



PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM
COMPUTACIONAL E TECNOLOGIA INDUSTRIAL
Doutorado em Modelagem Computacional e Tecnologia Industrial

Tese de doutorado

SmartChair: Cadeira de Rodas Inteligente com
Interface Flexível

Apresentada por: Oberdan Rocha Pinheiro
Orientadora: Lynn Rosalina Gama Alves
Co-orientador: Josemar Rodrigues de Souza

Dezembro de 2016

Oberdan Rocha Pinheiro

SmartChair: Cadeira de Rodas Inteligente com Interface Flexível

Tese de doutorado apresentada ao Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial, Curso de Doutorado em Modelagem Computacional e Tecnologia Industrial do SENAI CIMATEC, como requisito parcial para a obtenção do título de **Doutor em Modelagem Computacional e Tecnologia Industrial**.

Área de conhecimento: Interdisciplinar

Orientadora: Lynn Rosalina Gama Alves

Co-orientador: Josemar Rodrigues de Souza

SENAI CIMATEC

Salvador
SENAI CIMATEC
2016

Ficha catalográfica elaborada pela Biblioteca do Centro Universitário SENAI CIMATEC

P645s Pinheiro, Oberdan Rocha

Smartchair: cadeira de rodas inteligente com interface flexível / Oberdan Rocha
Pinheiro. – Salvador, 2017.

131 f. : il. color.

Orientadora: Prof^ª. Dr^ª. Lynn Rosalina Gama Alves.
Coorientador: Prof. PhD. Josemar Rodrigues de Souza.

Tese (Doutorado em Modelagem Computacional e Tecnologia Industrial) – Programa
de Pós-Graduação, Centro Universitário SENAI CIMATEC, Salvador, 2017.

Inclui referências.

1. Eeg motor moviment. 2. Eletroencefalograma. 3. Imagery dataset. 4. Interface
cérebro-computador. 5. Mobilidade reduzida. I. Centro Universitário SENAI CIMATEC. II.
Alves, Lynn Rosalina Gama. III. Souza, Josemar Rodrigues de. IV. Título.

CDD: 620. 00113

Nota sobre o estilo do PPGMCTI

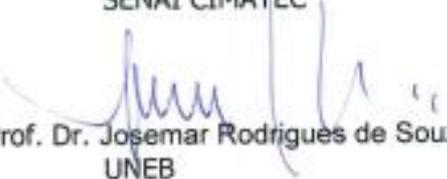
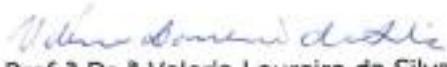
Esta tese de doutorado foi elaborada considerando as normas de estilo (i.e. estéticas e estruturais) propostas aprovadas pelo colegiado do Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial e estão disponíveis em formato eletrônico (*download* na Página Web http://ead.fieb.org.br/portal_faculdades/dissertacoes-e-teses-mcti.html ou solicitação via e-mail à secretaria do programa) e em formato impresso somente para consulta.

Ressalta-se que o formato proposto considera diversos itens das normas da Associação Brasileira de Normas Técnicas (ABNT), entretanto opta-se, em alguns aspectos, seguir um estilo próprio elaborado e amadurecido pelos professores do programa de pós-graduação supracitado.

Faculdade de Tecnologia SENAI CIMATEC

Doutorado em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, leu e aprovou a Tese de doutorado, intitulada "**SmartChair: cadeira de rodas inteligente com interface flexível**", apresentada no dia 16 de dezembro de 2016, como parte dos requisitos necessários para a obtenção do Título de Doutor em Modelagem Computacional e Tecnologia Industrial.

- Orientadora: 
Prof.^a Dr.^a Lynn Rosalina Gama Alves
SENAI CIMATEC
- Coorientador: 
Prof. Dr. Josemar Rodrigues de Souza
UNEB
- Membro Interno: 
Prof.^a Dr.^a Camila de Sousa Pereira-Guizzo
SENAI CIMATEC
- Membro Interno: 
Prof.^a Dr.^a Valeria Loureiro da Silva
SENAI CIMATEC
- Membro Externo: 
Prof.^a Dr.^a Esther Luna Colombini
UNICAMP
- Membro Externo: 
Prof.^a Dr.^a Tatiane Nogueira Rios
UFBA
-

Dedico este trabalho às pessoas com mobilidade reduzida portadoras de deficiências motoras graves.

Agradecimentos

Ao Prof. Ph. D. Josemar Rodrigues de Souza agradeço pela orientação científica, o incentivo constante, as sugestões, as condições proporcionadas para a realização deste trabalho e a confiança depositada ao longo dos últimos seis anos.

A Profa. Dra. Lynn Rosalina Gama Alves, pela orientação e paciência que dedicou a este trabalho, pelas sugestões valiosas, conduzindo, de forma coerente, a investigação e a elaboração desta tese de doutorado.

Aos professores do Programa de Pós-Graduação Stricto Sensu da Faculdade de Tecnologia SENAI CIMATEC, pelo incentivo e questionamentos, que foram fundamentais no processo da pesquisa e desenvolvimento desse trabalho.

Meus respeitosos agradecimentos aos membros da banca examinadora pela participação e contribuições na revisão deste trabalho de investigação.

À minha esposa e aos meus filhos, que, de forma grandiosa, me incentivaram a continuar, não importando quais fossem os problemas nos momentos difíceis dessa caminhada.

A todos os meus parentes e familiares, pelo apoio, incentivo e compreensão, ante a minha ausência em muitos momentos de confraternização.

Salvador, Brasil
16 Dezembro de 2016

Oberdan Rocha Pinheiro

Resumo

Na sociedade contemporânea, observa-se o aumento do número de pessoas diagnosticadas com algum tipo de deficiência física que possuem mobilidade reduzida, o que afeta sensivelmente a qualidade de vida e independência dessas pessoas. As atividades de controle do corpo realizadas pelo ser humano utilizam as vias neuromusculares e pessoas diagnosticadas com quadros clínicos como esclerose lateral amiotrófica, lesão na medula espinhal ou acidente vascular cerebral, por exemplo, têm as vias neuromusculares comprometidas, por esse motivo tem dificuldade em locomover-se. Uma das alternativas para contornar esse problema é o desenvolvimento de tecnologias que substituam, em parte, as funções perdidas das pessoas com deficiências motoras graves. Nesse contexto, as interfaces cérebro-computador, apresentam-se como solução para ajudar indivíduos com graves comprometimentos motores e neurológicos em suas atividades de vida diária. Entretanto, grande parte dessas interfaces é construída de forma isolada, possuindo estruturas próprias, impactando negativamente na capacidade de reuso, integração e extensão dessas interfaces. O objetivo geral desse trabalho de investigação é desenvolver um modelo computacional baseado em interface cérebro-computador que permita o estudo e desenvolvimento de novas tecnologias a partir dos sinais de eletroencefalograma para ajudar indivíduos com mobilidade reduzida a conduzir uma cadeira de rodas inteligente em suas atividades de vida diária. Os sinais de eletroencefalograma foram adquiridos através do banco de dados *eeegmidb - EEG Motor Movement/Imagery Dataset*. Amostras de sinais de eletroencefalograma de 106 indivíduos foram utilizadas para validação do modelo computacional. Os resultados alcançados nesse trabalho de investigação, evidenciam que a arquitetura proposta é viável e reusável para o desenvolvimento de interfaces cérebro-computador.

Palavras-chaves: Interface cérebro-computador, Cadeira de rodas inteligente, Eletroencefalograma.

Abstract

In contemporary society, the number of people diagnosed with some kind of physical disorder and who present motor impairment can be observed as it rises. These disorders affect considerably their life quality and their autonomy. Body control activities performed by human beings utilize neuromuscular traits. People diagnosed with clinical conditions such as Amyotrophic Lateral Sclerosis, Spinal Cord Lesion or Cerebrovascular Accident, for instance, present their neuromuscular traits compromised, and for that reason, they face difficulties regarding mobility. One of the alternatives to overcome that problem is the development of technologies that partially substitute the lost functions of the people with severe motor impairment. In this context the cerebro-computer interfaces, present themselves as a solution to aid individuals with severe motor and neurologic impairment of their daily activities. However, many of these interfaces are constructed isolatedly, thus having their own structures, what generates a negative impact in their reutilization capacity, integration and extension. The general aim of the present investigative paper is to develop a computational model based on a cerebro-computer interface which allows the study and the development of new technologies from electroencephalography signals in order to help individuals with motor impairment in conducting an intelligent wheelchair during their daily activities. Those signals were acquired through a database eegmmidb - EEG Motor Movement/Imagery Dataset. Samples of electroencephalography signals from 106 individuals were utilized in order to validate the computational model. The results obtained in this research work show that the proposed architecture is feasible and reusable for the development of brain-computer interfaces.

Keywords: Brain-Computer Interface, Intelligent wheelchair, Electroencephalography.

Sumário

1	Introdução	1
1.1	Definição do problema	2
1.2	Objetivo	5
1.3	Importância da pesquisa	6
1.4	Limites e limitações	6
1.5	Organização da Tese de doutorado	7
2	Interface cérebro-computador	8
2.1	Eletroencefalografia	10
2.2	Frequências dos sinais neurais	16
2.3	Potenciais evocados	18
2.4	Análise do sinal	20
2.5	Imaginação do movimento	21
2.6	Limites e possibilidades nas investigações sobre a interface cérebro-computador	22
2.7	Considerações finais	34
3	Projeto e desenvolvimento da pesquisa	37
3.1	Modelo computacional	37
3.1.1	Módulo de aquisição do sinal	38
3.1.2	Módulo de processamento do sinal	39
3.1.3	Módulo de acionamento do dispositivo	48
3.2	Projeto arquitetural	51
3.2.1	Camada de Integração	52
3.2.2	Camada Núcleo	53
3.2.3	Camada de Dispositivo	56
3.3	Implementação	57
3.3.1	Interface de acionamento	59
3.4	Considerações finais	62
4	Avaliação e resultados experimentais	64
4.1	Desempenho da RNA	65
4.1.1	Dados originados do eletrodo Fp1	65
4.1.2	Dados originados do eletrodo Fp2	70
4.2	Interface de acionamento	75
4.3	Integração do sistema	78
4.3.1	Validar camada de Integração	79
4.3.2	Validar camada Núcleo	79
4.3.3	Validar integração do sistema	80
5	Considerações finais	82
5.1	Conclusões	82
5.2	Contribuições	83
5.3	Atividades Futuras de Pesquisa	84
	Referências	86

A	Physionet EEG Motor Movement/Imagery Dataset	95
B	Class FastFourierTransformer	99
B.1	Constructor Summary	99
B.2	Method Summary	100
B.2.1	transformInPlace	100
B.2.2	transform	100
B.2.3	transform	101
B.2.4	transform	101
C	Class MultilayerPerceptron	102
C.1	Constructor Summary	103
C.2	Method Summary	103
C.2.1	main	103
C.2.2	setDecay	104
C.2.3	getDecay	104
C.2.4	setReset	104
C.2.5	getReset	104
C.2.6	setNormalizeNumericClass	104
C.2.7	getNormalizeNumericClass	105
C.2.8	setNormalizeAttributes	105
C.2.9	getNormalizeAttributes	105
C.2.10	setNominalToBinaryFilter	105
C.2.11	getNominalToBinaryFilter	105
C.2.12	setSeed	106
C.2.13	getSeed	106
C.2.14	setValidationThreshold	106
C.2.15	getValidationThreshold	106
C.2.16	setLearningRate	107
C.2.17	getLearningRate	107
C.2.18	setMomentum	107
C.2.19	getMomentum	107
C.2.20	setAutoBuild	108
C.2.21	getAutoBuild	108
C.2.22	setHiddenLayers	108
C.2.23	getHiddenLayers	108
C.2.24	setGUI	109
C.2.25	getGUI	109
C.2.26	setValidationSetSize	109
C.2.27	getValidationSetSize	109
C.2.28	setTrainingTime	109
C.2.29	getTrainingTime	110
C.2.30	blocker	110
C.2.31	getCapabilities	110
C.2.32	buildClassifier	110
C.2.33	distributionForInstance	111
C.2.34	listOptions	111
C.2.35	getOptions	112
C.2.36	toString	112
C.2.37	globalInfo	112

C.2.38	learningRateTipText	112
C.2.39	momentumTipText	113
C.2.40	autoBuildTipText	113
C.2.41	seedTipText	113
C.2.42	validationThresholdTipText	113
C.2.43	trainingTimeTipText	113
C.2.44	nominalToBinaryFilterTipText	113
C.2.45	hiddenLayersTipText	114

Lista de Tabelas

2.1	Funções do córtex cerebral.	9
2.2	Quantidade de estudos encontrados.	22
2.3	Trabalhos selecionados.	23
3.1	Comando de movimentação da CRI.	43
3.2	Preparação dos dados.	46
3.3	Tensão dos fios do barramento controlado pelo <i>joystick</i> e os movimentos.	50
3.4	Conteúdo do arquivo de propriedades: Integração.	53
3.5	Conteúdo do arquivo de propriedades: Núcleo.	55
3.6	Caractere x movimentação.	60
3.7	Valores aplicados aos pinos PWM do arduino e os movimentos.	61
4.1	Conteúdo do arquivo de propriedades do sistema.	65
4.2	Configuração da RNA otimizada (eletrodo Fp1).	68
4.3	Configuração ótima do arquivo de propriedades (eletrodo Fp1).	69
4.4	Matriz de confusão da classificação dados (eletrodo Fp1).	69
4.5	Configuração da RNA otimizada (eletrodo Fp2).	72
4.6	Configuração ótima do arquivo de propriedades (eletrodo Fp2).	73
4.7	Matriz de confusão da classificação dados (eletrodo Fp2).	73
4.8	Representação dos dados de teste para os requisitos a serem testado.	75
A.1	Dados dos participantes do experimento (Parte 1).	95
A.2	Dados dos participantes do experimento (Parte 2).	96
A.3	Dados dos participantes do experimento (Parte 3).	97
A.4	Dados dos participantes do experimento (Parte 4).	98

Lista de Figuras

2.1	Estrutura do córtex cerebral.	8
2.2	Formação e propagação do impulso nervoso.	10
2.3	Eletrodos intracranianos.	11
2.4	Eletroencefalograma não invasivo.	12
2.5	Artefato robótico.	13
2.6	<i>Design</i> experimental.	14
2.7	Sistema Internacional 10-20.	15
2.8	Frequência de onda delta.	16
2.9	Frequência de onda theta.	17
2.10	Frequência de onda alpha.	17
2.11	Frequência de onda beta.	17
2.12	Frequência de onda gamma.	18
2.13	Transmissão de diferentes intenções de sinais.	20
2.14	Componentes de controle.	24
2.15	CRI e seus componentes.	25
2.16	Estrutura de testes para seleção de dados.	26
2.17	Componentes da interface: (a) Arquitetura. (B) Apoio de cabeça. (c) Joystick. (d) Ajuste.	27
2.18	Arquitetura do sistema.	28
2.19	Esquema do sistema DTU-BCI.	29
2.20	Imagem da tela apresentada durante a fase de calibração.	30
2.21	Posicionamento dos sensores EEG e EOG.	31
2.22	Delineamento experimental.	31
2.23	Projeto mecatrônico do sistema.	32
2.24	Projeto da CRI.	33
2.25	Aplicação para registrar as expressões faciais.	34
3.1	Modelo computacional do projeto.	38
3.2	Esquema: aquisição do sinal.	39
3.3	Esquema: extração de características.	40
3.4	Frequência dominante.	41
3.5	Esquema: treinamento e classificação.	42
3.6	Distribuição dos eletrodos.	44
3.7	Cadeira de rodas motorizada.	48
3.8	Circuito do <i>joystick</i> visto de cima.	49
3.9	Esquema de ligação entre o <i>joystick</i> e o circuito do microcontrolador.	50
3.10	Visão geral da arquitetura.	51
3.11	Classes participantes do subsistema Integração.	52
3.12	Classes participantes do subsistema Núcleo.	54
3.13	Classes participantes do subsistema Dispositivo.	56
3.14	Visão de pacotes e subsistemas da aplicação.	58
3.15	Visão de implementação da aplicação.	59
3.16	Fluxograma do circuito de acionamento.	61
3.17	Implementação da caixa com o circuito de controle.	62

4.1	Localização dos eletrodos (Fp1 e Fp2) utilizados na pesquisa.	64
4.2	Curvas de aprendizagem médias para $\eta = 0.01$ (eletrodo Fp1).	66
4.3	Curvas de aprendizagem médias para $\eta = 0.1$ (eletrodo Fp1).	66
4.4	Curvas de aprendizagem médias para $\eta = 0.5$ (eletrodo Fp1).	67
4.5	Curvas de aprendizagem médias para $\eta = 0.9$ (eletrodo Fp1).	67
4.6	Melhores curvas de aprendizagem selecionadas. (eletrodo Fp1).	68
4.7	Curvas de aprendizagem médias para $\eta = 0.01$ (eletrodo Fp2).	70
4.8	Curvas de aprendizagem médias para $\eta = 0.1$ (eletrodo Fp2).	70
4.9	Curvas de aprendizagem médias para $\eta = 0.5$ (eletrodo Fp2).	71
4.10	Curvas de aprendizagem médias para $\eta = 0.9$ (eletrodo Fp2).	71
4.11	Melhores curvas de aprendizagem selecionadas. (eletrodo Fp2).	72
4.12	Esquema para acionar CRI a partir de 4 estados mentais.	74
4.13	Dimensões do módulo de referência.	74
4.14	Área para manobra sem deslocamento.	75
4.15	Abordagem para executar e avaliar o teste do componente.	76
4.16	Resultado do teste da interface de acionamento.	76
4.17	Implementação do método initialize()	77
4.18	Resultado do teste da interface de acionamento após modificação.	77
4.19	Abordagem para executar e avaliar o teste de integração.	78
4.20	Resultado do teste de integração: <i>Layer Integration</i>	79
4.21	Resultado do teste de integração: Integração e Núcleo.	80
4.22	Resultado do teste de integração do sistema.	81

Lista de Siglas

ACO	Auto-condicionamento do Operador
ADK	<i>Android Development Kit</i>
API	<i>Application Programming Interface</i>
AVC	Acidente Vascular Cerebral
BLC	<i>Bayes Linear Classifier</i>
CRI	Cadeira de Rodas Inteligente
CSP	<i>Common Spatial Patterns</i>
DAC	<i>Digital-to-Analog Converter</i>
E	Energia
EDF	European Data Format
EEG	Eletroencefalograma
ELA	Esclerose Lateral Amiotrófica
EOG	Eletrooculograma
FD	Frequência dominante
FFT	<i>Fast Fourier Transform</i>
FM	Mínimo valor da <i>Fast Fourier Transform</i>
Hz	Hertz
IBGE	Instituto Brasileiro de Geografia e Estatística
LAN	<i>Local Area Network</i>
LCS	<i>Lip Control System</i>
MA	Média aritmética
MLP	<i>Multi Layer Perceptron</i>
MS	Ministério da Saúde
MVF	Máximo valor da <i>Fast Fourier Transform</i>
ONU	Assembleia Geral das Nações Unidas
OMS	Organização Mundial da Saúde
PDA	<i>Personal Digital Assistant</i>
PIC	<i>Programmable Interface Controller</i>
PPGMCTI ..	Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial
PWM	<i>Pulse Width Modulation</i>
RNA	Rede Neural Artificial
RP	Reconhecimento de Padro
TF	Transformada de Fourier
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
SDK	<i>Software Development Kit</i>
STFT	<i>Short-time Fourier Transform</i>
SVM	<i>Support Vector Machine</i>

Introdução

O ser humano tem necessidade de interagir com o ambiente ao seu redor. Entretanto, pessoas com mobilidade reduzida acabam modificando sua rotina passando a ocupar-se de atividades pouco ativas, reduzindo seu desempenho físico, suas habilidades motoras e sua capacidade de coordenação. Esses efeitos não favorecem à manutenção de um estilo de vida saudável, levando essas pessoas ao isolamento social e à solidão. O aumento das atividades sociais voltadas para indivíduos com mobilidade reduzida, cada vez mais, revela a insuficiência funcional dos equipamentos atuais direcionados para este grupo de indivíduos. Segundo [Petry \(2013\)](#), equipamentos tradicionais para auxílio de mobilidade como cadeiras de rodas, muletas, bengalas e membros artificiais têm capacidade de auxílio limitado e em muitos casos não são capazes de prover o auxílio necessário para indivíduos que possuam combinações de deficiências físicas, cognitivas e de percepção.

Quando uma pessoa deseja realizar um determinado movimento, é através das vias neuromusculares que o cérebro envia sinais aos músculos do corpo para realizar o movimento; contudo, em alguns casos, essa comunicação é interrompida devido a danos provocados por doenças que danificam as vias através das quais o cérebro controla o corpo. Dentre elas, estão a Esclerose Lateral Amiotrófica (ELA), Acidente Vascular Cerebral (AVC), lesão do cérebro ou da medula espinhal, paralisia cerebral, distrofia muscular e esclerose múltipla ([WOLPAW et al., 2002](#)).

A imaginação do movimento é uma estratégia utilizada em interfaces cérebro-computador que permite à pessoa com graves comprometimentos motores ter algum controle sobre uma cadeira de rodas inteligente. Segundo [Decety e Boisson \(1990\)](#), a imaginação do movimento é um processo que depende da integridade das estruturas cerebrais, relacionadas com o planejamento e a execução do movimento. Segundo [Craje et al. \(2010\)](#), indivíduos com paralisia cerebral tem dificuldade de planejamento, porém, a capacidade de planejar ações é passível de mudanças e a imaginação do movimento é uma técnica promissora na melhora deste componente. Segundo [Danckert et al. \(2002\)](#), indivíduos com lesão no córtex parietal direito apresentam prejuízo na imagética motora. Contudo, a reabilitação do cérebro lesado pode promover reconexão de circuitos neuronais lesados. Segundo [Pascual-Leone et al. \(2005\)](#), o sistema nervoso central, na presença de lesões, utiliza-se da plasticidade neural¹ na tentativa de recuperar as funções perdidas e fortalecer as funções similares relacionadas às originais. Segundo [Donati et al. \(2016\)](#), pacientes (oito paraplégicos crônicos) submetidos a um período de longo treinamento (12

¹A plasticidade neural refere-se à capacidade que o sistema nervoso central possui em modificar algumas das suas propriedades morfológicas e funcionais em resposta às alterações do ambiente ([PASCUAL-LEONE et al., 2005](#)).

meses), fazendo uso de interfaces cérebro-máquina desenvolveram plasticidade cortical e espinhal, e a combinação dessa reorganização produzido pelo treinamento podem ter induzido também a geração de novas conexões cerebrais.

Nesse contexto, surge a necessidade de desenvolver novas tecnologias que venham atender as demandas das pessoas com comprometimentos motores e neurológicos.

1.1 Definição do problema

De acordo com a Organização Mundial da Saúde (OMS), existem, no mundo, um bilhão de pessoas com deficiência, representando cerca de 15% da população mundial, ou uma em cada sete pessoas. Deste número, cerca de 190 milhões de adultos sofrem dificuldades significativas de mobilidade. Estima-se que cerca de 93 milhões de crianças menores de 15 anos de idade vivem com alguma deficiência moderada ou grave. Ainda, segundo a OMS, o número de pessoas que sofrem de deficiência continuará a aumentar à medida que as populações envelhecem e com o aumento global das doenças crônicas. A Assembleia Geral das Nações Unidas (ONU) destacou que cerca de 80% das pessoas com deficiência vivem em países em desenvolvimento e reiterou a necessidade de assegurar que as pessoas com deficiência estejam incluídas em todos os aspectos do desenvolvimento ([WHO, 2015](#)).

No Brasil, a população atingida por algum tipo de deficiência é significativa. De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE), 6,2% da população brasileira têm algum tipo de deficiência. O estudo mostra que 1,3% da população tem algum tipo de deficiência física e quase a metade deste total, 46,8%, tem grau intenso ou muito intenso de limitações e que somente 18,4% desse grupo frequentam serviço de reabilitação ([IBGE, 2014](#)). A região Nordeste teve a maior taxa de prevalência de pessoas com pelo menos uma das deficiências (mental, auditiva, visual e motora): 26,3%.

De acordo com o IBGE, a proporção de pessoas, de 18 anos ou mais de idade, que usavam algum recurso para auxiliar na locomoção, tais como muleta, bengala ou cadeira de rodas foi de 2,5%. Por Região, este percentual variou de 1,7%, na Norte, a 2,8%, na Nordeste; mesmo com o uso do recurso para auxiliar a locomoção, 2,7% das pessoas não conseguiam ou tinham grande dificuldade para se locomover ([IBGE, 2014](#)).

Somado a isso, existem ainda as vítimas de AVC: 15 milhões de pessoas apresentam AVC por ano; destas, cinco milhões morrem em decorrência do evento e grande parte dos sobreviventes apresenta sequelas físicas e/ou mentais, sendo que muitas delas permanecerão em uma cadeira de rodas até o final de suas vidas ([RANGEL; BELASCO; DICCINI, 2013](#)). Existem ainda pessoas com paralisia cerebral, esclerose lateral amiotrófica, esclerose múltipla, distrofia muscular e outras doenças que atacam o sistema neuromotor.

No Brasil, fontes do Ministério da Saúde (MS) apontam o AVC como a principal causa de incapacidade das pessoas, com taxa elevada para ambos os sexos e, com o envelhecimento da população, a incidência tende a aumentar (SOUZA *et al.*, 2006; MS, 2009). A doença é uma das maiores causas de sequelas permanentes que geram incapacidades funcionais e representa a terceira causa de morte no mundo (LOTUFO; BENSENOR, 2004). Segundo Sá, Grave e Périco (2014), uma das sequelas motoras mais comuns em vítimas de AVC é a hemiplegia, que é a perda do controle motor do hemicorpo contralateral à lesão encefálica, situação, muitas vezes, incapacitante que se torna um desafio à reabilitação. A hemiplegia pode causar nas pessoas dependência de dispositivos auxiliares para locomoção e em outras atividades da vida diária. Segundo Dalpian, Grave e Périco (2013), as consequências são graves: a sensibilidade, a visão, a audição, a expressão facial e a fala podem estar ausentes, ou acentuadamente diminuídas, pois o paciente não percebe os estímulos que chegam de seu lado afetado.

O AVC apresenta um grande impacto na sociedade e na saúde pública, devido à sua prevalência² e morbidade³, sendo atualmente uma das principais causas de incapacidade motora em pessoas no Brasil, bem como de incapacidades em adultos em idade produtiva, além de condicionar a pessoa ao isolamento social. Segundo Oliveira, Cacho e Borges (2006), as sequelas deixadas por um acidente vascular cerebral são variáveis e podem ser sensitivas, motoras e/ou cognitivas, gerando incapacidades e prejudicando a independência e a qualidade de vida das pessoas. As incapacidades causadas pelas sequelas do AVC ocorrem por consequência da morte neuronal e pela falta ou alteração da comunicação do sistema nervoso central com os órgãos efetores⁴ (RAFII; HILLIS, 2006). Algumas das limitações imediatas produzidas são a incapacidade de realizar marcha (EICH *et al.*, 2004) e déficit no controle postural (ZIJLSTRA *et al.*, 2010).

A fraqueza muscular tem sido reconhecida como fator limitante de pacientes pós-acidente vascular cerebral e é refletida pela incapacidade de gerar força muscular em níveis normais (OLIVEIRA; CACHO; BORGES, 2006). As limitações funcionais após um acidente vascular cerebral incluem redução da velocidade da marcha e da capacidade de andar, disfunção do equilíbrio corporal, modificação da confiança no próprio equilíbrio e preocupação com a possibilidade de cair (HAFER *et al.*, 2008). Declínio funcional em aspectos biomecânicos, de entradas sensoriais, integração e reponderação de informações, estratégias motoras, processamento cognitivo e percepção da verticalidade geram maior ou menor impacto no equilíbrio dos diferentes pacientes (JONSDOTTIR; CATTANEO, 2007). Isso torna a disfunção do equilíbrio corporal muito heterogênea na sua apresentação, entretanto, com uma consequência comum entre os indivíduos: a instabilidade postural (HASHIMOTO *et al.*, 2014).

²A Prevalência quantifica o número de casos existentes de uma doença em uma população. Fonte: PID (2015).

³Variável característica das comunidades de seres vivos, refere-se ao conjunto dos indivíduos que adquiriram doenças num dado intervalo de tempo. Fonte: FURB (2015).

⁴Constituídos por músculos e glândulas, são órgãos que recebem estímulos do sistema nervoso central e atuam sobre um sistema muscular ou glandular já existente na anatomia humana (FAUCONNIER; TURNER, 2002).

[Cordini, Oda e Furlanetto \(2005\)](#) apontam, que, quando as incapacidades geradas pelo AVC impedem que a pessoa se engaje em atividades que constituem um componente importante para sua identidade pessoal, existe redução do seu bem estar. [Ricci, Kubota e Cordeiro \(2005\)](#) definem dependência como a incapacidade da pessoa funcionar satisfatoriamente, sem a ajuda de um semelhante, sendo classificado em categorias: dependência por limitações físicas, funcionais e por limitações cognitivas. No caso do AVC, as três podem ser simultâneas ([PEDREIRA; LOPES, 2010](#)).

Segundo [Lima et al. \(2015\)](#), dados da OMS indicam que mais da metade das vítimas de AVC apresentam algum grau de incapacidade física, especialmente na extremidade superior. As sequelas, temporárias ou permanentes, limitam o movimento e a função dos vitimados. A presença de sequelas, após o AVC, gera dependência por parte dos pacientes para a realização das atividades de vida diária ([RANGEL; BELASCO; DICCINI, 2013](#)). As sequelas geram impacto econômico, social e familiar ([CRUZ; DIOGO, 2009](#)).

Os sintomas neurológicos, após um AVC, variam de acordo com a localização e extensão da lesão cerebral, mas a fraqueza motora é o sintoma mais frequentemente identificado. Após um AVC, até 85% dos pacientes apresentam hemiparesia com predomínio do déficit no membro superior, no qual 80% destes pacientes conseguem alcançar a melhora da função da extremidade dentro de 3 semanas da ocorrência do AVC e 95% alcançam dentro de 9 semanas. No entanto, entre 55% e 75% dos sobreviventes têm limitações funcionais persistentes da extremidade superior, resultando em dependência para as atividades da vida diária ([CHOI et al., 2014](#)).

Embora as dificuldades de mobilidade das pessoas com comprometimentos motores e neurológicos possam ser diminuídas com o uso de cadeiras de rodas inteligentes, em alguns casos, a sua utilização de forma independente ainda é um desafio. Nos últimos anos, houve um grande avanço no desenvolvimento tecnológico das interfaces cérebro-computador, sendo estas cada vez mais sofisticadas. Uma das áreas em que essas tecnologias têm tido grande aplicabilidade é a ampliação da mobilidade para pessoas com graves deficiências motoras. Segundo [Faria et al. \(2013\)](#), especificamente para pessoas que têm deficiência grave no seu sistema motor, aplicações baseadas em eletroencefalograma são mais indicadas.

Nesse sentido, o desenvolvimento de pesquisas que contribuam para ampliação da mobilidade dessas pessoas torna-se essencial para a sua ressocialização. O impacto social do uso de um sistema para auxílio à mobilidade de pessoas com comprometimentos motores e neurológicos severos seria significativo, devido ao alto número de portadores de deficiência motora existente. Esse fato justifica o interesse cada vez maior no desenvolvimento de tecnologias para auxílio à mobilidade.

Vários projetos relacionados com interfaces cérebro-computador foram desenvolvidos nos últimos anos (BAREA et al., 2002; RANI; SARKAR, 2005; FARIA et al., 2013). As interfaces cérebro-computador, apresentam-se como solução para ajudar indivíduos com graves comprometimentos motores e neurológicos, que não são capazes de conduzir uma cadeira de rodas inteligente (CRI), em suas atividades de vida diária. Segundo Wolpaw et al. (2002), uma interface cérebro-computador é um sistema que propicia um canal de comunicação ou controle para um indivíduo através de sinais puramente cerebrais, sem que haja necessidade de atividade neuromuscular para realizar a ação codificada no sinal cerebral. Segundo Petry (2013), cadeira de rodas inteligente pode ser definida como um dispositivo robótico desenvolvido a partir de uma cadeira de rodas motorizada comercial, sistemas sensoriais, dispositivos de processamento e atuação.

Apesar de existirem vários projetos de pesquisa relacionados com as interfaces cérebro-computador desenvolvidos para fornecer um meio de comunicação e capacidade de controle para pessoas com deficiências motoras graves, verifica-se a necessidade de desenvolver interfaces cérebro-computador com capacidades para validar teorias, aplicar novas tecnologias da informação e da robótica no apoio à ressocialização de pessoas com graves deficiências motoras. Embora muitos dos trabalhos de investigação tentem resolver problemas específicos dentro desta área, nenhum dos estudos, destacou a preocupação em desenvolver uma interface cérebro-computador que facilite o estudo e desenvolvimento de novas tecnologias nesta área de pesquisa. É neste ponto que este trabalho de investigação se destaca em relação a esses trabalhos.

1.2 Objetivo

O objetivo geral desse trabalho de investigação é desenvolver um modelo computacional baseado em interface cérebro-computador que permita o estudo e desenvolvimento de novas tecnologias a partir dos sinais de eletroencefalograma para ajudar indivíduos com mobilidade reduzida a conduzir uma cadeira de rodas inteligente, em suas atividades de vida diária.

Os objetivos específicos são:

- Produzir uma interface cérebro-computador baseada na imaginação de movimentos do punho esquerdo, punho direito, ambos os punhos e ambos os pés para controlar uma CRI.
- Delinear uma arquitetura modular adequada à definição de metodologias para o desenvolvimento de interfaces cérebro-computador;

- Criar uma interface que permita comandar a CRI de forma mais flexível. O *joystick* tradicional, não favorece o controle da CRI, principalmente para pessoas com o graves dificuldades motora.
- Avaliar a viabilidade técnica de controle da CRI com monitoramento da atividade cerebral através da medição dos sinais EEG do córtex pré-frontal;
- Testar o sistema proposto e seus módulos utilizando, os sinais EEG adquiridos do banco de dados: *eegmmidb - EEG Motor Movement/Imagery Dataset*.

