

UNIVERSIDADE FEDERAL DO PARANÁ

ALISSON RAVAGLIO SANTOS

**ABORDAGENS DE APRENDIZADO DE MÁQUINA APLICADAS À
CLASSIFICAÇÃO DE DADOS PROVENIENTES DE EQUIPAMENTO
ELETROENCEFALOGRÁFICO NÃO-INVASIVO**

CURITIBA

2017

ALISSON RAVAGLIO SANTOS

ABORDAGENS DE APRENDIZADO DE MÁQUINA APLICADAS À
CLASSIFICAÇÃO DE DADOS PROVENIENTES DE EQUIPAMENTO
ELETROENCEFALOGRÁFICO NÃO-INVASIVO

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Área de Concentração em sistemas eletrônicos, Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como parte das exigências para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Leandro dos Santos Coelho.

CURITIBA

2017

S237a

Santos , Alisson Ravaglio

Abordagens de aprendizado de máquina aplicadas à classificação de dados provenientes de equipamento eletroencefalográfico não-invasivo / Alisson Ravaglio Santos . – Curitiba, 2017.
123 f. : il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, 2017.

Orientador: Leandro dos Santos Coelho .
Bibliografia: p. 109-117.

1. Interface cérebro-computador. 2. Aprendizado de Máquinas. 3. Pêndulo. I. Universidade Federal do Paraná. II. Coelho, Leandro dos Santos. III. Título.

CDD: 003.5



UNIVERSIDADE FEDERAL DO PARANÁ
Programa de Pós-Graduação em Engenharia Elétrica, PPGE
Setor de Tecnologia

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de ALISSON RAVAGLIO SANTOS intitulada: Abordagens de aprendizado de máquina aplicadas à classificação de dados provenientes de equipamento eletroencefalográfico não-invasivo, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa. A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 01 de Dezembro de 2017.

LEANDRO DOS SANTOS COELHO
Presidente da Banca Examinadora (UFPR)

JULIO CÉSAR NIEVOLA
Avaliador Externo (PUC/PR)

MARLIO JOSÉ DO COUTO BONFIM
Avaliador Interno (UFPR)

Dedico este trabalho a minha
família e a minha empresa
YOUNIC Solutions

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus por tudo que tem feito em minha vida. Ao Professor Leandro dos Santos Coelho, pela orientação, empenho, apoio, confiança e pela oportunidade por me permitir elaborar este trabalho. Também agradeço ao professor Gilberto Reynozo-Meza pelo suporte complementar e a todos os *professores* por me proporcionarem o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

A UFPR, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior, com confiança e ética aqui presentes.

Aos meus pais, Willian da Silva Santos e Tania Mara Ravaglio, por terem me dado toda base, educação e amor necessário para chegar até aqui. Às minhas avós Cleuza da Silva Santos e Rose Mari Ravaglio e minha tia Daniele Ravaglio por terem me ensinado valores desde criança e aos familiares restantes por terem contribuído em partes para a minha educação.

À minha namorada Renata Duarte Crozeta, pelo amor, companheirismo, dedicação, incentivo e carinho que me deu durante todo o mestrado, me dando suporte e motivação nos momentos que eu mais precisava.

Aos meus amigos, mas em especial para meu sócio e amigo Felipe Hartcopp Betoni, pela ajuda técnica e revisão de todo documento do mestrado, na YOUNIC Solutions e pela força de vontade e lealdade que compartilha comigo. Também agradeço ao Gabriel Becchi, o qual co-orientei no seu projeto de iniciação científica por um ano, por toda a disposição e ajuda prestada.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

RESUMO

A interface cérebro-computador (ICC, do inglês *brain-computer interfaces*, BCIs) tem emergido como uma área promissora de pesquisa. As ICCs podem possibilitar aos seus usuários a capacidade de comunicação e controle que não dependem de músculos. Esta dissertação visa utilizar um eletroencefalograma não-invasivo para, por meio da atividade cerebral das pessoas, controlar um processo denominado pêndulo, utilizando-se de uma abordagem de comunicação entre uma pessoa e uma máquina (interface cérebro-computador) por meio de classificações realizadas por algoritmos de Aprendizado de máquina (*Machine Learning*). Desta forma, o objetivo principal é desenvolver algoritmos de aprendizado de máquina capazes de classificar dados de um eletroencefalograma para controle de um processo pêndulo. Os algoritmos que foram utilizados na classificação dos sinais são: (i) máquina de aprendizado extremo, (ii) K vizinhos mais próximos, (iii) árvore de decisão, (iv) máquina de vetores de suporte, (v) máquina de vetores de relevância e (vi) XgBoost. Neste contexto, por meio de dados coletados de eletroencefalograma de uma pessoa e com a utilização dos algoritmos de Aprendizado de Máquina que classificam os sinais recebidos, tem-se como objetivo determinar o mínimo de erro possível (de acordo com a acurácia da matriz de confusão) entre as saídas desejadas e as saídas obtidas e, em seguida, gerar comandos para o controle de sistema de pêndulo onde será utilizado um controlador PID (Proporcional, Integral e Derivativo) adaptativo de Dahlin. Os resultados apresentados obtiveram valores de erros baixos, tanto para a acurácia matriz de confusão dos algoritmos quanto para estabelecer o controle do processo do pêndulo, com uma resposta satisfatória, de modo que o erro de saída do controlador, ITAE (*Integral of Time Multiplied by the Absolute Value of Error* - Integral do erro absoluto vezes o tempo), tenha resultado em um valor baixo com um bom controle do processo. Esta abordagem pode ser promissora e irá servir como base para projetos de sistemas que visam ajudar pessoas com limitações vinculadas de coordenação motora, tais como em casos de pessoas tetraplégicas ou quadriplégicas, que possuem paralisia em todas as quatro extremidades, superiores e inferiores, juntamente à musculatura do tronco, mas que possuem atividade cerebral. Por fim, este trabalho de dissertação visa contribuir com o processo de aquisição dos dados do eletroencefalograma e a utilização de algoritmos de aprendizado de máquina para a classificação destes dados, expondo os melhores algoritmos e padrões a serem utilizados no controle de um processo.

Palavras-chave: Interface cérebro-computador. Aprendizado de Máquina. Pêndulo.

ABSTRACT

The brain-computer interface (BCIs) has emerged as a promising area of research. The ICCs can give their users the ability to communicate and control without depending on muscles. Thus, this work aims to use an electroencephalogram to control a suspended pendulum by means of the brain activity of the people, an approach to a system of communication between a person and a machine (computer-brain interface) through algorithm classifications of machine learning. In this way, the main objective is to develop machine learning algorithms capable of classifying data of an electroencephalogram to control an overdamped pendulum process. The algorithms that were used to classify the signals are: (i) extreme learning machine, (ii) K nearest neighbors, (iii) decision tree, (iv) support vector machine, (v) relevance vector machine and (vi) XgBoost. In this context, through data collected from a person's electroencephalogram and using the Machine Learning algorithms to classify the received signals, the main objective is to determine the minimum possible error (according to the confusion matrix) between the desired outputs and the outputs obtained, and then generate commands for the overdamped pendulum system control that will be used a Dahlin Adaptive, Proportional, Integral and Derivative (PID) controller. The results obtained low error values, both for the confusion matrix of the algorithms and to establish the process control of the overdamped pendulum, with a satisfactory response, so that the controller output error, ITAE (Integral of Time Multiplied by the Absolute Value of Error - Integral of the absolute error times times), has resulted in a low value with a good process control for overdamped pendulum. This approach may be promising and will serve as the basis for projects aimed at helping people with related limitations of motor skills, such as in cases of tetraplegic or quadriplegic people who have paralysis in all four extremities, upper and lower, along the muscles of the trunk, but which have brain activity. Finally, this work aims to contribute with the process of electroencephalogram data acquisition and use machine learning algorithms to classify the electroencephalogram data, exposing the best algorithms and standards in systems without control of a process.

Key words: Brain-computer interface. Machine Learning. Overdamped Pendulum.

LISTA DE ILUSTRAÇÕES

Figura 1 – Fluxo geral dos módulos do trabalho	14
Figura 2 – Fundamentação teórica tratada em cada módulo do projeto	27
Figura 3 - Ilustração de um neurônio e suas sinapses	28
Figura 4 - Posicionamento de eletrodos dos sistemas 10-20 para medição EEG.....	29
Figura 5 - As 5 principais frequências e suas relações	30
Figura 6 - Mapa funcional do cérebro humano.....	32
Figura 7 - Fluxo do processo ICC para controle do pêndulo	34
Figura 8 - Modelo de aprendizado de máquina	35
Figura 9 - Estrutura de um neurônio biológico.....	38
Figura 10 - Modelo matemático de um neurônio	39
Figura 11 - Redes feedforward com camada única.....	41
Figura 12 - Redes feedforward com múltiplas camadas	42
Figura 13 - Funcionamento de aprendizagem supervisionada.....	43
Figura 14 - Modelo de aprendizagem por back-propagation.....	44
Figura 15 - Representação do método de aprendizado baseado em instâncias.....	49
Figura 16 - modelo de funcionamento do algoritmo K vizinho mais próximo	50
Figura 17 - Modelo de árvore de decisão para jogo de tênis	53
Figura 18 – Hiperplano de separação ótima (na cor vermelha), com os hiperplanos suporte (tracejados)	55
Figura 19 - Comparação de um modelo SVM com um modelo RVM.....	59
Figura 20 - Gráfico da posição angular em função do tempo para um pêndulo físico por um momento de atrito constante.....	61
Figura 21 - Comparativo custo x benefício de eletroencefalogramas.....	66
Figura 22 - Dispositivo EMOTIV EPOC.....	67
Figura 23 - Posicionamento dos eletrodos do EMOTIV	68
Figura 24 - Protótipo do pêndulo	69
Figura 25 - Arduino Mega 2560.....	70
Figura 26 – Gráfico de violino plotado na ferramenta MATLAB dos sensores do EMOTIV.....	75
Figura 27 - Amostra do sinal do EEG sem suavização	77
Figura 28 - Amostra do sinal do EEG com suavização	78

Figura 29 – Gráfico com valores de saída do EEG	80
Figura 30 - Matriz de confusão para o algoritmo ELM.....	83
Figura 31 - Matriz de confusão para o algoritmo K-NN	84
Figura 32 - Matriz de confusão para o algoritmo árvore de decisão.....	84
Figura 33 - Matriz de confusão para o algoritmo SVM	85
Figura 34 - Matriz de confusão para o algoritmo RVM	85
Figura 35 - Matriz de confusão para o algoritmo XgBoost	86
Figura 36 - Sinal de resposta e sinal de referência do controlador para $\tau = 0,3$	90
Figura 37 - Sinal de controle para $\tau = 0,3$	91
Figura 38 - Sinal de erro do controlador para $\tau = 0,3$	92
Figura 39 - Sinal de resposta e sinal de referência do controlador para $\tau = 0,4$	93
Figura 40 - Sinal de erro do controlador para $\tau = 0,4$	94
Figura 41 - Sinal de controle para $\tau = 0,4$	95
Figura 42 - Sinal de resposta e de referência para $\tau = 50$	96
Figura 43 - Sinal de controle para $\tau = 50$	97
Figura 44 - Sinal de erro do controlador para $\tau = 50$	97
Figura 45 - Sinal do EEG com padrão de pensamento para desligar pêndulo.....	98
Figura 46 - Sinal do EEG com padrão de pensamento para mover pêndulo em 45°	100
Figura 47 - Sinal de resposta e de referência para controle do pêndulo em 45°	102
Figura 48 - Sinal $u(t)$ de controle aplicado para movimento do motor com ângulo de 45°	103
Figura 49 - Sinal do EEG com padrão de pensamento para mover pêndulo em 48°	104
Figura 50 - Sinal de resposta e de referência para controle do pêndulo em 48°	105
Figura 51 - Sinal $u(t)$ de controle aplicado para movimento do motor com ângulo de 48°	106
Figura 52 - Esquema elétrico de ligação do pêndulo	123

LISTA DE TABELAS

Tabela 1 - Dados capturados utilizando o EMOTIV com MATLAB	73
Tabela 2 - Intervalo de valores utilizados para a coleta de dados.....	76
Tabela 3 - Medidas estatísticas obtidas do sinal do EEG	81
Tabela 4 - Resultados da acurácia da matriz de confusão dos algoritmos	82
Tabela 5 - Valores de erro do controlar PID adaptativo de Dahlin	89
Tabela 6 - Valores de média, mediana e variância para os sinais coletados para o pensamento de desligar o pêndulo	99
Tabela 7 - Valores de média, mediana e variância para os sinais coletados para o pensamento de ligar o pêndulo em 45°	101
Tabela 8 - Valores de média, mediana e variância para os sinais coletados para o pensamento de ligar o pêndulo em 48°	104

LISTA DE SIGLAS

EEG	Eletroencefalografia
ELM	<i>Extreme Learning Machine</i>
EMOTIV	<i>EMOTIV® EPOC Headset</i>
IA	Inteligência Artificial
ICC	Interface Cérebro-Computador
K-NN	<i>K-nearest neighbors</i>
MATLAB	Software MATLAB®
PUCPR	Pontifícia Universidade Católica do Paraná
RNA	Rede Neural Artificial
RVM	<i>Relevance vector machine</i>
SIMULINK	Software SIMULINK®
SVM	<i>Support vector machine</i>
<i>toolbox</i>	Ferramenta desenvolvida para o MATLAB
<i>Workspace</i>	Local onde os dados ficam armazenados no MATLAB

SUMÁRIO

1	INTRODUÇÃO	12
1.1	CONTEXTO	14
1.2	OBJETIVOS	14
1.2.1	Objetivo geral.....	15
1.2.2	Objetivos específicos.....	15
1.3	JUSTIFICATIVA	15
1.4	CONTRIBUIÇÃO DESTA DISSERTAÇÃO	16
1.5	ESTRUTURA DA DISSERTAÇÃO.....	16
2	REVISÃO DA LITERATURA.....	18
2.1	Adaptive Training Using an Artificial Neural Network and EEG Metrics for Within- and Cross-Task Workload Classification	18
2.2	EMOTIV Epop Servo Motor Control	19
3	FUNDAMENTAÇÃO TEÓRICA.....	27
3.1	Tecnologia EEG: Como a Eletroencefalografia Funciona	27
3.1.1	Introdução ao EEG e como a Atividade do Cérebro pode ser medida ..	28
3.1.2	Identificando Diferentes Padrões de Atividade Cerebral	29
3.1.3	Aplicação Prática da Tecnologia EEG	31
3.2	Interface Cérebro-Computador	33
3.2.1	Cinco Fatores Centrais de um Projeto ICC.....	34
3.3	Fundamentos de Aprendizado de Máquina.....	35
3.4	Redes Neurais Artificiais	36
3.4.1	Modelo Artificial de Neurônios	37
3.4.2	Arquitetura de uma Rede Neural Artificial.....	39
3.4.3	O Mapa de Aprendizado	42
3.4.4	Máquina de Aprendizado Extremo.....	46
3.5	Aprendizado Baseado em Instâncias	47

3.5.1	K Vizinhos mais próximos (K-NN).....	49
3.6	Árvore de Decisão.....	51
3.7	XgBoost.....	53
3.8	Máquina de vetor de suporte (SVM)	54
3.9	Máquina de vetor de relevância	56
3.10	Processo Pêndulo.....	59
3.11	Controlador PID Dahlin Adaptativo	61
4	MATERIAIS E MÉTODOS	66
4.1	MATERIAIS.....	66
4.1.1	Neuroheadset	66
4.1.1.1	Posicionamento de Eletrodos.....	67
4.1.1.2	EMOTIV Epoc Headset	67
4.1.2	Pêndulo.....	69
4.1.2.1	Arduino Mega 2560	70
4.1.3	Configurações do computador utilizado	71
4.2	MÉTODOS.....	72
4.2.1	Aquisições dos Sinais Eletroencefálicos	72
4.2.2	Validação dos Dados	74
4.2.3	Classificação dos Dados	76
4.2.4	Controle do Pêndulo.....	78
4.2.5	Integração Final.....	79
5	ANÁLISE DOS RESULTADOS E DISCUSSÃO.....	80
5.1	Classificação dos dados.....	82
5.1.1	Matriz de confusão das melhores acurácias obtidas.....	83
5.1.2	Parâmetros e ajustes utilizados	86
5.2	Controle do pêndulo.....	88
5.3	Integração entre todos os módulos do projeto	98

6	CONCLUSÕES E TRABALHOS FUTUROS	107
	REFERÊNCIAS	109
	APÊNDICE A – Outras coletas de Dados Utilizando o EMOTIV	118
	APÊNDICE B – Algoritmo para controle do pêndulo	120
	APÊNDICE C – Esquema elétrico geral para ligação do pêndulo	123

1 INTRODUÇÃO

O cérebro humano é particularmente complexo e extenso. Compreender como ele funciona, isto é, compreender a interação de suas moléculas, células, circuitos e os sistemas que permitem o comportamento, a percepção, o pensamento e a emoção, é um objetivo relevante da neurociência (estudo do sistema nervoso). Esta meta (Cascio e Falk, 2016) permanece em estudo e diversos pesquisadores tem realizado novas descobertas e percepções sobre o tema (Ramadan e Vasilakos, 2017), embora ainda existam algumas limitações das tecnologias disponíveis atualmente.

Ao longo da última década, no entanto, notáveis avanços tecnológicos têm criado novas possibilidades para o estudo e a compreensão do cérebro. Isso faz com que não só os pesquisadores da área médica tenham interesse na área como também os pesquisadores da área de tecnologia e computação (Roy et al., 2016; Kropotov, 2009). Uma das tecnologias mais conhecidas é a eletroencefalografia (*EEG*, do inglês *Electroencephalography*), onde um dispositivo capta correntes elétricas desenvolvidas no encéfalo, por meio de eletrodos posicionados no couro cabeludo (Cooper et al., 2014).

Existem algumas aplicações para este tipo de pesquisa. Um exemplo, em Waltz (2012), é uma aplicação que se resume em uma forma de ajudar na identificação de crianças com autismo. Luauté et al. (2015) mostram a potencialidade da tecnologia ICC (interface cérebro computador) em fornecer ao usuário a capacidade de comunicação através do pensamento, pedindo ajuda, alimento, ou qualquer outra coisa que ele não seria capaz de afirmar sem o auxílio de algum tipo de interpretador. Outros exemplos são apresentados em Peck (2013) e Li et al. (2015).

As ICCs podem possibilitar aos seus usuários capacidades de comunicação e controle que não dependem de músculos. A constante melhoria das técnicas de registro da atividade cerebral fornece sinais com crescentes relações sinal-ruído e o aumento da capacidade computacional permite o uso de técnicas de processamento de sinal complexas em tempo real.

Além disso, no estudo recente de Ramadan e Vasilakos (2017), constatou-se que o estudo de ICCs pode ajudar pessoas com lesões neuromusculares ou doenças neurodegenerativas como a esclerose lateral amiotrófica (ELA, do inglês

amyotrophic lateral sclerosis), que afeta as células nervosas no cérebro e na espinha, mas deixa a parte cognitiva intacta, o que torna totalmente possível a aplicação de projetos com *EEGs*.

O *EMOTIV*, EEG escolhido para ser utilizado neste trabalho, apresenta 14 eletrodos capazes de extrair informações cognitivas de diferentes regiões do cérebro e 2 eletrodos que capturam a referência da posição com uma frequência de amostragem de 128 Hz. Desta forma, os dados são coletados a cada 7,81 ms e torna-se possível a realização do estudo proposto por este trabalho de dissertação.

Para o desenvolvimento desta dissertação foi necessário o entendimento do funcionamento do *Neuroheadset* que será utilizado na aquisição dos dados, além da implementação e adaptação de algoritmos para fazer a classificação dos dados captados, como por exemplo, uma abordagem de rede neural artificial. A classificação dos dados é uma subárea de aprendizado de máquina e, nesta dissertação, são implementados e adaptados modelos de algoritmos de classificação baseado em rede neural artificial, árvore de decisão, máquina de vetores de suporte e aprendizado por instâncias. Estes algoritmos foram escolhidos pelo fato destes serem bastante estudados (Bishop et al., 2006; Lotte et al., 2007) quando se trata de reconhecimento de padrões. Estes modelos de algoritmo de classificação são adaptativos.

Sendo assim, pode-se resumir este projeto em três partes, das quais podem ser vistas na Figura 1, conforme seguinte sequência: (i) o processo de aquisição de sinal *EEG*, (ii) utilização dos dados obtidos e aplicação de diversas abordagens de Inteligência Artificial, dentro da subárea de Aprendizado de máquina, para reconhecimento de padrões, e (iii) utilização da classificação de sinais para controlar um processo (sistema dinâmico). O método de solução no projeto usa o *neuroheadset EMOTIV* para a Parte 1, algoritmos de Aprendizado de máquina incluindo Máquina de Aprendizado Extremo, K vizinho(s) mais próximo(s), Árvore de Decisão, Máquina de Vetor de Suporte, Máquina de Vetor de Relevância e XgBoost para classificar os padrões de ondas cerebrais na Parte 2 e um processo pêndulo (Santos et al., 2017) a ser controlado na Parte 3 para demonstrar o funcionamento e eficiência da classificação dos algoritmos utilizados.

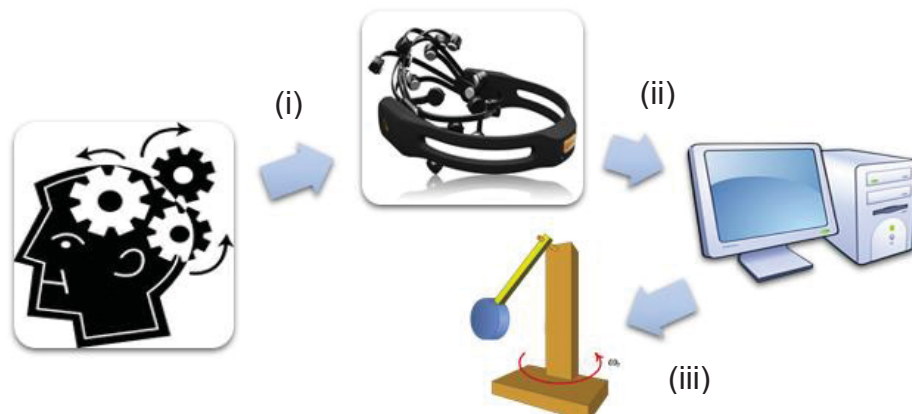


Figura 1 – Fluxo geral dos módulos do trabalho
 FONTE: O autor (2017)

1.1 CONTEXTO

Recentemente, os avanços da tecnologia e eletrônica e o aparecimento de sensores e de microcontroladores miniaturizados, permitiram a construção de equipamentos sofisticados para a aquisição de dados do cérebro, sendo estes capazes de extrair dados em tempo real das informações trocadas entre neurônios.

Logo, são utilizados dados cerebrais das pessoas de forma a controlar um processo pêndulo. Uma nova abordagem utilizando algoritmos de aprendizado de máquina e para um sistema de comunicação entre um pêndulo e uma pessoa por interface cérebro-máquina é adotada.

Para essa nova abordagem são utilizados algoritmos de aprendizado de máquina, onde são divididos em duas etapas: treinamento (*off-line*) e classificação (*on-line*). Na fase de treinamento são gerados os classificadores, baseado no processo de aprendizagem do algoritmo, de todos os algoritmos para poder utilizá-los na etapa seguinte (classificação). Na etapa de classificação os algoritmos recebem os dados cerebrais, estes dados são classificados e, então, gera-se os dados de saída que servem como entrada para o controle do pêndulo.

1.2 OBJETIVOS

A seguir são listados o objetivo geral e os objetivos específicos deste trabalho de dissertação.

1.2.1 Objetivo geral

Coletar e classificar, utilizando abordagens de aprendizado de máquina, os sinais de um EEG de forma a controlar um processo pêndulo em diferentes ângulos.

1.2.2 Objetivos específicos

Os objetivos específicos são:

- a) Revisão sobre as tecnologias de eletroencefalogramas não-invasivas disponíveis;
- b) Análisar o Eletroencefalograma escolhido;
- c) Capturar dados do cérebro utilizando o *EMOTIV* (EEG escolhido);
- d) Desenvolver algoritmo para aquisição de dados em ambiente computacional MATLAB, da *MathWorks*;
- e) Estudar os algoritmos de Aprendizado de Máquina para a classificação dos sinais coletados, incluindo Máquina de Aprendizado Extremo, Árvore de decisão, K Vizinho mais próximo, Máquina de Vetor de Suporte, Máquina de Vetor de Relevância e XgBoost;
- f) Implementar os algoritmos de Inteligência Artificial escolhidos (Máquina de Aprendizado Extremo, Árvore de decisão, K Vizinho mais próximo, Máquina de Vetor de Suporte, Máquina de Vetor de Relevância e XgBoost);
- g) Avaliar os melhores algoritmos, em termos da acurácia da matriz de confusão, para a classificação dos dados de eletroencefalograma;
- h) Aplicar os algoritmos para controle de um pêndulo;
- i) Obter a melhor modelagem de controle para o processo do pêndulo;
- j) Integrar todos os módulos e obter um controle em tempo real do pêndulo.

1.3 JUSTIFICATIVA

As ICCs têm se mostrado cada vez mais uma área promissora de pesquisa (Kryger et al., 2017; Choon Guan Lim et al., 2010). Elas podem possibilitar aos seus usuários capacidades de comunicação e controle que não dependem de músculos.

Além das ICCs, as técnicas de aprendizado de máquina têm se mostrado uma excelente ferramenta para entender o comportamento humano. O processamento das informações obtidas do cérebro, utilizando algoritmos de aprendizado de máquina, consegue atingir um resultado que é capaz de interpretar o pensamento humano sem nenhum erro, mostrando que é possível interpretar o pensamento humano e, com isso, aplicar em ações que possam facilitar as atividades humanas.

1.4 CONTRIBUIÇÃO DESTA DISSERTAÇÃO

Uma das contribuições desta dissertação é o algoritmo para o processo de aquisição dos dados do eletroencefalograma. Inicialmente, foi utilizado um modelo em SIMULINK, software integrado ao MATLAB. Entretanto, este era um modelo de um software pronto e funcionava apenas em uma versão específica para a coleta de dados. Sendo assim, foi desenvolvido um novo código, em C++ com integração no MATLAB, que captura o sinal das bibliotecas do *EMOTIV (.dlls)*. Desta forma, o processo de aquisição se tornou mais simples e a integração com os outros códigos do trabalho foi facilitada.

A segunda contribuição desta dissertação foi a utilização de algoritmos de aprendizado de máquina para o treinamento (*off-line*) e classificação (*on-line*) dos dados obtidos do *EEG*.

Também foram constatados os algoritmos de aprendizado de máquina que obtiveram os melhores valores de acurácia. Além disso, foi possível identificar os melhores padrões de pensamento, podendo assim tornarem-se mais fáceis de serem treinados e classificados.

Como última contribuição, pode-se citar o controle do pêndulo utilizando a saída dos algoritmos de aprendizado de máquina, onde foi possível estabelecer um excelente controle do pêndulo, juntamente com o controlador adaptativo utilizado.

1.5 ESTRUTURA DA DISSERTAÇÃO

O restante deste documento de dissertação está organizado da seguinte forma. O capítulo 2 detalha a fundamentação teórica, que expõe o conteúdo relacionado a este projeto referente ao EEG, aos algoritmos de Aprendizado de

máquina e ao controle do sistema denominado pêndulo. O capítulo 3 apresenta uma revisão da literatura importante para conhecer algumas aplicações já realizadas referentes ao estudo de ondas cerebrais aplicadas com algoritmos de inteligência artificial.

No capítulo 4 são fundamentados os materiais e métodos utilizados no trabalho em questão referentes aos módulos do EEG e do pêndulo. Nos métodos é apresentada toda a metodologia do trabalho para obtenção dos resultados mostrados.

O capítulo 5 mostra e analisa os resultados, baseados na acurácia da matriz de confusão, encontrados após processar os dados do EEG utilizando os algoritmos de aprendizado de máquina. Em seguida, no capítulo 6, tem-se a conclusão geral do trabalho, bem como o cronograma de atividades a serem desenvolvidas com intuito da escrita da dissertação para a defesa e as considerações sobre futuros trabalhos que possam ser desenvolvidos baseado nos resultados encontrados.

2 REVISÃO DA LITERATURA

Com a ideia de aplicar algoritmos de inteligência artificial em sinais eletroencefálicos, foram desenvolvidos diversos projetos. Além disso, foram pesquisados projetos que utilizassem as ondas cerebrais para o controle de um processo. Pode-se assim citar alguns deles que mais se destacaram pela eficiência e praticidade do seu uso. Para todos os projetos analisados citou-se seus aspectos positivos, bem como os possíveis aspectos negativos. Os projetos analisados estão descritos a seguir.

2.1 Adaptive Training Using an Artificial Neural Network and EEG Metrics for Within- and Cross-Task Workload Classification

O projeto em questão, desenvolvido por Penaranda e Baldwin (2012), visa reconhecer os padrões neurofisiológicos de pessoas que realizam tarefas de trabalho similares e diferentes. Para isso, foram utilizados algoritmos de Inteligência Artificial, mais especificamente um treino adaptativo baseado em redes neurais artificiais, utilizando um eletroencefalograma (EEG). Similar a este trabalho de dissertação, o intuito deste trabalho é que a Rede Neural identifique o nível de dificuldade de uma determinada tarefa em comparação com outras diferentes tarefas quando o operador não sabe realizar nenhuma das tarefas pretendidas. Além disso, é possível determinar a capacidade de aprendizagem e emoção cognitiva que envolve as atividades e, por este motivo, faz-se necessário a utilização de um treinamento adaptativo.

A utilização de um eletroencefalograma torna possível a extração de diversas características e padrões, possibilitando a aplicação para o cenário proposto por Baldwin e Penaranda (2012). Além disso, torna-se possível realizar a separação dos sinais eletroencefálicos, de forma a conseguir separar o tipo de característica da qual se deseja avaliar. Por exemplo, é possível separar os espectros do sinal do EEG em diversas faixas de frequências diferentes, ondas theta, por exemplo, das quais são possíveis capturar características relacionadas ao estado de frustrações e decepções.

Os resultados do experimento do trabalho mostraram que 8 pessoas não realizaram corretamente pelo menos 1 atividade das quais envolviam todos os trabalhos. Isso significa que o EEG identificou algum padrão que resultou em uma performance negativa para essas pessoas. Além disso, foi concluído que os resultados foram considerados altos para classificação de desempenho de pessoas que não entendem sobre a tarefa que estão executando, realizando esta tarefa. Quanto aos resultados de participantes individuais realizando uma tarefa, a Rede Neural pode distinguir muito claramente os níveis fáceis e difíceis de tarefas.

Por fim, pode-se dizer que este estudo é útil e promissor e pode ser aplicado em empresas das quais querem avaliar de maneira sistemática o desempenho e qualidade de serviços dos seus funcionários em realizar uma dada tarefa. Além disso, o artigo mostra muito bem a importância em saber realizar a coleta dos dados do EEG bem como o algoritmo correto, sendo ele adaptativo, para este tipo de estudo.

2.2 EMOTIV *Epoc Servo Motor Control*

O projeto desenvolvido por Peterman (2013) utiliza-se de um *Arduino* e um *EMOTIV* para controlar dois servos motores a partir de ondas cerebrais. Basicamente, uma aplicação desenvolvida em C++ é responsável pela comunicação entre o *EMOTIV* e o *Arduino*. Esta aplicação recebe e interpreta os sinais do *headset* utilizado e os envia para o *Arduino*, que irá gerar os movimentos nos servos motores.

Um outro projeto, com uma ideia interessante, foi o projeto de conclusão do curso de Engenharia de Controle e Automação com o título “Exame Distante”, desenvolvido por Rodrigues (2013), na Pontifícia Universidade Católica do Paraná, cujo tema envolve uma solução para a área de diagnóstico médico. O *software* criado, “Exame Distante”, permite um diagnóstico médico, a distância, sobre informações cerebrais do paciente. Desse modo, os custos de um tratamento presencial seriam poupados e o tempo de ambas as partes também. Para a aquisição das informações cerebrais do paciente, foi utilizado o *neuroheadset EMOTIV*.

O projeto desenvolvido na Universidade de Minnesota (*Control of a Virtual Helicopter Using a Motor Imagery Based BCI*), no Departamento de Engenharia Biomédica, consiste em controlar um *drone* utilizando os sensores de posição do *EMOTIV*, *GYROX* e *GYROY*, por meio dos eixos *x*, *y*, *z*, não sendo necessário classificá-los para realizar uma aprendizagem profunda com o uso de algoritmos específicos de aprendizado de máquina (Doud et al., 2011).

Um trabalho muito similar ao desta dissertação, "*Balancing a simulated inverted pendulum through motor imagery: An EEG-based real-time control paradigm*", é o trabalho proposto por Yue et al. (2012), onde o objetivo é o controle de um pêndulo invertido virtual. Foram coletados dados de estudantes com uma média de 24,2 anos, sendo quatro homens e uma mulher, onde o pêndulo era mostrado em um monitor de 22 polegadas e os estudantes, já informados sobre todos os procedimentos que deveriam fazer, deveriam ter como objetivo o controle de um pêndulo invertido.

Um passo similar ao que foi usado nesta dissertação, também foi utilizado no trabalho em questão, que foi a exclusão dos sensores de posição onde notou-se que estes, por sua vez, geram muito ruído no processo de classificação dos dados obtidos do EEG. Os algoritmos de classificação conseguem classificar melhor os dados adquiridos sem estes sensores. Para o processo de classificação dos dados, foi utilizado o algoritmo análise discriminante linear (LDA, do inglês *linear discriminant analysis*).

Os resultados obtidos deste trabalho demonstraram que, a partir da metodologia bem definida relacionada a aquisição dos dados, é possível a utilização deste algoritmo para o controle do pêndulo invertido. Como este trabalho é aplicado a um pêndulo virtual, o próximo passo proposto por Yue et al. (2012) é aplicar os algoritmos deste trabalho a um pêndulo invertido real.

Em Sanchez-Fraire et al. (2015), foi proposto um modelo de controle de robôs, utilizando o *EMOTIV*, com cinco diferentes experimentos (com três diferentes pessoas – média de 26 anos). Foi proposto um modelo capturando os 14 eletrodos do EEG, eliminando os sensores de posição (mesma proposta do trabalho de dissertação) e classificando os dados utilizando uma MEF (máquina de estados finita) com duas abordagens: automático (não há iteração com o usuário) e semiautomática (há uma certa iteração com o usuário).

O primeiro experimento consiste no treinamento para a formação do estado cognitivo sob diferentes tipos de imagens mentais. Para isso, são consideradas as seguintes imagens:

- 1) Aparece uma tela de cor vermelho / amarelo intermitente para o usuário como estímulo visual;
- 2) Um estímulo de movimento circular exerce uma força de contato com um dispositivo háptico.

Com os dados acima, é possível determinar as seguintes medidas:

- Poder espectral;
- Tempo de ativação do comando: intervalo de tempo que os usuários são capazes de manter um estado cognitivo ativo (mesmo padrão de pensamento);
- Frustração do usuário.

