do sistema atuante na realização do exame. Exemplo: mau contato de eletrodos e o ruído térmico gerado pelo aquecimento de componentes do amplificador de EEG;

- Fisiológicos: este tipo de artefato é caracterizado por sinais elétricos de origem não cerebral detectados durante o EEG, tendo sido gerados no organismo do paciente. Estes artefatos incluem movimentos oculares, musculares espontâneos ou não, respiratórios, pulso e artefatos glossocinéticos, entre outros.

Um ruído que frequentemente compromete o registro do PESS é o artefato de estímulo, relacionado à capacidade condutiva da pele em contato com os eletrodos. Este artefato apresenta elevada amplitude e banda larga (Figura. 2.7), além de ocorrer nos primeiros milissegundos pós-estímulo, geralmente os 5 ms iniciais para o PESS do nervo tibial [35]. Por ser síncrono com a excitação, este artefato não pode ser atenuado através da média coerente, tendo implicações na análise do PESS tanto no domínio do tempo quanto no da frequência.

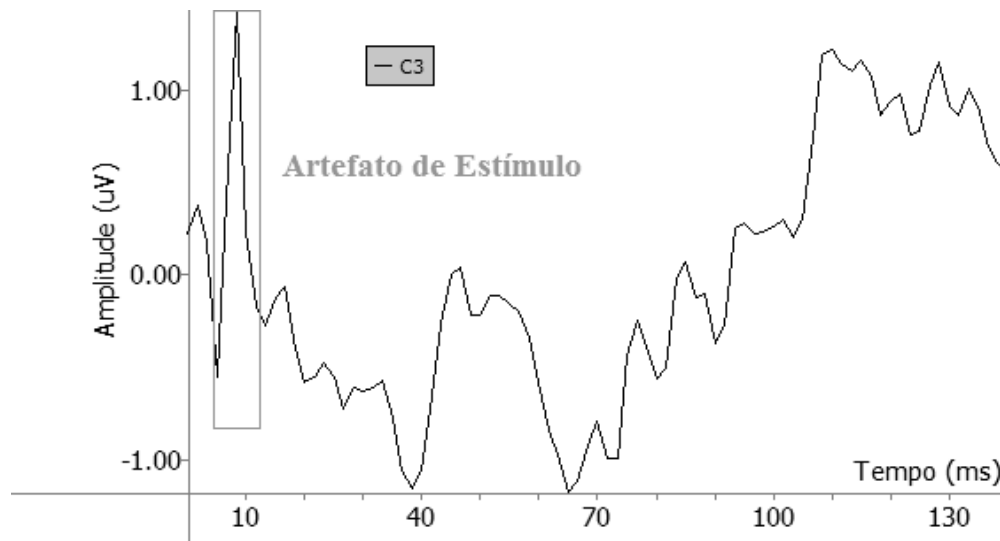


Figura 2.7 – Exemplo de artefato de estímulo presente na morfologia do PESS

Fonte: Elaboração própria

## Capítulo 3

# Técnicas para Análise do PESS e Detecção da Resposta

A monitorização contínua do EEG é justificada por sua sensibilidade a muitas condições patológicas tais como isquemia cortical, hidrocefalia e hipertensão intracraniana. Todavia, os sinais EEG não apresentam padrão bem definido, embora, dependendo do protocolo do exame, algumas características possam ser notadas. No caso do PESS, as características indicativas de anormalidade podem estar relacionadas, por exemplo, ao aumento da latência e à redução de amplitudes de ondas características. Entretanto, o PESS é mascarado pelo EEG de fundo.

Na prática clínica, a forma de onda do PESS é avaliada mediante inspeção visual; logo, é um procedimento subjetivo. Assim, a qualidade do EEG, o regime de anestesia e a variabilidade inter-observador e inter-paciente acabam por influenciar na interpretação do exame [20]. Esses fatores justificam a implementação de técnicas que permitam extrair características de interesse do EEG coletado e, conseqüentemente, facilitem a interpretação do mesmo.

Nesse contexto, está inserida a técnica conhecida como Média Coerente, que permite realizar a estimativa da resposta evocada no domínio do tempo. Por outro lado, a resposta somato-sensitiva pode ser analisada no domínio da frequência usando-se técnicas de detecção objetiva da resposta evocada (ORD, de *Objective Response Detection*).

Este capítulo apresenta uma descrição detalhada das técnicas Média Coerente e Magnitude Quadrática da Coerência (ORD), bem como justifica sua utilização neste trabalho.



### 3.1 – Análise Morfológica: Média Coerente

A despeito de sua ampla aplicabilidade, o PESS apresenta baixa amplitude (5  $\mu\text{V}$ ) se comparada à magnitude do EEG espontâneo (100  $\mu\text{V}$ ), que o mascara. Por este motivo, comumente realiza-se a média coerente, com o intuito de se obter uma estimativa do PESS.

Essa técnica se baseia na realização da média de  $M$  épocas do sinal coletado, sincronizadas com o instante de estimulação. Isto se justifica por assumir-se o EEG espontâneo como um ruído branco gaussiano de média zero e a resposta evocada síncrona com o estímulo e idêntica em todas as épocas. Uma ilustração desta técnica é apresentada na Figura 3.1, na qual se pode observar um esboço da forma de onda característica do PESS por estimulação do nervo tibial contendo o pico N45 e o vale P37.

Considerando o modelo linear de geração de resposta cortical ao estímulo (Figura 3.2), temos, matematicamente [21]:

$$x_i[n] = s[n] + r_i[n], \quad \text{Equação 3.1}$$

onde  $x_i[n]$  é a  $i$ -ésima época de EEG coletado,  $s[n]$  é a resposta evocada (note que não foi adotado o índice  $i$  para este termo em conformidade com o fato de essa resposta ser idêntica em todas as épocas) e  $r_i[n]$  o ruído.

Assim, a estimativa do PE pode ser obtida realizando-se a Média Coerente, calculada por:

$$\hat{s}[n] = \frac{1}{M} \sum_{i=1}^M x_i[n] = \frac{1}{M} \sum_{i=1}^M s[n] + \frac{1}{M} \sum_{i=1}^M r_i[n], \quad \text{Equação 3.2}$$

onde  $M$  é o número de épocas e o sobrescrito  $\hat{\phantom{x}}$  indica que se trata de uma estimação. Na Equação 3.2, nota-se que, quando  $M$  tende ao infinito, a média do ruído tende a zero e, portanto, o PE estimado tenderá à resposta evocada, ou seja,  $\lim_{M \rightarrow \infty} \hat{s}[n] = s[n]$ . A

promediação (média coerente) de cerca de 500 épocas permite a avaliação do PESS por um profissional bem treinado [21].

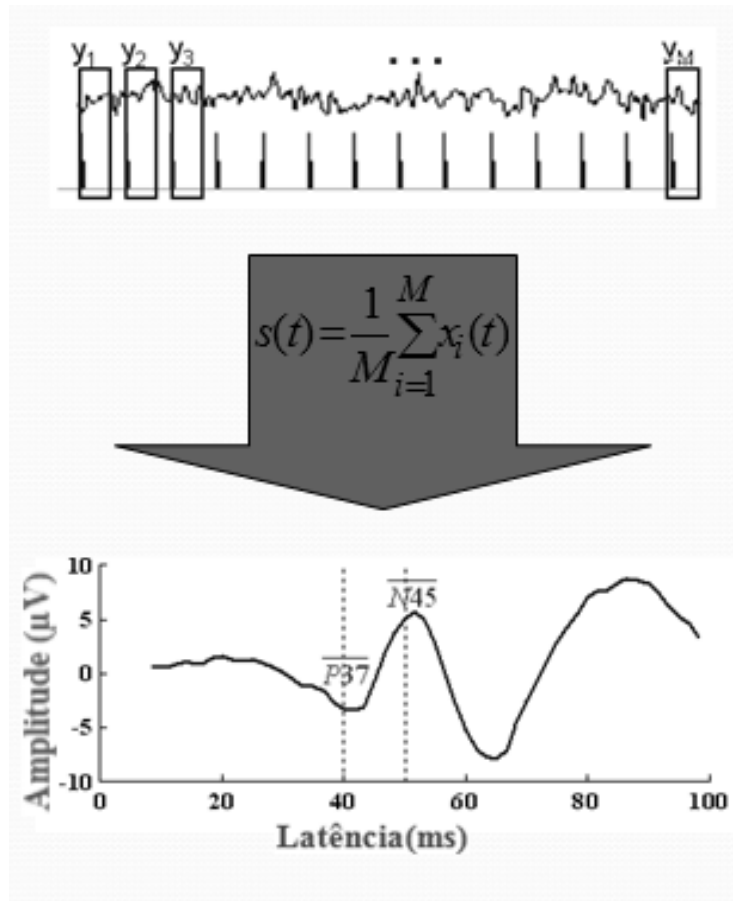


Figura 3.1 – Média Coerente

Fonte: Adaptado de [22]

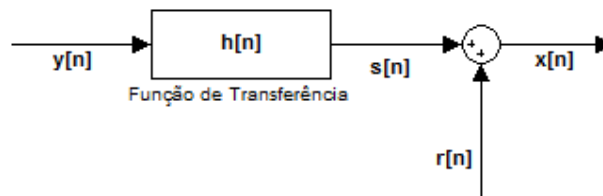


Figura 3.2 – Modelo Linear:  $y[n]$  representa o estímulo (trem de impulsos);  $h[n]$ , a função de transferência;  $s[n]$ , a resposta evocada;  $r[n]$ , o EEG espontâneo;  $x[n]$ , o EEG coletado

Fonte: Elaboração própria

A aplicação desta técnica faz com que o PESS estimado seja menos suscetível a razões sinal-ruído (RSR) baixas, quando comparado à monitorização direta do EEG, e permita a avaliação de estruturas subcorticais e corticais [13].

### 3.2 – Evolução Temporal do PESS

A monitorização da forma de onda do PESS se dá pelo acompanhamento das latências e amplitudes de seus potenciais de curta latência, em função de sua menor variabilidade em condições de normalidade (Seção 2.3). Assim, a ocorrência de elevadas variações durante a monitorização poderia indicar a iminência ou ocorrência de danos neurais durante uma cirurgia, por exemplo.

Dessa forma, a evolução temporal da forma de onda do PESS fundamenta-se na realização de  $N$  promediações através do descarte da época mais antiga (primeira época utilizada na estimativa anterior) e acréscimo da época atual (sucessora da última época da estimativa anterior no exame) acompanhados do armazenamento dos valores das amplitudes e dos instantes de ocorrência (latências) dos potenciais de interesse. No caso do presente trabalho, esses potenciais foram o vale P37 e o pico N45. A ilustração desse procedimento é apresentada na Figura 3.3 considerando  $N = 4$  estimações.

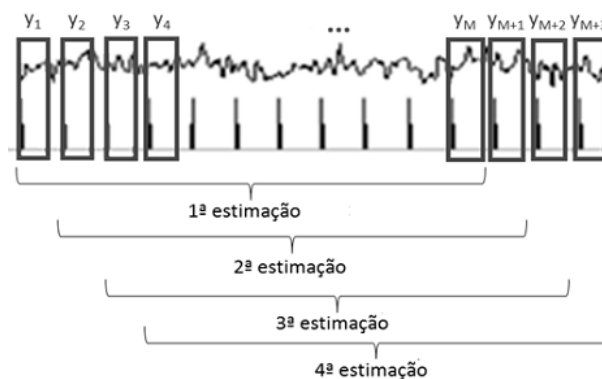


Figura 3.3 – Procedimento para avaliação da evolução temporal da forma de onda do PESS para  $N = 4$  estimações: a cada estimativa são extraídas as amplitudes e latências do P37 e N45 após realização da Média Coerente de  $M$  épocas

Fonte: Elaboração própria

### 3.3 – Técnicas de Detecção Objetiva da Resposta Evocada (ORD)

A análise morfológica, descrita nas seções anteriores, é uma técnica de avaliação da resposta evocada onde há significativa influência de fatores como experiência, habilidade e critérios informalmente estabelecidos pelo profissional envolvido. Dessa forma, a variabilidade entre observadores pode gerar resultados divergentes, uma vez que a inspeção é realizada visualmente.

A fim de mitigar esse problema, técnicas de detecção objetiva da resposta evocada (ORD, de *Objective Response Detecion*) vêm sendo propostas por diversos estudos. As técnicas ORD são baseadas em testes estatísticos para detecção de resposta, o que lhes confere desempenho frequentemente superior ao das técnicas convencionais, baseadas na observação humana. Essa característica se torna notória, em especial, na extração de informações relevantes em casos considerados duvidosos e na taxa máxima de falsos-positivos pré-estabelecida. Além disso, essas técnicas podem ser utilizadas no treinamento dos observadores humanos, reduzindo tendências para aqueles com critérios rígidos, relaxados ou flutuantes [10].

As técnicas ORD apresentam certa variedade de métodos, dentre os quais se destaca a Magnitude Quadrática da Coerência (MSC). Em 1989, DOBIE & WILSON propuseram sua aplicação a potenciais evocados auditivos. A MSC leva em consideração informações de potência e fase do sinal coletado para a identificação de frequências que contribuam para a resposta evocada. Dessa forma, esse método apresenta-se mais adequado para a análise do PE por conciliar potência e sincronismo entre a resposta cortical e o estímulo, permitindo melhor compreensão das atividades cerebrais envolvidas no processamento de informação sensorial.

Quando comparada a outros métodos, a MSC apresentou, em muitos casos, desempenho superior na avaliação tanto do PE transiente [34] quanto do PE em regime permanente [10]. Cabe ainda ressaltar a simplicidade computacional da MSC [10], o que viabiliza sua implementação em sistemas para uso *online*.

### 3.3.1 – Magnitude Quadrática da Coerência (MSC)

A Magnitude Quadrática da Coerência é caracterizada como a parcela do valor médio quadrático do EEG devida à estimulação, sendo definida matematicamente como o quadrado da função coerência complexa. Isto é, sendo  $y[n]$  uma função de entrada e  $x[n]$  a saída correspondente (Figura 3.2), o módulo ao quadrado da função de coerência é a razão entre o valor absoluto ao quadrado do espectro cruzado  $G_{xy}(f)$  e o produto dos autoespectros de  $y[n]$  e  $x[n]$ , respectivamente,  $G_{yy}(f)$  e  $G_{xx}(f)$ :

$$\gamma_{xy}^2(f) = \frac{|G_{xy}(f)|^2}{G_{xx}(f)G_{yy}(f)} \quad \text{Equação 3.3}$$

Logo, a MSC é uma função real e normalizada, variando de 0 a 1 [2].

Segundo [24], a estimativa de  $\hat{\gamma}_{xy}^2(f)$  para sinais discretos de duração finita pode ser calculada mediante a segmentação dos sinais de entrada e saída:

$$\hat{\gamma}_{xy}^2(f) = \frac{\left| \sum_{i=1}^M X_i(f)Y_i^*(f) \right|^2}{\sum_{i=1}^M |X_i(f)|^2 \sum_{i=1}^M |Y_i(f)|^2}, \quad \text{Equação 3.4}$$

onde  $X_i(f)$  e  $Y_i(f)$  são as Transformadas Discretas de Fourier (DFT) da  $i$ -ésima época dos sinais  $x[n]$  e  $y[n]$ , respectivamente,  $M$  é o número de épocas envolvidas no cálculo,  $\hat{\phantom{x}}$  representa a estimativa realizada e  $*$ , o complexo conjugado.

Assumindo-se uma estimulação periódica e que o sinal está segmentado de acordo com o período da estimulação, tem-se que  $Y_i(f)$  será idêntico em todas as épocas, o que permite redefinir a Equação 3.4, de modo simplificado como:

$$\hat{K}^2(f) = \frac{\left| \sum_{i=1}^M X_i(f) \right|^2}{M \sum_{i=1}^M |X_i(f)|^2}, \quad \text{Equação 3.5}$$

da qual fica evidente que a estimativa da MSC depende apenas de  $X_i(f)$ , ou seja, da saída do sistema mostrado na Figura 3.2.

Na Equação 3.5, o numerador é o módulo ao quadrado da soma complexa de  $M$  Transformadas de Fourier dos trechos de EEG sincronizados com os instantes de estimulação para cada frequência  $f$  – que contém informação de magnitude e fase (sincronismo) – e o denominador é  $M$  vezes a soma das magnitudes quadráticas correspondentes a cada  $f$  (normalização pela potência média). Assumindo que o sistema ao qual é aplicada a estimulação  $y[n]$  é linear e invariante no tempo, tem-se que caso haja resposta correspondente na frequência  $f$ , esta é a mesma para todas as  $M$  épocas, embora contaminada pelo EEG espontâneo (assumido gaussiano e de média zero). Assim, quando a potência do EEG de fundo tender a zero, a razão da Equação 3.5 se aproxima da unidade. Por outro lado, caso não haja resposta em uma dada frequência  $f$ ,  $X_i(f)$  não apresentará sincronismo com o estímulo, o que se reflete na distribuição aleatória de fases. Este fato faz com que o quociente expresso na Equação 3.5 tenda a zero, já que seu numerador será pequeno. Essa explicação está ilustrada na Figura 3.4, ao se utilizar o plano de Argand-Gauss.

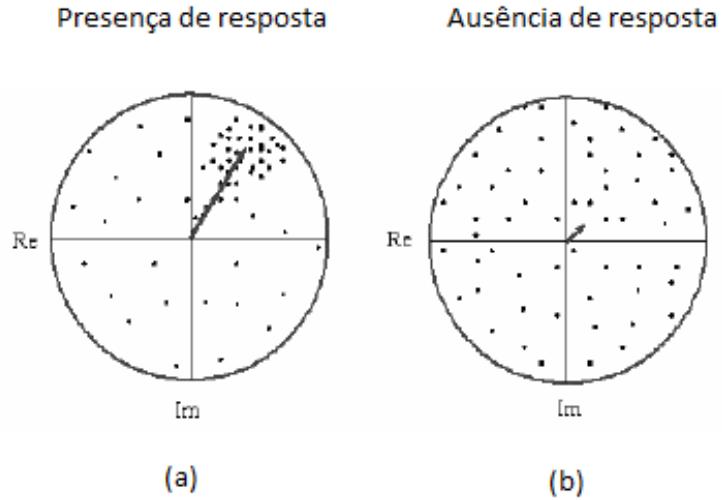


Figura 3.4 – Transformada de Fourier de  $Y_i(f)$ , representada pelos pontos, e a soma dos  $Y_i(f)$ , representada pela seta, no plano de Argand-Gauss: (a) Presença de resposta explícita pelo sincronismo de fase (b) Ausência de resposta pela falta de sincronismo de fase

Fonte: Modificada de [21].

Para a interpretação dos resultados provenientes do cálculo da MSC, estabeleceu-se a Hipótese Nula de Ausência de Resposta, ou seja, de que o valor verdadeiro da MSC,  $\kappa^2(f)$ , seria nulo. Para testar esta hipótese, estima-se um limiar com base no número de épocas  $M$  empregado e em um nível de significância  $\alpha$ , que pode ser expresso por:

$$\hat{K}_{crit}^2 = 1 - \alpha^{\frac{1}{M-1}} \quad \text{Equação 3.6}$$

O nível de significância representa a probabilidade de ocorrência de detecção mesmo não havendo resposta. Assim, espera-se a ocorrência de uma taxa máxima de falsos-positivos igual a  $\alpha$  em todas as frequências, mesmo na ausência de estimulação [21].

Dessa forma, as frequências para as quais os valores de  $\hat{\kappa}^2(f)$  estiverem abaixo desse limiar são aquelas para as quais se pode assumir que a Hipótese Nula é verdadeira, ou seja, que não há resposta à estimulação sensorial. Por outro lado, a

ultrapassagem do  $\hat{\kappa}_{crit}^2$  em certas frequências permite rejeitar a Hipótese Nula e, por simplicidade, assumir a presença de resposta.

### 3.3.2 – Considerações sobre a FFT (*Fast Fourier Transform*)

Conforme as equações 3.4 e 3.5, o cálculo da MSC envolve a obtenção da DFT (*Discrete Fourier Transform*) do EEG coletado. Para tal, pode-se utilizar a *Fast Fourier Transform* (FFT), um algoritmo criado em 1965 [8] que é muito eficiente computacionalmente se comparado com o implementado pela definição da DFT (*Discrete Fourier Transform*), expressa matematicamente por:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad \text{Equação 3.7}$$

onde  $0 \leq k \leq N-1$ ,  $W_N = e^{-j\frac{2\pi}{N}}$ ,  $N$  é o tamanho em amostras do sinal coletado,  $X[k]$  representa a DFT do sinal (domínio das frequências discretas) e  $x[n]$  representa o sinal amostrado (domínio do tempo discreto).

O algoritmo da FFT conhecido como radix-2 com decimação no tempo resulta em  $N \log_2 N$  operações de multiplicação e soma, enquanto que para a DFT têm-se  $N^2$  operações [9]. Isto se deve ao fato de o radix-2 com decimação no tempo ser efetuado por repetidas separações de somatórios em duas partes, a partir daquele descrito pela Equação 3.7.

A única restrição deste algoritmo é que o número de amostras obedeça à seguinte relação:

$$N = 2^p, \quad \text{Equação 3.8}$$

onde  $p \geq 0$ , ou seja, o número de amostras da janela deve ser uma potência de 2.



No caso de o número de amostras não obedecer a essa restrição, são acrescentados zeros (*zero-padding*), aumentando a resolução de amostragem do espectro, que é proporcional ao número total de amostras, ou seja, o *zero-padding* produz uma interpolação no domínio da frequência.

### **3.4 – Evolução Temporal da MSC**

A fim de investigar a ocorrência de resposta em determinadas frequências pode-se analisar a evolução temporal da MSC. De maneira análoga à evolução temporal da forma de onda do PESS (Seção 3.2), esta técnica consiste em realizar  $N$  estimativas da MSC pelo descarte da época mais antiga e inclusão da época atual, seguidas de armazenamento do valor da MSC para a frequência de interesse.

Cabe ressaltar que a diferença entre esta técnica e aquela exposta na Seção 3.2 é sua aplicação tanto no período estimulado como no não-estimulado, o que permite identificar momentos de transição entre estes períodos, mimetizando presença e ausência de resposta, respectivamente. Desta forma, o período não-estimulado deve estar segmentado em épocas de duração idêntica ao do estimulado para que, então, seja aplicado o cálculo da evolução temporal da MSC.

Com vistas a se obter melhor caracterização da resposta evocada, usualmente são avaliadas frequências localizadas na banda de máxima resposta [16]. Essa banda compreende as frequências em que há maior “responsividade” à estimulação somato-sensitiva e, portanto, maior probabilidade de se obter detecção na condição de estimulação. Resultados obtidos por [16] e [21] indicam que, para intensidade de estimulação correspondente ao limiar motor, pode-se considerar como banda de máxima resposta a faixa de 30 a 60 Hz. Essa banda corresponde aos potenciais de curta latência (P37 e N45) [16]. A Figura 3.5 ilustra a aplicação da MSC, sendo possível identificar a banda de máxima resposta.

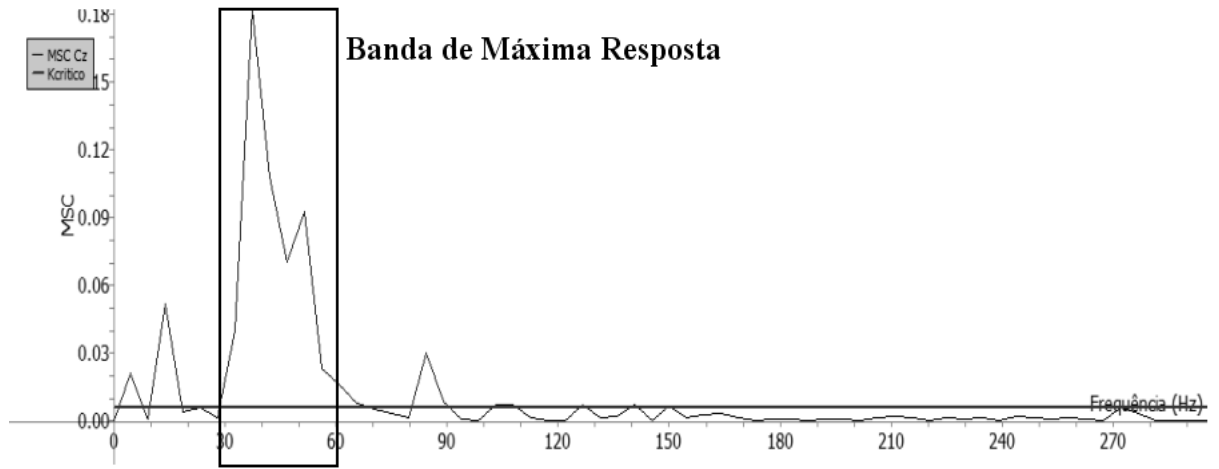


Figura 3.5 – Identificação da banda de máxima resposta (30-60 Hz)

Fonte: Elaboração própria

# Capítulo 4

## Materiais e Métodos

O desenvolvimento de um sistema multiplataforma que realiza processamento de sinais cria a necessidade de uso de ferramentas que permitam conjugar versatilidade e agilidade computacional.

Ademais, para avaliar as rotinas desenvolvidas para o sistema proposto, foram utilizados sinais simulados e sinais EEG (Sistema Internacional 10-20), espontâneo e durante a estimulação do nervo tibial direito de voluntários adultos. Neste capítulo, serão descritas a criação e a obtenção destes sinais.