1.3 Importância da pesquisa

Atualmente, observa-se o aumento do número de pessoas da população mundial diagnosticadas com algum tipo de deficiência física que possuem mobilidade reduzida. Em virtude disso, diversas técnicas vêm sendo propostas e utilizadas na construção de interfaces cérebro-computador. Contudo, o desenvolvimento de sistemas baseados em interfaces cérebro-computador sem apoio automatizado de ferramentas constitui uma atividade onerosa, propensa a erros que pode dispendir muito tempo na construção desse tipo de sistema. Além disso, dificuldades quanto à integração, evolução, manutenção e reutilização desses sistemas são muito comuns. Dessa forma, a automatização de procedimentos para a construção de interfaces cérebro-computador é importante, contribuindo para melhorar a qualidade e produtividade de atividades relacionadas ao desenvolvimento desse tipo de sistemas.

Considerando a necessidade em se construir interfaces cérebro-computador de qualidade para validação e testes de teorias, esforços para o estabelecimento de modelos computacionais e arquiteturais são ferramentas importantes. Nesse contexto, esse trabalho de investigação apresenta um modelo computacional que visa contribuir para a área de interfaces cérebro-computador com a disponibilização de uma arquitetura que facilite o desenvolvimento, reúso, manutenção, testes e extensão de sistemas baseados em interfaces cérebro-computador.

1.4 Limites e limitações

O estudo desenvolvido nesse trabalho de investigação apresentou limitações quanto à aquisição de sinais EEG envolvendo seres humanos. Dessa forma, utilizamos o banco de dados *eegmmidb - EEG Motor Movement/Imagery Dataset* para aquisição e tratamento dos sinais EEG. Outra limitação, proveniente da utilização dessa base de dados é em relação a quantidade de estados mentais que a mesma disponibiliza, a saber: movimento

dos punhos esquerdo e direito, ambos os punhos e ambos os pés. Dessa forma, mesmo sendo possível acionar a CRI em cinco direções (frente, parar, ré, esquerda e direita), utilizaremos nos testes para validação do modelo computacional apenas quatro direções: frente, parar, esquerda e direita.

1.5 Organização da Tese de doutorado

A tese apresenta 5 capítulos e está estruturada da seguinte forma:

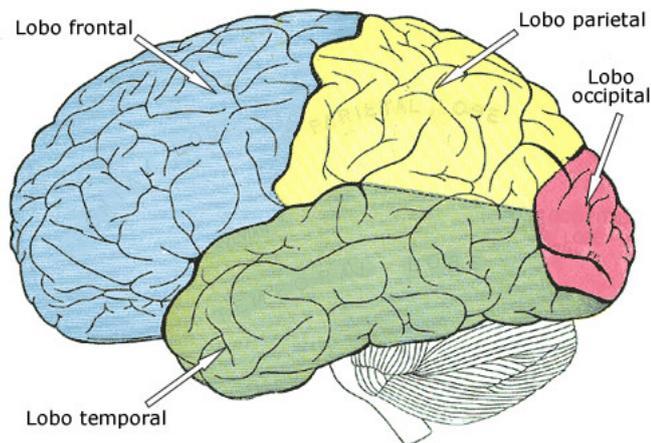
- **Capítulo 1 - Introdução:** Contextualiza o âmbito, no qual a pesquisa proposta está inserida. Apresenta, portanto, a definição do problema, objetivos e justificativas da pesquisa e como esta tese de doutorado está estruturada;
- **Capítulo 2 - Interface cérebro-computador:** São apresentados os conceitos fundamentais sobre sinais cerebrais e discutidos os fundamentos das interfaces cérebro-computador. Apresentam-se os trabalhos de investigação que tratam de interfaces cérebro-computador para auxílio de pessoas com graves comprometimentos motores;
- **Capítulo 3 - Projeto e desenvolvimento da pesquisa:** Neste capítulo, é apresentado o projeto e desenvolvimento da interface cérebro-computador para controlar cadeira de rodas inteligente;
- **Capítulo 4 - Avaliação e resultados experimentais:** Neste capítulo, são apresentados os resultados experimentais da interface cérebro-computador proposta;
- **Capítulo 5 - Considerações finais:** Apresenta as conclusões, contribuições e algumas sugestões de atividades de pesquisa a serem desenvolvidas no futuro.

Interface cérebro-computador

Para o desenvolvimento de um sistema de interface cérebro-computador, faz-se necessário o estudo do cérebro, ou encéfalo. O encéfalo de uma pessoa adulta pesa cerca de 1400 gramas e representa todo o conteúdo do crânio, incluindo cérebro, cerebelo e tronco encefálico (MARK; BARRY; MICHAEL, 2008). A parte mais significativa do cérebro, no que diz respeito às interfaces cérebro-computador, é o córtex cerebral, uma camada de 1,5 a 4 mm de substância cinzenta que cobre a superfície exterior do cérebro (WEBSTER, 2009). Ele também controla os movimentos voluntários do corpo, sendo responsável pelos processos cognitivos (memória, raciocínio e inteligência) (RIZZO, 2010).

A Figura 2.1 mostra a estrutura do córtex cerebral com suas divisões. O córtex cerebral está dividido em dois hemisférios cerebrais e cada hemisfério está dividido em 4 lobos distintos: frontal, parietal, occipital e temporal.

Figura 2.1: Estrutura do córtex cerebral.



Fonte: Nicida (2015).

A Tabela 2.1 apresenta as funções dos lobos cerebrais: no lobo frontal, localizado na parte da frente do cérebro (testa), acontece o planejamento de ações e movimento.

Tabela 2.1: Funções do córtex cerebral.

Lobo	Função
frontal	planejamento de ações e movimento
occipital	processam os estímulos visuais
temporal	processar os estímulos auditivos
parietal	percepção de sensações como o tato, a dor e o calor

Fonte: [Nicida \(2015\)](#).

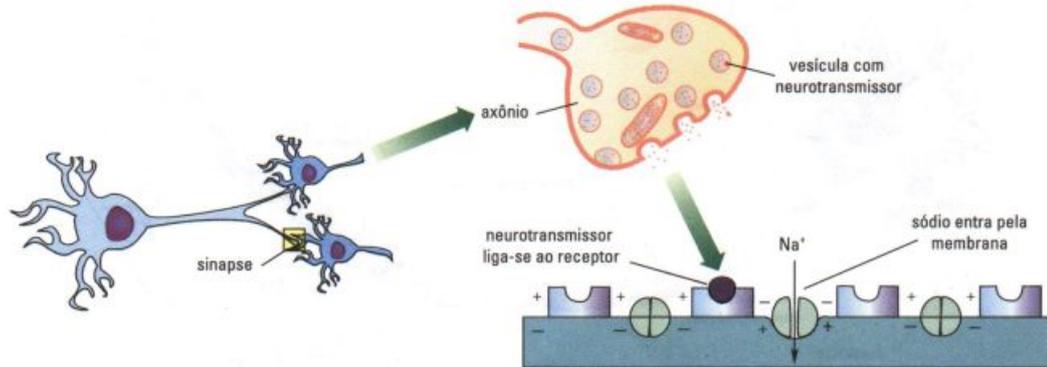
No lobo frontal estão incluídos o córtex motor e o córtex pré-frontal. O córtex motor controla e coordena a motricidade voluntária, sendo que o córtex motor do hemisfério direito controla o lado esquerdo do corpo do indivíduo, enquanto o do hemisfério esquerdo controla o lado direito ([NICIDA, 2015](#)). Um trauma nesta área pode causar fraqueza muscular ou paralisia. Segundo [Vilela \(2008\)](#), a aprendizagem motora e os movimentos de precisão são executados pelo córtex pré-motor, que fica mais ativo do que o restante do cérebro quando se imagina um movimento sem executá-lo. Lesões nesta área não chegam a comprometer a ponto do indivíduo sofrer uma paralisia ou ter problemas para planejar ou agir; no entanto a velocidade de movimentos automáticos, como a fala e os gestos, é perturbada.

Segundo [Degroot \(1994\)](#), o predomínio de qualquer pulso nervoso específico pode modificar as funções controladas pelo cérebro. Embora existam funções atribuídas aos pulsos nervosos em áreas específicas do cérebro, todo o sistema nervoso é altamente conectado. O neurônio é a célula fundamental do sistema nervoso e tem como função receber, processar e enviar as informações ([RIZZO, 2010](#)). Os neurônios são unidades funcionais básicas do sistema nervoso, responsáveis pelos impulsos nervosos. A transmissão desses impulsos se dá por sinapses¹, através dos dendritos e dos axônios.

Os dendritos são ramificações que se estendem a partir do corpo e recebe informações de outros neurônios, enquanto os axônios efetuam um processo mais longo de conexão para passar as informações às outras células (Figura 2.2).

¹Sinapse é um tipo de junção especializada em que um terminal axonal faz contato com outro neurônio ou tipo celular.

Figura 2.2: Formação e propagação do impulso nervoso.



Fonte: Vilela (2008).

Segundo Rizzo (2010), a comunicação intercelular é feita por sinapses através de lacunas entre as células; por neurotransmissores, que são agentes químicos liberados pelo neurônio. Todo esse processo produz fluxos elétricos que permeiam todo o sistema nervoso. A atividade elétrica do neurônio, distribuída pelos axônios, pode se espalhar diretamente a outros neurônios vizinhos que tenham contato elétrico (RIZZO, 2010; VILELA, 2008). A propagação de impulsos cerebrais acontece pelas alterações de potencial interno em relação ao externo da célula, que inicialmente se encontra com as distribuições iônicas equilibradas (VILELA, 2008).

O cérebro apresenta-se como um órgão com maior potencial de interação do ser humano com o meio que o cerca. O mapeamento das zonas cerebrais é essencial para a correspondência da atividade elétrica do cérebro. Essa atividade é identificada como frequências de ondas cerebrais que serão capturadas e interpretadas pelos sensores de eletroencefalografia (EEG).

2.1 Eletroencefalografia

A eletroencefalografia é um método para medição de potenciais elétricos no escalpo (couro cabeludo) do indivíduo, resultantes de atividade elétrica neuronal no encéfalo. Os primeiros estudos relacionados aos potenciais elétricos do cérebro foram realizados no ano de 1875, pelo fisiologista inglês Richard Caton (JACKS; MILLER, 2003). Ele fez registros destes potenciais, capturados no encéfalo de cães e coelhos, utilizando um dispositivo com grande sensibilidade à tensão elétrica. Em 1929, o psiquiatra austríaco Hans Berger realizou os primeiros testes de EEG envolvendo seres humanos e relatou a distinção clara observada entre o EEG do sono e o da vigília (MARK; BARRY; MICHAEL, 2008).

Segundo [Wolpaw e Wolpaw \(2012\)](#), Hans Berger foi o primeiro estudioso a usar EEG qualitativamente. Seu uso regular é no diagnóstico de problemas neurológicos, porém, uma combinação com métodos de imaginação motora permite investigar e identificar estados mentais.

Segundo [Silva e Niedermeyer \(1982\)](#), a aquisição dos sinais cerebrais utiliza técnicas e métodos para o procedimento, que se dividem em duas principais categorias: invasiva (Figura 2.3) e não invasiva (Figura 2.4). A técnica invasiva é utilizada quando o foco está na captação de dados com o mínimo de perda de sinais. Nesse método há uma diminuição de ruídos. Todavia, existe a necessidade de procedimento cirúrgico para que os eletrodos sejam colocados diretamente no cérebro.

Figura 2.3: Eletrodos intracranianos.



Fonte: [Wen \(2006\)](#).

A vantagem na utilização da técnica não invasiva é que não há a necessidade da introdução de eletrodos no interior do crânio ou mesmo de qualquer tipo de procedimento cirúrgico. Esta técnica é menos precisa no que se refere à qualidade de sinal e mesmo quanto ao maior nível de ruído produzido. Por outro lado, ela é mais simples ([WOLPAW, 2007](#)). Os sinais são obtidos com uma largura de banda mais limitada, suscetíveis a sobreposições e interferências de sinais diversos. Entretanto, eles possuem aplicabilidade suficiente para ativar comandos computacionais, como, por exemplo, mover uma cadeira de rodas inteligente.

Um gorro comercial com os eletrodos já instalados e distribuídos pode ser utilizado, para facilitar o posicionamento dos mesmos. O sinal de eletroencefalograma é filtrado e amplificado imediatamente após a sua aquisição. A faixa de frequência para análises clínicas é de 0,1 a 50 Hz. Essa baixa amplitude é promovida, principalmente, pelas camadas

existentes entre o cérebro e o escalpo ([WEBSTER, 2009](#)).

Figura 2.4: Eletroencefalograma não invasivo.



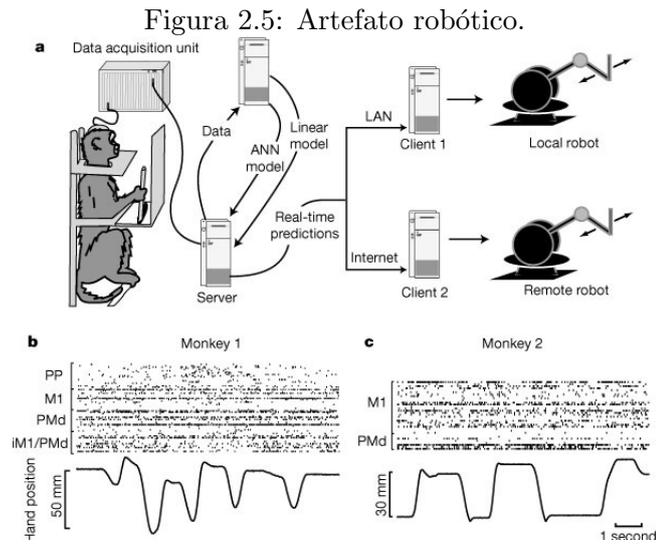
Fonte: [Wladimir \(2013\)](#).

Segundo [Wolpaw \(2007\)](#), os estudos com eletroencefalografia invasiva tiveram início na década de 60 pelo cientista alemão Eberhard Fetz, através da análise dos neurônios localizados no córtex motor primário, associados ao processo de movimento corporal. Esses estudos tinham a intenção de levar os movimentos comandados pelo cérebro a dispositivos eletrônicos.

A análise desses sinais permitiu determinar o intervalo entre o início das atividades das células do córtex e dos músculos envolvidos no processo, sendo possível analisar o impacto das regiões cerebrais nas áreas do corpo relativas às posições onde os membros do corpo se encontravam e à sua força motora.

[Johan et al. \(2000\)](#) desenvolveram estudos utilizando macacos para controlar um braço robótico, mostrando ser possível também utilizar a mesma técnica em seres humanos. Essa pesquisa abriu caminho para a criação de próteses humanas controladas pelo pensamento e para o desenvolvimento de novos tratamentos para pessoas paralisadas.

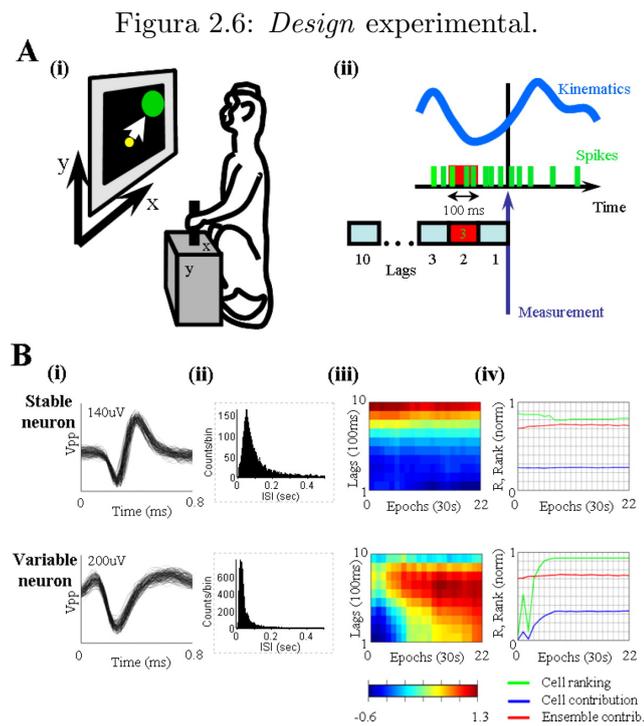
A Figura 2.5 ilustra o primeiro artefato robótico usado, ou seja, um braço mecânico, que realizava os movimentos pensados por uma macaca. O diagrama esquemático (a) representa o aparelho experimental utilizado para realizar a aquisição dos dados, para controlar os movimentos dos dispositivos robotizados locais (LAN) e remotos (internet). (b) e (c) registram simultaneamente a atividade neural.



Fonte: [Johan et al. \(2000\)](#).

[Schmidt \(1980\)](#) desenvolveu uma pesquisa utilizando implantes cranianos para permitir o condicionamento de animais para o controle de dispositivos extracorpóreos utilizando padrões neurais. Essas pesquisas demonstraram a importância em tornar a tecnologia de eletroencefalografia aplicável a seres humanos com problemas de paralisia, indicando a possibilidade de desenvolvimento de interfaces funcionais.

[Carmena et al. \(2005\)](#) desenvolveram pesquisa utilizando macacos, na qual os referidos animais foram condicionados a responder por sinais neuronais à iluminação de um conjunto de oito sinais luminosos, em que cada um estaria associado a um estímulo cerebral específico, com correspondências sucessivas entre os padrões cerebrais e as luzes. A Figura 2.6 ilustra a forma como os ensaios foram desenvolvidos. Cada ensaio iniciou-se com um alvo apresentado em uma posição aleatória na tela. O macaco tinha que usar o polo para colocar o cursor sobre o alvo; se o macaco atravessasse o alvo muito rápido, o alvo desaparecia e o resultado não era registrado. Os macacos tinham 5 segundos para acertar o alvo, com um atraso de 0,5 segundos entre os ensaios. Eles recebiam recompensas para o desempenho correto.

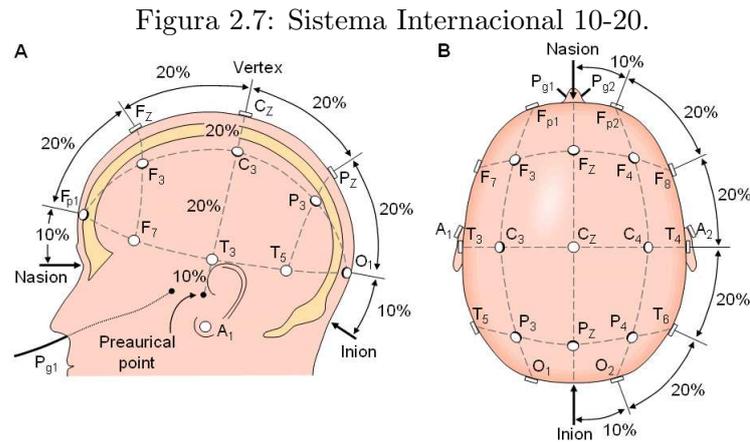


Fonte: [Carmenta et al. \(2005\)](#).

Apesar das leituras de sinais utilizando essa técnica fornecerem informações mais detalhadas sobre o funcionamento dos neurônios, a configuração do córtex cerebral pode variar de pessoa para pessoa ([WEN, 2006](#)).

Segundo [Webster \(2009\)](#), o sistema internacional 10-20² (Figura 2.7) foi proposto para assegurar a padronização do local de fixação dos eletros para captura dos sinais EEG, tais posições são localizadas de acordo as áreas cerebrais. Dessa forma, independentemente do desenvolvimento craniano da pessoa, a posição dos eletrodos mantém-se suficientemente consistente entre os diversos tipos de biotipos das pessoas ([WEN, 2006](#)).

²O sistema internacional 10-20 é utilizado no mapeamento das posições onde serão fixados os eletrodos para registrar os sinais do Eletroencefalograma.



Fonte: Webster (2009).

Nesse sistema, os números pares indicam eletrodos do lado direito da cabeça e os ímpares do esquerdo. As letras F, C, P, T e O são utilizadas para identificar as diferentes regiões do córtex, sendo frontal, central, parietal, temporal e occipital, respectivamente (WEBSTER, 2009).

Segundo Wolpaw (2007), os primeiros estudos com eletroencefalografia não invasiva foram realizados durante os anos 60 e permitiram que pessoas controlassem os seus níveis de concentração e atenção. As ondas cerebrais são produzidas principalmente quando estamos acordados de olhos fechados e estão associadas à inativação de áreas do córtex que não estão sendo utilizadas (WOLPAW, 2007). Assim, pessoas foram capazes de controlar os padrões das ondas alfa através de técnicas de relaxamento.

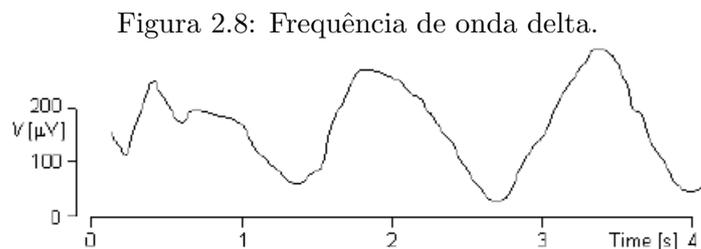
As interfaces cérebro-computador possuem características segundo vários aspectos. No controle temporal síncrono, um estímulo externo, geralmente visual ou auditivo, indica o momento em que a pessoa deve gerar o padrão cerebral que será analisado, causando confinamento deste padrão em uma janela temporal e na maioria das vezes, melhores taxas de acerto (PFURTSCHELLER; NEUPER, 2001). Esse método gera uma dependência do indivíduo em relação ao dispositivo externo. Já no modo de operação assíncrono, a pessoa envia comandos cerebrais para a interface cérebro-computador sem que haja associação temporal a estímulos externos (MILLAN; MOURINO, 2003). Quando o processo de aquisição dos dados, o pré-processamento do sinal, a extração de características e a classificação são realizados durante o tempo em que o indivíduo está utilizando o sensor, a interface é classificada como *online* (VIDAURRE et al., 2006). Segundo Vidaurre et al. (2006), quando a aquisição dos dados é feita para análise posterior e os sinais capturados não são utilizados para gerar ações imediatas, a interface cérebro-computador é dita *offline*. Já na aprendizagem homem-máquina, as abordagens utilizadas são a de auto-

condicionamento do operador (ACO) e reconhecimento de padrões (RP). Na abordagem ACO, a pessoa recebe uma realimentação dos sinais gerados e aprende, ou se condiciona, a controlá-los de forma a produzir um sinal que será mais facilmente reconhecido pela interface cérebro-computador (BIRBAUMER et al., 2000). Já na abordagem por RP, a maior carga de aprendizado, ou treinamento, está na interface cérebro-computador, que deve associar, de maneira correta, os diferentes estados mentais da pessoa nas ações pré-estabelecidas (MILLAN et al., 2002).

2.2 Frequências dos sinais neurais

As frequências captadas via eletroencefalograma representam atividades elétricas geradas no córtex cerebral e se correlacionam com estados do comportamento humano, como os níveis de atenção, sono, vigília, concentração ou processos cognitivos (PINTO, 2006).

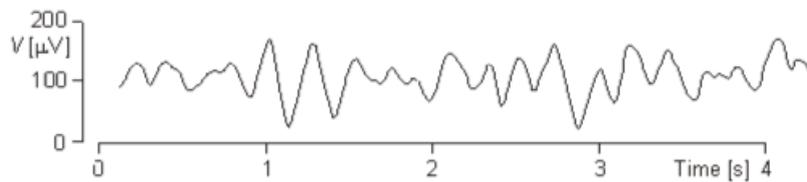
Segundo Mark, Barry e Michael (2007), os sinais neurais são categorizados por faixa de frequência, sendo que cada uma delas é representada por uma letra do alfabeto grego. As frequências delta, theta, alpha, beta e gamma têm origem cortical e bandas medidas em hertz (Hz) (MARK; BARRY; MICHAEL, 2007). A análise dessas frequências é complexa devido a grande quantidade de informações percebidas pelos eletrodos. As Figuras 2.8 a 2.12 mostram as frequências das ondas: delta na Figura 2.8, theta na Figura 2.9, alpha na Figura 2.10, beta na Figura 2.11 e gamma na Figura 2.12 respectivamente.



Fonte: Mark, Barry e Michael (2007).

Segundo Mark, Barry e Michael (2007), o ritmo delta (frequências de 0,1 Hz a 4 Hz) normalmente não ocorre em adultos durante um estado de vigília. São observadas durante o sono em crianças com menos de 1 ano de idade.

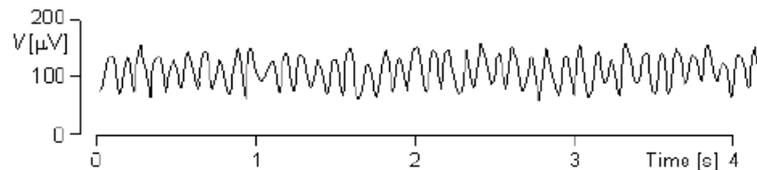
Figura 2.9: Frequência de onda theta.



Fonte: [Mark, Barry e Michael \(2007\)](#).

Segundo [Mark, Barry e Michael \(2007\)](#), o ritmo theta (frequências de 4 Hz a 8 Hz) está relacionado à sonolência ou sono leve. É observado assim que a pessoa acorda ou imediatamente antes de dormir. Em geral, ocorre nas regiões parietal e temporal em crianças. Em adultos, esse ritmo se manifesta em situações de estresse mental como desapontamento ou frustração.

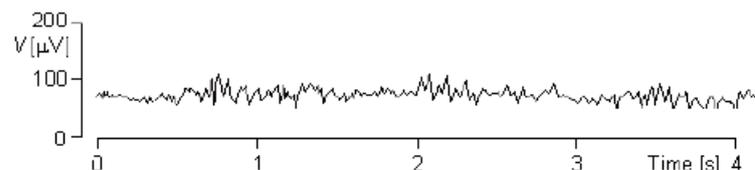
Figura 2.10: Frequência de onda alpha.



Fonte: [Mark, Barry e Michael \(2007\)](#).

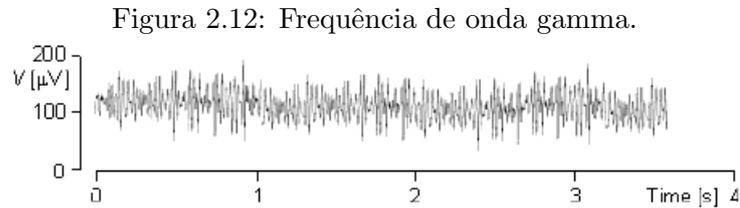
Segundo [Mark, Barry e Michael \(2007\)](#), o ritmo alpha (frequências de 8 Hz a 13 Hz), que é verificado em uma pessoa normal relaxada, ocorre na região posterior e com maior amplitude sobre a região occipital. O ritmo alpha é mais bem visualizado quando o indivíduo está com os olhos fechados, em situação de baixa atividade mental e relaxamento físico.

Figura 2.11: Frequência de onda beta.



Fonte: [Mark, Barry e Michael \(2007\)](#).

Segundo [Mark, Barry e Michael \(2007\)](#), o ritmo beta (frequências de 13 Hz a 30 Hz) predominante na região frontal e central, está associado à atividade e concentração.



Fonte: [Mark, Barry e Michael \(2007\)](#).

Segundo [Mark, Barry e Michael \(2007\)](#), o ritmo gama (frequências de 30 Hz a 70 Hz) está relacionado à elevada atividade mental, à resolução de problemas e também ao medo.

As frequências captadas são amplificadas e filtradas por um sensor de eletroencefalograma para remoção de ruído que possa atrapalhar a leitura dos sinais, que são convertidos de analógicos para digitais utilizando frequência de amostragem. Nesse processamento digital, são aplicados algoritmos matemáticos para transformar os dados obtidos, que se apresentam com características diversificadas, transformando-os em informações básicas. Assim, o conjunto de características dos sinais passa por um processo de classificação para o reconhecimento de padrões.

2.3 Potenciais evocados

Esse potencial é uma resposta inerente do cérebro, o que tem como consequência a estabilidade ao longo do tempo e a possibilidade de qualquer pessoa controlar tais sistemas ([BAYLISS, 2001](#)). Em uma interface cérebro-computador os potenciais evocados podem gerar sinais de controle quando a pessoa é submetida a um estímulo. Este método tem como vantagem um pequeno período de treinamento e adaptação aos sistemas baseados nesses potenciais, ao custo dessa pessoa ter que esperar pelo estímulo externo.

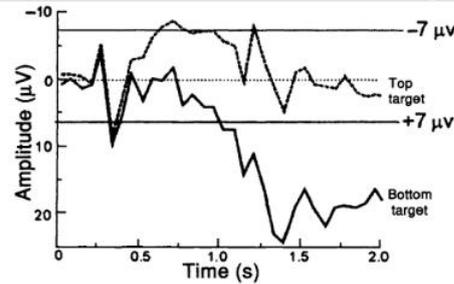
As atividades cerebrais mais utilizadas pelas interfaces cérebro-computador atuais baseadas em eletroencefalograma são:

- **P300**: este potencial surge em decorrência de um evento externo, geralmente sob a forma visual ou auditiva ([DONCHIN; SPENCER; WIJESINGHE, 2000](#)). Este sistema de comunicação permite à pessoa escolher letras em tabuleiros pictográficos, movimentar cursor em telas de computador ou controlar uma cadeira de rodas por meio

de um *Personal Digital Assistant* (PDA). O P300 recebe esse nome por ser um potencial que surge 300ms após o estímulo desencadeador, ao contrário do potencial visual evocado. O P300 não surge necessariamente com todo e qualquer estímulo, apenas com aqueles que têm algum grau de novidade ou significância para a pessoa. O P300 é mais evidente no córtex parietal e tem seu uso em implementações de interfaces cérebro-computador, onde a expectativa da pessoa é um fator importante. Uma vantagem desse método sobre os outros é que o P300 não precisa ser aprendido, já que é um tipo de resposta cerebral intrínseca em resposta ao aparecimento de um estímulo escolhido ou desejado (DONCHIN; SPENCER; WIJESINGHE, 2000).

- **Potencial evocado visual:** foi a base da primeira tentativa sistemática de desenvolver uma forma de interface entre o cérebro e o computador. Vidal (1977) propôs registrar o potencial visual evocado da pessoa para estímulos visuais que piscavam com frequências distintas. Estas frequências seriam sincronizadas com o sinal de eletroencefalograma captado no córtex visual, possibilitando a identificação do objeto para o qual a pessoa estivesse olhando. Esse método é utilizado na detecção da direção de visão do usuário e exige, conseqüentemente, que a pessoa tenha bom controle visual (DONCHIN; SPENCER; WIJESINGHE, 2000; BEVERINA et al., 2003).
- **Potencial evocado visual estado-permanente:** estes sinais são respostas naturais para estímulos visuais em frequências específicas. Quando a retina é excitada por um estímulo visual dentro da faixa de 3,5 a 50 Hz, o cérebro gera um potencial com a mesma frequência do estímulo, ou múltiplos dela. Isso é utilizado para identificação de qual estímulo visual externo o usuário do sistema está olhando dentre vários estímulos (BEVERINA et al., 2003).
- **Potencial lento do córtex:** caracterizado por variações lentas do potencial geradas no córtex depois de 0,5 a 10 segundos do início do estímulo (Figura 2.13); esses estímulos são produzidos geralmente por movimentos (MILLAN et al., 2004). Apesar de suas alterações lentas serem um fator limitante, assim como o extensivo treinamento necessário para aprender a controlar esse tipo de potencial, não são dependentes de algum estímulo (BEVERINA et al., 2003). Estudos comprovam que é possível, mediante treinamento da pessoa, controlar estes sinais produzindo sinais de controle (BEVERINA et al., 2003).

Figura 2.13: Transmissão de diferentes intenções de sinais.



Fonte: Vallabhaneni, Wang e He (2005).

O potencial cortical lento negativo indica a soma dos potenciais sincronizados, enquanto o positivo indica redução de potenciais sincronizados.

2.4 Análise do sinal

As interfaces cérebro-computador utilizam diferentes técnicas de processamento para analisar os sinais EEG. Essas técnicas podem ser agrupadas como transientes no domínio do tempo ou da frequência. As ferramentas de análise de sinal mais utilizadas pelas interfaces cérebro-computador baseadas em eletroencefalograma são:

- **Transformada de Fourier:** permite decompor um sinal complexo em um conjunto de sinais senoidais bem localizados no domínio da frequência. Dessa forma, pode-se verificar a quantidade de energia que o sinal possui nas diversas componentes de frequência. Segundo Leuchter et al. (1993), a Transformada de Fourier (TF) é uma boa ferramenta para análise de sinais estacionários³. Aplicando-se a TF em um sinal, a informação de tempo é omitida; nesse caso, apenas a informação de frequência está presente. Dessa forma, a TF é ineficiente para representar eventos transientes em um sinal, já que não é capaz de discriminar o instante em que o evento ocorreu, como: tendências, mudanças abruptas e começo e fim de eventos, sendo, portanto, perdidos quando analisados no domínio da frequência (LEUCHTER et al., 1993).
- **Transformada de Fourier de Curto Termo:** Gabor em 1946, propôs, para resolver o problema da TF em sinais não estacionários⁴, analisar o sinal em janelas (KAMMLER, 2008). A adaptação de Gabor chamou-se de *Short-time Fourier Transform* (STFT). A STFT fornece a informação de quando e em qual frequência ocorreu um evento no sinal. Contudo, essa informação tem precisão limitada pelo tamanho da janela. A STFT popularizou-se a partir de 1965, quando uma implementação

³Sinais estacionários: Apresentam os mesmos componentes de frequência durante toda sua duração.

⁴Sinais não estacionários: Apresentam os componentes de frequência variável durante sua duração.

algorítmico-computacional da STFT foi desenvolvida. Essa solução ficou conhecida como: *Fast Fourier Transform* (FFT) (KAMMLER, 2008). Na FFT intervalos de tempo mais longos para cada janela permitem melhor resolução de frequência, mas quanto mais longa a janela, menor fica a resolução temporal.

2.5 Imaginação do movimento

A imaginação do movimento é considerada como um processo mental dinâmico no qual a pessoa simula internamente uma tarefa motora, sem que ocorra o movimento de quaisquer segmentos corporais associados a esta tarefa (GENTILI; PAPAXANTHIS; POZZO, 2006). Segundo Dechent, Merboldt e Frahm (2004), a imaginação do movimento está contida na mesma categoria de processos que envolvem a programação, planejamento e preparação para execução de tarefas motoras. A utilização do EEG tem se mostrado uma importante ferramenta de análise do funcionamento cerebral durante a imaginação do movimento (BIRCH; BOZORGZADEH; MASON, 2002).