No segundo experimento, foi utilizado o método automático da MEF, onde o objetivo era controlar, por meio dos dados do EEG, um robô que começava em um ponto A e deveria chegar até um ponto B. O terceiro experimento foi parecido com o segundo, diferindo-se através da utilização do método semiautomático e invertendo a ordem dos pontos.

O quarto experimento foi utilizando parte do primeiro experimento com um dispositivo háptico para alterar o estado mental cognitivo. Por fim, o quinto e último experimento foi controlar um braço de robô (similar aos movimentos do pêndulo), onde o usuário deveria pensar em girar o braço no sentido horário e anti-horário. Todos os experimentos mostraram bons resultados.

O trabalho “*Control a Robot via VEP Using Emotiv EPOC*” é uma tese desenvolvida por Ingel (2015) que descreve um sistema com potencial evocado visualmente em estado estacionário (SSVEP, do inglês *steady-state visual evoked potential*) de forma a controlar um robô. Em outras palavras, um sistema que descreve a resposta de um sinal do EEG baseado em estimulações visuais para obter o controle de um robô.

Para o desenvolvimento desta tese, foi utilizado o EEG *EMOTIV* e algoritmos para a extração de características. Estes algoritmos são PSDA (do inglês, *power*

spectrum density analysis) e CCA (do inglês, *canonical correlation analysis*) e os mesmos foram combinados de forma a gerar uma nova abordagem para este método.

Com os resultados obtidos, taxa de acerto $85.81 \pm 6.39 \%$, foi estabelecer o controle do robô.

O projeto “*Brain-Machine Interfaces for Robotic Control*” (GALBRAITH, 2016), desenvolvido com a colaboração de *Guenther lab* e *Neuromorphics lab* na Universidade de Boston, vislumbra o controle de um robô utilizando uma tecnologia EEG. O grande diferencial deste projeto é que o robô é um agente móvel adaptativo aprimorado com uma câmera rotativa e um braço robótico, onde o usuário pode navegar em um espaço com a possibilidade de enxergar em diversos ângulos o ambiente ao redor.

A tese denominada “*A Brain Controlled Wheelchair to Navigate in Familiar Environments*” (Rebsamen, 2009) tem o intuito de ajudar pessoas que não são capazes de movimentar os membros, que possuem problemas como Esclerose Lateral Amiotrófica (ALS), acidente vascular cerebral, lesão cerebral ou medular, paralisia cerebral, distrofias musculares ou esclerose múltipla. O objetivo principal é controlar sua cadeira de rodas utilizando a força do pensamento, com o intuito de utilizar este trabalho em hospitais e casas sem a necessidade de muita modificação no estado atual em que as pessoas que precisam disto encontram-se.

Para esta tese, foram desenvolvidos algoritmos capazes de realizar o controle da cadeira de rodas. Como conclusão, Rebsamen (2009) descreve as melhores opções de algoritmos para esta aplicação e explica o seu funcionamento.

Um trabalho com similaridades com o proposto por Penaranda e Baldwin (2012) foi o trabalho de Zhang e Yang (2014), onde foi proposto um modelo para a utilização de um *EEG* de forma a avaliar os sinais do cérebro humano (afetivos, ansiedade, entre outros) para entender o comportamento das pessoas durante o trabalho. Desta forma, é possível prevenir riscos de trabalho em atividades que exigem muito das pessoas.

No desenvolvimento deste projeto, foram utilizadas cinco diferentes pessoas para completar dois experimentos. Para a realização dos experimentos foram utilizados 11 eletrodos e, em seguida, foi aplicado o algoritmo análise de componentes independentes (ICA, do inglês *independent component analysis*), técnica de redução do número de instâncias, onde no trabalho em questão foi

observado que isso reduziria o piscar de olhos das pessoas no momento em que seus dados cerebrais eram coletados. Os dados foram filtrados e processados utilizando modelos *fuzzy*.

Com os modelos *fuzzy* adquiridos, foi proposto um novo modelo de controlador PID incremental com conceitos de técnicas de *swarm* (baseados nos comportamentos de pássaros e peixes) junto com PSO (do inglês, *particle swarm optimization*), o qual foi denominado de IPID-PSO. Este modelo proposto obteve bons resultados e mostrou que é possível a utilização destes algoritmos juntos de forma a obter um melhor desempenho e, quando necessário, uma redução de ritmo de trabalho das pessoas, baseado nos dados coletados do *EEG*.

Este trabalho, com o título “*Trajectory Path Planning of EEG Controlled Robotic Arm Using GA*” (Roy et al., 2016), tem o objetivo de controlar um braço robótico utilizando um eletroencefalograma. Em Sanchez-Fraire et al. (2015), foi proposto um trabalho muito parecido, entretanto, os algoritmos utilizados, tanto para a aquisição dos dados quanto para a classificação dos mesmos, são diferentes.

Os algoritmos utilizados neste trabalho foram:

- Algoritmo genético: utilizado para auxiliar no processo de aquisição de dados para o planejamento da posição que o objeto irá alcançar. Desta forma, é possível obter a melhor trajetória possível para o braço robótico.
- Máquina de vetor de suporte: foi utilizada no processo de classificação dos sinais do *EEG*.

Os resultados obtidos atingiram 75,77% de acurácia utilizando os algoritmos propostos neste trabalho (algoritmo genético e máquina de vetor de suporte). Haja vista, esses valores podem ser melhorados em um trabalho futuro, apesar de não ser um trabalho fácil de se obter valores maiores de acurácia.

O trabalho (*Brain computer interface: control signals review*) de Ramadan e Vasilakos (2017) tem intuito de dar uma abordagem geral a respeito das BCIs (*brain computer interface*). Primeiramente é mostrado um apanhado geral da literatura sobre o funcionamento e arquitetura do cérebro. Em seguida, são mostradas tecnologias *EEGs* disponíveis, bem como suas funcionalidades e características.

Para um início de trabalho, este artigo tem uma abordagem muito didática e interessante, podendo auxiliar no início dos estudos sobre BCIs e *EEGs*.

Na tese de Le (2015), é proposto um modelo utilizando um *EEG* para o controle de um carro robótico. Este trabalho tem um intuito parecido com o de Rebsamen (2009), para ajudar pessoas com incapacidade motora.

Para o desenvolvimento da tese, foi utilizado o algoritmo de máquina de vetor de suporte (SVM, do inglês *Support Vector Machine*) e foram treinados dados sobre diferentes direções e letras, de forma exaustiva, até que o algoritmo de SVM ficasse robusto o suficiente. Em seguida, foi aplicado a um display LCD os comandos treinados.

O artigo “*The future of brain-computer interfacing (keynote paper)*” apresenta um breve resumo (Nijholt, 2016) sobre as BCIs e tecnologias *EEG*. Similiar ao trabalho de Ramadan e Vasilakos (2017), este trabalho é mais reduzido e mostra um pouco dos conceitos iniciais de BCIs, um explanado geral desde que o conceito começou a se propagar (1970) até os dias atuais, comparando a evolução de possíveis aplicações e de como as aplicações que já existem têm usado essa tecnologia. O trabalho também relata a colaboração com o estado da arte dos pesquisadores da área médica e da área da computação.

O capítulo 7 da tese de Duvinage (2015), com o título “*Towards Effective Non-Invasive Brain-Computer Interfaces Dedicated to Ambulatory Applications*”, mostra uma comparação interessante entre o *EMOTIV* e um sistema de grau médico (dispositivo ANT, do inglês *Advanced Neuro Technology*) utilizando os mesmos eletrodos em um sistema P300 BCI. Além disso, é apresentado um relevante estudo da literatura sobre o *EMOTIV*, apresentando suas principais características.

Como resultado desta comparação, o *EMOTIV* ficou um pouco abaixo. Entretanto, ainda assim apresenta-se como uma excelente opção para realizar estes estudos. Um outro ponto a ser citado sobre o *EMOTIV* é que ele apresentou uma oxidação dos sensores no experimento.

Similar a alguns trabalhos mostrados anteriormente sobre o controle de robôs utilizando tecnologias *EEG* (Roy et al., 2016). Chae et al. (2011), propuseram um trabalho sobre o controle de um humanoide (robô que simula os movimentos humanos) através de uma tecnologia *EEG*. O título deste trabalho é “*Noninvasive Brain-Computer Interface-based control of humanoid navigation*”.

A tese de Sarkisyan (2013) propõe o controle de uma mão de prótese utilizando um *EEG*. O *EEG* utilizado foi o *EMOTIV* e, para a extração de dados no cérebro, foram pensadas as seguintes ações para o controle da prótese:

- Piscar olho esquerdo;
- Piscar olho direito;
- Imaginar a mão direita se movimentando;
- Apertar;
- Erguer as sobrancelhas;
- Sorrir.

Em seguida, os movimentos foram combinados da seguinte forma:

- Piscar olho esquerdo e imaginar a mão esquerda se movimentando;
- Piscar olho direito e imaginar a mão esquerda se movimentando;
- Apertar e piscar olho esquerdo;
- Apertar e piscar olho direito;
- Apertar e olhar para a direita;
- Erguer as sobrancelhas e olhar para a direita;
- Piscar olho direito e sorrir do lado esquerdo.

Também foi citado as combinações que não obtiveram resultados apropriados e promissores:

- Apertar e erguer as sobrancelhas;
- Apertar e imaginar o movimento da mão;
- Piscar com os dois olhos.

Os resultados apresentados nesta tese demonstraram que é possível o controle da prótese. Entretanto, para melhorar a precisão foi citado a utilização de EEG com 32 e 64 eletrodos.

O trabalho de Agarwal et al. (2015), com o título “*Classification of mental tasks from EEG data using backtracking search optimization based neural classifier*”, propõe a utilização de um classificador neural com uma busca otimizada (BSANN, do inglês *Backtracking Search optimization Algorithm*), evolução do algoritmo BSA (*Backtracking Search algorithm*), para a classificação de três atividades: movimentar a mão esquerda, movimentar a mão direita e a gerar palavras.

O trabalho proposto por Majumdar (2011) visa utilizar diversos algoritmos de aprendizado de máquina para reconhecimento de padrões dos dados de um EEG. Dentre as possíveis abordagens, foram utilizados algoritmos baseados em redes neurais, sistemas *fuzzy*, computação evolutiva, discriminação estatística e inferência Bayesiana. Também foram utilizadas técnicas de redução de dimensionalidade (PCA, ICA) para um controle maior manipulação dos dados e uma conversão do domínio da frequência (wavelets).

Como no trabalho de dissertação em questão foi utilizado o algoritmo de máquina de aprendizado extremo (ELM, do inglês *Extreme Learning Machine*), é conveniente citar o trabalho de Liang et al. (2006), "*Classification of Mental Tasks from EEG Signals Using Extreme Learning Machine*", onde foi utilizado o algoritmo ELM para a classificação de cinco tarefas mentais. Os resultados da acurácia do algoritmo ELM foram comparados com a abordagem de redes neurais artificiais com *backpropagation* e com máquina de vetor de suporte.

No processo de aquisição dos dados, foram gravadas sessões de 10 segundos para cada uma das tarefas mentais com 7 pessoas diferentes. As tarefas mentais são: relaxamento, composição de uma carta, multiplicação, contagem e rotação. Desta forma, os resultados obtidos mostraram que o algoritmo de SVM apresentou uma melhor média de valores. Entretanto, para alguns movimentos, os resultados da máquina de aprendizado extremo apresentaram-se melhores.

Estes resultados de comparação entre ELM e SVM mostraram-se próximos aos obtidos neste trabalho de dissertação.

Considerando todos os trabalhos propostos estudados, pode-se afirmar que este trará novas abordagens para o tema e complementarará ainda mais o estudo referente a interfaces cérebro-máquina, com novas abordagens de algoritmos de aprendizado de máquina, conclusões referentes a melhor forma de se obter dados do EEG e comunicação com o processo de pêndulo, bem como seu sistema de controle.

3 FUNDAMENTAÇÃO TEÓRICA

Na sessão a seguir, são embasados os aspectos teóricos da dissertação. Nas seções 3.1 e 3.2, são tratados os conceitos referentes ao EEG. Nas seções 3.3, 3.4, 3.5, 3.6, 3.7 e 3.8, os aspectos teóricos são baseados nos algoritmos de Aprendizado de Máquina que são utilizados. Por fim, nas seções 3.9 e 3.10, são explicados os conceitos sobre o pêndulo e o controlador utilizado. A Figura 2 ilustra todos os conceitos de cada módulo desta etapa do trabalho.

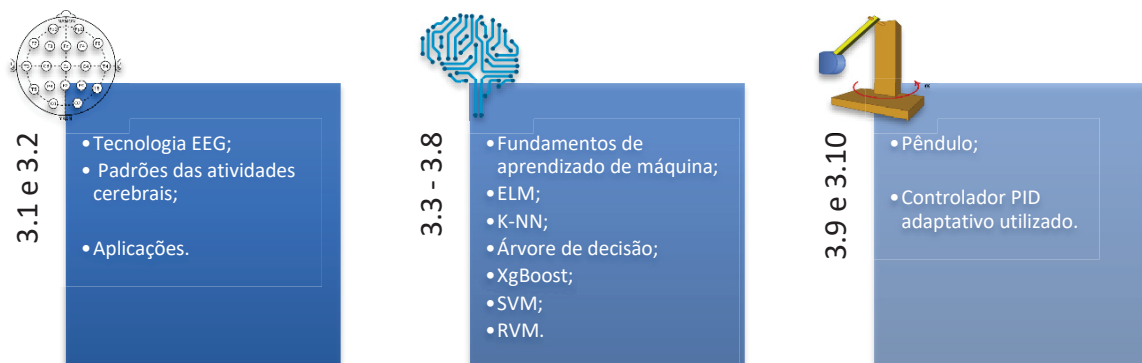


Figura 2 – Fundamentação teórica tratada em cada módulo do projeto

FONTE: O autor (2017)

3.1 Tecnologia EEG: Como a Eletroencefalografia Funciona

Este módulo explica como o eletroencefalograma (EEG) funciona e os diferentes padrões de atividade cerebral que são uma base para a análise de EEG, seguido por algumas palavras sobre aplicações práticas do mesmo. Em primeiro lugar, uma distinção rápida: EEG é sobre as próprias ondas cerebrais. No contexto do projeto, significa que o EEG é falar somente do cérebro humano e dos dados que são capturados e ICC é um sistema que interpreta estes sinais de uma forma computadorizada onde é possível utilizá-los no processamento de algoritmos.



Figura 3 - Ilustração de um neurônio e suas sinapses

FONTE: Portal Online do *Midwest Institute* (2016)¹

3.1.1 Introdução ao EEG e como a Atividade do Cérebro pode ser medida

O cérebro humano sempre fascinou os seres humanos e, particularmente, um cientista alemão denominado Hans Berger, que descobriu a eletroencefalografia cerca de 80 anos atrás. Depois disso, novos métodos para explorá-la foram encontrados, podendo ser classificados em dois grupos principais: Invasiva e não-invasiva. Uma abordagem invasiva requer implantes físicos de eletrodos em seres humanos ou animais, tornando-se possível medir neurônios individuais em locais potenciais do cérebro. Uma abordagem não invasiva faz uso de, por exemplo, imagem por ressonância magnética e a tecnologia EEG para fazer medições. Ambos fornecem diferentes perspectivas, nos permitindo olhar dentro do cérebro e observar o que acontece (Kropotov, 2009). Em EEG, os potenciais elétricos entre neurônios são captados a partir de eletrodos posicionados no couro cabeludo. Os pares de eletrodos condutores, conforme Figura 4, concebidos usando a prata, por exemplo, são usados para ler esta eletricidade. A diferença de tensão entre os eletrodos é medida, e uma vez que o sinal é fraco (30-100 μV) tem que ser amplificada. A corrente ocorre quando os neurônios se comunicam. O caso mais simples é chamado potencial de ação, uma descarga rápida de tensão causada por uma abertura e fechamento de canais Na^+ e K^+ ion na membrana neuronal. Se a membrana despolarizar a algum limiar, o neurônio pode sofrer uma descarga de

¹ Disponível em: <www.midwestinstituteforaddiction.org/learn-about-addiction/physiological-adaptations>

tensão não prevista e parar de funcionar. Acompanhando estas descargas ao longo do tempo, pode-se monitorar a atividade cerebral.

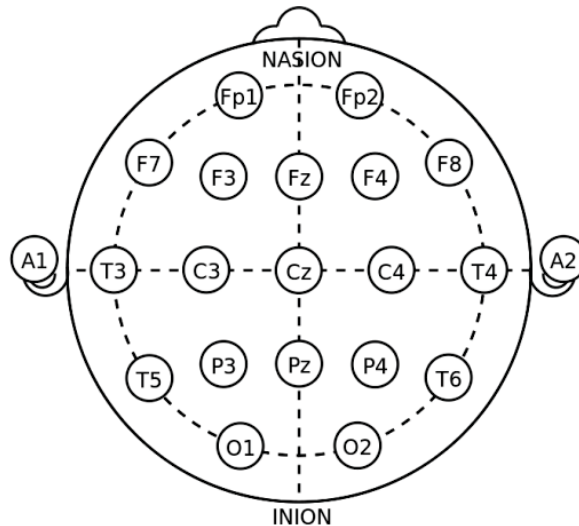


Figura 4 - Posicionamento de eletrodos dos sistemas 10-20 para medição EEG

FONTE: Repositório de mídia livre do Wikipédia (2016)²

3.1.2 Identificando Diferentes Padrões de Atividade Cerebral

Voltando à Hans Berger (Karbowski, 2002), ele descobriu que diferentes frequências elétricas poderiam ser conectadas às ações e diferentes estágios da consciência (baseados nos lobos específicos do córtex cerebral). Esta noção foi obtida por meio da observação de indivíduos realizando tarefas diferentes, tal como a resolução de problemas matemáticos, durante a gravação do seu EEG. A Figura 5 mostra as faixas de frequências, largura de banda medidas em Hz (ciclos por segundo), das ondas Delta, Theta, Alpha, Beta e Gama.

² Disponível em: <commons.wikimedia.org/wiki/File:21_electrodes_of_International_10-20_system_for_EEG.svg>

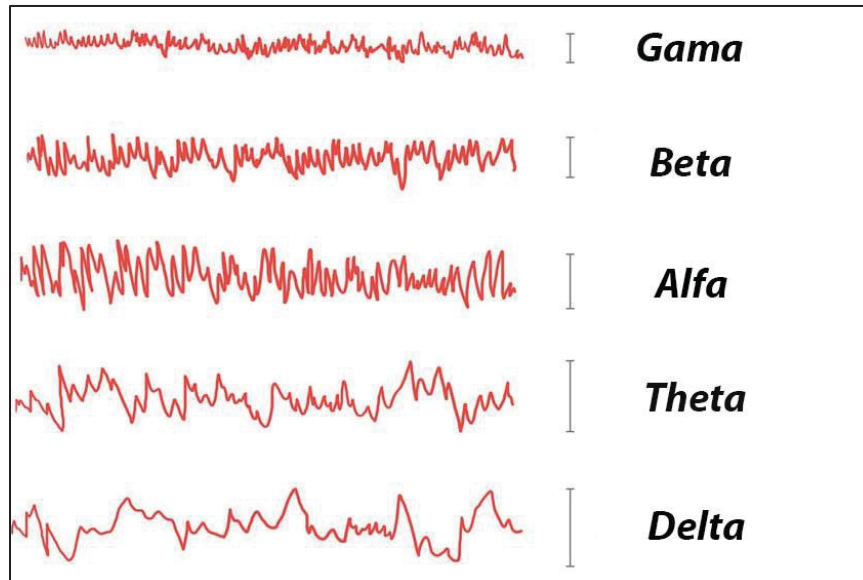


Figura 5 - As 5 principais frequências e suas relações
 FONTE: Adaptado de acervo online piramidal (2017)³

As ondas *gama* estão na faixa de frequência dos 31Hz ou superior. Pensa-se que isto reflete o mecanismo de consciência. Ondas *beta* e *gama* em conjunto têm sido associados com atenção, percepção e cognição.

As ondas *beta* estão na faixa de 12 Hz a 30 Hz de frequência, mas são geralmente divididos em β_1 e β_2 para se obter uma faixa mais específica. As ondas são pequenas e rápidas e associadas com a concentração e o foco. Podem ser melhor definidas em áreas centrais e frontais do cérebro. Ao resistir, evitar um movimento, ou resolver um problema matemático, se obtém um aumento significativo da atividade *beta* (Zhang et al., 2008).

As ondas *alfa* variam entre 7,5-12 Hz, são mais lentas e associadas com o relaxamento e desligamento. Pensando em algo pacífico com os olhos fechados deve-se obter um aumento da atividade *alfa*, mais profunda na parte de trás da cabeça (O1 e O2, conforme ilustrado na Figura 4) e no lobo frontal.

As ondas *theta* variam de 3,5 a 7,5 Hz, estão ligados à ineficiência, sonhar acordado, e as ondas *theta* de baixa frequência representam o estado intermediário entre estar acordado e estar dormindo. As ondas *theta* surgem do estresse emocional, especialmente frustrações ou decepções. Essas ondas também têm sido associadas com o subconsciente, inspiração criativa e profunda meditação. Altos

³ Disponível em:
 <https://piramidalcwb.files.wordpress.com/2017/09/brain_waves_large.jpg?w=1000&h=680>

níveis de *theta* são considerados anormais em adultos, e é, por exemplo, muito relacionada com o *transtorno do déficit de atenção com hiperatividade* (Heinrich et al., 2007).

As ondas *delta* variam entre a faixa de frequência de 0,5 a 3,5 Hz, são as ondas lentas e ocorrem durante o sono (Hammond, 2007). Se essas ondas ocorrem no estado de vigília, isto pode indicar defeitos físicos no cérebro.

A unidade motora está associada com atividades motoras, e é também encontrada na faixa de frequências de ondas *alfa*, mas em que a amplitude máxima é registrada pelo córtex motor. Por isso, basicamente, é disparada quando há um movimento real ou uma intenção de se mover (Hammond, 2007).

Todos estes grupos de ondas ocorrem em diferentes partes do cérebro e ajudam a compreender o estudo referente às ICCs. Estas ondas cerebrais podem ser extraídas de sinais com frequência de amostragem próximas ao EEG utilizado neste trabalho (128 Hz) e podem ser separadas a cada segundo de coleta de dados, respeitando suas bandas de frequência.

3.1.3 Aplicação Prática da Tecnologia EEG

Uma aplicação muito utilizada de EEG é observar e estudar os registros manualmente, para pesquisar, ou para entender, danos cerebrais e vários distúrbios, como, por exemplo, a epilepsia (Sundaram et al., 1999). Pesquisas e estudos de caso ao longo das décadas levaram a mapas cerebrais funcionais, segundo mostrado na Figura 6, que combinado com eletrodos colocados de acordo com o sistema 10-20, conforme apresentado na Figura 4, faz com que as atividades nessas áreas possam ser observadas. EEG é também um equipamento utilizado em hospitais para declarar quando o cérebro de um paciente está morto, quando nenhuma atividade é monitorada.

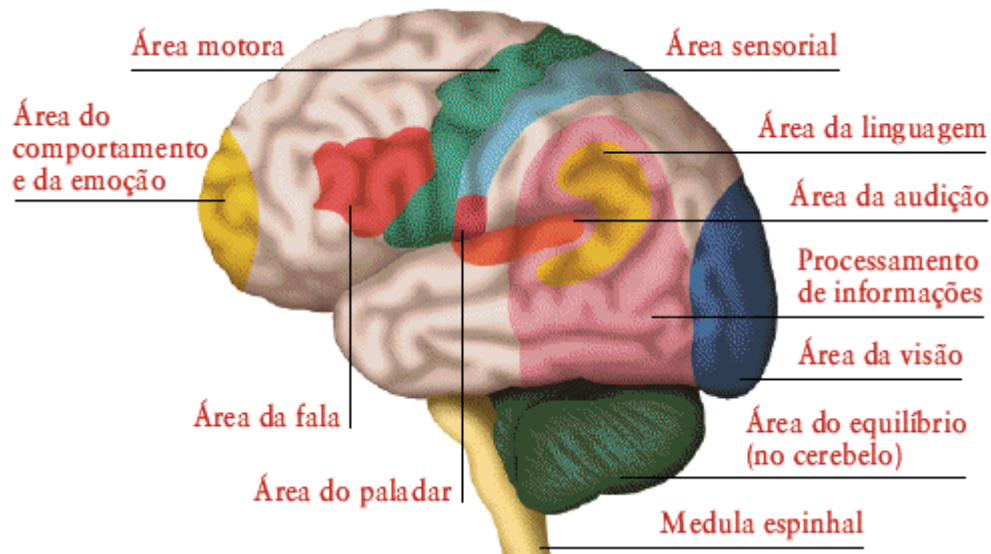


Figura 6 - Mapa funcional do cérebro humano

FONTE: Portal online *Sophia of Nature* (2016)⁴

O estudo das ondas cerebrais e como elas se relacionam com diferentes estados mentais têm levado a vários métodos e crenças alternativas sobre como manipular essas ondas. Por exemplo, a fim de tornar-se mais relaxado, concentrado e mais esperto, pode-se escutar músicas que tocam em faixas de frequências específicas que prometem fazer exatamente isso, como por exemplo, deixar um bebê escutar Mozart, na fase de crescimento, torna-se possível obter os efeitos mencionados (Braverman, 1990).

Uma área que tem sido muito pesquisada é a relação das drogas e medicamentos com o EEG. E de acordo com Braverman (1990), um aumento da atividade *alfa* é encontrada quando se toma antidepressivos e drogas que causam dependência, como a morfina, heroína e maconha. Também foi identificado que em usuários de drogas, muitas vezes falta uma quantidade natural de ocorrência de ondas *alfa*, e, assim, pode-se explicar porque esses usuários se tornam viciados. Foi observado em alcoólicos um excesso de ocorrência de ondas *beta*, e que isto pode inibir a sua capacidade de relaxar. A pesquisa com álcool mostra que a sua utilização aumenta a amplitude das frequências de ondas lentas e diminui as ondas rápidas. Assim, sugere-se que o álcool se torna um remédio rápido para relaxar.

Além disso, Braverman (1990) menciona que "como as ondas cerebrais simbolizam as várias partes da nossa consciência, e que se nós obtermos o

⁴ Disponível em: <<https://sophiaofnature.files.wordpress.com/2014/12/funcoesdocerebro.gif>>

conhecimento e tratamento necessário para alterá-los, podemos chegar mais perto de controlar nossas ondas cerebrais e mantê-las equilibradas".

3.2 Interface Cérebro-Computador

Essa interface foi explicada por Wolpaw (2010), da seguinte forma:

A interface cérebro-computador é um método de comunicação baseado em atividade neural gerada pelo cérebro e é independente dos seus percursos de saída dos nervos e músculos. O objetivo da ICC não é determinar a intenção de uma pessoa através da espionagem de sua atividade cerebral, mas sim fornecer um novo canal de saída para o cérebro que requer controle adaptativo voluntário por parte do usuário.

Além disso, Wolpaw (2010) identifica quatro diferentes áreas de aplicação da ICC, os quais são:

Aplicações de Bioengenharia: Dispositivos com fins de assistência para pessoas com deficiência (Rebsamen, 2009; Stamps e Hamam, 2010).

Monitoramento do ser humano: investigação e detecção de distúrbios do sono, doenças neurológicas, monitoramento de atenção, e/ou "estado mental" global (Sundaram et al., 1999).

Pesquisas da neurociência: métodos em tempo real para correlacionar o comportamento observável com sinais neurais gravados (Penaranda e Baldwin, 2012).

Interação homem-máquina: dispositivos de interface entre os seres humanos, computadores ou máquinas (Santos e Cordeiro; et al., 2017).

Busca-se neste trabalho seguir os passos do processo na Figura 7.

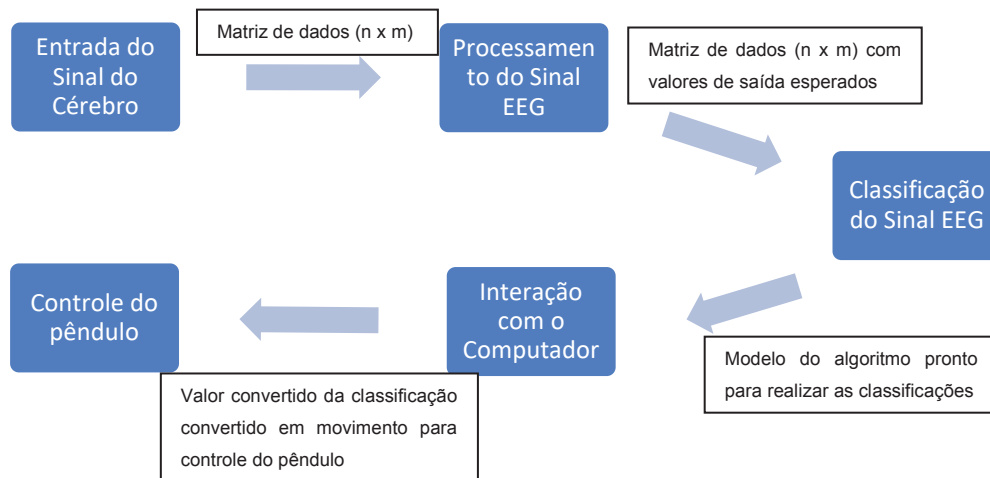


Figura 7 - Fluxo do processo ICC para controle do pêndulo

FONTE: O autor (2017)

3.2.1 Cinco Fatores Centrais de um Projeto ICC

Algumas características que são comuns e críticas ao design ICC foram explicadas por Lotte et al. (2007), as quais estão listados a seguir:

- Ruído e valores discrepantes: As características da ICC são barulhentas ou contêm valores extremos, pois os sinais de EEG têm uma baixa relação sinal-ruído;
- Alta dimensionalidade: em sistemas ICC, vetores de características são muitas vezes de alta dimensionalidade. De fato, vários recursos são geralmente extraídos de vários canais e de vários segmentos de tempo antes de ser concatenados em um único vetor de características;
- Informações temporais: características ICC devem conter informações de tempo porque padrões de atividade do cérebro estão geralmente relacionadas a variações específicas de tempo;
- Não estacionariedade: características ICC são não-estacionárias, porque sinais de EEG pode variar rapidamente ao longo do tempo;
- Pequenos conjuntos de treinamento: os conjuntos de treinamentos são relativamente pequenos, uma vez que o processo de treinamento é demorado e exigente.

A alta dimensionalidade é realçada porque o dispositivo *EMOTIV*, mais explicado na sessão 4.1.1, utilizado para este projeto é um dispositivo de vários canais. Isto aumenta não só a complexidade de processamento, como também aumenta a precisão do sinal, pois cada sinal é obtido em um canal diferente.

3.3 Fundamentos de Aprendizado de Máquina

Segundo Coppin (2010), o aprendizado está diretamente ligado com a inteligência, pois se um sistema é capaz de aprender a exercer determinada tarefa, este sistema merece então ser chamado de inteligente.

O Aprendizado de máquina é uma área de estudo que fornece aos computadores a capacidade de se modificar ao longo do curso de suas interações com o mundo, bem como pela experiência de seus próprios estados e processos internos. Em outras palavras, a capacidade de aprender sem a necessidade de se programar explicitamente este processo de aprendizagem. A definição de Mitchell (1997) para aprendizado de máquina: “um programa de computador é feito para aprender de uma experiência *A* sobre alguma tarefa *T* e sobre alguma medida de performance *P*, se esta performance em *T*, como medido por *P*, melhora com uma experiência *E*”. À medida que uma certa tarefa melhora com a experiência, esta performance pode ser medida e validada para realizar alguma ação. Na Figura 8 pode-se observar o modelo de aprendizado de máquina.

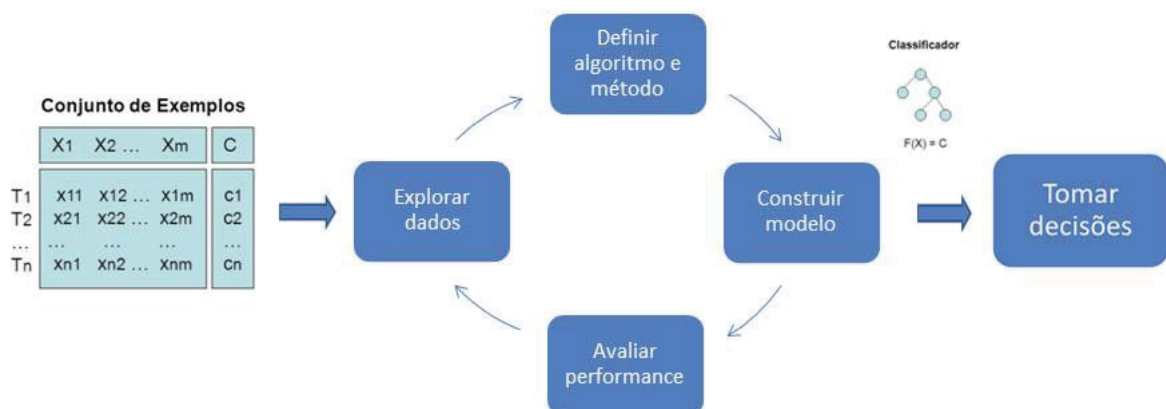


Figura 8 - Modelo de aprendizado de máquina

FONTE: O autor (2017)

Conforme sugere Coppin (2010), existem vários métodos de aprendizado de máquina. Entre eles o aprendizado por hábito, que tem como característica o programa aprender por experiência de acordo com o que foi informado anteriormente. Porém, o programa apenas armazena os dados que podem ser classificados, e caso não consiga classificar os valores informados, o método falhará. Há o método de aprendizado por conceito, que analisa todas as hipóteses e demonstra qual é a correta. No método por conceito existe uma subdivisão, que é a “hipótese mais geral”, o que significa que se não existe nenhuma possibilidade correta, o programa determina a que mais se aproxima do correto. Mas estes métodos têm alguns problemas. Por exemplo, nem sempre o usuário quer saber a hipótese correta, e sim a mais comum.

Um algoritmo de Aprendizado de máquina, o qual deve ser um modelo computacional que pode adaptar-se e aprender a partir da experiência, tem como principal objetivo descobrir o relacionamento entre as variáveis de um sistema (entrada e saída) a partir de dados amostrados.

3.4 Redes Neurais Artificiais

Existem cinco categorias que abrangem os algoritmos mais utilizados em sistemas de classificação ICC, os quais são: classificadores lineares, classificadores bayesianos não lineares, classificadores vizinhos mais próximos, redes neurais e uma combinação de classificadores (Lotte et al., 2007). Em todas as categorias, foram alcançados bons resultados em sistemas de classificação ICC. No entanto, as redes neurais são as mais populares em pesquisa ICC, e é um dos métodos que escolhemos para classificar os dados deste projeto.

As redes neurais artificiais funcionam de uma maneira muito eficiente quando são aplicadas em reconhecimento de padrões. Esse modelo de algoritmo de classificação é um modelo adaptativo inspirado no próprio cérebro humano, especificamente tomando como base as características de processamento de informação encontradas nos neurônios. O cérebro humano possui uma capacidade de processar enormes quantidades de informações, e isso despertou o interesse das pessoas em basear um algoritmo de classificação no comportamento das redes neurais biológicas.