Por fim, serão discutidas as etapas envolvidas no pré-processamento do sinal do exame de PESS, que envolvem a segmentação do exame em épocas sincronizadas com a estimulação e a mitigação de efeitos referentes aos artefatos (Seção 2.5). Esta última fase é de especial relevância para a melhoria da razão sinal-ruído (RSR) do PESS, influenciando diretamente tanto a análise por inspeção visual quanto a realizada por meio da MSC (Seção 3.3.1).

### 4.1 – Sinais Simulados

Foram gerados sinais mimetizando um exame de PESS: sinais senoidais (representando a resposta eliciada pelo estímulo) com ruído aditivo gaussiano e de média nula (simulando o EEG de fundo). Além disso, a simulação conteve trechos na condição de não-estimulação (com presença apenas do EEG espontâneo) de maneira a tornar-se mais próxima da realidade e permitir o teste de rotinas de sincronismo. Esses sinais foram armazenados em arquivo binário como dois canais distintos. Foi, ainda, armazenado um terceiro canal, representando a estimulação aplicada.

A Figura 4.1 apresenta o conjunto de sinais simulados em um trecho estimulado, uma senóide com RSR (razão entre potências) de 0,0152 (Canal 1) e uma senóide amortecida<sup>5</sup> com RSR de 4,0 (Canal 2). Essa apresentação foi possível mediante leitura do arquivo de teste e armazenamento em arquivo texto através da rotina desenvolvida em C++. Estas informações foram, então, empregadas para exibição gráfica com auxílio do MATLAB<sup>®</sup>.

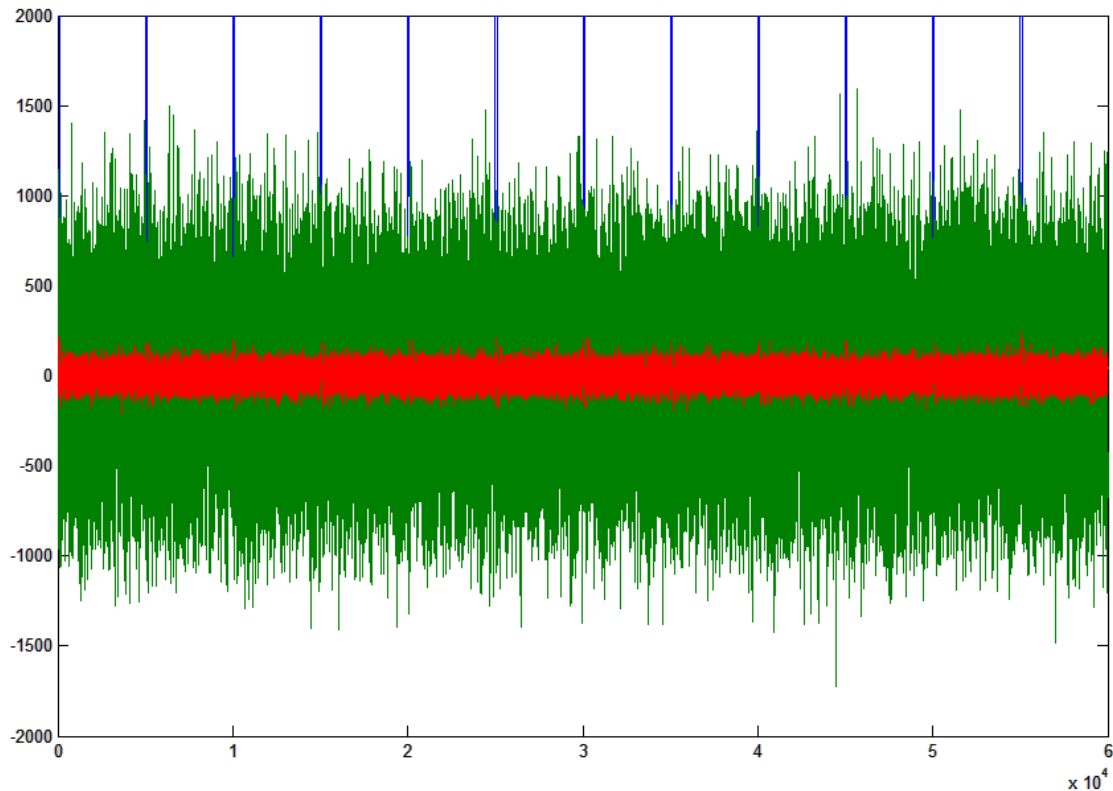


Figura 4.1 – Parte do sinal lido pela rotina de leitura desenvolvida em C++: Em azul está a estimulação; em verde, uma senóide com  $RSR = 0,0152$ ; em vermelho, uma senóide amortecida com  $RSR = 4,0$

Fonte: Elaboração própria

---

<sup>5</sup> Pela Figura não se pode notar a senóide amortecida, uma vez que a mesma é repetida para cada estímulo e a Figura 4.1 apresenta um registro de longa duração.

## 4.2 – Sinais EEG

Os sinais EEG utilizados foram coletados durante o doutorado de Danilo Melges [22] no Laboratório de Processamento de Imagens e Sinais (LAPIS) do Programa de Engenharia Biomédica (PEB/COPPE). Estes sinais foram coletados durante estimulação somato-sensitiva de 40 voluntários na faixa etária de 21 a 41 anos, sem histórico de neuropatias. O exame foi realizado com os voluntários relaxados e posicionados em decúbito dorsal (deitados de costas), a fim de evitar a contaminação do EEG por artefatos musculares. O comitê de ética local (CEP-HUCFF/UFRJ) aprovou esta pesquisa e todos os voluntários assinaram consentimento livre e esclarecido.

Os eletrodos de superfície foram utilizados para estimulação e registro com impedância inferior a 5 k $\Omega$ . A importância deste parâmetro se reflete diretamente na qualidade do exame, sendo o desbalanceamento de impedâncias responsável por boa parte dos registros de má qualidade [21]. A coleta foi realizada com posicionamento de eletrodos de acordo com o Sistema Internacional 10-20 (Figura 4.2), precedida pela extração do extrato córneo (camada superior da pele composta por células mortas). Os eletrodos de estimulação foram colocados sobre o nervo tibial direito (entre a borda medial do tendão de Aquiles e a borda posterior do maléolo medial - Figura 4.3), o que garantiu a geração de PESS de maiores amplitudes. Já o eletrodo de terra foi localizado na fossa poplíteia.

A estimulação foi realizada através do estimulador Atlantis Four ([www.emsamed.com.br](http://www.emsamed.com.br), EMSA, Brasil) e caracterizada por pulsos de corrente de 200  $\mu$ s de duração. A intensidade de estimulação foi definida como o limiar motor (menor intensidade capaz de produzir leve contração involuntária dos músculos intrínsecos do pé, produzindo flexão do hálux) de maneira a preservar o tecido e, ainda assim, verificar a presença da resposta evocada. A taxa de estimulação envolveu frequências nominais de 2, 5, 7 e 9 Hz, (valores reais: 2,0, 4,83, 6,71 e 8,55 Hz).



Figura 4.2 – Posicionamento de eletrodos

Fonte: [22]



Figura 4.3 – Aplicação de estímulo

Fonte: [22]

Os sinais coletados e amplificados foram submetidos a um filtro passa-faixas, com frequência de corte inferior de 0,1 Hz e superior de 100 Hz. Em seguida, estes sinais foram digitalizados (resolução de 16 bits) à taxa de 600 Hz com o amplificador EEG BNT-36 (EMSA, Brasil). Esta taxa de amostragem representa 3 vezes a taxa de Nyquist (200 Hz), o que evita efeitos de *aliasing*.

Embora tenha sido realizada a coleta de todas as derivações indicadas pelo Sistema Internacional 10-20, neste trabalho, somente foram utilizadas as derivações C3,

C4 e Cz. Isto porque estas derivações se encontram localizadas em região correspondente ao córtex somato-sensitivo primário (Figura 4.4), região na qual se espera obter respostas à estimulação sensorial somática com maior razão sinal-ruído.

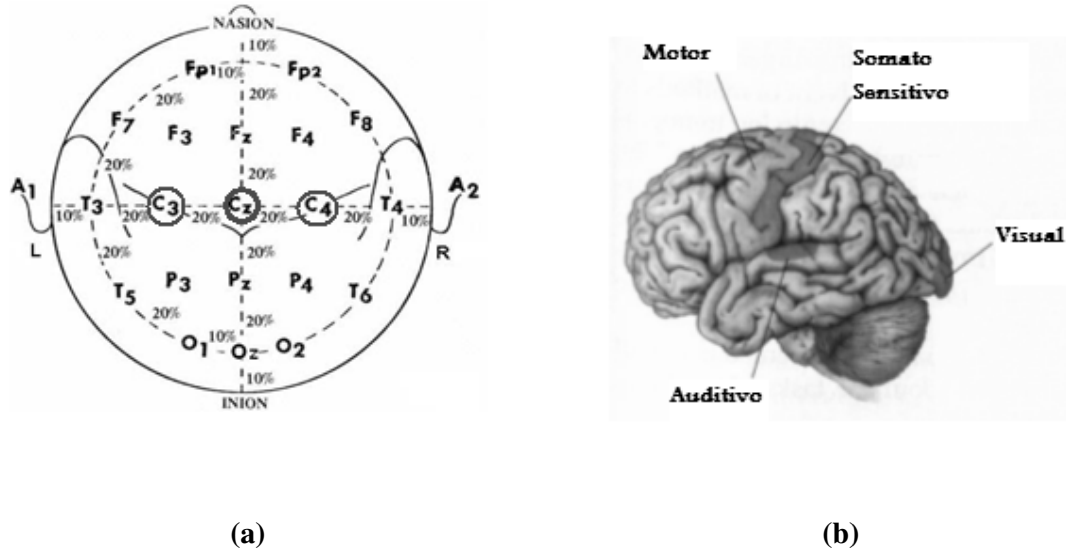


Figura 4.4 – (a) Derivação C3, C4 e Cz do Sistema Internacional 10-20 localizadas sobre o (b) córtex somato-sensitivo

Fonte: Adaptado de [17] e [4]

### 4.3 – Pré-processamento

A aplicação de técnicas de pré-processamento tem o intuito de tornar possível ou melhorar as etapas subsequentes de análise do sinal. Considerando que as etapas de processamento descritas no Capítulo 3 se baseiam em sinais janelados em épocas sincronizadas com a estimulação, fez-se necessária uma etapa responsável pela separação prévia desses trechos de sinal. A fim de manter a generalidade do sistema desenvolvido com relação à frequência de estímulo, a segmentação do sinal de exame em épocas foi realizada através da identificação dos instantes de estímulo. Assim, a obtenção da derivada do sinal contido no canal de estimulação (*trigger*) permite identificar esses instantes em que o valor da derivada é positivo, o que caracteriza a transição de subida do pulso de estímulo. Além disso, foi realizada a remoção do valor médio (nível DC) do sinal EEG. Adicionalmente, com o intuito de mitigar os efeitos dos

diversos tipos de artefatos (Seção 2.4) na interpretação dos resultados dos exames de PE, são aplicados dois procedimentos: rejeição automática de artefatos e janela Tukey.

### 4.3.1 – Janela Tukey

Como os componentes mais importantes do PESS (potenciais de curta latência) acontecem nos primeiros 100 ms pós estímulo [16] e a presença do artefato de estímulo se dá justamente dentro desse intervalo de tempo, faz-se necessária a minimização de seus efeitos. Dada sua característica impulsiva (elevada amplitude e banda larga), este artefato introduz distorções ainda mais agravantes no domínio da frequência e, portanto, na detecção da resposta realizada pela MSC. O simples descarte desses instantes iniciais resultaria em perda de resolução espectral, em relação ao uso das épocas completas.

Assim, com vistas a reduzir o efeito desse artefato na análise do PESS, sem perda de resolução espectral, é aplicada a janela Tukey. Essa janela é caracterizada pelos seguintes parâmetros (Figura 4.5):

- Tempo de subida ( $t_{\text{trans}}$ ): intervalo de tempo em que há transição de 0 para 1;
- Tempo de janela ( $t_{\text{janela}}$ ): intervalo de tempo total compreendido pela janela, incluindo  $t_{\text{trans}}$ .

O parâmetro  $t_{\text{zero}}$  se constitui do tempo fixo em zero (*zero padding*).



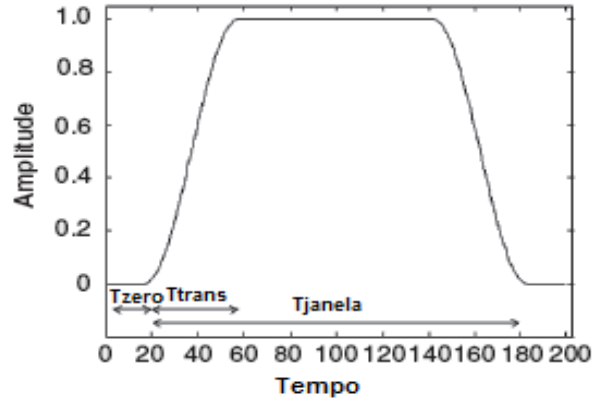


Figura 4.5 – Janela Tukey

Fonte: Adaptado de [16]

A janela Tukey é definida matematicamente por:

$$\omega[k] = \begin{cases} \frac{1}{2} \left[ 1 + \cos \left( \frac{2\pi}{r} \frac{k-1}{N-1} - \pi \right) \right] & k < \frac{r}{2}(N-1)+1 \\ 1 & \frac{r}{2}(N-1)+1 \leq k \leq N - \frac{r}{2}(N-1) \\ \frac{1}{2} \left[ 1 + \cos \left( \frac{2\pi}{r} - \frac{2\pi}{r} \frac{k-1}{N-1} - \pi \right) \right] & N - \frac{r}{2}(N-1) < k \end{cases} \quad \text{Equação 4.1}$$

onde  $k$  pertence ao intervalo de  $[1, N]$ ,  $N$  é o número de amostras no comprimento da janela e  $r$  é dado pela relação:

$$r = \frac{2t_{trans}}{t_{janela}} \quad \text{Equação 4.2}$$

### 4.3.2 – Rejeição Automática de Artefatos

A rejeição automática de artefatos consiste no descarte de épocas com reduzida razão sinal-ruído (RSR), isto é, com elevada variância, que poderiam comprometer a estimação do PESS (Média Coerente) e a detecção da resposta (MSC). Assim, este algoritmo consiste do caso particular de uma ponderação onde o peso dado à época ruidosa é nulo [10]. A fim de obter trechos de EEG livres de ruído, são rejeitadas as

épocas em que mais de 5% das amostras consecutivas ou 10% das amostras totais se encontram fora da faixa de  $\pm 3$  vezes o desvio padrão ( $\delta$ ) de 20 s de EEG espontâneo livre de ruído. Este limiar é justificado por conter 99,5% das amostras, assumindo-se que o EEG tem a amplitude normalmente distribuída.

A seleção do sinal de referência (20 s de EEG espontâneo livre de ruído) foi realizada através do Módulo Linha de Base do *software* desenvolvido em LabVIEW [21] para monitorização do PESS. Neste sistema, após a escolha do sinal de referência, foi calculado o desvio padrão correspondente, o qual foi armazenado em arquivo binário para posterior utilização na rejeição de artefatos. Na Figura 4.6, que ilustra este procedimento, se pode observar que o trecho entre os cursores verticais vermelhos está sendo selecionado.

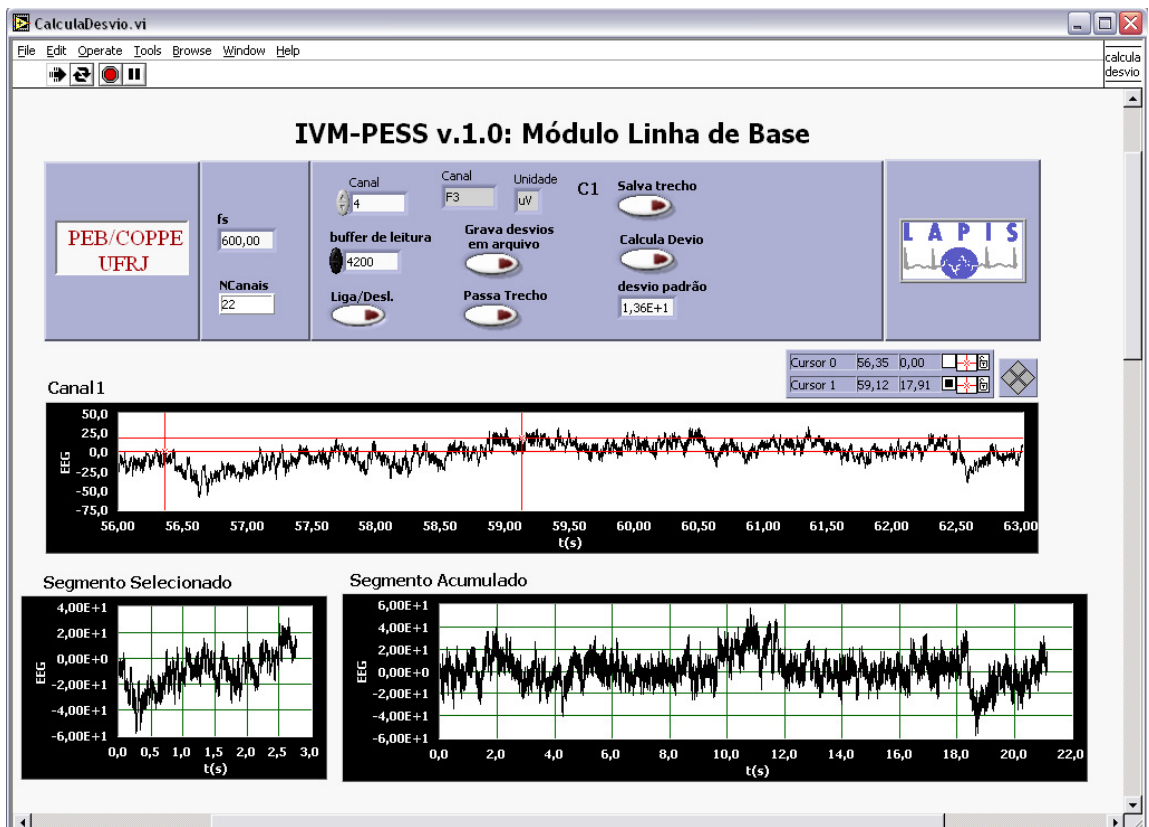


Figura 4.6 – Cálculo do desvio padrão ( $\delta$ ) por meio de *software* desenvolvido em LabVIEW [21]

Fonte: Elaboração própria

Embora seja eficiente na redução de efeitos de artefatos de origem externa, instrumental ou fisiológica, esta técnica se mostra ineficaz quando se trata de artefato de estímulo. Isso se deve ao fato de este artefato apresentar curta duração e, portanto, ser representado por poucas amostras. A frequência de estimulação de 4,83 Hz, por exemplo, corresponde a épocas de 207 ms. Dessa forma, o artefato de estímulo ocorrido em seus primeiros 5 ms significa um percentual equivalente a 2,4% de suas amostras, não atendendo aos critérios de descarte estabelecidos pela técnica de rejeição automática de artefatos. Por outro lado, se atendessemos a esses critérios, todas as épocas seriam descartadas, uma vez que esse artefato é síncrono com a estimulação. Daí a importância do janelamento Tukey apresentado anteriormente.

# Capítulo 5

## Avaliação do Sistema Multiplataforma para Análise do PESS

A avaliação do sistema desenvolvido é fundamental para garantir seu funcionamento correto em campo (na prática clínica ou na monitorização cirúrgica). Assim, para a realização de testes da ferramenta confeccionada, foram utilizados sinais simulados (Seção 4.1) e sinais EEG com e sem estimulação do nervo tibial direito (Seção 4.2).

O presente capítulo apresenta os resultados obtidos com a aplicação do sistema e analisa o seu desempenho. Cabe ressaltar que as rotinas implementadas estão disponíveis no Apêndice D (D.2 a D.13).

### 5.1 – Média Coerente

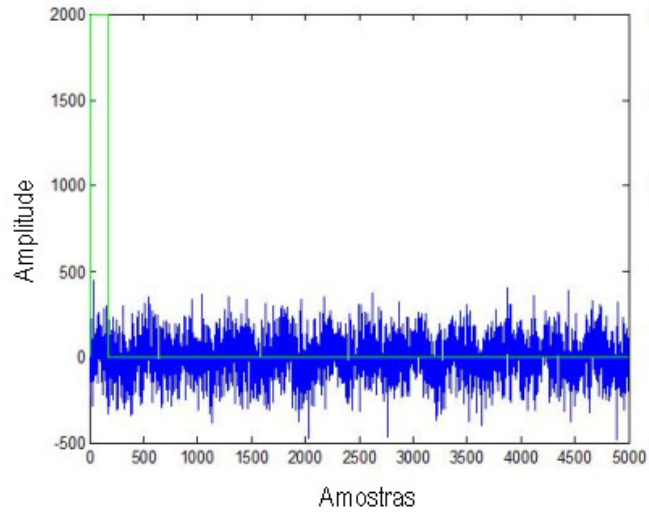
A rotina que implementa a Média Coerente, bem como a etapa de pré-processamento de detecção dos instantes de estimulação e segmentação do exame em épocas foi testada primeiramente com sinais simulados (Seção 4.1). Dessa forma, após a leitura dos sinais a partir do arquivo binário, a média coerente lhes foi aplicada.

Mediante observação das Figuras 5.1 e 5.2, pode-se verificar o resultado da aplicação da Média Coerente à senóide e à senóide amortecida de 20 Hz, respectivamente, descritas na Seção 4.1. Estes sinais foram previamente segmentados e promediados para diferentes números de épocas  $M$ . Nota-se claramente que o aumento do número de épocas leva a um aumento da razão sinal-ruído no sinal promediado, conforme previsto pela teoria (Seção 3.1). Isso se deve ao fato de a potência do ruído tender a zero com o aumento da quantidade de épocas. As rotinas de teste geraram

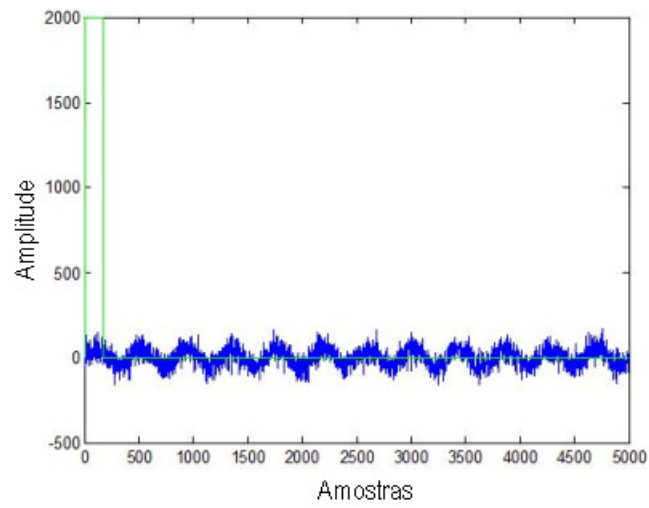
arquivos de texto com resultados que foram lidos no MATLAB<sup>®</sup> para apresentação gráfica.

Após a verificação dos resultados com os sinais simulados, esta rotina foi submetida a testes com sinais EEG coletados em laboratório (Seção 4.2). A Figura 5.3 apresenta a média coerente ( $M = 500$  épocas) das derivações C3, C4 e Cz do voluntário #1. Pode-se observar a formação do vale em torno de 37 ms correspondente ao P37 e do pico em torno de 45 ms correspondente ao N45. Estes gráficos foram obtidos por meio da própria interface gráfica do sistema desenvolvido (neste caso, não foram empregadas as técnicas de pré-processamento para redução de efeitos de artefatos). Adicionalmente, é possível comparar as respostas corticais de derivações distintas.