Algumas pesquisas utilizam o conceito da imaginação do movimento para gerar padrões distintos dos sinais de eletroencefalograma (HUNG et al., ; REZAEI et al., 2006; XIAO; MU; HU, 2009; YI et al., 2013). Elas utilizam os ritmos beta e gamma, conforme mencionado na Seção 2.2, a diminuição da intensidade do ritmo gamma está associada ao movimento, real ou imaginário (STEPHEN et al., 2001). De forma semelhante, os ritmos alpha e beta também tende a ter sua potência reduzida com a presença de um movimento (PFURTSCHELLER et al., 2006). O aumento desta energia caracteriza um estado de sincronização, enquanto a diminuição da energia representa uma dessincronização.

Segundo Stephen et al. (2001), a realização de um movimento ou da simples preparação de um movimento gera uma dessincronização na atividade neuronal, o que reduz a presença dos ritmos gamma e beta. De forma parecida, o fim do movimento, seguido de relaxamento, acaba por gerar uma maior sincronia dessa atividade, o que causa um aumento dos ritmos gamma e beta, esse comportamento aparece também na imaginação do movimento (PFURTSCHELLER et al., 2006).

Através das técnicas de processamento de sinais e reconhecimento de padrões, é possível extrair informações importantes dos sinais emitidos por um processo de eletroencefalografia. Entretanto, a interpretação das diversas faixas de frequências dos sinais cerebrais de uma forma mais precisa ainda representa grandes desafios. Segundo Tehovnik, Woods e Slocum (2013), são necessários mais estudos sobre o assunto a fim de compreender como o cérebro, por meio dos seus sinais, se comporta para que a transferência das informações para um dispositivo externo se torne mais confiável.

É importante ressaltar que, devido à baixa resolução espacial, da grande atenuação e espalhamento sofridos pelo sinal na passagem pelo crânio, é necessário usar técnicas de processamento de sinais e reconhecimento de padrões para melhorar a relação sinal-ruído e o desempenho das interfaces cérebro-computador baseadas em eletroencefalografia. O reconhecimento de padrões é um recurso usual nos casos onde a parametrização ou modelagem matemática envolve alto nível de complexidade, como é o caso dos sinais cerebrais relacionados a imaginação de movimento. As técnicas de inteligência artificial, como as Redes Neurais Artificiais, Máquinas de Vetor de Suporte e Árvores de Decisão, proporcionam uma solução eficiente na extração de características (VALLABHANENI; WANG; HE, 2005).

2.6 Limites e possibilidades nas investigações sobre a interface cérebro-computador

Esta seção apresenta trabalhos de investigação que tratam de interfaces cérebro-computador para auxílio de pessoas com graves comprometimentos motores, cuja finalidade é o controle de uma cadeira de rodas inteligente.

Durante este estudo foram selecionados artigos publicados entre 2012 e 2016 nas bases de dados: ACM Digital Library (ACM), IEEE Xplore Digital Library (IEEE Xplore), Scientific Electronic Library Online (SciELO), Springer Link (Springer) e US National Library of Medicine (PubMed). Utilizaram-se como descritores: interface cérebro-computador, cadeira de rodas inteligente e EEG, não se fazendo restrição aos idiomas português, espanhol e inglês. A Tabela 2.2 apresenta a quantidade de estudos encontrados durante a busca para as bases selecionadas, totalizando 373 estudos.

Tabela 2.2: Quantidade de estudos encontrados.

Fonte	Quantidade
ACM	1
IEEE Xplore	259
PubMed	3
SciELO	2
Springer	108
TOTAL	373

Fonte: Autor.

Deste número de publicações foram selecionados 11 estudos. O critério de seleção ocorreu a partir da identificação dos estudos que tratam de interfaces cérebro-computador

para pessoas com graves deficiências motoras (FARIA; REIS; LAU, 2012); outros que envolvem interfaces que podem ser utilizadas por essas pessoas (BHATTACHARYYA; SHIMODA; HAYASHIBE, 2016; EDELMAN; BAXTER; HE, 2016; LAY; PIZARRO, 2015; EL-MADANI et al., 2015; WITKOWSKI et al., 2014); outros que envolvem interfaces cuja finalidade é o controle de cadeira de rodas inteligente (EID; GIAKOU MIDIS; SADDIK, 2016; KAUFMANN; HERWEG; KÜBLER, 2014; WANG et al., 2014; JOSE; LOPES, 2015; CARLSON; MILLAN, 2013) e estar disponível *online* gratuitamente. A Tabela 2.3 apresenta os trabalhos selecionados.

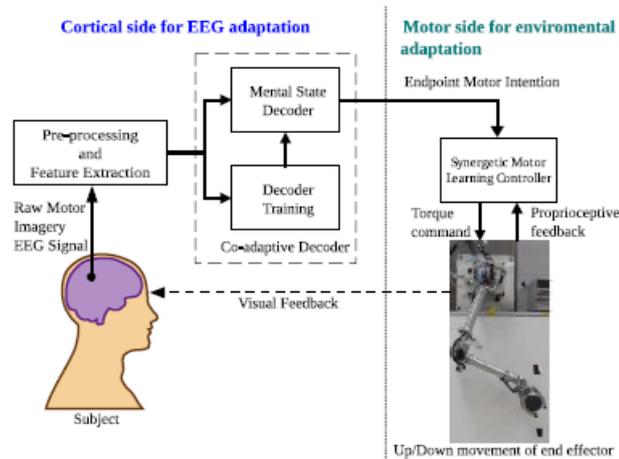
Tabela 2.3: Trabalhos selecionados.

Publicação/Ano	Método
Bhattacharyya, Shimoda e Hayashibe (2016)	SVM
Eid, Giakoumidis e Saddik (2016)	-
Edelman, Baxter e He (2016)	Bayes
Jose e Lopes (2015)	-
Lay e Pizarro (2015)	BCI2000
El-Madani et al. (2015)	Bayes
Kaufmann, Herweg e Kübler (2014)	Árvore de decisão
Witkowski et al. (2014)	BCI2000
Wang et al. (2014)	SVM
Carlson e Millan (2013)	Gaussiano
Faria, Reis e Lau (2012)	RNA; SVM; Bayes

Fonte: Autor.

Bhattacharyya, Shimoda e Hayashibe (2016), desenvolveram uma interface cérebro-computador utilizando imaginação do movimento para controlar um sistema robótico. Nesse trabalho, foi utilizado a imaginação da mão direita e esquerda, correspondendo às direções x e y, respectivamente. 14 eletrodos (F3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 e F4), localizados nas regiões do córtex frontal e motor, foram utilizados para capturar os sinais EEG, com uma taxa de amostragem de 128 Hz e um filtro passa-banda de 0,2 - 45 Hz. A Figura 2.14 apresenta os componentes de controle que foram desenvolvidos nesse estudo.

Figura 2.14: Componentes de controle.



Fonte: [Bhattacharyya, Shimoda e Hayashibe \(2016\)](#).

O participante observa a posição atual do robô, em seguida, realiza tentativas para gerar o sinal relacionado à imaginação do movimento necessário para movimentar o robô. O processo envolve a filtragem e extração de características do sinal EEG. Em seguida, as características são alimentadas como entradas para o decodificador, para identificar o estado (esquerda / direita). A saída decodificada é, então, transmitida ao robô como comandos para movê-lo para cima ou para baixo no plano vertical. O sistema emprega a transformada *wavelet*, para extração de características e um classificador: *Support Vector Machine* (SVM), para o reconhecimento de padrão entre os dois estados mentais (imaginação da mão direita e esquerda). Os usuários alcançaram com sucesso os seus objetivos com uma precisão média de 75%.

No trabalho de [Eid, Giakoumidis e Saddik \(2016\)](#), o sistema de controle da CRI é baseado no rastreamento dos olhos. Os pesquisadores preocuparam-se em auxiliar pessoas com deficiências motoras advindas de doenças, como ELA. A Figura 2.15 apresenta o a cadeira de rodas destacando os vários componentes.

Figura 2.15: CRI e seus componentes.

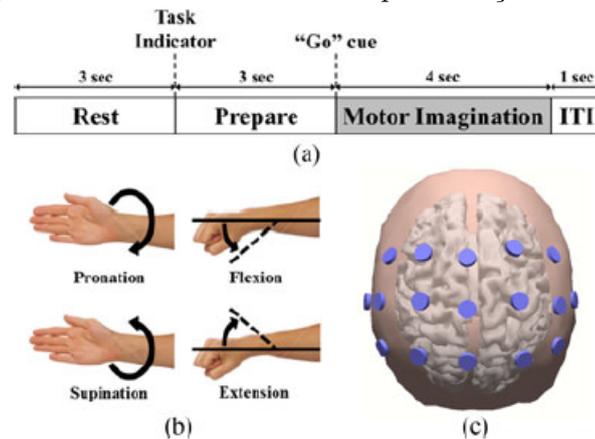


Fonte: [Eid, Giakoumidis e Saddik \(2016\)](#).

Nesse trabalho, o sistema permite que uma pessoa com deficiência motora possa controlar uma cadeira de rodas através do rastreamento dos olhos. O sistema proporciona uma navegação contínua em tempo real em ambientes desconhecidos; o processo de calibração é realizado através de uma interface gráfica. A interface funciona da seguinte maneira: cada célula contém um único elemento para ser controlado (tal como um botão, um campo de texto ou uma imagem). Sempre que o usuário olha para uma célula em particular, o elemento correspondente é destacado e assim, um piscar duplo corresponderia a clicar sobre o elemento de controle. Por exemplo, para o usuário mover a cadeira de rodas para a frente, tem que olhar para a célula correspondente ao botão de “seta para cima”, e uma vez em destaque, deve piscar duas vezes os olhos para executar a ação. O objetivo dos testes era completar uma viagem entre o quarto, sala de jantar e o escritório. Os usuários tiveram uma precisão média de 70% no controle da CRI. O *software* do sistema de controle da CRI é uma combinação de códigos: LabVIEW, MATLAB, C++ e C.

[Edelman, Baxter e He \(2016\)](#) utilizaram, para o desenvolvimento de uma interface cérebro-computador, a imaginação de quatro movimentos relacionados à mão direita: flexão, extensão, supinação e pronação. Essas tarefas foram escolhidas para representar ações complexas envolvidas no controle motor fino da mão que poderia ser aplicada para o controle de dispositivos. Cinco indivíduos (três deles homens, com idade média de 25 anos) com experiência na utilização de interfaces cérebro-computador participaram deste estudo. Os indivíduos foram sentados em uma cadeira na frente de uma tela de computador e instruídos a realizar movimentos reais e imaginados, relacionados à flexão, extensão, supinação ou pronação do lado direito. Cada ensaio foi concebido da seguinte forma: um descanso de 3 segundos; em seguida, durante 3 segundos um ícone indica que tarefa que será executada. Finalmente durante 4 segundos o indivíduo realiza de forma contínua a tarefa de imaginação do movimento indicada. Durante este período, o indivíduo foi instruído a se concentrar na tela do computador e relaxar sua mente como preparação para a próxima tarefa (Figura 2.16 [(a) e (b)]).

Figura 2.16: Estrutura de testes para seleção de dados.

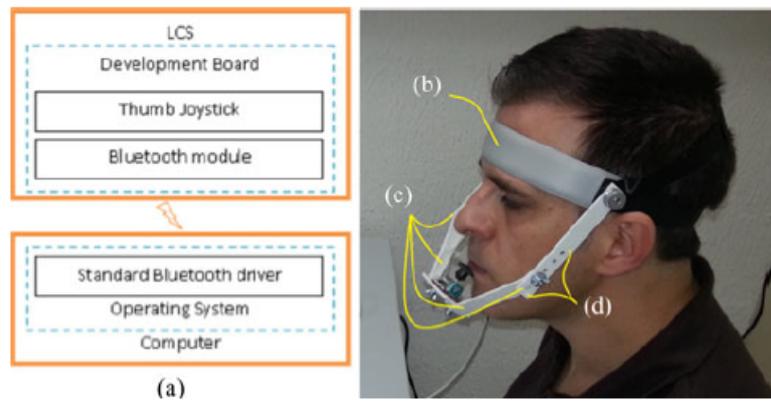


Fonte: [Edelman, Baxter e He \(2016\)](#).

Os dados EEG foram adquiridos a partir de 17 eletrodos (FC3, FC1, FCz, FC2, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CP1, CPz, CP2, CP4), localizados nas regiões do córtex motor e parietal, com uma taxa de amostragem de 100 Hz e um filtro passa-banda de 2 - 30 Hz, em destaque na Figura 2.16 (c). O sistema utiliza a transformada wavelet, para extração de características e um classificador Naive Bayes para o reconhecimento de padrão, a precisão apresentada foi de 82,2%.

No trabalho de [Jose e Lopes \(2015\)](#), foi desenvolvida uma interface para acionamento de CRI controlada pelo lábio, especialmente concebida para pessoas com tetraplegia. O objetivo desse estudo foi de medir a capacidade humana em controlar a interface e ainda obter a taxa de transferência (bits/s) comparável às interfaces desenvolvidas para pessoas com tetraplegia. A Figura 2.17 apresenta os componentes da interface controlada pelo lábio ou *Lip Control System* (LCS).

Figura 2.17: Componentes da interface: (a) Arquitetura. (B) Apoio de cabeça. (c) Joystick. (d) Ajuste.



Fonte: Jose e Lopes (2015).

O *joystick* foi utilizado como um sensor para controlar as direções de movimento da cadeira de rodas; um dispositivo foi fixado à cabeça como um *headset* com um micro *joystick* próximo à boca para o controle pelo lábio inferior. A plataforma para o desenvolvimento da interface utilizou os componentes: placa de desenvolvimento Arduino Mega ADK, módulo *Bluetooth* Bluegiga WT12, módulo *Bluetooth* Roving RN-42 e *joystick* de polegar. O *hardware* do LCS consiste em uma placa de desenvolvimento microcontrolada que recebe os dados do *joystick*, interpretando-os e convertendo-os em valores compatíveis com o movimento desejado. Os dados interpretados são transmitidos por meio do módulo *Bluetooth*. Com o LCS controlado pelo lábio, foi possível atingir 2,59 bits/s, 2,60 bits/s e 1,06 bits/s respectivamente, para as tarefas unidirecional horizontal, unidirecional vertical e multidirecional. Em comparação com interfaces controladas pelo polegar, o lábio conseguiu atingir em média 62,2% dos valores obtidos pelo polegar.

Lay e Pizarro (2015), apresentaram uma interface cérebro-computador usando o dispositivo Emotiv EEG. Nesse estudo foi utilizado o reconhecimento de expressões faciais para controle de um robô móvel. O Emotiv EEG é um produto oferecido pela empresa Emotiv constituído por 14 eletrodos (F3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, F4), localizados nas regiões do córtex frontal e motor. Ele possui um giroscópio de dois eixos, se comunica com o computador através da comunicação por *Bluetooth*, o reconhecimento de padrões é realizado por um *software* proprietário, que pode ser processados através do *software development kit* (SDK), que está disponível para diferentes plataformas de programação. Para a implementação do sistema a linguagem de programação utilizada foi C#. A arquitetura de *hardware* utilizada no sistema é apresentada na Figura 2.18.

Figura 2.18: Arquitetura do sistema.

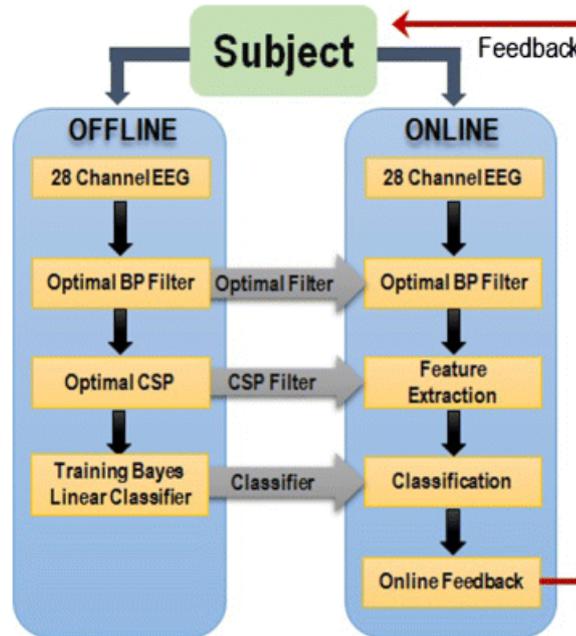


Fonte: [Lay e Pizarro \(2015\)](#).

O sistema considera cinco ações de movimento disponíveis: frente, esquerda, direita, ré e parar, cada um relacionado a um evento. Quando um evento é gerado, o Emotiv detecta um conjunto de sinais correspondente à ação determinada, por exemplo, abrir e fechar os olhos, mastigar, sorrir, pestanejar e franzir a testa. Para validar o sistema, foram utilizados seis pessoas do sexo masculino, com idade entre 24 e 26 anos de idade. Os resultados demonstraram uma precisão média de 81% na classificação das expressões faciais.

[El-Madani et al. \(2015\)](#) desenvolveram uma interface cérebro-computador baseada na imaginação dos movimentos das mãos esquerda e direita, resultando em um sistema com duas classes de saída. A Figura 2.19 lustra o esquema do sistema, nomeado como esquema DTU-BCI.

Figura 2.19: Esquema do sistema DTU-BCI.



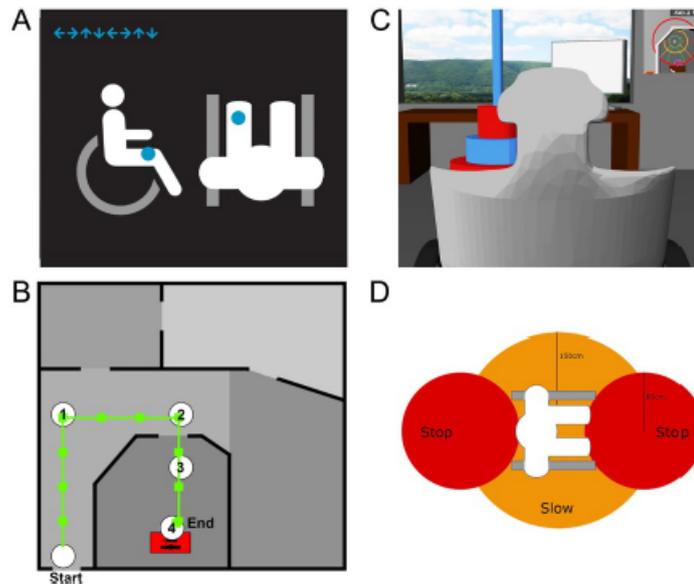
Fonte: [El-Madani et al. \(2015\)](#).

Experimentos *offline* foram realizados utilizando a imaginação dos movimentos da mão esquerda/direita, para :(i) determinar as frequências ideais que deram melhor discriminação entre as classes, (ii) criar um filtro para extração de características e (iii) treinar um classificador para medições. Nas medições *online*, os dados são filtrados por filtros passa-banda (obtidos a partir da análise *offline*); em seguida, as características são extraídas utilizando o procedimento de extração de características a partir da análise *offline* e finalmente, o vetor característica é classificado usando o classificador treinado. No sistema DTU-BCI, foram utilizados 28 eletrodos (posicionados ao redor do córtex motor). Doze indivíduos saudáveis participaram dos testes (sete homens e cinco mulheres com uma idade média de 23 anos). Para o processo de extração de característica, foi utilizado o algoritmo *Common Spatial Patterns* (CSP), o classificador utilizado foi o *Bayes linear classifier* (BLC). O sistema apresentou precisão média de 80%.

[Kaufmann, Herweg e Kübler \(2014\)](#), validaram uma interface cérebro-computador baseada em potenciais relacionados a eventos evocados-tátil para controlar uma cadeira de rodas. Foram posicionados quatro estimuladores táteis para representar direções de navegação (coxa esquerda: mover para a esquerda; coxa direita: mover para a direita; abdômen: avançar; parte inferior do pescoço: mover para trás). Os sinais EEG foram adquiridos a partir de 16 eletrodos (Fz, FC1, FC2, C3, Cz, C4, CP1, CP2, P7, P3, Pz, P4, P8, O1, Oz e O2), localizados na região do córtex motor, a uma taxa de amostragem de 512 Hz. Foi utilizado um filtro passa-banda entre 0,1 e 60 Hz, o sistema foi implementado utilizando

a linguagem de programação C++. Esse sistema exige uma calibração inicial por parte dos usuários para geração dos parâmetros de referência. A Figura 2.20 apresenta o *design* experimental proposto (A: tela do visor apresentada durante a fase de calibração. B: Vista de cima da planta. C: *Screenshot* do ambiente virtual (vista de trás do pescoço, apoio da cadeira de rodas. D: Colisão da cadeira de rodas. Quando frontalmente aproximando de um objeto (ou seja, se um objeto entra na zona de “stop” marcados em laranja), a cadeira de rodas fica parada para evitar a colisão.).

Figura 2.20: Imagem da tela apresentada durante a fase de calibração.

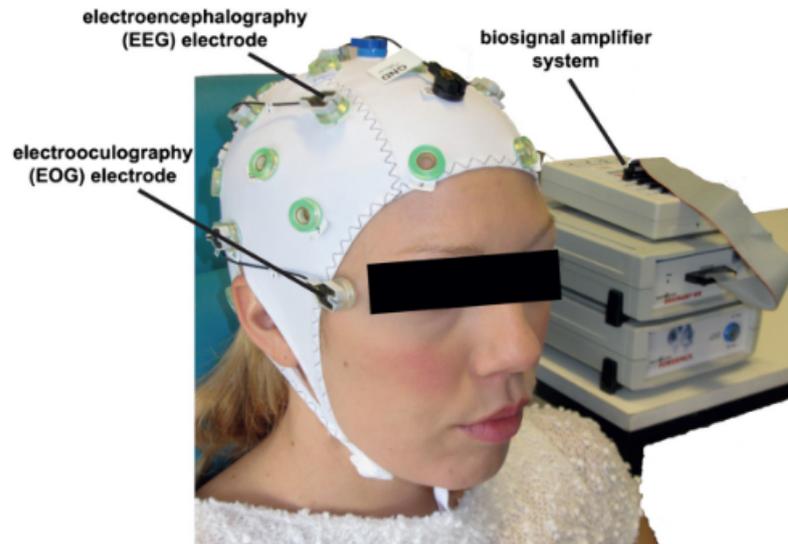


Fonte: Kaufmann, Herweg e Kübler (2014).

Os dados para a classificação da direção foram originados a partir da etapa de calibração. Uma árvore de decisão foi utilizada para classificar os dados de entrada. Quinze participantes foram utilizados para validação do sistema. Os resultados demonstraram uma precisão média de 87,5% classificação dos dados.

Witkowski et al. (2014) desenvolveram uma interface cérebro-computador híbrida, baseada na fusão dos sinais EEG e EOG com o objetivo de aumentar a confiabilidade do sistema. Doze voluntários saudáveis (8 masculinos, 4 femininos, com idade média de 28 anos) foram convidados para participar de uma sessão experimental de 1 hora. Os sinais EEG foram adquiridos a partir de 5 eletrodos (F3, T3, C3, P3, e CZ), localizados nas regiões do córtex frontal e motor. O EEG foi gravado em uma taxa de amostragem de 200Hz. Foi utilizado um filtro passa-banda entre 0.4 e 70Hz. A Figura 2.21 ilustra o posicionamento dos sensores EEG e EOG.

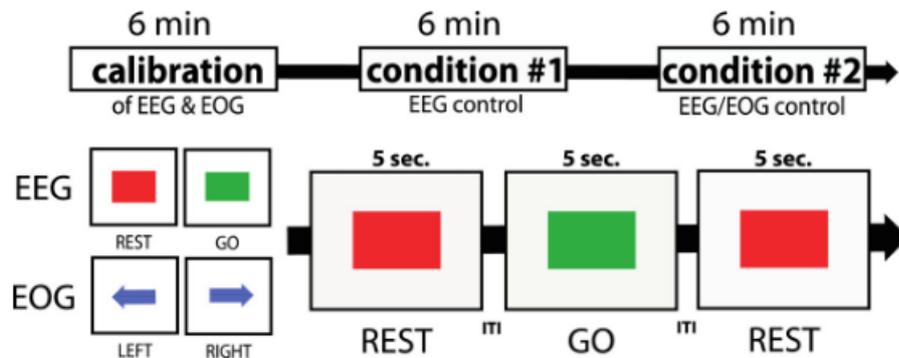
Figura 2.21: Posicionamento dos sensores EEG e EOG.



Fonte: Witkowski et al. (2014).

Durante a calibração EEG, um quadrado vermelho (indicando para descansar) ou quadrado verde (indicando a imaginação do movimento) foi mostrado ao participante. Para a calibração EOG, o participante foi convidado a olhar para a esquerda (seta azul à esquerda) ou para a direita (seta azul à direita). A calibração do sistema foi composta por duas partes: na primeira parte, os participantes foram instruídos a descansar ou imaginar o movimento de fechar a mão, na sequência de uma sugestão visual (quadrado vermelho: *REST*; quadrado verde: *GO*) exibido na tela do computador (Figura 2.22). Os resultados demonstraram uma precisão média de 63,59% de acerto na classificação dos dados.

Figura 2.22: Delineamento experimental.



Fonte: Witkowski et al. (2014).

O trabalho de [Wang et al. \(2014\)](#) consiste em uma interface que utiliza EEG com um sistema síncrono por meio do P300. O sistema é constituído por quatro componentes: um sistema de aquisição de EEG, um computador principal, um módulo de comunicação e uma cadeira de rodas robótica. O projeto mecatrônico do sistema é ilustrado na [Figura 2.23](#).

Figura 2.23: Projeto mecatrônico do sistema.

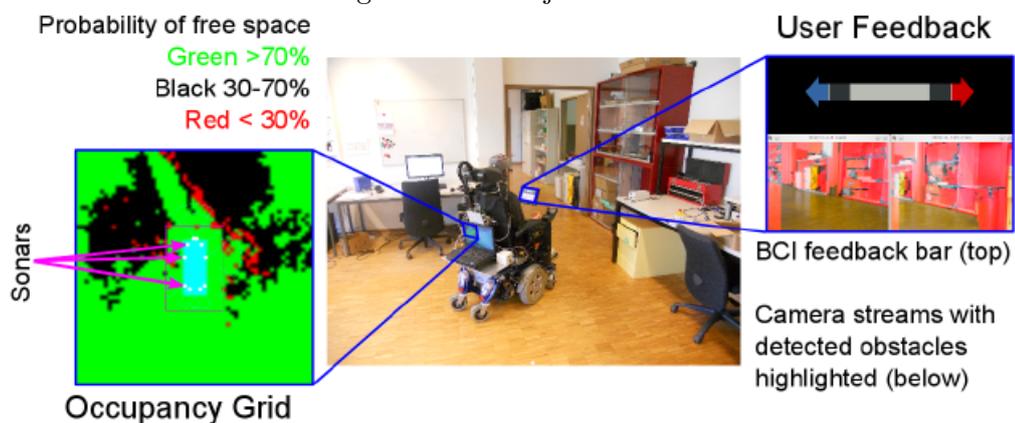


Fonte: [Wang et al. \(2014\)](#).

Os sinais de EEG são amplificados, amostrados a uma taxa de 250 Hz e filtrados entre 0,5 e 100 Hz (filtro passa-banda). Nesse sistema, 15 eletrodos (FC3, FCz, FC4, C3, Cz, C4, CP3, CPZ, CP4, P3, Pz, P4, O1, OZ, e O2), localizados nas regiões do córtex frontal e motor, foram utilizados para captar os sinais EEG relacionados a imaginação do movimento (mão direita e esquerda), enquanto os potenciais P300 são detectados a partir do piscar dos olhos, fazendo uso de mais 2 eletrodos (FP1 e FP2). O sistema foi desenvolvido utilizando a linguagem de programação C++. O objetivo deste trabalho foi testar a capacidade dos usuários em realizar tarefas de movimentação: frente, aceleração, desaceleração e deslocamento para trás. Para avaliar o desempenho do sistema, quatro indivíduos saudáveis (quatro homens com idade entre 24 a 32 anos) participaram de duas experiências *online*. Os resultados demonstraram uma precisão média de 93%.

No projeto desenvolvido por [Carlson e Millan \(2013\)](#) a interface cérebro-computador utiliza imaginação de movimentos das mãos esquerda e direita para controlar a CRI (Figura 2.24). 16 eletrodos (Fz, FC3, FC1, FCz, FC2, FC4, C3, C1, Cz, C2, C4, CP3, CP1, CPz, CP2 e CP4), localizados nas regiões do córtex frontal e motor, foram utilizados para captura dos sinais EEG a uma taxa de 512 Hz, um filtro passa-banda entre 4 e 48Hz foi utilizado para filtrar os sinais EEG. A análise de variáveis canônicas (CVA) foi utilizada para selecionar as características relacionadas à imaginação dos movimentos. Essas características foram usadas para treinar um classificador Gaussiano. O sistema foi testado com 4 indivíduos, os resultados demonstraram uma precisão média de 94%.

Figura 2.24: Projeto da CRI.



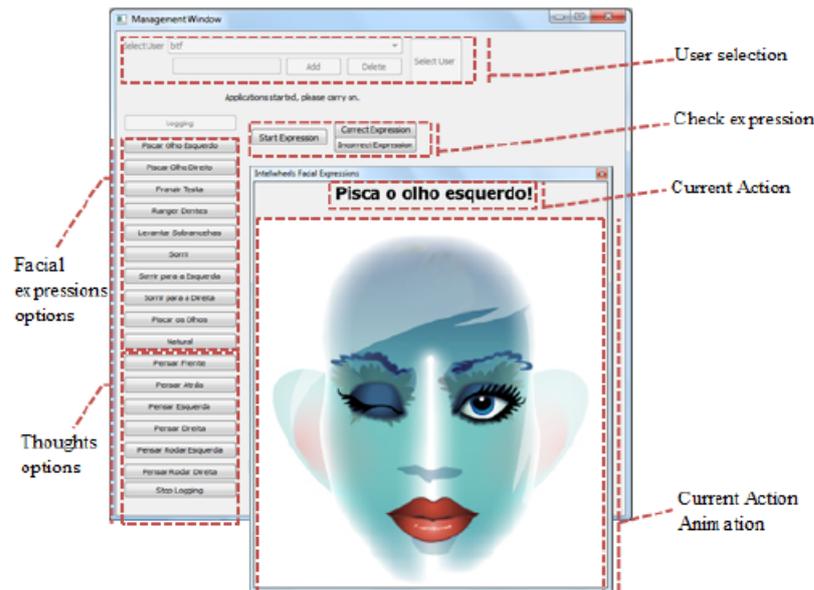
Fonte: [Carlson e Millan \(2013\)](#).

O sistema da CRI conta com um *joystick* remoto que atua como uma interface entre um computador portátil e cadeiras de rodas. Dessa forma, é possível controlar a cadeira de rodas remotamente. Foi adicionado um par de *encoders* nas rodas motoras centrais, a fim de fornecer à cadeira de rodas um *feedback* sobre o seu próprio movimento, um array de sonares e duas câmeras foram adicionadas à cadeira de rodas para fornecer *feedback* do ambiente para o controlador.

A interface cérebro-computador, desenvolvida por [Faria, Reis e Lau \(2012\)](#), utilizou algoritmos de classificação, tais como: Naive Bayes, Rede Neural Artificial e Máquina de Vetor de Suporte para o reconhecimento de padrões em sinais EEG baseados em expressões faciais. Experimentos foram realizados utilizando 30 indivíduos que sofrem de paralisia cerebral. Os resultados obtidos mostraram que o classificador baseado em RNA foi mais eficiente em relação aos outros métodos, permitindo melhorar os resultados em 57%. A aquisição dos sinais EEG foi realizada fazendo uso do dispositivo Emotiv EEG, 14 eletrodos (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 e AF4), localizados nas regiões do córtex frontal e motor, foram utilizados. A taxa de amostragem utilizada foi de 128Hz. Os dados extraídos dos sinais EEG correspondem às seguintes expressões

faciais: sorriso; piscar os olhos; piscar o olho esquerdo; piscar o olho direito e cerrar as sobrancelhas. Uma aplicação foi desenvolvida para registrar os dados brutos do sensor EEG (Figura 2.25). A metodologia consistiu em solicitar aos utilizadores com paralisia cerebral que imitassem várias expressões faciais. Um terapeuta ocupacional foi envolvido no processo, com o objetivo de verificar se as expressões faciais foram realizadas corretamente. Apenas as expressões corretas foram adicionadas ao conjunto de dados final.

Figura 2.25: Aplicação para registrar as expressões faciais.



Fonte: Faria, Reis e Lau (2012).

2.7 Considerações finais

A interface de controle da CRI, desenvolvida por Eid, Giakoumidis e Saddik (2016), é baseada no rastreamento dos olhos. Os pesquisadores preocuparam-se em auxiliar pessoas com graves deficiências motoras. Nesse caso, o sistema requer olhar contínuo e consistente durante o tempo de navegação. A atenção do usuário é compartilhada entre a percepção do ambiente e a tomada de decisão, que consiste em identificar a seta correspondente à direção desejada e em seguida, confirmar a opção com um duplo piscar de olhos. Dessa forma, o usuário terá dificuldade para olhar em torno do meio ambiente durante a navegação.

Jose e Lopes (2015), desenvolveram uma interface especialmente para pessoas com tetraplegia: o dispositivo de entrada de dados é baseado em um *joystick* controlado pelo lábio inferior. Nesse caso, é necessário considerar que uma interface controlada pelo lábio inferior deve evitar o contato com áreas úmidas da boca por questões de higiene. Este é um fator muito importante, principalmente no caso das pessoas com tetraplegia que pre-

cisam da ajuda de outras pessoas para a limpeza da interface. No trabalho de [Kaufmann, Herweg e Kübler \(2014\)](#), a interface cérebro-computador é baseada em potenciais relacionados a eventos evocados-tátil. Dentro dessa área de investigação existe alguns pontos a serem tratados, tais como a otimização do tempo de treinamento do usuário para operar o sistema.

As interfaces cérebro-computador desenvolvidas nos trabalhos de [Edelman, Baxter e He \(2016\)](#) e [Faria, Reis e Lau \(2012\)](#) utilizaram pelo menos 16 eletrodos para aquisição dos sinais EEG. Quanto maior o número de eletrodos para aquisição e envio dos dados, maior será o consumo de energia do sistema. Conseqüentemente, isso reduz o tempo de utilização do dispositivo em atividades prolongadas.

Os trabalhos de [Wang et al. \(2014\)](#) e [Carlson e Millan \(2013\)](#), dentre os estudados, apresentaram melhores resultados, precisão: de 93% e 94% respectivamente, porém, em ambas as pesquisas, o número de voluntários que participaram dos experimentos representam um grupo pequeno: 4 indivíduos. Já na pesquisa desenvolvida por [Faria, Reis e Lau \(2012\)](#), foram utilizados 30 indivíduos, onde, foi possível observar uma precisão: 57% para uma RNA, 43,55% utilizando SVM e 54,64% com Bayes.