As redes neurais artificiais são constituídas por conjuntos de neurônios artificiais que interagem entre si, assemelhando-se ao comportamento dos neurônios biológicos. Em geral, "redes neurais são sistemas paralelos distribuídos, compostos por unidades de processamento simples (nodos) que calculam determinadas funções matemáticas, normalmente não lineares" (Braga et al., 2007).

As redes neurais artificiais têm, em geral, a capacidade de coletar, utilizar e armazenar informações baseadas em experimentos (processo de aprendizado). O processo de aprendizado realizado por meio de algoritmos de aprendizado é o processo fundamental de uma rede neural artificial, onde os pesos sinápticos da rede são modificados de uma forma ordenada para alcançar o resultado desejado" (Braga et al., 2007).

A partir da informação contida em pares de entradas e saídas, contidos em um banco de dados, denotados por $[(p_1, t_1), (p_2, t_2), \dots, (p_n, t_n)]$ (amostras de treinamento), as redes neurais devem "aprender" e generalizar o conhecimento contido nas amostras de forma a responder de forma adequada quando entradas não presentes no conjunto de treinamento forem apresentadas à rede neural artificial.

3.4.1 Modelo Artificial de Neurônios

Em geral, todos os diferentes modelos de rede neural se baseiam em criar uma rede de neurônios artificiais, que se aproximam o máximo possível de uma rede de neurônios biológicos. Para cada neurônio artificial, tem-se diversas entradas, as quais correspondem às conexões sinápticas com outras unidades similares a ele, e uma saída, cujo valor depende diretamente da somatória ponderada de todas as saídas dos outros neurônios a esse conectado.

Ao olhar para a própria biologia, conforme apresentado na Figura 9, temos os dendritos, dos quais são responsáveis por receber os sinais de entrada, o axônio que irá transmitir os sinais para as terminações do axônio e isso será repassado para outros neurônios. Ou seja, o processo será repetido para N neurônios.

Basicamente, tem-se várias entradas para várias saídas, ou até mesmo várias entradas para uma saída. Isso varia dependendo do tipo de solução que se deseja encontrar ou do tipo de aplicação.

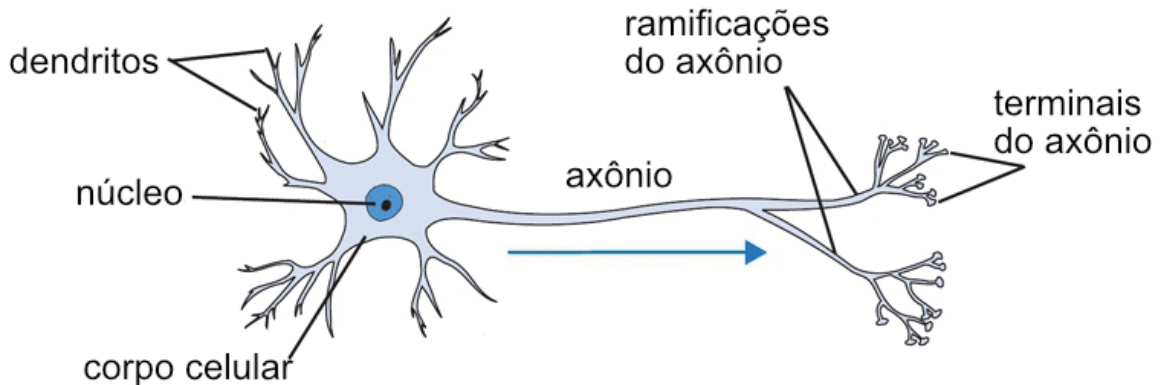


Figura 9 - Estrutura de um neurônio biológico
 FONTE: Portal online researchgate.net (2017)⁵

Na Figura 10 observa-se o modelo matemático tradicional de um neurônio artificial, baseado no modelo de McCulloch e Pitts (1943), onde x_1, x_2, \dots, x_m , são os sinais de entrada, oriundos da camada de entrada. O processo sináptico é representado pelos pesos (w) que amplificam cada um dos sinais recebidos.

Em seguida isso é dada como entrada para a Σ , também chamado de função agregadora. A função agregadora tem o objetivo de receber todos os sinais e realizar a soma dos produtos dos sinais. Ou seja, os sinais são recebidos na função agregadora e a função irá avaliar, através do sinal de *bias*, se os sinais recebidos são transmitidos, inibidos ou alterados.

Quando o sinal passa, obtém-se V_k , denominado potencial de ativação, e em seguida irá entrar na função de ativação [$\Psi(\cdot)$], que irá modelar a forma como o neurônio responde ao nível de excitação, limitando e definindo a saída da rede neural.

A variável V_k pode ser definido como:

$$V_k = \sum_{i=1}^n w_i * x_i - B_k \quad (1)$$

Onde w_i são os pesos e B_k as bias de polarização.

⁵ Disponível em: <
https://www.researchgate.net/profile/Sabrina_Bezerra/publication/307578398/figure/fig1/AS:401970222387203@1472848577750/Figura-22-Representacao-de-um-neuronio-biologico-Fonte-extraido-e-adaptado-de.ppm>

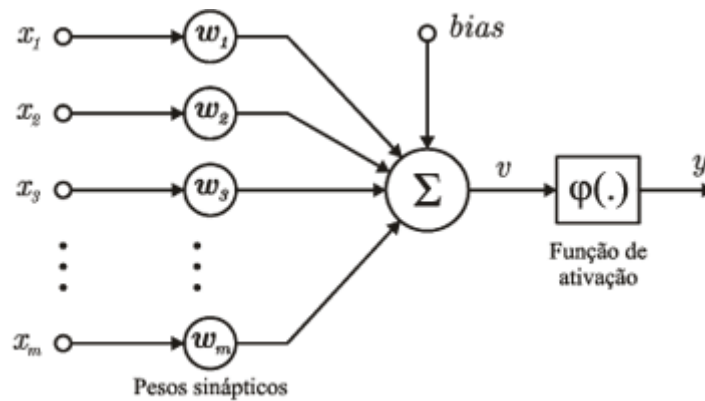


Figura 10 - Modelo matemático de um neurônio

FONTE: Documento online (2016)⁶

Conforme observa-se na Figura 10, obtém-se o modelo da rede neural a partir da função de ativação, onde y são as classes desejadas no processo de classificação.

3.4.2 Arquitetura de uma Rede Neural Artificial

Há diversos modelos diferentes de redes neurais artificiais. Cada modelo funciona melhor conforme a necessidade de sua aplicação. Em geral, define-se uma estrutura que compõe uma rede neural (Haykin, 1999):

- Camadas Intermediárias;
- Quantidades de Neurônios;
- Função de Transferência;
- Algoritmo de Aprendizado.

Para o desenvolvimento de uma rede neural, deve-se seguir um fluxo de trabalho para qualquer que seja a sua aplicação. O fluxo normalmente é dividido em sete etapas principais, as quais são (MathWorks, 2016):

- Coletar dados;
- Criar a rede;
- Configure a rede;
- Iniciar os pesos e *biases* da rede;
- Treinar a rede;

⁶ Disponível em: <<http://www.scielo.br/img/revistas/ca/v21n4/a03fig17.gif>>

- Validar a rede;
- Usar a rede.

O modelo típico de rede neural é estruturado em camadas de neurônios, onde alguns apresentam apenas uma camada, e são chamados de redes neurais de camada única. Outros, apresentam múltiplas camadas de neurônios, chamados de redes neurais de múltiplas camadas. A partir dessa estrutura topológica, à camada inicial é dado o nome de camada de entrada, à camada final, camada de saída, e às camadas intermediárias, camadas escondidas.

O modelo de rede neural *feedforward* foi o primeiro e o mais simples modelo de rede neural artificial. Eles são mais comumente usados para a previsão, reconhecimento de padrões, e função de montagem não-linear. Nesta rede, a informação se move em apenas uma direção, para a frente, a partir dos nós de entrada, através dos nós escondidos e para os nós de saída. Não há ciclos ou loops na rede.

2.4.2.1 Redes *Feedforward* com Camada Única

Em uma rede neural em camadas (multicamadas), supervisionada, onde os neurônios são organizados na forma de camadas, o termo “camada única” se refere à camada de saída de nós computacionais (neurônios). Sendo assim, neste modelo, a única camada é a de saída.

O tipo mais simples de rede neural é uma rede *feedforward* de camada única, conforme apresentado na Figura 11, que consiste numa única camada de nós de saída, as entradas são alimentadas diretamente às saídas através de uma série de pesos.

A soma dos produtos dos pesos e das entradas é calculado em cada nó. Caso o valor for superior a um limiar (tipicamente 0) o neurônio “ativa” e converte o valor ativado (normalmente 1). Caso contrário, assume o valor desativado (normalmente -1). Neurônios com este tipo de função de ativação também são chamados neurônios artificiais ou unidades de limiar linear.

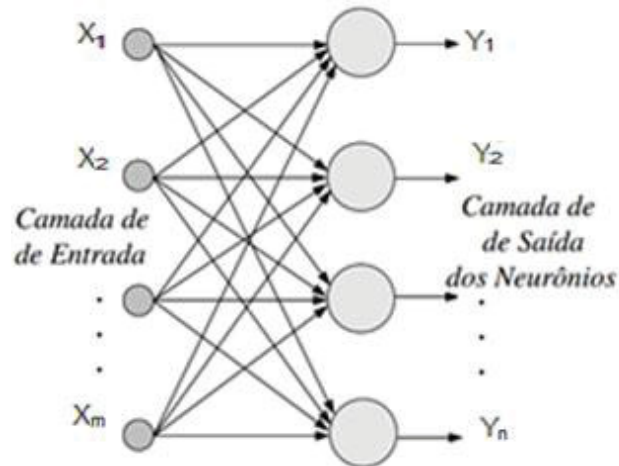


Figura 11 - Redes *feedforward* com camada única

FONTE: O autor (2016)

2.4.2.2 Redes *Feedforward* com Múltiplas Camadas

A arquitetura de uma *feedforward* com múltiplas camadas se difere pela presença de uma ou mais camadas intermediárias (mais conhecidas como camadas ocultas), cujos nós computacionais são chamados de neurônios ocultos ou unidades ocultas. A função dos neurônios ocultos é intervir entre a entrada externa e a saída da rede de uma maneira útil. Ao adicionar uma ou mais camadas ocultas, a rede se torna capaz de extrair informações estatísticas de ordem mais elevada, mas em compensação torna-se complexa, sendo complexo o seu funcionamento nas estruturas internas, e também lenta, pois o algoritmo demora um tempo maior para processar essas camadas adicionais.

Este tipo de rede neural artificial é um dos métodos mais populares e mais amplamente utilizados em muitas aplicações práticas (Setiono, 2001; Setiono e Lu, 1994; Xiaoping e He, 2011) . Ele é composto por várias camadas de neurônios, interligados em uma maneira *feedforward*. Cada neurônio em uma camada possui conexões com os neurônios da camada subsequente.

Redes de multicamadas usam uma variedade de técnicas de aprendizagem, sendo o mais popular a técnica de *back-propagation*.

Observa-se na Figura 12 um exemplo de rede neural com uma camada oculta.

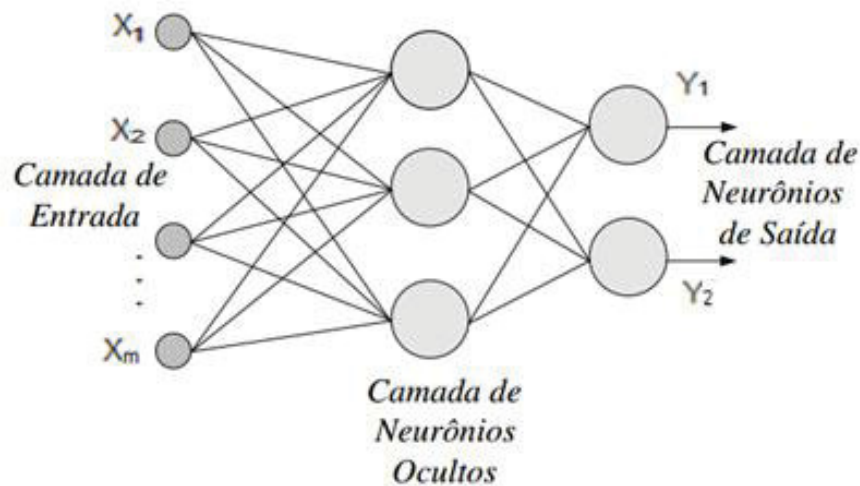


Figura 12 - Redes *feedforward* com múltiplas camadas

FONTE: O autor (2016)

3.4.3 O Mapa de Aprendizado

“O aprendizado em Redes Neurais Artificiais é de relevante importância e alvo de intenso trabalho de pesquisa, desenvolvido sob ambos os pontos de vistas: o biológico e o artificial” (Kartalopoulos, 1996).

Em geral, o aprendizado é um processo de constante ajuste das redes neurais aos estímulos recebidos. “Na realidade, durante o processo de aprendizagem, a rede ajusta continuamente seus parâmetros, os pesos sinápticos, em resposta aos estímulos de entrada de tal forma que a resposta real encontrada converge para a resposta desejada para o problema” (Kartalopoulos, 1996).

Em outras palavras, “funções de treinamento e aprendizado são procedimentos matemáticos usados para ajustar automaticamente pesos e *biases* da rede. A função de treinamento dita um algoritmo global que afeta todos os pesos e *biases* de uma determinada rede. A função de aprendizagem pode ser aplicada a pesos individuais e *biases* dentro de uma rede. Quando se obtém como saída a resposta desejada, a fase da aprendizagem da rede é finalizada, caso contrário a rede é ajustada e o processo continua.

O aprendizado de uma rede neural pode ser por dois métodos: aprendizagem supervisionada ou aprendizagem não supervisionada.

O aprendizado supervisionado é um tipo de algoritmo de Aprendizado de máquina que utiliza um conjunto de dados conhecidos (chamado de conjunto de

dados de treinamento) para fazer previsões. O conjunto de dados de treinamento inclui dados de entrada e valores de resposta. A partir dele, o algoritmo de aprendizado supervisionado busca construir um modelo que pode fazer previsões dos valores de resposta para um novo conjunto de dados. Um conjunto de dados de teste é muitas vezes usado para validar o modelo. Usando conjuntos de dados de treinamento maiores, muitas vezes pode-se produzir modelos com maior poder preditivo que pode funcionar bem para novos conjuntos de dados.

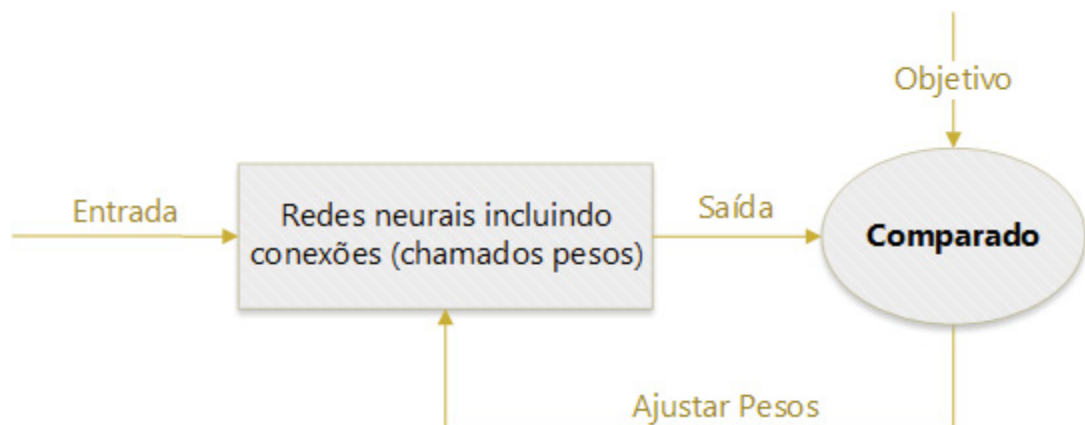


Figura 13 - Funcionamento de aprendizagem supervisionada

FONTE: O autor (2016)

Aprendizado não supervisionado é um tipo de algoritmo de aprendizado de máquina utilizado para fazer inferências a partir de conjuntos de dados que consistem em dados de entrada sem respostas definidas. Assim, a rede funciona de forma a organizar as entradas em categorias. Ao receber um novo estímulo de entrada, a rede irá indicar, como resposta, a classe da qual ela pertence. Caso a rede não encontre uma classe que represente essa entrada, ela irá imediatamente criar uma nova classe.

O método mais comum de aprendizagem não supervisionada é a análise de cluster, que é usado para a análise exploratória de dados para encontrar padrões ocultos ou agrupamento de dados. Os clusters são modelados usando uma medida de similaridade que é definida por métricas tais como distância euclidiana ou distância probabilística.

2.4.3.1 Modelo de Aprendizado por Retropropagação do Erro

Essa técnica busca iterativamente a mínima diferença entre as saídas desejadas e as saídas obtidas pela rede neural, com o mínimo de erro possível. Para obter o melhor resultado, a rede ajusta os pesos entre as camadas através da retropropagação do erro encontrado em cada iteração,

Na técnica de aprendizagem de retropropagação do erro (*error back-propagation*), os valores de saída são comparados com a resposta correta para calcular o valor de alguma função de erro predefinida. Através de várias técnicas, o erro é então transmitido para trás através da rede. Utilizando esta informação, o algoritmo ajusta os pesos sinápticos de cada conexão, a fim de reduzir o valor da função de erro para o menor valor possível. Depois de repetir este processo para um número “suficientemente grande” de ciclos (épocas) de treinamento, a rede irá geralmente convergir para um estado em que o erro entre o resultado obtido e o resultado desejado é baixo. Neste caso, pode-se dizer que a rede obteve aprendizado.

Detalhadamente, o algoritmo funciona da seguinte maneira, conforme apresentado na Figura 14 (Rumelhart et al., 1986):

- Um padrão de entrada é aplicado como um estímulo aos elementos da primeira camada da rede, que é propagado por cada uma das outras camadas até que a saída seja gerada.

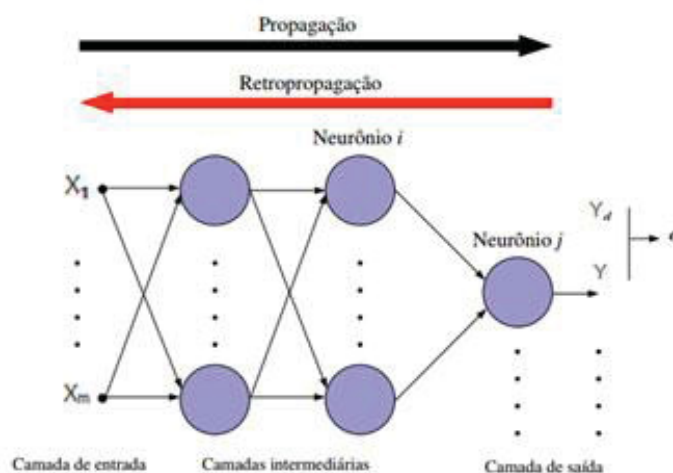


Figura 14 - Modelo de aprendizagem por *back-propagation*

FONTE: O autor (2016)

- Ela é então comparada com a saída desejada, a_d (gerando um sinal de erro para cada elemento da saída).
- O sinal de erro (δ_k) é então “retropropagado” da camada de saída para cada elemento da camada intermediária anterior que contribui diretamente para a formação da saída (equação 2).

$$\delta_k \leftarrow O_k(1 - O_k)(a_d - O_k) \quad (2)$$

onde O_k é a saída obtida e a_d é a saída desejada.

- Cada elemento da camada intermediária recebe apenas uma porção do sinal de erro total (δ_h), proporcional apenas à contribuição relativa de cada elemento na formação da saída original (equação 3).

$$\delta_h \leftarrow O_h(1 - O_h) \sum_{k \in \text{outputs}} w_{kh} \delta_k \quad (3)$$

onde w_{kh} são os pesos da camada de saída e da camada intermediária.

- Este processo se repete, camada por camada, até que cada elemento da rede receba um sinal de erro que descreva sua contribuição relativa para o erro total.
- Com base no sinal de erro recebido, os pesos sinápticos (w_{ji}) são então atualizados (de acordo com uma regra de correção de erro) para cada elemento de modo a fazer a rede convergir para o valor de saída desejada a_d (equação 4).

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji} \quad (4)$$

onde

$$\Delta w_{ji} = n \delta_h x_{ji} \quad (5)$$

onde n é igual a taxa de e x_{ji} é o vetor de entrada

3.4.4 Máquina de Aprendizado Extremo

A Máquina de Aprendizado Extremo (ELM) proposta por Huang et al. (2006), é um tipo de projeto de rede neural com arquitetura de redes *feedforward*, que visa resolver alguns problemas encontrados nas abordagens mais tradicionais de redes neurais. Estes problemas podem ser sobre-treinamento, do inglês *overfitting*, onde o comportamento pode se tornar generalizado a ponto de não fazer um bom reconhecimento dos padrões, convergência lenta (quando a taxa de treinamento é muito pequena), instabilidade e divergência da rede (quando a taxa de treinamento é muito grande).

A Máquina de Aprendizado Extremo foi desenvolvida para resolver estes problemas citados anteriormente, onde o objetivo principal é que com múltiplos neurônios na camada oculta seja possível mapear os dados obtidos para uma dimensão elevada onde os padrões possam ser separados por um hiperplano. Essa separação do hiperplano ocorre atribuindo os pesos da camada escondida aleatoriamente (não supervisionada), enquanto que os pesos da camada de saída são determinados analiticamente utilizando uma solução de mínimos quadrados (Huang et al., 2011). Essa forma de treinamento ocorre de maneira simples e com baixo custo temporal.

Com N amostras distintas, n neurônios e função de ativação $g(x)$ obtemos o seguinte modelo:

$$\sum_{i=1}^n w_i^2 g_i(x_j) = \sum_{i=1}^n w_i^2 g_i(w_i^1 x_i + \theta_i) = y_j, \quad j = 1, 2, 3, \dots, N \quad (6)$$

onde i corresponde ao índice do neurônio da camada escondida, w_i^1 aos pesos gerados de forma aleatória conectado ao neurônio da camada de entrada, w_i^2 aos pesos conectando os neurônios na camada de saída e θ_i ao limiar do neurônio. A seguir um pseudocódigo do funcionamento da ELM.

Pseudocódigo da ELM	
Entrada:	x : conjunto de treinamento $g(x)$: função de ativação n : número de neurônios na camada escondida
Saída:	w^2 : matriz de pesos de saída
	<ol style="list-style-type: none"> 1. <i>Atribuir pesos aleatoriamente para w_i^1 e θ_i</i> 2. <i>Calcular a matriz de saída da camada escondida</i> 3. <i>Calcular w^2</i>

A ELM pode reduzir o sobretreinamento, mas ainda assim pode-se gerar um conjunto não-ótimo de pesos e limiares e, desta forma, ocasionar no sobretreinamento. Sendo assim, faz-se necessário escolher o número de neurônios ocultos de forma a obter a melhor eficácia possível para o problema a ser resolvido.

3.5 Aprendizado Baseado em Instâncias

O processo de Aprendizado Baseado em Instâncias armazena as instâncias, calcula a distância entre as instâncias de treinamento e a instância desconhecida para avaliar o valor da função de classificação a partir dos valores das instâncias próximas, possui o conceito de aprendizado preguiçoso (indutivo). Esta abordagem não possui um modelo de construção de um modelo de classificação explícito. Os exemplos do algoritmo são classificados baseados em comparações usando similaridade com os exemplos de treinamento.

Para o cálculo da similaridade, as métricas mais utilizadas são distância Euclidiana (equação 7), distância de Mahalanobis (equação 8), distância de Bhattacharyya e divergência (equação 9), Similaridade de Cosseno (equação 10) e Correlação de Pearson (equação 11).

$$d(x_i, x_j) = \sqrt{\sum_{p=1}^n (a_p(x_i) - a_p(x_j))^2} \quad (7)$$

onde $a_p(x)$ é o valor do p -ésimo atributo da instância x .

$$d(x_i, x_j) = \sqrt{\sum_{p=1}^n \frac{(a_p(x_i) - a_p(x_j))^2}{\sigma_i^2}} \quad (8)$$

onde σ_i , é o desvio padrão de $a_p(x_i)$ no conjunto amostral.

$$d(p, q) = \sqrt{\sum_{i=1}^n \sqrt{p(x)q(x)}} \quad (9)$$

onde p e q são as distribuições de probabilidade discreta em relação há um conjunto de entrada X .

$$\cos(x_i, x_j) = \frac{\sum_{i,j}^n a_p(x_i) \cdot a_p(x_j)}{\sqrt{\sum_{i,j}^n a_p(x_i)^2} \sqrt{\sum_{i,j}^n a_p(x_j)^2}} \quad (10)$$

$$r(x_i, x_j) = \frac{\sum_{i,j}^n (x_i - \bar{x}_i) \cdot (x_j - \bar{x}_j)}{\sqrt{(\sum_{i,j}^n (x_i - \bar{x}_i)^2) \cdot (\sum_{i,j}^n (x_j - \bar{x}_j)^2)}} \quad (11)$$

onde \bar{x}_i e \bar{x}_j são as médias dos valores de x_i e x_j .

Na Figura 15 é possível observar uma representação do aprendizado baseado em instâncias, onde, a partir de um cálculo de similaridade das instâncias são geradas as hipóteses do modelo.

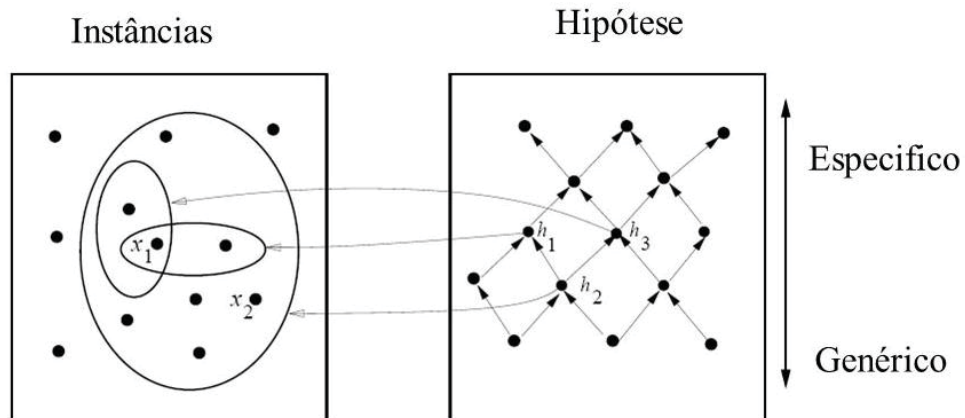


Figura 15 - Representação do método de aprendizado baseado em instâncias

FONTE: imagem da internet⁷

3.5.1 *K* Vizinhos mais próximos (K-NN)

Seguindo o conceito de Aprendizado Baseado em Instâncias, a técnica mais comum conhecida é a de *K* vizinhos mais próximos, aprendizado preguiçoso, difundida por e Cover e Hart (1967), onde o *K* define a quantidade de nós (vizinhos próximos) que são avaliados para tomar a decisão da classe da qual o nó está definido. Desta forma, quanto maior o valor de *K*, menos suscetível a ruído o algoritmo se torna. Entretanto, torna-se menor a flexibilidade de discriminação entre classes. A vizinhança pode ser definida pela função de distância euclidiana, medida de similaridade citada anteriormente. A seguir o algoritmo de funcionamento do método K-NN:

Algoritmo do K-NN
Entrada: <i>x</i> : conjunto de treinamento K: informar o valor de
Saída: rótulos de classificação
Para cada nova amostra:
<ol style="list-style-type: none"> 1. Calcular a distância para todas as amostras 2. Determinar o conjunto das <i>k</i>'s instâncias mais próximas 3. O rótulo com mais representantes no conjunto dos <i>k</i>'s vizinhos será o escolhido

⁷ Disponível em: <<http://slideplayer.com.br/slide/5213488/>>

Apesar desta técnica ser facilmente implementada e bastante flexível, classificar um exemplo desconhecido pode ser um processo computacionalmente complexo pelo motivo de que requer um cálculo de distância para cada exemplo de treinamento, o que pode consumir muito tempo quando o conjunto de treinamento é extenso. Além disso, apesar de idealmente os dados serem normalizados no processo de treinamento, a precisão da classificação pode ser muito prejudicada quando há a presença de ruídos ou características indesejadas nos dados.

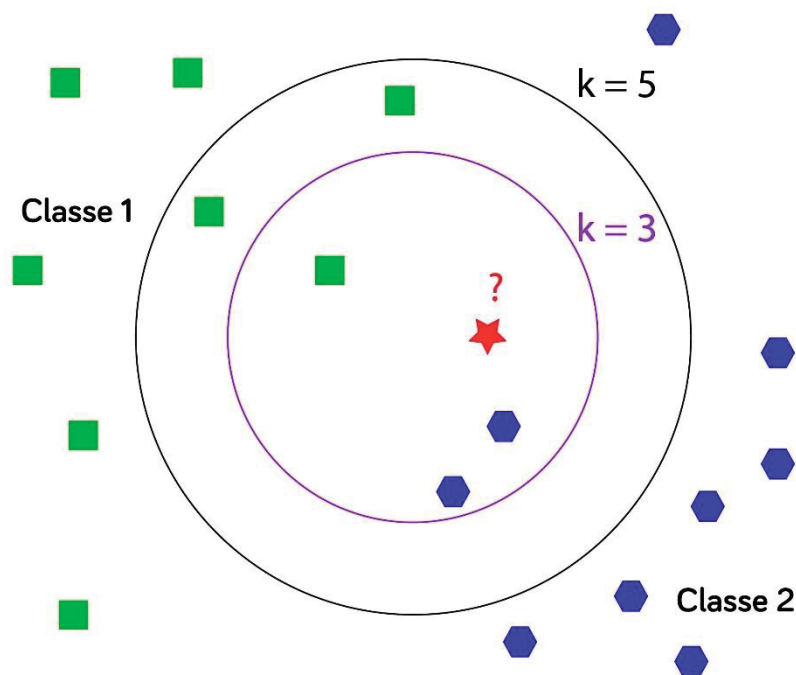


Figura 16 - modelo de funcionamento do algoritmo K vizinho mais próximo

FONTE: Imagem da internet⁸

Conforme pode-se observar, na Figura 16 têm-se uma representação do algoritmo K vizinho mais próximos com $K=3$ e $K=5$. Existem duas classes das quais as instâncias podem ser classificadas e, neste exemplo representativo, com $K=3$ a escolha seria para a classe 2 e para $K=5$ a classe escolhida seria a classe 1.

⁸ Disponível em: <
<http://www.coxdocs.org/lib/exe/fetch.php?media=perseus:user:activities:matrixprocessing:learning:knn.png>>

3.6 Árvore de Decisão

Entre os principais algoritmos de aprendizagem de máquina supervisionada para classificação, a árvore de decisão tem sido um dos principais e mais utilizados métodos. A árvore é construída num conceito simples: a cada nó, o conjunto de dados é dividido baseado em uma regra que minimize a entropia dos dois subconjuntos restantes. Isto significa que a cada divisão o dado se torna mais homogêneo, com parâmetros mais próximos. Os subconjuntos são divididos até que tenham acontecido N divisões desde a raiz ou que não seja possível encontrar em tempo limitado uma divisão que cause redução de N^2 na entropia. Neste caso, a classificação predominante dos dados de teste deste conjunto será a classificação desta folha.

A árvore de decisão é um método de aprendizagem indutiva que é relativamente robusto a ruídos nos dados. Seu modelo é composto por uma estrutura hierárquica de dados do qual representa uma estratégia de dividir para conquistar, baseado em busca gulosa. Para seguir a estratégia de dividir para conquistar, as árvores de decisão classificam instâncias de forma a ordenar árvore abaixo, a partir da raiz até alguma folha. Além disso, cada nó da árvore especifica um teste de algum atributo da instância e cada ramo segue um nó correspondente a um dos possíveis valores dos atributos.

O algoritmo base de árvore de decisão, o ID3 (Quinlan, 1986), foi um dos primeiros implementados e segue a premissa de que cada vértice (nó) corresponde a um atributo, e cada aresta da árvore a um valor do possível atributo. Além disso, também segue a lógica de que uma folha da árvore corresponde a um valor esperado da decisão segundo os dados de treino utilizados (classe). A seguir o fluxo que representa os passos do algoritmo ID3.

Algoritmo ID3
<ol style="list-style-type: none"> 1. Seleciona um atributo como sendo o nodo raiz 2. Arcos são criados para todos os diferentes valores do atributo selecionado no passo 1 3. Se todos os exemplos de treinamento (registros) sobre uma folha pertencerem a uma mesma classe, esta folha recebe o nome de classe. Se todas as folhas possuem uma classe, o algoritmo termina 4. Senão, o nodo é determinado com um atributo que não ocorra no trajeto da raiz, e arcos são criados para todos os valores. O algoritmo retorna ao passo 3

Para avaliar qual classe deverá iniciar na raiz, calcula-se a entropia de forma a obter o menor valor médio para os atributos avaliados. A seguir a fórmula da entropia que é utilizada.

$$H = - \sum p(x) \cdot \log p(x) \quad (12)$$

onde H é a entropia e $p(x)$ é a probabilidade do atributo.

Uma evolução do ID3 proposta por Quinlan (1992) é o C4.5, um dos algoritmos utilizados para este trabalho de dissertação. As principais diferenças com relação ao ID3 são:

- Trata atributos categóricos e contínuos;
- Trata valores desconhecidos com “?”, de forma a não utilizar esses valores nos cálculos de ganho e de entropia;
- Utiliza a medida de razão de ganho (medida que se mostrou melhor por gerar árvores mais precisas e menos complexas) para a seleção de atributos que divide os exemplos;
- Trata problemas em que os atributos possuem custos diferenciados;
- Pós-poda.

O algoritmo C4.5 é um dos mais utilizados na literatura para os algoritmos baseados em método de aprendizagem indutiva (Taherkhani, 2010). Na Figura 17 é representado um modelo de árvore de decisão para um jogo de tênis.

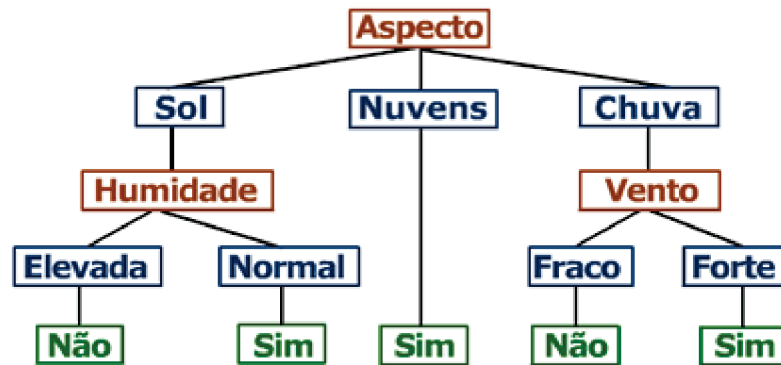


Figura 17 - Modelo de árvore de decisão para jogo de tênis

FONTE: Acervo de GFBioinfo⁹

Conforme pode-se observar, cada nó interno (representado pela cor vermelha), testa um novo atributo para obter o ramo (representado pela cor azul) e, então, obtém-se a folha (cor verde) da qual será a resposta final da classificação da árvore. Por exemplo, caso o aspecto (nó interno) seja “nuvens” a classificação final (folha) será “sim” e, logo, o jogo poderá ocorrer.