(a)



(b)



(c)

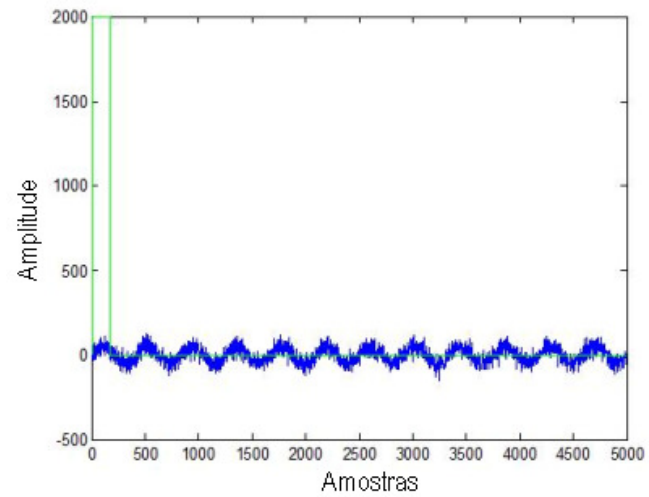


Figura 5.1 – Senóide de 20 Hz com aplicação da Média Coerente (em verde, estimulação; em azul, sinal promediado): (a)  $M = 10$  épocas (b)  $M = 100$  épocas (c)  $M = 200$  épocas

Fonte: Elaboração própria

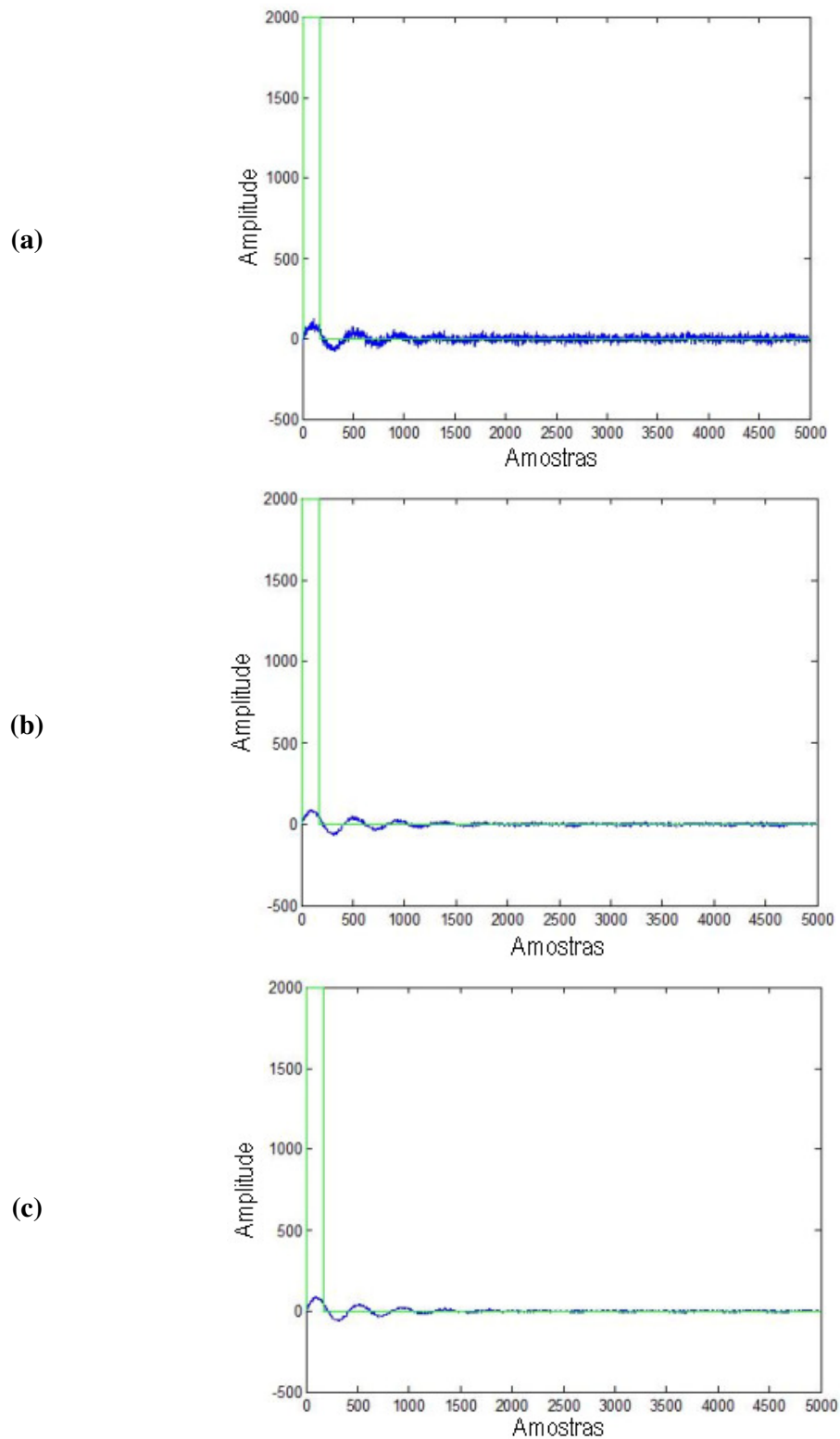


Figura 5.2 – Senóide amortecida de 20 Hz com aplicação da Média Coerente (em verde, estimulação; em azul, sinal promediado): (a)  $M = 10$  épocas (b)  $M = 100$  épocas (c)  $M = 200$  épocas

Fonte: Elaboração própria

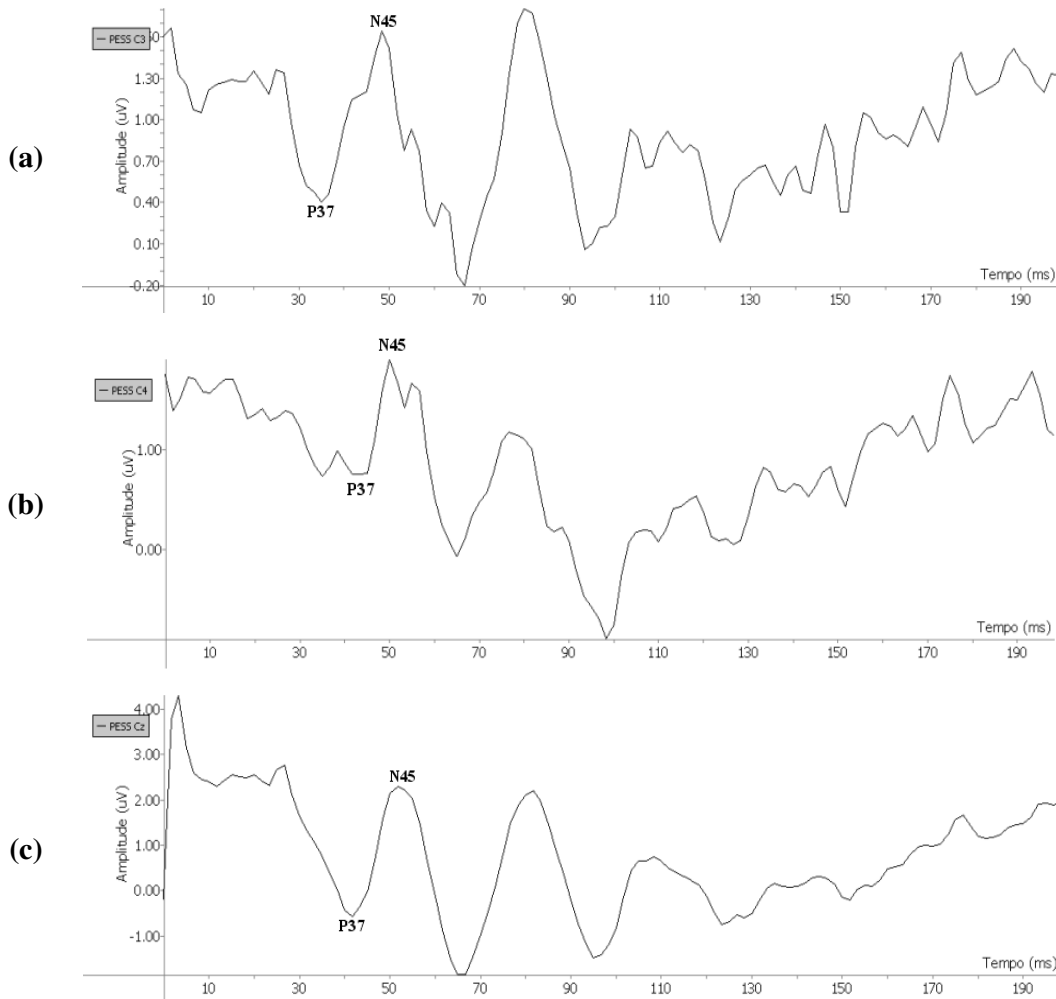


Figura 5.3 – Estimativa do PESS por Média Coerente ( $M = 500$  épocas) para o Voluntário #1 estimulado a  $11 \text{ mA}$ <sup>6</sup> com frequência de estimulação nominal de 5 Hz: (a) Derivação C3 (b) Derivação C4 (c) Derivação Cz

Fonte: Elaboração própria

## 5.2 – Rejeição de Artefatos

Esta técnica tem o intuito de atenuar efeitos de artefatos externos, fisiológicos e instrumentais, como exposto na Seção 4.3.1. O resultado da aplicação deste método e o consequente descarte de épocas ruidosas pode ser observado na Figura 5.4 que apresenta a média coerente ( $M = 1000$  épocas) para o voluntário #2, sem (Figura 5.4a) e

<sup>6</sup> Os valores referentes ao limiar motor estão em função da amplitude do pulso de corrente usado na estimulação.



com rejeição (Figura 5.4b). Nota-se um “alisamento” do sinal. Dessa forma é possível verificar mais facilmente as amplitudes e latências em que ocorrem os potenciais de curta latência do PESS.

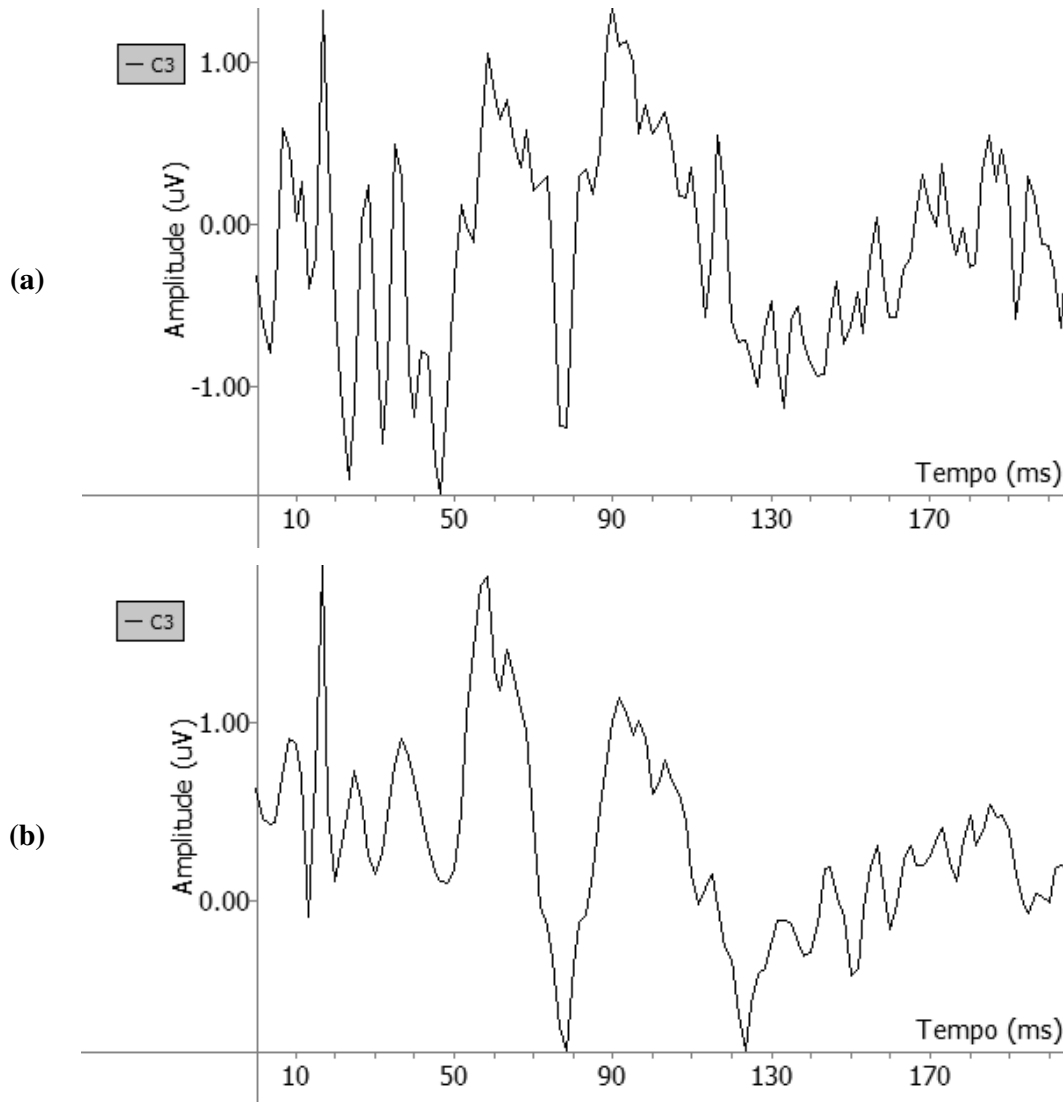


Figura 5.4 – Efeito da Rejeição Automática de Artefatos na Média Coerente do PESS para derivação C3 do Voluntário #2 estimulado a 11 mA com frequência de estimulação nominal de 5 Hz ( $M = 1000$  épocas): (a) Antes e (b) depois da aplicação da rejeição de artefatos

Fonte: Elaboração própria

### 5.3 – Remoção do Valor Médio

Conforme o exposto na Seção 4.3, esta função tem o intuito de remover o nível DC (médio) do sinal antes da aplicação da FFT de modo a facilitar a identificação de todas as componentes de frequência que compõem o PESS. Embora se tenha aplicado um filtro passa-faixa ao sinal EEG e, portanto, este sinal apresente valor médio nulo, ao segmentá-lo, os trechos podem ter valor médio diferente de zero e variando entre si. O teste consistiu na aplicação desta rotina a uma senóide com nível DC. Os resultados da rotina de teste foram registrados em um arquivo de texto e, então, lidos com o MATLAB<sup>®</sup> para visualização gráfica. A Figura 5.5 permite observar o funcionamento adequado da rotina na retirada do nível médio.

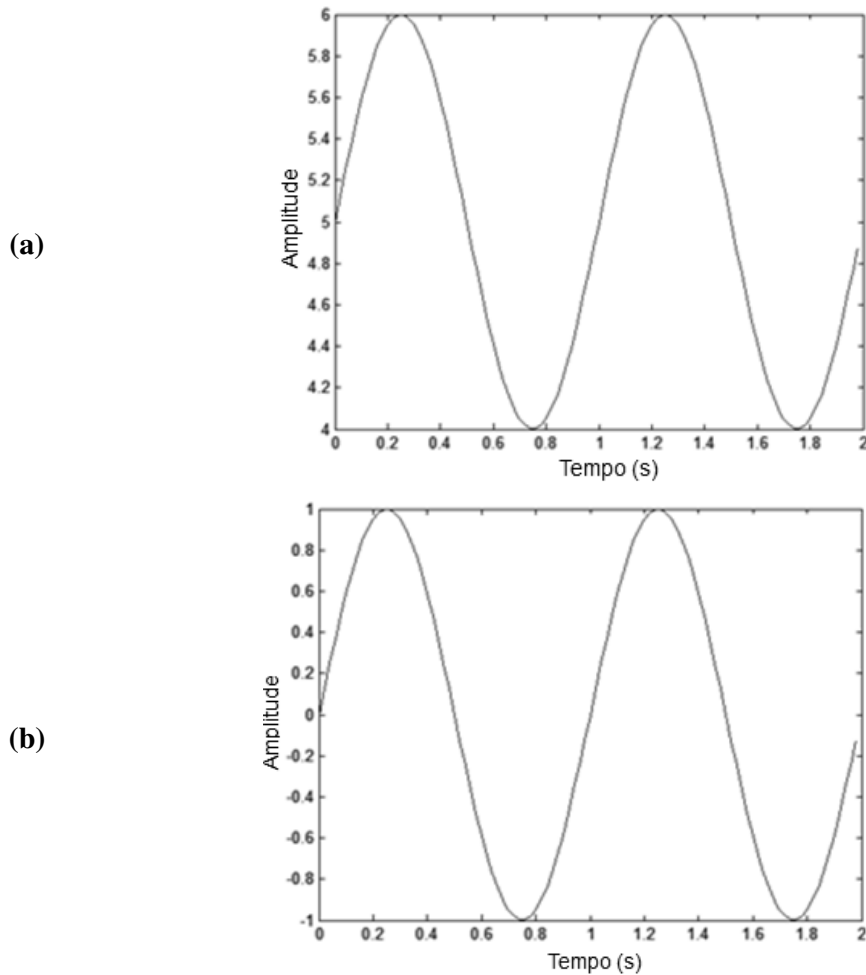


Figura 5.5 – Remoção de nível médio de senóide de 1 Hz: (a) com nível DC (b) sem nível DC

Fonte: Elaboração própria

Os resultados da aplicação desta rotina ao sinal coletado e promediado são expostos na Figura 5.6.

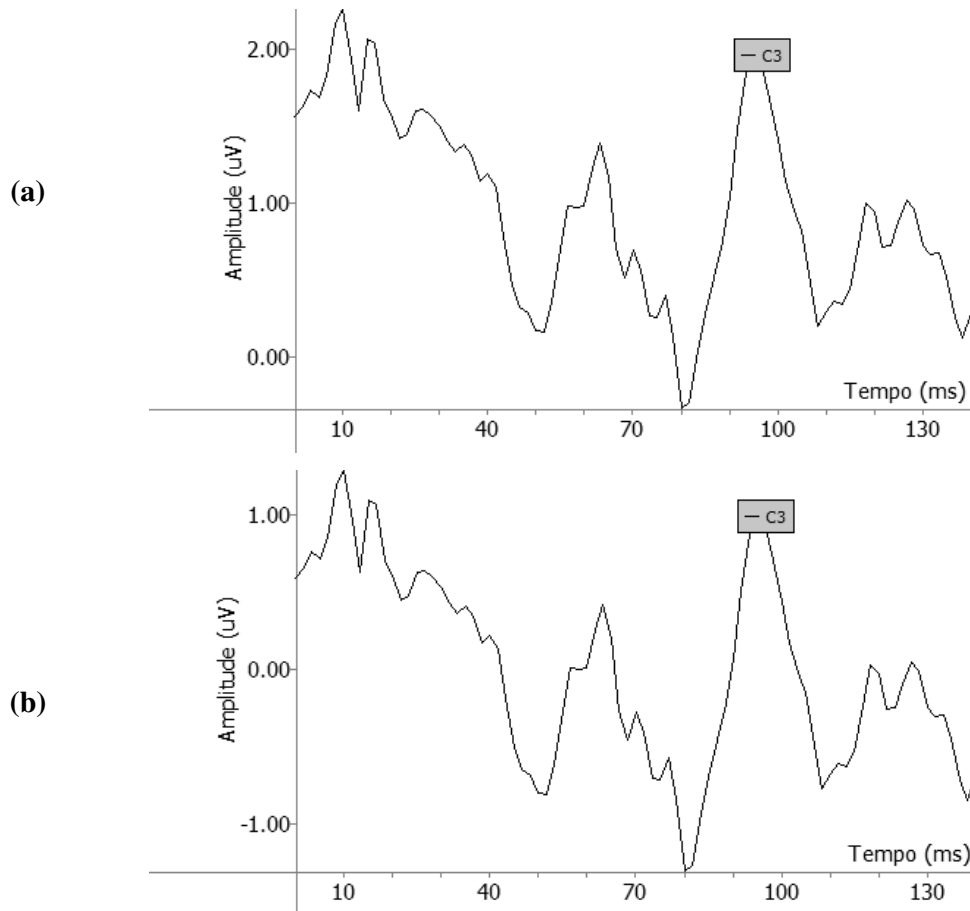


Figura 5.6 – Remoção de nível médio da média coerente: (a) com nível DC (b) sem nível DC

Fonte: Elaboração própria

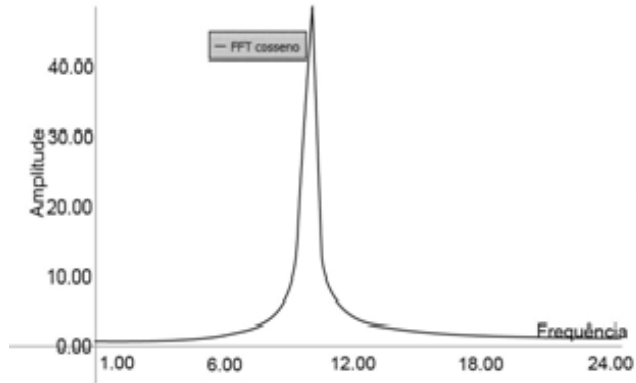
Cabe ressaltar que a remoção do valor médio foi aplicada a cada época antes do janelamento Tukey (Seção 4.3.1), caso contrário poderia introduzir componentes de alta frequência na estimativa espectral e, portanto, na MSC.

## 5.4 – Magnitude Quadrática da Coerência

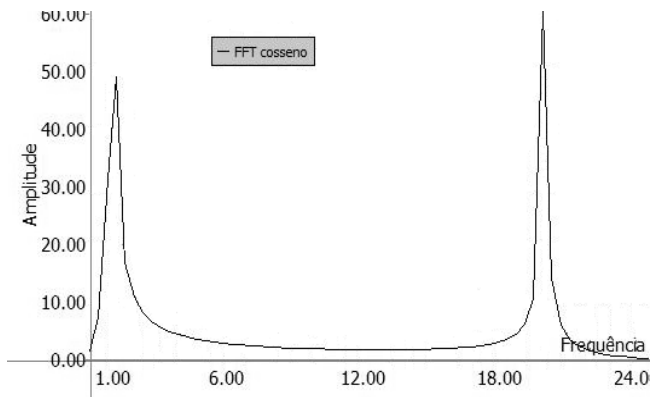
A avaliação da rotina de MSC foi realizada em duas etapas: apreciação do funcionamento da FFT através de sinais simulados e da MSC com sinais EEG. A primeira etapa se fez necessária por se tratar de uma função fundamental para a realização das etapas de cálculo seguintes. Assim, utilizando-se uma rotina de teste, foram gerados sinais contendo senóides, senóides sobrepostas (adição) com e sem ruído branco gaussiano aditivo de média zero. A FFT foi aplicada a estes sinais para fins de teste, conforme ilustrado na Figura 5.7, respectivamente, para uma senóide de 10 Hz, a soma de senóides de 1 e 20 Hz e senóide de 20 Hz corrompida por ruído, sendo possível verificar o funcionamento correto da função. Neste trabalho, a FFT foi implementada conforme algoritmo descrito nas *Numerical Recipes* (<http://www.nr.com/>).

A rejeição da Hipótese Nula de Ausência de Resposta em qualquer frequência acontece quando o valor da MSC para dada frequência ultrapassa o limiar estabelecido como valor crítico, conforme exposto na Seção 3.3.1. As Figuras 5.8 e 5.9 ilustram este procedimento para a detecção de resposta por meio da MSC. Para  $M = 50$  épocas, houve detecção principalmente na banda de 40 a 50 Hz para C3 (Figura 5.8a), de 35 a 50 Hz para C4 (Figura 5.8b) e de 30 a 50 Hz para Cz (Figura 5.8c). Analogamente, para  $M = 500$  épocas, a detecção aconteceu para as bandas de 30 a 60 Hz para C4 (Figura 5.9b) e Cz (Figura 5.9c) e de 20-50 para C3 (Figura 5.9a).