Observa-se que os projetos (Tabela 2.2), de modo geral, possuem características semelhantes, como a adaptação de diferentes tipos de interfaces de controle para facilitar a interação de pessoas com diferentes tipos de deficiência. Entretanto, muitos desses projetos foram desenvolvidos de maneira isolada, sobre arquiteturas e ferramentas próprias. Como consequência, dificuldades quanto à integração, evolução, manutenção, reutilização e testes desses projetos são comuns. Apenas [Lay e Pizarro \(2015\)](#) e [Witkowski et al. \(2014\)](#), utilizaram o sistema BCI2000 para o desenvolvimento das suas pesquisas. O BCI2000 é um ambiente de desenvolvimento para sistemas baseados em interfaces cérebro-computador, o projeto foi desenvolvido pelo *Wadsworth Center, New York State Dpt of Health, Albany, NY* ([SCHALK et al., 2004](#)). Um aspecto, que torna o BCI2000 uma plataforma pouco eficiente, é o fato de não permitir a análise *offline* de dados, sendo esta feita por programas especializados no tratamento e processamento de sinal, como por exemplo o MatLab. Outro aspecto está relacionado ao fato de que os sistemas desenvolvidos a partir do BCI2000 só executam em ambiente Windows.

Devido à relevância destes fatores, a utilização de um modelo computacional na automação de tarefas que permita o desenvolvimento de novas interfaces cérebro-computador apresenta-se como uma iniciativa promissora, podendo viabilizar a utilização de diferentes ferramentas para que pesquisadores possam validar hipóteses em um ambiente integrado. Este trabalho de investigação procura preencher algumas destas lacunas, neste sentido foca-se na criação de um modelo computacional de forma a auxiliar no desenvolvimento de novas tecnologias baseadas em interface cérebro-computador, provendo vantagens quanto

ao reúso, produtividade e evolução das aplicações.

Com o objetivo de validar o modelo computacional proposto, desenvolvemos uma interface cérebro-computador para acionar uma CRI, tendo como ponto de diferenciação em relação aos trabalhos apresentados na Tabela 2.3, a utilização de dois eletrodos (Fp1 e Fp2) para aquisição dos sinais EEG (córtex pré-frontal). Isso permitirá um menor consumo de energia por parte do sistema, possibilitando ao indivíduo a utilização do dispositivo durante longos períodos de tempo. Nenhum dos trabalhos anteriores (Tabela 2.3) utilizou o córtex pré-frontal para aquisição de sinais EEG. É no córtex pré-frontal que ocorre planejamento da ação e do controle do movimento, sendo essa uma das áreas motoras envolvidas na imaginação do movimento. Outro fato que justifica a escolha do córtex pré-frontal para aquisição dos sinais EEG, é que ele fica localizado na área imediatamente atrás da testa. Segundo Webster (2009), a aquisição dos sinais cerebrais consiste no registro da atividade elétrica cerebral, com boa relação sinal/ruído. Para que a qualidade do sinal registrado seja adequada, é necessário que haja um bom contato entre a pele e o eletrodo.

Outro ponto importante, ainda em relação à diminuição do número de eletrodos envolvidos no processo de construção das interfaces cérebro-computador, são os aspectos relacionados à representação social e pessoal do indivíduo com deficiência⁵. A necessidade dos eletrodos para aquisição dos sinais EEG incrementa a percepção simbólico da condição de deficiência que o indivíduo enfrenta. Segue no próximo capítulo a proposta de solução.

⁵A representação social e pessoal da deficiência é uma espécie de lente pela qual o indivíduo é visto e enxerga seu mundo. Essas percepções podem ser repletas de preconceitos, de ambas as partes, pois é um ser social, participando da construção e manutenção das concepções, status e valores da sociedade (FECHIO et al., 2009).

Projeto e desenvolvimento da pesquisa

Este capítulo apresenta o projeto, arquitetura e implementação da interface cérebro-computador desenvolvida no âmbito deste trabalho de investigação, . Como o objetivo geral desse trabalho de investigação consistiu no desenvolvimento de um modelo computacional baseado em interface cérebro-computador que permita o estudo e desenvolvimento de novas tecnologias a partir dos sinais de eletroencefalograma para acionar CRI, pode-se considerar este modelo como um *Framework*¹ que permite o desenvolvimento de aplicações baseadas em interfaces cérebro-computador. O modelo foi projetado com características modulares para facilitar a utilização de parte ou de todos os componentes do sistema.

Não obstante, este trabalho de investigação está inserido no projeto de pesquisa em desenvolvimento no Núcleo de Arquitetura de Computadores e Sistemas Operacionais (ACSO)² intitulado: “Cadeira de Rodas Inteligente com Interface Flexível”, utilizando como premissas a Engenharia de *Software* e Inteligência Artificial. O ACSO é um órgão suplementar de natureza interdisciplinar, previsto no Regimento Geral da Universidade do Estado da Bahia - UNEB, vinculado ao Departamento de Ciências Exatas e da Terra - DCET-I, cadastrado no CNPQ, liderado pelo Prof. Ph. D. Josemar Rodrigues de Souza. Atua em pesquisa, ensino e extensão. Busca a integração efetiva entre grupos de pesquisas, pesquisadores e professores da UNEB e da comunidade científica nacional e internacional.

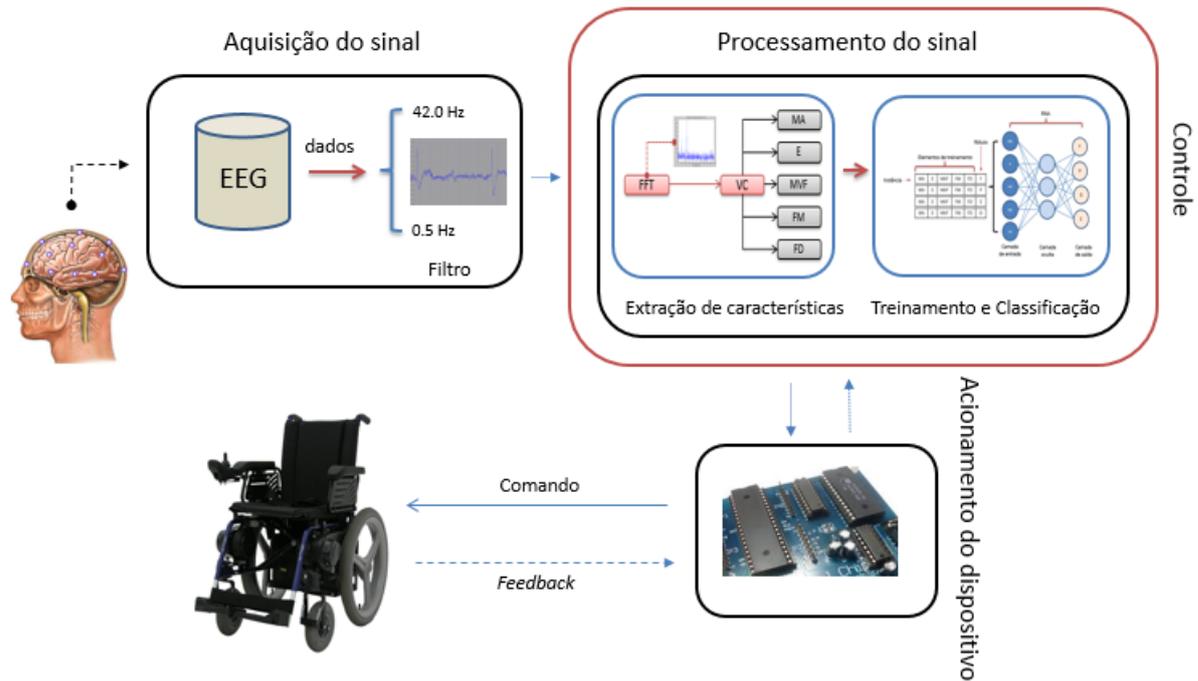
3.1 Modelo computacional

O modelo computacional, ilustrado na Figura 3.1, segue o fluxo proposto por [Norani, Mansor e Khuan \(2010\)](#), que consiste nas etapas de aquisição e pré-processamento dos sinais, extração de características e classificação.

¹Um *Framework* é uma estrutura de classes inter-relacionadas, que corresponde a uma implementação incompleta para um conjunto de aplicações de um determinado domínio. Esta estrutura de classes deve ser adaptada para a geração de aplicações específicas ([JOHNSON, 1992](#))

²Disponível na URL: <http://www.acso.uneb.br/acso>. Acessado em 20/11/2016.

Figura 3.1: Modelo computacional do projeto.



Fonte: Autor.

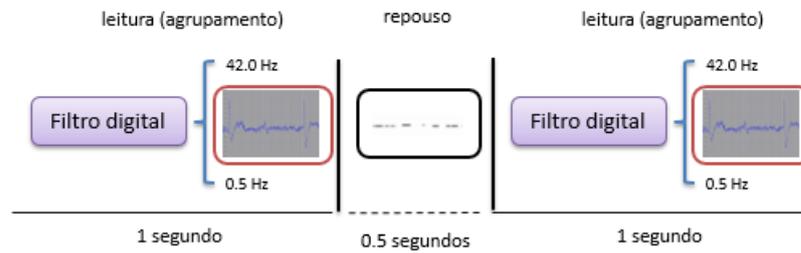
Os sinais EEG brutos são lidos e armazenados em arquivos seguindo o formato *European Data Format* (EDF)³; em seguida, calcula-se a FFT dos sinais EEG. O sistema de controle recebe as informações de amplitude e frequência do sinal EEG e os armazenam em um vetor. Essas informações serão utilizadas na fase de treinamento e caracterização do modelo para acionamento da CRI. Os componentes do modelo computacional, especificados a seguir, têm funções específicas.

3.1.1 Módulo de aquisição do sinal

O primeiro módulo a ser considerado foi o módulo relativo à aquisição dos dados para análise. Os sinais EEG são filtrados por um filtro digital passa-faixa, cujas frequências variam em uma faixa de 0,5 a 42 Hz e amostrados em uma taxa de 160 Hz. O modelo permite a parametrização dos valores do filtro (mínimo e máximo) e taxa de amostragem (detalhado na Subseção 3.2.1), esquema ilustrado na Figura 3.2.

³*European Data Format* é um formato de arquivo padrão projetado para troca e armazenamento de sinais biológicos.

Figura 3.2: Esquema: aquisição do sinal.



Fonte: Autor.

Os sinais EEG filtrados são agrupados em janelas de 1 segundo, com sobreposição de 50% (0.5 segundos). Esses registros são digitalizados e entregues ao módulo de processamento do sinal. Para a realização do processamento digital do sinal foi utilizada a biblioteca: *Java Digital Signal Processing (DSP) collection*⁴, trata-se de um conjunto de classes *open source* desenvolvidas utilizando a linguagem de programação Java. O filtro digital passa-faixa utilizado nesse trabalho de investigação é baseado na API (*Application Programming Interface*) presente no componente *Java (DSP) collection*.

3.1.2 Módulo de processamento do sinal

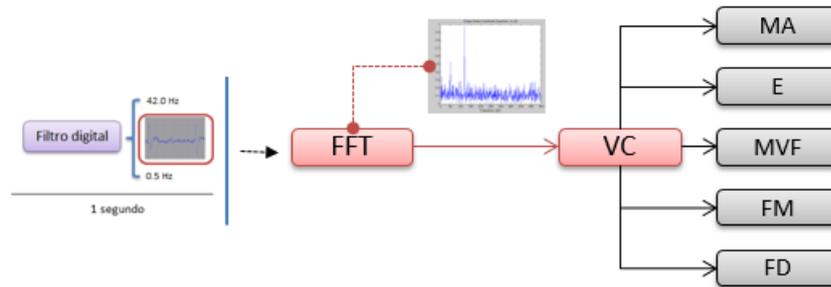
O módulo de processamento do sinal já inclui o pré-processamento aplicado ao sinal adquirido, que passa pela filtragem da faixa de interesse. Essa etapa consiste no processo de extração de características, treinamento e classificação dos dados.

Após a recepção dos dados, calcula-se a FFT a partir dos dados de entrada. Para o cálculo da FFT foi utilizada a biblioteca: *The Apache Commons Mathematics Library*⁵ (Documentação no Anexo B). O sistema de controle, baseado em informações de amplitude e frequência dos sinais EEG, calcula os atributos que contém os padrões de entrada para o vetor de características (VC), esquema ilustrado na Figura 3.3.

⁴Disponível para download na URL: <http://www.source-code.biz/dsp/java>. Acessado em 25/09/2015.

⁵Disponível para download na URL: <http://commons.apache.org/proper/commons-math/>. Acessado em 28/09/2015.

Figura 3.3: Esquema: extração de características.



Fonte: Autor.

O vetor de características é composto por cinco atributos, que contém o padrão a ser analisado pelo processo de treinamento e classificação do modelo. Os atributos utilizados nesse trabalho de investigação, segue o modelo definido por [George e Azevedo \(2007\)](#). Esses atributos são definidos como:

- **Média aritmética (MA)**: representa o valor médio do sinal. O atributo é calculado pela Equação 3.1.

$$m(s) = \frac{1}{N} \sum_{n=1}^{n=N} S[n] \quad (3.1)$$

na qual s é o valor do sinal da FFT e N o tamanho do vetor de características (S).

- **Energia (E)**: representa a área sob a curva descrita pela função do sinal elevada ao quadrado, calculado pela Equação 3.2.

$$E_2(s) = \frac{1}{N} \sum_{n=1}^{n=N} |s[n]|^2 \quad (3.2)$$

- **Máximo valor da FFT (MVF)**: representa o ponto onde ocorre o máximo valor da curva da FFT, calculado pela Equação 3.3.

$$MVF(s) = \max(S[n]) \quad (3.3)$$

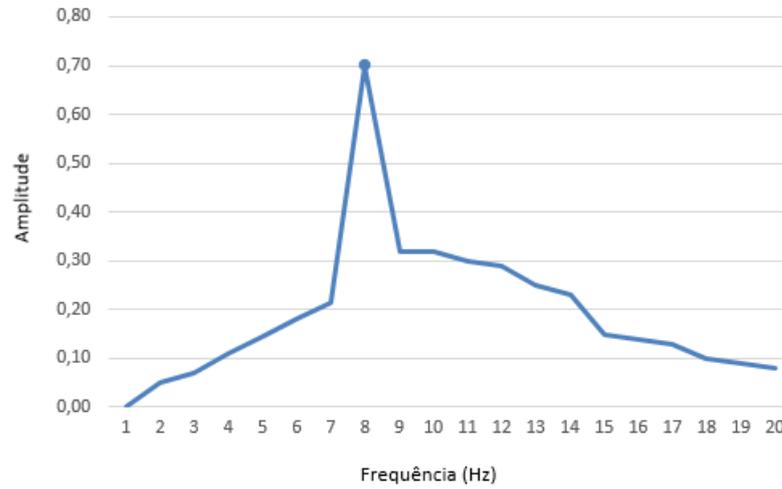
- **Mínimo valor da FFT (FM)**: representa o ponto onde ocorre o valor mínimo da FFT, calculado pela Equação 3.4.

$$FM(s) = \min(S[n]) \quad (3.4)$$

- **Frequência dominante (FD)**: representa a frequência que possui maior amplitude,

é a coordenada X do ponto onde ocorre o máximo valor da curva da FFT. A Figura 3.4 ilustra a frequência principal: 8 Hz.

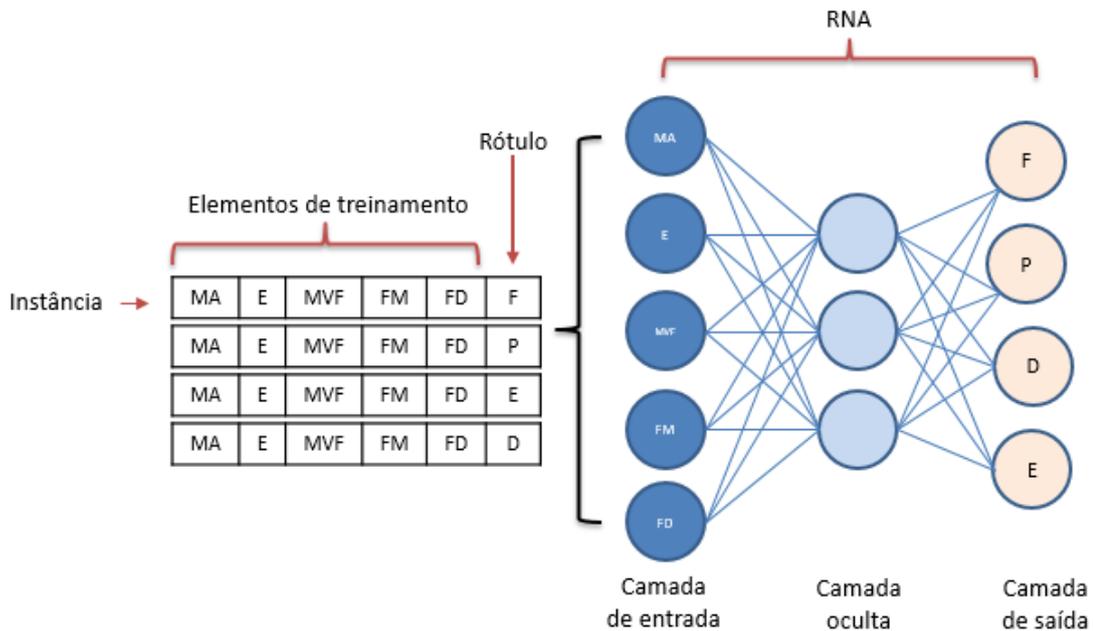
Figura 3.4: Frequência dominante.



Modificado de: [George e Azevedo \(2007\)](#).

O processo de classificação é realizado em duas etapas: treinamento (etapa de aprendizado) e classificação dos dados (testes). Na etapa de treinamento, o modelo do classificador é construído descrevendo um conjunto pré-determinado de classes. A Figura 3.5 apresenta o esquema de comportamento do processo de classificação.

Figura 3.5: Esquema: treinamento e classificação.



Fonte: Autor.

A RNA utilizada é composta por 3 camadas (entrada, oculta e saída). O número de neurônios da camada de entrada da RNA é determinado pela dimensionalidade do vetor de características, sendo cinco. A quantidade de neurônios na camada de saída da rede foi definida como quatro neurônios, visto que o objetivo da rede é discriminar entre quatro estados mentais, produzindo quatro respostas por parte da RNA, sendo: F (frente), P (parar), E (esquerda) e D (direita). Já para a definição do número de neurônios da camada oculta da RNA, utilizaremos a metodologia definida por [Han e Kamber \(2006\)](#), Equação: 3.5, sendo está a opção padrão do modelo. O modelo permite a parametrização da quantidade do número de neurônios da camada oculta da RNA (detalhado na Subseção 3.2.2).

$$N_{hidden} = 2N_{in} + 1 \quad (3.5)$$

Onde N_{in} é o número de neurônios da camada de entrada e N_{hidden} representa o número de neurônios da camada oculta.

Dessa forma, o sistema de controle extrai os atributos do vetor de características (MA, E, MVF, FM e FD) para realizar o treinamento da RNA, ou seja, construir o modelo da RNA, através do conjunto de dados de treinamento. Aos elementos de treinamento,

são associados rótulos de classes às quais cada um pertence (E, D, F e P). Esta etapa é conhecida como aprendizagem supervisionada, pois o rótulo da classe de cada elemento de treinamento é fornecido ao classificador.

No modelo, a RNA será treinada, através de exemplos, para classificar uma determinada imaginação de movimento. A saída dessa RNA indicará o comando de movimentação da CRI (Tabela 3.1).

Tabela 3.1: Comando de movimentação da CRI.

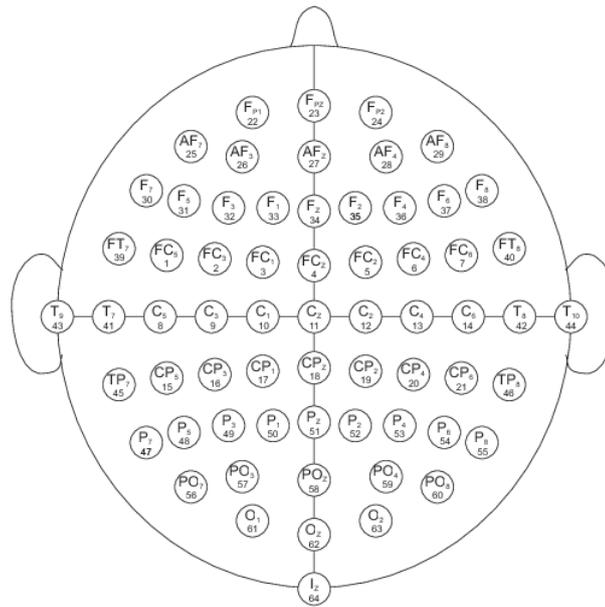
Saída da RNA	Imaginação movimento	Direção (CRI)
E	Punho esquerdo	Esquerda
D	Punho direito	Direita
F	Ambos os punhos	Frente
P	Ambos os pés	Parar

Fonte: Autor.

Os sinais dos registros EEG utilizados para formatação dos elementos de treinamento foram adquiridos através do banco de dados *eegmmidb - EEG Motor Movement/Imagery Dataset*, capturado utilizando o sistema BCI2000 (SCHALK et al., 2004), disponível através do PhysioBank (GOLDBERGER et al., 2000). Este banco de dados, composto por mais de 1500 registros de um e dois minutos de sinais EEG, obtidos a partir de 109 voluntários, sendo quarenta e três pessoas do sexo masculino e sessenta pessoas do sexo feminino, com idade entre 19 e 67 anos de idade. Dados de seis indivíduos (S104, S105, S106, S107, S108 e S109) não foram disponibilizados para determinação do sexo (Mais informações no Anexo A). Todos os indivíduos, não tinham distúrbios neurológicos e não foram treinados anteriormente para a utilização da interface cérebro-computador. A população de voluntários foi extraída dos funcionários do Departamento de Saúde do Estado de Nova York. Todos os indivíduos tinham concluído o ensino médio e a maioria tinha completado quatro anos de ensino superior.

Para a formação da base de dados, os voluntários realizaram diferentes tarefas motoras e imaginadas, registradas pelo sistema BCI2000, fazendo uso de 64 eletrodos, ilustrados na Figura 3.6

Figura 3.6: Distribuição dos eletrodos.



Fonte: [Goldberger et al. \(2000\)](#).

Cada voluntário realizou 14 sessões experimentais: dois ensaios de um minuto, sendo, um minuto com os olhos abertos e um minuto com os olhos fechados (sessões *baseline*), e três ensaios de dois minutos de cada uma das seguintes tarefas:

1. Um alvo aparece no lado esquerdo ou direito da tela. O voluntário abre e fecha o punho correspondente até que o alvo desapareça. Em seguida, o voluntário relaxa.
2. Um alvo aparece no lado esquerdo ou direito da tela. O sujeito imagina abrindo e fechando o punho correspondente até que o alvo desapareça. Em seguida, o voluntário relaxa.
3. Um alvo aparece na parte superior ou inferior da tela. O voluntário abre e fecha ambos os punhos (alvo no topo) ou ambos os pés (alvo na parte inferior) até que o alvo desapareça. Em seguida, o voluntário relaxa.
4. Um alvo aparece na parte superior ou inferior da tela. O voluntário imagina abrindo e fechando ambos os punhos (alvo no topo) ou ambos os pés (alvo na parte inferior) até que o alvo desapareça. Em seguida, o voluntário relaxa.

Em resumo, as sessões experimentais foram as seguintes:

1. *Baseline*, olhos abertos.

2. *Baseline*, olhos fechados.
3. Tarefa 1 (abre e fecha o punho esquerdo ou direito).
4. Tarefa 2 (imagina abrindo e fechando o punho esquerdo ou direito).
5. Tarefa 3 (abre e fecha ambos os punhos ou ambos os pés).
6. Tarefa 4 (imagina abrindo e fechando ambos os punhos ou ambos os pés).
7. Tarefa 1 (abre e fecha o punho esquerdo ou direito).
8. Tarefa 2 (imagina abrindo e fechando o punho esquerdo ou direito).
9. Tarefa 3 (abre e fecha ambos os punhos ou ambos os pés).
10. Tarefa 4 (imagina abrindo e fechando ambos os punhos ou ambos os pés).
11. Tarefa 1 (abre e fecha o punho esquerdo ou direito).
12. Tarefa 2 (imagina abrindo e fechando o punho esquerdo ou direito).
13. Tarefa 3 (abre e fecha ambos os punhos ou ambos os pés).
14. Tarefa 4 (imagina abrindo e fechando ambos os punhos ou ambos os pés).

Os registros são disponibilizados no padrão *European Data Format* (EDF), contendo registros de 64 eletrodos, cada um amostrado em uma taxa de 160 Hz e um canal de anotação. Cada anotação contém um dos códigos:

- T0: corresponde ao descanso, com duração de 4,2 segundos.
- T1: corresponde ao início do movimento real ou imaginário, com duração de 4,1 segundos, do punho esquerdo (sessões 3, 4, 7, 8, 11, e 12) e ambos os punhos (sessões 5, 6, 9, 10, 13 e 14).
- T2: corresponde ao início do movimento real ou imaginário, com duração de 4,1 segundos, do punho direito (sessões 3, 4, 7, 8, 11 e 12) e ambos os pés (sessões 5, 6, 9, 10, 13 e 14).

Neste trabalho de investigação, serão avaliados os registros individuais dos eletrodos posicionados na região do córtex pré-frontal (Fp1 e Fp2). Os registros relacionados à imaginação de movimentos (punho esquerdo, punho direito, ambos os pés e ambos os punhos) serão utilizados para validação do modelo e serão extraídos seguindo a orientação disposta na Tabela 3.2. Esses dados serão submetidos ao processo descrito na Subseção 3.1.1.

Tabela 3.2: Preparação dos dados.

Imaginação movimento	Sessões	Tarefa	Anotação
punho esquerdo	4;8;12	2	T1
punho direito	4;8;12	2	T2
ambos os punhos	6;10;14	4	T1
ambos os pés	6;10;14	4	T2

Fonte: Autor.

No módulo de processamento do sinal, foi utilizada, a ferramenta *Waikato Environment for Knowledge Analysis* (Weka) desenvolvido pelo grupo de aprendizado de máquina da Universidade de Waikato (Nova Zelândia). O Weka⁶ é uma biblioteca *open source*, sob licença GNU (*General Public License*), desenvolvido fazendo uso da linguagem de programação Java. Neste trabalho de investigação, foram utilizados os métodos de classificação e treinamento da RNA, disponível na biblioteca Weka.

Na etapa de classificação dos dados, será utilizado um conjunto de dados formado com elementos de testes e seus rótulos de classe associados (Figura 3.5). A técnica de Validação Cruzada (*10-fold cross-validation*) disponível na biblioteca Weka, foi utilizada para realizar os processos de treinamento e classificação do modelo. No *10-fold cross-validation*, o conjunto de dados é dividido aleatoriamente em 10 partes nas quais cada classe é representada na mesma proporção do conjunto inteiro, aproximadamente. Destas 10 partes, 9 partes são utilizadas para o treinamento e uma serve como base de testes. O processo é repetido 10 vezes, de forma que cada parte seja usada uma vez como conjunto de testes. Ao final, a correção total é calculada pela média dos resultados obtidos em cada etapa, obtendo-se assim uma estimativa da qualidade do modelo (HALL et al., 2009).

Segundo Han e Kamber (2006), a acurácia⁷ e a precisão⁸ são bons indicadores de desempenho na tarefa de classificar dados desconhecidos (não apresentados anteriormente ao classificador). Este conjunto de dados é chamado de conjunto de testes. Para cada elemento de teste, o atributo classe é confrontado com a predição da classe realizada pelo classificador em análise.

Neste trabalho de investigação, será utilizada, como medida de desempenho dos classificadores (Tabela 3.1) a precisão. Nesse caso, o sistema de controle determina a classe de cada instância (Figura 3.5): se a classificação está correta, conta-se como um acerto; caso contrário, conta-se como um erro. Pretende-se conhecer o desempenho do classificador a

⁶Disponível para download na URL: <http://www.cs.waikato.ac.nz/ml/weka/downloading.html>. Acessado em 16/03/2015.

⁷A acurácia de um classificador de um dado conjunto de testes é a porcentagem de elementos do conjunto de testes que são corretamente classificados por ele.

⁸Precisão é a taxa com que todos os exemplos classificados como positivos são realmente positivos.

partir de dados não utilizados na fase de treinamento, pois os dados utilizados na fase de treinamento, já são conhecidos pelo classificador.

Para esse propósito, a sensibilidade (Equação 3.6) e a especificidade (Equação 3.7) podem ser usadas:

- **Sensibilidade (SSB)**: taxa de reconhecimento de verdadeiros positivos, ou seja, é a proporção de instâncias positivas que são corretamente identificadas.
- **Especificidade (ESP)**: taxa de reconhecimento de falsos negativos, ou seja, é a proporção de instâncias negativas que são corretamente identificadas.

Segundo [Han e Kamber \(2006\)](#), essas medidas são definidas como:

$$SSB = \frac{vp}{p} \quad (3.6)$$

$$ESP = \frac{vn}{n} \quad (3.7)$$

Onde **vp** é o número de verdadeiros positivos; **p** é o número de instâncias positivas; **vn** é o número de verdadeiros negativos; **n** é o número de instâncias negativas. Além disso, é possível usar a precisão (Equação 3.8) para obter o percentual de instâncias rotuladas, por exemplo, como: imaginação de movimento do punho esquerdo, que de fato, são instâncias da classe imaginação de movimento do punho esquerdo = E.

$$PCS = \frac{vp}{vp + fp} \quad (3.8)$$

Onde **fp** é o número de falsos positivos. O cálculo da acurácia é definido em função da sensibilidade e da especificidade, calculado pela Equação 3.9.

$$ACR = SSB \frac{p}{p+n} + ESP \frac{n}{p+n} \quad (3.9)$$

O modelo da rede neural artificial utilizado neste trabalho de investigação será a *Multi Layer Perceptron* (MLP) com o algoritmo de retropropagação do erro denominado *Back-propagation*, porposto por [David, Geoffrey e Ronald \(1986\)](#) e desenvolvida utilizando a

implementação *MultilayerPerceptron* proposta por [Hall et al. \(2009\)](#), disponível na biblioteca Weka (Documentação no Anexo C). Segundo [Han e Kamber \(2006\)](#), a principal característica que torna a MLP interessante para solução de problemas é a capacidade de aprender através de exemplos e de generalizar este aprendizado de forma que seja capaz de reproduzir um resultado a uma entrada diferente da apresentada. [Hall et al. \(2009\)](#), afirma que a rede MLP permite a solução de problemas não lineares, como é o caso dos sinais EEG.

3.1.3 Módulo de acionamento do dispositivo

A cadeira de rodas motorizada utilizada neste trabalho é a *Freedom Carbon*, ilustrada na Figura 3.7.

Figura 3.7: Cadeira de rodas motorizada.



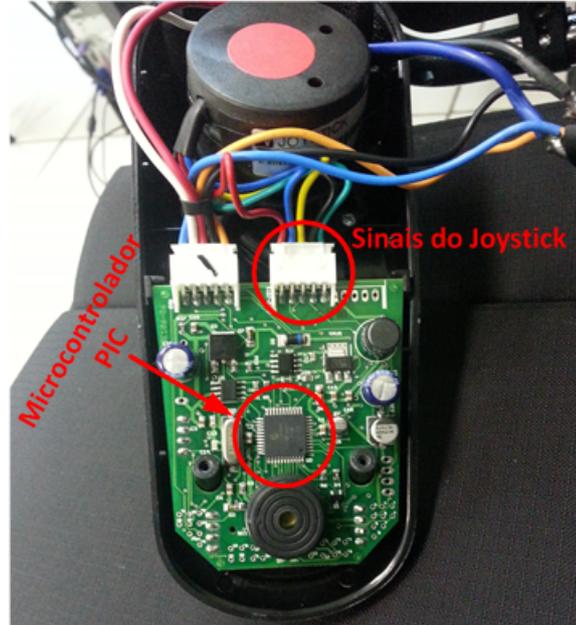
Fonte: [Freedom \(2012\)](#).

A primeira etapa do processo de adaptação da cadeira de rodas foi o desenvolvimento do módulo de acionamento do dispositivo. Por se tratar de um equipamento comercial, não foi possível compreender o módulo de potência para acionamento, uma vez que o fabricante não fornece esse tipo de informação. Sendo assim, para desenvolver o circuito que simula o *joystick* original da cadeira de rodas foi necessário entender quais os valores de tensão específicos, responsáveis por realizar os movimentos na cadeira de rodas através *joystick*.

O *joystick* da cadeira de rodas (*Freedom Carbon*) envia sinais para um circuito embarcado que, de acordo com o movimento executado na alavanca do *joystick*, dois sinais de

tensão são enviados ao circuito do microcontrolador que, por sua vez, interpreta e envia outros sinais ao sistema de motorização da cadeira de rodas para realizar os movimentos desejados. A Figura 3.8 ilustra o circuito do microcontrolador, destacando-se o *Programmable Interface Controller* (PIC) e barramento, de onde são enviados os sinais do *joystick* diretamente ao circuito.

Figura 3.8: Circuito do *joystick* visto de cima.

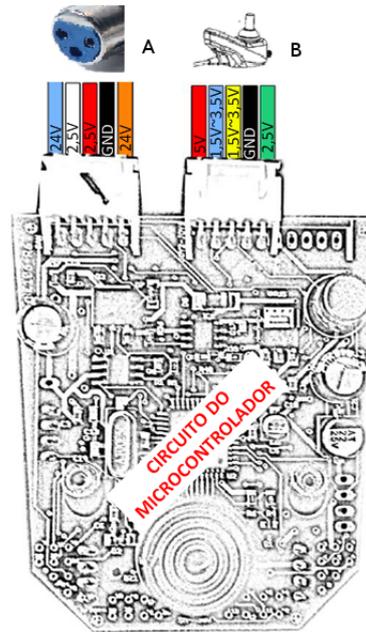


Fonte: Freedom (2012).

Sendo assim, desenvolvemos um circuito que realiza a mesma função do *joystick*, enviando tensões analógicas ao circuito do microcontrolador para controlar os movimentos da cadeira de rodas. Em resumo, o circuito simula em sua saída os mesmos sinais analógicos do *joystick* original da cadeira de rodas, consistindo em uma adaptação do circuito já existente.