3.7 XgBoost

A árvore de decisão tem bom funcionamento em dados com baixo ruído, baixo índice de erro nos dados de treinamento e em casos com parâmetros bem definidos linearmente. Uma grande vantagem das árvores são a rápida classificação (comparando à KNN), classificador ocupando pequeno espaço de memória (comparando à KNN) e relativamente rápido tempo de treinamento (comparando à SVM). Entretanto, poucos conjuntos de dados possuem tais características, tornando, em geral, as árvores de decisão pouco precisas para aplicações reais.

Uma forma de melhorar o índice de acerto são os ensembles, conjunto de classificadores trabalhando em conjunto. A teoria por trás dos ensembles segue dois

⁹ Disponível em:

<<http://web.tecnico.ulisboa.pt/ana.freitas/bioinformatics.ath.cx/bioinformatics.ath.cx/indexf23d.html?id>>

pressupostos: diferentes classificadores cometerão erros em pontos diferentes do espaço amostral e a decisão de um classificador não influencia a decisão de outros. Assim, garante-se estatisticamente que, caso os classificadores sejam capazes de ter um resultado de acerto acima de uma escolha simples aleatória, o ensemble terá um índice de acerto pelo menos igual ao pior classificador.

Existem técnicas de ensembles que buscam otimizar esta teoria para maximizar o índice de acerto na classificação. O ensemble em paralelo segue um sistema simples: cada classificador recebe apenas parte dos dados de treinamento, garantindo generalização da classificação e divergindo as decisões do algoritmo. A cada nova entrada, todos os classificadores são utilizados e um sistema de votação faz a decisão da classificação final. Ou seja, cada classificador seria independente do outro. No caso da árvore de decisão, um dos algoritmos mais famosos de ensemble paralelo é a *Random Forest*.

Outra técnica de ensemble seria a iterativa (boosting). Os classificadores são criados baseado no índice de erro dos classificadores já existentes no ensemble, ou seja, a escolha dos dados que são alimentados como treinamento para o próximo classificador que participará da decisão não será mais randômica, criando ensembles mais especializados.

A XGBoost traz o conceito de ensemble iterativo para um novo nível: além de utilizar a técnica de boosting, todas as árvores são binárias (um contra todos), independentemente do número de classificações, garantindo ainda mais especialização de cada classificador. Um sistema de "poda" garante que cada árvore não tenha tamanho maior que N definido e garante que os resultados sejam generalizados. Este sistema faz a remoção de algumas divisões no final das árvores e compara o índice final de acerto, não causando perdas por se tratar de um número grande de árvores. O ensemble foi implementado por Chen e Guestrin (2016) em C++, mas possui suporte para, também, Python, R, Java, Scala e Julia.

3.8 Máquina de vetor de suporte (SVM)

Os fundamentos de SVM são provenientes da teoria de aprendizagem estatística, também conhecida como teoria de Vapnik e Lerner (1963), da qual explica o processo de aprendizagem do ponto de vista estatístico. Esta teoria consiste em outras quatro teorias para sua implementação: a teoria da consistência

dos processos de aprendizagem, teoria da taxa de convergência dos processos de aprendizagem, teoria da minimização dos risco estrutural e teoria da otimização.

Estas teorias são utilizadas para a criação dos algoritmos baseados em SVM, que é um método de classificação para dados lineares e não-lineares. A SVM utiliza um mapeamento não-linear para transformar os dados originais, etapa de treinamento, em uma dimensão maior. Na nova dimensão formada é buscado um hiper-plano que gere uma separação linear ótima. O processo de separação das classes é formado de forma que o hiper-plano tenha a maior distância possível entre as classes, utilizando vetores suportes e margens definidas pelos vetores suportes. Desta forma, o SVM é resistente ao hiper-ajustamento (overfitting), porque eles aprendem uma superfície de decisão linear específica (hiper-plano marginal máximo) e realizam predições baseadas na seguinte função:

$$y(x) = \sum_{n=0}^n w_n K(x, x_n) + w_0 \quad (13)$$

onde w_n são os pesos do modelo e $k(.,.)$ é a função kernel utilizada.

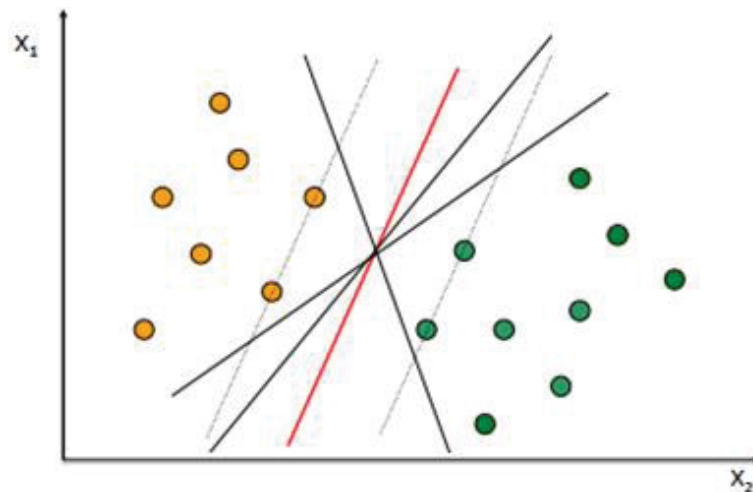


Figura 18 – Hiperplano de separação ótima (na cor vermelha), com os hiperplanos suporte (tracejados)

FONTE: http://ead.liberato.com.br/~mitza/rt_cb_33a.pdf¹⁰

¹⁰ Conteúdo disponível em:
ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia004_1s10/notas_de_aula/topico8_IA004_1s2010.pdf

Conforme pode-se observar na Figura 18, o hiperplano de separação ótima (linha vermelha) foi construído tendo como base dois outros hiperplanos que passam por alguns pontos, denominados vetores de suporte (linhas pontilhadas), em ambas as classes.

3.9 Máquina de vetor de relevância

A máquina de vetor de relevância (RVM, do inglês *relevance vector machine*) é um algoritmo similar ao SVM, entretanto, atribui um tratamento bayesiano a um modelo linear generalizado. Este tratamento diminui algumas desvantagens da SVM, como prevenir os modelos relativamente esparsos, complexidade computacional e decisões binárias para algoritmos de classificação.

O funcionamento do algoritmo RVM (Tipping, 2000) é baseado na probabilidade de um novo ponto pertencer a um grupo. Para isso, é aplicada uma função sigmoide logística para obter o modelo linear da saída encontrada através da equação 14 [$y(x)$], conforme equação a seguir:

$$\sigma(y) = 1/(1 + e^{-y}) \quad (14)$$

Em seguida é calculada a verossimilhança dos pontos a serem classificados.

$$P(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^N \sigma(y(\mathbf{X}_n; \mathbf{w}))^{t_n} [1 - \sigma(y(\mathbf{X}_n; \mathbf{w}))]^{1-t_n} \quad (15)$$

Não é possível integrar os pesos analiticamente para obter a verossimilhança marginal $P(\mathbf{t}|\alpha)$. Desta forma, torna-se necessário utilizar um procedimento utilizando outra aproximação (MacKay, 1992), onde são encontrados os pesos mais prováveis do modelo (moda).

$$\nabla_w \nabla_w \log P(w|t, \alpha)|_{w_{MP}} = -(\Phi^T B \Phi + A)^{-1} \quad (16)$$

onde $B = \sigma\{y(\mathbf{X}_n)\}[1 - \sigma\{y(\mathbf{X}_n)\}]$ e $A = \text{diag}(\alpha_1, \alpha_2, \dots, \alpha_N)$. Desta forma, é possível escrever as seguintes expressões:

$$\sum = (\Phi^T B \Phi + A)^{-1} \quad (17)$$

$$w_{MP} = \sum \Phi^T B t \quad (18)$$

Por fim, com as equações 8 e 9, faz-se possível estimar o valor de saída (y) para cada valor de entrada x' .

$$y = w_{MP}^T \Phi(x') \quad (19)$$

Este procedimento é repetido até que uma convergência adequada seja satisfeita. A seguir são demonstrados os algoritmos para treinamento e classificação da RVM (Mehrotra et al., 2012).

Algoritmo para treinamento da RVM

Entrada: x : Matriz de entrada de N instâncias com dimensão d

t : valores de saída equivalente

Saída: R : Modelo de vetores de relevância
 W_{mp} : matriz de pesos mais prováveis

Gerar $\Phi = [\phi_1(x), \phi_2(x), \dots, \phi_N(x)]$

Inicializar δ_T // limiar de convergência de

//inicialização de hiper-parâmetros

$$\alpha \leftarrow \left(\frac{1}{N}\right)^2$$

$$\beta \leftarrow 0$$

Executar

$$A = \text{diag}(\alpha), B = \text{diag}(\beta)$$

$$\Sigma = (\Phi^T B \Phi + A)^{-1}$$

$$w_{MP} = \Sigma \Phi^T B t$$

$$\max |L(\alpha) = -\frac{1}{2} [N \log 2\pi + \log |C| + t^T C^{-1} t]|$$

$$C_{-i} = C - \alpha_i^{-1} \phi_i \phi_i^T$$

$$s_i = \phi(x_i)^T C_{-i}^{-1} \phi(x_i)$$

$$q_i = \phi(x_i)^T C_{-i}^{-1} t$$

Se $q_i^2 > s_i$ **então**

Adiciona x_i para R **se** $x_i \notin R$

$$\alpha_i = \frac{s_i^2}{q_i^2 - s_i}$$

Senão

Remove x_i de R **se** $x_i \in R$

$$\alpha_i = \infty$$

Fim

Atualiza β

$$\delta = \sum_{i=1}^n \alpha_i^{n+1} - \alpha_i^n$$

Enquanto $\delta < \delta_T$

Algoritmo para classificação da RVM

Entrada: x' : Matriz de teste para classificação

R : modelo de vetores de relevância

W_{mp} : matriz de pesos mais prováveis

Saída: y : valor da classe previsto

$r \leftarrow |R|$ // número de vetores de relevância

// gerar a matriz com os modelos de relevância

$$\phi(x') \leftarrow [\phi_1(x'), \phi_2(x'), \dots, \phi_r(x')]$$

$$y = w_{MP}^T \phi(x')$$

Na Figura 19 é possível visualizar uma comparação entre um modelo svm, figura da esquerda, com um umdelo RVM, figura da direita.

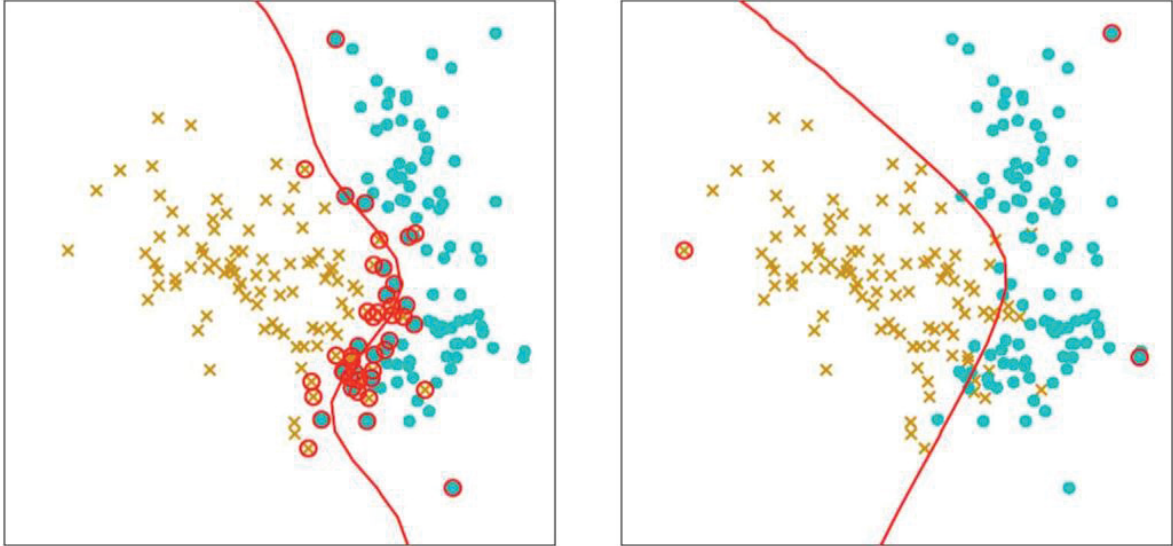


Figura 19 - Comparação de um modelo SVM com um modelo RVM

FONTE: imagem da internet¹¹

Neste exemplo da Figura 19 é possível visualizar que o modelo RVM obteve um desempenho muito próximo do desempenho da SVM. Além disso, como o cálculo realizado para a RVM (cálculo estatístico) é diferente do cálculo da SVM, foram utilizados apenas 3 vetores de suporte (pontos circulado em vermelho) para a RVM e para a SVM foram utilizados 44 vetores de suporte. Isto mostra que, neste exemplo, o modelo de RVM obteve um desempenho computacionalmente menos complexo e com uma melhor eficiência.

3.10 Processo Pêndulo

Na mecânica, pêndulo é um mecanismo que oscila em torno de um ponto fixo, movimentando-se no plano xy . Quando um pêndulo está parado, sem nenhuma excitação, ele está em uma posição de equilíbrio. Por outro lado, quando é submetido a alguma excitação, ele realiza movimentos oscilatórios caracterizados por uma função *seno*, realizando movimentos alternados em torno da posição de equilíbrio.

¹¹ Disponível em: <http://images.slideplayer.com/29/9428537/slides/slide_11.jpg>

Na prática, não existem pêndulos que não sejam amortecidos, isto se dá ao fato de que não existam pêndulos perfeitos (aqueles que não sofrem influência de forças contrárias que o façam parar, como por exemplo, o atrito). Assim todo e qualquer pêndulo real, é um PAM.

Segundo Lemes et al. (2010) o PAM é um experimento para auxiliar no estudo e na avaliação de projetos de controladores digitais. As características dinâmicas não-lineares e visuais deste sistema mecatrônico motivam o emprego no estudo do controle da posição com controle por realimentação (Bucher e Balemi, 2008).

Os pêndulos são utilizados há muitos séculos como objeto de estudo, aplicado a inúmeros casos científicos e acadêmicos por seus diferentes tipos e particularidades. Além de sua utilização típica em disciplinas e cursos de física, os pêndulos são requisitados na engenharia, cursos de tecnologia e outras ciências aplicadas, tanto por possibilitar testes e aplicações de algoritmos de controle como também para comparar respostas práticas com as obtidas na teoria.

Quando o pendulo está parado, ele se mantém em equilíbrio, sem apresentar nenhuma oscilação. Com a realização do movimento de giro (propulsão) gerado através da hélice do motor, é adicionada uma excitação ao sistema, resultando na alteração do ângulo Θ , caracterizado por uma função *seno*, implicando assim diretamente na variação de sua amplitude, conforme apresentado na Figura 20, com um exemplo de resposta oscilatória de um processo pêndulo.

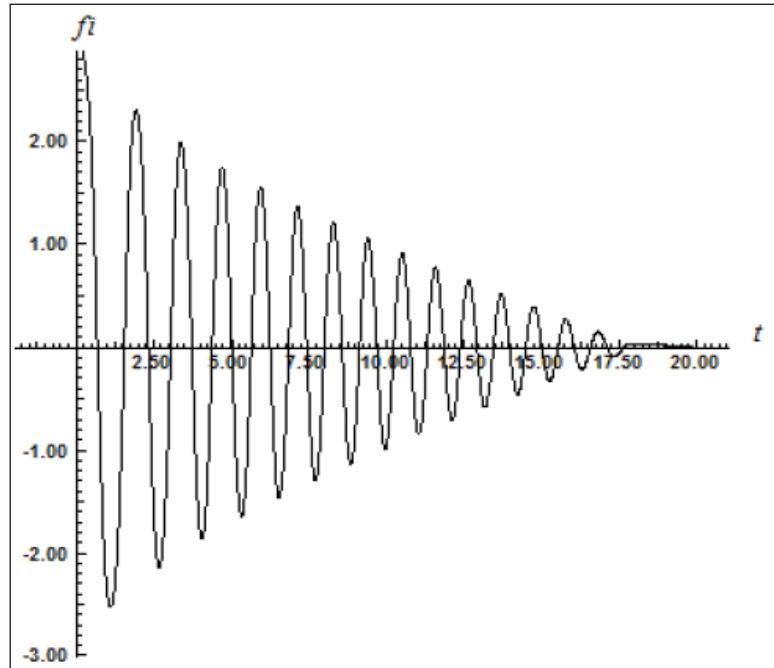


Figura 20 - Gráfico da posição angular em função do tempo para um pêndulo físico por um momento de atrito constante

FONTE: http://ead.liberato.com.br/~mitza/rt_cb_33a.pdf¹²

Todos os pêndulos em prática (Li et al., 2013; Medina et al., 2005) sofrem influência do meio, seja este em forma de atrito ou mesmo a resistência com o ar ou fluido que esteja envolto.

3.11 Controlador PID Dahlin Adaptativo

Em DAHLIN (1968), foi proposta uma estratégia de controle que possui como objetivo tratar a dinâmica do sistema em malha fechada como um sistema de primeira ordem com atraso de transporte ao invés de uma entrada com atraso (sinal de entrada de um sistema sendo um degrau, por exemplo, atrasado em X instantes). A resposta a uma entrada do tipo degrau de um sistema de primeira ordem, com um ganho unitário e um tempo morto de $(f + 1)$ períodos é dada pela equação (20.10).

$$y(Z^{-1}) = \frac{(1-\lambda)*(Z^{-f-1})}{(1-\lambda*Z^{-1})*(1-Z^{-1})}, \quad (20.10)$$

¹² Conteúdo disponível em: http://ead.liberato.com.br/~mitza/rt_cb_33a.pdf, acessado em novembro de 2015.

Nas equações 20.11, 20.12, 20.13 e 20.14, serão detalhados os elementos da equação (20.10).

A equação (20.11) representa o tempo morto da resposta. A equação (20.12), por sua vez, demonstra a mudança do degrau unitário. Já a equação (20.13) indica que a resposta é de primeiro grau e a equação (20.14) presente no numerador da equação (20.10) é o que tenta forçar o ganho unitário da resposta do sistema.

$$Z^{-f-1} \quad (20.11)$$

$$\frac{1}{(1 - Z^{-1})} \quad (20.12)$$

$$(1 - \lambda * Z^{-1}) \quad (20.13)$$

$$(1 - \lambda) \quad (20.14)$$

A estratégia proposta por Dahlin possui prós e contras, como qualquer estratégia. Uma das desvantagens deste método é o fato de não possuir um bom desempenho frente a sistemas com fase não mínima. Apesar disso, responde bem a sistemas com atraso de transporte, sistemas analiticamente simples e ao tempo de estabilização do sistema em relação ao *set-point* desejado variando apenas um único parâmetro de projeto (Souza, 2010).

Visando reduzir o erro em estado estacionário, a saída do sistema empregando o controlador proposto por Dahlin é detalhada na equação (20.18), onde Ni (20.15) é um inteiro truncado, τ_t é o atraso de transporte, T_s é o período de amostragem, p (20.16), λ (20.17) sendo este o parâmetro de ajuste do método de Dahlin e τ é a constante de tempo do sistema.

$$Ni = \frac{\tau_t}{T_s} \quad (20.15)$$

$$p = e^{-\lambda * T_s} \quad (20.16)$$

$$\lambda = \frac{1}{\tau} \quad (20.17)$$

$$y[t] = p * y[t - 1] + (1 - p) * u[t - N - 1] \quad (20.18)$$

Para realizar o ajuste do controlador de Dahlin, deve-se levar em consideração qual é a velocidade de reposta do sistema desejada. Levando em consideração que, na teoria, λ é o parâmetro de ajuste do controlador, ao variar o valor de τ , o valor de λ é inversamente afetado, portanto para a sintonia do controlador de Dahlin, deve-se variar τ tendo em vista que:

- Ao diminuir o valor de τ , o valor de λ aumenta, com isso o valor de p tende a zero e isso faz com que o controle do sistema torne-se mais rápido.
- Ao aumentar o valor de τ , o valor de λ diminui, com isso o valor de p tende a um, fazendo com que o controle torne-se mais lento.
- Zerando o valor de λ , tendendo o valor de τ ao infinito, o controlador de Dahlin passa a ser um controlador *Dead Beat*.

Assim, variando o valor de τ , o valor de p também é alterado, percebendo-se que a velocidade de resposta do sistema está diretamente ligada ao valor de p , assim, na prática altera-se simplesmente o valor deste, pois possui uma faixa de valores aceitados limitada entre zero e um - valores definidos pela curva exponencial negativa. Em vista disso, passa-se a tratar p como o parâmetro de Dahlin, ao invés de λ (Meira, 2010).

O método de Dahlin, em malha fechada, pode ser expresso da seguinte maneira (Bobàl et al., 2005):

$$\frac{Y}{U} = \frac{(1 - p) * Z^{-N-1}}{1 - p * Z^{-1}} \quad (20.19)$$

Convertendo a equação (20.19) de malha fechada para malha aberta (Zafiriou e Morari 1985) e aproximando-a com uma modelagem matemática de segunda ordem, encontra-se a equação do controlador (20.20), tal que:

$$G(Z) = \frac{1 + a_1 * Z^{-1} + a_2 * Z^{-2}}{b_0 * Z^{-1} + b_1 * Z^{-2}} * \frac{(1-p) * Z^{-N-1}}{(1-p * Z^{-1})} \quad (20.20)$$

Para evitar a presença de oscilações (também denominado *ringing*) no sinal de controle, a equação (20.20) é modificada e fica conforme a equação (20.21) (Ho et al., 1999).

$$G(Z) = \frac{1 + a_1 * Z^{-1} + a_2 * Z^{-2}}{1 - p * Z^{-1} - (1-p) * Z^{-N-1}} * \frac{(1-p)}{(b_0 + b_1)} \quad (20.21)$$

Para encontrar os valores dos ganhos do controlador (K_p , K_i e K_d), deve-se primeiro considerar a função transferência do controlador PID digital na forma ideal (20.22).

$$G_c(Z) = K_c * \left[\frac{\left(1 + \frac{T_s}{T_i} + \frac{T_d}{T_s}\right) - \left(1 + 2 * \frac{T_d}{T_s}\right) * Z^{-1} + \left(\frac{T_d}{T_s}\right) * Z^{-2}}{(1 - Z^{-1})} \right] \quad (20.22)$$

Relacionando as equações (20.21) e (20.22), encontram-se as seguintes relações:

$$K_{aux} = \frac{(1 - p)}{(b_0 + b_1) * (1 + N * (1 - p))} \quad (20.23)$$

$$K_c = K_{aux} * (a_1 + 2 * a_2) \quad (20.24)$$

$$T_i = - \frac{T_s * (a_1 + 2 * a_2)}{(1 + a_1 + 2 * a_2)} \quad (20.25)$$

$$T_d = -\frac{a_2 * T_s}{(a_1 + 2 * a_2)} \quad (20.26)$$

Para mais detalhes sobre o procedimento do cálculo dos ganhos do controlador PID de Dahlin, ver referência (Ho et al., 1999).

4 MATERIAIS E MÉTODOS

Nesta seção são contempladas as principais tecnologias que foram escolhidas da melhor forma possível para o desenvolvimento do projeto e como estas tecnologias foram utilizadas.

4.1 MATERIAIS

Os materiais utilizados são: um *neuroheadset* para a leitura do nível dos sinais eletro encefálicos do cérebro humano, o circuito de ligação do sistema pêndulo, que será controlado utilizando os sinais do EEG, e por fim, os algoritmos de processamento.

4.1.1 Neuroheadset

A fim de captar os sinais elétricos gerados pelos estímulos dos neurônios do nosso cérebro, utilizaremos um capacete que possui a capacidade de captação do sinal de EEG. Pelo estudo realizado por Stamps e Hamam (2010), podemos perceber, conforme a Figura 21, que o *EMOTIV* teve maior pontuação entre os sistemas avaliados. Isso ocorre por possuir um rápido tempo de resposta, excelente resolução de dados, ser não invasivo e ter uma interface simplificada e agradável.

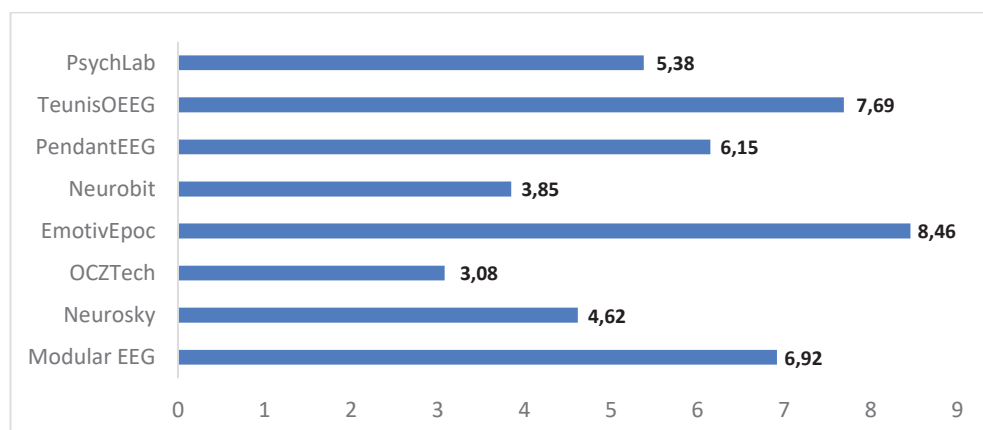


Figura 21 - Comparativo custo x benefício de eletroencefalogramas

FONTE: O autor (2017)

Portanto, para este projeto será utilizado o *neuroheadset* denominado *EMOTIV EPOC*. Seus detalhes de organização e funcionamento são demonstrados a seguir.

4.1.1.1 Posicionamento de Eletrodos

A Federação Internacional de Neuropsicologia recomenda a utilização do modelo 10-20 para até 21 eletrodos de acordo com a Figura 4. Neste modelo, todos os eletrodos se encontram a uma distância de seus eletrodos vizinhos de 10% ou 20% da distância total da cabeça. A cada posição é dado um identificador de uma ou duas letras e um número de acordo com sua proximidade a alguma parte do cérebro. Números pares se encontram à direita e números ímpares a esquerda.

A sociedade americana de EEG sugeriu que novos eletrodos de maneira equidistante dos 21 eletrodos já definidos na Figura 4 na sessão 3.1.1, caso seja necessário um número maior de eletrodos.

4.1.1.2 *EMOTIV* Epoc Headset

O *EMOTIV* Epoc Headset, Figura 22, é um *hardware* desenvolvido pela *EMOTIV* Systems que tem como proposta ser o dispositivo para controle de videogames da próxima geração.



Figura 22 - Dispositivo *EMOTIV* EPOC

FONTE: Portal Online *emotiv.com/epoc/* (2017)¹³

¹³ Disponível em: <https://www.emotiv.com/wp-content/uploads/2016/02/emotiv_epoc_600.png>

O dispositivo *EMOTIV* está equipado com 14 eletrodos, conforme ilustrado na Figura 23, fixos nas posições AF3, AF4, F3, F4, F7, F8, FC5, FC6, P7, P8, T7, T8, O1, O2 e mais dois eletrodos, CMS e DRL utilizados como sensores de referência de posicionamento, e também funcionam com o objetivo de reduzir o ruído adequado o posicionamento dos demais eletrodos.

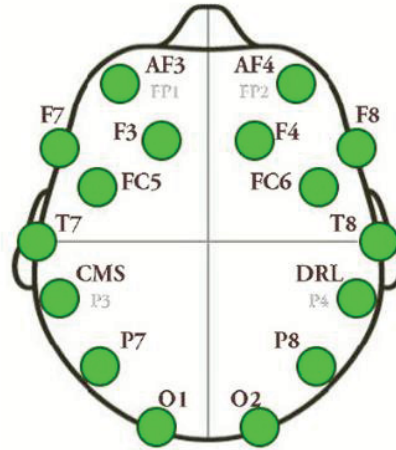


Figura 23 - Posicionamento dos eletrodos do *EMOTIV*

FONTE: Imagem da internet (2016)¹⁴

O *EMOTIV* amostra os sinais de EEG a uma taxa de 2048Hz (sinal interno), que é convertida para uma frequência de amostragem de 128Hz (utilizada neste trabalho), com o tempo de coleta a cada 7,18 ms ou de 256Hz, com tempo de coleta a cada 3,91 ms. O seguinte pré-processamento é realizado diretamente no controlador neural antes de ser transmitido a uma determinada aplicação:

- Filtro com frequência de corte de 85 Hz;
- Filtro com frequência de corte de 0.16 Hz;
- Filtro com rejeição em torno de 50 Hz e 60 Hz.

Ao utilizar o dispositivo, é necessário ungir os eletrodos do dispositivo com algumas gotas de alguma solução salina. Após deixar os eletrodos úmidos, o *EMOTIV* reconhecerá os sensores de modo a obter a melhor qualidade dos sinais.

O *neuroheadset EMOTIV* também possui uma ótima resposta para a captura de dados, comunicação wireless (Bluetooth® Smart), presença de múltiplos canais e tecnologia não-invasiva. Além disso, o eletroencefalograma vem equipado com licenças para o uso de *softwares* próprios do produto, denominados “*EMOTIV EPOC Control Panel*”, “*EmoComposer*” e o “*EmoKey*”.

¹⁴ Disponível em: <http://web.snauka.ru/wp-content/uploads/2013/09/091913_1211_1.png>

4.1.2 Pêndulo

O pêndulo será o processo que será controlado pelo EEG e consiste em uma haste rígida de material leve, suspensa em seu eixo de giro a um potenciômetro o qual é fixado a uma estrutura que sustenta o conjunto. Na haste é adaptado um motor com hélice utilizado para realizar a propulsão de giro, movimentando-se bidimensionalmente no plano xy .

Na Figura 24, pode ser visualizada a descrição das partes principais que compõe o PAM. Uma hélice acoplada a um motor (propulsor – item *a*) está acoplada na extremidade da haste girante (item *b*), a barra está fixada a um potenciômetro em seu centro (item *c*) e um contrapeso na outra extremidade da haste (item *d*) é utilizado para balancear o peso do pêndulo.

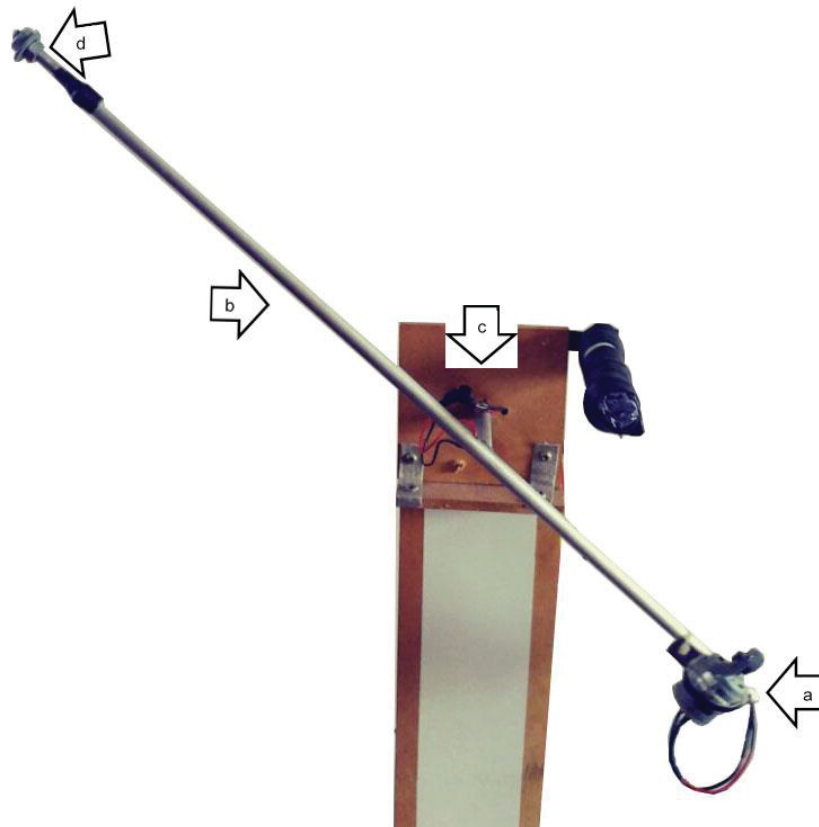


Figura 24 - Protótipo do pêndulo

FONTE: O autor (2017)

4.1.2.1 Arduino Mega 2560

Para a leitura dos sinais do projeto, foi utilizado o Arduino MEGA 2560, a Figura 25, mostra um detalhamento maior do microcontrolador que tem sua placa baseado no ATmega2560.

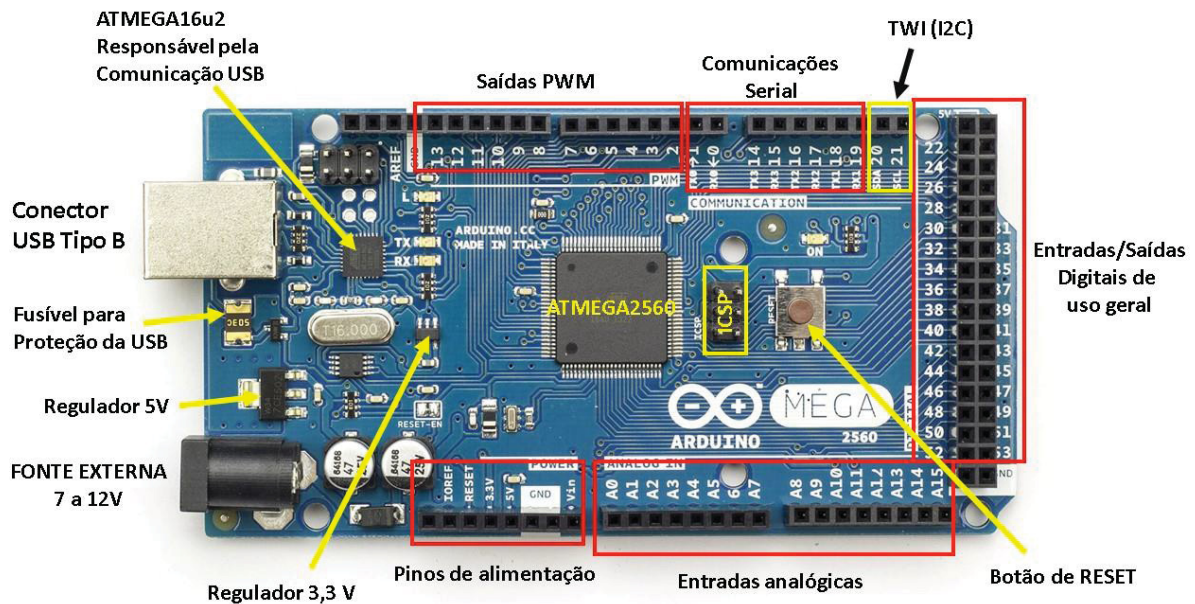


Figura 25 - Arduino Mega 2560

FONTE: Imagem da Internet (2017)¹⁵

A placa possui 54 pinos de entradas/saídas digitais, 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, uma entrada de alimentação, uma conexão ICSP e um botão de reset.