(a)



(b)



(c)

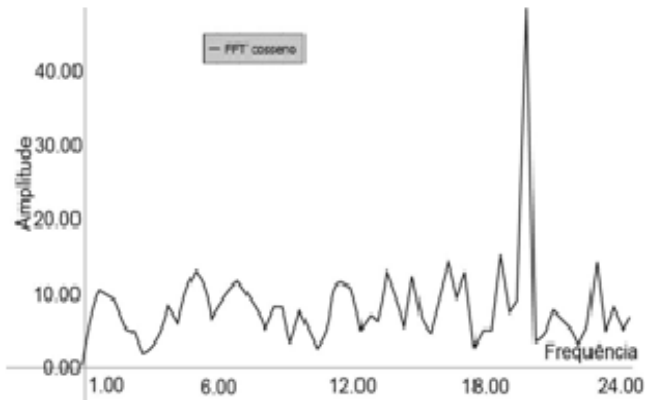


Figura 5.7 – FFT obtida através de rotina desenvolvida: (a) Senóide de 10 Hz  
(b) Senóides de 1 e 20 Hz somadas (c) Senóide de 20 Hz com ruído aditivo

Fonte: Elaboração própria

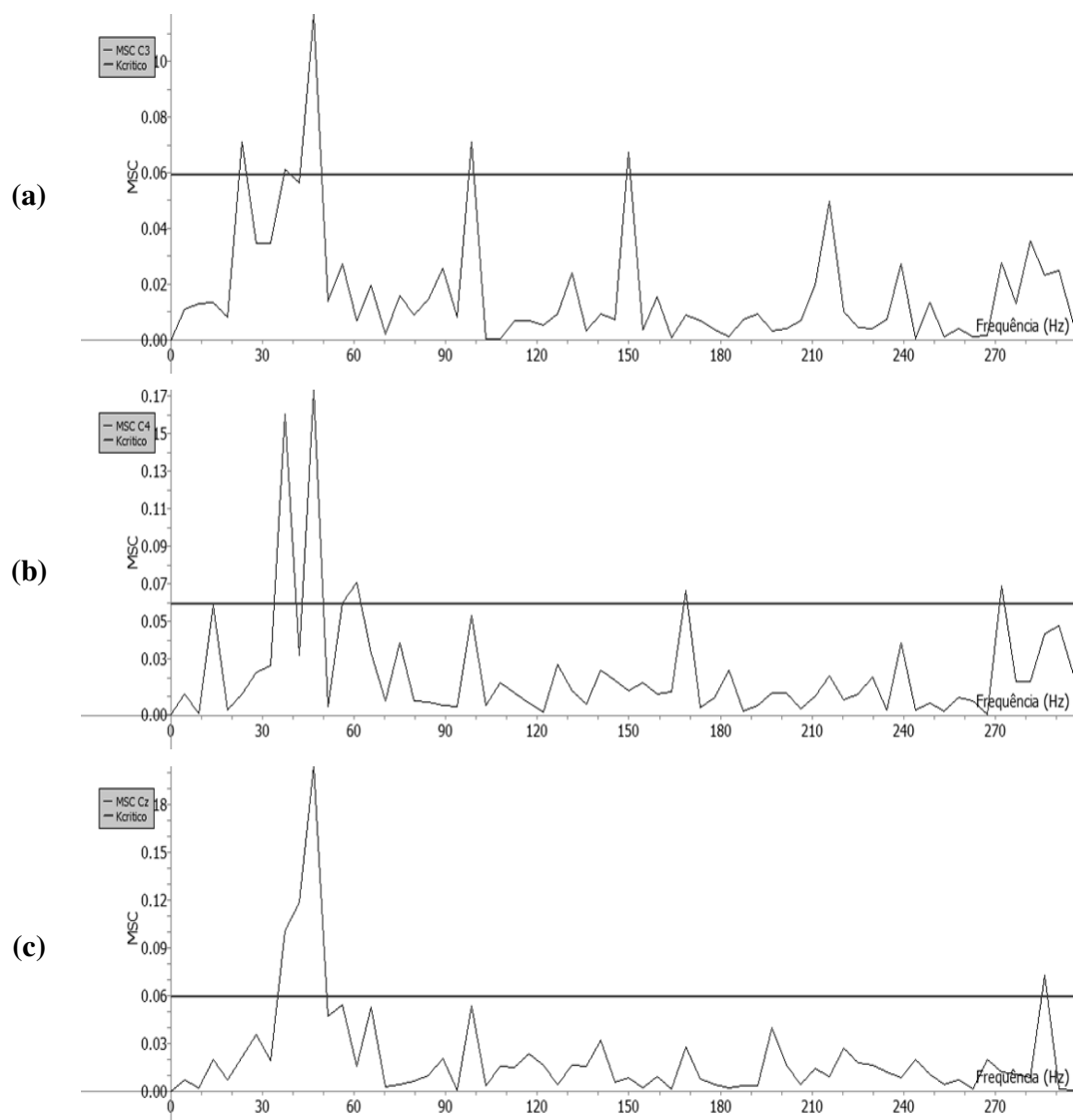


Figura 5.8 – MSC da derivação (a) C3 (b) C4 (c) Cz do Voluntário #3 estimulado a 10 mA com frequência de estimulação nominal de 5 Hz ( $M = 50$  épocas e  $\alpha = 0,05$ ;  $\hat{\kappa}_{crit} = 0,06$ )

Fonte: Elaboração própria

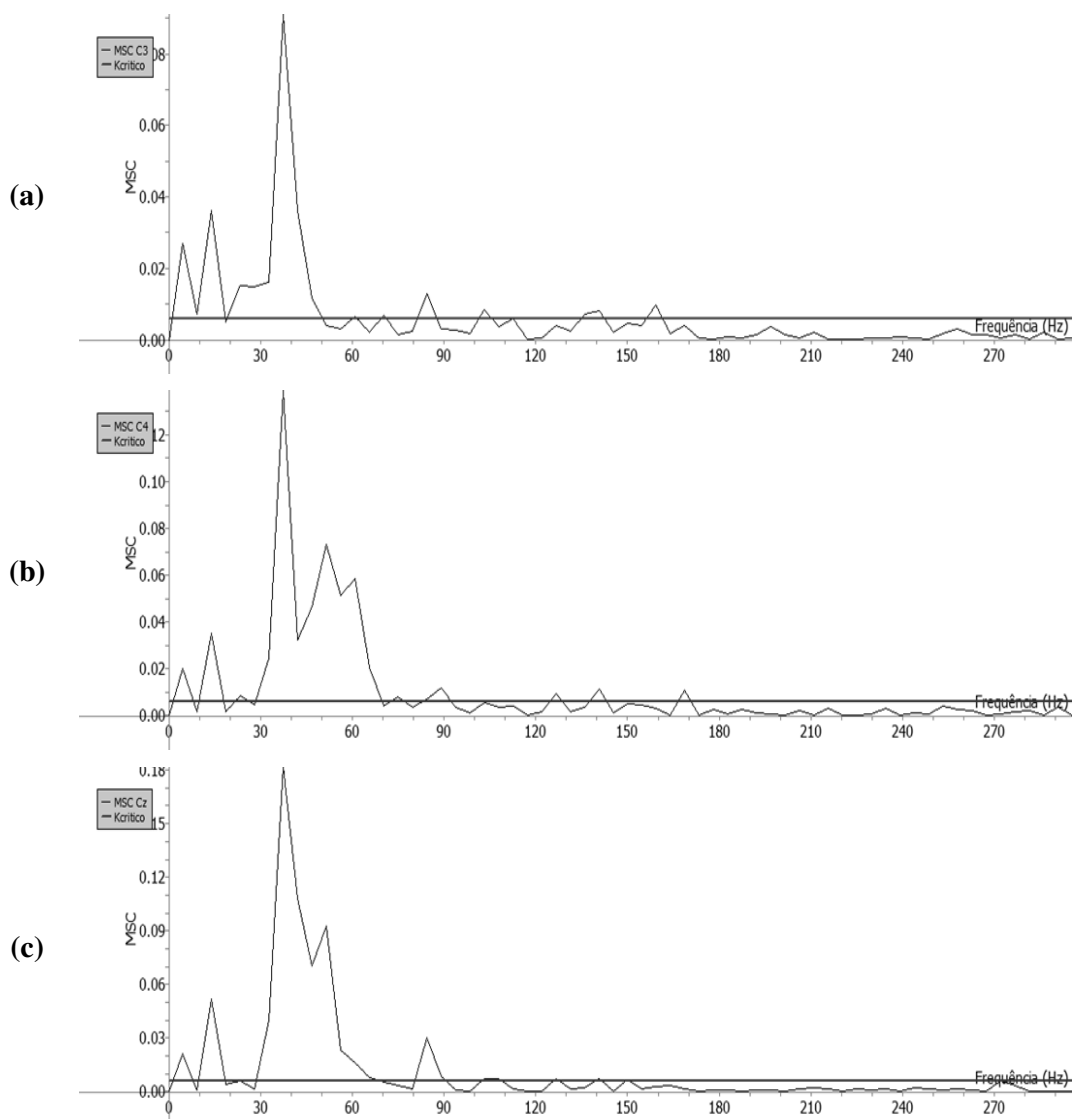


Figura 5.9 – MSC da derivação (a) C3 (b) C4 (c) Cz do Voluntário #3 estimulado a 10 mA com frequência de estimulação nominal de 5 Hz ( $M = 500$  épocas e  $\alpha = 0,05$ ;  $\hat{\kappa}_{crit} = 0,006$ )

Fonte: Elaboração própria

Das Figuras 5.8 e 5.9, nota-se que o desempenho da MSC, mesmo após a aplicação de apenas  $M = 50$  estímulos, apresenta resultados relevantes para a detecção de resposta. Este aspecto torna interessante a utilização desta técnica, em especial no monitoramento cirúrgico, por implicar reduzido intervalo de tempo para a detecção da resposta evocada.

## 5.5 – Janela Tukey

Conforme discutido na Seção 4.3.2, o janelamento Tukey é aplicado com o intuito de reduzir o efeito do artefato de estímulo, que, em função de sua característica impulsiva, apresenta consideráveis efeitos na MSC, dado que esta é aplicada no domínio da frequência.

A Figura 5.10 apresenta a Média Coerente e a MSC para a derivação C3 do voluntário #4, sendo possível observar o efeito deste artefato. Verifica-se que a presença do artefato de estímulo leva à detecção em praticamente toda a banda de frequências, devido a este artefato ser de banda larga. Logo, a presença deste prejudica a análise no domínio da frequência, dificultando a identificação da banda na qual ocorre de fato a resposta. Além disso, é fácil notar que a elevada amplitude do artefato de estímulo dificulta a visualização da forma de onda do PESS que, devido à “autoescala” do gráfico, apresenta os potenciais de curta latência proporcionalmente menores.

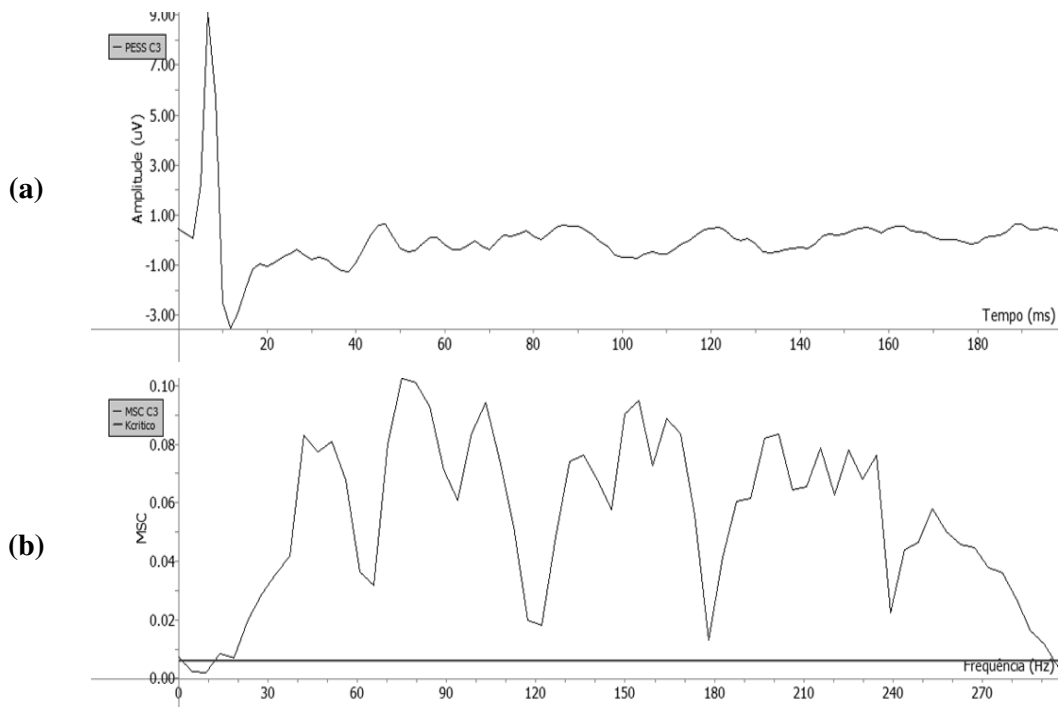


Figura 5.10 – (a) PESS e (b) MSC da derivação C3 para o Voluntário #4 estimulado a 10 mA com frequência de estimulação nominal de 5Hz ( $M = 500$  épocas e  $\alpha = 0,05$ ;  $\hat{k}_{crit} = 0,006$ )

Fonte: Elaboração própria



A aplicação da janela Tukey resulta em redução do efeito do artefato (Figura 5.11) e, portanto, melhora a estimativa da MSC (Figura 5.11b), permitindo identificar as bandas em que efetivamente há resposta. Ademais, é possível visualizar melhor a forma de onda e as características de interesse do PESS (Figura 5.11a). Os parâmetros da janela utilizada encontram-se na Tabela 5.1.

Parâmetro	Valor (ms)
Tamanho da janela	200
Tempo de transição	7
Tempo em zero	15

Tabela 5.1 – Parâmetros da janela Tukey utilizada

Fonte: Elaboração própria

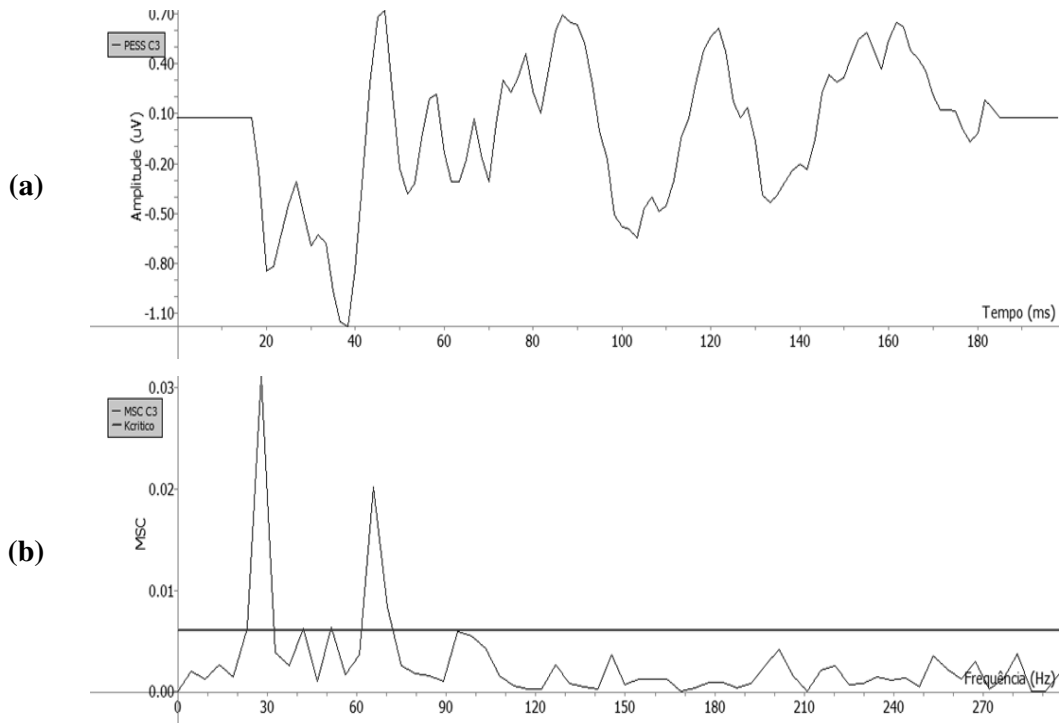


Figura 5.11 – (a) PESS e (b) MSC da derivação C3 para o Voluntário #4 estimulado a 10 mA com frequência de estimulação nominal de 5 Hz ( $M = 500$  épocas e  $\alpha = 0,05$ ;  $\hat{\kappa}_{crit} = 0,006$ )

Fonte: Elaboração própria

## 5.6 – Evolução Temporal do PESS

A monitorização da evolução temporal de características de interesse do PESS, tais como P37 e N45, é relevante para a verificação de condições de normalidade, dado que variações em amplitude ou latência podem ser indicativas de danos ao sistema nervoso (Seção 2.4).

As Figuras 5.12 e 5.13 ilustram a monitorização das latências e amplitudes de P37 e N45, tanto para  $M = 50$ , quanto para  $M = 500$  épocas. Nota-se que as latências apresentam menor variabilidade ao longo do tempo do que as amplitudes. A importância deste tipo de análise pode ser evidenciada pelo trabalho de [19], que ao investigar a hipotermia durante cirurgia, indicou haver maior correlação entre latência e temperatura.

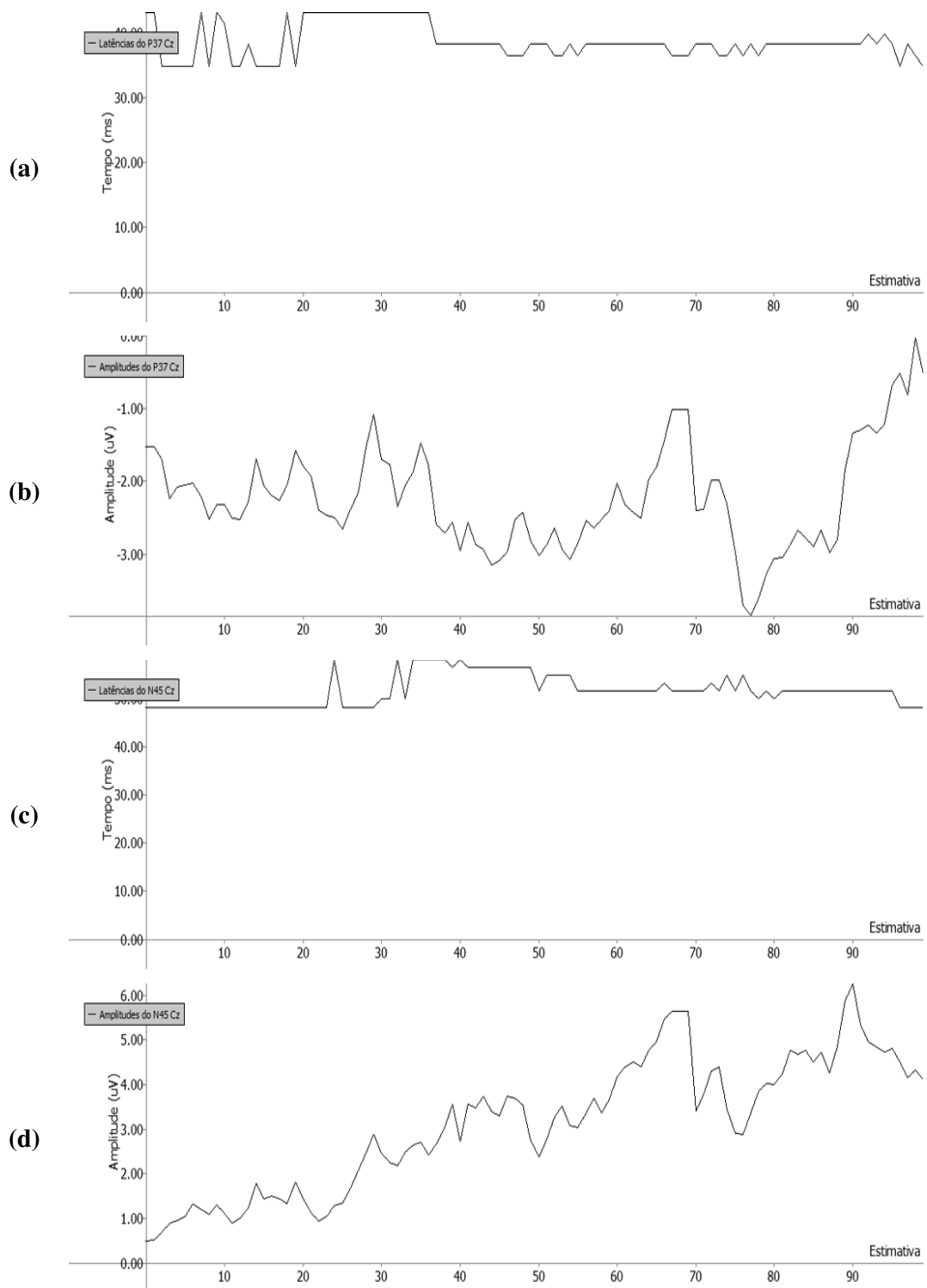


Figura 5.12 – Evolução Temporal do PESS da derivação Cz para o Voluntário #5 estimulado a 11 mA com frequência de estimulação nominal de 5 Hz ( $M = 50$  épocas por estimativa): (a) Latência e (b) Amplitude do P37 (c) Latência e (d) Amplitude do N45

Fonte: Elaboração própria

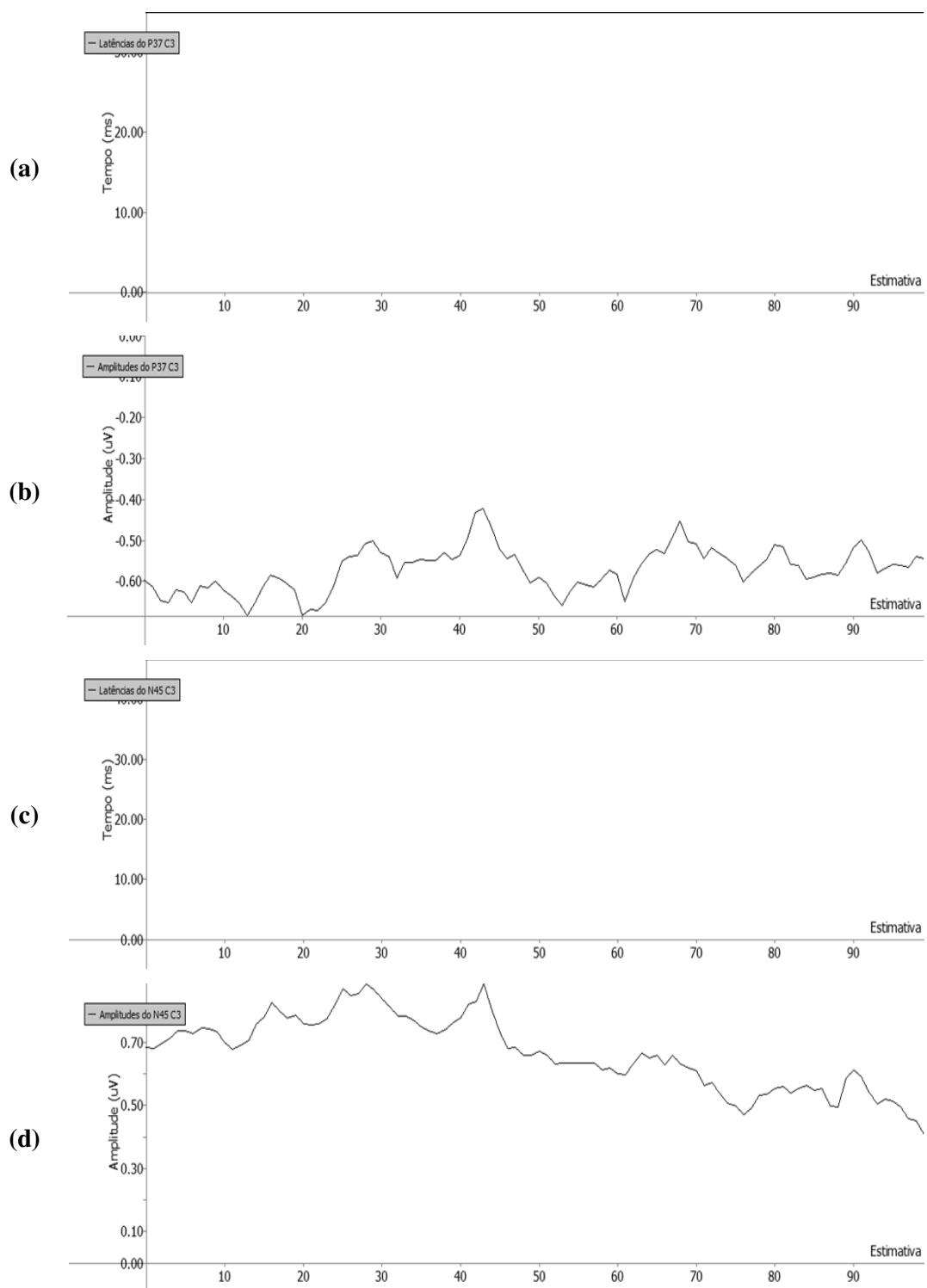


Figura 5.13 – Evolução Temporal do PESS da derivação Cz para o Voluntário #5 estimulado a 11 mA com frequência de estimulação nominal de 5 Hz ( $M = 500$  épocas por estimativa): (a) Latência e (b) Amplitude do P37 (c) Latência e (d) Amplitude do N45

Fonte: Elaboração própria

Dos exemplos anteriores, nota-se que mesmo para estimativas envolvendo poucas épocas, obtém-se conformação do PESS por meio da Média Coerente. Isso pode ser verificado observando-se que as latências e amplitudes dos potenciais P37 e N45 variam em torno do valor previsto na teoria (Figura 5.12), mesmo para estimativas calculadas com 50 épocas (10 s de exame). Este aspecto é ainda mais relevante quando se leva em consideração a monitorização durante cirurgias, em que o tempo para detecção da iminência ou ocorrência de danos neurológicos é um fator crítico.

## 5.7 – Evolução Temporal da MSC

A evolução temporal da MSC foi avaliada para frequências dentro da banda de máxima resposta (Seção 3.4). Exemplos da evolução temporal da MSC são mostrados na Figura 5.14 para  $M = 50$ , 100 e 200 épocas na frequência de 35 Hz (nominal), na qual as retas verticais pontilhadas indicam os instantes de início e término da estimulação. A frequência de 35 Hz foi escolhida por pertencer à banda de máxima resposta (30-60 Hz) para o PESS (Figura 5.15).

Ainda na Figura 5.14, observa-se que menores valores de  $M$  implicam maior variabilidade da MSC, conforme descrito por [21]. Todavia, estes valores também apresentam menores atrasos para a detecção de resposta em relação ao instante em que foi iniciada a aplicação de estímulo. Utilizando-se  $M = 50$  épocas, tem-se um atraso na detecção menor que os atrasos com  $M = 100$  épocas e 200 épocas, que foram, respectivamente, de 14 s e 24 s<sup>7</sup>.

Conforme discutido na Seção 5.6, o atraso na detecção de resposta é um aspecto crucial na aplicação das técnicas de evolução temporal em campo (monitorização intra e pós-cirúrgica).

---

<sup>7</sup> Este cálculo foi feito tomando por base a sobreposição entre estimativas. A diferença de tempo entre cada estimativa é igual a uma época (no caso, 200 ms). Assim, um atraso na detecção de aproximadamente 70 estimativas,  $M = 100$  épocas, equivaleria a 14 s. Analogamente, para  $M = 200$  épocas, um atraso de 120 estimativas corresponderia a 24 s.