Pode-se observar na Figura 3.9 um esquema com os fios de ligação no circuito do microcontrolador, destacando os barramentos provenientes do conector *canon* (A) e do *joystick* (B) e seus respectivos valores de tensão, medidos por um multímetro digital.

Figura 3.9: Esquema de ligação entre o *joystick* e o circuito do microcontrolador.



Fonte: Autor.

O fio azul proveniente do *joystick* (B) controla as ações de movimentos: frente ou ré, enquanto que o fio amarelo controla as ações de movimentos: esquerda ou direita. Para cada movimento, foram realizadas medidas nas tensões de cada fio com o auxílio de um multímetro digital. A Tabela 3.3 apresenta os respectivos valores de tensão dos fios do barramento controlado pelo *joystick*. Assim, entende-se de que forma esses valores alteram o movimento da cadeira de rodas.

Tabela 3.3: Tensão dos fios do barramento controlado pelo *joystick* e os movimentos.

	Parado	Direita	Esquerda	Frente	Ré
Fio azul	2,5V	2,5V	2,5V	3,5V	1,5V
Fio amarelo	2,5V	1,5V	3,5V	2,5V	2,5V

Fonte: Autor.

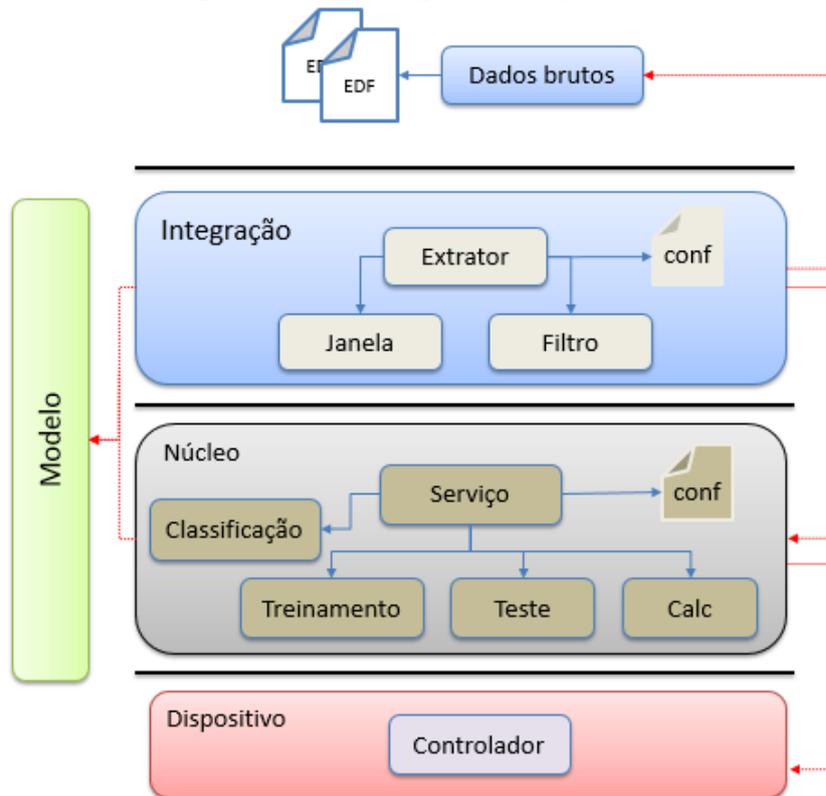
Nota-se que é possível controlar os movimentos da cadeira de rodas, variando os valores de tensão aplicados nos fios azul e amarelo localizados no barramento proveniente do *joystick* (Figura 3.9). Dessa forma, com a implementação de um circuito capaz de enviar sinais analógicos para esses pinos, pode-se movimentar a cadeira de rodas. O circuito que simula as ações do *joystick* original deve ser projetado respeitando-se os limites operacionais das tensões a serem enviadas ao circuito do microcontrolador da cadeira de rodas, evitando, assim, que ocorra o impedimento de movimento por conta de uma tensão enviada fora da

faixa de operação.

3.2 Projeto arquitetural

O projeto arquitetural, proposto nesse trabalho de investigação (Figura 3.10), tem como foco principal atender às necessidades do modelo computacional, descritas na Seção 3.1. O projeto fornece uma visão arquitetural abrangente da aplicação interface cérebro-computador (sistema), usando diversas visões de arquitetura para representar diferentes aspectos do sistema. Assim, propõe-se uma divisão em camadas (Integração, Núcleo e Dispositivo) com componentes (Interfaces) bem definidos, proporcionando alta coesão⁹ e baixo acoplamento¹⁰ em diversos níveis.

Figura 3.10: Visão geral da arquitetura.



Fonte: Autor.

O componente Modelo provê as funcionalidades diretamente ligadas ao contexto do domínio

⁹Coesão: Refere-se ao princípio da responsabilidade única, um componente deve ter apenas uma única responsabilidade e realizá-la de maneira satisfatória, ou seja, um componente não deve assumir responsabilidades que não são suas.

¹⁰Acoplamento: Significa o quanto um componente depende de outro para funcionar, ou seja, quanto menor for esta dependência entre ambos, dizemos que estes componentes estão fracamente acoplados.

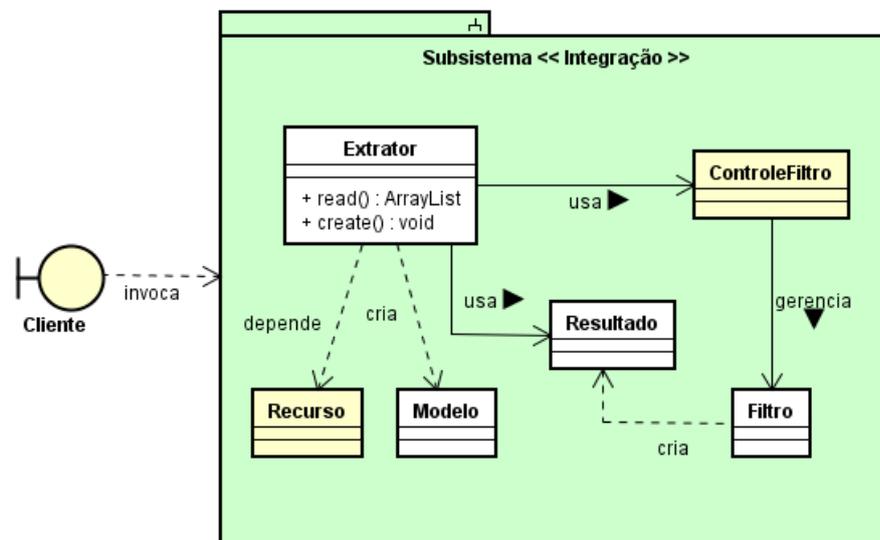
do sistema. O Modelo contém as abstrações que representam as entidades do sistema, como: vetor de características, média aritmética, energia, máximo valor da transformada de Fourier, mínimo valor da transformada de Fourier, frequência dominante, tensão e os atributos da CRI. Os dados originados do Modelo são utilizados pelas camadas: Integração e Núcleo.

3.2.1 Camada de Integração

A camada de Integração é responsável pela recepção dos dados brutos, no formato EDF, referentes aos sinais EEG. O componente **Extrator** realiza a extração dos dados brutos, transformando-os em elementos de modelo do sistema (**Modelo**), o **Extrator** utiliza outros dois componentes para completar sua tarefa, são eles: **Filtro**, esse componente tem por finalidade filtrar os sinais a uma faixa configurável (valores mínimo e máximo) e **Janela**, que comporta-se como uma janela móvel no eixo temporal, limitando o sinal a ser analisado. A Figura 3.11 apresenta as classes participantes do subsistema Integração.

A classe de fronteira **Cliente**, representa o processo que requer acesso ao sistema, ou seja, que deseja comandar a CRI através dos sinais EEG.

Figura 3.11: Classes participantes do subsistema Integração.



Fonte: Autor.

A classe **Extrator**, disponibiliza métodos genéricos para concentrar as operações comuns, com o objetivo de simplificar o esquema de acesso aos recursos. A classe **ControleFiltro**, gerencia o processamento do filtro, essa classe, administra o tempo que o filtro ficará ativo

(gerencia o tamanho da janela). Já a classe **Filtro**, filtra (elimina) tudo à exceção de um determinado intervalo de frequências (por exemplo: faixa de 0,5 a 42 Hz). A classe **Recurso**, é responsável pela leitura dos parâmetros de configurações externos à aplicação, o arquivo de configuração é denominado como: **conf.properties**¹¹. O mesmo deve ser salvo em local pré-determinado, para que ele possa ser lido pela aplicação. A Tabela 3.4 apresenta as chaves (identificadores) e os respectivos valores padrão dos parâmetros da aplicação, podendo serem alterados conforme os requisitos da aplicação.

Tabela 3.4: Conteúdo do arquivo de propriedades: Integração.

Chave	Valor
prop.filtro.minimo	0.5
prop.filtro.maximo	42.0
prop.taxa.amostragem	160.0

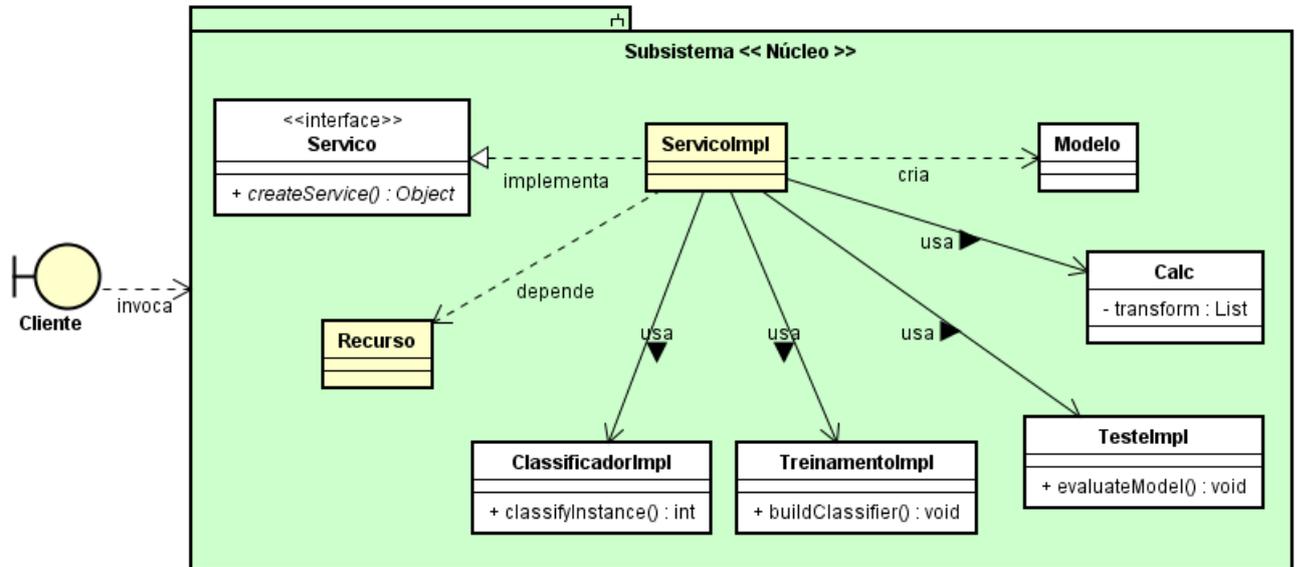
A chave **prop.filtro.minimo** representa o valor da frequência mínima do filtro digital passa-faixa, a chave **prop.filtro.maximo** representa o valor da frequência máxima do filtro digital passa-faixa. Já a chave **prop.taxa.amostragem**, representa o valor da taxa de amostragem para leitura dos sinais EEG. A classe **Resultado**, representa o resultado dos dados filtrados, que será utilizado para criação da classe **Modelo**, utilizada para a transferência de dados à camada Núcleo.

3.2.2 Camada Núcleo

A camada Núcleo está publicamente exposta através da interface **Servico**, a classe de fronteira **Cliente**, representa o subsistema de Integração (Figura 3.11). O **ServicoImpl** é a classe que organiza um conjunto de métodos relacionados ao negócio. Essa classe é responsável por acessar recursos (classificação, treinamento e teste), atualizar dados e processar requisições (solicitações do **Cliente**) com uma interação limitada com o **Cliente**. O **ClassificadorImpl** é utilizado para identificar a classe à qual pertence uma determinada observação (conjunto de dados), a partir de suas características (atributos). A classe **TrainamentoImpl** é utilizada na fase de aprendizado, como resultado, obtem-se a construção do classificador (modelo de classificação). Após o classificador ser construído, inicia-se a etapa de teste, que visa avaliar seu desempenho através do emprego de um conjunto de dados de testes, a classe **TesteImpl** realiza essa tarefa. A classe **Clac** encapsula os métodos matemáticos do sistema e a classe **Modelo** representa os dados (entidades) do sistema.

¹¹O arquivo properties na plataforma Java, representa um conjunto persistente de propriedades. As propriedades podem ser salvas em um fluxo ou carregadas de um fluxo. Cada chave e seu valor correspondente na lista de propriedades é uma String.

Figura 3.12: Classes participantes do subsistema Núcleo.



Fonte: Autor.

Na camada Núcleo, o componente **Servico** provê as funcionalidades referentes à lógica do sistema, gerenciando os componentes de classificação (**ClassificadorImpl**), treinamento (**TreinamentoImpl**), teste (**TesteImpl**), fórmulas matemáticas (**Calc**) e a classe **Recurso**. Os componentes dessa camada tem funções específicas, descritas a seguir:

- **Servico**: esse componente tem por finalidade simplificar a complexidade de acesso (utilização) aos recursos da camada Núcleo. O **Servico** fornece uma interface unificada para o conjunto de componentes (**ClassificadorImpl**, **TreinamentoImpl**, **TesteImpl** e **Calc**). O objetivo do **Servico** é desacoplar a camada de Integração da camada Núcleo (acesso direto aos componentes complexos), dessa forma, é possível reduzir a dependência entre as camadas do sistema, permitindo realizar manutenções nos componentes dessa camada isoladamente, sem que a camada de Integração precise sofrer alterações. Basicamente, o componente **Servico** permite a camada de Integração acesso aos componentes da camada Núcleo de forma isolada, ou seja, sem que a camada de Integração conheça os detalhes de implementação desses componentes.
- **ClassificadorImpl**: o classificador (implementação da RNA), tem como tarefa realizar a associação automática de um objeto (instância) a uma determinada classe (E, D, F e P), pertencente a um conjunto pré-definido de classes (esquerda, direita, frente e parar).
- **TreinamentoImpl**: esse componente processa uma base de treinamento, composta

por instâncias já classificadas. Como resultado, este algoritmo constrói um modelo capaz de classificar novos objetos (instâncias cuja classe é desconhecida). O modelo produzido nessa fase, será utilizado pelo componente: **ClassificadorImpl** que implementa o algoritmo de classificação.

- **TesteImpl**: esse componente, após a fase de treinamento, testa o modelo para que seu desempenho seja avaliado. Isso é feito com o uso de uma base de testes, que também possui instâncias classificadas.
- **Calc**: é responsável pela implementação das funções matemáticas utilizadas pelo sistema.
- **Recurso**: esse componente é responsável pela leitura dos parâmetros de configurações externos à aplicação.

A classe **Recurso**, utilizada na camada Núcleo, é responsável pela leitura dos parâmetros: número de neurônios presentes na camada oculta da rede¹², taxa de aprendizado¹³, o termo de *momentum*¹⁴ e o número de épocas¹⁵. A Tabela 3.5 apresenta as chaves (identificadores) e os respectivos valores padrão dos parâmetros da aplicação, podendo serem alterados conforme os requisitos da aplicação em uso.

Tabela 3.5: Conteúdo do arquivo de propriedades: Núcleo.

Chave	Valor
prop.taxa.aprendizado	0.01;0.1;0.5;0.9
prop.termo.momentum	0.0;0.1;0.5;0.9
prop.numero.neuronio.oculto	11
prop.numero.epocas	1000

A chave **prop.taxa.aprendizado** representa o valor da taxa de aprendizagem, podendo ser um único valor ou um conjunto de valores, separados por “;”. A chave **prop.termo.momentum** representa o valor da constante de *momentum*. A chave **prop.numero.neuronio.oculto**, representa o valor do número de neurônios da camada oculta da RNA. Já a chave **prop.numero.epocas**, representa o valor do número de ciclos de treinamento da RNA.

Essa parametrização permite determinar através de simulações computacionais: o número

¹²O número de neurônios da camada oculta da rede é analisado para selecionar o melhor método a ser utilizado pelo sistema.

¹³A taxa de aprendizado é uma constante de proporcionalidade no intervalo 0 - 1. Uma taxa de aprendizado muito baixa torna o aprendizado da rede muito lento, ao passo que uma taxa de aprendizado muito alta provoca oscilações no treinamento e impede a convergência do processo de aprendizado.

¹⁴O termo *momentum* tem por objetivo aumentar a velocidade de treinamento da rede neural e seu valor varia de 0 - 1. É uma constante que determina o efeito das mudanças passadas dos pesos na direção atual do movimento no espaço de pesos.

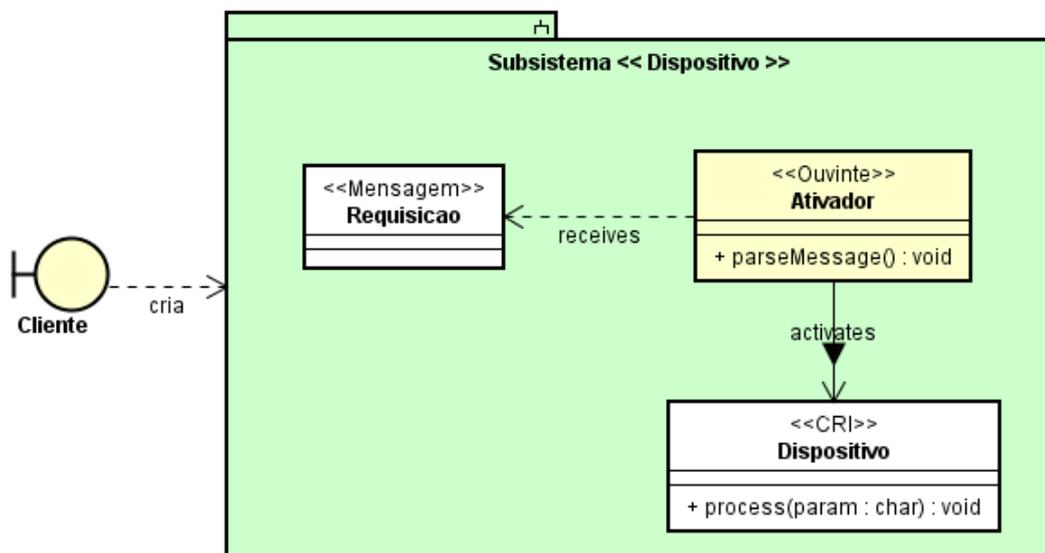
¹⁵Define o número de ciclos de treinamento, ou seja, o número de vezes em que o conjunto de treinamento é apresentado à rede. Um número excessivo de ciclos pode levar a rede à perda da capacidade de generalização.

ótimo de neurônios presentes na camada oculta da RNA, o parâmetro da taxa de aprendizagem e a constante de *momentum*. Dessa forma, utilizando os valores configurados no arquivo **conf.properties**, são simuladas combinações do parâmetro da taxa de aprendizagem e da constante de *momentum*. O processo de treinamento é executado de 1 até o valor definido para o número de épocas.

3.2.3 Camada de Dispositivo

A camada de Dispositivo, encapsula a lógica relacionada a integração do sistema com a CRI. É acoplada a camada Núcleo sempre que esta camada precisar de serviços que residem na camada de Dispositivo. Assim, essa camada fica responsável em receber requisições do cliente (Núcleo). Ao receber uma mensagem, o **Ativador** localiza e chama os métodos necessários nos componentes para atender à requisição, de forma assíncrona. A Figura 3.13 apresenta as classes participantes do subsistema Dispositivo.

Figura 3.13: Classes participantes do subsistema Dispositivo.



Fonte: Autor.

O subsistema Dispositivo, propõe como solução chamar serviços da CRI de modo assíncrono, o uso de processamento assíncrono nesse caso coloca em uma fila as requisições, que são processadas pelo dispositivo de controle da CRI. Dessa forma, em situações em que o sistema venha a receber um número grande de requisições (comandos de movimentos), o processamento em paralelo não irá comprometer o desempenho do sistema. Em resumo, a classe **Ativador** comporta-se com um ouvinte, essa classe analisa a requisição (classe **Requisicao**) e chama o componente (**Dispositivo**) para processar a solicitação.

3.3 Implementação

Os componentes relacionados as camadas: Integração e Núcleo, foram implementados utilizando a linguagem de programação Java, justifica-se a escolha dessa linguagem por ser multiplataforma, ou seja, é independente de sistema operacional. Assim, não será preciso reescrever os códigos do sistema quando for instalá-los em outro sistema operacional, pois a conversão é feita pela Máquina Virtual do Java. As classes relacionadas a camada Dispositivo, foram implementadas utilizando a linguagem de programação C++, representando a interface para acionamento da CRI. Nesse caso, a linguagem C++ foi escolhida em razão da utilização da plataforma *Arduino*.

A troca de mensagens entre as camadas Núcleo e Dispositivo é baseada na comunicação serial (UART)¹⁶. Todas as camadas do sistema, apresentadas na Seção 3.2, são compostas por um ou mais pacotes. Deste modo, a estrutura de pacotes da aplicação segue o formato presente no diagrama da Figura 3.14.

Os pacotes do sistema seguem a hierarquia do pacote **br.com.asco**, pacote de primeiro nível. Esses pacotes, agrupam classes de acordo suas responsabilidades, a saber:

- **Integração:** o pacote *br.com.asco.integracao* representa o local, onde ficam as classes da camada de integração, implementadas de forma desacoplada (independente das classes de outras camadas) visto que é importante fornecer a possibilidade de troca da implementação das classes desse pacote sem a necessidade de adaptações no restante do sistema. Os métodos de acesso as regras do sistema são fornecidos através de uma interface *Servico* (camada Núcleo). A organização geral dos pacotes são descritas a seguir:
 - *br.com.asco.integracao.controle*: neste local, ficam as classes de controle da camada de integração.
 - *br.com.asco.integracao.controle.filtro*: neste local, ficam as classe responsáveis pela aplicação do filtro.
- **Núcleo:** os elementos do pacote *br.com.asco.negocio*, são classes que implementam as regras de negócio do sistema. Sendo importante ressaltar que este pacote é subdividido em outros pacotes, descritos a seguir:
 - *br.com.asco.negocio.servico*: neste local, fica a interface para acesso as regras de negócio do sistema e acesso a camada de dispositivo.
 - *br.com.asco.negocio.servico.nucleo*: neste local, ficam os serviços (treino, classificação e teste) que implementam as regras de negócio do sistema.

¹⁶UART (*Universal Asynchronous Receiver/Transmitter*): Transmissor/Receptor Universal Assíncrono. É um padrão para comunicação de dados de forma serial. Na forma assíncrona, dois fios são usados para transmitir dados, um em cada direção, em regime *full-duplex*, ou seja, totalmente bi-direcional.

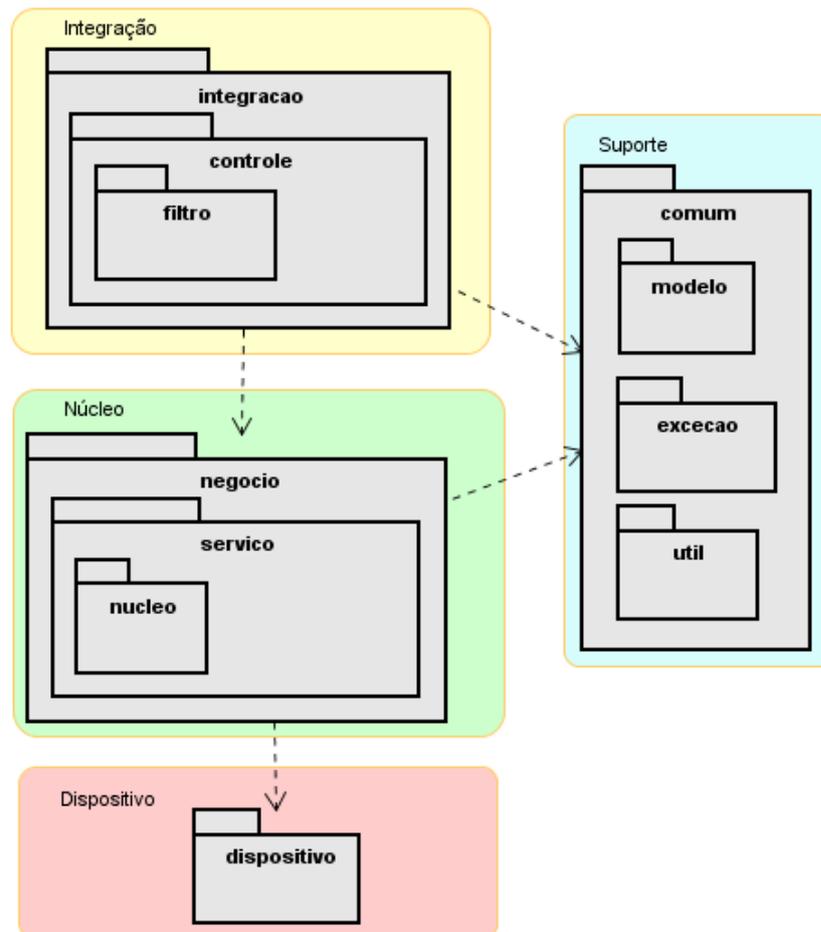
- **Dispositivo**

- `br.com.asco.dispositivo`: neste local, são organizadas as classes que executam as tarefas de acionamento da CRI.

- **Suporte**: o pacote `br.com.asco.comum` representa o local, onde ficam as classes que, em tempo de execução, transitam pelas camadas do sistema: Integração e Núcleo. A organização geral dos pacotes são descritas a seguir:

- `br.com.asco.comum.modelo`: neste local, ficam as classes responsáveis pela transferência de dados entre as camadas do sistema: Integração e Núcleo.
- `br.com.asco.comum.excecao`: neste local, ficam as classes exceção do sistema.
- `br.com.asco.comum.util`: neste local, ficam as classes utilitárias do sistema.

Figura 3.14: Visão de pacotes e subsistemas da aplicação.

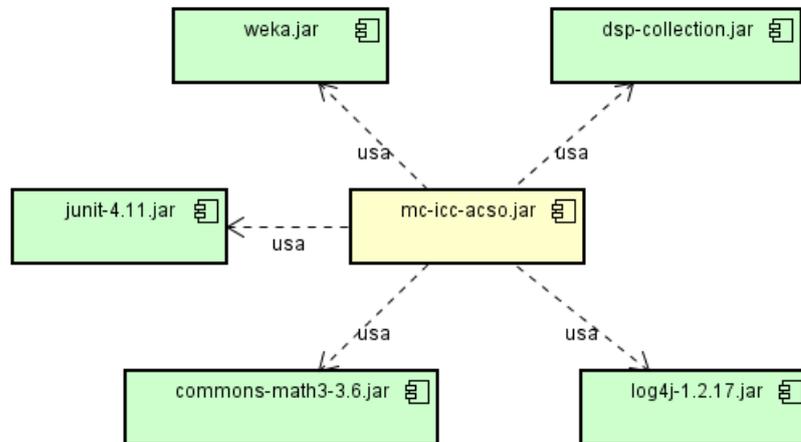


Fonte: Autor.

A Figura 3.15 ilustra os principais componentes de *software* que o *framework mc-icc-acco* tem relação de dependência, portanto, so necessários para utilização de sistemas

baseados na proposta desse trabalho de investigação.

Figura 3.15: Visão de implementação da aplicação.



Fonte: Autor.

Cada componente, têm responsabilidade específica, a saber:

- **weka.jar**: apresenta uma coleção de algoritmos de aprendizado de máquina. Os algoritmos podem ser aplicados diretamente a um conjunto de dados ou chamados de seu próprio código Java.
- **dsp-collection.jar**: biblioteca *open-source* que contém classes para processamento de sinal digital.
- **junit-4.11.jar**: é um *framework open-source*, com suporte à criação de testes automatizados.
- **commons-math3-3.6.jar**: é uma biblioteca de componentes matemáticos e estatísticos, que abordam os problemas mais comuns não disponíveis na linguagem de programação Java.
- **log4j-1.2.17.jar**: fornece uma API para que o desenvolvedor de *software* possa fazer *log* de dados na sua aplicação.

3.3.1 Interface de acionamento

Para o desenvolvimento do circuito de acionamento da CRI, foi utilizado o *Arduino UNO*. Justifica-se a escolha do *Arduino*, por se tratar de uma plataforma eletrônica de código

aberto baseada em *hardware* e *software* de fácil utilização. Dois pinos *Pulse Width Modulation* (PWM): 5 e 6 foram utilizados, para enviar os sinais de tensão ao circuito do microcontrolador da cadeira de rodas motorizada.

Segundo [Hirzel \(2016\)](#), para calcular o valor médio da tensão de saída de um sinal PWM, pode-se utilizar a seguinte Equação 3.10:

$$V_{out} = (duty\ cycle/100) * V_{cc} \quad (3.10)$$

Onde: V_{out} é a tensão de saída em *Volts*, *duty cycle* é o valor do ciclo ativo do PWM em % e V_{cc} é a tensão de alimentação em *Volts*.

Quando o *duty cycle* está em 0% o valor médio da saída encontra-se em 0 *Volts* e conseqüentemente para um *duty cycle* de 100% a saída assume o valor máximo de 5 *Volts*. Para um *duty cycle* de 50% a saída assumirá 50% o valor da tensão de 2,5 *Volts* e assim sucessivamente para cada variação no *duty cycle* ([HIRZEL, 2016](#)).

Dessa forma, a programação em C++ implementada no arduíno, leva esse cálculo em consideração para fornecer a tensão desejada no circuito do microcontrolador da cadeira de rodas. Duas saídas PWM (pinos 5 e 6), foram utilizados, sendo que o pino 5 envia a tensão ao fio azul para comandar os movimentos: frente e ré. Enquanto o pino 6 aplica as tensões no fio amarelo, para controlar os movimentos: direita e esquerda.

Para movimentar a cadeira de rodas, foram definidos cinco caracteres, cada um representando uma ação: parar, frente, ré, direita e esquerda. Escolheram-se os caracteres de “0” a “4” que representam, na tabela ASCII os decimais de 48 a 52. A Tabela 3.6, apresenta os caracteres selecionados, seu valor equivalente na tabela ASCII e o comando representativo de cada um deles dentro do circuito de controle.

Tabela 3.6: Caractere x movimentação.

Caractere	ASCII	Comando
‘0’	48	parar
‘1’	49	frente
‘2’	50	ré
‘3’	51	direita
‘4’	52	esquerda

Fonte: Autor.

De acordo com o caractere recepcionado pelo arduíno, serão aplicados dois valores nos

pinos 5 e 6 (Tabela 3.7), que deverão ser aplicados no circuito da cadeira de rodas.

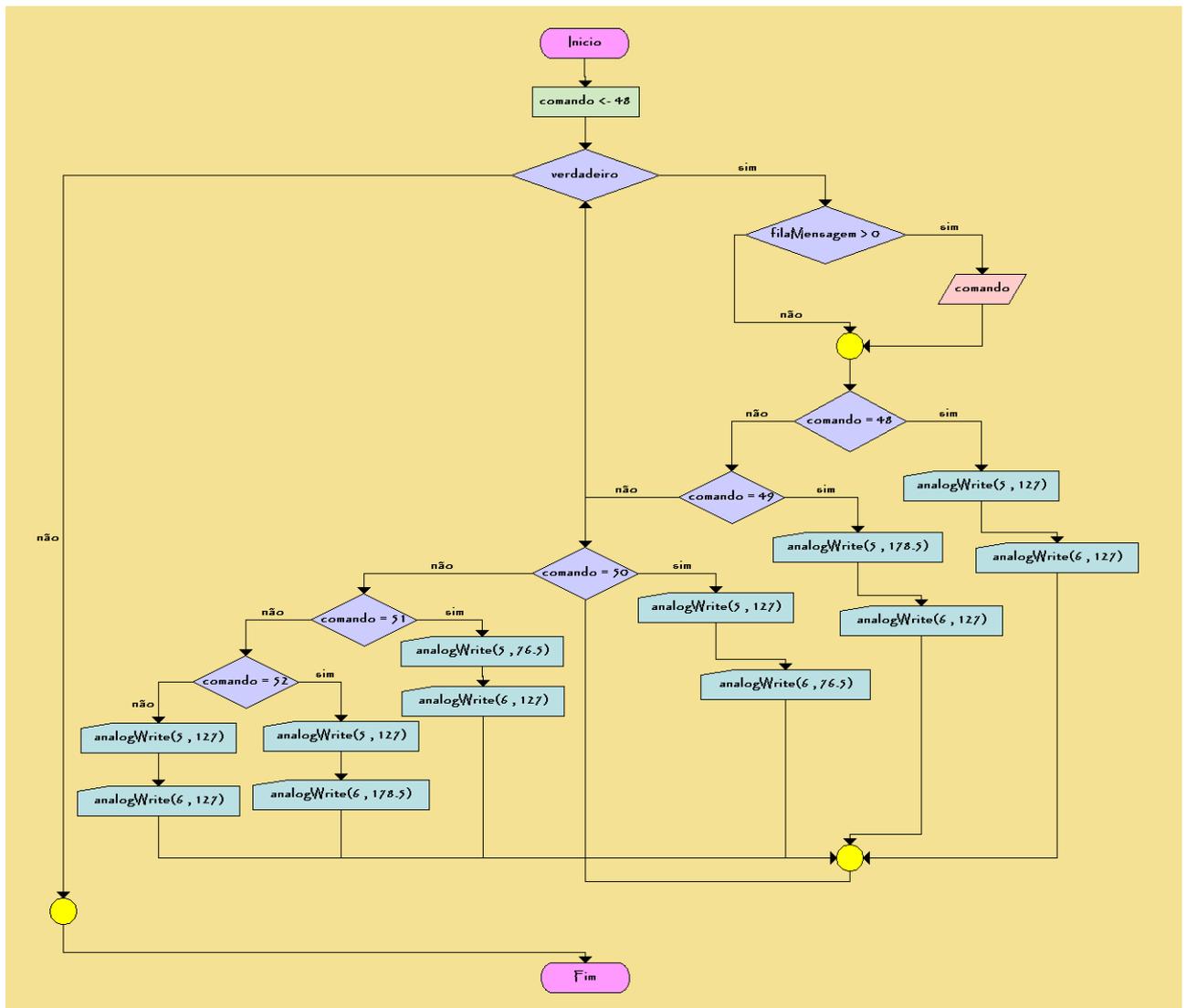
Tabela 3.7: Valores aplicados aos pinos PWM do arduino e os movimentos.