¹⁵ Disponível em: <<https://www.embarcados.com.br/wp-content/uploads/2014/04/Arduino-Mega-2560-recursos.jpg>>

4.1.3 Configurações do computador utilizado

Para o desenvolvimento da dissertação e para a execução dos algoritmos utilizados em todas as etapas do projeto (aquisição de dados, treinamento, classificação, controle do pêndulo), foi utilizado um computador com as seguintes configurações:

Sistema operacional: Windows 10.

Processador: Intel Core™ i7-5500U 2.4GHz.

Memória RAM (do inglês Random Access Memory): 8GB.

Placa de vídeo: NVIDIA GeForce 920M com 2GB.

4.2 MÉTODOS

A seguir são explicados os métodos utilizados para a captura de sinais do EEG, os algoritmos de processamento para executar a classificação dos sinais eletro encefálicos e o controle do pendulo.

4.2.1 Aquisições dos Sinais Eletroencefálicos

Para o desenvolvimento do projeto, o primeiro passo é referente a aquisição dos dados utilizando o eletroencefalograma *EMOTIV*. Para realizar a aquisição, o usuário deve pensar nos movimentos que deseja executar de maneira concentrada e, de preferência, sem que haja distrações. Por exemplo, ao pensar no movimento “ligar motor do pêndulo em 48°”, que o pêndulo irá executar, o usuário deverá se concentrar ao máximo pensando nessa ação, ou no pensamento proporcional a esta ação, sem que haja interferências do mundo externo. Sendo assim, quanto mais concentração, menores são os ruídos gerados no processo de aquisição. Outro ponto importante são as “*dlls*” do *EMOTIV*, que são o conjunto de bibliotecas que as funções precisam acessar para conseguir capturar as informações em tempo real.

Para a aquisição de dados do EEG, foi implementado um código em MATLAB (MATHWORKS, INC, 2015) de forma a capturar os valores das sinapses que ocorrem em cada sensor do EMOTIV. Os dados adquiridos do *EMOTIV* são colocados em um formato de matriz. Sendo assim, é possível obter para cada t instante de tempo os valores do conjunto sensores. Os dados são dispostos conforme matriz da Tabela 1. Desta forma, como existem 14 sensores que capturam os sinais sinápticos e 2 sensores com valor do giroscópio (DRL e CMS), gerou-se uma matriz com 14 colunas (o número de linhas irá variar de acordo com o tempo de aquisição), entretanto, os valores dos sensores DRL e CMS foram desconsiderados em vista de que não são utilizados valores de posição para a classificação dos dados.

	EEG1	EEG2	EEG3	...	EEG12	EEG13	EEG14
1	4147,18	4614,87	4451,28	3968,21	4151,79	4433,85	4327,18
2	4142,05	4617,44	4448,72	3969,23	4145,64	4423,59	4291,28
3	4140,51	4618,46	4450,77	3969,23	4148,72	4426,15	4310,77
4	4144,1	4616,41	4454,87	3969,23	4152,31	4433,33	4329,23
N

Tabela 1 - Dados capturados utilizando o EMOTIV com MATLAB
Fonte: O autor (2017)

Em seguida, após o processo de criação do algoritmo de captura de sinal, foram coletados dados de cinco pessoas diferentes de forma que a base de dados gerada fosse robusta o suficiente para a etapa de treinamento, haja vista que, cada pessoa pensa diferente da outra e os valores das sinapses podem alterar para o mesmo tipo de pensamento. As coletas foram realizadas com três diferentes padrões de pensamento, sendo eles:

- Pêndulo desligado: usuário é instruído a se concentrar em um objeto (preferencialmente um motor) parado, sem nenhuma ação e, tampouco, movimento;
- Pêndulo em 45°: usuário é instruído a se concentrar em um objeto (preferencialmente um motor) ligado, dando a partida e se movimentando de forma constante sem muita velocidade. Para facilitar o processo de aquisição, o usuário pode visualizar o pêndulo na posição de 45° enquanto pensa no movimento do motor;
- Pêndulo em 48°: usuário é instruído a se concentrar em um objeto (preferencialmente um motor) ligado com velocidade duas vezes maior do que o pensamento do ângulo de 45°. Para facilitar o processo de aquisição, o usuário pode visualizar o pêndulo na posição de 48° enquanto pensa no movimento do motor.

Cada pensamento foi extraído durante o tempo de 20 segundos. Onde o usuário foi instruído a se concentrar, com o mínimo possível de variação do pensamento, na ação que executaria durante este tempo.

Após a coleta dos cinco indivíduos faz-se necessária a construção da matriz de dados que servirá como entrada dos algoritmos na etapa de treinamento e classificação dos dados. Para a construção da matriz de dados a partir dos dados

coletados do EEG, os dados de todos os usuários foram agrupados em uma única matriz e, em seguida, as linhas foram reordenadas. Desta forma, pode-se afirmar que não há distinção entre os dados que serão classificados dos usuários, haja vista que, a medida que os dados são combinados, não é mais possível identificar a quem eles pertencem.

Com a matriz de dados construída, o objetivo é determinar um valor específico para cada tipo de ação que será executado pelo pêndulo. Desta forma, faz-se necessário aplicar um algoritmo para a classificação dos dados coletados. O algoritmo irá determinar qual o resultado para o problema e os dados são coletados de forma a representarem o pensamento do usuário, e, em seguida, são classificados em movimentos reais para a ação do pêndulo. No APÊNDICE A, para entender um pouco mais do funcionamento do EEG e das ondas cerebrais que podem ser coletadas, é possível visualizar diversos gráficos (*Tempo x Valores*) relacionados a coletas de dados utilizando o EMOTIV com padrões relativos a atividades motoras sem utilizar apenas o pensamento.

4.2.2 Validação dos Dados

De forma a obter uma melhor compreensão dos dados capturados do EEG, é possível obter uma visualização detalhada da distribuição dos valores coletados pelos sensores, com o auxílio do gráfico de violino plotado na ferramenta MATLAB. Na Figura 26, foram plotados os valores de amplitude, com o sinal já filtrado, para todos os sensores do EMOTIV (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) onde cada cor representa um padrão diferente de pensamento (motor desligado, motor ligado em 45° e motor ligado em 48°). A cor azul representa o pensamento “motor desligado”, onde foi atribuído para o pêndulo desligado, a cor preta representa um pensamento de “motor ligado em 45°”, a amarela representa o pensamento “motor ligado em 48°”.

Para um melhor entendimento da distribuição dos valores de amplitudes, também foram plotados, para cada padrão de pensamento e sensor, os valores de média (cruz vermelha) e mediana (quadrado verde).

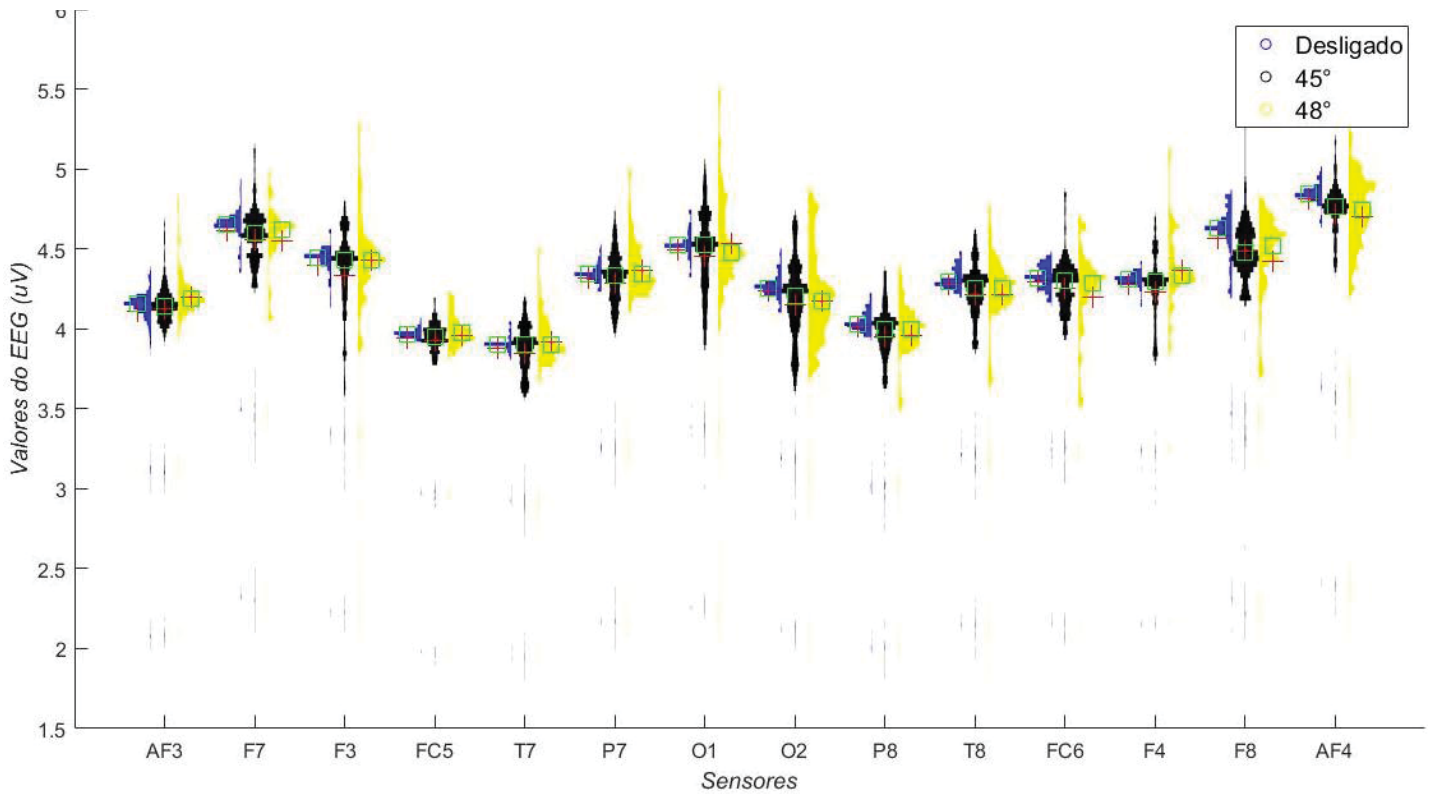


Figura 26 – Gráfico de violino plotado na ferramenta MATLAB dos sensores do *EMOTIV*

FONTE: O autor (2017)

Conforme pode-se observar na Figura 26, a maior parte dos valores ficam concentrados em uma parte e depois se dissipam. A maior parte dos valores da amostra pode ser vista quando o gráfico está com uma largura maior no eixo x. No eixo y este valor representa a amplitude em μV dos sensores analisados. Logo, obtém-se valores concentrados em uma faixa de valores, onde torna-se possível analisar esse intervalo de valores e descartar valores que não fazem parte deste intervalo. Este processo facilita a classificação e diminui a probabilidade de ruídos nos dados, obtendo menores valores de erro e, desta forma, um controle melhor do processo utilizado. Na Tabela 2 pode-se observar o intervalo de valores utilizado para cada sensor, onde os valores que se distanciam de forma considerável dos valores da maior parte da amostra são descartados.

Sensores	Intervalo - pêndulo desligado	Intervalo - pêndulo 45 graus	Intervalo - pêndulo 48 graus
AF3	$[3,93 - 4,37] \in \mathbb{R}$	$[3,90 - 4,42] \in \mathbb{R}$	$[3,98 - 4,43] \in \mathbb{R}$
F7	$[4,45 - 4,77] \in \mathbb{R}$	$[4,35 - 4,88] \in \mathbb{R}$	$[4,13 - 4,79] \in \mathbb{R}$
F3	$[4,20 - 4,57] \in \mathbb{R}$	$[3,93 - 4,67] \in \mathbb{R}$	$[3,93 - 4,73] \in \mathbb{R}$
FC5	$[3,92 - 4,05] \in \mathbb{R}$	$[3,77 - 4,10] \in \mathbb{R}$	$[3,83 - 4,24] \in \mathbb{R}$

T7	$[3,81 - 4,85] \in \mathbb{R}$	$[3,57 - 4,20] \in \mathbb{R}$	$[3,65 - 4,20] \in \mathbb{R}$
P7	$[4,23 - 4,50] \in \mathbb{R}$	$[4,00 - 4,70] \in \mathbb{R}$	$[4,10 - 4,75] \in \mathbb{R}$
O1	$[4,32 - 4,74] \in \mathbb{R}$	$[3,99 - 4,99] \in \mathbb{R}$	$[4,13 - 5,00] \in \mathbb{R}$
O2	$[4,12 - 4,47] \in \mathbb{R}$	$[3,78 - 4,68] \in \mathbb{R}$	$[3,72 - 4,87] \in \mathbb{R}$
P8	$[3,95 - 4,20] \in \mathbb{R}$	$[3,65 - 4,36] \in \mathbb{R}$	$[3,54 - 4,39] \in \mathbb{R}$
T8	$[4,19 - 4,48] \in \mathbb{R}$	$[3,87 - 4,62] \in \mathbb{R}$	$[3,66 - 4,67] \in \mathbb{R}$
FC6	$[4,19 - 4,46] \in \mathbb{R}$	$[3,98 - 4,49] \in \mathbb{R}$	$[3,90 - 4,58] \in \mathbb{R}$
F4	$[4,16 - 4,37] \in \mathbb{R}$	$[4,06 - 4,47] \in \mathbb{R}$	$[3,95 - 4,59] \in \mathbb{R}$
F8	$[4,22 - 4,84] \in \mathbb{R}$	$[4,14 - 4,88] \in \mathbb{R}$	$[3,84 - 4,79] \in \mathbb{R}$
AF4	$[4,66 - 4,96] \in \mathbb{R}$	$[4,38 - 5,03] \in \mathbb{R}$	$[4,27 - 5,15] \in \mathbb{R}$

Tabela 2 - Intervalo de valores utilizados para a coleta de dados

Com isso, quando os valores dos eletrodos ficarem fora dos intervalos da Tabela 2, os mesmos serão desconsiderados.

4.2.3 Classificação dos Dados

Para o processo de classificação dos dados, foram utilizados os padrões de pensamentos citados anteriormente sendo estes com as classes pêndulo desligado, pêndulo em 45° e pêndulo em 48° , o que caracteriza este problema como um problema de classificação multi-classes. Neste processo, o primeiro passo é realizar o treinamento dos dados e, em seguida, a classificação em tempo real para o controle do pêndulo. Para o treinamento dos algoritmos de classificação, foram utilizados como entrada dos algoritmos os dados obtidos das cinco pessoas.

Primeiramente, ocorreu um tratamento nos dados de entrada com uma suavização dos sinais do EEG. Para isto, utilizou-se o filtro de Delorme e Makeig (2004) denominado “*EEGFILT*”. Este filtro, dependendo do sinal obtido do EEG, funciona como filtro passa alta, passa baixa e passa faixa.

Para obter o tratamento dos sinais do EEG, foram utilizados como parâmetros de entrada para a função do filtro, implementada em MATLAB, os dados do EEG, a frequência de amostragem utilizada (128 Hz), os tipos de filtros utilizados para o tratamento do sinal do EEG, neste caso o filtro utilizado foi o passa faixa e o número de quadros por época. A partir destes parâmetros, obtém-se os valores de saída em forma de matriz ([0.0157 0.0157]).

Em seguida, utiliza-se o comando “*filtfilt*”, filtro nativo do MATLAB, com a matriz, obtida anteriormente no filtro de Delorme e Makeig (2004), sendo utilizada

como um dos parâmetros. Além da matriz, também é necessário determinar o fator de multiplicação do filtro, valor pelo qual todos os dados serão multiplicados após o processo de filtragem (será utilizado o valor igual a um) e, por fim, os dados do EEG que serão suavizados. Desta forma, é possível aplicar nos valores de saída do EEG o filtro e perceber uma leve suavização do sinal, conforme Figura 27 e Figura 28.

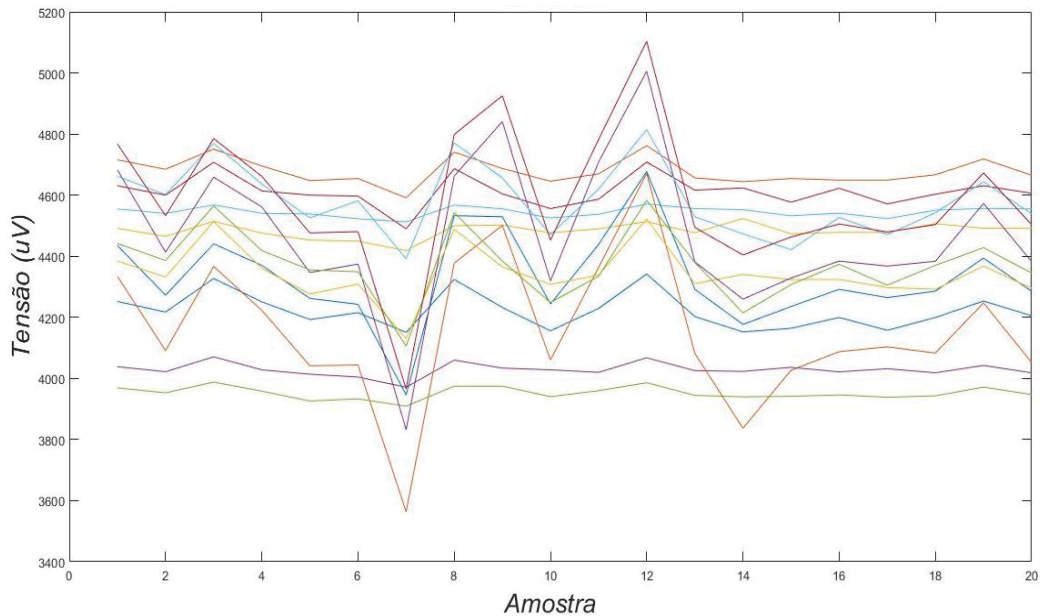


Figura 27 - Amostra do sinal do EEG sem suavização

FONTE: O autor (2017)

Em geral o filtro suavizou alguns sinais em alta frequência e reduziu alguns picos no sinal. Estes picos acentuados do sinal podem gerar erros processo de classificação dos dados. Este processo de filtragem se faz necessário para diminuir esse valor alto do pico (que pode ser considerado um ruído) e, desta forma, melhorar a classificação dos dados e o controle do pêndulo.

Desta forma, o sinal filtrado sofreu um número reduzido de variações do que o sinal classificado não filtrado, e como este sinal será enviado para o *pêndulo* em forma de comando, se obteve como resultado um controle mais preciso com um número reduzido de oscilações entre o estado ligado e desligado do pêndulo.

O filtro foi aplicado para a etapa de treinamento, validação e classificação em tempo real dos dados obtidos do EEG.

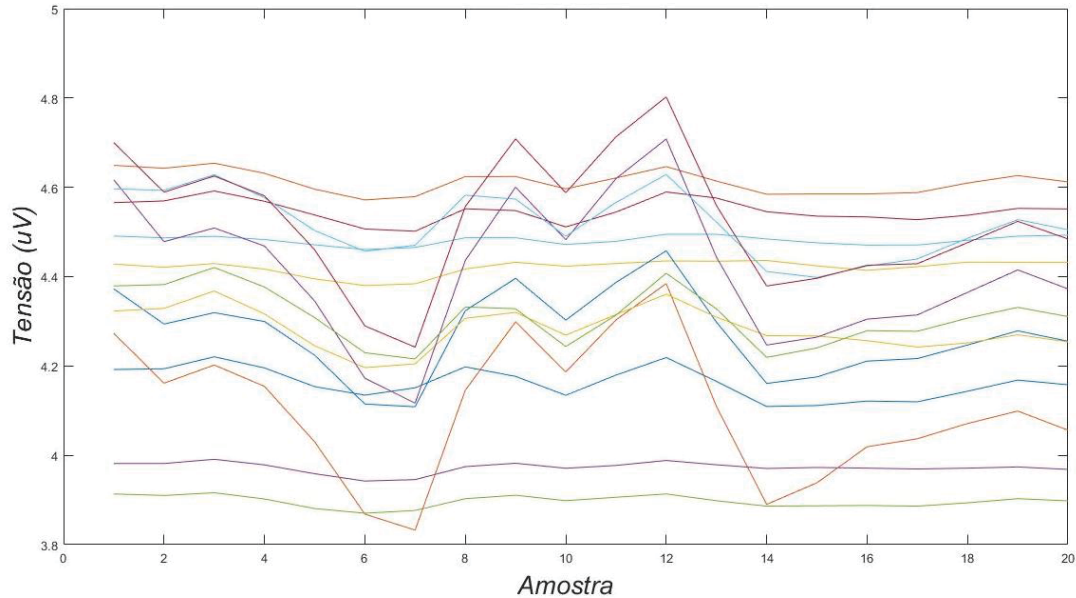


Figura 28 - Amostra do sinal do EEG com suavização

FONTE: O autor (2017)

Na Figura 28 pode-se observar que o sinal se comportou de maneira mais suave sem alto picos. Além disso, no processo de filtragem todos os valores da matriz são divididos por 1000. Esta divisão ocorre porque os valores obtidos sem o filtro são valores redondos altos (na escala de milhar), sendo estes desnecessários no processo de classificação dos dados e computacionalmente mais custosos.

Em seguida, os dados foram separados em 70% para treinamento e 30% para validação. Após rodar os algoritmos, com os valores de treinamento são obtidos os modelos dos algoritmos e com os valores de validação, gera-se os resultados de acurácia da matriz de confusão (ferramenta para avaliação de modelos estatísticos como os modelos de aprendizado de máquina), encontrados na sessão de análise dos resultados e discussão.

4.2.4 Controle do Pêndulo

Para o controle do pêndulo o sinal de acionamento do propulsor é gerado pelo controlador implementado no MATLAB (Dahlin adaptativo) e é enviado ao propulsor por meio de um sinal PWM (*Pulse Width Modulation* - Modulação por largura de pulso) gerado pelo conversor DA (Arduino). O Esquema elétrico do projeto do pêndulo pode ser visto no APÊNDICE C.

4.2.5 Integração Final

Por fim, com todos os módulos configurados individualmente, implementa-se a solução final para integração entre todos os módulos do projeto. Para isto, foi implementado um código em MATLAB que integra todos os módulos.

5 ANÁLISE DOS RESULTADOS E DISCUSSÃO

Na etapa de aquisição dos sinais do EEG, com os sinais dos sensores de posição fora da matriz de dados gerada (Tabela 1), foram atribuídos como entrada dos algoritmos de classificação os sinais conforme Figura 29.

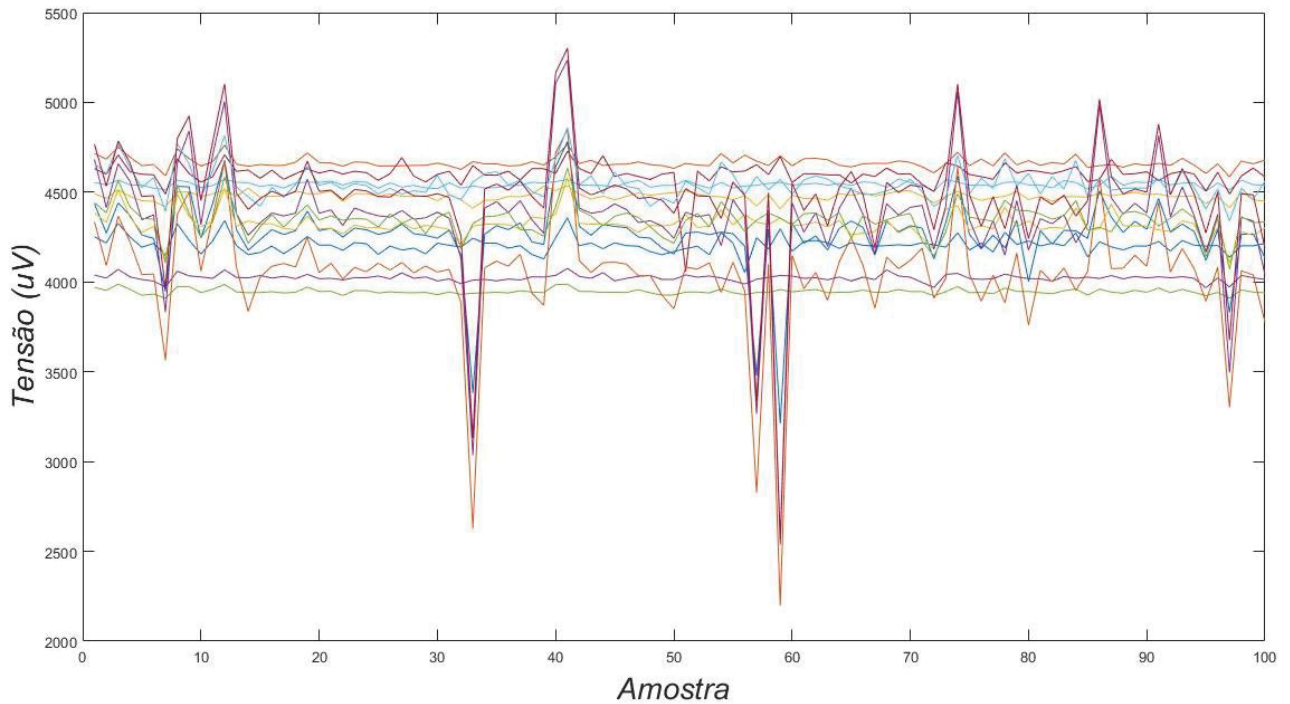


Figura 29 – Gráfico com valores de saída do EEG

FONTE: O autor (2017)

Para a aquisição dos dados da Figura 29, o padrão de pensamento utilizado foi o de “pensar em algo em movimento” e a partir dos dados desta figura, pode-se compreender o comportamento dos dados adquiridos do EEG e, também, de cada sensor individualmente. Na Tabela 3 foram extraídas algumas características estatísticas dos dados obtidos de cada sensor do EEG para o pensamento neutro.

Sensores	Média	Mediana	Variância
AF3	4198,444	4200	2827,748
F7	4664,363	4657,44	898,5314
F3	4480,132	4479,49	526,5128
FC5	4021,773	4022,05	335,3726
T7	3945,725	3943,08	186,4328
P7	4545,111	4543,59	225,0875
O1	4597,195	4604,1	5332,895
O2	4260,485	4274,87	26285,11
P8	4048,393	4066,41	70862,92
T8	4322,352	4314,87	4039,795
FC6	4368,024	4371,28	70093,73
F4	4342,857	4354,36	6814,758
F8	4539,412	4524,62	6682,383
AF4	4481,628	4490,77	67906,91

Tabela 3 - Medidas estatísticas obtidas do sinal do EEG

FONTE: O autor (2017)

Apesar desta coleta ter sido realizada somente para uma melhor compreensão dos dados do EEG e de os dados coletados não terem sido usados para o controle do pêndulo, a partir da Tabela 3 é possível observar a média, mediana e variância de cada sensor a partir de todos os dados coletados para o pensamento “pensar em algo em movimento”. Com isso, pode-se determinar os sensores que mais apresentam variações (P8, FC6 e AF4). A média dos valores dos sensores e mediana, em μV , foi útil para determinar quais os valores que mais irão ocorrer para o pensamento neutro, com isso é possível distinguir os valores para cada coleta.

Estes cálculos estatísticos foram úteis para melhor compreender os dados que foram classificados pelos algoritmos de aprendizado de máquina. Além disso, estas medidas estatísticas foram aplicadas para todos os padrões de pensamento que foram utilizados para controle do pêndulo, conforme pode-se ver na sessão 5.3.

Em seguida, com os dados, já com os filtros de suavização aplicados, separados em 70% para treinamento e 30% para validação, os algoritmos de aprendizado de máquina foram executados na ferramenta MATLAB e gerou-se as matrizes de confusão, juntamente com os resultados de acurácia da mesma, a partir dos dados de validação.

5.1 Classificação dos dados

A seguir, Tabela 4, são representados os resultados da classificação multi-classes dos algoritmos de Aprendizado Extremo, K-NN, Árvore de decisão, Máquina de vetor de suporte, máquina de vetor de relevância e *XgBoost* em forma da acurácia da matriz de confusão. Para gerar estes valores de acurácia da matriz de confusão foram utilizados os dados de validação, das coletas realizadas das cinco pessoas, e foram aplicadas três abordagens diferentes para cada vez que o algoritmo foi executado.

Algoritmos	Abordagem 1	Abordagem 2	Abordagem 3
Máquina de aprendizado extremo	0,8770	0,8764	0,8787
K-NN	0,8989 com k=2	0,9132 com k=5	0,9034 com k=15
Árvore de decisão	0,9973	0,9985	0,9982
Máquina de vetor de suporte	0,9314	0,9236	0,9307
Máquina de vetor de relevância	0,8738	0,8790	0,8793
XgBoost	0,9871	0,9912	0,9986

Tabela 4 - Resultados da acurácia da matriz de confusão dos algoritmos
FONTE: O autor (2017)

A Tabela 4 tem o intuito de mostrar os melhores valores de acurácia obtidos da matriz de confusão, onde, quanto menor o valor, pior será a classificação do algoritmo em tempo real. Para os casos onde os resultados da acurácia da matriz de confusão ficaram próximos do valor um, obtém-se uma excelente classificação dos dados de entrada onde é possível obter um acerto próximo a 100% dos dados que são classificados. Sendo assim, é possível obter um controle do pêndulo com erro próximo a zero e, desta forma, o pêndulo será acionado nos ângulos corretos de acordo com o pensamento do usuário.

Como avaliação da Tabela 4, pode-se concluir que, baseados no percentual de acerto da matriz de confusão, os melhores resultados obtidos foram os resultados dos algoritmos baseados em árvore de decisão e *XgBoost*.

O melhor resultado foi a abordagem 3 do algoritmo XgBoost (0,9986), onde, é possível dizer que o pensamento do qual será interpretado pelo EEG terá um percentual de acerto de 99,86%. O pior caso foi a abordagem 1 do algoritmo

máquina de vetor de relevância (0,8738), do qual pode-se obter um erro de até 12,62% dos dados que são classificados pelo algoritmo. Este erro pode gerar uma imprecisão para o controle do pêndulo e classificar os pensamentos obtidos do EEG de maneira que quando o usuário pensar um determinado padrão de pensamento o algoritmo pode interpretar outro padrão.

5.1.1 Matriz de confusão das melhores acurácias obtidas

Além de verificar a acurácia geral de cada algoritmo, através da matriz de confusão é possível verificar a acurácia de cada um dos padrões de pensamentos utilizados. Por exemplo, para o padrão desligado (representado pelo número 1), pode-se verificar a quantidade de valores classificados corretamente e a quantidade de valores que foram interpretados erroneamente para os outros padrões utilizados, sendo eles ligado em 45° (representado pelo número 2) e ligado em 48° (representado pelo número 3). A seguir serão mostradas as matrizes de confusão dos melhores resultados de acurácia obtidos de cada algoritmo.

Na Figura 30, matriz de confusão do algoritmo ELM, pode-se observar que a classe 3 (pensamento ligado em 48°), obteve o maior erro de classificação (13,39%) em relação as classes 1 e 2.

Classe de saída	1	3367 30,12%	198 1,77%	200 1,79%	89,43% 10,57%
	2	243 2,17%	3259 29,16%	221 1,98%	87,54% 12,46%
	3	262 2,34%	232 2,08%	3196 28,59%	86,61% 13,39%
		86,96% 13,04%	88,34% 11,66%	88,36% 11,64%	87,87% 12,13%
		1	2	3	
		Classe estimada			

Figura 30 - Matriz de confusão para o algoritmo ELM

FONTE: O autor (2017)

A partir da matriz de confusão do algoritmo K-NN (Figura 31), pode-se notar que o maior erro está na classe 3, conforme observado anteriormente na Figura 30. Além disso, é possível afirmar que o maior percentual de acerto (92,91%) está na classe 1 (pêndulo desligado).

Classe de saída	1	3498 31,29%	128 1,15%	139 1,24%	92,91% 7,09%
	2	176 1,57%	3387 30,30%	160 1,43%	90,98% 9,02%
	3	185 1,66%	182 1,63%	3323 29,73%	90,05% 9,95%
		90,65% 9,35%	91,61% 8,39%	91,74% 8,26%	91,32% 8,68%
		1	2	3	
		Classe estimada			

Figura 31 - Matriz de confusão para o algoritmo K-NN

FONTE: O autor (2017)

Para o algoritmo de árvore de decisão, um dos melhores algoritmos em termos de acurácia da matriz de confusão, observa-se que o percentual de acerto foi alto para todas as classes. Entretanto, o mesmo padrão citado anteriormente foi encontrado.

Classe de saída	1	3762 33,66%	1 0,01%	2 0,02%	99,92% 0,08%
	2	2 0,02%	3717 33,25%	4 0,04%	99,84% 0,16%
	3	5 0,04%	3 0,03%	3682 32,94%	99,78% 0,22%
		99,81% 0,19%	99,89% 0,11%	99,84% 0,16%	99,85% 0,15%
		1	2	3	
		Classe estimada			

Figura 32 - Matriz de confusão para o algoritmo árvore de decisão

FONTE: O autor (2017)

A Figura 33 e Figura 34, mostram as matrizes dos algoritmos SVM e RVM das quais pode-se observar que o percentual de acerto para cada classe se mantém iguais aos algoritmos anteriores sendo o maior percentual de acerto para a classe 1 (pêndulo desligado), em seguida a classe 2 (pêndulo em 45°) e por último a classe 3 (pêndulo em 48°).

Classe de saída	1	3544 31,71%	106 0,95%	115 1,03%	94,13% 5,87%
	2	143 1,28%	3452 30,88%	128 1,15%	92,72% 7,28%
	3	142 1,27%	133 1,19%	3415 30,55%	92,55% 7,45%
		92,56% 7,44%	93,52% 6,48%	93,36% 6,64%	93,14% 6,86%
		1	2	3	
		Classe estimada			

Figura 33 - Matriz de confusão para o algoritmo SVM

FONTE: O autor (2017)

Também é possível salientar que os algoritmos SVM e RVM, embora tenham conceitos similares, apresentaram resultados bem diferentes para todas as classes, onde o SVM atingiu os melhores resultados em todos os casos.

Classe de saída	1	3354 30,01%	201 1,80%	210 1,88%	89,08% 10,92%
	2	234 2,09%	3268 29,24%	221 1,98%	87,78% 12,22%
	3	255 2,28%	228 2,04%	3207 28,69%	86,91% 13,09%
		87,28% 12,72%	88,40% 11,60%	88,15% 11,85%	87,93% 12,07%
		1	2	3	
		Classe estimada			

Figura 34 - Matriz de confusão para o algoritmo RVM

FONTE: O autor (2017)

Na Figura 35 obtém-se a matriz de confusão para o algoritmo XgBoost, o qual demonstrou-se ser o mais eficiente. Pode-se observar que os percentuais de acerto para todas as classes foram altos e para a classe 1 não houve nenhum dado classificado incorretamente como sendo da classe 2, o que faz com que o percentual de acerto da classe estimada 2 seja maior do que o das outras classes.