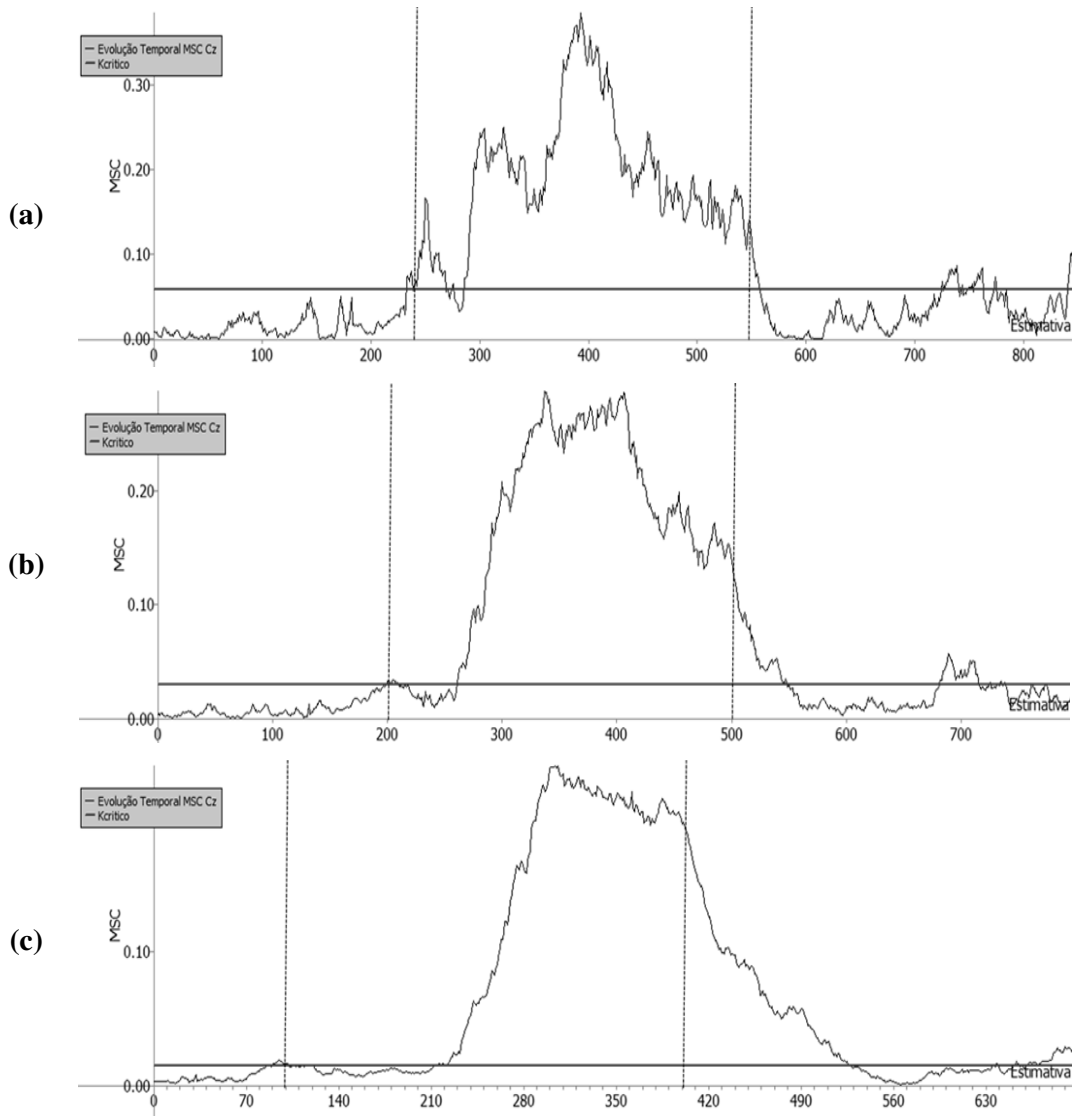


Figura 5.14 – Evolução Temporal da MSC da derivação Cz para o Voluntário #6 estimulado a 11 mA com frequência de estimulação nominal de 5Hz ( $\alpha=0,05$ ): (a)  $M = 50$  épocas (1 estimativa = 10 s de exame) (b)  $M = 100$  épocas (1 estimativa = 20 s de exame) (c)  $M = 200$  épocas (1 estimativa = 40 s de exame)

Fonte: Elaboração própria

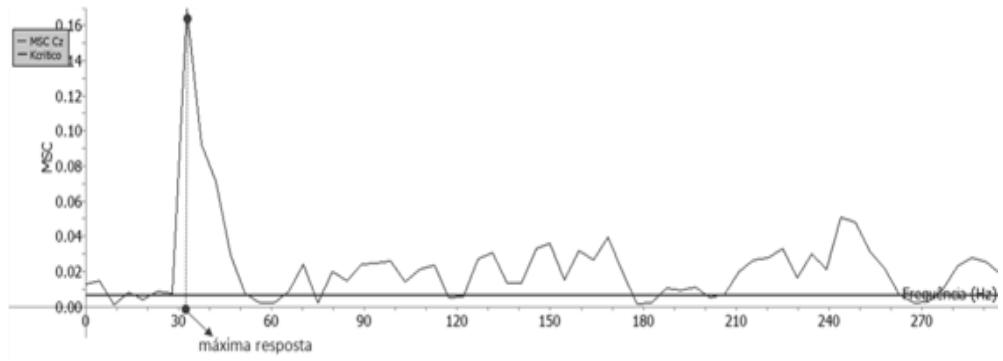


Figura 5.15 – MSC da derivação Cz para o Voluntário #6 estimulado a 11 mA com frequência de estimulação nominal de 5Hz ( $M = 500$  épocas,  $\alpha=0,05$  e  $\hat{\kappa}_{crit} = 0,006$ )

Fonte: Elaboração própria

# Capítulo 6

## Interface com o usuário

O sistema desenvolvido prevê sua utilização tanto por profissionais da área de Ciências Exatas quanto das Ciências da Saúde. Portanto, faz parte de seus requisitos, dentre outras características, a facilidade de utilização. Com este objetivo, foi desenvolvida uma interface empregando a biblioteca gráfica wxWidgets, ferramenta livre e multiplataforma, o que provê também flexibilidade e baixo custo de desenvolvimento. Esta ferramenta faz uso, ainda, da API (*Application Programming Interface*) nativa de cada plataforma, tornando a interface mais amigável ao usuário.

Dessa forma, o sistema desenvolvido pode ser usado por usuários de diferentes perfis profissionais, por conjugar interface amigável e utilização tanto na plataforma Windows como na Linux. Este capítulo visa apresentar essa interface e o acesso às funcionalidades do sistema.

### 6.1 – Leitura de Informações do Exame

Além do sinal EEG, arquivos com extensão “.peb” (gravados durante a coleta) armazenam dados de interesse sobre o protocolo experimental empregado no exame. O sistema desenvolvido possibilita o acesso ao “cabeçalho do exame”, que contém as informações listadas na Tabela 6.1, as quais se encontram armazenadas sequencialmente. Tais informações são requeridas para as etapas de processamento (e.g.: frequência de amostragem e coeficientes de calibração do amplificador). O acesso ao referido cabeçalho é realizado em tela específica da interface (Figura 6.1), sendo que a Figura 6.2 mostra o funcionamento desta interface, tanto no Windows quanto em Linux. A rotina implementada encontra-se no Apêndice D (D.1).



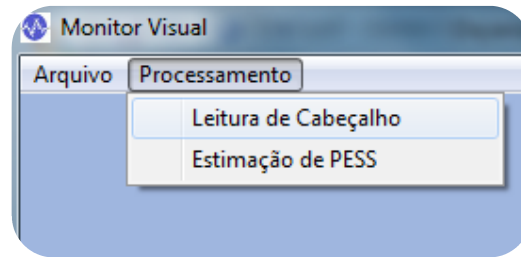


Figura 6.1 – Menu principal do sistema: Seleção de funcionalidade de leitura do cabeçalho do exame

Fonte: Elaboração própria

Tipo de Variável	Conteúdo
<b>Int8</b>	Número de caracteres do nome do tipo de exame
<b>Char*</b>	Nome do tipo de exame
<b>Real</b>	Frequência de amostragem
<b>Int8</b>	Número de caracteres da unidade de tempo
<b>Char*</b>	Unidade de tempo
<b>Int8</b>	Número de canais gravados no arquivo
<b>Real</b>	Coefficiente angular da reta de calibração
<b>Real</b>	Coefficiente linear da reta de calibração
<b>Int8</b>	Número de caracteres dos nomes dos canais gravados no arquivo
<b>Char*</b>	Nomes dos canais armazenados no arquivo
<b>Int8</b>	Número de caracteres dos nomes das unidades do eixo das ordenadas de cada canal
<b>Char*</b>	Nome da unidade do eixo das ordenadas de cada canal
<b>Int8</b>	Número de canais de anotação
<b>Int8</b>	Soma do número de canais, ou seja, número de canais de anotação somados ao número de canais gravados no arquivo
<b>Real</b>	Frequência de corte superior do filtro utilizado
<b>Real</b>	Frequência de corte inferior do filtro utilizado
<b>Int16</b>	Data de realização do exame – Ano, mês e dia
<b>Int16</b>	Horário de realização do exame – hora e minutos
<b>Int8</b>	Número de caracteres dos comentários inseridos no arquivo de exame
<b>Char*</b>	Comentários inseridos no arquivo de exame

Tabela 6.1 – Informações do cabeçalho de um arquivo \*.peb

Fonte: Elaboração própria

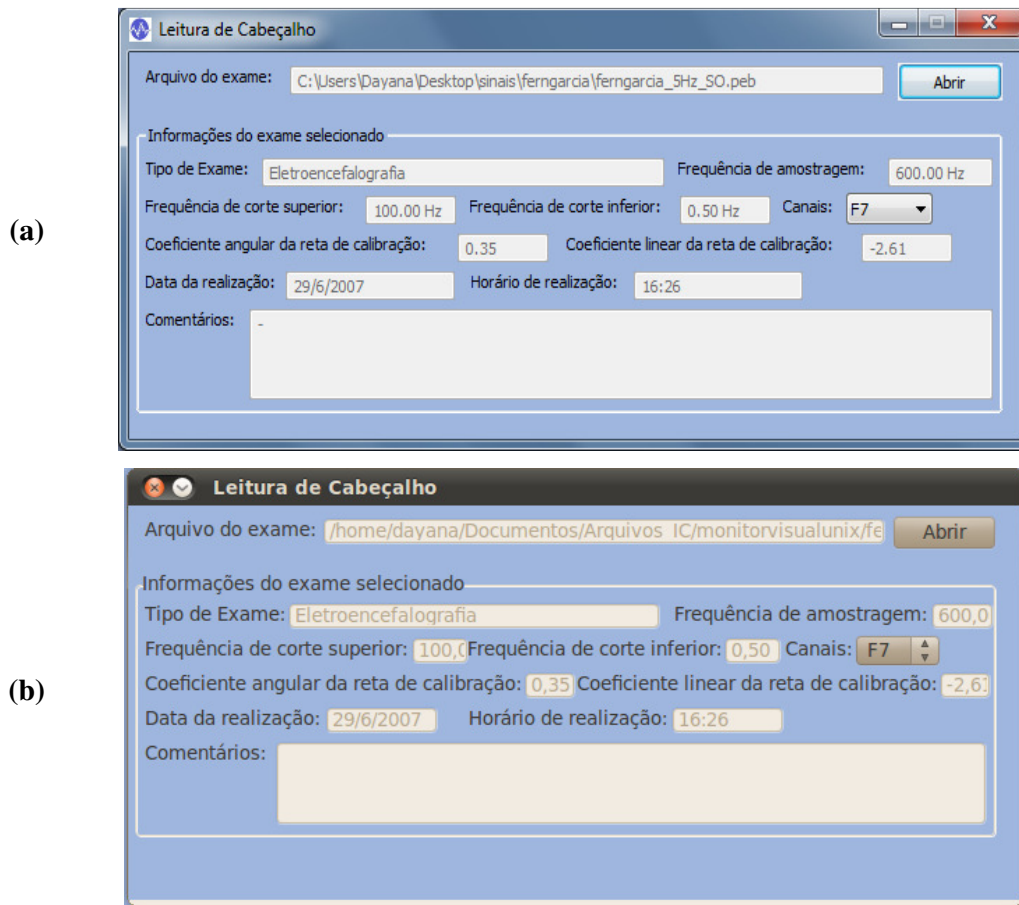


Figura 6.2 – Interface de apresentação de informações do cabeçalho de arquivo \*.peb nas plataformas (a) Windows e (b) Linux

Fonte: Elaboração própria

## 6.2 – Estimação do PESS

A seleção da funcionalidade de “Estimação de PESS” no *menu* principal (Figura 6.1), leva à abertura de uma janela (Figura 6.3) para a inserção de parâmetros necessários para o processamento (e.g.: Estimação do PESS, MSC), bem como para o pré-processamento (e.g.: Janela Tukey e rejeição de artefatos). A Tabela 6.2 contém a explicação de cada campo da etapa de processamento.

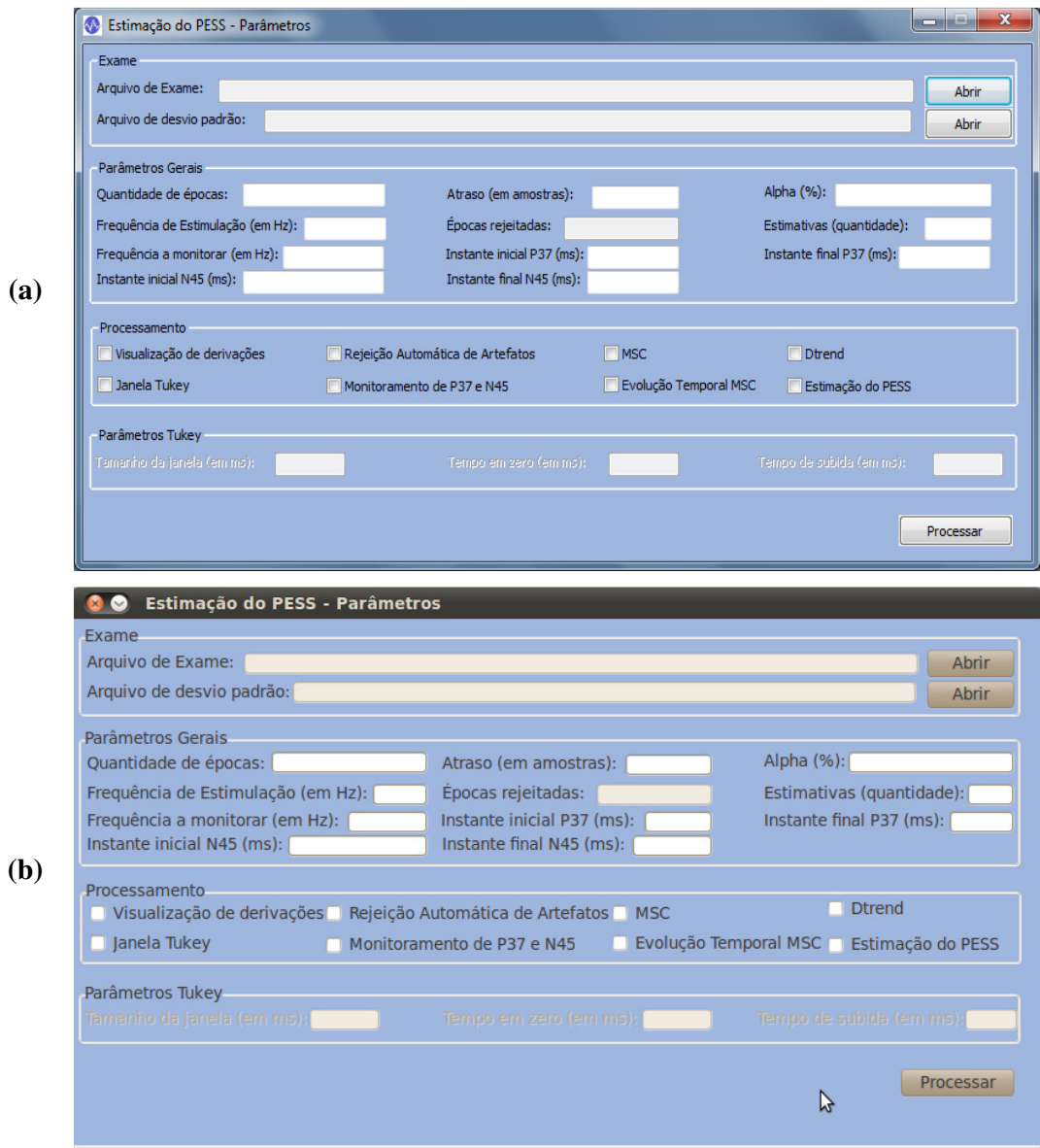


Figura 6.3 – Interface para inserção de parâmetros para processamento do exame nas plataformas (a) Windows e (b) Linux

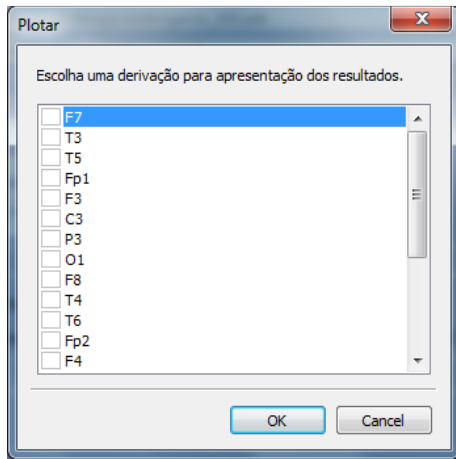
Fonte: Elaboração própria

Uma vez preenchidos os campos com os parâmetros obrigatórios e/ou desejados é requerida a confirmação, mediante o uso do botão “Processar”. Então, é possível selecionar a(s) derivação(ões) desejada(s) por meio da lista apresentada (Figura 6.4). Ao confirmar a(s) escolha(s), são gerados gráficos (Figura 6.5) correspondentes às opções de processamento previamente selecionadas. Conforme se pode observar na Figura 6.6, cada gráfico apresenta *menu* com funcionalidades tais como *zoom*, *grid*, salvar imagem (em formatos \*.png, \*.bmp ou \*.jpg), prever impressão e imprimir.

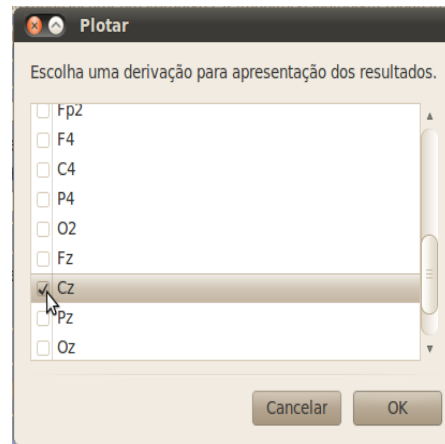
<b>Campo</b>	<b>Conteúdo</b>
<b>Arquivo de Exame</b>	Campo de preenchimento obrigatório. Informa o arquivo de exame através da navegação pelo botão “Abrir”, dado que seu preenchimento manual é bloqueado.
<b>Arquivo de desvio padrão</b>	Campo de preenchimento obrigatório. Informa o arquivo de desvio padrão através da navegação pelo botão “Abrir”, dado que seu preenchimento manual é bloqueado.
<b>Quantidade de épocas</b>	Campo de preenchimento obrigatório. Contém o número de épocas a ser utilizada na promediação e/ou por estimativa (para o caso de evolução temporal).
<b>Frequência de Estimulação (em Hz)</b>	Campo de preenchimento obrigatório. Esta informação não está contida no cabeçalho do exame, podendo ser variável. É um valor real com casas decimais separadas por vírgula.
<b>Atraso (em amostras)</b>	Campo de preenchimento opcional. Diz respeito à quantidade de amostras que se pode atrasar cada época. Caso não seja preenchido pelo usuário, este valor é “setado” em zero.
<b>Épocas rejeitadas</b>	Caso a checkbox de rejeição automática de artefatos seja marcada, informa ao usuário quantas épocas foram rejeitadas no último processamento realizado.
<b>Alpha</b>	Nível de significância para o cálculo do limiar de detecção na MSC. Seu preenchimento é obrigatório caso qualquer cálculo de processamento que envolva a MSC seja assinalado.
<b>Estimativas</b>	Quantidade de estimativas realizadas na Evolução Temporal da Morfologia.
<b>Frequência a monitorar (em Hz)</b>	Frequência a ser analisada na Evolução Temporal da MSC, sendo seu preenchimento obrigatório caso esta opção esteja assinalada.
<b>Instante inicial P37</b>	Instante de tempo, ao longo do sinal promediado, para começar a varredura à procura do vale P37. O preenchimento é obrigatório, caso a opção de Evolução Temporal de P37 e N45 esteja sinalizada.
<b>Instante final P37</b>	Instante de tempo, ao longo do sinal promediado, para começar a varredura à procura do vale P37. O preenchimento é obrigatório, caso a opção de Evolução Temporal de P37 e N45 esteja sinalizada.
<b>Instante inicial N45</b>	Análogo ao instante inicial P37.
<b>Instante final N45</b>	Análogo ao instante final P37.
<b>Visualização de derivações</b>	Permite a visualização do período estimulado do exame, sem qualquer processamento.
<b>Janela Tukey</b>	Possibilita aplicação de Janela Tukey a cada época processada. Ativa o preenchimento dos parâmetros da janela.
<b>Rejeição Automática de Artefatos</b>	Comporta aplicação de rejeição de artefatos durante o processamento.
<b>Monitoramento de P37 e N45</b>	Seleciona funcionalidade de Evolução Temporal da Morfologia ao longo do exame.
<b>MSC</b>	Permite a aplicação da técnica de MSC.
<b>Evolução Temporal da MSC</b>	Admite utilização da técnica de Evolução Temporal da MSC ao longo do exame.
<b>Dtrend</b>	Realiza a remoção de valor médio do sinal promediado.
<b>Estimação do PESS</b>	Permite a apresentação do sinal promediado para o número de épocas selecionado.
<b>Tamanho de Janela (em ms)</b>	Campo de preenchimento obrigatório mediante seleção da funcionalidade de Janela Tukey. Tamanho da janela aplica, devendo ter a mesma duração que uma época.
<b>Tempo em zero (em ms)</b>	Campo de preenchimento obrigatório mediante seleção da funcionalidade de Janela Tukey. Tempo da janela fixo em zero.
<b>Tempo de subida (em ms)</b>	Campo de preenchimento obrigatório mediante seleção da funcionalidade de Janela Tukey. Tempo de transição da janela aplicada.

Tabela 6.2 – Parâmetros para estimação do PESS

Fonte: Elaboração própria



(a)

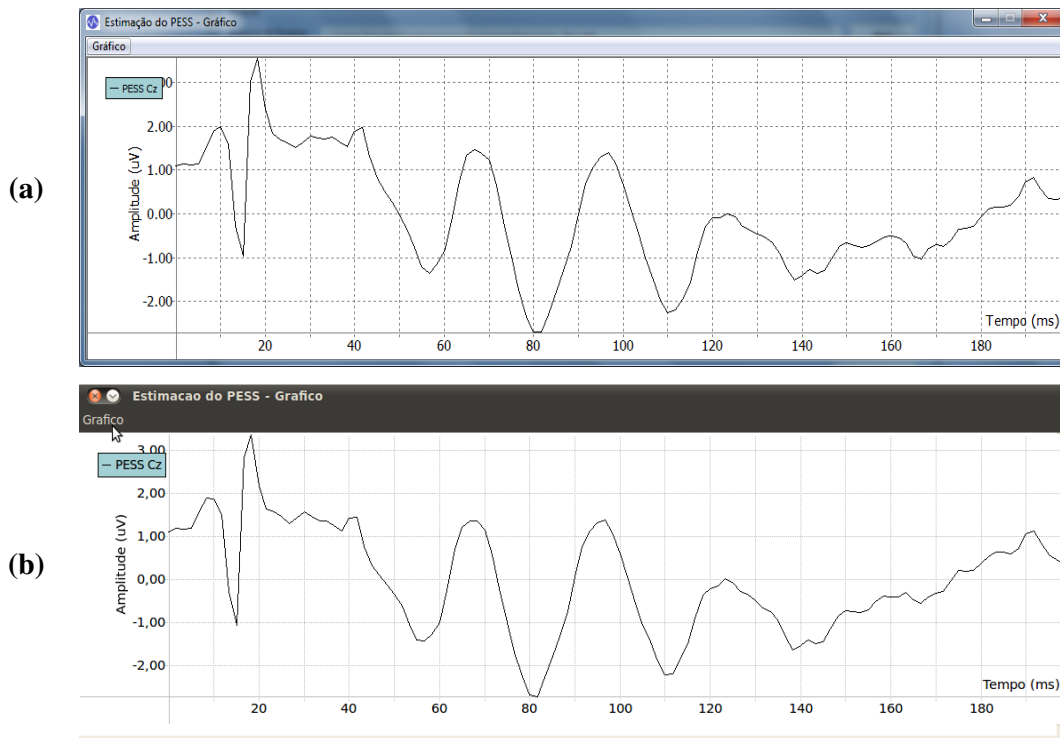


(b)

Figura 6.4 – Seleção de derivações desejadas nas plataformas (a) Windows e (b)

Linux

Fonte: Elaboração própria



(a)

(b)

Figura 6.5 – Apresentação gráfica de resultados nas plataformas (a) Windows e

(b) Linux

Fonte: Elaboração própria

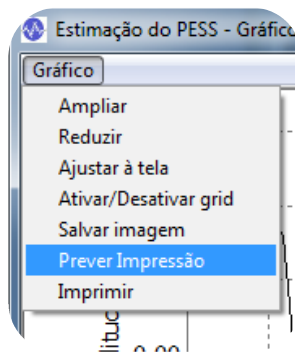


Figura 6.6 – Opções de melhor visualização gráfica e impressão

Fonte: Elaboração própria

# Capítulo 7

## Discussão e Conclusão

O sistema desenvolvido se mostrou adequado para a investigação das atividades corticais em resposta à estimulação sensorial, em especial somato-sensitiva. As técnicas de pré-processamento implementadas resultaram em estimativas do PESS e da MSC compatíveis com aquelas da literatura. Cabe salientar que a Evolução Temporal da MSC deve ser calculada levando-se em consideração o compromisso entre rapidez na detecção e menor variabilidade dos valores estimados, ou seja, uma escolha adequada do número de épocas  $M$  se faz necessária. Com vistas a aumentar a rapidez de detecção quando do uso da MSC, [36] propõe o emprego desta técnica ORD com esquecimento exponencial.