	Parado	Direita	Esquerda	Frente	Ré
Pino 5	127	127	127	178.5	76.5
Pino 6	127	76.5	178.5	127	127

Fonte: Autor.

A Figura 3.16 ilustra o fluxograma do circuito de acionamento, transcrito para a linguagem de programação C++ e embarcado no arduino UNO.

Figura 3.16: Fluxograma do circuito de acionamento.

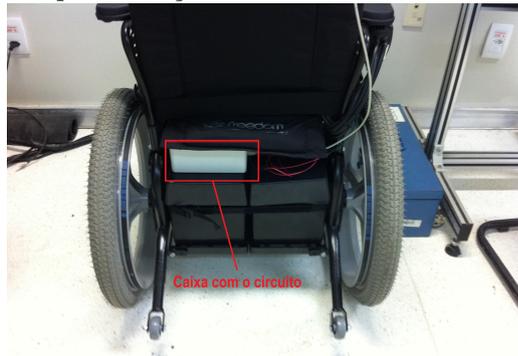


Fonte: Autor.

Observa-se, a partir da Figura 3.16, que inicialmente a variável comando é definida com o valor igual a 48, esse comportamento determina inicialmente que o estado de operação da CRI será de parar, conforme dados apresentados na Tabela 3.6. Em seguida, um laço de repetição contínua é estabelecido, na sequência é verificado se existe pelo menos uma mensagem (solicitação de acionamento da CRI) na fila para ser consumida (`filaMensagem > 0`). Em caso positivo, o sistema aplica os valores de tensão nos pinos 5 e 6, conforme valores estabelecidos na Tabela 3.7, através do comando *analogWrite*. Por exemplo, se o valor lido para a variável comando, for igual a 49 (acionar CRI para frente), os comandos executados serão: *analogWrite(5, 178.5)* e *analogWrite(6, 127)*, respectivamente 3.5V no pino 5 e 2.5V no pino 6. Caso o valor lido para variável comando seja diferente dos valores: 48 (parar), 49 (frente), 50 (ré), 51 (direita) e 52 (esquerda), os comandos executados serão: *analogWrite(5, 127)* e *analogWrite(6, 127)*, respectivamente 2.5V no pino 5 e 2.5V no pino 6, ou seja, o comando será de parar a CRI.

A Figura 3.17 ilustra como foi colocada a caixa com o circuito de controle junto à cadeira de rodas e a aparência final da cadeira com a plataforma implementada.

Figura 3.17: Implementação da caixa com o circuito de controle.



Fonte: Autor.

O circuito foi colocado em uma caixa de plástico, perfurada para que os fios de comunicação e sinais de controle fossem conectados à cadeira de rodas e o computador. A caixa com o circuito foi colocada na parte traseira da cadeira de rodas, para que não ficasse visível ao usuário, garantindo que a estética original da cadeira não fosse modificada.

3.4 Considerações finais

O projeto da interface cérebro-computador, desenvolvido neste trabalho de investigação, utilizou o sinal EEG para realizar a medição das atividades cerebrais. A aquisição

dos sinais EEG foi realizada através do banco de dados *egmmidb - EEG Motor Movement/Imagery Dataset*. O passo seguinte à aquisição dos sinais EEG foi o pré-processamento. Para realizar esse procedimento foi desenvolvido um filtro digital passa-faixa, cujas frequências variam em uma faixa de 0,5 a 42 Hz. Assim, o filtro possibilita a redução da influência de frequências fora das regiões de interesse.

O conjunto de dados adquiridos por meio dos sinais EEG alimenta o sistema com uma grande quantidade de informação. O mecanismo de extração de características implementado nesse projeto, utiliza a FFT no sinal EEG. Em seguida, associa um conjunto de informação que caracteriza os fenômenos observados a uma estrutura denominada vetor de características. As etapas de identificação e extração de características dos sinais são de extrema importância para o bom desempenho dos classificadores.

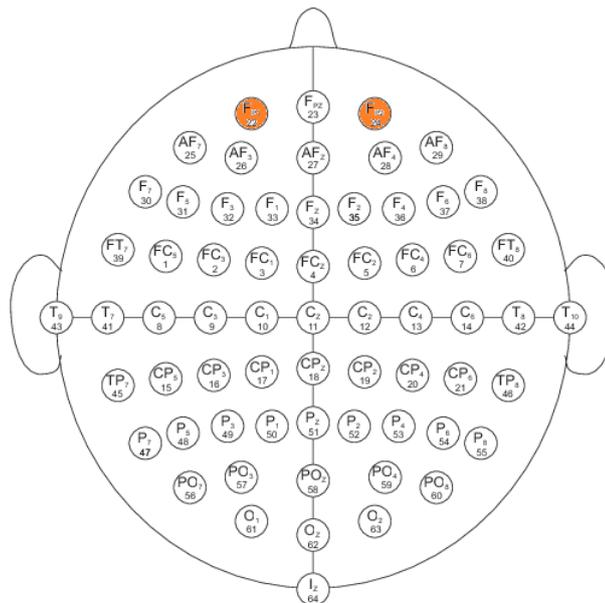
A etapa de classificação dos sinais EEG, refere-se à tradução das características extraídas em comandos. O mecanismo de classificação implementado no projeto utiliza como solução uma RNA, *perceptron* multicamadas. O objetivo da classificação é associar automaticamente uma instância do vetor de características a uma determinada classe. Esta classe identifica a modalidade da tarefa cognitiva imaginada pelo usuário do sistema. Uma vez que o estado mental é determinado, associa-se a este a execução de um movimento da CRI, através da interface de acionamento.

Avaliação e resultados experimentais

Neste capítulo, será apresentado o experimento computacional realizado para avaliar o comportamento da interface cérebro-computador, relacionado ao desempenho da RNA, interface de acionamento e integração dos módulos e componentes do sistema. A arquitetura do sistema, utiliza a técnica de Validação Cruzada (*10-fold cross-validation*) para a validação da RNA do tipo *Multi Layer Perceptron* (MLP), com o algoritmo de retropropagação do erro (*backpropagation*).

Os sinais dos registros EEG utilizados para validar a interface cérebro-computador foram adquiridos através do banco de dados *egmmidb - EEG Motor Movement/Imagery Dataset*, capturado utilizando o sistema BCI2000 (SCHALK et al., 2004), disponível através do PhysioBank (GOLDBERGER et al., 2000). Este banco de dados é composto por mais de 1500 registros de sinais EEG, obtidos a partir de 109 voluntários. Os registros de 106 voluntários foram utilizados nos testes. Dados de 3 voluntários (88, 92 e 100) apresentaram problemas na etapa de aquisição; por esse motivo, tais registros não foram considerados. Neste trabalho de investigação, foram utilizados apenas os eletrodos posicionados na região do córtex pré-frontal (Fp1 e Fp2), ilustrados na Figura 4.1.

Figura 4.1: Localização dos eletrodos (Fp1 e Fp2) utilizados na pesquisa.



Modificado de: Goldberger et al. (2000)

O computador utilizado foi um DELL® modelo Latitude E6430 com processador Intel R CORE TM i7-3540M - 3.00GHz 2 núcleos e 4 *threads*, placa de vídeo NVidia R NVS TM 5200M - 1GB, 64bit - memória RAM com capacidade para 8GB e sistema operacional Linux Ubuntu 14.04.2 LTS.

4.1 Desempenho da RNA

Uma vez construída a RNA, sua precisão e capacidade de generalização foram testadas utilizando os dados processados dos sinais EEG que compõe o conjunto de amostras para treinamento e testes. A Tabela 4.1 lista os parâmetros variáveis do sistema, definidos no arquivo **conf.properties**.

Tabela 4.1: Conteúdo do arquivo de propriedades do sistema.

Chave	Valor
prop.filtro.minimo	0.5
prop.filtro.maximo	42.0
prop.taxa.amostragem	160.0
prop.taxa.aprendizado	0.01;0.1;0.5;0.9
prop.termo.momentum	0.0;0.1;0.5;0.9
prop.numero.neuronio.oculto	11
prop.numero.epocas	1000

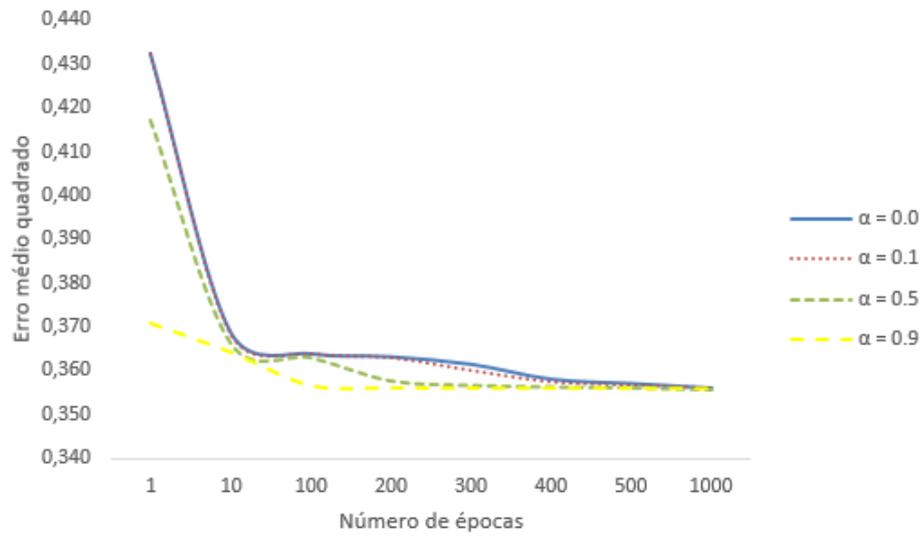
Fonte: Autor.

A coleta dos registros foi realizada com a utilização de um filtro digital passa-faixa, cujas frequências variam em uma faixa de 0.5 a 42.0 Hz e frequência de amostragem de 160.0 Hz. A RNA foi configura com 5 neurônios na camada de entrada, 11 neurônios na camada oculta e 4 neurônios na camada de saída. Foram simuladas combinações do parâmetro da taxa de aprendizagem (0.01, 0.1, 0.5, 0.9) e da constante de *momentum* (0.0, 0.1, 0.5, 0.9) para observar o efeito da convergência da RNA. Cada combinação foi treinada até 1000 épocas. Para avaliação da RNA, foram selecionadas 28.614 instâncias, dos dados originados de cada um dos eletrodos Fp1 e Fp2.

4.1.1 Dados originados do eletrodo Fp1

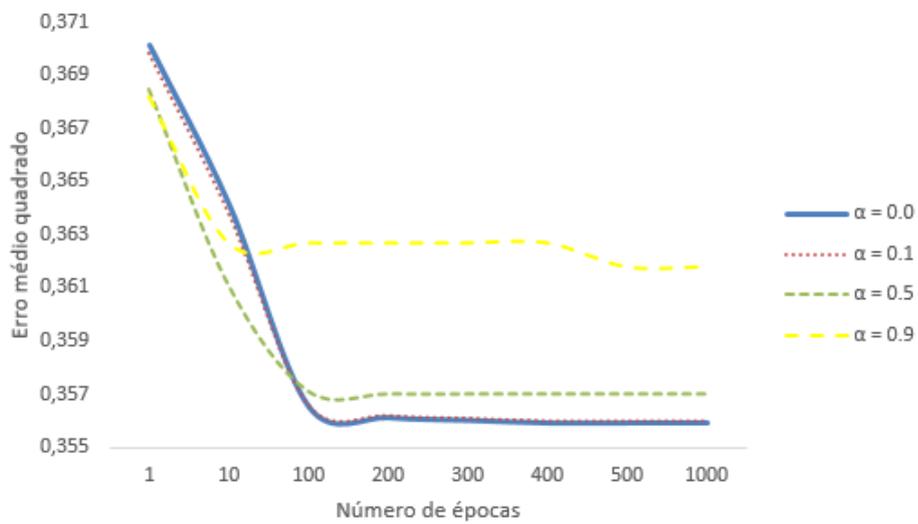
As curvas de aprendizagem médias referentes ao dados originados do eletrodos Fp1, são ilustradas nas Figuras 4.2, 4.3, 4.4 e 4.5. Agrupadas pela taxa de aprendizagem (η) em função das constantes de *momentum* (α).

Figura 4.2: Curvas de aprendizagem médias para $\eta = 0.01$ (eletrodo Fp1).

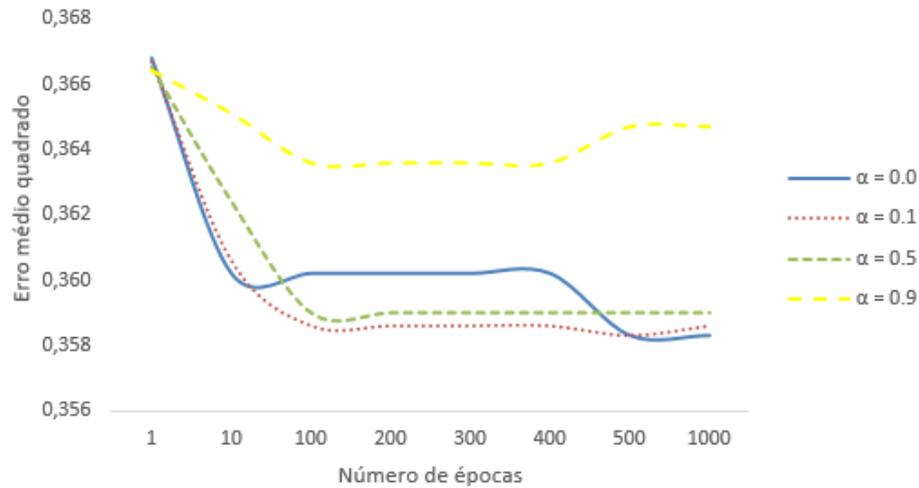


Fonte: Autor.

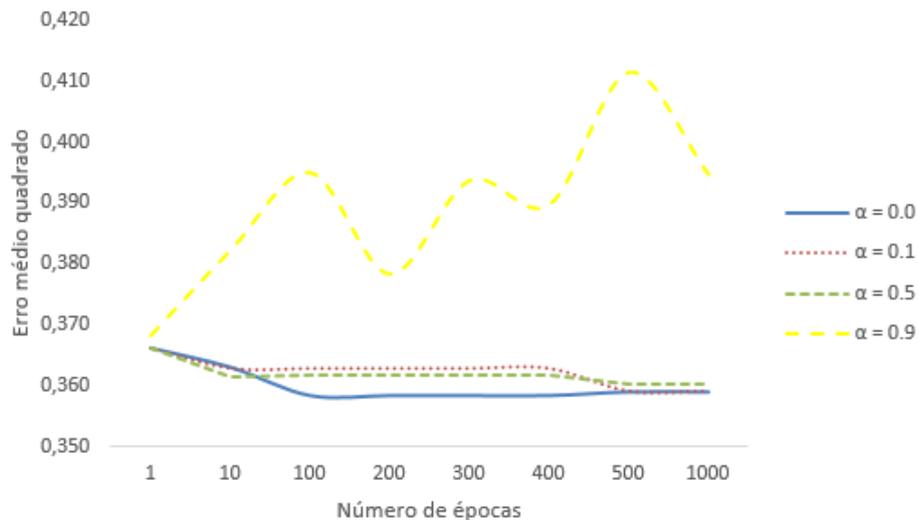
Figura 4.3: Curvas de aprendizagem médias para $\eta = 0.1$ (eletrodo Fp1).



Fonte: Autor.

Figura 4.4: Curvas de aprendizagem médias para $\eta = 0.5$ (eletrodo Fp1).

Fonte: Autor.

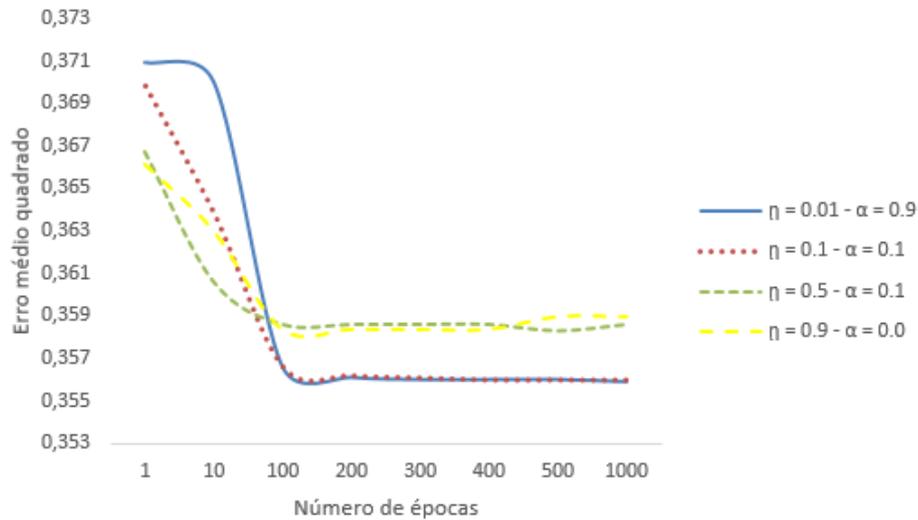
Figura 4.5: Curvas de aprendizagem médias para $\eta = 0.9$ (eletrodo Fp1).

Fonte: Autor.

Observa-se, a partir dos experimentos, ilustrados nas Figuras 4.2, 4.3, 4.4 e 4.5, que em geral, para um valor pequeno (0.01), atribuído à taxa de aprendizagem (η), resulta em uma convergência mais lenta. O uso das taxas de aprendizagem $\eta = 0.5$ e $\eta = 0.9$, sendo $\alpha = 0.9$, observa-se oscilações no erro médio quadrado durante o processo de aprendizagem, ocasionando um valor mais alto do erro médio quadrado na convergência da RNA, sendo ambos efeitos indesejáveis.

A Figura 4.6 apresenta as melhores curvas de aprendizagem para cada grupo das curvas ilustradas nas Figuras 4.2, 4.3, 4.4 e 4.5, para determinar a melhor curva de aprendizagem global.

Figura 4.6: Melhores curvas de aprendizagem selecionadas. (eletrodo Fp1).



Fonte: Autor.

Observa-se, a partir da Figura 4.6, que o melhor parâmetro da taxa de aprendizado foi $\eta = 0.1$ e a constante de *momentum* foi $\alpha = 0.1$. Nesse contexto, o fato de o erro médio quadrado da curva não variar muito, sugere que os parâmetros são os mais adequados para esse problema. A Tabela 4.2 apresenta um resumo dos parâmetros otimizados para a RNA.

Tabela 4.2: Configuração da RNA otimizada (eletrodo Fp1).

Parâmetro	Valor
Número de neurônios da camada oculta	11
Taxa de aprendizagem	0.1
Constante de <i>momentum</i>	0.1

Fonte: Autor.

Nesse contexto, a RNA final foi avaliada para determinar a sua precisão e capacidade de generalização. A Tabela 4.3 lista os parâmetros, configurados para esse experimento, definidos no arquivo **conf.properties**. O mesmo conjunto de dados com 28.614 instâncias usados anteriormente foram utilizados novamente.

Tabela 4.3: Configuração ótima do arquivo de propriedades (eletrodo Fp1).

Chave	Valor
prop.taxa.aprendizado	0.1
prop.termo.momentum	0.1
prop.numero.neuronio.oculto	11
prop.numero.epocas	1000

Fonte: Autor.

Findado o experimento, a RNA demonstrou uma precisão média de 50,40% e um percentual de instâncias classificadas corretamente igual a 50,77%. O número de instâncias classificadas corretamente foi de 14527, contra 14087 instâncias classificadas incorretamente. Ao analisarmos a matriz de confusão (Tabela 4.4), observa-se, que a classe E (imaginação de movimento do punho esquerdo), obteve 3571 instâncias classificadas corretamente, contra 3647 instâncias classificadas incorretamente. A classe D (imaginação de movimento do punho direito), obteve 1264 instâncias classificadas corretamente, contra 5822 instâncias classificadas incorretamente. A classe F (imaginação de movimento de ambos os punhos), obteve 7149 instâncias classificadas corretamente, contra 15 instâncias classificadas incorretamente. Já a classe P (imaginação de movimento de ambos os pés), obteve 2543 instâncias classificadas corretamente, contra 4603 instâncias classificadas incorretamente.

Tabela 4.4: Matriz de confusão da classificação dados (eletrodo Fp1).

Dados da amostragem				
Classes	E	D	F	P
E	3571	1281	20	2346
D	3448	1264	30	2344
F	3	2	7149	10
P	3235	1342	26	2543

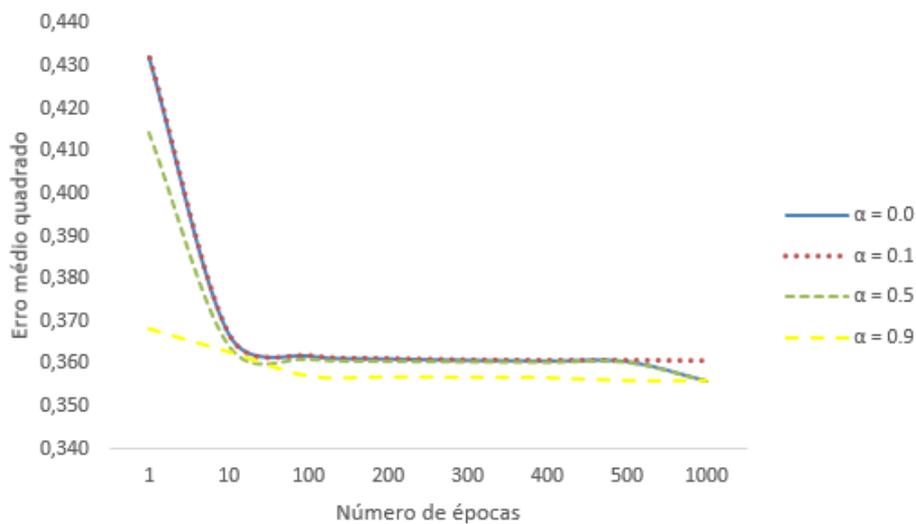
Fonte: Autor.

O método apresentou uma precisão de 98,9% para a classe F (Tabela 4.4), ou seja, a RNA demonstrou boa capacidade de classificação correta para o estado mental relacionado a imaginação de movimento de ambos os punhos, seguida das classes P (35,1%), E (34,8%) e D (32,5%). Conseqüentemente, a classe D apresentou alta taxa de confusão, onde 3448 instâncias foram consideradas como E, 2344 instâncias foram consideradas como P e 30 instâncias foram consideradas como F.

4.1.2 Dados originados do eletrodo Fp2

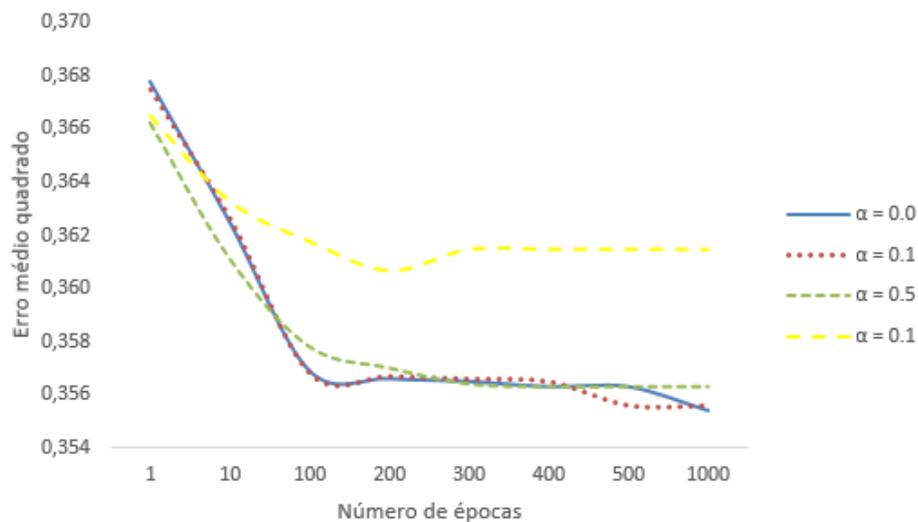
Repetimos todo o experimento computacional para avaliar a precisão e capacidade de generalização da RNA a partir dos dados originados do eletrodos Fp2. As curvas de aprendizagem médias são ilustradas nas Figuras 4.7, 4.8, 4.9 e 4.10. Agrupadas pela taxa de aprendizagem (η) em função das constantes de *momentum* (α).

Figura 4.7: Curvas de aprendizagem médias para $\eta = 0.01$ (eletrodo Fp2).

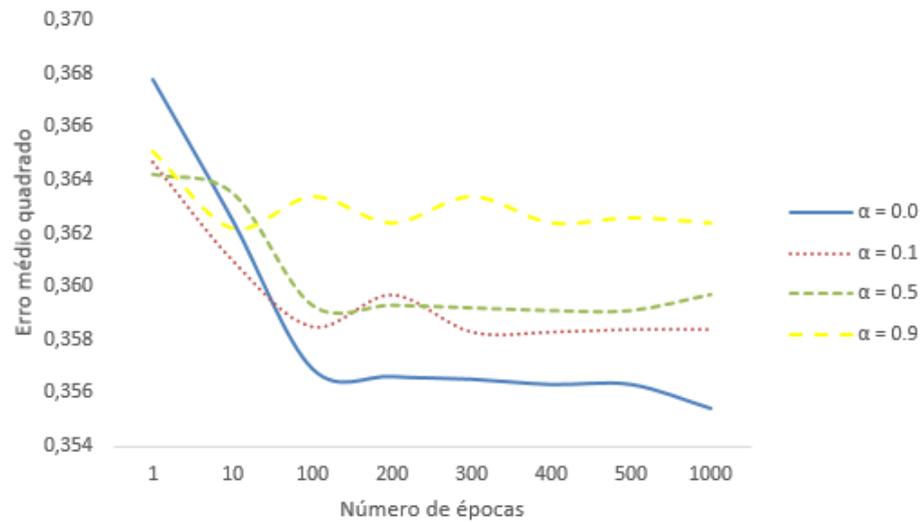


Fonte: Autor.

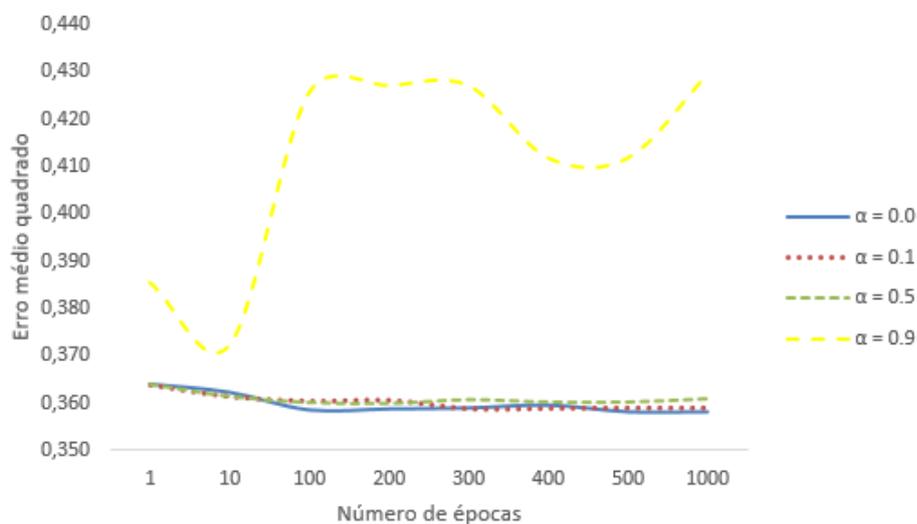
Figura 4.8: Curvas de aprendizagem médias para $\eta = 0.1$ (eletrodo Fp2).



Fonte: Autor.

Figura 4.9: Curvas de aprendizagem médias para $\eta = 0.5$ (eletrodo Fp2).

Fonte: Autor.

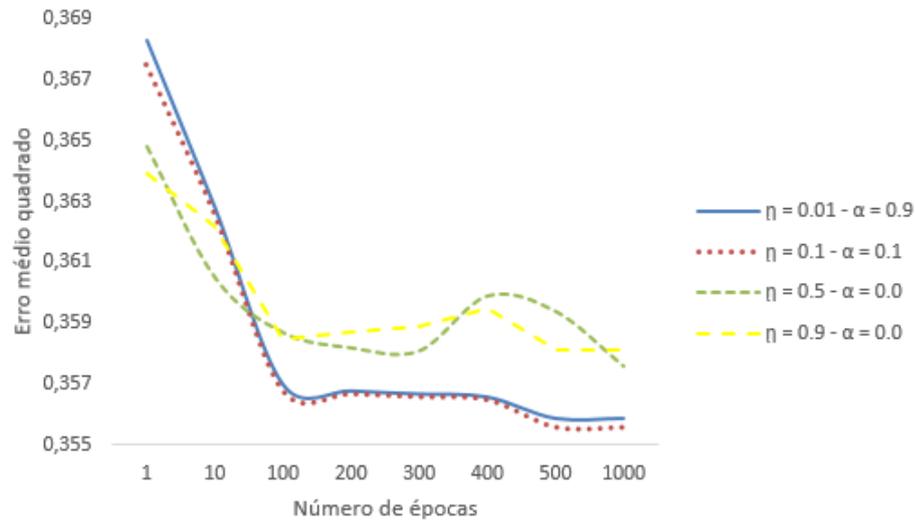
Figura 4.10: Curvas de aprendizagem médias para $\eta = 0.9$ (eletrodo Fp2).

Fonte: Autor.

Observa-se, a partir dos experimentos, ilustrados nas Figuras 4.7, 4.8, 4.9 e 4.10, o mesmo comportamento apresentado na Subseção 4.1.1. Em geral, para um valor pequeno (0.01), atribuído à taxa de aprendizagem (η), resulta em uma convergência mais lenta. O uso das taxas de aprendizagem $\eta = 0.5$ e $\eta = 0.9$, sendo $\alpha = 0.9$, observa-se oscilações no erro médio quadrado durante o processo de aprendizagem, ocasionando um valor mais alto do erro médio quadrado na convergência da RNA.

A Figura 4.11 apresenta as melhores curvas de aprendizagem para cada grupo das curvas ilustradas nas Figuras 4.7, 4.8, 4.9 e 4.10, para determinar a melhor curva de aprendizagem global.

Figura 4.11: Melhores curvas de aprendizagem selecionadas. (eletrodo Fp2).



Fonte: Autor.

Observa-se, a partir da Figura 4.11, que o melhor parâmetro da taxa de aprendizado foi $\eta = 0.1$ e a constante de *momentum* foi $\alpha = 0.1$. A Tabela 4.5 apresenta um resumo dos parâmetros otimizados para a RNA.

Tabela 4.5: Configuração da RNA otimizada (eletrodo Fp2).

Parâmetro	Valor
Número de neurônios da camada oculta	11
Taxa de aprendizagem	0.1
Constante de <i>momentum</i>	0.1

Fonte: Autor.

Dessa forma, a RNA foi avaliada para determinar a sua precisão e capacidade de generalização. A Tabela 4.6 lista os parâmetros, configurados para esse experimento, definidos no arquivo **conf.properties**. O mesmo conjunto de dados com 28.614 instâncias usados anteriormente foram utilizados novamente.

Tabela 4.6: Configuração ótima do arquivo de propriedades (eletrodo Fp2).

Chave	Valor
prop.taxa.aprendizado	0.1
prop.termo.momentum	0.1
prop.numero.neuronio.oculto	11
prop.numero.epocas	1000

Fonte: Autor.

Ao final do experimento, a RNA demonstrou uma precisão média de 50,30% e um percentual de instâncias classificadas corretamente igual a 50,70%. O número de instâncias classificadas corretamente foi de 14510, contra 14104 instâncias classificadas incorretamente. Ao analisarmos a matriz de confusão (Tabela 4.7), observa-se, que a classe E (imaginação de movimento do punho esquerdo), obteve 3288 instâncias classificadas corretamente, contra 3930 instâncias classificadas incorretamente. A classe D (imaginação de movimento do punho direito), obteve 1336 instâncias classificadas corretamente, contra 5750 instâncias classificadas incorretamente. A classe F (imaginação de movimento de ambos os punhos), obteve 7151 instâncias classificadas corretamente, contra 13 instâncias classificadas incorretamente. Já a classe P (imaginação de movimento de ambos os pés), obteve 2735 instâncias classificadas corretamente, contra 4411 instâncias classificadas incorretamente.

Tabela 4.7: Matriz de confusão da classificação dados (eletrodo Fp2).

Dados da amostragem				
Classes	E	D	F	P
E	3288	1428	23	2479
D	3234	1336	26	2490
F	2	0	7151	11
P	3022	1364	25	2735

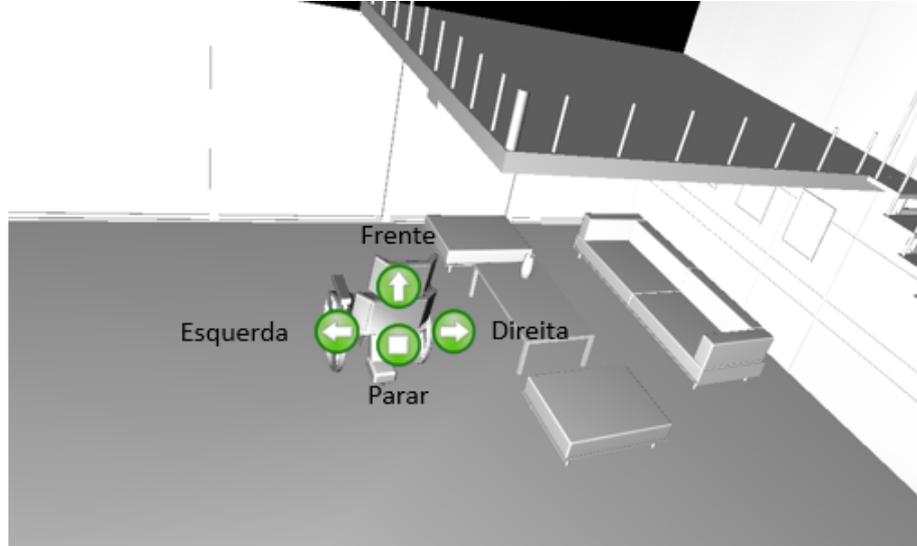
Fonte: Autor.