Classe de saída	1	3763 33,66%	0 0,00%	2 0,02%	99,95% 0,05%
	2	1 0,01%	3719 33,27%	3 0,03%	99,89% 0,11%
	3	7 0,06%	3 0,03%	3680 32,92%	99,73% 0,27%
		99,79% 0,21%	99,92% 0,08%	99,86% 0,14%	99,86% 0,14%
		1	2	3	
		Classe estimada			

Figura 35 - Matriz de confusão para o algoritmo XgBoost

FONTE: O autor (2017)

Por fim, faz-se possível atestar que a classe da qual mais se obteve os maiores erros em todos os casos foi a classe 3 (pêndulo em 48°) e a classe que se obteve o maior percentual de acerto foi a classe 1 (pêndulo desligado).

5.1.2 Parâmetros e ajustes utilizados

Os ajustes dos parâmetros de cada algoritmo foram feitos baseados em tentativa e erro e, também, do conhecimento obtido através das revisões da literatura. A seguir, os valores adotados para os parâmetros de cada algoritmo utilizado.

Para abordagem 1 do algoritmo ELM foram utilizados os seguintes parâmetros:

- *Elm_type* = 1 para classificação;

- *Coeficiente de regularização* = **0.3**;
- *Kernel_type* = **lin_kernel**;
- *Kernel_para* = **100**;

Para abordagem 2 do algoritmo ELM foram utilizados os seguintes parâmetros:

- *Elm_type* = **1 para classificação**;
- *Coeficiente de regularização* = **0.5**;
- *Kernel_type* = **lin_kernel**;
- *Kernel_para* = 1000;

Para abordagem 3 do algoritmo ELM foram utilizados os seguintes parâmetros:

- *Elm_type* = **1 para classificação**;
- *Coeficiente de regularização* = **1**;
- *Kernel_type* = **RBF_Kernel**;
- *Kernel_para* = **10**.

Conforme visualizado na Tabela 4, no algoritmo de K vizinho mais próximo foram alterados os valores do *K* com *K*=2 na abordagem 1, *K*=5 na abordagem 2 e *K*=15 na abordagem 3. Para o algoritmo de árvore de decisão, a abordagem 1 foi definida com a atribuição do valor 30 para o parâmetro *MaxNumSplits*, parâmetro que define um número máximo de divisões da árvore. Desta forma, a árvore, mesmo sendo determinística, tem uma variável que define o número máximo de ramos, podendo variar das outras árvores. A abordagem 2 foi definida com a Árvore de Decisão sem alterações na sua estrutura (nativa do MATLAB). A abordagem 3 foi definida com o parâmetro de validação cruzada ligado (*CrossVal* = on) o qual divide a árvore de decisão em 10 árvores que são utilizadas para a validação e, desta forma, tira-se a média de todas estas subárvores geradas para se obter o resultado de acurácia da matriz de confusão.

No algoritmo de SVM, foram aplicadas três diferentes abordagens. Na abordagem 1 foram utilizados os parâmetros com o parâmetro “standardize”, padronização dos valores de entrada, igual a falso, com a “KernelFunction”, função do hiper-plano de separação, sendo RBF e “KernelScale”, divide os valores de

entrada em um valor automático ou escalar para calcular a função Kernel, com valor automático. Na segunda abordagem, foram classificados e testados os valores de entrada sem normalização, mas continuando com a função Kernel sendo RBF e “KernelScale” automático. Na última abordagem (terceira) o parâmetro de otimização dos hyperparâmetros foi habilitado e demorou em torno de 4 horas para o término da execução do algoritmo e os valores encontrados da acurácia da matriz de confusão foram piores que os valores da abordagem 1.

Por fim, nas abordagens do algoritmo de RVM, foram alterados os valores de “alpha”, parâmetro de prioridade para 100, 10, 1 para as abordagens 1, 2 e 3 respectivamente. Entretanto, os valores da acurácia da matriz de confusão não mostraram muita diferença com relação a mudança dos parâmetros.

5.2 Controle do pêndulo

Para o processo de controlar o pêndulo, primeiramente, foi aplicado um sinal com ângulo de 62° , escolhido arbitrariamente, para verificar quais os melhores valores de parâmetros para que o controlador siga o sinal de referência e obtenha o menor erro possível. Ao aplicar a rotina do controlador PID Adaptativo de Dahlin no PAM, foram geradas, através do controlador implementado, do sinal de resposta da planta (linha contínua é a resposta da planta e a linha tracejada é o sinal de referência), para cada valor de τ . Onde τ é uma constante de tempo da ação derivativa do controlador PID e pode ser definido pela seguinte equação:

$$\tau = \frac{T_d}{N} \quad (21.10)$$

Além do critério do erro médio para validação dos melhores valores obtidos do controlador do pêndulo, foi utilizado o erro ITAE (do inglês Integral of Time multiplied by Absolute Error) definido pela seguinte equação:

$$ITAE = \sum_{k=1}^N k \cdot |e(K)| \quad (21.20)$$

Na Tabela 5, é possível visualizar os valores de erro médio e ITAE para todos os valores de τ adotados.

Valor de τ	ITAE	Erro Médio
0,05	97669,3	123,457
0,3	82773	-0,167
0,33	94687	0,246
0,35	262193	5,24
0,38	89573	-0,314
0,4	83530	-0,493
0,5	249881	5,007
0,75	88457,8	0,289
5	7164917	196,8
50	33410028	666,211

Tabela 5 - Valores de erro do controlar PID adaptativo de Dahlin

FONTE: O autor (2017)

Conforme pode-se observar o melhor valor de erro médio e de ITAE foi para a constante de tempo $\tau = 0,3$. A seguir os gráficos do sinal de resposta da planta, de erro do controlador e dos parâmetros do controlador PID para a melhor constante de tempo encontrada de acordo com a Tabela 5.

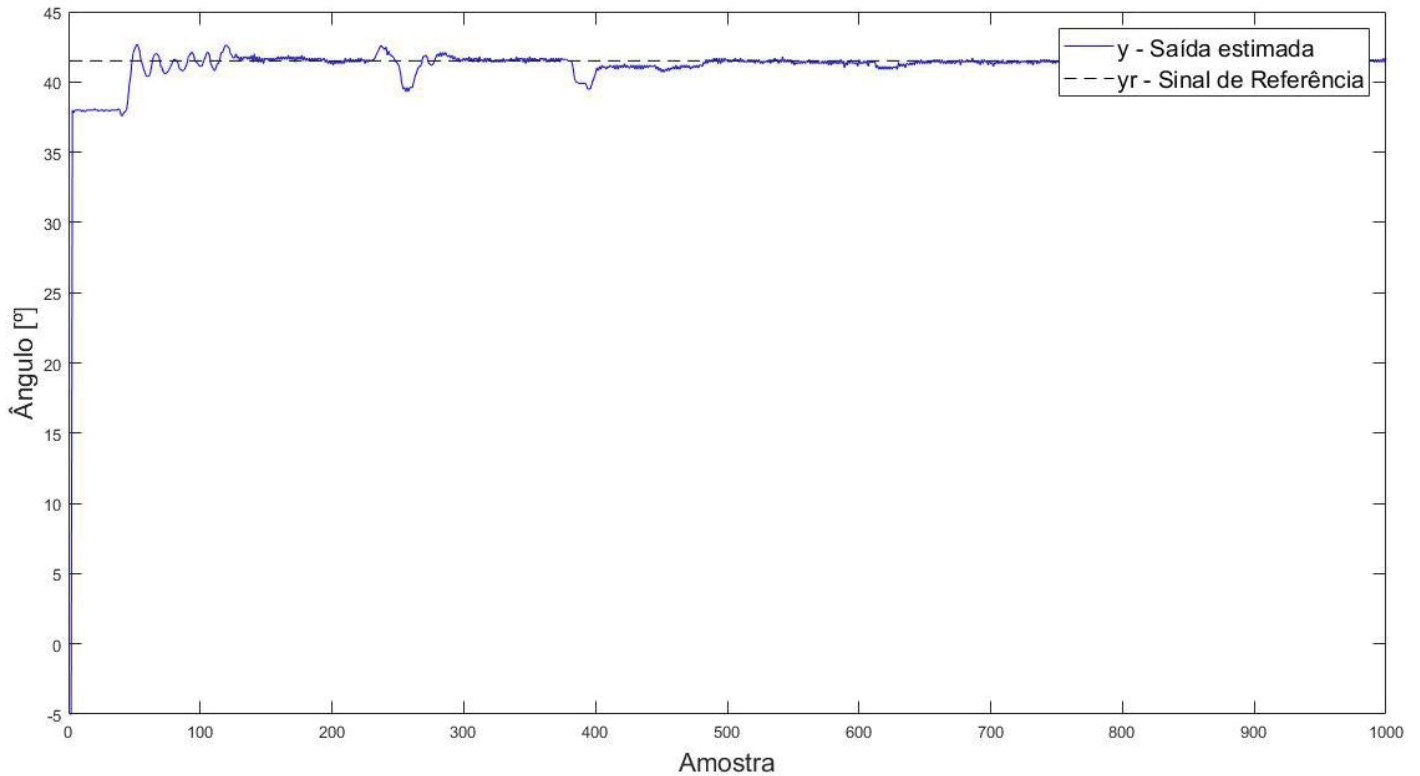


Figura 36 - Sinal de resposta e sinal de referência do controlador para $\tau = 0,3$

FONTE: O autor (2017)

Na Figura 36, é possível observar que o controlador conseguiu atingir um valor muito próximo da referência, com ângulo de 42° . Desta forma, o controlador com o parâmetro de $\tau = 0,3$ pode ser utilizado para a integração final do pêndulo com a classificação dos dados do EEG pelo motivo de que obteve o melhor controle, com o menor valor de erro.

Para entender o comportamento do controlador para $\tau = 0,3$ no ângulo de 42° , o gráfico da Figura 37 mostra o sinal de controle.

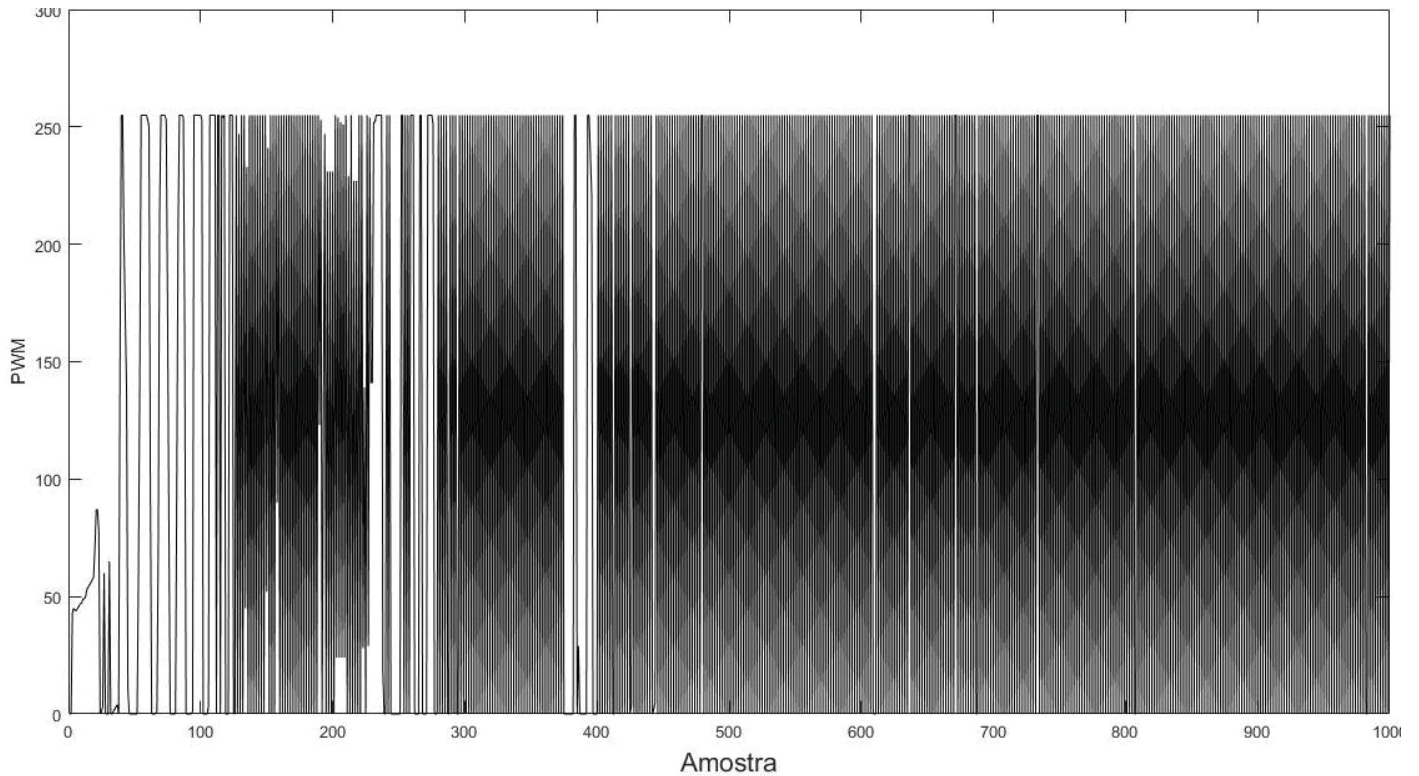


Figura 37 - Sinal de controle para $\tau = 0,3$

FONTE: O autor (2017)

Conforme pode-se observar, para manter-se no ângulo proposto, houve um esforço considerável do controlador (variações entre diferentes velocidades grandes). Entretanto, este esforço foi necessário para manter o pêndulo na posição desejada. Na Figura 38 pode-se visualizar os erros obtidos ao longo do tempo para o controlador.

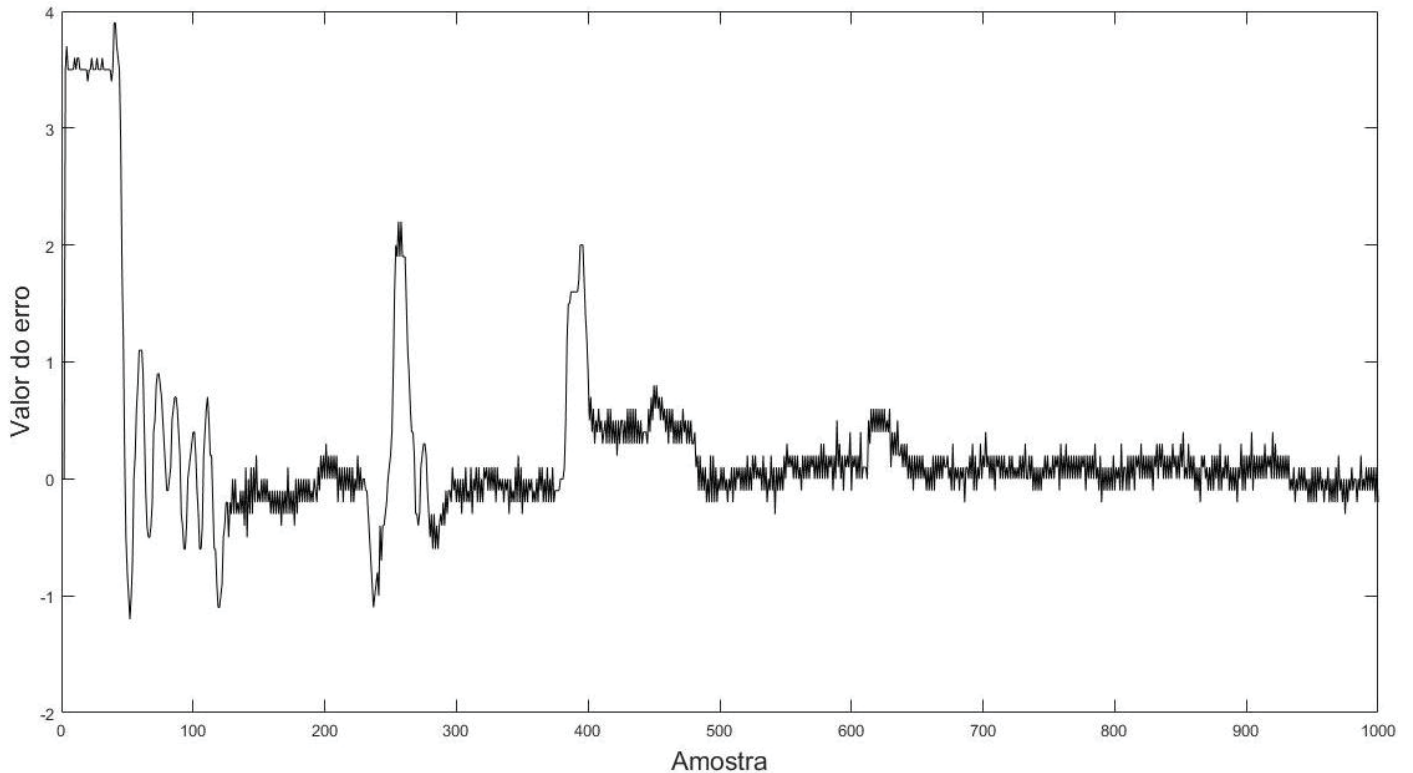


Figura 38 - Sinal de erro do controlador para $\tau = 0,3$

FONTE: O autor (2017)

É possível observar na Figura 38 que houve uma variação maior do erro (entre -1 e 4) até a amostra 500 e, após este instante, esta variação diminuiu. Isto significa que até o instante 500 o controlador teve uma maior dificuldade em se manter na posição correta e, em seguida, conseguiu se manter.

Para visualizar a resposta do ângulo de 42° com outros valores de τ , foram plotados os mesmos gráficos com $\tau = 0,4$ para visualizar se existe muita diferença no processo de controle do pêndulo com valores de ITAE e erro médio próximos. Na Figura 39, é possível observar o sinal de controle e de referência para este instante de tempo.

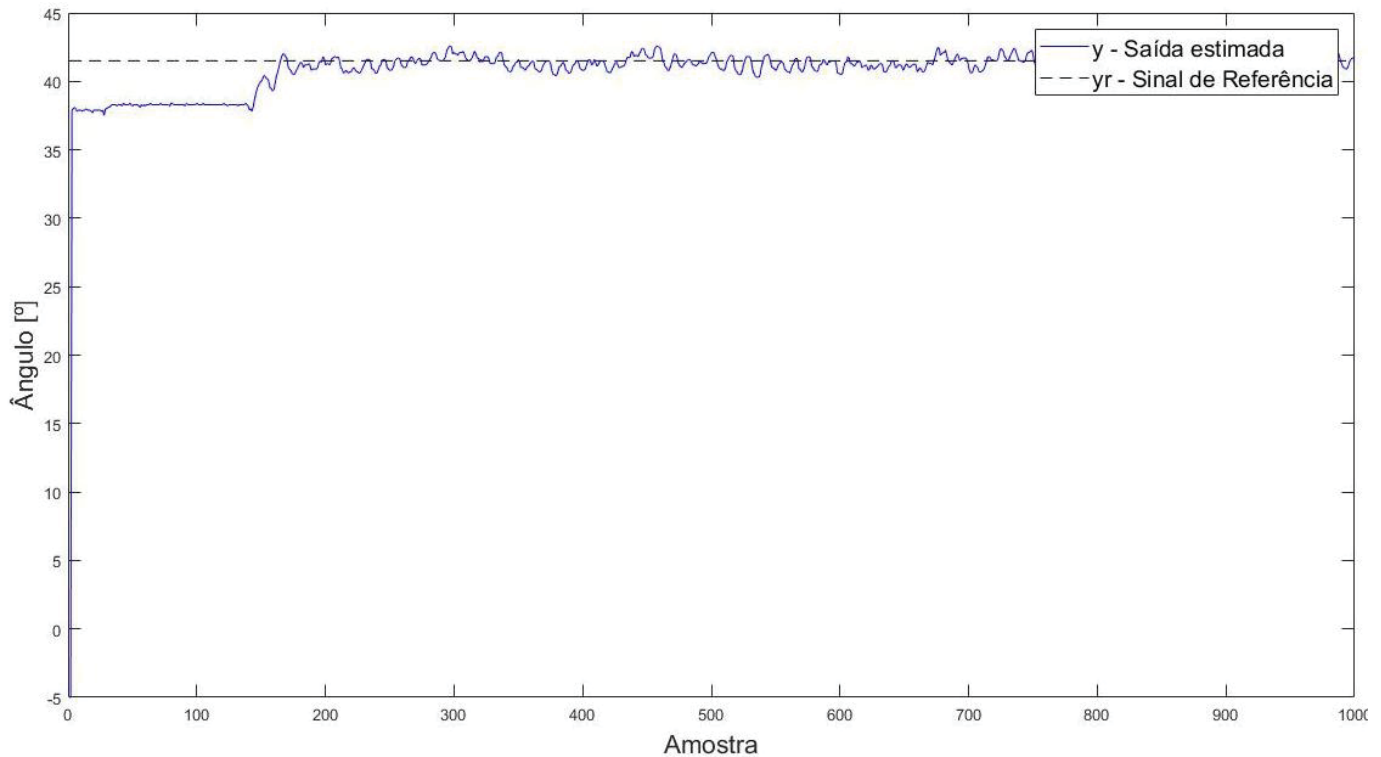


Figura 39 - Sinal de resposta e sinal de referência do controlador para $\tau = 0,4$

FONTE: O autor (2017)

Na Figura 39 observa-se que não há muita diferença entre o controle para os parâmetros de $\tau = 0,3$ e $\tau = 0,4$. Por isso, os valores de erro encontrados, Figura 40, também são muito próximos aos encontrados na Figura 38 com a diferença de que os erros para $\tau = 0,3$ diminuíram mais após um certo instante de tempo.

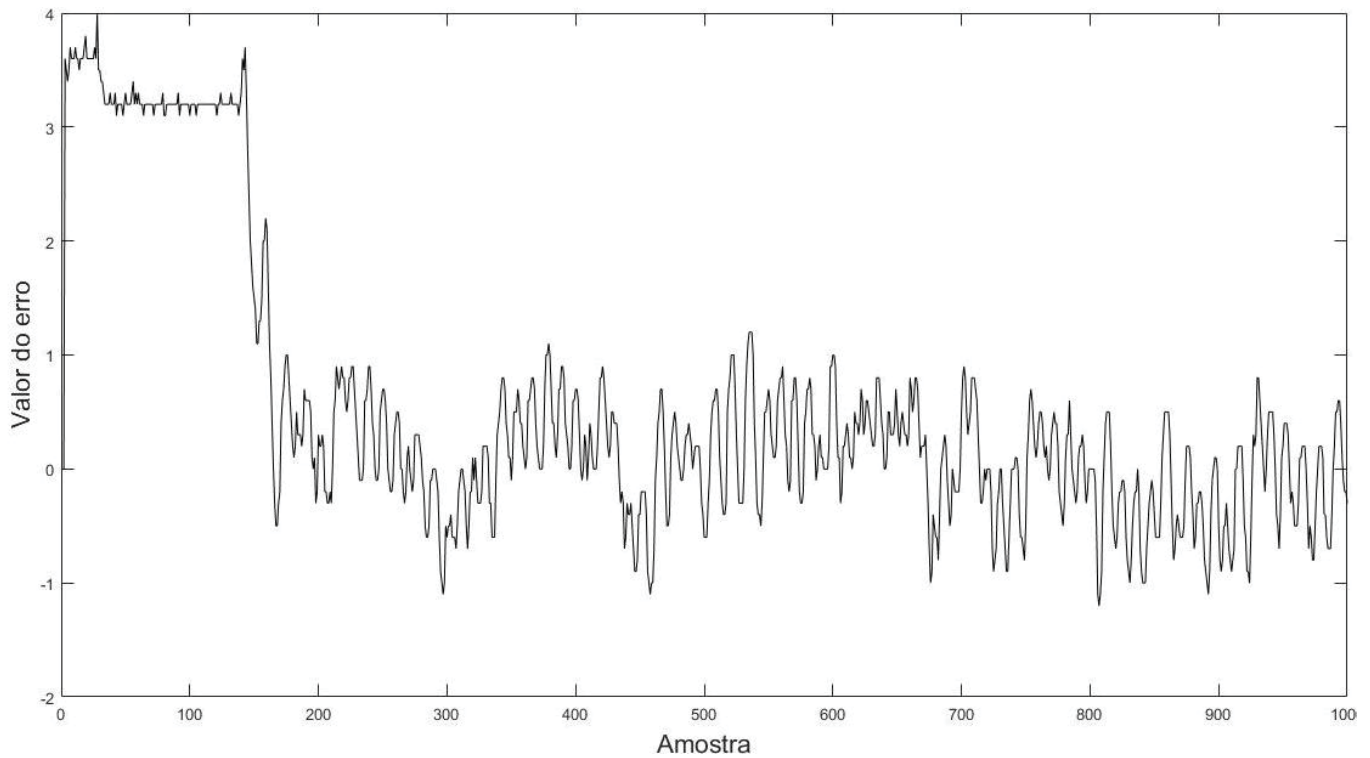


Figura 40 - Sinal de erro do controlador para $\tau = 0,4$

FONTE: O autor (2017)

Além dos valores de erro obtidos, também foram plotados a resposta de controle para $\tau = 0,4$. Na Figura 41, pode-se visualizar a resposta do controlador e observou-se que a resposta da obtida na Figura 37 ($\tau = 0,4$), haja vista que o sinal $u(t)$ se manteve durante um tempo maior no valor do PWM, ocasionando variações menores.

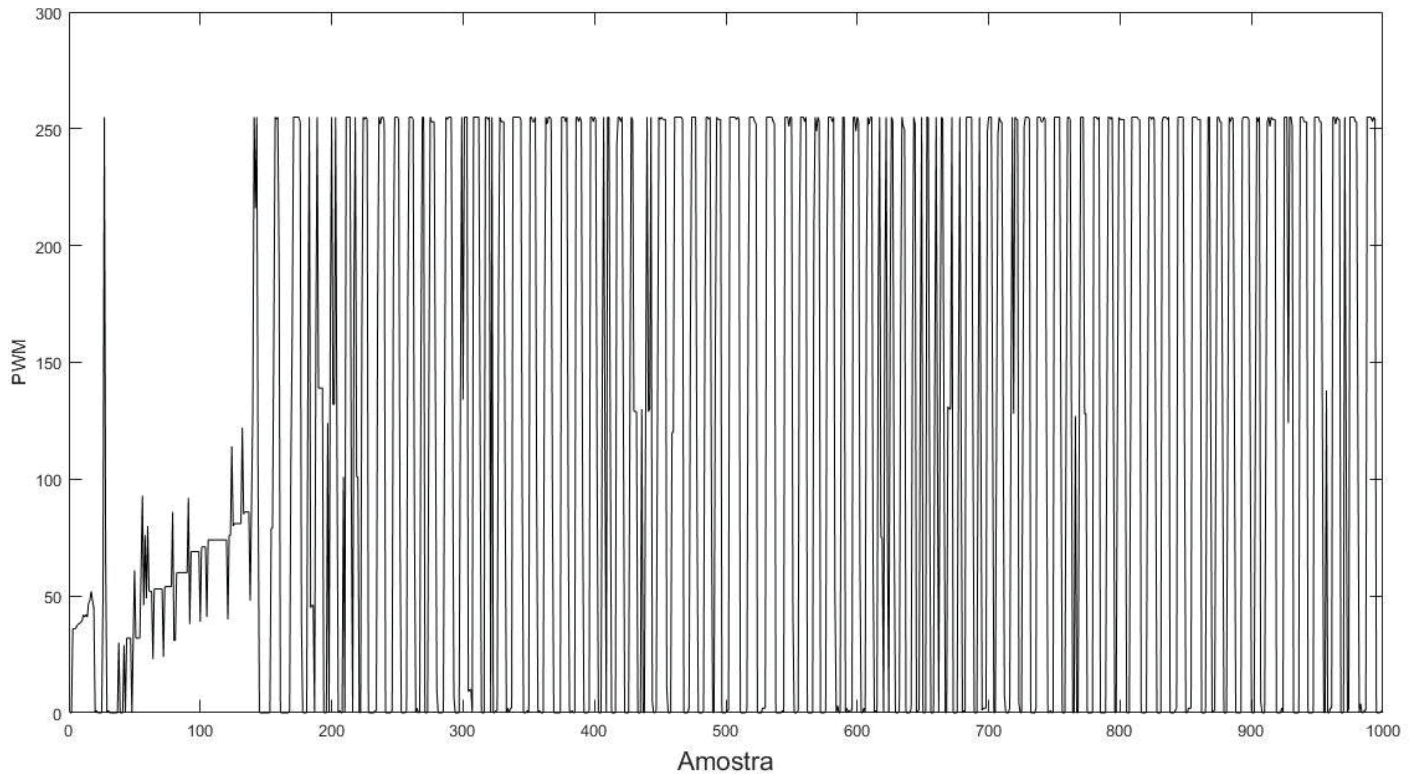


Figura 41 - Sinal de controle para $\tau = 0,4$

FONTE: O autor (2017)

Por fim, foram gerados os mesmos gráficos para mostrar o sinal de saída e os ganhos do controlador para o pior caso obtido, $\tau = 50$, onde foram obtidos os maiores valores de ITAE (33410028) e de erro médio (666,211).

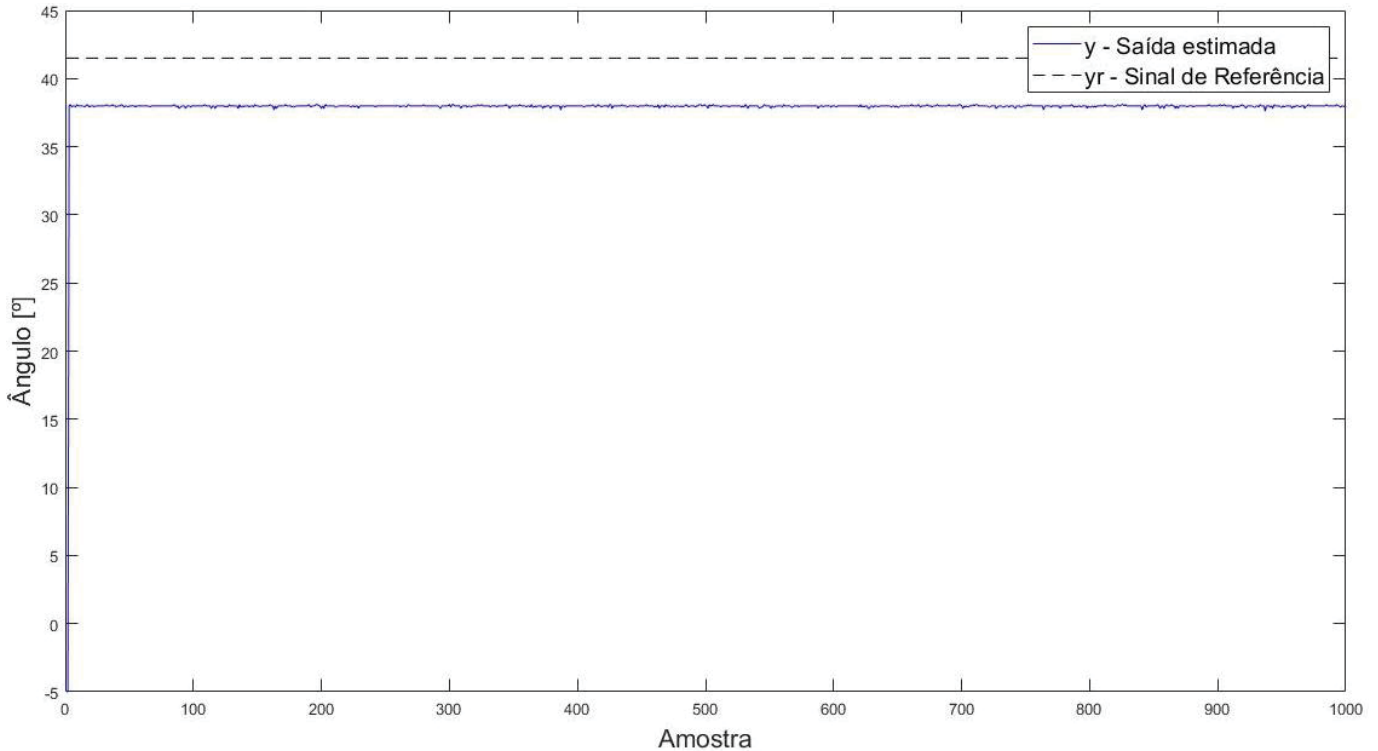
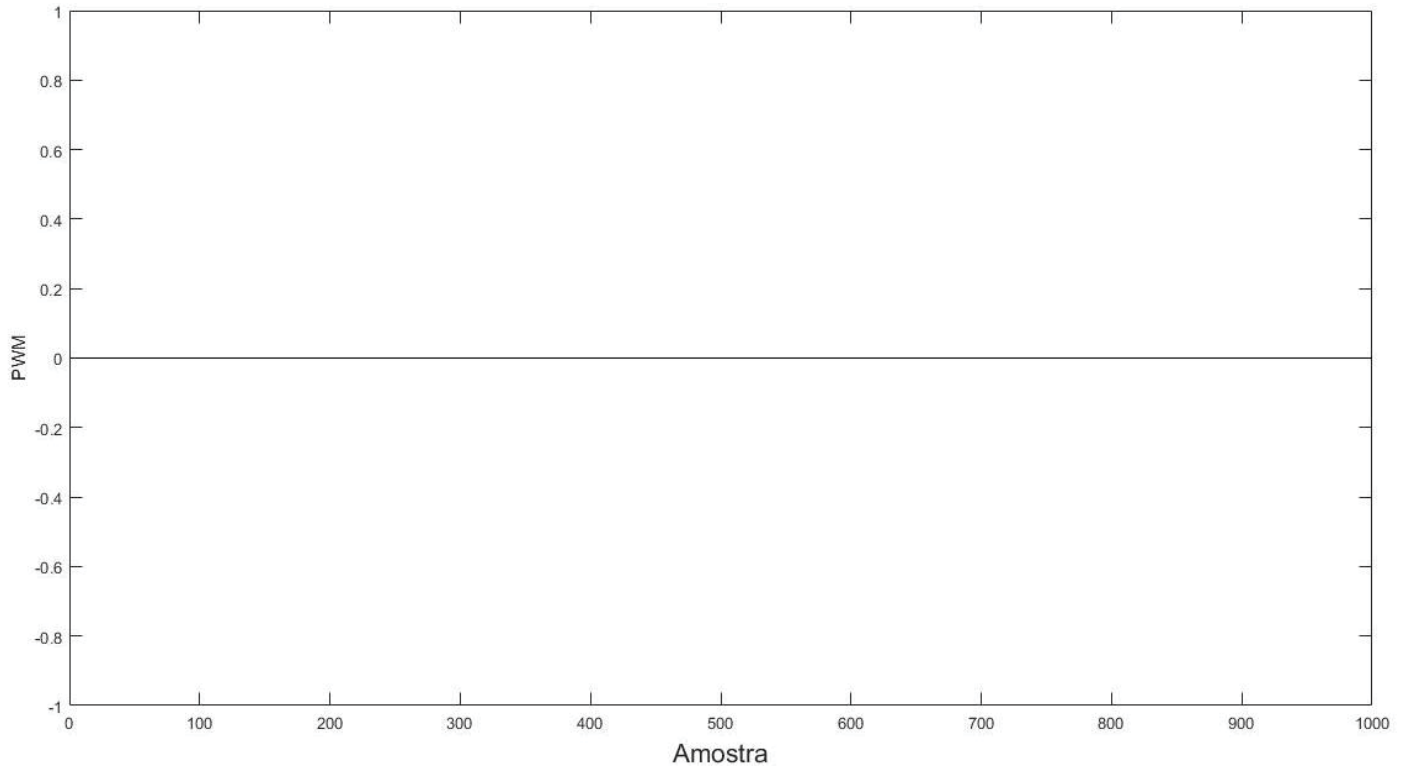


Figura 42 - Sinal de resposta e de referência para $\tau = 50$

FONTE: O autor (2017)

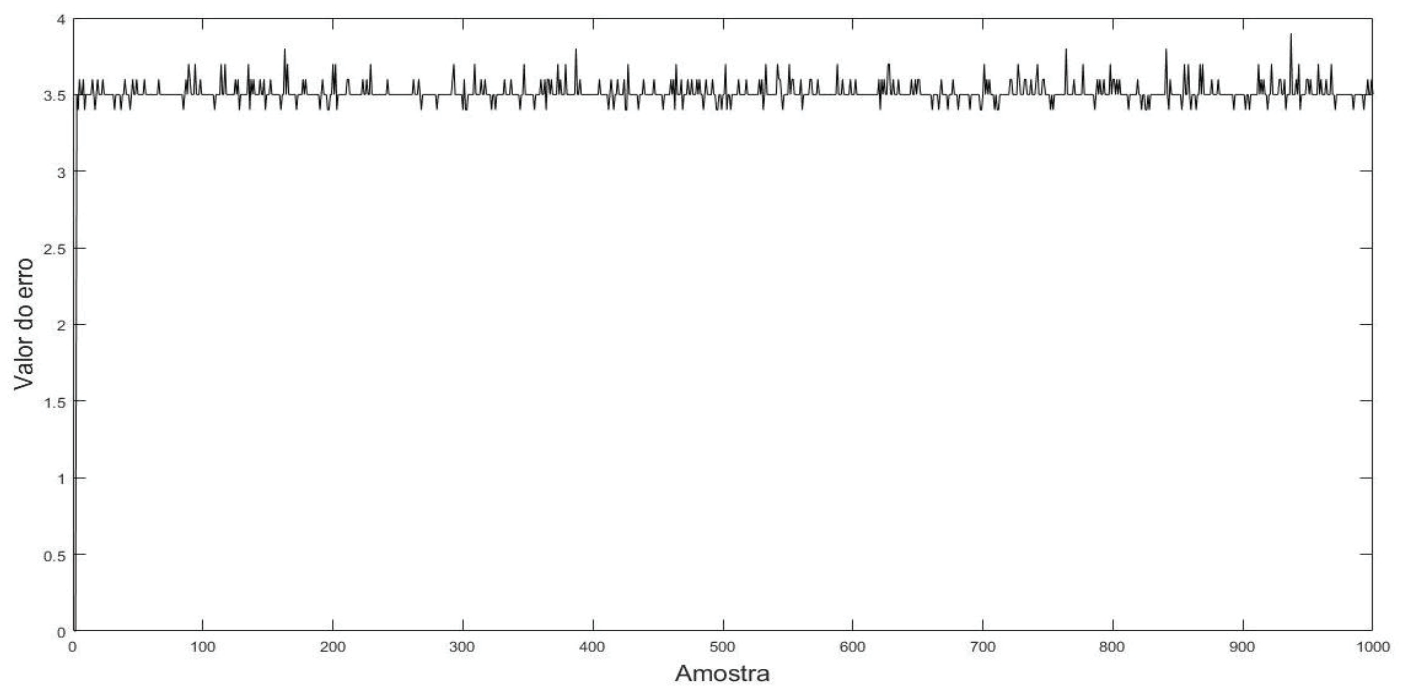
Na Figura 42, é possível observar que o sinal de saída não atingiu o valor do sinal de referência. Por isso, não foi possível obter um bom controle do pêndulo para $\tau = 50$. Na Figura 43 e Figura 44 é possível visualizar a resposta do controlador, que se manteve em 0, e os erros obtidos para o controlador que, neste caso, foram valores altos, ficando praticamente estáveis nos valores altos encontrados.