Embora tenha sido validado com sinais EEG durante estimulação somato-sensitiva, o sistema desenvolvido, por sua concepção modular, poderá vir a ser utilizado com outros tipos de potencial evocado, desde que sejam realizadas alterações que o adequem ao tipo de estímulo empregado, bem como ao protocolo em questão. Além disso, o benefício de se ter um sistema concebido com arquitetura modular garante a reutilização das funções desenvolvidas em outros sistemas, bem como maior facilidade de manutenção.

O sistema desenvolvido provê versatilidade de uso por basear-se em ferramentas de desenvolvimento não-proprietárias e multiplataforma (Windows e Linux/Unix), sendo essa sua principal vantagem frente aos *softwares* atualmente disponíveis. A utilização de tais ferramentas garante, ainda, a isenção de pagamento de licenças, o que é fundamental no âmbito científico. Adicionalmente, a ferramenta foi desenvolvida nas linguagens de programação C/C++, amplamente utilizadas no meio comercial e acadêmico; e apresenta código aberto, o que assegura auditabilidade e maior facilidade de expansão.

A biblioteca gráfica wxWidgets, por ser nativa, garante maior agilidade na execução, além de permitir que o visual do sistema seja adaptado à plataforma em que se encontra. Esta biblioteca possui uma comunidade de desenvolvedores em expansão e ampla documentação na *web*.

O desempenho computacional das rotinas depende das especificações do computador em uso e das versões do sistema operacional disponíveis. Tais aspectos são relevantes em especial para as funções de cálculo, que utilizam intensamente recursos computacionais de processamento e memória. Este é o caso, por exemplo, das rotinas de Média Coerente, FFT, MSC, Evolução Temporal da MSC e de características extraídas do PESS. O sistema apresentou bom desempenho quando testado em computadores com processadores Dual Core Intel<sup>®</sup> e AMD<sup>®</sup> Turion X2, com *clocks* de 2,2 GHz e 2,1 GHz, respectivamente. Além disso, o sistema pode ser executado em computadores PC de 32 bits ou 64 bits.

Para o desenvolvimento do sistema, a instalação da biblioteca wxWidgets requer que a máquina possua pelo menos 100 MB de espaço em disco, além de espaço adicional para a aplicação. Os compiladores utilizados compreenderam MinGW, para Windows, e GCC, para Linux.

Finalmente, o teste do sistema envolveu os sistemas operacionais Windows XP/Vista/7 e Ubuntu 10.4 LTS (Lucid Lynx)/10.10 (Maverick Meerkat). Ao ser executado em um AMD<sup>®</sup> Turion X2 com *clock* de 2,2 GHz, o *software* desenvolvido apresentou melhor desempenho neste último sistema operacional. As ferramentas utilizadas ainda comportam sua utilização no sistema operacional Apple Mac OS, embora essa possibilidade não tenha sido testada.



# Capítulo 8

## Trabalhos Futuros

Dada a modularidade da arquitetura do projeto e tendo em vista o estágio em que se encontra, propõe-se a melhoria das técnicas de evolução temporal da MSC pela inserção da característica de esquecimento exponencial, visando aprimorar esta análise.

Ademais, podem ser desenvolvidos módulos de aquisição de sinais EEG. Isto permitiria utilizar o presente sistema na análise do PESS *online*.

# Capítulo 9

## Trabalhos Apresentados e Prêmio

Durante o desenvolvimento deste projeto, os seguintes trabalhos foram/serão publicados:

1. "Sistema para Análise do Potencial Evocado Somato-Sensitivo Baseado em Software Multiplataforma", Lole, D.S., Melges, D.B., Infantosi, A.F.C., Livro de Resumos da XXX Jornada Giulio Massarani de Iniciação Científica, Artística e Cultural – UFRJ, 2008, p. 118, (Inscrição: no. 2737), 3-7 Novembro 2008.

2. "Sistema para Detecção Objetiva da Resposta Evocada Somato-sensitiva utilizando Software Multiplataforma", Lole, D.S., Melges, D.B., Infantosi, A.F.C., Livro de Resumos da XXXI Jornada Giulio Massarani de Iniciação Científica, Artística e Cultural – UFRJ, 2009, p. 226-227 (Código: 750), 6-9 Outubro 2009.

3. "Sistema multiplataforma para obtenção da média coerente do EEG durante estimulação somato-sensitiva", Lole, D.S., Melges, D.B., Infantosi, A.F.C., Livro de Programa e Resumo do XXI Congresso Brasileiro de Engenharia Biomédica - CBEB2008, v. 1. p. 180, Salvador /Bahia, 16-20 Novembro 2008.

4. "Sistema para Detecção Objetiva da Resposta Evocada Somato-sensitiva utilizando Software Multiplataforma", Lole, D.S., Melges, D.B., Infantosi, A.F.C., Livro eletrônico dos Anais e Resumos da 62ª Reunião Anual da SBPC, Natal/Rio Grande do Norte, 25-30 Julho 2010 – com previsão de publicação para Outubro 2010 no endereço eletrônico do evento [[www.sbpnet.org.br/natal/home/](http://www.sbpnet.org.br/natal/home/)].

5. “Monitorização da Evolução Temporal da Resposta à Estimulação Somato-Sensitiva usando a Magnitude Quadrática da Coerência”, Lole, D.S., Melges, D.B., Infantosi, A.F.C., Livro de Resumos da XXXII Jornada Giulio Massarani de Iniciação Científica, Artística e Cultural – UFRJ, 2010, p. 11, (Código: 733), 4-8 Outubro 2010.

6. “Monitorização da Resposta à Estimulação Somato-sensitiva nos Domínios do Tempo e da Frequência”, Lole, D.S., Melges, D.B., Infantosi, A.F.C., XXII Congresso Brasileiro de Engenharia Biomédica – CBEB 2010, Tiradentes/Minas Gerais (aceito), 21-25 Novembro 2010.

A apresentação do trabalho “Sistema para Detecção Objetiva da Resposta Evocada Somato-sensitiva utilizando Software Multiplataforma” na XXXI Jornada Giulio Massarani de Iniciação Científica, Artística e Cultural (JICAC 2009 - UFRJ) recebeu o Prêmio de Melhor Trabalho do Centro de Tecnologia.

## Bibliografia

- [1] AVELAR, F., GOMES, V.C., POZZER, C.T., *Estudo Comparativo de Bibliotecas Gráficas Integradas com OPENGL*, Disponível em: [http://www.wusr.inf.ufsm.br/~avelar/arquivos/artigo\\_cricte\\_2007.pdf](http://www.wusr.inf.ufsm.br/~avelar/arquivos/artigo_cricte_2007.pdf) (Acesso em 22 de novembro de 2009).
- [2] AZEVEDO, E.P., *Determinação das bandas de máxima resposta do potencial evocado auditivo de curta latência por meio da magnitude quadrática da coerência*, Dissertação de Mestrado, Rio de Janeiro, Brasil, COPPE/UFRJ, 2003.
- [3] BARRY, S. R., *Eyes on the Brain*, Disponível em: <http://www.psychologytoday.com/> (Acesso em 07 de outubro de 2010).
- [4] BEAR, M.F., CONNORS, B.W., PARADISO, M.A., *Neurociências: Desvendando o Sistema Nervoso*, Porto Alegre, Brasil, Artmed, 2002.
- [5] BENDAT, J.S., PIERSOL, A.G., *Random data analysis and measurement procedures*, 3 ed., New York, USA, Wiley-Interscience, 2000.
- [6] CAHN, B.R., POLICH J., “Meditation states and traits: EEG, ERP, and neuroimaging studies”, *Psychological Bulletin*, vol. 132, n. 2, pp. 180-211, 2006.
- [7] CHIAPPA, K.H., *Evoked Potentials in Clinical Medicine*, chapter Short-latency somatosensory evoked potentials: Methodology, New York, USA, Raven, pp. 283-340, 1997.
- [8] COOLEY, J.W., TUKEY, J.W., *An algorithm for the machine calculation of complex Fourier series*, *Math Comput.*, n. 19, pp. 297-301, 1965.
- [9] DINIZ, P., SILVA, E. B., NETTO, S.L., *Processamento Digital de Sinais: Projeto e Análise de Sistemas*. Porto Alegre, Brasil, Bookman, 2004.

- [10] DOBIE, R.A., WILSON, M.J., “Objective response detection in the frequency domain”, *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, v. 88, n. 6, pp. 516-524, 1993.
- [11] FACCO, E., BEHR, A.U., MUNARI, M., BARATTO, F., VOLPIN, S.M., GALLO, F., LANZILLOTTA, M.A., GIRON, G.P., “Auditory and somatosensory evoked potentials in coma following spontaneous cerebral hemorrhage: early prognosis and outcome”, *Electroencephalography and clinical Neurophysiology*, v. 107, pp. 332-338, 1998.
- [12] FESHCHENKO, V.A., REINSEL, R.A., VESELIS, R.A., “Multiplicity of the alpha rhythm in normal humans”, *J Clin Neurophysiol*, vol. 18, n. 4, pp. 331–44, 2001.
- [13] FLORENCE, G., GUÉRIT, J.M., GUEGUEN, B., “Electroencephalography and somatosensory evoked potentials to prevent cerebral ischaemia in the operating room”, *Neurophysiologie Clinique/Clinical Neurophysiology*, v. 34, n.1, pp. 17-32, 2004.
- [14] GALLA, J.D., ERGIN, M.A., LANSMAN, S.L. *et al*, “Use of somatosensory evoked potentials for thoracic and thoracoabdominal aortic resections”, In: *The Annals of Thoracic Surgery*, v. 67, n. 6, pp. 1947-1952, 1999.
- [15] GIUGNO, K.M., MAIA, T.R., KUNRATH, C.L. *et al*, “Tratamento da hipertensão intracraniana”, *Jornal de Pediatria*, v. 79, n. 4, pp. 287-296, 2003.
- [16] INFANTOSI, A.F.C., MELGES, D.B., TIERRA-CRIOLLO, C.J., “Use of magnitude-squared coherence to identify the maximum driving response band of the somatosensory evoked potential”, *Brazilian Journal of Medical and Biological Research*, vol. 39, pp.1593-1603, 2006.
- [17] KLAS, D.W., DALY, D.D., *Current Practice of Clinical Electroencephalography*. New York, USA, Raven, 1980.

- [18] LOPES, T., *Programming with wxDev-C++*, SoftT, 2006.
- [19] MARKAND, O.N., WARREN, C., MALIK, G.S., KING, R.D., BROWN, J.W., MAHOMED, Y., “Effects of hypothermia on short latency somatosensory evoked potentials in humans”, *Electroencephalography and Clinical Neurophysiology*, v. 77, pp. 416-424, 1990.
- [20] MARTIN, C.J., SINSON, G., PATTERSON, T., ZAGER, E.L., STECKER, M.M., “Sensitivity of scalp EEG, cortical EEG and somatosensory evoked responses during surgery for intracranial aneurysms”, *Surgical Neurology*, v.58, n. 5, pp. 317-321, 2002.
- [21] MELGES, D. B., *Instrumento Virtual para Monitorização da Resposta à Estimulação Somato-Sensitiva*. Dissertação de Mestrado, Rio de Janeiro, Brasil, COPPE/UFRJ, 2005.
- [22] MELGES, D.B., *Aplicação de técnicas de detecção objetiva uni e multivariadas no domínio da frequência ao EEG durante estimulação somato-sensitiva*, Tese de doutorado, Rio de Janeiro, Brasil, COPPE/UFRJ, 2009.
- [23] MELGES, D. B., INFANTOSI, A. F. C., MIRANDA DE SÁ, A. M. F. L., *Topographical distribution of the somatosensory evoked potential: an objective response detection approach*. In: IV Congreso Latinoamericano de Ingeniería Biomédica 2007, Isla de Margarita. IFMBE Proceedings. Isla de Margarita, v. 18. pp. 34-37, 2007.
- [24] MIRANDA DE SÁ, A.M.F.L., INFANTOSI, A.F.C., “A coherence-based technique for evaluating the degree of synchronism in the EEG during sensory stimulation”, *Revista Brasileira de Engenharia Biomédica*, v. 18, n.1, pp. 39-49, 2002.
- [25] MISULIS, K., “Normal SEP's to leg stimulation”. In: *Spehlmann's Evoked Potential Primer: Visual, Auditory and Somatosensory Evoked Potentials in Clinical Diagnosis*, Boston, USA, Butterworth-Heinemann, 1994.

- [26] NIEDERMEYER, E., “Alpha rhythms as physiological and abnormal phenomena”, *Int J Psychophysiol*, vol. 26 (1-3), 31–49, 1997.
- [27] NIEDERMEYER, E., *Eletroencefalography – Basic Principles, Clinical Applications and Related Fields*, chapter The normal EEG of the waking adult, Baltimore, USA, Williams & Wilkins, pp. 149-173, 1999.
- [28] NUWER, M.R., AMINOFF, M., DESMEDT, J., EISEN, A.A., GOODIM, D., MAUGUIÉRE, F., SHIBASAKI, H., SUTHERLING, W., VIBERT, J.F., “IFCN recommended standards for short latency somatosensory evoked potentials. Report of IFCN committee”, *Electroencefalography and Clinical Neurophysiology*, vol. 13, n. 4, pp. 6-11, 1994.
- [29] PFURTSCHELLER, G., LOPES DA SILVA, F.H. “Event-related EEG/MEG synchronization and desynchronization: basic principles” *Clin Neurophysiol*, vol. 110, n. 11, pp. 1842-1857, 1999.
- [30] REGAN, D., “Some characteristics of average steady-state and transient responses evoked by modulated light”, *Electroencefalography and Clinical Neurophysiology*, vol. 20, pp. 238–248, 1966.
- [31] SAUNDERS, M.G., *Current Practice of Clinical Electroencefalography*, chapter Artifacts: Activity of Noncerebral Origin in the EEG, New York, USA, Raven Press, pp. 37-67, 1979.
- [32] SCAPPINI JUNIOR, W., *Potenciais Evocados*, Disponível em: <http://www.neurocamed.com.br/>, (Acesso em 07 de outubro de 2010).
- [33] SHACKMAN, A.J., MCMENAMIN, B.W., MAXWELL, J.S., GREISCHAR, L.L., DAVIDSON, R.J. “Identifying robust and sensitive frequency bands for interrogating neural oscillations”, *NeuroImage*, vol. 4, n. 51, pp. 1319-1333, 2010.
- [34] SIMPSON, D.M., TIERRA-CRIOLLO, C.J., LEITE, R.T., *et al.*, “Objective Response Detection in an Electroencefalogram during Somatosensory

- Stimulation”, In: *Annals of Biomedical Engineering*, vol. 28, n. 6, pp. 691-698, 2000.
- [35] TIERRA-CRIOLLO, C.J., INFANTOSI, A.F.C., “Determinação da banda de máxima resposta para estimulação do nervo tibial”, In: *Anais do XVIII Congresso Brasileiro de Engenharia Biomédica*, pp. 476-479, São José dos Campos, 2002.
- [36] TIERRA-CRIOLLO, C.J., *Monitorização objetiva da resposta à estimulação somato-sensitiva utilizando parâmetros espectrais*, Tese de doutorado, Rio de Janeiro, Brasil, COPPE/UFRJ, 2001.



# Apêndice A

## Resultados da utilização do Sistema para diversas frequências de estimulação

A fim de manter a compatibilidade dos exemplos, ao longo do texto foram apresentados apenas resultados referentes à frequência de estimulação de 5Hz. Neste apêndice são apresentados resultados da utilização de sinais com outras frequências de estimulação para ilustrar a abrangência da ferramenta desenvolvida.

Cabe ressaltar que as observações a respeito dos resultados analisados anteriormente também são válidas para estes exemplos. Os resultados aqui ilustrados dizem respeito à Média Coerente e à MSC, dado que a Evolução Temporal (Morfologia e MSC) são extensões destas. Foram utilizadas as técnicas de pré-processamento (Janela Tukey e Rejeição de Artefatos) para todos os exemplos.

### A.1 - Frequência de estimulação de 2 Hz

#### *Média Coerente*

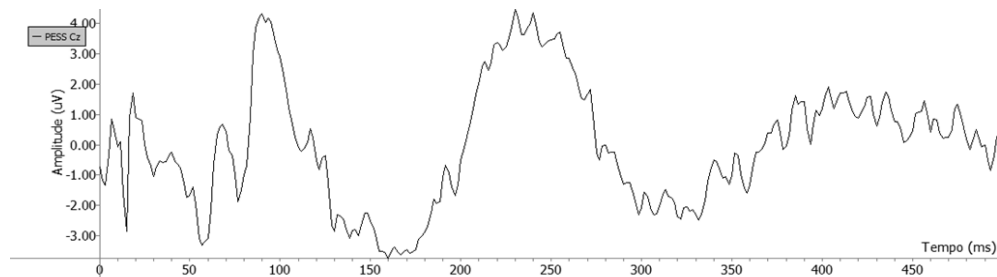


Figura A.1 – Estimativa do PESS (derivação Cz) por Média Coerente ( $M = 500$  épocas) para o Voluntário #7 estimulado a 11 mA com frequência de estimulação nominal de 2Hz

Fonte: Elaboração própria

## MSC

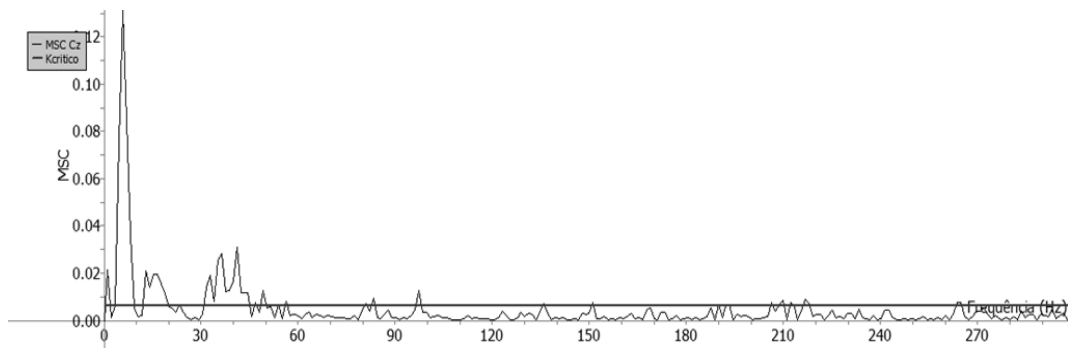


Figura A.2 – MSC do PESS (derivação Cz) para o Voluntário #7 estimulado a 11 mA com frequência de estimulação nominal de 2Hz ( $M = 500$  épocas e  $\alpha = 0,05$ :  $\hat{\kappa}_{crit} = 0,006$ )

Fonte: Elaboração própria

## A.2 - Frequência de estimulação de 7 Hz

### Média Coerente

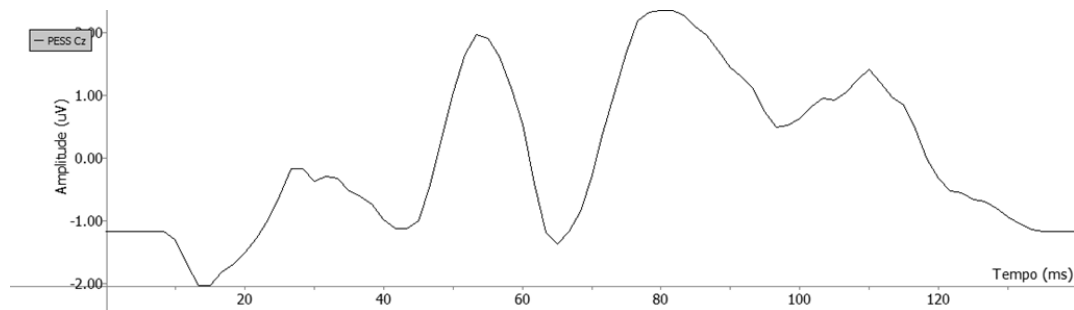


Figura A.3 – Estimativa do PESS (derivação Cz) por Média Coerente ( $M = 500$  épocas) para o Voluntário #8 estimulado a 11 mA com frequência de estimulação nominal de 7Hz

Fonte: Elaboração própria

MSC

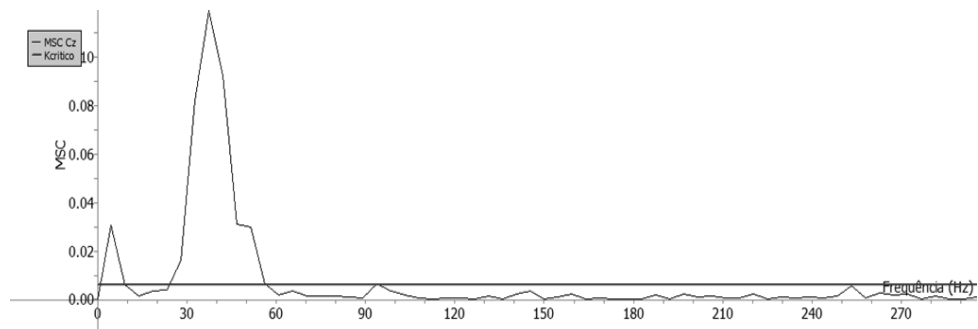


Figura A.4 – MSC do PESS (derivação Cz) para o Voluntário #8 estimulado a 11 mA com frequência de estimulação nominal de 7Hz ( $M = 500$  épocas e  $\alpha = 0,05$ :  $\hat{\kappa}_{crit} = 0,006$ )

Fonte: Elaboração própria

### A.3 - Frequência de estimulação de 9 Hz

*Média Coerente*

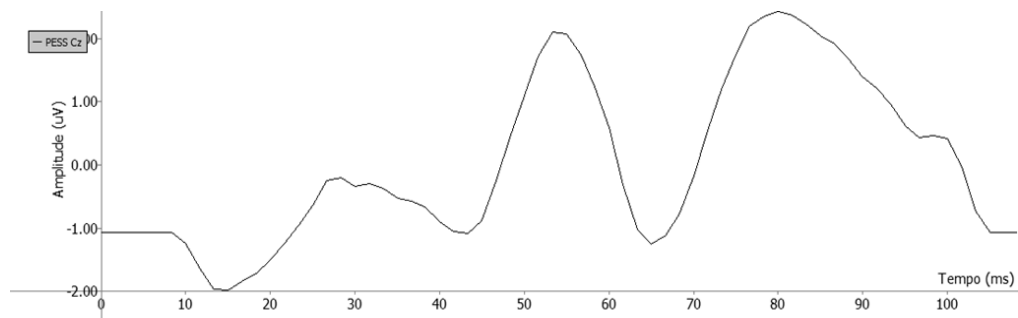


Figura A.5 – Estimativa do PESS (derivação Cz) por Média Coerente ( $M = 500$  épocas) para o Voluntário #9 estimulado a 11 mA com frequência de estimulação nominal de 9Hz

Fonte: Elaboração própria

## MSC

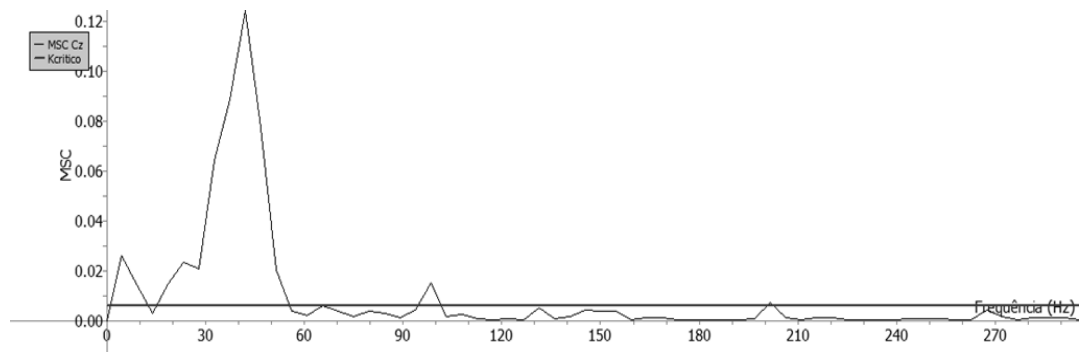


Figura A.6 – MSC do PESS (derivação Cz) para o Voluntário #9 estimulado a 11 mA com frequência de estimulação nominal de 9Hz (M = 500 épocas e  $\alpha = 0,05$ :  $\hat{k}_{crit} = 0,006$ )

Fonte: Elaboração própria

# Apêndice B

## Guia de instalação de wxWidgets

### B.1 – Linux

Para instalação desta biblioteca em Linux, primeiramente é necessária a obtenção da versão mais recente do wxGTK no endereço eletrônico de seus desenvolvedores ([www.wxwidgets.org/downloads/](http://www.wxwidgets.org/downloads/)). Neste ponto, deve-se verificar se a versão do GTK disponível na máquina é compatível com a requerida para instalação do wxGTK (através do comando `gtk-config --version`). Para a instalação, os procedimentos abaixo são necessários:

1. Extrair o pacote na localização de interesse (pasta base).
2. Abrir o terminal e ir até a pasta base.
3. Criar um novo diretório através do comando `mkdir buildgtk`.
4. Entrar no diretório criado: `cd buildgtk`.
5. Verificar a necessidade de dependências: `./configure --with-gtk`.
6. Compilar a biblioteca: `make`.
7. Para que o acesso às bibliotecas não seja restrito pelo diretório criado e esteja disponível a todos os usuários, usa-se o comando: `sudo make install`. Este comando deve ser seguido de `sudo ldconfig` para concluir. Cabe ressaltar que para a execução deste último procedimento é necessário que o usuário tenha acesso de administrador.