O método apresentou uma precisão de 99,4% para a classe F (Tabela 4.7), ou seja, a RNA demonstrou boa capacidade de classificação correta para o estado mental relacionado a imaginação de movimento de ambos os punhos, seguida das classes E (39,2%), P (36,8%) e D (23,8%). Conseqüentemente, a classe D apresentou alta taxa de confusão, onde 3234 instâncias foram consideradas como E, 2490 instâncias foram consideradas como P e 26 instâncias foram consideradas como F.

A Figura 4.12 ilustra como poderia ser possível acionar a CRI a partir de 4 estados mentais:

imaginação de movimento do punho esquerdo, direito, ambos os punhos e ambos os pés para acionar uma CRI nas direções: esquerda, direita, frente e parar.

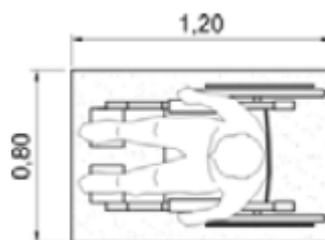
Figura 4.12: Esquema para acionar CRI a partir de 4 estados mentais.



Fonte: Autor.

De acordo com a Associação Brasileira de Normas Técnicas (ABNT), as dimensões referenciais para cadeiras de rodas manuais ou motorizadas é de 0,80 m por 1,20 m no piso, ocupada por uma pessoa utilizando cadeira de rodas, conforme Figura 4.13.

Figura 4.13: Dimensões do módulo de referência.

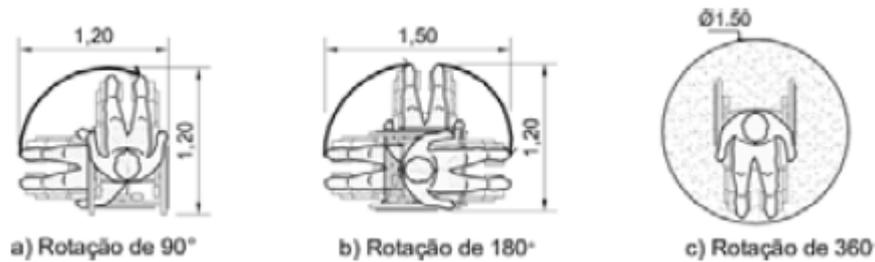


Fonte: [ABNT \(2004\)](#).

As medidas necessárias para a manobra de cadeira de rodas, conforme a Figura 4.14, são:

- para rotação de 90 graus = 1,20 m x 1,20 m;
- para rotação de 180 graus = 1,50 m x 1,20 m;
- para rotação de 360 graus = diâmetro de 1,50 m.

Figura 4.14: Área para manobra sem deslocamento.



Fonte: [ABNT \(2004\)](#).

Dessa forma, é possível guiar uma CRI com quatro ações (esquerda, direita, frente e parar), sem a utilização da opção ré.

4.2 Interface de acionamento

Esse experimento visa avaliar o comportamento da interface de acionamento da CRI. O propósito é validar se os requisitos (Subseção: [3.1.3](#)) estão apropriadamente implementados. A técnica utilizada foi baseada em teste caixa-preta, cujo objetivo é avaliar o comportamento externo do componente, ou seja, validar a saída do componente a partir dos dados de entrada, sem considerar o comportamento interno do mesmo. A Tabela [4.8](#) apresenta o conjunto de entradas para os testes e os resultados esperados para uma determina entrada. O opção ré não foi considerada, pois nesse momento, essa situação não está prevista como uma das possíveis saídas da RNA.

Tabela 4.8: Representação dos dados de teste para os requisitos a serem testado.

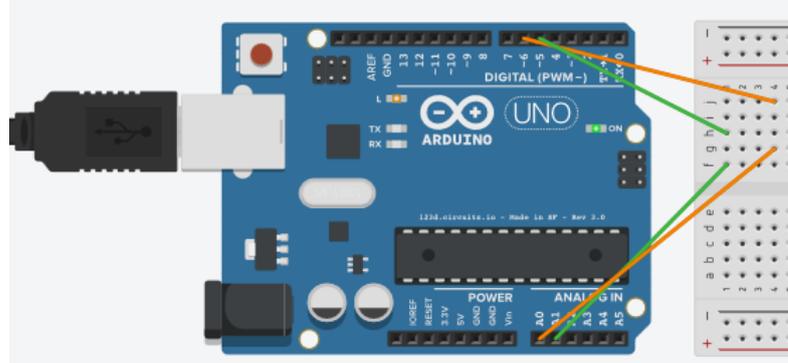
Entrada	Tensão saída		Ação
	5	6	
'0'	2,5 volts	2,5 volts	parar
'1'	3,5 volts	2,5 volts	frente
'3'	2,5 volts	1,5 volts	direita
'4'	2,5 volts	3,5 volts	esquerda

Fonte: Autor.

A Figura [4.15](#) ilustra as tarefas requeridas para executar e avaliar o comportamento do componente. Os dados de entrada são fornecidos ('0', '1', '3' e '4'), o teste é executado e o resultado obtido é comparado a um resultado esperado previamente conhecido. Para

realizar a validação de forma automática, os valores de tensão aplicados nos pinos 5 e 6 foram conectados aos pinos A0 e A1 do microcontrolador e comparados com os valores dos pinos 5 e 6 (Tabela 4.8).

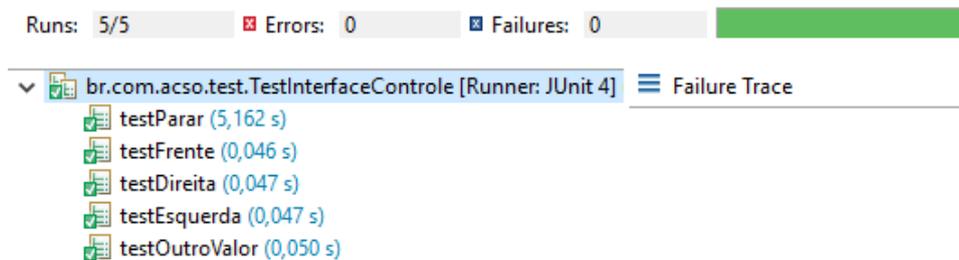
Figura 4.15: Abordagem para executar e avaliar o teste do componente.



Fonte: Autor.

O *framework* JUnit e a IDE Eclipse foram utilizados para a criação e execução das suítes de teste. Quando os testes foram executados em modo gráfico, os métodos testados apresentaram sucesso (Figura 4.16). Os métodos testados poderiam apresentar os seguintes resultados: verde para sucesso, roxo para falha e vermelho para exceção.

Figura 4.16: Resultado do teste da interface de acionamento.



Fonte: Autor.

O caso de teste, `testParar()`, apesar de ter passado no teste, apresentou o tempo de 5,162 segundos para finalizar a execução do teste, esse número não está adequado, visto que o sistema dispõe de apenas 0,5 segundos (sobreposição da janela) para atender a uma requisição de acionamento da CRI. Investigando as classes do Subsistema Dispositivo (Figura 3.13), percebemos que o custo de execução do método `initialize()` da classe Ativador (Figura 4.17) demorava em média 5,163 segundos para abrir a conexão com a porta serial responsável pela comunicação entre o sistema e a interface de acionamento da CRI.

Figura 4.17: Implementação do método `initialize()`.

```

/**
 * Método que verifica se a comunicação com a porta serial está ok
 * @throws PortInUseException
 * @throws IOException
 * @throws UnsupportedOperationException
 */
private void initialize() throws PortInUseException, IOException, UnsupportedOperationException {

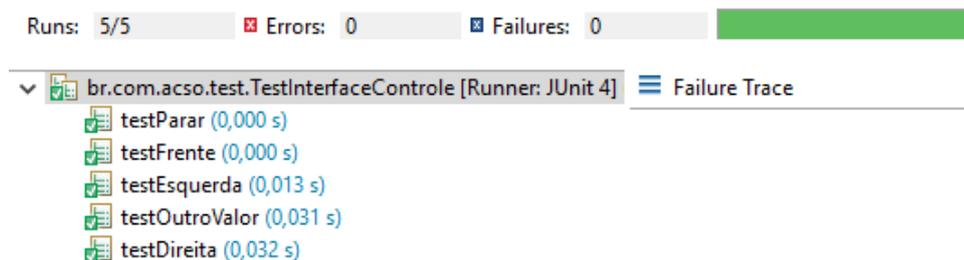
    //Define uma variável portId do tipo CommPortIdentifier para realizar a comunicação serial
    CommPortIdentifier portId = null;
    try {
        //Tenta verificar se a porta COM informada existe
        portId = CommPortIdentifier.getPortIdentifier(this.portaCOM);
    } catch (NoSuchPortException npe) {
        //Caso a porta COM não exista será exibido um erro
        JOptionPane.showMessageDialog(null, "Porta COM não encontrada.",
            "Porta COM", JOptionPane.PLAIN_MESSAGE);
    }
    //Abre a porta COM
    SerialPort port = (SerialPort) portId.open("CRI", this.taxa);
    serialOut = port.getOutputStream();
    port.setSerialPortParams(this.taxa, //taxa de transferência da porta serial
        SerialPort.DATABITS_8, //taxa de 10 bits 8 (envio)
        SerialPort.STOPBITS_1, //taxa de 10 bits 1 (recebimento)
        SerialPort.PARITY_NONE); //receber e enviar dados
}
}

```

Fonte: Autor.

Diante do exposto, realizamos uma alteração na classe `Ativador`, implementamos o padrão de projeto *Singleton* (FREEMAN et al., 2004). Em resumo, a ideia é de que tenhamos uma classe capaz de ser instanciada uma única vez e com visibilidade global. Assim, fizemos o instanciamento da classe `Ativador` (criada em memória e pronta para seu uso) no método `main()` da classe principal do sistema. A Figura 4.18, ilustra o resultado da execução dos testes após as alterações na classe `Ativador`. Observa-se que cada teste (`testParar()`, `testFrente()`, `testDireita()`, `testEsquerda()` e `testOutroValor()`) apresentou tempo abaixo de 0,5 segundos.

Figura 4.18: Resultado do teste da interface de acionamento após modificação.



Fonte: Autor.

Os casos de testes, `testParar()`, `testFrente()`, `testDireita()`, `testEsquerda()` e `testOutroValor()`, foram programados para utilizar o serviço de acionamento da CRI, passando como valor de entrada uma variável do tipo `char`. O retorno a chamada ao método

de acionamento é um vetor de duas posições, do tipo *float*, sendo: o índice [0] o valor aplicado ao pino 5 e o índice [1] o valor aplicado ao pino 6. A classe de teste compara os valores presentes nos índices do vetor com os respectivos valores esperados (Tabela 4.8), em caso positivo o JUnit assinala o resultado com sucesso. O caso de teste, **testOutro-Valor()**, envia o valor '5' ao método de acionamento, ou seja, um valor desconhecido pelo componente de acionamento e o resultado esperado pelo teste é uma exceção contendo a mensagem: valor inválido. Nessa situação, o sistema de acionamento envia envias os sinais correspondente a opção parar (Tabela 4.8).

4.3 Integração do sistema

O objetivo desse experimento é avaliar o comportamento do sistema através da integração dos componentes do modelo definidos na arquitetura do sistema (Seção 3.2). Nesse experimento, a cadeira de rodas *Freedom Carbon* também foi integrada ao sistema, porém, foi colocada sobre uma base fixa (Figura 4.19), mantendo-se as rodas da cadeira livres.

Figura 4.19: Abordagem para executar e avaliar o teste de integração.



Fonte: Autor.

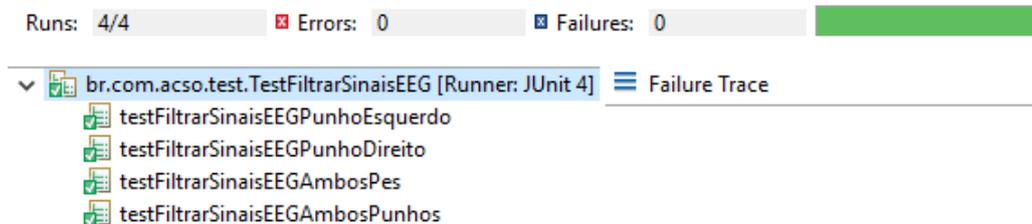
O *framework* JUnit e a IDE Eclipse, foram utilizados para a criação e execução das suítes de teste. Para esse experimento, foram definidos três casos de testes, a saber:

- **Validar camada de Integração:** esse caso de teste tem por objetivo avaliar os componentes participantes da camada de Integração (Figura 3.11).
- **Validar camada Núcleo:** esse caso de teste tem por objetivo avaliar a integração dos componentes participantes entre as camadas Integração e Núcleo (Figura 3.12).
- **Validar integração do sistema:** esse caso de teste tem por objetivo avaliar a integração dos componentes participantes de todo o sistema: Integração, Núcleo e Dispositivo (Figura 3.13).

4.3.1 Validar camada de Integração

A Figura 4.20 ilustra o resultado da execução do caso de teste: Filtrar os sinais EEG. Os métodos testados apresentaram sucesso, ou seja, a partir dos testes, não foram encontradas falhas provenientes da integração interna dos componentes do sistema da camada de Integração.

Figura 4.20: Resultado do teste de integração: *Layer Integration*.



Fonte: Autor.

4.3.2 Validar camada Núcleo

Para a execução do caso de teste Validar camada Núcleo, foram utilizados os dados dos sinais EEG de 5 indivíduos (005, 009, 012, 055 e 066), selecionado de maneira aleatória. A Figura 4.21 ilustra o resultado da execução desse caso de teste. O método testado apresentou sucesso, ou seja, a partir dos testes, não foram encontradas falhas provenientes da integração interna entre os componentes das camadas Integração e Núcleo.

Figura 4.21: Resultado do teste de integração: Integração e Núcleo.



Fonte: Autor.

O método de teste `testDesempenhoRNA()` envolveu os seguintes procedimentos:

1. Filtrar os dados relacionados a imaginação de movimentos do punho esquerdo, punho direito, ambos os punhos e ambos os pés.
2. Aplicar a FFT sobre os dados.
3. Criar o vetor de características.
4. Treinar a RNA .
5. Testar o desempenho da RNA.

4.3.3 Validar integração do sistema

A Figura 4.22 ilustra o resultado da execução do caso de tese Validar integração do sistema. Nesse experimento, a integração entre os três módulos do sistema foram testados: Integração, Núcleo e Dispositivo. Foram utilizados os dados dos sinais EEG de 5 indivíduos (055, 069, 081, 095 e 102), selecionado de maneira aleatória. O método de teste `testAcionarCRI()`, envolveu os seguintes procedimentos:

1. Filtrar os dados relacionados a imaginação de movimentos do punho esquerdo, amostrados em uma taxa de 160 Hz (apenas 1 segundo de coleta).
2. Aplicar a FFT sobre os dados.
3. Criar o vetor de características.
4. Classificar os dados através do resultado da saída da RNA.
5. Acionar a CRI com o valor de saída da RNA.

O método testado apresentou sucesso. Assim, conclui-se que os métodos apresentaram sucesso, ou seja, a partir dos testes, não foram encontradas falhas provenientes da integração interna dos componentes do sistema: Integração, Núcleo e Dispositivo.

Figura 4.22: Resultado do teste de integração do sistema.



Fonte: Autor.

Após a conclusão dos experimentos, de acordo com os resultados, observa-se que a RNA apresentou precisão média de 50,40% para dados do eletrodo Fp1 e 50,30% para dados do eletrodo Fp2, no processo de reconhecimento de padrões a partir dos sinais EEG imaginados do punho direito, punho esquerdo, ambos os punhos e ambos os pés. Foi possível observar ainda, que, a RNA demonstrou boa capacidade de classificação correta para o estado mental relacionado a imaginação de movimento de ambos os punhos, para os dados originados dos dois eletrodos (Fp1 e Fp2). Apresentando 98,9% de precisão para o eletrodo Fp1 e 99,4% de precisão para o eletrodo Fp2. Os dados relacionados a imaginação de movimento do punho direito, apresentou pior desempenho para ambos os eletros: 32,5% de precisão para o eletrodo Fp1 e 23,8% de precisão para o eletrodo Fp2. Nesse contexto, foi possível observar que o desempenho da RNA utilizando dados dos eletrodos Fp1 e Fp2 apresentam valores próximos. Observa-se a partir dos resultados, que é possível reconhecer padrões nos sinais EEG originados no córtex pré-frontal, esse fato sugere a possibilidade de pessoas com mobilidade reduzida, através da imaginação de movimentos poder para acionar uma CRI. Outro fator importante é o fato, por exemplo, de uma pessoa que tenha um lado do cérebro (direito ou esquerdo) paralisado, sendo essa uma das sequelas mais comuns em vítimas de AVC, poder fazer uso de um dos eletrodos (Fp1 ou Fp2) contrário ao lado onde a lesão se apresenta e dessa forma, será possível acionar a CRI e ampliar sua mobilidade.

Considerações finais

Este trabalho de investigação apresenta um modelo computacional baseado em interface cérebro-computador que permite o estudo e desenvolvimento de novas tecnologias a partir dos sinais de EEG com o objetivo de minimizar a dependência social que as pessoas com graves comprometimentos motores têm da sociedade e na ampliação da mobilidade dessas pessoas. A interface cérebro-computador desenvolvida é baseada na imaginação de movimentos do punho esquerdo, punho direito, ambos os punhos e ambos os pés para acionar uma cadeira de rodas inteligente. A imaginação do movimento é considerada como um estado cognitivo que corresponde à simulação mental de uma ação motora sem que haja qualquer manifestação motora real. A metodologia proposta é composta de uma etapa de pré-processamento, extração de características, classificação e acionamento da CRI. Na etapa de pré-processamento, os sinais EEG foram filtrados por um filtro digital passa-faixa, cujas frequências variam de 0,5 a 42 Hz e agrupados em janelas de um segundo. A extração de características é realizada através da FFT, após a decomposição do sinal, informações foram separadas em um conjunto de cinco atributos: média aritmética, energia, máximo valor da FFT, mínimo valor da FFT e frequência dominante. Esses atributos contém o padrão a ser analisado por uma RNA do tipo *Multi Layer Perceptron* (MLP), com o algoritmo de retropropagação do erro (*backpropagation*). A RNA foi configurada com 5 neurônios na camada de entrada, 11 neurônios na camada oculta e 4 neurônios na camada de saída. O processo de classificação foi baseado em aprendizado de máquina, onde o sistema de reconhecimento de padrões (RNA) foi treinado de forma supervisionada, para realizar a correta identificação das intenções de movimentos a partir das tarefas mentais. A classificação dos dados incluem respostas a quatro tarefas mentais distintas, ampliando o grau de liberdade do indivíduo em quatro direções: esquerda, direita, frente e parar. O processo de acionamento da CRI é responsável por realizar os movimentos na cadeira de rodas.

5.1 Conclusões

A interface cérebro-computador proposta nesta pesquisa, através dos experimentos realizados, como detalhado no Capítulo 4, mostrou ser possível identificar nos sinais EEG, a partir dos eletrodos Fp1 e Fp2 localizados na região do córtex pré-frontal, características que expressem a intenção de quatro movimentos para acionar uma CRI. Conforme os objetivos propostos inicialmente nesta pesquisa, uma arquitetura de *software* foi proposta para atender as necessidades do modelo computacional. Três subsistemas foram desenvol-

vidos: Integração, Núcleo e Dispositivo, responsáveis pelo funcionamento do sistema. O subsistema Integração foi desenvolvido para extrair e filtrar os dados brutos referentes aos sinais EEG. O subsistema Núcleo provê as funcionalidades referentes à lógica do sistema, integra os componentes de classificação, treinamento, teste e fórmulas matemáticas. O subsistema Dispositivo encapsula a lógica relacionada a integração do sistema com a CRI.

A capacidade do sistema proposto em discernir entre quatro intenções de movimentos, foi avaliada nos experimentos, conforme detalhado na Seção 4.1. A RNA apresentou precisão média de 50,40% para dados do eletrodo Fp1 e 50,30% para dados do eletrodo Fp2, a RNA demonstrou boa capacidade de classificação correta para o estado mental relacionado a imaginação de movimento de ambos os punhos, 98,9% de precisão para o eletrodo Fp1 e 99,4% de precisão para o eletrodo Fp2. Os dados relacionados a imaginação de movimento do punho direito, apresentou pior desempenho para ambos os eletros: 32,5% de precisão para o eletrodo Fp1 e 23,8% de precisão para o eletrodo Fp2.

Pôde-se notar, conforme detalhado na Seção 4.3, que não foram encontradas falhas provenientes da integração interna dos componentes do sistema: Integração, Núcleo e Dispositivo. Os métodos das classes participantes do sistema foram testados e apresentaram sucesso em todos os casos de testes definidos.

Dessa forma, a interface cérebro-computador baseado nos resultados, mostrou-se útil para reconhecer padrões a partir de sinais EEG e acionar uma cadeira de rodas. Para a navegação da cadeira de rodas, grande autonomia é atribuída a CRI, cujo sistema integrado, equipado com vários sensores, é capaz de executar uma navegação segura em um ambiente semi-estruturado.

As técnicas desenvolvidas nesta pesquisa são promissoras, o sistema destina-se a contribuir como completo na ampliação da mobilidade de pessoas com graves comprometimentos motores. Uma fase experimental mais longa, com pessoas deficientes faz-se necessária, com o objetivo de uma avaliação mais completa.

5.2 Contribuições

As principais contribuições desenvolvidas neste trabalho de investigação são:

- Desenvolvimento de uma arquitetura para sistemas baseados em interfaces cérebro-computador. Os projetos que utilizam essa tecnologia apresentaram diferentes aspectos no seu desenvolvimento e de forma restrita. Nesse trabalho de investigação procurou-se uma aplicação mais ampla dos conceitos que envolvem sistemas base-

ados em interfaces cérebro-computador, promovendo direções no desenvolvimento desses sistemas.

- A proposta de uma interface cérebro-computador utilizando apenas dois eletrodos para aquisição de sinais EEG, permite um menor consumo de energia permitindo a utilização do sistema por um período de tempo mais longo.
- Outra contribuição em relação a utilização de dois eletrodos está relacionada com a representação social e pessoal do indivíduo com deficiência. A diminuição do número de eletrodos no processo de aquisição de sinais EEG decrementa a percepção simbólica da condição de deficiência que indivíduo deficiente enfrenta.

Nesse contexto, a utilização de ferramentas de *software*, que automatizem o desenvolvimento e testes de sistemas baseados em interfaces cérebro-computador, pode aprimorar a qualidade desses sistemas, o que pode trazer benefícios aos utilizadores desses sistemas, no sentido de devolver-lhe o direito de locomoção, presenteando-os não só com a autonomia para vários atos da vida, como também lhe devolve a dignidade, tão essencial à vida humana.

5.3 Atividades Futuras de Pesquisa

Como proposta de atividade futura desta pesquisa, uma questão importante a ser tratada é o desenvolvimento do sistema de navegação e controle da CRI em ambientes *indoor*. A abordagem adotada para a implementação do sistema de navegação será baseada em [Souza et al. \(2015\)](#), adaptando-a às necessidades do projeto de pesquisa envolvendo a cadeira de rodas inteligente.

Na camada de Integração, a etapa de aquisição de sinais EEG, utiliza uma janela de tamanho fixo. Uma possível proposta de continuidade dessa pesquisa seria a possibilidade de implementar uma janela dinâmica para melhorar a resolução temporal. Na camada de Núcleo, uma proposta seria o desenvolvimento de outros algoritmos para o reconhecimento de padrões, por exemplo, Árvore de decisão e Máquina de vetores de suporte.

Outra proposta de atividade futura, seria a realização de testes em pessoas com graves deficiências motoras para melhorar a confiabilidade dos resultados. Dessa forma, mais experimentos, com pessoas deficientes será de grande valor na continuidade do desenvolvimento da interface cérebro-computador e na ampliação da base de dados para treinamento dos classificadores.

Embora as interfaces cérebro-computador seja uma ferramenta útil para auxiliar indivíduos com graves deficiências motoras, algumas questões relacionadas a sua utilização

por um período prolongado de tempo ainda é um desafio. Nesse contexto, sugere-se ainda, o desenvolvimento de um *sensor* contendo dois eletrodos, localizados nas posições: Fp1 e Fp2 (Figura 2.7) respectivamente. Esse sensor é proposto para melhorar o conforto do indivíduo em utilizações prolongadas e menor consumo de energia. Estas duas vertentes de investigação pretendem contribuir para a ergonomia da interface cérebro-computador e permitir maior mobilidade aos indivíduos.

Referências Bibliográficas

- ABNT. *Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos*. [S.l.], 2004.
- BAREA, R.; BOQUETE, L.; MAZO, M.; LOPEZ, E. System for assisted mobility using eye movements based on electrooculography. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, v. 10, n. 4, p. 209–218, Dec 2002. ISSN 1534-4320.
- BAYLISS, J. D. *A Flexible Brain-Computer Interface*. Rochester, NY, USA, 2001.
- BEVERINA, F.; PALMAS, G.; SILVONI, S.; PICCIONE, F.; GIOVE, S. User adaptive bcis: Ssvep and p300 based interfaces. *PsychNology Journal*, v. 1, n. 4, p. 331–354, 2003.
- BHATTACHARYYA, S.; SHIMODA, S.; HAYASHIBE, M. A synergetic brain-machine interfacing paradigm for multi-dof robot control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 46, n. 7, p. 957–968, July 2016. ISSN 2168-2216.
- BIRBAUMER, N. et al. The thought translation device (ttd) for completely paralyzed patients. *IEEE Transactions on Rehabilitation Engineering*, v. 8, n. 2, p. 190–193, Jun 2000. ISSN 1063-6528.
- BIRCH, G. E.; BOZORGZADEH, Z.; MASON, S. G. Initial on-line evaluations of the lf-asd brain-computer interface with able-bodied and spinal-cord subjects using imagined voluntary motor potentials. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 10, n. 4, p. 219–224, Dec 2002. ISSN 1534-4320.
- CARLSON, T.; MILLAN, J. del R. Brain-controlled wheelchairs: A robotic architecture. *IEEE Robotics Automation Magazine*, v. 20, n. 1, p. 65–73, March 2013. ISSN 1070-9932.
- CARMENA, J. M.; LEBEDEV, M. A.; HENRIQUEZ, C. S.; NICOLELIS, M. A. L. Stable ensemble performance with single-neuron variability during reaching movements in primates. *The Journal of Neuroscience*, v. 25, n. 46, p. 10712–10716, 2005. Disponível em: <http://www.jneurosci.org/content/25/46/10712.abstract>.
- CHOI, J. et al. Effectiveness of commercial gaming-based virtual reality movement therapy on functional recovery of upper extremity in subacute stroke patients. *Ann Rehabil Med*, v. 38, p. 485 – 493, 2014.
- CORDINI, K. L.; ODA, E. Y.; FURLANETTO, L. M. Quality of life of patients with previous history of stroke: description of cases. *Jornal brasileiro de psiquiatria*, v. 54, n. 4, 2005.

CRAJÉ, C. et al. Compromised motor planning and motor imagery in right hemiparetic cerebral palsy. *Research in Developmental Disabilities*, v. 31, n. 6, p. 1313 – 1322, 2010. ISSN 0891-4222. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0891422210001666>>.

CRUZ, K. C. T. d.; DIOGO, M. J. D. Avaliação da capacidade funcional de idosos com acidente vascular encefálico. *Acta Paulista de Enfermagem*, scielo, v. 22, p. 666 – 672, 10 2009. ISSN 0103-2100.

DALPIAN, A.; GRAVE, M.; PÉRICO, E. Avaliação da percepção corporal em pacientes pós-acidente vascular cerebral (avc). *Revista Neurociências*, v. 21, p. 377–382, October 2013.

DANCKERT, J. et al. Selective, non-lateralized impairment of motor imagery following right parietal damage. *Neurocase*, v. 8, n. 3, p. 194–204, 2002.

DAVID, E. R.; GEOFFREY, E. H.; RONALD, J. W. Learning representations by back-propagating errors. *Nature*, v. 323, p. 533–536, 1986.

DECETY, J.; BOISSON, D. Effect of brain and spinal cord injuries on motor imagery. In: . [S.l.]: Eur Arch Psychiatry Clin Neurosci, 1990. v. 1, p. 39–43.

DECHENT, P.; MERBOLDT, K.-D.; FRAHM, J. Is the human primary motor cortex involved in motor imagery? *Cognitive Brain Research*, v. 19, n. 2, p. 138 – 144, 2004. ISSN 0926-6410. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0926641003002945>>.

DEGROOT, J. *Neuroanatomia*. Rio de Janeiro: Editora Guanabara, 1994. ISBN 9781285174150.

DONATI, A. R. C. et al. Long-term training with a brain-machine interface-based gait protocol induces partial neurological recovery in paraplegic patients. *Scientific Reports*, v. 6, n. 30383, 2016. Disponível em: <<http://www.nature.com/articles/srep30383>>.

DONCHIN, E.; SPENCER, K. M.; WIJESINGHE, R. The mental prosthesis: assessing the speed of a p300-based brain-computer interface. *IEEE Transactions on Rehabilitation Engineering*, v. 8, n. 2, p. 174–179, Jun 2000. ISSN 1063-6528.

EDELMAN, B. J.; BAXTER, B.; HE, B. Eeg source imaging enhances the decoding of complex right-hand motor imagery tasks. *IEEE Transactions on Biomedical Engineering*, v. 63, n. 1, p. 4–14, Jan 2016. ISSN 0018-9294.

EICH, H.-J.; MACH, H.; WERNER, C.; HESSE, S. Aerobic treadmill plus bobath walking training improves walking in subacute stroke: a randomized controlled trial. *Clinical Rehabilitation*, v. 18, n. 6, p. 640–651, 2004.

- EID, M. A.; GIAKOUMIDIS, N.; SADDIK, A. E. A novel eye-gaze-controlled wheelchair system for navigating unknown environments: Case study with a person with als. *IEEE Access*, v. 4, p. 558–573, 2016. ISSN 2169-3536.
- EL-MADANI, A.; SORENSEN, H. B.; W., K. T.; THOMSEN, C. E.; PUTHUS-SERYPADY, S. Real-time brain computer interface using imaginary movements. *EPJ Nonlinear Biomedical Physics*, v. 9, n. 3, 2015.
- FARIA, B. M.; REIS, L. P.; LAU, N. Cerebral palsy eeg signals classification: Facial expressions and thoughts for driving an intelligent wheelchair. In: *2012 IEEE 12th International Conference on Data Mining Workshops*. [S.l.: s.n.], 2012. p. 33–40. ISSN 2375-9232.
- FARIA, B. M.; REIS, L. P.; LAU, N.; SOARES, J. C.; VASCONCELOS, S. Patient classification and automatic configuration of an intelligent wheelchair. In: _____. *Agents and Artificial Intelligence: 4th International Conference, ICAART 2012, Vilamoura, Portugal, February 6-8, 2012. Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 268–282. ISBN 978-3-642-36907-0. Disponível em: http://dx.doi.org/10.1007/978-3-642-36907-0_18.
- FAUCONNIER, G.; TURNER, M. *The way we Think: Conceptual Blending and the Mind's Hidden Complexities*. New York: Basic Books, 2002.
- FECHIO, M.; PACHECO, K.; KAIHAMI, H.; ALVES, V. A repercussão da lesão medular na identidade do sujeito. *Acta Fisiátrica*, v. 16, n. 1, p. 38–42, 2009.
- FREEDOM. *FREEDOM CARBON*: Manual do proprietário. Porto Alegre, 2012. 24 p.
- FREEMAN, E.; ROBSON, E.; BATES, B.; SIERRA, K. *Head First Design Patterns: A Brain-Friendly Guide*. [S.l.]: OREILLY, 2004.
- FURB. *Conceitos*. 2015. FURB. Disponível em: <http://www.inf.furb.br/sias/saude/Textos/Conceitos.htm>. Acesso em: 05 janeiro 2016.
- GENTILI, R.; PAPAXANTHIS, C.; POZZO, T. Improvement and generalization of arm motor performance through motor imagery practice. *Neuroscience*, v. 137, n. 3, p. 761 – 772, 2006. ISSN 0306-4522. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306452205011541>.
- GEORGE, A.; AZEVEDO, F. M. Identificação automática de complexos k e fusos do sono em sinais de eeg utilizando redes neurais e transformada wavelet. In: *Anais do 8 Congresso Brasileiro de Redes Neurais*. Florianópolis: SBRN, 2007. p. 1–6.
- GOLDBERGER, A. L. et al. Physiobank, physiotookit, and physionet. *Circulation, American Heart Association Journals*, v. 101, n. 23, p. e215–e220, 2000. ISSN 0009-7322. Disponível em: <http://circ.ahajournals.org/content/101/23/e215>.

HAFER, C. E.; RYAN, A. S.; IVEY, F. M.; MACKO, R. F. Skeletal muscle changes after hemiparetic stroke and potential beneficial effects of exercise intervention strategies. *Journal of Rehabilitation Research and Development*, v. 45, p. 261 – 272, 2008. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2978978/>>.

HALL, M. et al. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Disponível em: <<http://doi.acm.org/10.1145/1656274.1656278>>.

HAN, J.; KAMBER, M. *Data Mining: Concepts and Techniques*. USA: Morgan Kaufmann, 2006.

HASHIMOTO, M.; TAKASHIMA, Y.; UCHINO, A.; YUZURIHA, T.; YAO, H. Dual task walking reveals cognitive dysfunction in community-dwelling elderly subjects: The sefuri brain mri study. *Journal of stroke and cerebrovascular disease*, v. 23, p. 1770 – 1775, 2014.

HIRZEL, T. *PWM*. [S.l.], 2016. Disponível em: <<https://www.arduino.cc/en/Tutorial/PWM>>. Acesso em: 20 nov. 2016.

HUNG, C.-I. et al. Recognition of motor imagery electroencephalography using independent component analysis and machine classifiers. *Annals of Biomedical Engineering*, v. 33, n. 8, p. 1053 – 1070. ISSN 1573-9686. Disponível em: <<http://dx.doi.org/10.1007/s10439-005-5772-1>>.

IBGE. *Pesquisa nacional de saúde: 2013: percepção do estado de saúde, estilos de vida e doenças crônicas : Brasil, grandes regiões e unidades da federação*. Rio de Janeiro: [s.n.], 2014. ISBN 9788524043345. Disponível em: <URL: <http://loja.ibge.gov.br/pesquisa-nacional-de-saude-2013-percepc-o-do-estado-de-saude-estilos-de-vida-e-doencas-cronicas.html>>.