A resposta do controlador se manteve em 0 pelo motivo de que para $\tau = 50$ a resposta se tornou lenta, fazendo com que a posição inicial (aproximadamente 36°) se mantesse sem alteração da resposta do controlador.

Figura 43 - Sinal de controle para $\tau = 50$

FONTE: O autor (2017)

O sinal de erro variou pouco, entretanto o valor se manteve errado em todo instante de tempo, não atingindo em nenhuma amostra o valor 0 ou próximo a ele.

Figura 44 - Sinal de erro do controlador para $\tau = 50$

FONTE: O autor (2017)

Encontram-se valores próximos de ITAE e erro médio para outros valores de τ , conforme demonstrado na Figura 36 e Figura 39 e, também, conforme pode-se visualizar na Tabela 5. Entretanto, ainda assim, embora muito próximos, o controlador tem um melhor comportamento para $\tau = 0,3$ e, sendo assim, este será o valor utilizado para o controle do pêndulo em tempo real com os dados adquiridos do EEG e classificados com os algoritmos de aprendizado de máquina.

5.3 Integração entre todos os módulos do projeto

Com os parâmetros do controlador corretamente configurados, foram obtidos os resultados finais do trabalho proposto. Inicialmente, foram adquiridos e classificados os dados para o controle do pêndulo desligado. Na Figura 45, Figura 46 e Figura 49, é possível visualizar os valores de tensão (μV) para todas as amostras com frequência de amostragem de 128Hz para o padrão de pensamento do pêndulo desligado, pêndulo em 45° e pêndulo em 48° , respectivamente. Os sinais mostrados já estão com o filtro para suavização dos sinais do EEG.

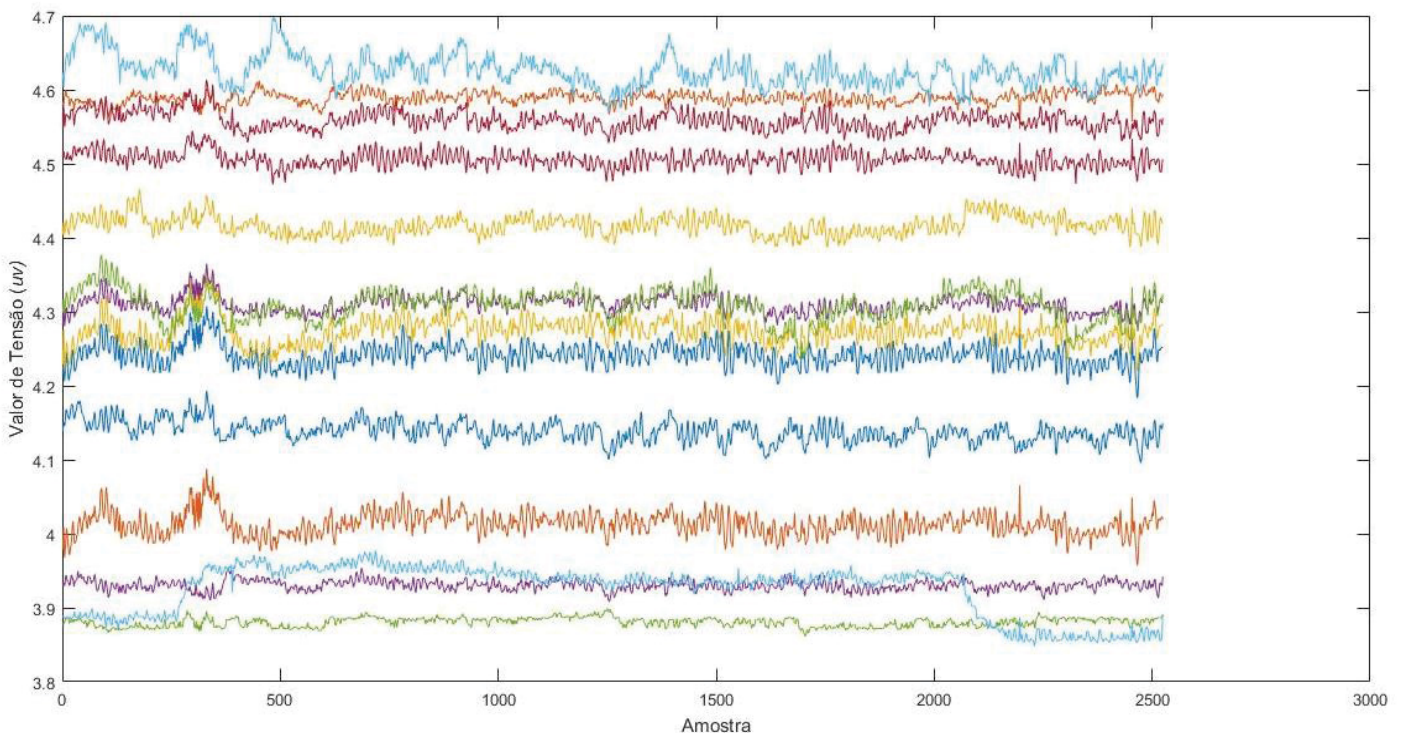


Figura 45 - Sinal do EEG com padrão de pensamento para desligar pêndulo

FONTE: O autor (2017)

Na Figura 45, pode-se ver que houve pequenas variações entre os sensores. Para compreender analiticamente os valores obtidos a partir do padrão de pensamento para desligar o pêndulo, foram calculadas as medidas de média, mediana e variância de cada sensor. Na Tabela 6, pode-se visualizar os valores obtidos.

Sensores	Média	Mediana	Variância
AF3	4,13873327	4,138756708	0,000195646
F7	4,589009571	4,589138179	5,90E-05
F3	4,417060535	4,417227205	0,000157891
FC5	3,930410106	3,930693867	4,53E-05
T7	3,880560561	3,881395867	3,59E-05
P7	3,924257472	3,936761313	0,001170249
O1	4,505939716	4,50558439	0,000106779
O2	4,241355654	4,240765646	0,000222674
P8	4,013223544	4,01311001	0,000239397
T8	4,274286193	4,274263005	0,000322248
FC6	4,309948815	4,309909251	0,000122358
F4	4,308395094	4,310288467	0,00044322
F8	4,626552895	4,624278805	0,000476549
AF4	4,558993111	4,558232126	0,000152537

Tabela 6 - Valores de média, mediana e variância para os sinais coletados para o pensamento de desligar o pêndulo

FONTE: O autor (2017)

Conforme pode-se observar, obteve-se pouca variância do sinal ao pensar em algo desligado e estático, onde a maior variância obtida foi para o sensor P7 (0,001170249). Desta forma, é possível assumir que ocorreu um certo padrão com os dados para este tipo de pensamento o que facilita a classificação dos algoritmos de aprendizado de máquina.

Para o controle do pêndulo em 45°, foram adquiridos os pensamentos com padrão de ligar. Na Figura 46 é possível visualizar como os dados foram plotados e percebe-se que os valores ficaram dispostos de forma diferente do que os valores da Figura 45. Desta forma, obtém-se a classificação dos dados com diferentes movimentos para cada coleta.

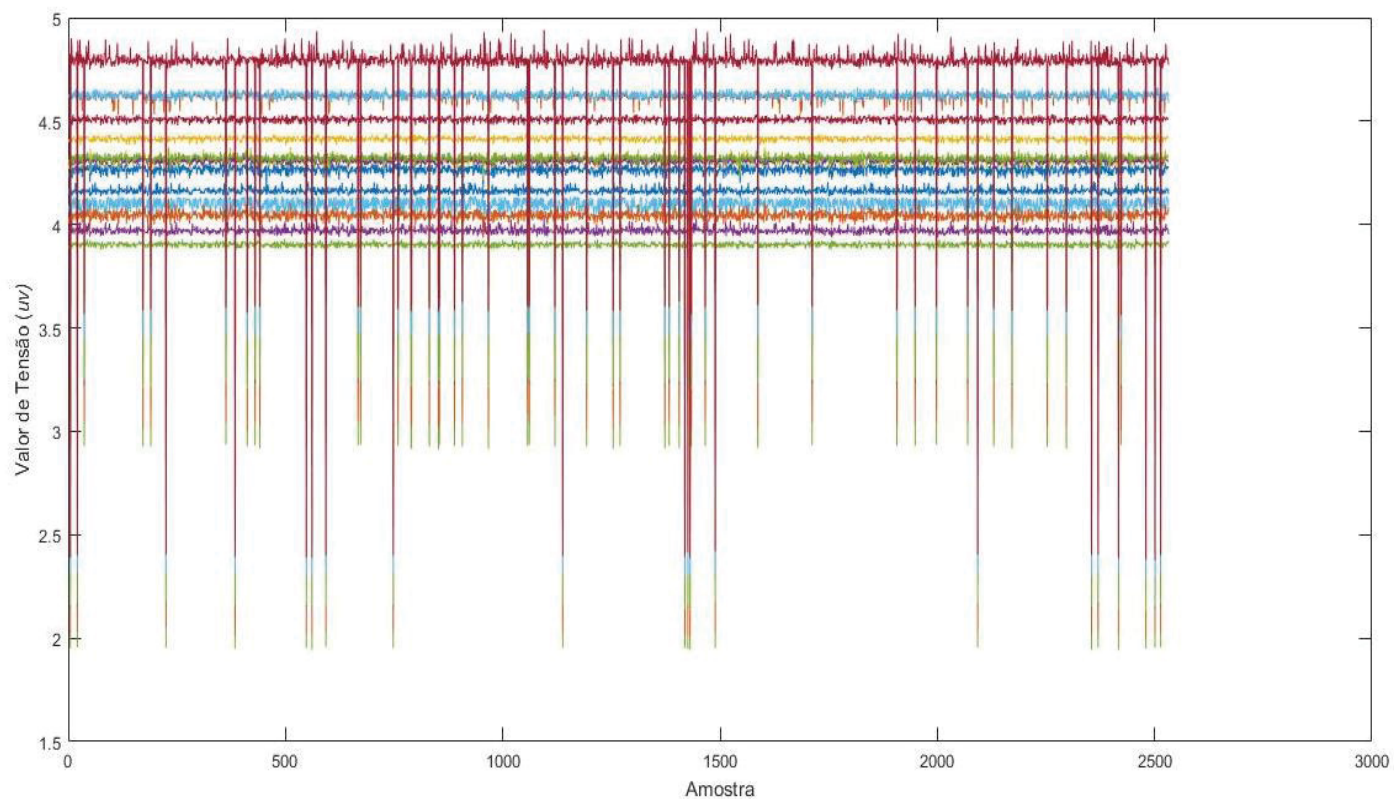


Figura 46 - Sinal do EEG com padrão de pensamento para mover pêndulo em 45°

FONTE: O autor (2017)

De forma a obter a mesma avaliação estatística citada anteriormente, foram calculadas as medidas de média, mediana e variância para o padrão de pensamento de ligar o pêndulo em 45° (Tabela 7).

Sensores	Média	Mediana	Variância
AF3	4,129806	4,162205	0,050537
F7	4,582763	4,62232	0,062459
F3	4,379102	4,413372	0,056676
FC5	3,93918	3,970069	0,046026
T7	3,872541	3,903896	0,04435
P7	4,065582	4,09344	0,049379
O1	4,472586	4,509882	0,059142
O2	4,231408	4,265794	0,053147
P8	4,012442	4,044585	0,047876
T8	4,277951	4,312564	0,05442
FC6	4,275055	4,307634	0,054086
F4	4,288118	4,322929	0,054482
F8	4,589674	4,626554	0,062449
AF4	4,759499	4,794926	0,067694

Tabela 7 - Valores de média, mediana e variância para os sinais coletados para o pensamento de ligar o pêndulo em 45°

FONTE: O autor (2017)

A partir da Tabela 7, pode-se afirmar que a variação para este padrão de pensamento foi maior do que a variação obtida para o padrão de pensamento do pêndulo desligado (Tabela 6). Esta variação também pode ser vista através dos gráficos da Figura 45 e Figura 46 e, essa diferença entre os dois gráficos, representa que os dois padrões conseguem ser distinguidos, o que facilita o trabalho do classificador.

Outro fator importante a ser citado são as variações bruscas que ocorrem em alguns sensores desta coleta, das quais também ficam nítidas analisando a Tabela 7. Estas variações ocorrem pela dificuldade em manter-se um pensamento constante e por ruídos externos que ocorrem no momento da coleta. Embora estes valores prejudiquem a coleta, os mesmos não são considerados no treinamento e classificação em tempo real dos dados, haja vista que estes valores serão desconsiderados, conforme explicado e observado na sessão de validação dos dados (4.2.2).

Após realizada a coleta e classificação dos dados, o pêndulo foi controlado com os dados da Figura 46 onde obteve-se o controle do pêndulo conforme pode ser visualizado na Figura 47.

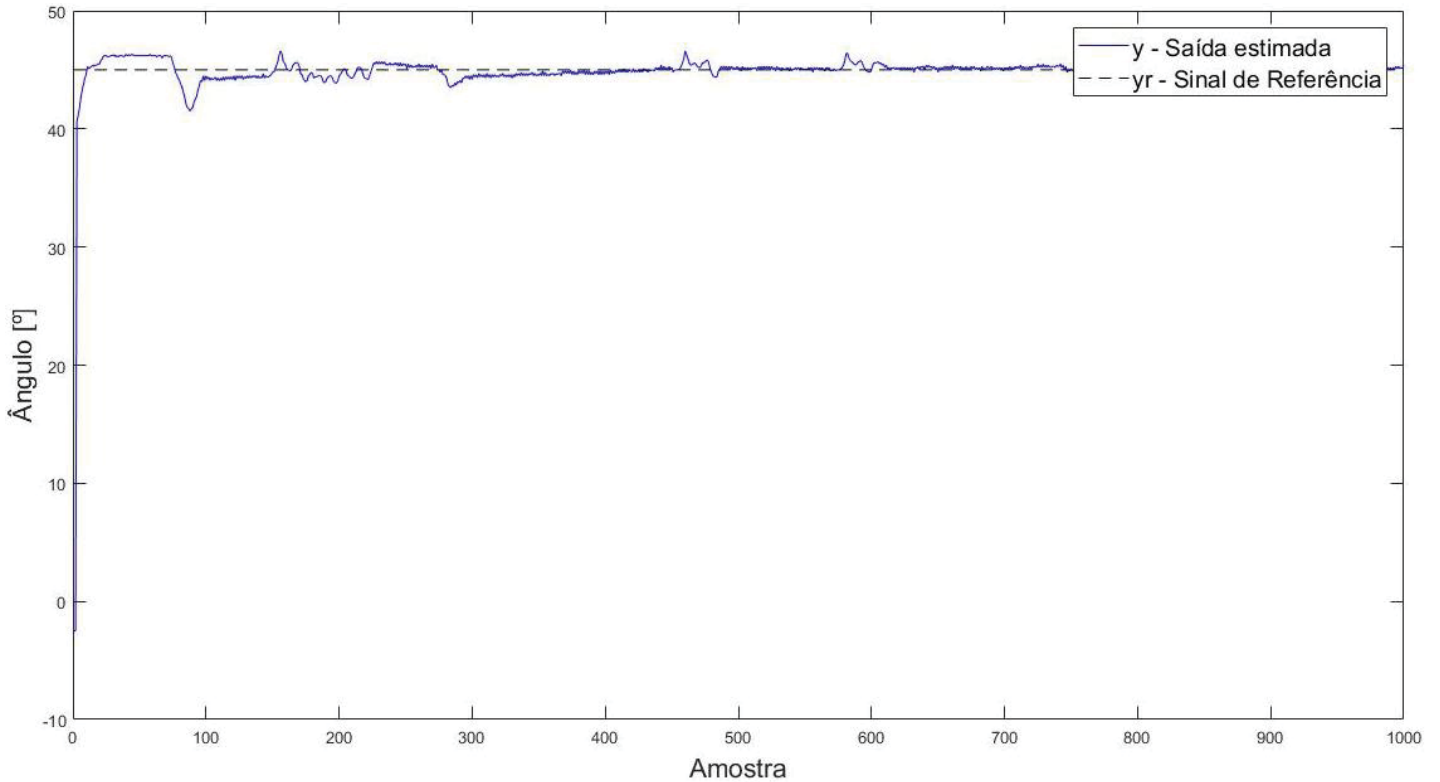


Figura 47 - Sinal de resposta e de referência para controle do pêndulo em 45°

FONTE: O autor (2017)

Em seguida, conforme pode ser visualizado na Figura 48, foi gerado o gráfico do sinal $u(t)$ que foi aplicado para o controle do pêndulo em 45° onde pode-se visualizar que o esforço do controlador para manter o pêndulo no ângulo correto foi intenso, haja vista que o sinal variou de 0 a 250 em vários instantes da amostra.

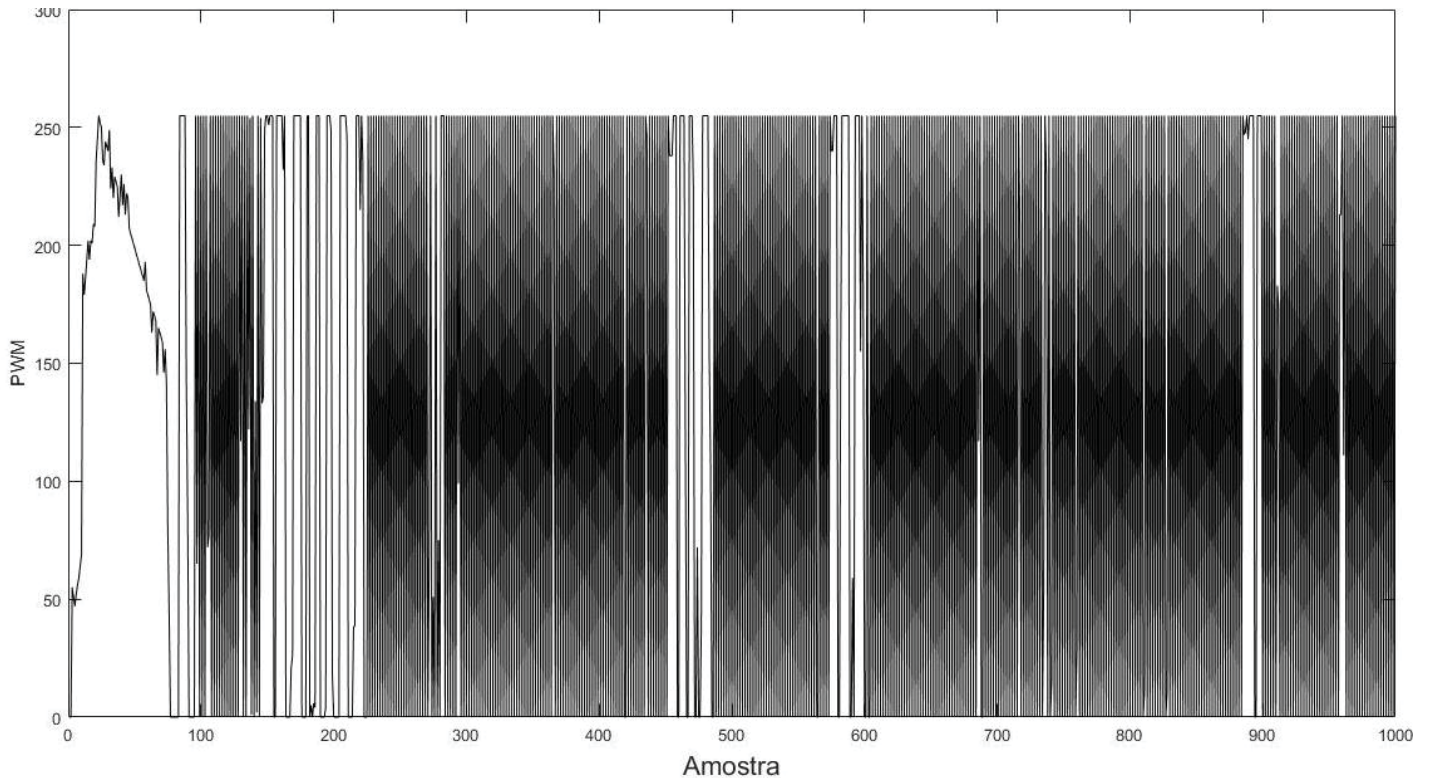


Figura 48 - Sinal $u(t)$ de controle aplicado para movimento do motor com ângulo de 45°

FONTE: O autor (2017)

Como última etapa para controle do pêndulo foram adquiridos os dados para o controle do pêndulo em 48° com padrão de pensamento ligado duas vezes mais forte, onde o usuário deveria pensar em algo em movimento duas vezes mais rápido do que para o padrão de pensamento utilizado para o ângulo de 45° , conforme Figura 49. Pode-se observar visualmente que os padrões adquiridos foram diferentes das coletas da Figura 45 e Figura 46. Por este motivo, é possível obter uma excelente classificação dos dados, conforme pode ser visto na sessão anterior na Tabela 4.

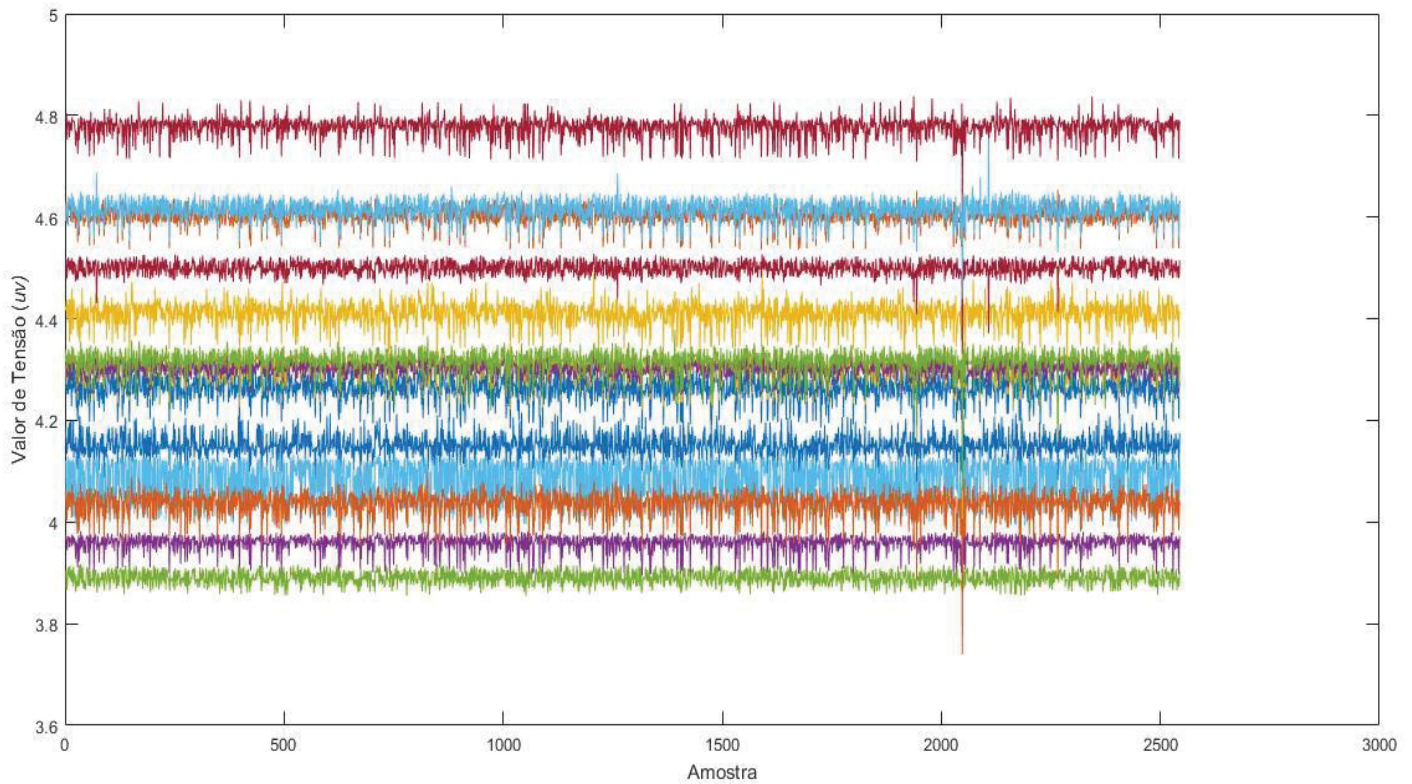


Figura 49 - Sinal do EEG com padrão de pensamento para mover pêndulo em 48°

FONTE: O autor (2017)

Para esta padrão de pensamento, também foram calculadas as mesmas medidas estatísticas dos outros padrões citados anteriormente (Tabela 8).

Sensores	Média	Mediana	Variância
AF3	4,15036	4,14849	0,000473
F7	4,603657	4,60279	0,000392
F3	4,410175	4,412803	0,000569
FC5	3,95856	3,96179	0,000258
T7	3,890169	3,888222	0,000173
P7	4,078204	4,095084	0,001854
O1	4,49963	4,498506	0,000186
O2	4,265458	4,264024	0,000746
P8	4,035252	4,039402	0,000621
T8	4,302534	4,304979	0,000559
FC6	4,299691	4,300682	0,000251
F4	4,31772	4,320148	0,000441
F8	4,614645	4,614419	0,000403
AF4	4,77722	4,781148	0,000373

Tabela 8 - Valores de média, mediana e variância para os sinais coletados para o pensamento de ligar o pêndulo em 48°

FONTE: O autor (2017)

Pode-se observar na Tabela 8 que a variância para o padrão de pensamento para manter o pêndulo em 48° foi menor do que a obtida para o pêndulo e 45° . Além disso, os valores obtidos para as medidas de média e mediana também mostram que, para cada padrão de pensamento coletado, os valores destas medidas foram diferentes umas das outras. Com isso pode-se afirmar que os dados coletados para cada padrão de pensamento foram diferentes o que facilitou o trabalho do classificador para obter os resultados obtidos conforme Tabela 4.

Com o padrão adquirido da Figura 49, foi possível controlar o pêndulo no ângulo de 48° . A Figura 50 mostra como foi a resposta do controlador com relação a referência para o controle do pêndulo em 48° .

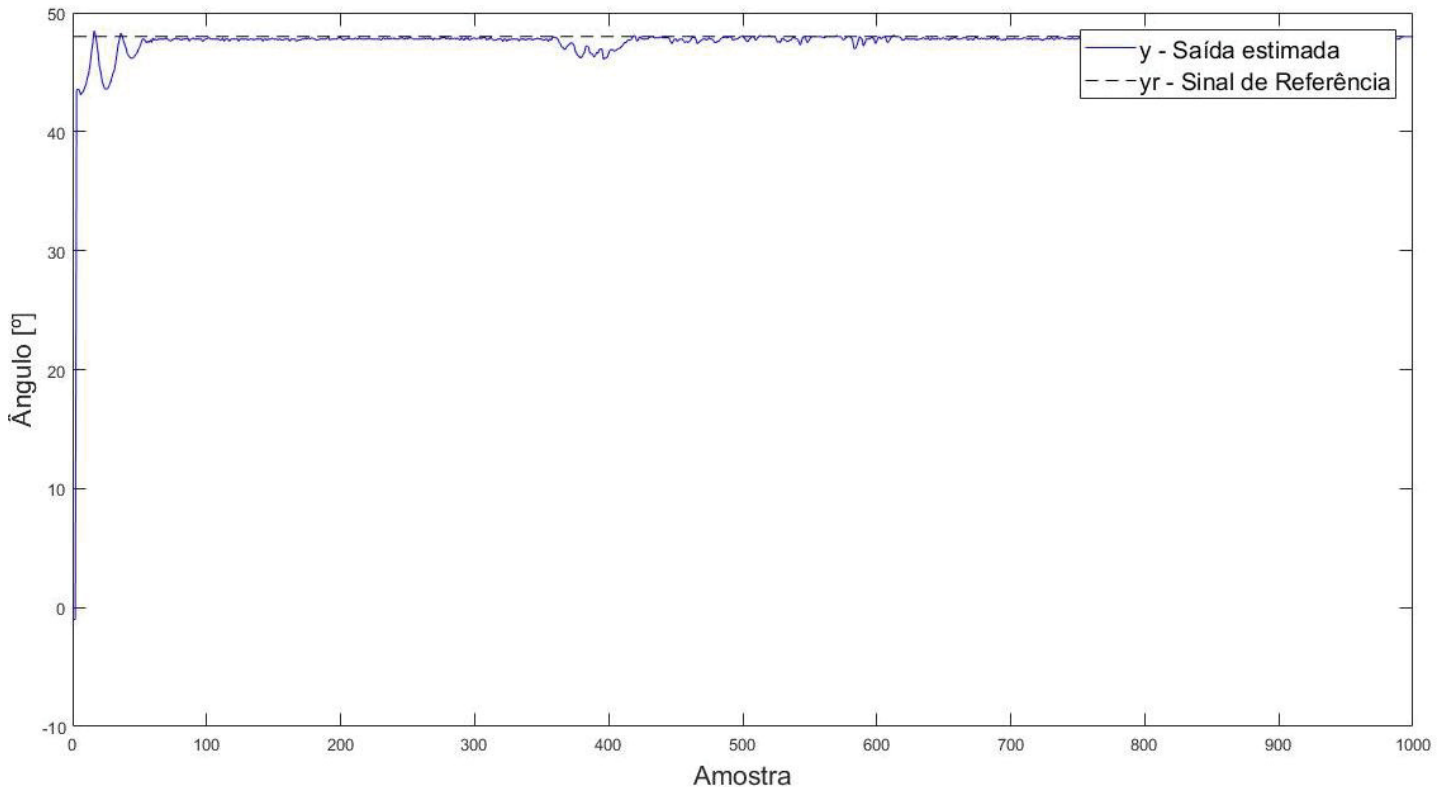


Figura 50 - Sinal de resposta e de referência para controle do pêndulo em 48°

FONTE: O autor (2017)

Na Figura 51, observa-se a resposta do sinal $u(t)$ para o controle do motor com ângulo de 48° . Conforme é possível observar, não foi necessário um esforço muito grande do controlador como foi visto na Figura 48. Logo, o controlador conseguiu se manter no ângulo de 48° sem muita variação do sinal $u(t)$.