### B.2 – Windows

Para a instalação da biblioteca em Windows, o endereço eletrônico dos desenvolvedores ([www.wxwidgets.org/downloads/](http://www.wxwidgets.org/downloads/)) dá acesso direto a um arquivo executável que realiza todas as tarefas autonomamente.

# Apêndice C

## *Makefile* para compilação

### C.1 – Linux

```
# Universidade Federal do Rio de Janeiro
# Programa de Engenharia Biomedica - COPPE
#
# Autor: Dayana Sant Anna Lole
# Description: Makefile for linux

CPP          = g++

OBJ          = Objects/PessEstimation.o Objects/PessEstimFrm.o
Objects/PessEstimGraphFrm.o Objects/MonitorVisualFrm.o
Objects/HeaderReadingFrm.o Objects/Util.o Objects/GraphPlot.o
Objects/Tukey.o Objects/MonitorVisualApp.o Objects/mathplot.o

LINKOBJ      = "Objects/PessEstimation.o" "Objects/PessEstimFrm.o"
"Objects/PessEstimGraphFrm.o" "Objects/MonitorVisualFrm.o"
"Objects/HeaderReadingFrm.o" "Objects/Util.o"
"Objects/GraphPlot.o" "Objects/Tukey.o"
"Objects/MonitorVisualApp.o" "Objects/mathplot.o"

#This item must use the value returned by the command
#wx-config --libs (on the terminal)
LIBS        = -L"/usr/local/lib" -pthread -lwx_gtk2_richtext-2.8
-lwx_gtk2_aui-2.8 -lwx_gtk2_xrc-2.8 -lwx_gtk2_qa-2.8 -
lwx_gtk2_html-2.8 -lwx_gtk2_adv-2.8 -lwx_gtk2_core-2.8 -
lwx_base_xml-2.8 -lwx_base_net-2.8 -lwx_base-2.8

BIN          = Output/monitorVisualUnix

#This item must use the value returned by the command
#wx-config --cxxflags (on the terminal)
CXXFLAGS    = -I"/usr/local/lib/wx/include/gtk2-ansi-release-2.8"
-I"/usr/local/include/wx-2.8" -D_FILE_OFFSET_BITS=64 -
D_LARGE_FILES -D__WXGTK__ -pthread

RM          = rm -f

LINK        = g++

.PHONY: all all-before all-after clean clean-custom
all: all-before $(BIN) all-after

clean: clean-custom
```

```

$(RM) $(LINKOBJ) "$$(BIN)"

$(BIN): $(OBJ)
    $(LINK) $(LINKOBJ) -o "$$(BIN)" $(LIBS)

Objects/PessEstimation.o: $(GLOBALDEPS) PessEstimation.cpp
    $(CPP) -c PessEstimation.cpp -o Objects/PessEstimation.o
    $(CXXFLAGS)

Objects/PessEstimFrm.o: $(GLOBALDEPS) PessEstimFrm.cpp
    $(CPP) -c PessEstimFrm.cpp -o Objects/PessEstimFrm.o
    $(CXXFLAGS)

Objects/PessEstimGraphFrm.o: $(GLOBALDEPS) PessEstimGraphFrm.cpp
    $(CPP) -c PessEstimGraphFrm.cpp -o
Objects/PessEstimGraphFrm.o $(CXXFLAGS)

Objects/MonitorVisualFrm.o: $(GLOBALDEPS) MonitorVisualFrm.cpp
    $(CPP) -c MonitorVisualFrm.cpp -o
Objects/MonitorVisualFrm.o $(CXXFLAGS)

Objects/HeaderReadingFrm.o: $(GLOBALDEPS) HeaderReadingFrm.cpp
    $(CPP) -c HeaderReadingFrm.cpp -o
Objects/HeaderReadingFrm.o $(CXXFLAGS)

Objects/Util.o: $(GLOBALDEPS) Util.cpp
    $(CPP) -c Util.cpp -o Objects/Util.o $(CXXFLAGS)

Objects/GraphPlot.o: $(GLOBALDEPS) GraphPlot.cpp
    $(CPP) -c GraphPlot.cpp -o Objects/GraphPlot.o $(CXXFLAGS)

Objects/Tukey.o: $(GLOBALDEPS) Tukey.cpp
    $(CPP) -c Tukey.cpp -o Objects/Tukey.o $(CXXFLAGS)

Objects/MonitorVisualApp.o: $(GLOBALDEPS) MonitorVisualApp.cpp
    $(CPP) -c MonitorVisualApp.cpp -o
Objects/MonitorVisualApp.o $(CXXFLAGS)

Objects/mathplot.o: $(GLOBALDEPS) mathplot.cpp
    $(CPP) -c mathplot.cpp -o Objects/mathplot.o $(CXXFLAGS)

```

## C.2 – Windows

```
# Universidade Federal do Rio de Janeiro
# Programa de Engenharia Biomedica - COPPE
#
# Autor: Dayana Sant Anna Lole
# Description: Makefile for windows

WXLIBNAME = wxmsw28

CPP        = g++.exe

CC         = gcc.exe

WINDRES    = "windres.exe"

OBJ        = Objects/MingW/PessEstimation.o
Objects/MingW/PessEstimFrm.o Objects/MingW/PessEstimGraphFrm.o
Objects/MingW/MonitorVisualFrm.o
Objects/MingW/HeaderReadingFrm.o Objects/MingW/Util.o
Objects/MingW/GraphPlot.o Objects/MingW/Tukey.o
Objects/MingW/MonitorVisualApp.o Objects/MingW/mathplot.o

LINKOBJ    = "Objects/MingW/PessEstimation.o"
"Objects/MingW/PessEstimFrm.o"
"Objects/MingW/PessEstimGraphFrm.o"
"Objects/MingW/MonitorVisualFrm.o"
"Objects/MingW/HeaderReadingFrm.o" "Objects/MingW/Util.o"
"Objects/MingW/GraphPlot.o" "Objects/MingW/Tukey.o"
"Objects/MingW/MonitorVisualApp.o" "Objects/MingW/mathplot.o"

LIBS       = -L"C:/Program Files/Dev-Cpp/Lib" -mwindows -
l$(WXLIBNAME) -l$(WXLIBNAME)_gl -lwxtiff -lwjpeg -lwxpng -
lwxzlib -lwregex -lwexpat -lkernel32 -luser32 -lgdi32 -
lcomdlg32 -lwinpool -lwinmm -lshell32 -lcomctl32 -lole32 -
loleaut32 -luuid -lrpcrt4 -ladvapi32 -lwsock32 -lodbc32 -
lopengl32

INCS       = -I"C:/Program Files/Dev-Cpp/Include"

CXXINCS    = -I"C:/Program Files/Dev-
Cpp/lib/gcc/mingw32/3.4.5/include" -I"C:/Program Files/Dev-
Cpp/include/c++/3.4.5/backward" -I"C:/Program Files/Dev-
Cpp/include/c++/3.4.5/mingw32" -I"C:/Program Files/Dev-
Cpp/include/c++/3.4.5" -I"C:/Program Files/Dev-Cpp/include" -
I"C:/Program Files/Dev-Cpp/" -I"C:/Program Files/Dev-
Cpp/include/common/wx/msw" -I"C:/Program Files/Dev-
Cpp/include/common/wx/generic" -I"C:/Program Files/Dev-
Cpp/include/common/wx/html" -I"C:/Program Files/Dev-
Cpp/include/common/wx/protocol" -I"C:/Program Files/Dev-
Cpp/include/common/wx/xml" -I"C:/Program Files/Dev-
Cpp/include/common/wx/xrc" -I"C:/Program Files/Dev-
Cpp/include/common/wx" -I"C:/Program Files/Dev-
```



```

Cpp/include/common" -I"C:/Program Files/Dev-
Cpp/include/wxMathPlot"

RCINCS      = --include-dir "C:/PROGRA~1/Dev-Cpp/include/common"

BIN         = Output/MingW/monitorVisualUnix.exe

DEFINES     = -D__WXMSW__ -D__GNUWIN32__ -D__WIN95__

CXXFLAGS    = $(CXXINCS) $(DEFINES) -fno-exceptions -fno-pcc-
struct-return -fstrict-aliasing -Wall -D__WXMSW__ -D__GNUWIN32__
-D__WIN95__ -fexpensive-optimizations -O3

CFLAGS      = $(INCS) $(DEFINES) -fno-exceptions -fno-pcc-struct-
return -fstrict-aliasing -Wall -D__WXMSW__ -D__GNUWIN32__ -
D__WIN95__ -fexpensive-optimizations -O3

GPROF       = gprof.exe

RM          = rm -f

LINK        = g++.exe

.PHONY: all all-before all-after clean clean-custom
all: all-before $(BIN) all-after

clean: clean-custom
      $(RM) $(LINKOBJ) "$(BIN)"

$(BIN): $(OBJ)
      $(LINK) $(LINKOBJ) -o "$(BIN)" $(LIBS)

Objects/MingW/PessEstimation.o: $(GLOBALDEPS) PessEstimation.cpp
      $(CPP) -c PessEstimation.cpp -o
Objects/MingW/PessEstimation.o $(CXXFLAGS)

Objects/MingW/PessEstimFrm.o: $(GLOBALDEPS) PessEstimFrm.cpp
      $(CPP) -c PessEstimFrm.cpp -o Objects/MingW/PessEstimFrm.o
$(CXXFLAGS)

Objects/MingW/PessEstimGraphFrm.o: $(GLOBALDEPS)
PessEstimGraphFrm.cpp
      $(CPP) -c PessEstimGraphFrm.cpp -o
Objects/MingW/PessEstimGraphFrm.o $(CXXFLAGS)

Objects/MingW/MonitorVisualFrm.o: $(GLOBALDEPS)
MonitorVisualFrm.cpp
      $(CPP) -c MonitorVisualFrm.cpp -o
Objects/MingW/MonitorVisualFrm.o $(CXXFLAGS)

Objects/MingW/HeaderReadingFrm.o: $(GLOBALDEPS)
HeaderReadingFrm.cpp
      $(CPP) -c HeaderReadingFrm.cpp -o
Objects/MingW/HeaderReadingFrm.o $(CXXFLAGS)

Objects/MingW/Util.o: $(GLOBALDEPS) Util.cpp

```

```
$(CPP) -c Util.cpp -o Objects/MingW/Util.o $(CXXFLAGS)

Objects/MingW/GraphPlot.o: $(GLOBALDEPS) GraphPlot.cpp
    $(CPP) -c GraphPlot.cpp -o Objects/MingW/GraphPlot.o
$(CXXFLAGS)

Objects/MingW/Tukey.o: $(GLOBALDEPS) Tukey.cpp
    $(CPP) -c Tukey.cpp -o Objects/MingW/Tukey.o $(CXXFLAGS)

Objects/MingW/MonitorVisualApp.o: $(GLOBALDEPS)
MonitorVisualApp.cpp
    $(CPP) -c MonitorVisualApp.cpp -o
Objects/MingW/MonitorVisualApp.o $(CXXFLAGS)

Objects/MingW/mathplot.o: $(GLOBALDEPS) mathplot.cpp
    $(CPP) -c mathplot.cpp -o Objects/MingW/mathplot.o
$(CXXFLAGS)
```

# Apêndice D

## Rotinas de Processamento

Neste ponto são apresentadas as principais rotinas do sistema.

### D.1 - Leitura de cabeçalho de arquivo \*.peb

```
void PessEstimation::readExamFile(unsigned long delay,
                                   double fe,
                                   bool estimatePess,
                                   bool saveUnstimulatedPeriod)
{
    //It opens and reads the file, after sets the values
    FILE *file;
    short int **point;
    unsigned int length;

    //It converts wxString to String
    char pebFileString[pebFile.Len()+1];

    for (unsigned int i = 0; i < (pebFile.Len()); i++)
    {
        pebFileString[i] = pebFile.GetChar(i);
    }
    pebFileString[pebFile.Len()] = '\\0';

    file=fopen(pebFileString,"r+b");

    fread(&nCh,1,1,file);
    nChar=(int)nCh;

    if (nChar!=0)
    {
        typeExam=new char[EXAM_TYPE_MAX_LENGTH];
        fread(typeExam,1,nChar,file);    //Read the type
of exam
        for(int i = nChar; i<=EXAM_TYPE_MAX_LENGTH; i++)
        {
            typeExam[i]='\\0';
        }
    }
    else
    {
        typeExam=new char;
```

```

        typeExam[0]='\0';
    }

    fread(&fs,4,1,file);
    fread(&nCh2,1,1,file);
    nChar2=(int)nCh2;
    fread(&timeUnity,1,nChar2,file);
    fread(&nChannel,1,1,file);
    nChannels=(int)nChannel;

    cAng=new double[nChannels];

    for(unsigned int i=0; i<nChannels; i++)
    {
        fread(&cAng[i],8,1,file);
    }

    cLin=new double[nChannels];
    for(unsigned int i=0; i<nChannels; i++)
    {
        fread(&cLin[i],8,1,file);
    }

    chName=new char*[nChannels];

    for (unsigned int i=0; i<nChannels;i++)
    {
        fread(&len,1,1,file);
        length=(int)len;
        chName[i]=new char[length+1];
        fread(chName[i],1,length,file);
        chName[i][length]='\0';
    }

    chUnity=new char*[nChannels];

    for (unsigned int i=0;i<nChannels;i++)
    {
        fread(&len,1,1,file);
        length=(int)len;
        chUnity[i]=new char[length+1];
        fread(chUnity[i],1,length,file);
        chUnity[i][length]='\0';
    }

    fread(&nCh2,1,1,file);
    nChAnot = (int)nCh2;

    nTotal = nChannels + nChAnot;
    fread(&lowCutOff,4,1,file);
    fread(&highCutOff,4,1,file);

```

```

fread(&yearExam, 2, 1, file);
fread(&month, 1, 1, file);
monthExam=(int)month;
fread(&day, 1, 1, file);
dayExam=(int)day;
fread(&hour, 1, 1, file);
hourExam=(int)hour;
fread(&minute, 1, 1, file);
minuteExam=(int)minute;

fread(&nCh2, 1, 1, file);
nComment=(int)nCh2;

if (nComment!=0)
{
    comment=new char[nComment];
    fread(comment, 1, nComment, file);
}
else
{
    comment=new char;
    comment[0]='\0';
}

//windowing the exam (only if it is necessary to
//estimate PESS)
if (estimatePess)
{
    point = new short int*[nTotal];

    pointsQt = (unsigned int)(fs/fe);

    peakDetector(point, file, delay);
}

fclose(file);

if (estimatePess)
    removeNormalizedForm();
}

```

## D.2 - Separação do sinal em épocas

```
void PessEstimation::peakDetector(short int **point,
                                  FILE *file, unsigned long int delay)
{
    short int **pointAux;
    short int difference=0;
    short int ***completePointWindowedCopy;
    unsigned int epochsQt,
    unstimulatedEpochsQt, unstimulatedEpochsQtLater, unstimulated
    EpochsQtBefore;
    unsigned int firstStimulationEpoch;
    unsigned long int unstimulatedPointsQt=0;
    int begin;
    unsigned long int size = 0;
    unsigned long int counter = pointsQt*SIZE_FACTOR;

    epochsTotal = stimulatedEpochsTotal =
    unstimulatedEpochsQtBefore = unstimulatedEpochsQtLater = 0;

    point = new short int*[nTotal];

    for (unsigned int i=0; i<nTotal; i++)
    {
        point[i] = new short int [counter];
    }

    while (!feof(file))
    {
        if(size < counter)
        {
            for (unsigned int i=0; i<nTotal; i++)
                fread(&point[i][size], 2, 1, file);

            difference = point[nTotal-1][size]-
            point[nTotal-1][size-1];

            if (difference > LIMAR_PEAK_DETECTION)
                stimulatedEpochsTotal++;

            size++;
        }
        else
        {
            pointAux = point;
            point = new short int*[nTotal];
            counter *= 2;

            for (unsigned int i=0; i<nTotal; i++)
                point[i] = new short int [counter];
        }
    }
}
```

```

        for (unsigned int i=0; i<nTotal; i++)
        {
            for (unsigned int j=0; j<(counter/2); j++)
                point[i][j] = pointAux[i][j];
        }

        for (unsigned int i = 0; i<nTotal; i++)
            delete []pointAux[i];
        delete []pointAux;
    }

}
stimulatedEpochsTotal--;
epochsTotal = (unsigned int)size/pointsQt;

// separation into epochs
epochsQt=unstimulatedEpochsQt=0;
difference=0;
completePointWindowedCopy = new short int**[nTotal];
stimulatedPointWindowed = new short int **[nTotal];

for (unsigned int i=0;i<nTotal;i++)
{
    completePointWindowedCopy[i] = new short
int*[epochsTotal];
    stimulatedPointWindowed[i] = new short int
*[stimulatedEpochsTotal];
}

for (unsigned int i=0;i<nTotal;i++)
{
    for(unsigned int j=0;j<epochsTotal;j++)
    {
        completePointWindowedCopy[i][j]=new short
int[pointsQt];

        if(j<stimulatedEpochsTotal)
            stimulatedPointWindowed[i][j]=new short
int[pointsQt];
    }
}

for (unsigned int j=0;j<size;)
{
    difference = point[nTotal-1][j+1]-point[nTotal-
1][j];

    if (difference > LIMIAR_PEAK_DETECTION && epochsQt
< stimulatedEpochsTotal)
    {

```

```

        if(unstimulatedPointsQt != 0 &&
unstimulatedEpochsQt!=0)
        {
            unstimulatedPointsQt = 0;
            unstimulatedEpochsQt--;
            unstimulatedEpochsQtBefore =
unstimulatedEpochsQt;
        }
        for(unsigned int i=0;i<nTotal;i++)
        {
            begin=j+delay;
            for(unsigned int k=0;k<pointsQt;k++)
            {

completePointWindowedCopy[i][epochsQt+unstimulatedEpochsQt]
[k]=point[i][begin];

stimulatedPointWindowed[i][epochsQt][k]=point[i][begin];
                begin++;
            }
        }
        j+=pointsQt;

        if (epochsQt == 0)
            firstStimulationEpoch = epochsQt +
unstimulatedEpochsQt;

            epochsQt++;
        }
        else
        {
            if (epochsQt ==0 || epochsQt ==
stimulatedEpochsTotal)
            {
                for (unsigned int i=0; i<nTotal; i++)

                    completePointWindowedCopy[i][unstimulatedEpochsQt+epoc
hsQt][unstimulatedPointsQt] = point[i][j+delay];

                    unstimulatedPointsQt++;

                    if(unstimulatedPointsQt == pointsQt)
                    {
                        unstimulatedPointsQt = 0;
                        unstimulatedEpochsQt++;
                    }
                }
            }
        }
        j++;
    }
}

```



```

    unstimulatedEpochsQtLater = unstimulatedEpochsQt -
unstimulatedEpochsQtBefore;

    //this part is only to keep simetry in the graph, but
    //in a online situation, this may not be executed
    //since, in this case, completePointWindowed =
    //completePointWindowedCopy
    completePointWindowed = new short int **[nTotal];

    if (unstimulatedEpochsQtBefore >
unstimulatedEpochsQtLater)
        unstimulatedEpochsQt = unstimulatedEpochsQtLater;
    else
        unstimulatedEpochsQt = unstimulatedEpochsQtBefore;

    for (unsigned int i = 0; i<nTotal; i++)
    {
        completePointWindowed[i] = new short int*
[stimulatedEpochsTotal+2*unstimulatedEpochsQt];
    }

    for (unsigned int i = 0; i < nTotal; i++)
        for (unsigned int j = 0; j <
(stimulatedEpochsTotal+2*unstimulatedEpochsQt); j++)
            completePointWindowed[i][j] = new short int
[pointsQt];

    //unstimulated period before stimulation
    for (unsigned int i = 0; i < nTotal; i++)
        for (unsigned int j = 0; j < unstimulatedEpochsQt;
j++)
            for (unsigned int p = 0; p<pointsQt;p++)
                completePointWindowed[i][j][p] =
completePointWindowedCopy[i][j][p];

    //stimulated period
    for (unsigned int i = 0; i < nTotal; i++)
        for (unsigned int j = 0; j < stimulatedEpochsTotal;
j++)
            for (unsigned int p = 0; p<pointsQt;p++)

                completePointWindowed[i][j+unstimulatedEpochsQt][p] =
completePointWindowedCopy[i][j+firstStimulationEpoch][p];

    //unstimulated period before stimulation
    for (unsigned int i = 0; i < nTotal; i++)
        for (unsigned int j = 0; j < unstimulatedEpochsQt;
j++)
            for (unsigned int p = 0; p<pointsQt;p++)

                completePointWindowed[i][j+unstimulatedEpochsQt+stimul

```

```

atedEpochsTotal][p] =
completePointWindowedCopy[i][j+firstStimulationEpoch+stimul
atedEpochsTotal][p];

    for (unsigned int i = 0; i < nTotal; i++)
        for (unsigned int j = 0; j < epochsTotal; j++)
            delete [] completePointWindowedCopy[i][j];

    for (unsigned int i = 0; i < nTotal; i++)
        delete [] completePointWindowedCopy[i];

    delete [] completePointWindowedCopy;

    epochsTotal =
stimulatedEpochsTotal+2*unstimulatedEpochsQt;

    for (unsigned int i=0; i<nTotal; i++)
        delete [] point[i];

    delete []point;
}

```

### D.3 - Remoção da forma normalizada

```
void PessEstimation::removeNormalizedForm()
{
    completeExam = new double **[nTotal];
    stimulatedExam = new double**[nTotal];

    for(unsigned int i=0;i<nTotal;i++)
    {
        completeExam[i]=new double*[epochsTotal];
        stimulatedExam[i]=new
double*[stimulatedEpochsTotal];
    }

    for(unsigned int i=0;i<nTotal;i++)
    {
        for(unsigned int j=0;j<epochsTotal;j++)
        {
            completeExam[i][j]=new double[pointsQt];

            if (j<stimulatedEpochsTotal)
            {
                stimulatedExam[i][j] = new
double[pointsQt];
            }
        }
    }

    for(unsigned int i=0;i<nTotal;i++)
    {
        for(unsigned int j=0;j<pointsQt;j++)
        {
            for(unsigned int w=0;w<epochsTotal;w++)
            {

                completeExam[i][w][j]=(completePointWindowed[i][w][j]-
cLin[i])*cAng[i];

                if (w <stimulatedEpochsTotal)
                    stimulatedExam[i][w][j] =
(stimulatedPointWindowed[i][w][j]-cLin[i])*cAng[i];
            }
        }
    }

    // free memory
    for(unsigned int i=0;i<nTotal;i++)
    {
        for(unsigned int w=0;w<epochsTotal;w++)
        {
```

```
        delete []completePointWindowed[i][w];
        if(w < stimulatedEpochsTotal)
            delete []stimulatedPointWindowed[i][w];
    }
}

for (unsigned int i=0;i<nTotal;i++)
{
    delete [] completePointWindowed[i];
    delete [] stimulatedPointWindowed[i];
}

delete []completePointWindowed;
delete []stimulatedPointWindowed;
}
```