JACKS, A. S.; MILLER, N. R. Spontaneous retinal venous pulsation: aetiology and significance. *Journal of Neurology, Neurosurgery & Psychiatry*, v. 74, n. 1, p. 9, 2003. Disponível em: <<http://jnnp.bmj.com/content/74/1/9.short>>.

JOHAN, W. et al. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, v. 408, p. 361–365, November 2000.

JOHNSON, R. E. Documenting frameworks using patterns. *SIGPLAN Notices*, v. 27, n. 10, 1992.

JONSDOTTIR, J.; CATTANEO, D. Reliability and validity of the dynamic gait index in persons with chronic stroke. *Arch Phys Med Rehabil*, v. 88, p. 1410 – 1415, 2007.

JOSE, M. A.; LOPES, R. de D. Human - computer interface controlled by the lip. *IEEE Journal of Biomedical and Health Informatics*, v. 19, n. 1, p. 302–308, Jan 2015. ISSN 2168-2194.

KAMMLER, D. W. *A First Course in Fourier Analysis*. 2nd. ed. [S.l.]: Cambrigde, 2008. ISBN 978-0521709798.

KAUFMANN, T.; HERWEG, A.; KÜBLER, A. Toward brain-computer interface based wheelchair control utilizing tactually-evoked event-related potentials. *Journal of NeuroEngineering and Rehabilitation*, v. 11, n. 1, p. 1–17, 2014. ISSN 1743-0003. Disponível em: <<http://dx.doi.org/10.1186/1743-0003-11-7>>.

LAY, S. M.; PIZARRO, D. A. Robotic motion control with cognitive and facial detection via emotiv eeg. *Ingeniare. Revista chilena de ingeniería*, scielo, v. 23, p. 496 – 504, 10 2015. ISSN 0718-3305.

LEUCHTER, A. et al. Regional differences in brain electrical activity in dementia: use of spectral power and spectral ratio measures. *Electroencephalogr Clin Neurophysiol*, v. 87, n. 6, p. 385 – 393, 1993.

LIMA, N. M. et al. Motor and sensory effects of ipsilesional upper extremity hypothermia and contralesional sensory training for chronic stroke patients. *Topics in Stroke Rehabilitation*, v. 22, n. 1, p. 44–55, 2015. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1179/1074935714Z.0000000023>>.

LOTUFO, P. A.; BENSENOR, I. M. Mortalidade pela doença cerebrovascular em são paulo (1997-2003): descrição utilizando a décima classificação internacional de doenças. *Arquivos de Neuro-Psiquiatria*, scielo, v. 62, p. 1008 – 1011, 12 2004. ISSN 0004-282X.

MARK, F. B.; BARRY, W. C.; MICHAEL, A. P. *Neuroscience: Exploring the Brain*. 3rd. ed. [S.l.]: Lippincott Williams & Wilkins, 2007. ISBN 978-0781760034.

MARK, F. B.; BARRY, W. C.; MICHAEL, A. P. *Neurociências: Desvendando o Sistema Nervoso*. Porto Alegre: Artmed, 2008. ISBN 9788536313337.

MILLAN, J. d. R. et al. A local neural classifier for the recognition of eeg patterns associated to mental tasks. *IEEE Transactions on Neural Networks*, v. 13, n. 3, p. 678–686, May 2002. ISSN 1045-9227.

MILLAN, J. R.; MOURINO, J. Asynchronous bci and local neural classifiers: an overview of the adaptive brain interface project. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 11, n. 2, p. 159–161, June 2003. ISSN 1534-4320.

MILLAN, J. R.; RENKENS, F.; MOURINO, J.; GERSTNER, W. Noninvasive brain-actuated control of a mobile robot by human eeg. *IEEE Transactions on Biomedical Engineering*, v. 51, n. 6, p. 1026–1033, June 2004. ISSN 0018-9294.

MS. *Saúde Brasil 2008 : 20 anos de Sistema Único de Saúde (SUS) no Brasil / Ministério da Saúde, Secretaria de Vigilância em Saúde, Departamento de Análise de Situação em Saúde*. Braslia: [s.n.], 2009. ISBN 978-85-334-1600-0. Disponível em: <URL: http://bvsmms.saude.gov.br/bvs/publicacoes/saude_brasil_2008.pdf>.

- NICIDA, D. P. *Neurociência: compreendendo o funcionamento do sistema nervoso*. 2015. Neurociências da Aprendizagem. Disponível em: <<http://www.neuroeducacao.com.br/neurociencias.asp>>. Acesso em: 15 maio 2015.
- NORANI, N. A. M.; MANSOR, W.; KHUAN, L. Y. A review of signal processing in brain computer interface system. In: *Biomedical Engineering and Sciences (IECBES), 2010 IEEE EMBS Conference on*. [S.l.: s.n.], 2010. p. 443–449.
- OLIVEIRA, R. d.; CACHO, E. W. A.; BORGES, G. Post-stroke motor and functional evaluations: a clinical correlation using fugl-meyer assessment scale, berg balance scale and barthel index. *Arquivos de Neuro-Psiquiatria*, scielo, v. 64, p. 731 – 735, 09 2006. ISSN 0004-282X.
- PASCUAL-LEONE, A.; AMEDI, A.; FREGNI, F.; MERABET, L. B. The plastic human brain cortex. *Annual Review of Neuroscience*, v. 28, n. 1, p. 377–401, 2005. PMID: 16022601. Disponível em: <<http://dx.doi.org/10.1146/annurev.neuro.27.070203.144216>>.
- PEDREIRA, L. C.; LOPES, R. L. M. Cuidados domiciliares ao idoso que sofreu acidente vascular cerebral. *Revista Brasileira de Enfermagem*, scielo, v. 63, p. 837 – 840, 10 2010. ISSN 0034-7167.
- PETRY, M. R. *A Vision-based Approach Towards Robust Localization for Intelligent Wheelchairs*. Doctor in Informatics Engineering — University of Porto, Porto, 2013.
- PFURTSCHELLER, G.; BRUNNER, C.; SCHLÖGL, A.; SILVA, F. Mu rhythm (de)synchronization and {EEG} single-trial classification of different motor imagery tasks. *NeuroImage*, v. 31, n. 1, p. 153 – 159, 2006. ISSN 1053-8119. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1053811905025140>>.
- PFURTSCHELLER, G.; NEUPER, C. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, v. 89, n. 7, p. 1123–1134, Jul 2001. ISSN 0018-9219.
- PID. *Pesquisa, Informação e Desenvolvimento em Epidemiologia*. 2015. E-PID. Disponível em: <<http://www.uff.br/e-pid/prevalencia.htm>>. Acesso em: 05 janeiro 2016.
- PINTO, L. C. *Neurofisiologia Clínica - Princípios Básicos e Aplicações*. [S.l.]: Atheneu, 2006. ISBN 9788538801436.
- RAFII, M. S.; HILLIS, A. E. Compendium of cerebrovascular diseases. *International Review of Psychiatry*, v. 18, n. 5, p. 395–407, 2006. Disponível em: <<http://dx.doi.org/10.1080/09540260600935405>>.
- RANGEL, E. S. S.; BELASCO, A. G. S.; DICCINI, S. Qualidade de vida de pacientes com acidente vascular cerebral em reabilitação. *Acta Paulista de Enfermagem*, scielo, v. 26, p. 205 – 212, 2013. ISSN 0103-2100.

RANI, P.; SARKAR, M. Emg-based high level human-robot interaction system for people with disability. In: *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*. [S.l.: s.n.], 2005. p. 280–285.

REZAEI, S.; TAVAKOLIAN, K.; NASRABADI, A. M.; SETAREHDAN, S. K. Different classification techniques considering brain computer interface applications. *Journal of Neural Engineering*, v. 3, n. 2, p. 139, 2006. Disponível em: <http://stacks.iop.org/1741-2552/3/i=2/a=008>.

RICCI, N. A.; KUBOTA, M. T.; CORDEIRO, R. C. Concordância de observações sobre a capacidade funcional de idosos em assistência domiciliar. *Revista de Saúde Pública*, scielo, v. 39, p. 655 – 662, 08 2005. ISSN 0034-8910.

RIZZO, D. C. *Fundamentals of Anatomy and Physiology*. 4th edition. ed. New Jersey: Prentice Hall, 2010. ISBN 9781285174150.

Sá, B.; GRAVE, M.; PÉRICO, E. Profile of patients hospitalized with stroke in a hospital of vale do taquari/rs. *Revista Neurociências*, v. 22, p. 381–387, September 2014.

SCHALK, G.; MCFARLAND, D. J.; HINTERBERGER, T.; BIRBAUMER, N.; WOLPAW, J. R. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on Biomedical Engineering*, v. 51, n. 6, p. 1034–1043, June 2004. ISSN 0018-9294.

SCHMIDT, E. M. Single neuron recording from motor cortex as a possible source of signals for control of external devices. *Biomedical Engineering*, v. 8, n. 4, p. 339–349, July 1980.

SILVA, F. L.; NIEDERMEYER, E. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. New Jersey: [s.n.], 1982.

SOUZA, J. R. de; MERCÊS, R. S. M. R. C.; PINHEIRO, O. R.; oES, M. A. C. S. Bahiart@home 2016 team description paper. In: ROBOCUP FEDERATION. *Proceedings of RoboCup 2015*. Hefei, China, 2015. Disponível em: http://www.acso.uneb.br/bill/uploads/Main/BahiaRTHome-RoboCup-2016_TDP_2016-02-19a.pdf.

SOUZA, M. d. F. M.; ALENCAR, A. P.; MALTA, D. C.; MOURA, L.; MANSUR, A. d. P. Análise de séries temporais da mortalidade por doenças isquêmicas do coração e cerebrovasculares, nas cinco regiões do Brasil, no período de 1981 a 2001. *Arquivos Brasileiros de Cardiologia*, scielo, v. 87, p. 735 – 740, 12 2006. ISSN 0066-782X.

STEPHEN, P. D.; ALBERT, H. K.; DANIEL, L. S.; JOHN, W. M. Characterization of the mu rhythm during rapid eye movement sleep. *Clinical Neurophysiology*, v. 112, n. 3, p. 528 – 531, 2001.

TEHOVNIK, E.; WOODS, L.; SLOCUM, W. Transfer of information by {BMI}. *Neuroscience*, v. 255, p. 134 – 146, 2013. ISSN 0306-4522. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306452213008476>.

- VALLABHANENI, A.; WANG, T.; HE, B. Brain-computer interface. In: _____. *Neural Engineering*. Boston, MA: Springer US, 2005. p. 85–121. ISBN 978-0-306-48610-4. Disponível em: <http://dx.doi.org/10.1007/0-306-48610-5_3>.
- VIDAL, J. J. Real-time detection of brain events in eeg. *Proceedings of the IEEE*, v. 65, n. 5, p. 633–641, May 1977. ISSN 0018-9219.
- VIDAURRE, C.; SCHLOGL, A.; CABEZA, R.; SCHERER, R.; PFURTSCHELLER, G. A fully on-line adaptive bci. *IEEE Transactions on Biomedical Engineering*, v. 53, n. 6, p. 1214–1219, June 2006. ISSN 0018-9294.
- VILELA, A. L. M. *Sinapses elétricas*. 2008. Anatomia e Fisiologia Humanas. Disponível em: <<http://www.afh.bio.br/nervoso/nervoso2.asp>>. Acesso em: 20 maio 2015.
- WANG, H.; LI, Y.; LONG, J.; YU, T.; GU, Z. An asynchronous wheelchair control by hybrid eeg/eog brain-computer interface. *Cognitive Neurodynamics*, v. 8, n. 5, p. 399409, 2014. Disponível em: <<http://doi.org/10.1007/s11571-014-9296-y>>.
- WEBSTER, J. G. *Medical Instrumentation Application and Design*. 4th edition. ed. Nova York: John Wiley & Sons, 2009. ISBN 978-0471676003.
- WEN, D. *Monitorização contínua do paciente por VÍDEO-EEG*. 2006. Epilepsia. Disponível em: <http://www.epilepsia-cirurgia.com.br/etapas_investigacao.htm>. Acesso em: 12 novembro 2015.
- WHO. *WHO global disability action plan 2014-2021. Better health for all people with disability*. Switzerland: [s.n.], 2015. ISBN 978 92 4 150961 9. Disponível em: <URL: http://apps.who.int/iris/bitstream/10665/199544/1/9789241509619_eng.pdf?ua=1>.
- WITKOWSKI, M. et al. Enhancing brain-machine interface (bmi) control of a hand exoskeleton using electrooculography (eog). *Journal of NeuroEngineering and Rehabilitation*, v. 11, n. 1, p. 1–6, 2014. ISSN 1743-0003. Disponível em: <<http://dx.doi.org/10.1186/1743-0003-11-165>>.
- WLADIMIR. *Cientistas controlam o corpo de uma pessoa com cérebro de outra*. 2013. Nerdices. Disponível em: <<http://nerdices.com.br/42/2013/08/28/cientistas-controlam-corpo-cerebro/>>. Acesso em: 15 novembro 2015.
- WOLPAW, J.; BIRBAUMER, N.; MCFARLAND, D.; PFURTSCHELLER, G.; VAUGHAN, T. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, v. 113, n. 6, p. 767–791, 2002. ISSN 1388-2457.
- WOLPAW, J.; WOLPAW, E. W. *Brain-Computer Interfaces: Principles and Practice*. New York: Oxford University Press, 2012.
- WOLPAW, J. R. Brain-computer interfaces as new brain output pathways. *Physiology*, v. 579, n. 3, p. 613619, Jan 2007.

XIAO, D.; MU, Z.; HU, J. Classification of motor imagery eeg signals based on energy entropy. p. 61–64, May 2009.

YI, W. et al. Eeg feature comparison and classification of simple and compound limb motor imagery. *Journal of NeuroEngineering and Rehabilitation*, v. 10, n. 1, p. 1 – 12, 2013. ISSN 1743-0003. Disponível em: <http://dx.doi.org/10.1186/1743-0003-10-106>.

ZIJLSTRA, A.; MANCINI, M.; CHIARI, L.; ZIJLSTRA, W. Biofeedback for training balance and mobility tasks in older populations: a systematic review. *NeuroEngineering and Rehabilitation*, v. 7, n. 58, 2010.

Physionet EEG Motor Movement/Imagery Dataset

As Tabelas: [A.1](#), [A.2](#), [A.3](#) e [A.4](#), apresentam os dados dos indivíduos que participaram da formação do banco de dados *eegmdb* - *EEG Motor Movement/Imagery Dataset*. Informações desconhecidas esto vazias. Na coluna **Destreza**, a letra **L** significa: *left* (esquerda), já a letra **R** significa: *right* (direita). Na coluna **Sexo**, a letra **M** significa: *male* (masculino), já a letra **F** significa: *female* (feminino).

Tabela A.1: Dados dos participantes do experimento (Parte 1).

Indivíduo	Idade	Sexo	Destreza
S001	19	M	R
S002	26	F	L
S003	36	M	R
S004	44	M	R
S005	50	F	R
S006	47	F	R
S007	34	F	R
S008	23	F	R
S009	42	F	R
S010	26	M	R
S011	23	M	R
S012	36	F	R
S013	32	F	R
S014	41	F	R
S015	38	F	R
S016	51	F	R
S017	50	F	L
S018	44	F	R
S019	48	F	R
S020	27	M	R
S021		M	R
S022	45	F	R
S023		F	R
S024	54	M	R
S025	39	F	R
S026	34	F	R
S027	31	M	R
S028	44	M	R

Tabela A.2: Dados dos participantes do experimento (Parte 2).

Indivíduo	Idade	Sexo	Destreza
S029		F	R
S030	57	M	R
S031	31	M	R
S032	56	F	R
S033	49	F	R
S034	19	M	R
S035	30	M	R
S036	48	F	R
S037	54	F	R
S038	33	M	L
S039	27	F	R
S040	34	M	R
S041	55	F	R
S042	22	F	R
S043	26	F	R
S044	36	F	R
S045	47	F	R
S046	40	F	L
S047	44	F	R
S048	39	F	R
S049	46	F	R
S050	55	F	R
S051	50	M	R
S052	25	M	L
S053	32	F	R
S054	27	F	R
S055	30	M	R
S056	51	F	R
S057	52	M	R
S058	67	M	R
S059	48	M	R
S060	25	F	R
S061	42	F	R
S062	27	M	R
S063	51	F	R
S064	24	M	R
S065	57	M	R
S066	30	M	R

Tabela A.3: Dados dos participantes do experimento (Parte 3).

Indivíduo	Idade	Sexo	Destreza
S067	31	F	R
S068	43	M	R
S069	52	M	R
S070	30	M	L
S071	47	M	L
S072	28	M	R
S073	38	F	R
S074	48	F	R
S075	26	M	R
S076	58	F	R
S077	37	F	R
S078	23	F	R
S079	33	F	R
S080	39	M	R
S081	31	F	R
S082	51	F	R
S083	49	F	R
S084	36	M	R
S085	28	F	R
S086		F	R
S087	32	M	R
S088	47	M	
S089	59	M	R
S090	53	M	L
S091	43	F	R
S092	43	M	L
S093	27	F	R
S094	20	M	R
S095		F	
S096	32	F	L
S097	32	F	R
S098	34	F	R
S099	44	M	L
S100	20	F	R
S101		M	R
S102	56	F	R
S103		M	

Tabela A.4: Dados dos participantes do experimento (Parte 4).

Indivíduo	Idade	Sexo	Destreza
S104			
S105			
S106			
S107			
S108			
S109			

Este conjunto de dados¹ foi criado para o PhysioBank (GOLDBERGER et al., 2000), por Gerwin Schalk e seus colegas do Programa de Pesquisa e Desenvolvimento da Brain-computer interface (BCI) do Wadsworth Center, no Departamento de Saúde do Estado de Nova York, em Albany, NY. W.A. Sarnacki coletou os dados. Aditya Joshi compilou o conjunto de dados e preparou a documentação. D.J. McFarland e J.R. Wolpaw foram responsáveis pela concepção experimental e supervisão do projeto, respectivamente. Este trabalho foi apoiado por bolsas de NIH / NIBIB ((EB006356 (GS) e EB00856 (JRW e GS)).

¹Disponível na URL: <https://physionet.org/pn4/eegmidb>. Acessado em 10/06/2016.

Class `FastFourierTransformer`

public class `FastFourierTransformer` ¹

extends `Object`

implements `Serializable`

Implements the Fast Fourier Transform for transformation of one-dimensional real or complex data sets. For reference, see Applied Numerical Linear Algebra, ISBN 0898713897, chapter 6.

There are several variants of the discrete Fourier transform, with various normalization conventions, which are specified by the parameter `DftNormalization`.

The current implementation of the discrete Fourier transform as a fast Fourier transform requires the length of the data set to be a power of 2. This greatly simplifies and speeds up the code. Users can pad the data with zeros to meet this requirement. There are other flavors of FFT, for reference, see S. Winograd, On computing the discrete Fourier transform, Mathematics of Computation, 32 (1978), 175 - 199.

Since: 1.2

See Also: `DftNormalization`, Serialized Form

B.1 Constructor Summary

`FastFourierTransformer(DftNormalization normalization)`

Creates a new instance of this class, with various normalization conventions.

Parameters: `normalization` - the type of normalization to be applied to the transformed data

¹Disponível na URL: <http://commons.apache.org/proper/commons-math/apidocs/org/apache/commons/math4/transform/>
Acessado em 28/09/2015.

B.2 Method Summary

B.2.1 transformInPlace

```
public static void transformInPlace(double[][] dataRI, DftNormalization normalization,
TransformType type)
```

Computes the standard transform of the specified complex data. The computation is done in place. The input data is laid out as follows

`dataRI[0][i]` is the real part of the *i*-th data point, `dataRI[1][i]` is the imaginary part of the *i*-th data point.

Parameters: `dataRI` - the two dimensional array of real and imaginary parts of the data
`normalization` - the normalization to be applied to the transformed data
`type` - the type of transform (forward, inverse) to be performed

Throws: `DimensionMismatchException` - if the number of rows of the specified array is not two, or the array is not rectangular
`MathIllegalArgumentException` - if the number of data points is not a power of two

B.2.2 transform

```
public Complex[] transform(double[] f, TransformType type)
```

Returns the (forward, inverse) transform of the specified real data set.

Parameters: `f` - the real data array to be transformed
`type` - the type of transform (forward, inverse) to be performed

Returns: the complex transformed array

Throws: `MathIllegalArgumentException` - if the length of the data array is not a power of two

B.2.3 *transform*

```
public Complex[] transform(UnivariateFunction f, double min, double max, int n, TransformType type)
```

Returns the (forward, inverse) transform of the specified real function, sampled on the specified interval.

Parameters: f - the function to be sampled and transformed min - the (inclusive) lower bound for the interval max - the (exclusive) upper bound for the interval n - the number of sample points type - the type of transform (forward, inverse) to be performed

Returns: the complex transformed array

Throws: `NumberIsTooLargeException` - if the lower bound is greater than, or equal to the upper bound `NotStrictlyPositiveException` - if the number of sample points n is negative `MathIllegalArgumentException` - if the number of sample points n is not a power of two

B.2.4 *transform*

```
public Complex[] transform(Complex[] f, TransformType type)
```

Returns the (forward, inverse) transform of the specified complex data set.

Parameters: f - the complex data array to be transformed type - the type of transform (forward, inverse) to be performed

Returns: the complex transformed array

Throws: `MathIllegalArgumentException` - if the length of the data array is not a power of two

Class MultilayerPerceptron

public class **MultilayerPerceptron**¹

extends Classifier

implements OptionHandler, WeightedInstancesHandler, Randomizable

A Classifier that uses backpropagation to classify instances.

This network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time. The nodes in this network are all sigmoid (except for when the class is numeric in which case the the output nodes become unthresholded linear units).

Valid options are:

-L (learning rate) Learning Rate for the backpropagation algorithm. (Value should be between 0 - 1, Default = 0.3).

-M (momentum) Momentum Rate for the backpropagation algorithm. (Value should be between 0 - 1, Default = 0.2).

-N (number of epochs) Number of epochs to train through. (Default = 500).

-V (percentage size of validation set) Percentage size of validation set to use to terminate training (if this is non zero it can pre-empt num of epochs. (Value should be between 0 - 100, Default = 0).

-S (seed) The value used to seed the random number generator (Value should be ≥ 0 and a long, Default = 0).

-E (threshold for number of consecutive errors) The consecutive number of errors allowed for validation testing before the network terminates. (Value should be >0 , Default = 20).

¹Disponível na URL: <http://weka.sourceforge.net/doc.stable/weka/classifiers/functions/MultilayerPerceptron.html>. Acessado em 02/02/2016.

- G GUI will be opened. (Use this to bring up a GUI).

- A Autocreation of the network connections will NOT be done. (This will be ignored if -G is NOT set)

- B A NominalToBinary filter will NOT automatically be used. (Set this to not use a NominalToBinary filter).

- H (comma separated numbers for nodes on each layer) The hidden layers to be created for the network. (Value should be a list of comma separated Natural numbers or the letters 'a' = (attribs + classes) / 2, 'i' = attribs, 'o' = classes, 't' = attribs .+ classes) for wildcard values, Default = a).

- C Normalizing a numeric class will NOT be done. (Set this to not normalize the class if it's numeric).

- I Normalizing the attributes will NOT be done. (Set this to not normalize the attributes).

- R Reseting the network will NOT be allowed. (Set this to not allow the network to reset).

- D Learning rate decay will occur. (Set this to cause the learning rate to decay).

Author: Malcolm Ware (mfw4@cs.waikato.ac.nz)

C.1 Constructor Summary

MultilayerPerceptron() The constructor.

C.2 Method Summary

C.2.1 main

```
public static void main(java.lang.String[] argv)
```

Main method for testing this class.

Parameters: argv - should contain command line options (see setOptions)

C.2.2 *setDecay*

```
public void setDecay(boolean d)
```

Parameters: d - True if the learning rate should decay.

C.2.3 *getDecay*

```
public boolean getDecay()
```

Returns: the flag for having the learning rate decay.

C.2.4 *setReset*

```
public void setReset(boolean r)
```

This sets the network up to be able to reset itself with the current settings and the learning rate at half of what it is currently. This will only happen if the network creates NaN or infinite errors. Also this will continue to happen until the network is trained properly. The learning rate will also get set back to it's original value at the end of this. This can only be set to true if the GUI is not brought up.

Parameters: r - True if the network should restart with it's current options and set the learning rate to half what it currently is.

C.2.5 *getReset*

```
public boolean getReset()
```

Returns: The flag for resetting the network.

C.2.6 *setNormalizeNumericClass*

```
public void setNormalizeNumericClass(boolean c)
```

Parameters: *c* - True if the class should be normalized (the class will only ever be normalized if it is numeric). (Normalization puts the range between -1 - 1).

C.2.7 getNormalizeNumericClass

```
public boolean getNormalizeNumericClass()
```

Returns: The flag for normalizing a numeric class.

C.2.8 setNormalizeAttributes

```
public void setNormalizeAttributes(boolean a)
```

Parameters: *a* - True if the attributes should be normalized (even nominal attributes will get normalized here) (range goes between -1 - 1).

C.2.9 getNormalizeAttributes

```
public boolean getNormalizeAttributes()
```

Returns: The flag for normalizing attributes.

C.2.10 setNominalToBinaryFilter

```
public void setNominalToBinaryFilter(boolean f)
```

Parameters: *f* - True if a nominalToBinary filter should be used on the data.

C.2.11 getNominalToBinaryFilter

```
public boolean getNominalToBinaryFilter()
```

Returns: The flag for nominal to binary filter use.

C.2.12 setSeed

```
public void setSeed(int l)
```

This seeds the random number generator, that is used when a random number is needed for the network.

Specified by: `setSeed` in interface `Randomizable`

Parameters: `l` - The seed.

C.2.13 getSeed

```
public int getSeed()
```

Description copied from interface: `Randomizable` Gets the seed for the random number generations

Specified by: `getSeed` in interface `Randomizable`

Returns: The seed for the random number generator.

C.2.14 setValidationThreshold

```
public void setValidationThreshold(int t)
```

This sets the threshold to use for when validation testing is being done. It works by ending testing once the error on the validation set has consecutively increased a certain number of times.

Parameters: `t` - The threshold to use for this.

C.2.15 getValidationThreshold

```
public int getValidationThreshold()
```

Returns: The threshold used for validation testing.

C.2.16 setLearningRate

```
public void setLearningRate(double l)
```

The learning rate can be set using this command. NOTE That this is a static variable so it affect all networks that are running. Must be greater than 0 and no more than 1.

Parameters: l - The New learning rate.

C.2.17 getLearningRate

```
public double getLearningRate()
```

Returns: The learning rate for the nodes.

C.2.18 setMomentum

```
public void setMomentum(double m)
```

The momentum can be set using this command. THE same conditions apply to this as to the learning rate.

Parameters: m - The new Momentum.

C.2.19 getMomentum

```
public double getMomentum()
```

Returns: The momentum for the nodes.

C.2.20 *setAutoBuild*

```
public void setAutoBuild(boolean a)
```

This will set whether the network is automatically built or if it is left up to the user. (there is nothing to stop a user from altering an autobuilt network however).

Parameters: a - True if the network should be auto built.

C.2.21 *getAutoBuild*

```
public boolean getAutoBuild()
```

Returns: The auto build state.

C.2.22 *setHiddenLayers*

```
public void setHiddenLayers(java.lang.String h)
```

This will set what the hidden layers are made up of when auto build is enabled. Note to have no hidden units, just put a single 0, Any more 0's will indicate that the string is badly formed and make it unaccepted. Negative numbers, and floats will do the same. There are also some wildcards. These are 'a' = (number of attributes + number of classes) / 2, 'i' = number of attributes, 'o' = number of classes, and 't' = number of attributes + number of classes.

Parameters: h - A string with a comma seperated list of numbers. Each number is the number of nodes to be on a hidden layer.

C.2.23 *getHiddenLayers*

```
public java.lang.String getHiddenLayers()
```

Returns: A string representing the hidden layers, each number is the number of nodes on a hidden layer.

C.2.24 *setGUI*

```
public void setGUI(boolean a)
```

This will set whether A GUI is brought up to allow interaction by the user with the neural network during training.

Parameters: a - True if gui should be created.

C.2.25 *getGUI*

```
public boolean getGUI()
```

Returns: The true if should show gui.

C.2.26 *setValidationSetSize*

```
public void setValidationSetSize(int a)
```

This will set the size of the validation set.

Parameters: a - The size of the validation set, as a percentage of the whole.

C.2.27 *getValidationSetSize*

```
public int getValidationSetSize()
```

Returns: The percentage size of the validation set.

C.2.28 *setTrainingTime*

```
public void setTrainingTime(int n)
```

Set the number of training epochs to perform. Must be greater than 0.

Parameters: n - The number of epochs to train through.

C.2.29 getTrainingTime

```
public int getTrainingTime()
```

Returns: The number of epochs to train through.

C.2.30 blocker

```
public void blocker(boolean tf)
```

A function used to stop the code that called buildclassifier from continuing on before the user has finished the decision tree.

Parameters: tf - True to stop the thread, False to release the thread that is waiting there (if one).

C.2.31 getCapabilities

```
public Capabilities getCapabilities()
```

Returns default capabilities of the classifier.

Specified by: getCapabilities in interface CapabilitiesHandler

Overrides: getCapabilities in class Classifier

Returns: the capabilities of this classifier

See Also: Capabilities

C.2.32 buildClassifier

```
public void buildClassifier(Instances i) throws java.lang.Exception
```

Call this function to build and train a neural network for the training data provided.

Specified by: `buildClassifier` in class `Classifier`

Parameters: `i` - The training data.

Throws: `java.lang.Exception` - if can't build classification properly.

C.2.33 distributionForInstance

```
public double[] distributionForInstance(Instance i) throws java.lang.Exception
```

Call this function to predict the class of an instance once a classification model has been built with the `buildClassifier` call.

Overrides: `distributionForInstance` in class `Classifier`

Parameters: `i` - The instance to classify.

Returns: A double array filled with the probabilities of each class type.

Throws: `java.lang.Exception` - if can't classify instance.

C.2.34 listOptions

```
public java.util.Enumeration listOptions()
```

Returns an enumeration describing the available options.

Specified by: `listOptions` in interface `OptionHandler`

Overrides: `listOptions` in class `Classifier`

Returns: an enumeration of all the available options.

C.2.35 *getOptions*

```
public java.lang.String[] getOptions()
```

Gets the current settings of NeuralNet.

Specified by: `getOptions` in interface `OptionHandler`

Overrides: `getOptions` in class `Classifier`

Returns: an array of strings suitable for passing to `setOptions()`

C.2.36 *toString*

```
public java.lang.String toString()
```

Overrides: `toString` in class `java.lang.Object`

Returns: string describing the model.

C.2.37 *globalInfo*

```
public java.lang.String globalInfo()
```

This will return a string describing the classifier.

Returns: The string.

C.2.38 *learningRateTipText*

```
public java.lang.String learningRateTipText()
```

Returns: a string to describe the learning rate option.

C.2.39 momentumTipText

```
public java.lang.String momentumTipText()
```

Returns: a string to describe the momentum option.

C.2.40 autoBuildTipText

```
public java.lang.String autoBuildTipText()
```

Returns: a string to describe the AutoBuild option.

C.2.41 seedTipText

```
public java.lang.String seedTipText()
```

Returns: a string to describe the random seed option.

C.2.42 validationThresholdTipText

```
public java.lang.String validationThresholdTipText()
```

Returns: a string to describe the validation threshold option.

C.2.43 trainingTimeTipText

```
public java.lang.String trainingTimeTipText()
```

Returns: a string to describe the learning rate option.

C.2.44 nominalToBinaryFilterTipText

```
public java.lang.String nominalToBinaryFilterTipText()
```

Returns: a string to describe the nominal to binary option.

C.2.45 hiddenLayersTipText

```
public java.lang.String hiddenLayersTipText()
```

Returns: a string to describe the hidden layers in the network.

Trabalhos Publicados

Capítulos de livros:

PINHEIRO, O. R.; Josemar Rodrigues de Souza. Accesibilidad y usuabilidad en la TDI. SIMULADOR DE CADEIRA DE RODAS INTELIGENTE PARA PESSOAS COM MOBILIDADE REDUZIDA. 1ed.Córdoba (España): CITEC Universidad de Crdoba, 2014, v. 1, p. 140-144, ISBN 978-84-697-0302-1.

Artigos publicados em anis de congressos:

R. M. F. MOURA JNIOR; J. R. SOUZA; O. R. PINHEIRO. CADEIRA DE RODAS INTELIGENTE: VENCENDO A TERCEIRA IDADE COM JOGOS SRIOS, JAPEX 2016.

Oberdan R. Pinheiro; Lynn R. G. Alves; Romero M. F. M. Jnior; Josemar R. de Souza. Wheelchair simulator game for training people with severe disabilities, International Conference on Technology and Innovation in Sports, Health and Wellbeing, 2016 - UTAD, Vila Real, Portugal. TISHW 2016.

SOUZA, J. R. de; MERCS, R. S. M. R. C.; PINHEIRO, O. R.; M. A. C. S. Bahiart@home 2016 team description paper. In: ROBOCUP FEDERATION. Proceedings of RoboCup 2015. Hefei, China, 2015.

Romero Moura Júnior; Ramon Mercês; Oberdan Pinheiro; Josemar Rodrigues. *CRI Project: Intelligent Wheelchair*, BRASERO 2015.

R. B. S; T. A. C; PINHEIRO, O. R; Josemar Rodrigues de Souza. MULTIPLATAFORMA DE CONTROLE DE DIREÇÃO PARA O PROJETO SMARTCHAIR. PTI 2014.

Artigos submetidos para revistas qualificadas:

Oberdan R. Pinheiro; Lynn R. G. Alves; Josemar R. de Souza, Interface cérebro-computador: imaginação de movimentos para treinamento de pessoas com graves deficiências motoras, RENOTE, 2016, ISSN 1679-1916.

Oberdan R. Pinheiro; Lynn R. G. Alves; Josemar R. de Souza, *EEG Signals Classification: Motor Imagery for Driving an Intelligent Wheelchair*, IEEE Amrica Latina, 2016, ISSN 1548-0992.

SmartChair: Cadeira de Rodas Inteligente com Interface Flexível

Oberdan Rocha Pinheiro

Salvador, Dezembro de 2016.