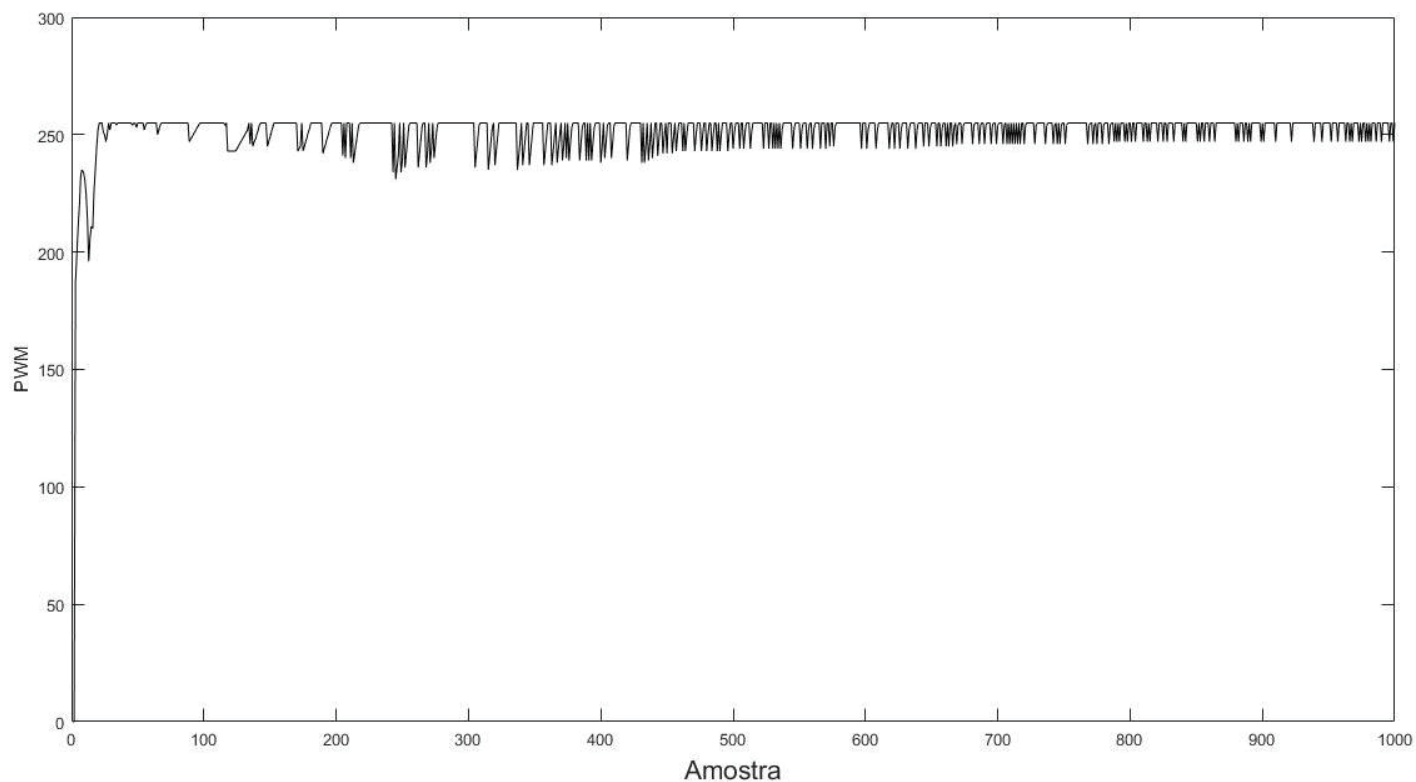


Figura 51 - Sinal $u(t)$ de controle aplicado para movimento do motor com ângulo de 48°

FONTE: O autor (2017)

Por fim, pode-se obter como resultado um bom controle do pêndulo em tempo real para diferentes posições (desligado, 45° e 48°).

6 CONCLUSÕES E TRABALHOS FUTUROS

O projeto como um todo foi de grande importância para aprendizagem de diversos conceitos, desde o funcionamento do sistema nervoso, a aquisição de dados do cérebro através das sinapses, até a utilização dos dados capturados para a execução de uma ação, que no caso deste projeto foi o controle de um *PAM*. Assuntos como algoritmos de aprendizagem de máquina, funcionamento do encéfalo e medula espinhal, utilização de *softwares* como MATLAB, aquisição e dados de maneira não-invasiva, foram todos muito úteis para o desenvolvimento do projeto, e também, um complemento para aprendizagem.

Os resultados obtidos mostraram que cada etapa do trabalho alcançou os objetivos esperados. Durante o estudo de algoritmos de classificação, ficou definido o uso dos algoritmos que poderiam ser as melhores opções quando se trata do processamento de dados do cérebro humano. Porém, os algoritmos que apresentaram os melhores resultados, em termos da acurácia da matriz de confusão, foram os baseados em árvore de decisão e a técnica de ensemble XgBoost, seguidos de SVM.

Para o controle do pêndulo, foi utilizado o controlador PID Dahlin Adaptativo, entretanto, existem diversos algoritmos de controle que podem ser aplicados, mas, como o objetivo principal deste trabalho ficou na aquisição de dados do cérebro e adaptação de algoritmos de aprendizado de máquina para processamento. Não foram implementadas e testadas outras abordagens de controlador. Além disso, para a ligação elétrica do pêndulo, podem ser projetados novos modelos que reduzam o tamanho físico de todo o sistema do pêndulo.

O Algoritmo principal, que integra todos os módulos deste trabalho, aquisição, processamento, classificação e controle do processo, foi de grande valia para que o sistema funcionasse em tempo real. Além disso, torna-se fácil alterar módulos específicos do projeto e integrar no algoritmo principal novamente.

Com base nas principais funcionalidades do projeto, é possível apontar que a ideia é inovadora e pode ajudar consideravelmente a vida de pessoas com necessidades físicas. Além disso, é possível adicionar outros módulos ao projeto

com novas funcionalidades e isso pode gerar um impacto positivo ainda maior, o que torna a sua implementação viável e atraente para o público ao qual está voltado.

É importante destacar a importância da coleta de dados em indivíduos diferentes para o treinamento dos algoritmos de aprendizado de máquina, haja vista que pensamentos são particulares de cada indivíduo e os dados coletados podem apresentar-se com valores diferentes. Além disso, observou-se que os dados se tornam mais fácil a classificação quando os usuários pensam algo que movimentem a parte emotiva do cérebro. Um exemplo seria a coleta realizada em indivíduos que pensam em atos felizes e atos tristes, onde fica fácil a classificação dos dados.

Até esta etapa do projeto, é possível verificar que seu funcionamento atendeu as expectativas. Foi gratificante observar a superação das adversidades encontradas inicialmente e ver que é possível, através de dados do cérebro, utilizar os conhecimentos obtidos de forma a criar aplicações que podem auxiliar pessoas com necessidades físicas.

REFERÊNCIAS

AGARWAL, S. K.; SHAH, S.; KUMAR, R. Classification of mental tasks from EEG data using backtracking search optimization based neural classifier. **Neurocomputing**, v. 166, n. 14, p. 397–403, 2015.

ASCIO, C.; FACLK, E. B. Neuroscience. **International Encyclopedia of Communication Theory and Philosophy.**, v. 37, n. 46, p. 11074–11084, 2016.

BALDWIN, C. L.; PENARANDA, B. N. NeuroImage Adaptive training using an artificial neural network and EEG metrics for within- and cross-task workload classification. **NeuroImage**, v. 59, n. 1, p. 48–56, 2012. Elsevier Inc. Disponível em: <<http://dx.doi.org/10.1016/j.neuroimage.2011.07.047>>.

BISHOP, C. M.; JORDAN, M.; KLEINBERG, J.; SCHÖLKOPF, B.; CHRISTOPHER, M. **Pattern Recognition and Machine Learning**. 1º ed. Inglaterra, 2006.

BOBÁL, V.; BÖHM, J.; FESSL, J.; MACHÁCEK, J. **Digital self-tuning controllers: algorithms, implementation and applications**. 2005, 2005.

BRAGA, A. DE P.; CARVALHO, A. C. P. DE L. F.; LUDERMIR, T. B. **Redes neurais artificiais : teoria e aplicações**. 2ª ed. Rio de Janeiro: LTC Editora, 2007.

BRAVERMAN, E. R. Brain Mapping: A Short Guide to Interpretation, Philosophy and Future. **Journal of Orthomolecular Medicine**, v. 5, n. 4, 1990. Disponível em: <<https://pdfs.semanticscholar.org/0330/a4732a35fba8f5a1fa7b395e72b67357aaa0.pdf>>. Acesso em: 23/9/2017.

BUCHER, R.; BALEMI, S. CAN-bus based rapid control prototyping system for education laboratories. IFAC Proceedings Volumes (IFAC-PapersOnline). **Anais...** , 2008.

CHAE, Y.; JEONG, J.; JO, S. Noninvasive brain-computer interface-based control of humanoid navigation. IEEE International Conference on Intelligent Robots and

Systems. **Anais...** . p.103–115, 2011. São Francisco: 25/09/2011.

CHEN, T.; GUESTRIN, C. XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16. **Anais...** . p.785–794, 2016. San Francisco, CA, USA: 2016.

CHOON GUAN LIM, B.; LEE, T.-S.; GUAN, C.; et al. Effectiveness of a Brain-Computer Interface Based Programme for the Treatment of ADHD: A Pilot Study. **Psychopharmacology Bulletin**, v. 43, p. 73–82, 2010.

COOPER, R.; OSSELTON, J. W.; SHAW, J. C. **Fundamentals of EEG Technology**. 2º ed. Bloomfield: 1974, 1974.

COPPIN, B. **Inteligência Artificial**. Rio de Janeiro, 2010.

COVER, T. M.; HART, P. Nearest neighbor pattern classification. **IEEE Transactions on Information Theory**, v. 13, n. 1, p. 21–27, 1967.

DAHLIN, E. B. **Designing and tuning digital controllers**. Intrum, **Control Syst.** 1968.

DELORME, A.; MAKEIG, S. EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. **Journal of Neuroscience Methods**, v. 134, n. 1, p. 9–21, 2004.

DOUD, A. J.; LUCAS, J. P.; PISANSKY, M. T.; HE, B. Continuous three-dimensional control of a virtual helicopter using a motor imagery based Brain-Computer interface. **PLoS ONE**, 2011.

DUVINAGE, M. **Towards Effective Non-Invasive Brain-Computer Interfaces Dedicated to Ambulatory Applications**, 2015. University of Mons.

GALBRAITH, B. V. Brain-Machine Interfaces for Robotic Control | Guenther Lab. Disponível em: <<http://sites.bu.edu/guentherlab/research-projects/bmi-robots/>>.

Acesso em: 24/9/2017.

HAMMOND, D. C. What Is Neurofeedback? **Journal of Neurotherapy: Investigations in Neuromodulation, Neurofeedback and Applied Neuroscience**, 2007.

HAYKIN, S. S. **Neural networks : a comprehensive foundation**. 2º ed. Prentice Hall, 1999.

HEINRICH, H.; GEVENSLEBEN, H.; STREHL, U. Annotation: Neurofeedback - Train your brain to train behaviour. **Journal of Child Psychology and Psychiatry and Allied Disciplines**, 2007.

HO, H. L.; RAD, A. B.; CHAN, C. C.; WONG, Y. K. Comparative studies of three adaptive controllers. **ISA Transactions**, 1999.

HUANG, G. BIN; WANG, D. H.; LAN, Y. Extreme learning machines: A survey. **International Journal of Machine Learning and Cybernetics**, v. 13042, n. 6, p. 120–131, 2011.

HUANG, G. BIN; ZHU, Q. Y.; SIEW, C. K. Extreme learning machine: Theory and applications. **Neurocomputing**, v. 261, n. 4, p. 1–276, 2006.

INGEL, A. **Control a Robot via VEP Using Emotiv EPOC**, 2015. UNIVERSITY OF TARTU.

KARBOWSKI, K. Hans Berger (1873-1941). **Journal of Neurology**, v. 249, n. 8, p. 1130–1131, 2002.

KARTALOPOULOS, S. V. **ARTIFICIAL NEURAL NETWORKS: CONCEPTS. Understanding neural networks and fuzzy logic: basic concepts and applications**, 1996.

KROPOTOV, J. D. **Quantitative EEG, Event-Related Potentials and**

Neurotherapy. 2009.

KRYGER, M.; WESTER, B.; POHLMAYER, E. A.; et al. Flight simulation using a Brain-Computer Interface: A pilot, pilot study. **Experimental Neurology**, v. 287, p. 473–478, 2017.

LE, Q. M. **EEG-Controlling Robotic Car and Alphabetic Display by Support Vector Machine for Aiding Amyotrophic Lateral Sclerosis Patie**, 2015. Arkansas. Disponível em: <<http://scholarworks.uark.edu/cgi/viewcontent.cgi?article=1044&context=eleguht>>.

LEMES, A.; PRADO, F.; KIAMETIS, D. Software Livre Na Educação Em Engenharia De Controle: Um Estudo De Caso Na Análise E Projeto De Controle De Um Pêndulo Amortecido. XVIII Congresso Brasileiro de Automática. **Anais...** , 2010.

LI, X.; HU, B.; XU, T.; SHEN, J.; RATCLIFFE, M. A study on EEG-based brain electrical source of mild depressed subjects. **Computer Methods and Programs in Biomedicine**, 2015.

LI, Z.; YANG, C.; FAN, L. **Advanced control of wheeled inverted pendulum systems**. 2013.

LIANG, N.-Y.; SARATCHANDRAN, P.; HUANG, G.-B.; SUNDARARAJAN, N. Classification of Mental Tasks From EEG Signals Using Extreme Learning Machine. **International Journal of Neural Systems**, v. 16, n. 1, p. 29–38, 2006.

LOTTE, F.; CONGEDO, M.; LÉCUYER, A.; LAMARCHE, F.; ARNALDI, B. A review of classification algorithms for EEG-based brain-computer interfaces. **J Neural Eng**, v. 2, n. 1, p. 1–13, 2007.

LUAUTÉ, J.; MORLET, D.; MATTOUT, J. BCI in patients with disorders of consciousness: Clinical perspectives. **Annals of Physical and Rehabilitation Medicine**, 2015.

MACKAY, D. J. C. The Evidence Framework Applied to Classification Networks. **Neural Computation**, v. 4, n. 5, p. 720–736, 1992.

MAJUMDAR, K. Human scalp EEG processing: Various soft computing approaches. **Applied Soft Computing Journal**, 2011.

MATHWORKS, I. Workflow for Neural Network Design - MATLAB & Simulink. Disponível em: <<http://www.mathworks.com/help/nnet/ug/workflow-for-neural-network-design.html>>. Acesso em: 23/9/2017.

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, 1943. Kluwer Academic Publishers. Disponível em: <<http://link.springer.com/10.1007/BF02478259>>. Acesso em: 23/9/2017.

MEDINA, C.; VELAZCO, S.; SALINAS, J. Experimental control of simple pendulum model. In: Springer (Org.); **The Pendulum: Scientific, Historical, Philosophical and Educational Perspectives**. p.67–76, 2005. 2005.

MEHROTRA, H.; VATSA, M.; SINGH, R.; MAJHI, B. Biometric match score fusion using RVM: A case study in multi-unit iris recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. **Anais...** . p.96–184, 2012. India: IEEE.

MEIRA, A. S. Controle adaptativo aplicado em um robô manipulador eletropneumático de três graus de liberdade cartesiano. , 19. dez. 2010. João Pessoa: Universidade Federal da Paraíba. Disponível em: <http://tede.biblioteca.ufpb.br/handle/tede/5410?locale=pt_BR>. Acesso em: 23/9/2017.

MITCHELL, T. **Machine Learning**. Nova Iorque: McGraw-Hill, 1997.

NIJHOLT, A. The future of brain-computer interfacing (keynote paper). 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV). **Anais...** .

p.156–161, 2016. Dhaka, Bangladesh: IEEE. Disponível em: <<http://ieeexplore.ieee.org/document/7759987/>>. Acesso em: 24/9/2017.

PECK, M. Refuted: EEG Study Detecting Responsiveness In Vegetative Patients - IEEE Spectrum. Disponível em: <<https://spectrum.ieee.org/tech-talk/biomedical/diagnostics/refuted-eeg-study-detecting-responsiveness-in-vegetative-patients>>. Acesso em: 23/9/2017.

PENARANDA, B. N.; BALDWIN, C. L. Temporal Factors of EEG and Artificial Neural Network Classifiers of Mental Workload. **Proceedings of the Human Factors and Ergonomics Society Annual Meeting**, v. 56, n. 1, p. 188–192, 2012. Disponível em: <<http://pro.sagepub.com/content/56/1/188%5Cnhttp://pro.sagepub.com/content/56/1/188.full.pdf>>.

PETERMAN, R. Controlling Servos with Emotiv Epoc Headset (BCI) and Arduino. Disponível em: <<https://github.com/renatopeterman/emotiv-epoc-servomotor-control>>. Acesso em: 23/9/2017.

QUINLAN, J. R. Induction of Decision Trees. **Machine Learning**, 1986.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. 1992.

RAMADAN, R. A.; VASILAKOS, A. V. Brain computer interface: control signals review. **Neurocomputing**, v. 223, n. 1, p. 26–44, 2017.

REBSAMEN, B. **A Brain Controlled Wheelchair To Navigate in Familiar Environments a Thesis Submitted for the Degree of Doctor of Philosophy Department of Mechanical Engineering**, 2009. National university of singapore.

RODRIGUES, M. D.; COELHO, L. DOS S. Controle de software utilizando o Neuroheadset Emotiv Epoc. , 2013. Curitiba: Pontifícia Universidade Católica do Paraná.

ROY, R.; MAHADEVAPPA, M.; KUMAR, C. S. Trajectory Path Planning of EEG

Controlled Robotic Arm Using GA. In: U. S. Tiwary (Org.); *Procedia Computer Science*. **Anais...** . p.147–151, 2016. Elsevier Science.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, 1986.

SANCHEZ-FRAIRE, U.; PARRA-VEGA, V.; MARTINEZ-PEON, D.; et al. On the Brain Computer Robot Interface (BCRI) to Control Robots. In: C. C. Chan (Org.); *IFAC-PapersOnLine*. **Anais...** . p.154–159, 2015. Salvador-Bahia.

SANTOS, A.; ANDRADE, G.; REYNOSO-MEZA, G.; COELHO, L. USING MULTIOBJECTIVE OPTIMIZATION DESIGN FOR DRIVEN. COBEM-2017. **Anais...** . p.1–10, 2017. Curitiba.

SANTOS, A.; CORDEIRO, G.; BECCHI, G.; et al. Quadcopter Control Based on Electroencephalogram Headset and Hybrid Fuzzy Classifier. BFAL. **Anais...** . p.209–11, 2017. Patras.

SARKISYAN, H. **EEG based multi-finger prosthesis**, 2013. California State University. Disponível em: <<http://csus-dspace.calstate.edu/handle/10211.9/2128>>. .
 SETIONO, R. Feedforward neural network construction using cross validation. **Neural computation**, v. 13, n. 12, p. 365–375, 2001.

SETIONO, R.; LU, G. L. G. Image compression using a feedforward neural network. **Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)**, v. 5, n. 2, p. 15681–15684, 1994.

SOUZA, O. B. DE. Controle adaptativo aplicado em dois elos de um robô manipulador eletropneumático de três graus de liberdade cartesiano. , 3. ago. 2010. João Pessoa: Universidade Federal da Paraíba. Disponível em: <<http://tede.biblioteca.ufpb.br/handle/tede/5398?mode=full>>. Acesso em: 23/9/2017.

STAMPS, K.; HAMAM, Y. Towards inexpensive BCI control for wheelchair navigation in the enabled environment - A hardware survey. In: springer (Org.); *Lecture Notes in*

Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). **Anais...** . p.336–345, 2010. Toronto-Canada.

SUNDARAM, M.; SADLER, R. M.; YOUNG, G. B.; PILLAY, N. EEG in epilepsy: current perspectives. **The Canadian journal of neurological sciences. Le journal canadien des sciences neurologiques**, v. 26, n. 4, p. 152–183, 1999.

TAHERKHANI, A. Recognizing sorting algorithms with the C4.5 decision tree classifier. IEEE International Conference on Program Comprehension. **Anais...** , 2010. Braga.

TIPPING, M. E. The Relevance Vector Machine. Advances in Neural Information Processing Systems. **Anais...** . p.211–244, 2000.

VAPNIK, V.; LERNER, A. Pattern recognition using generalized portrait method. **Automation and remote control**, v. 1, n. 1, p. 10–19, 1963.

WALTZ, E. Simple EEG Might Help Spot Autism in Children - IEEE Spectrum. Disponível em: <<https://spectrum.ieee.org/tech-talk/biomedical/diagnostics/simple-eeg-test-might-help-spot-autism-in-children>>. Acesso em: 23/9/2017.

WOLPAW, J. R. Brain-computer interface research comes of age: Traditional assumptions meet emerging realities. **Journal of Motor Behavior**, v. 42, n. 6, p. 341–353, 2010.

XIAOPING, C.; HE, H. Immune Feedforward Neural Network for Fault Detection. **Tsinghua Science and Technology**, v. 16, n. 3, p. 272–277, 2011.

YUE, J.; ZHOU, Z.; JIANG, J.; LIU, Y.; HU, D. Balancing a simulated inverted pendulum through motor imagery: An EEG-based real-time control paradigm. **Neuroscience Letters**, v. 524, n. 2, p. 95–100, 2012.

ZAFIRIOU, E.; MORARI, M. Digital controllers for SISO systems: A review and a new algorithm. **International Journal of Control**, v. 5, n. 6, p. 81–138, 1985.

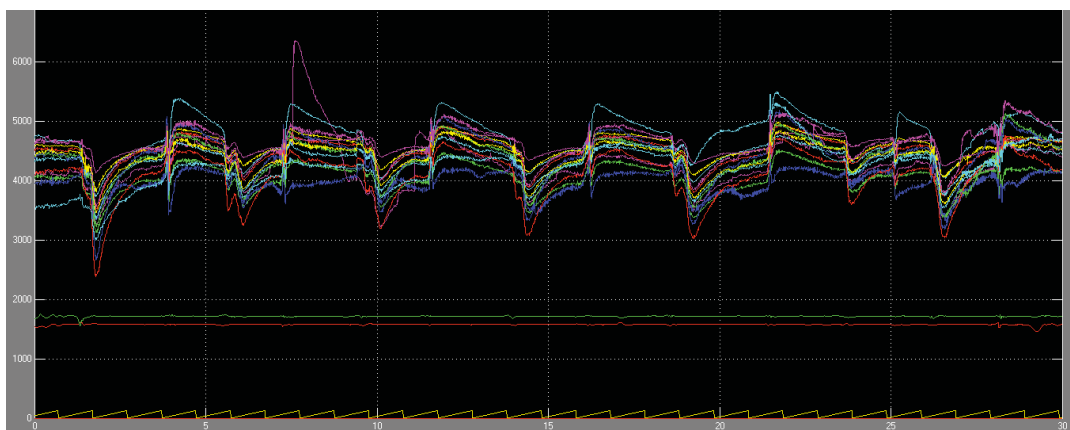
ZHANG, J.; YANG, S. An incremental-PID-controlled particle swarm optimization algorithm for EEG-data-based estimation of operator functional state. **Biomedical Signal Processing and Control**, v. 14, n. 11, p. 272–284, 2014.

ZHANG, Y.; CHEN, Y.; BRESSLER, S. L.; DING, M. Response preparation and inhibition: The role of the cortical sensorimotor beta rhythm. **Neuroscience**, v. 10, n. 8, p. 238–246, 2008.

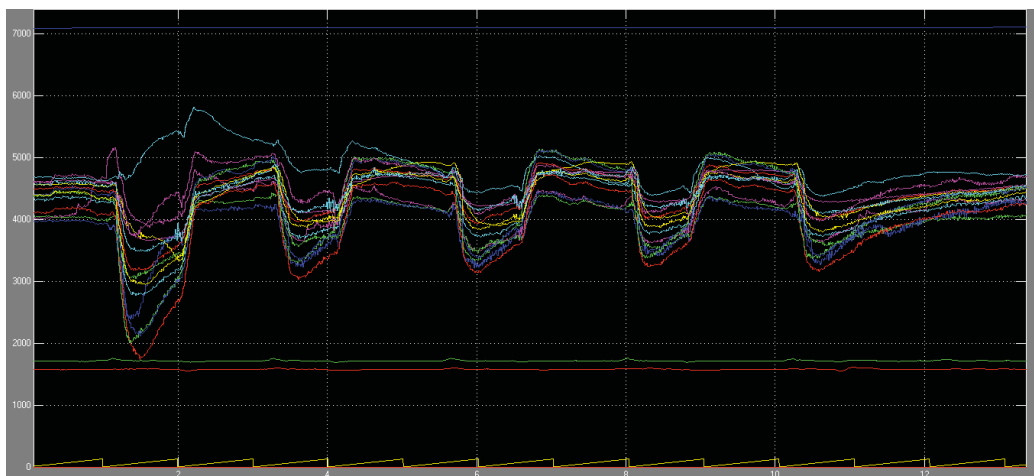
APÊNDICE A – Outras coletas de Dados Utilizando o *EMOTIV*

Exemplo de testes: dados captados utilizando o *headset EMOTIV* e plotados em um gráfico utilizando a ferramenta MATLAB. Os gráficos estão plotados no eixo x como sendo o *tempo* e no eixo y como tensão em uV (microvolts).

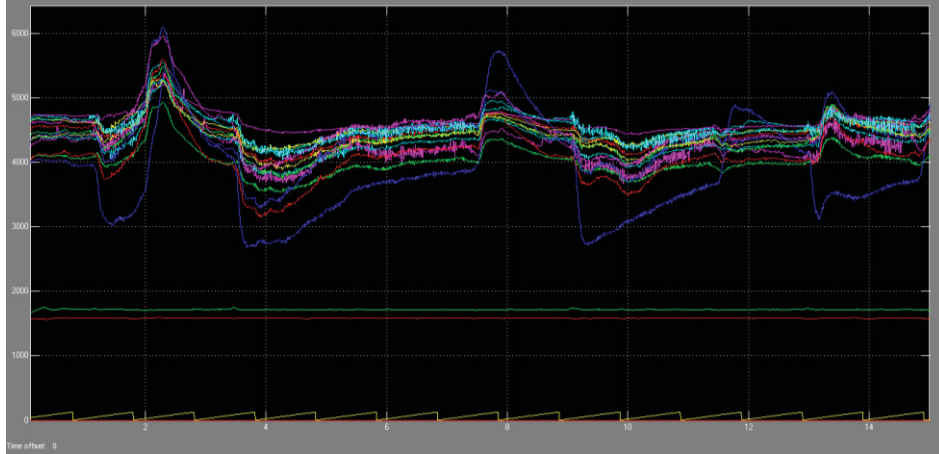
Abrir e fechar a boca:



Fechar olho com força:



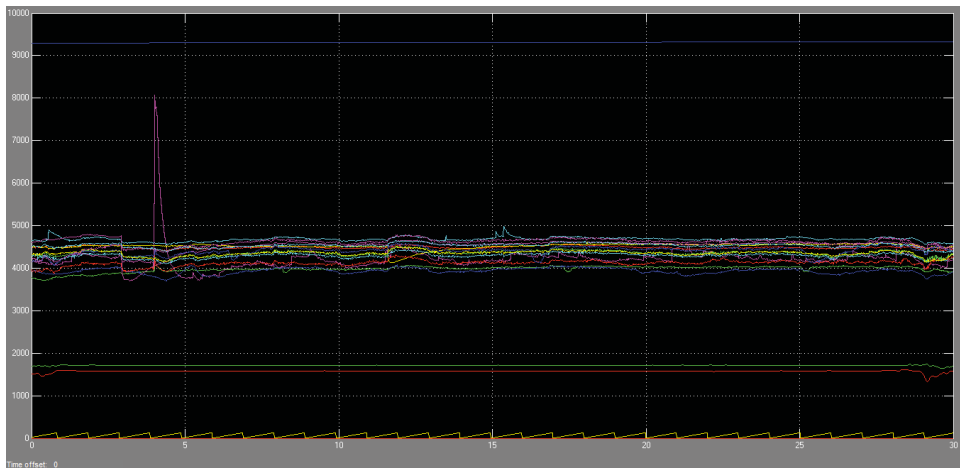
Levantar a sobrancelha:



Olhar para cima



Neutro:



APÊNDICE B – Algoritmo para controle do pêndulo

```

% -----
% Controle PID de Dahlin Adaptativo
% -----
clearvars

close all
clc

a=arduino('COM3','Mega2560');
a.configurePin('A5','AnalogInput');
a.configurePin('A8','Unset');
% a.flush;
a.writePWMMVotage(8,0) % write 0 to analog pin 8
%pause(5);
% Condições Iniciais
nit = 1000;
ts = 0.1;
umax = 255;
umin = 0;
u(1:2) = 0;
erro(1:2) = 0;
y(1:2) = 0;

% Parâmetros do Estimador
emq(1:2) = 0;
theta = [0.1 0.1 0.1 0.1];
p = 1000*eye(4,4);

% Inicia comunicação com Arduino

pause(5)
% Referência

yr(1:nit) = 800;

tau = input('tau = ');
% Parâmetros do Controlador
%tau = 0.3;
lambda = 1/tau;
que = exp(-lambda*ts);
N = 1; %Indica a presença de integradores

% Inicialização da Malha Fechada
for t = 3:nit
    % Recebe medição
    for j=1:3
        media(j) = a.readVoltage(5);
    end
    y(t) = round(mean(media));
    % Estimador dos MQR

```

```

vm = [-y(t-1) -y(t-2) u(t-1) u(t-2)];
yhat = vm*theta';
emq(t) = y(t) - yhat;
ganho = (p*vm') / (1 + vm*p*vm');
theta = theta + (ganho')*emq(t);
p = p - ganho*(1 + vm*p*vm')*(ganho');
a1(t) = theta(1,1);
a2(t) = theta(1,2);
b0(t) = theta(1,3);
b1(t) = theta(1,4);
% Calcula Lei de Controle
kax = (1 - que)/(b0(t) + b1(t)*(1 + N*(1 - que)));
kc(t) = -kax*(a1(t) + 2*a2(t));
ti(t) = -(a1(t) + 2*a2(t))*ts/(1 + a1(t) + a2(t));
td(t) = -(a2(t)*ts)/(a1(t) + 2*a2(t));
erro(t) = yr(t) - y(t);
u(t) = round(u(t-1) + (kc(t)*(1 + ts/ti(t) + td(t)/ts))*erro(t) + (-kc(t)*(1 +
2*td(t)/ts))*erro(t-1) + ...
(kc(t)*(td(t)/ts))*erro(t-2));
% Saturação
if u(t) >= umax
    u(t) = umax;
elseif u(t) <= umin
    u(t) = umin;
end

if(u(t)>0)
    u(t) = u(t)*(5/255);
end

a.writePWMMVltage(8,u(t));
end

a.writePWMMVltage(8,0);
% Resultados
figure(1)
t=1:nit;
y = y-50;
plot(t,y(t).*102.3/1023,'r')
hold on
yr = yr - 50;
plot(t,yr(t).*102.3/1023,'--k')
title('y e yr')
xlabel('Amostra');
ylabel('Angulo [°]')
figure(2)
plot(t,u(t)),title('u'),xlabel('Amostra');
ylabel('PWM');
figure(3)
plot(t,erro(t).*102.3/1023),title('Erro'),xlabel('Amostra');
figure(4)
t = 1: nit;
subplot(3,1,1),plot(t,kc(t)),title('Kc'),xlabel('Amostra');
subplot(3,1,2),plot(t,ti(t)),title('Ti'),xlabel('Amostra');
subplot(3,1,3),plot(t,td(t)),title('Td'),xlabel('Amostra');

u_medio = mean(u);

```

```
u_varia = cov(u);
u_esforco = std(u);
erro_medio = mean(erro);
erro_varia = cov(erro);

ITAE = sum(t.*abs(erro))/10;

% alguma análise de desempenho em malha fechada
fprintf(' Controlador PID Adaptativo de Dahlin\n');
fprintf(' Sinal de controle, u -> médio: %f\n esforço do controlador:
%f\n',u_medio,u_esforco);
fprintf(' Sinal de erro, e -> médio: %f\n',erro_medio);
fprintf(' ITAE: %f\n',ITAE)
```

APÊNDICE C – Esquema elétrico geral para ligação do pêndulo

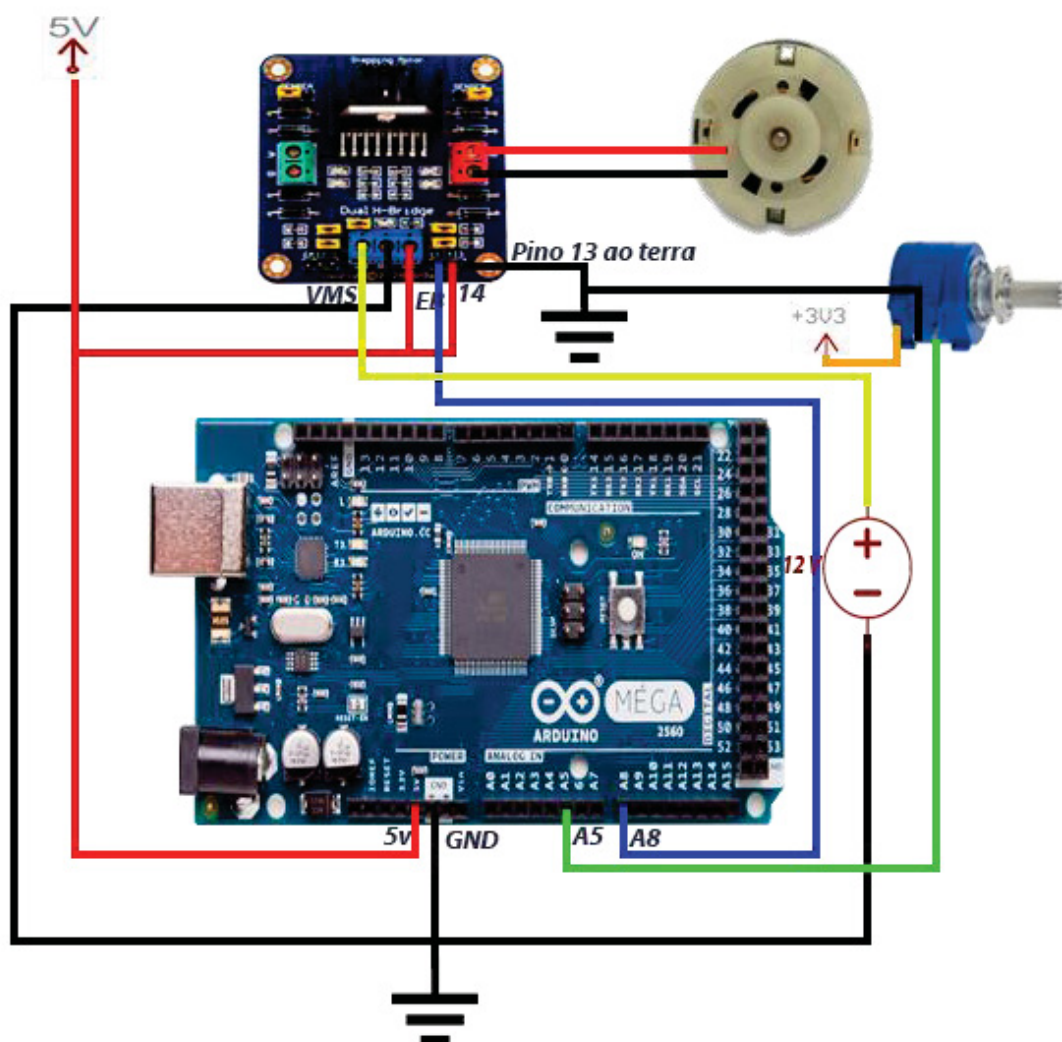


Figura 52 - Esquema elétrico de ligação do pêndulo

FONTE: O autor, 2017