## D.4 - Pré-processamento

```
void PessEstimation::preProcessing(
    bool applyDtrend,
    bool applyArtifactRejection,
    bool applyTukey,
    double *Tukey,
    unsigned int channel,
    unsigned int epochsOfExam,
    double*** exam,
    unsigned int epochsSelected,
    unsigned int epochsDelay,
    wxString *epochsRejected,
    double **signalToProcess)
//exam is the input and signalToProcess is the output to be
//processed in other steps, considering the selected
//channel
{
    unsigned int countRejected = 0;
    char * out = new char[10];

    if (applyArtifactRejection)
    {
        unsigned int epochsCounter = 0;
        for (unsigned int i=0; i < epochsOfExam &&
epochsCounter < epochsSelected; i++)
        {
            if (artifactRejection(channel, i+epochsDelay,
exam))
            {
                for (unsigned int j=0; j<pointsQt; j++)
                    signalToProcess[epochsCounter][j] =
exam[channel][i+epochsDelay][j];

                if (applyDtrend)
Dtrend(signalToProcess[epochsCounter],pointsQt);

                if (applyTukey)
                    for (unsigned int j=0; j<pointsQt;
j++)
signalToProcess[epochsCounter][j] *= Tukey[j];

                epochsCounter++;
            }
            else
                countRejected++;
        }
    }
    else
```

```

    {
        unsigned int epochsCounter = 0;
        for (unsigned int i=0; i < epochsOfExam &&
epochsCounter < epochsSelected; i++)
            {
                for (unsigned int j=0; j<pointsQt; j++)
                    signalToProcess[epochsCounter][j] =
exam[channel][i+epochsDelay][j];

                    if (applyDtrend)

Dtrend(signalToProcess[epochsCounter],pointsQt);

                    if (applyTukey)
                        for (unsigned int j=0; j<pointsQt; j++)
                            signalToProcess[epochsCounter][j] *=
Tukey[j];

                        epochsCounter++;
                    }
            }

        sprintf(out, "%d", countRejected);
        *epochsRejected = wxString::FromAscii(out);
    }

```

## D.5 - Remoção automática de artefatos

```
bool PessEstimation::artifactRejection(
    unsigned int channel,
    int epoch,
    double ***exam)
{
    double
    percentageBefore, percentageGeneral, counterBefore,
    counterGeneral;
    bool beforeHappened;

    beforeHappened=false;
    counterBefore=1.0;
    counterGeneral=0.0;

    for (unsigned int index = 0; index < pointsQt; index++)
    {
        if(3*sd[channel]<fabs(exam[channel][epoch][index]))
        {
            counterGeneral = counterGeneral+1.0;

            if(beforeHappened)
            {
                counterBefore = counterBefore + 1.0;
                percentageBefore =
double(counterBefore/pointsQt);
                if(percentageBefore >=
LIMIAR_BEFORE_REJECTION)
                    return false;

            }
            else
                counterBefore=1.0;

percentageGeneral=double(counterGeneral/pointsQt);

            if(percentageGeneral >= LIMIAR_GENERAL_REJECTION
|| percentageBefore >= LIMIAR_BEFORE_REJECTION)
                return false;

            beforeHappened = true;
        }
        else
            beforeHappened = false;
    }
    return true;
}
```

## D.6 - Janela Tukey

```
void Tukey::TukeyWindow(float Fs, double tzeroIn,
                       double trising, double twindow)
{
    double *tukeyWindowAux, r, samples;
    int samplesAux, k, tzero;

    samplesAux = (int)(twindow*Fs*0.001);

    tzero = (int)round(tzeroIn*Fs*0.001);

    samplesAux = samplesAux-2*tzero;
    samples = samplesAux;

    r = trising/twindow;
    r = r*2;
    tukeyWindowAux = new double[samplesAux];
    tukeyWindow = new double[(samplesAux+2*tzero+1)];

    //Computation of the window
    for(k=1; k<=samplesAux; k++)
    {
        if (k<(((r/2)*(samples-1))+1))
        {
            tukeyWindowAux[k]=0.5*(1+cos(((2*(PI/r))*((k-
1)/(samples-1)))-PI));
        }
        if((((r/2)*(samples-1))+1)<=k && k<=(samples-
(0.5*r*(samples-1))))
        {
            tukeyWindowAux[k]=1;
        }
        if (k>(samples-(0.5*r*(samples-1))))
        {
            tukeyWindowAux[k]=0.5*(1+cos((2*PI/r)-
((2*PI/r)*((k-1)/(samples-1)))-PI));
        }
    }

    for(k=0; k<tzero; k++)
        tukeyWindow[k]=0;

    for(k=1; k<=samplesAux; k++)
        tukeyWindow[k+tzero] = tukeyWindowAux[k];

    for(k=(int)(samples+tzero+1); k<=(samples+2*tzero); k++)
        tukeyWindow[k]=0;
}
```



## D.7 - Média Coerente

```
void PessEstimation::coherentAverage(bool applyDtrend,
                                     bool applyArtifactRejection,
                                     bool applyTukey,
                                     double *Tukey,
                                     unsigned int channel,
                                     unsigned int epochsSelected,
                                     unsigned int epochsOfExam,
                                     unsigned int epochsDelay,
                                     double *** exam,
                                     wxString *epochsRejected,
                                     double **&average)
{
    double **signalToProcess;

    signalToProcess = new double *[epochsSelected];

    for (unsigned int i = 0; i < epochsSelected; i++)
        signalToProcess [i] = new double [pointsQt];

    preProcessing(applyDtrend, applyArtifactRejection,
                  applyTukey,Tukey, channel, epochsOfExam, exam,
                  epochsSelected, epochsDelay,
                  epochsRejected, signalToProcess);

    //allocating memory
    average = new double*[AXES_QUANTITY];

    for (unsigned int index = 0; index < AXES_QUANTITY;
index++)
        average[index] = new double[pointsQt];

    for (unsigned int index = 0; index < pointsQt; index++)
    {
        average[X_AXIS][index]=(index/fs)*TIME_SCALE_MS;
        average[Y_AXIS][index] = 0;
    }

    for (unsigned int epoch = 0; epoch < epochsSelected;
epoch++)
        for (unsigned int index = 0; index < pointsQt;
index++)
            average[Y_AXIS][index] +=
signalToProcess[epoch][index];

    for (unsigned int index = 0; index < pointsQt; index++)
        average[Y_AXIS][index] =
average[Y_AXIS][index]/(epochsSelected);

    if (applyDtrend)
```

```
        Dtrend(average[Y_AXIS, pointsQt]);  
    for (unsigned int i=0; i < epochsSelected; i++)  
        delete []signalToProcess[i];  
  
    delete [] signalToProcess;  
  
}
```

## D.8 - Remoção de nível médio

```
void PessEstimation::Dtrend(double *vector,
                           int pointsNumber)
{
    double averageValue;
    averageValue = 0.0;

    for(int index = 0; index < pointsNumber; index++)
    {
        averageValue = averageValue + vector[index];
    }
    averageValue = averageValue/pointsNumber;

    for(int index = 0; index < pointsNumber; index++)
    {
        vector[index] = vector[index] - averageValue;
    }
}
```

## D.9 - FFT

```
void Util::fftFunction(double **&data,
                      unsigned long length, double **&fft,
                      int isign)
{
    //preparing the entry
    unsigned long tempLength=0;

    //variables for the fft
    unsigned long n, mmax, m, j, istep, i;
    double wtemp, wr, wpr, wpi, wi, theta, tempr, tempi;

    for (unsigned int i = 0; tempLength < length; i++)
    {
        tempLength = pow(2.0, i);
    }

    tempLength = 2*tempLength;

    fft = new double*[AXES_QUANTITY];

    for(unsigned int i=0; i < AXES_QUANTITY; i++)
    {
        fft[i] = new double[tempLength];
    }

    double T = data[X_AXIS][1];
    for (unsigned int i=0; i<tempLength;i++)
    {
        fft[X_AXIS][i] = (i/(tempLength*T))*TIME_SCALE_MS;
        i++;
        fft[X_AXIS][i] = fft[X_AXIS][i-1];
    }

    for(unsigned int i=0; i<tempLength; i++)
    {
        if (i < length)
        {
            fft[Y_AXIS][2*i] = data[Y_AXIS][i];
            fft[Y_AXIS][2*i+1] = 0;
        }
        else
            fft[Y_AXIS][i] = 0;

        if(i+1 == length)
        {
            i = 2*i+1;
        }
    }
}
```

```

//end preparing the entry

    //the complex array is real+complex so the array
    //has a size n = 2* number of complex samples
    //real part is the data[index] and
    //the complex part is the data[index+1]

    //begin of bit-reversal algoritm
    n = tempLength;
    j=0;
    for (i=0;i<n/2;i+=2)
    {
        if (j > i)
        {
            SWAP(fft[1][j],fft[1][i]);
            SWAP(fft[1][j+1],fft[1][i+1]);
            if((j/2)<(n/4))
            {
                SWAP(fft[1][(n-(i+2))],fft[1][(n-
(j+2))]));
                SWAP(fft[1][(n-(i+2))+1],fft[1][(n-
(j+2))+1]));
            }
        }
        m = n >> 1;
        while (m >= 2 && j >= m) {
            j -= m;
            m >>= 1;
        }
        j += m;
    }

//end of bit-reversal algoritm

//Danielson-Lanzcos routine
mmax=2;
while (n > mmax)
{
    istep=mmax << 1;
    theta=isign*(2*PI/mmax);
    wtemp=sin(0.5*theta);
    wpr = -2.0*wtemp*wtemp;
    wpi=sin(theta);
    wr=1.0;
    wi=0.0;
    for (m=1;m<mmax;m+=2)
    {
        for (i=m;i<=n;i+=istep)
        {
            j=i+mmax;
            tempr=wr*fft[1][j-1]-wi*fft[1][j];

```

```

        tempi=wr*fft[1][j]+wi*fft[1][j-1];
        fft[1][j-1]=fft[1][i-1]-tempr;
        fft[1][j]=fft[1][i]-tempi;
        fft[1][i-1] += tempr;
        fft[1][i] += tempi;
    }
    wr=(wtemp=wr)*wpr-wi*wpi+wr;
    wi=wi*wpr+wtemp*wpi+wi;
}
mmax=istep;
}
//end of the routine
}

```

## D.10 - MSC

```
unsigned long PessEstimation::msc(unsigned int channel,
                                unsigned int epochsOfExam,
                                unsigned int epochsSelected,
                                unsigned int epochsDelay,
                                double ***exam,
                                double **&mscResult,
                                bool applyArtifactRejection,
                                bool applyTukey,
                                double * Tukey,
                                bool applyDtrend,
                                wxString *epochsRejected)
{
    unsigned long tempLength=0;
    double ***dataAux;
    double ***fftAux;
    double **fftSum;
    double **moduleFftSum;
    double ***moduleFftAux;
    double **sumModuleFft;
    double **fft;
    double **signalToProcess;
    signalToProcess = new double *[epochsSelected];

    for (unsigned int i = 0; i < epochsSelected; i++)
        signalToProcess [i] = new double [pointsQt];

    preProcessing(applyDtrend, applyArtifactRejection,
applyTukey,Tukey, channel, epochsOfExam, exam,
epochsSelected, epochsDelay,
epochsRejected, signalToProcess);

    for (unsigned int i=0;tempLength<pointsQt;i++)
        tempLength=pow(2.0,i);

    tempLength = 2*tempLength;

    //Calculating the magnitude of sum of epochs
    dataAux=new double**[epochsSelected];

    for(unsigned int i=0;i<epochsSelected;i++)
        dataAux[i]=new double*[AXES_QUANTITY];

    for(unsigned int i=0;i<epochsSelected;i++)
        for(unsigned int j=0;j<AXES_QUANTITY;j++)
            dataAux[i][j]=new double[pointsQt];

    for(unsigned int i = 0; i<(epochsSelected);i++)
        for(unsigned int j=0;j<pointsQt;j++)
```

```

        {
            dataAux[i][X_AXIS][j]=(j/fs)*TIME_SCALE_MS;
            dataAux[i][Y_AXIS][j]=signalToProcess[i][j];
        }
fftAux=new double**[epochsSelected];

for(unsigned int i=0;i<epochsSelected;i++)
    fftAux[i]=new double*[AXES_QUANTITY];

for(unsigned int i=0;i<epochsSelected;i++)
    for(unsigned int j=0;j<AXES_QUANTITY;j++)
        fftAux[i][j]=new double[tempLength];

for(unsigned int i=0;i<epochsSelected;i++)
{
    Util util;
    util.fftFunction(dataAux[i], pointsQt,fft,1);

    for(unsigned j=0;j<tempLength;j++)
    {
        fftAux[i][X_AXIS][j]=fft[X_AXIS][j];
        fftAux[i][Y_AXIS][j]=fft[Y_AXIS][j];
    }
}

//Calculating the sum of fft's
fftSum=new double*[AXES_QUANTITY];

for(unsigned int j=0;j<AXES_QUANTITY;j++)
    fftSum[j]=new double[tempLength];

for(unsigned j=0;j<tempLength;j++)
{
    fftSum[X_AXIS][j]=fftAux[0][X_AXIS][j];
    fftSum[Y_AXIS][j]=0;
}

for(unsigned int i=0;i<epochsSelected;i++)
    for(unsigned int j=0;j<tempLength;j++)

fftSum[Y_AXIS][j]=fftSum[Y_AXIS][j]+fftAux[i][Y_AXIS][j];

//Calculating squared module of fftSum
moduleFftSum=new double*[AXES_QUANTITY];

for(unsigned int j=0;j<AXES_QUANTITY;j++)
    moduleFftSum[j]=new double[tempLength/2];

for(unsigned int i=0;i<(tempLength/2);i++)
{
    moduleFftSum[X_AXIS][i]=fftSum[X_AXIS][2*i];
}

```



```

moduleFftSum[Y_AXIS][i]=((fftSum[Y_AXIS][2*i])*(fftSum[Y_AXIS][2*i]))+((fftSum[Y_AXIS][(2*i)+1])*(fftSum[Y_AXIS][(2*i)+1]));
}

// Calculating the module of each fft
moduleFftAux=new double**[epochsSelected];

for(unsigned int i=0;i<epochsSelected;i++)
    moduleFftAux[i]=new double*[AXES_QUANTITY];

for(unsigned int i=0;i<epochsSelected;i++)
    for(unsigned int j=0;j<AXES_QUANTITY;j++)
        moduleFftAux[i][j]=new double[(tempLength/2)];

for(unsigned int i=0;i<epochsSelected;i++)
    for(unsigned int j=0;j<(tempLength/2);j++)
    {

moduleFftAux[i][X_AXIS][j]=fftAux[i][X_AXIS][2*j];

moduleFftAux[i][Y_AXIS][j]=((fftAux[i][Y_AXIS][2*j])*(fftAux[i][Y_AXIS][2*j]))+((fftAux[i][Y_AXIS][(2*j)+1])*(fftAux[i][Y_AXIS][(2*j)+1])));
    }

//Calculating the sum of modules
sumModuleFft=new double*[AXES_QUANTITY];

for(unsigned int i=0;i<AXES_QUANTITY;i++)
    sumModuleFft[i]=new double[tempLength/2];

for(unsigned int i=0;i<(tempLength/2);i++)
    {
        sumModuleFft[X_AXIS][i]=moduleFftAux[0][X_AXIS][i];
        sumModuleFft[Y_AXIS][i]=0;
    }

for(unsigned int i=0;i<epochsSelected;i++)
    for(unsigned int j=0;j<(tempLength/2);j++)

sumModuleFft[Y_AXIS][j]=sumModuleFft[Y_AXIS][j]+moduleFftAux[i][Y_AXIS][j];

// Calculating mscLastValue

for(unsigned int i=0;i<(tempLength/2);i++)

sumModuleFft[Y_AXIS][i]=epochsSelected*sumModuleFft[Y_AXIS][i];

```

```

mscResult=new double*[AXES_QUANTITY];

for(unsigned int i=0;i<AXES_QUANTITY;i++)
    mscResult[i]=new double[tempLength/2];

for(unsigned int i=0;i<tempLength/2;i++)
{
    mscResult[X_AXIS][i]=sumModuleFft[X_AXIS][i];
mscResult[Y_AXIS][i]=(moduleFftSum[Y_AXIS][i])/(sumModuleFft[Y_AXIS][i]);
}

for (unsigned int i = 0; i<epochsSelected; i++)
    for (unsigned int j = 0; j < AXES_QUANTITY; j++)
    {
        delete []dataAux[i][j];
        delete []fftAux[i][j];
    }

for (unsigned int i = 0; i<epochsSelected; i++)
{
    delete []dataAux[i];
    delete []fftAux[i];
}

delete [] dataAux;
delete [] fftAux;

for (unsigned int i = 0; i < AXES_QUANTITY; i++)
{
    delete []fftSum[i];
    delete []moduleFftSum[i];
    delete []moduleFftAux[i];
    delete []sumModuleFft[i];
    delete []fft[i];
}
delete []fftSum;
delete []moduleFftSum;
delete []moduleFftAux;
delete []sumModuleFft;
delete []fft;
return tempLength;
}

```

## D.11 - Evolução temporal da MSC

```
unsigned long PessEstimation::tempEvolMsc(bool applyDtrend,
    bool artifactRejection,
    bool applyTukey,
    double *Tukey,
    unsigned int channel,
    unsigned int epochs,
    double monFrequency,
    wxString *epochsRejected,
    double **&tempEvolutionMscOnSelectedChannel)
{
    unsigned long tempLength=0;
    unsigned int indexMonFrequency;
    Util util;
    double **mscResult;
    unsigned long numberOfEstimatives = epochsTotal -
epochs;

    //allocating memory
    tempEvolutionMscOnSelectedChannel = new double
*[AXES_QUANTITY];

    for (unsigned int index = 0; index < AXES_QUANTITY;
index++)
        tempEvolutionMscOnSelectedChannel[index] = new
double [numberOfEstimatives];

    //first estimative
    tempLength = msc(channel, epochsTotal, epochs, 0,
completeExam, mscResult, artifactRejection, applyTukey,
Tukey, applyDtrend, epochsRejected);

    //find the closest frequency value to monFrequency
    indexMonFrequency =
util.findClosestValue(mscResult[X_AXIS], monFrequency,
pointsQt);

    // saving first estimative
    tempEvolutionMscOnSelectedChannel[X_AXIS][0] = 0;
    tempEvolutionMscOnSelectedChannel[Y_AXIS][0] =
mscResult[Y_AXIS][indexMonFrequency];

    for (unsigned int estimativeCounter = 1;
estimativeCounter < numberOfEstimatives; estimativeCounter
++)
    {

        for (unsigned int i=0; i< AXES_QUANTITY; i++)
        {
            delete []mscResult[i];
```

```

    }
    delete []mscResult;

    tempLength = msc(channel,epochsTotal, epochs,
estimativeCounter, completeExam,
                    mscResult, artifactRejection,
applyTukey, Tukey, applyDtrend,epochsRejected);

tempEvolutionMscOnSelectedChannel[X_AXIS][estimativeCounter
] = estimativeCounter;

tempEvolutionMscOnSelectedChannel[Y_AXIS][estimativeCounter
] = mscResult[Y_AXIS][indexMonFrequency];
    }

    return numberOfEstimates;
}

```

## D.12 - Evolução temporal do PESS

```
void PessEstimation::tempEvolMorphology (bool applyDtrend,
                                         bool artifactRejection,
                                         bool applyTukey,
                                         double *Tukey,
                                         unsigned int channel,
                                         unsigned int epochs,
                                         unsigned int estimative,
                                         double initialInstantP37,
                                         double finalInstantP37,
                                         double initialInstantN45,
                                         double finalInstantN45,
                                         wxString *epochsRejected,
                                         double **&p37TimeOccurrences,
                                         double **&p37AmplitudeOccurrences,
                                         double **&n45TimeOccurrences,
                                         double **&n45AmplitudeOccurrences)
{
    Util util;
    double **average;

    p37TimeOccurrences = new double * [AXES_QUANTITY];
    p37AmplitudeOccurrences = new double * [AXES_QUANTITY];
    n45TimeOccurrences = new double * [AXES_QUANTITY];
    n45AmplitudeOccurrences = new double * [AXES_QUANTITY];

    for (unsigned int index = 0; index < AXES_QUANTITY;
index++)
    {
        p37TimeOccurrences[index] = new double
[estimative];
        p37AmplitudeOccurrences[index] = new double
[estimative];
        n45TimeOccurrences[index] = new double
[estimative];
        n45AmplitudeOccurrences[index] = new double
[estimative];
    }

    for (unsigned int estimativeCounter = 0;
estimativeCounter < estimative; estimativeCounter ++)
    {
        unsigned int initialIndexP37, initialIndexN45;
        unsigned int finalIndexP37, finalIndexN45;
        unsigned int p37Index, n45Index;

        coherentAverage(applyDtrend, artifactRejection,
applyTukey, Tukey, channel, epochs, epochsTotal,
                        estimativeCounter, stimulatedExam,
epochsRejected, average);
    }
}
```

```

        initialIndexP37 =
util.findClosestValue(average[X_AXIS], initialInstantP37,
pointsQt);
        initialIndexN45 =
util.findClosestValue(average[X_AXIS], initialInstantN45,
pointsQt);
        finalIndexP37 =
util.findClosestValue(average[X_AXIS], finalInstantP37,
pointsQt);
        finalIndexN45 =
util.findClosestValue(average[X_AXIS], finalInstantN45,
pointsQt);

        p37Index = initialIndexP37;
        n45Index = initialIndexN45;

        for (unsigned int index = initialIndexP37+1; index
< finalIndexP37; index++)
            if (average[Y_AXIS][index] <
average[Y_AXIS][p37Index])
                p37Index = index;

        for (unsigned int index = initialIndexN45+1; index
< finalIndexN45; index++)
            if (average[Y_AXIS][index] >
average[Y_AXIS][n45Index])
                n45Index = index;

        p37TimeOccurrences[X_AXIS][estimativeCounter] =
p37AmplitudeOccurrences[X_AXIS][estimativeCounter] =
estimativeCounter;
        p37AmplitudeOccurrences[Y_AXIS][estimativeCounter]
= average[Y_AXIS][p37Index];
        p37TimeOccurrences[Y_AXIS][estimativeCounter] =
average[X_AXIS][p37Index];

        n45TimeOccurrences[X_AXIS][estimativeCounter] =
n45AmplitudeOccurrences[X_AXIS][estimativeCounter] =
estimativeCounter;
        n45AmplitudeOccurrences[Y_AXIS][estimativeCounter]
= average[Y_AXIS][n45Index];
        n45TimeOccurrences[Y_AXIS][estimativeCounter] =
average[X_AXIS][n45Index];
    }
}

```

## D.13 - Definição da classe principal do sistema

```
class PessEstimation
{
    private:
        unsigned int epochsTotal, stimulatedEpochsTotal;
        unsigned int
nChar, nChannels, nChar2, nChAnot, nTotal, monthExam, dayExam;
        unsigned int hourExam, minuteExam, nComment, pointsQt;
        short unsigned int yearExam;
        char month, day, hour, minute, *comment;
        char
nCh, *typeExam, timeUnity, nChannel, nCh2, **chName, len, **chUnit
Y;
        float fs, highCutOff, lowCutOff;
        double
*cAng, *cLin, ***completeExam, ***stimulatedExam, *sd;
        wxString pebFile;
        wxString sdFile;
        short int ***completePointWindowed,
***stimulatedPointWindowed;
        double kCriticalSquared;

    public:
        PessEstimation();
        virtual ~PessEstimation();
        unsigned int getEpochsTotal();
        unsigned int getPointsQt();
        void setPebFile(wxString file);
        void setSdFile(wxString file);
        double *** getExamPoints();
        wxString getPebFile();
        wxString getSdFile();
        wxString getTypeExam();
        float getHighCutOff();
        float getLowCutOff();
        float getFs();
        double* getCAng();
        double* getCLin();
        unsigned int getNTotal();
        wxString getRealizationTime();
        wxString getRealizationDate();
        wxString getComments();
        unsigned int getNChannels();
        char** getChName();
        void readExamFile();
        void readExamFile(unsigned long delay, double fe,
bool estimatePess, bool saveUnstimulatedPeriod);
        void readSdFile();
        void coherentAverage(bool applyDtrend, bool
artifactRejection, bool applyTukey, double *Tukey,
```

```

        unsigned int channel, unsigned
int epochsSelected, unsigned int epochsOfExam, unsigned int
epochsDelay,
        double *** exam, wxString
*epochsRejected, double **&average);
        unsigned long msc(unsigned int channel, unsigned
int epochsOfExam, unsigned int epochsSelected,
        unsigned int epochsDelay,
double ***exam, double **average,
        double **&mscResult, bool
applyArtifactRejection, bool applyTukey,
        double * Tukey);
        void setKCriticalSquared(double alpha, unsigned int
epochs);
        double getKCriticalSquared();
        unsigned long tempEvolMsc (bool applyDtrend, bool
artifactRejection, bool applyTukey, double *Tukey,
        unsigned int channel,
unsigned int epochs,
        double monFrequency,
wxString *epochsRejected, double
**&tempEvolutionMscOnSelectedChannel);
        void tempEvolMorphology (bool applyDtrend, bool
artifactRejection, bool applyTukey, double *Tukey,
        unsigned int channel,
unsigned int epochs, unsigned int estimative,
        double
initialInstantP37, double finalInstantP37,
        double
initialInstantN45, double finalInstantN45,
        wxString
*epochsRejected, double **&p37TimeOccurrences,
        double
**&p37AmplitudeOccurrences, double **&n45TimeOccurrences,
        double
**&n45AmplitudeOccurrences);

private:
        void peakDetector(short int **point, FILE *file,
unsigned long int delay);
        void removeNormalizedForm();
        bool artifactRejection(unsigned int channel, int
epoch, double ***exam);
        void Dtrend(double *vector, int pointsNumber);
        void preProcessing(bool applyDtrend, bool
applyArtifactRejection, bool applyTukey,
        double *Tukey, unsigned
int channel, unsigned int epochsOfExam, double*** exam,
        unsigned int
epochsSelected, unsigned int epochsDelay, wxString
*epochsRejected,

```



```
double  
**signalToProcess);  
};
```