Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica
Curso de Engenharia Biomédica

LUCAS LEMOS FRANCO

SOFTWARE PARA VISUALIZAÇÃO DA TRAJETÓRIA DE UM CURSOR CONTROLADO POR MEIO DA ELETROMIOGRAFIA DE SUPERFÍCIE

Uberlândia
2018

LUCAS LEMOS FRANCO

SOFTWARE PARA VISUALIZAÇÃO DA TRAJETÓRIA DE UM CURSOR CONTROLADO POR MEIO DA ELETROMIOGRAFIA DE SUPERFÍCIE

Trabalho apresentado como requisito parcial de avaliação na disciplina Trabalho de Conclusão de Curso do Curso de Engenharia Biomédica da Universidade Federal de Uberlândia.

Orientador: Adriano de Oliveira Andrade

_____

Assinatura do Orientador

Uberlândia

2018

Dedico este trabalho a Deus, pois sem Ele nada poderia ter sido feito.

AGRADECIMENTOS

Agradeço, antes de tudo, ao meu Senhor Jesus Cristo. Agradeço à minha querida mãe, Izilda, por todo o auxílio, conselhos e paciência. Aos familiares, amigos, colegas e professores, em especial ao professor Edgard Afonso Lamounier Júnior e ao professor Stéfano Paschoal, por terem me dado apoio fundamental nesse período. Agradeço o auxílio dos professores Adriano de Oliveira Andrade e Carlos Magno Medeiros Queiroz neste trabalho, pois sem eles não seria possível ter sido feito.

RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação de software, a qual se trata de uma interface gráfica, que tem por objetivo exibir o percurso do cursor de um *mouse* controlado por sinais eletromiográficos. Trata-se de uma ferramenta para auxílio na exibição de dados previamente coletados. O trabalho proposto visa também correlacionar temporalmente o percurso do *mouse* com o sinal eletrofisiológico coletado do músculo do voluntário, a fim de que seja possível fazer uma avaliação visual dos movimentos do *mouse*, comparando-os com as inserções de comando feitas a partir da contração do músculo.

ABSTRACT

This work presents the development of a software application, a GUI (Graphic User Interface), which has as purpose the exhibition of the path of the mouse cursor controlled by electromyographic signals. In other words, it is about a tool to help on the display of previous collected data. This present work also aims to correlate this mouse cursor path with the electrophysiological signal collected from a volunteer's muscle, in order that it may be possible to make a visual assessment of the mouse movements, comparing them to the input commands that were made voluntarily by the muscle's contraction.

LISTA DE ILUSTRAÇÕES

FIGURA 1 - Modelo em Cascata 15
FIGURA 2 - Representação da interface de aprendizado..... 19
FIGURA 3 - Parâmetros de entrada..... 24
FIGURA 4 - Representação gráfica do percurso do mouse (cima) e sinal EMG (embaixo).....24
FIGURA 5 - Funcionalidades da interface.....26
FIGURA 6 - Percurso feito pelo usuário entre os Alvos 2 e 3.....27
QUADRO 1 – Subdivisões da coleta completa de um voluntário.....19

LISTA DE TABELAS

TABELA 1 – Comandos de mouse de acordo e suas respectivas durações de contração..... 18
TABELA 2 - Exemplos de ações extraídas de um arquivo gerado 20
TABELA 3 – Protocolos e medidas dos alvos em pixels..... 21
TABELA 4 – Tempo e Ação em um experimento..... 22
TABELA 5 – Siglas e suas correspondentes ações..... 25

LISTA DE ABREVIATURAS E SIGLAS

UFU – Universidade Federal de Uberlândia

FEELT – Faculdade de Engenharia Elétrica

ABNT – Associação Brasileira de Normas Técnicas

EMG – Eletromiografia

EEG – Eletroencefalografia

CVDS - Ciclo de Vida de Desenvolvimento de Software

PA – Potencial de Ação

UM – Unidade Motora

IHC– Interface Homem-Computador

IHM - Interface Homem-Máquina

SUMÁRIO

1 INTRODUÇÃO 11
1.1 MOTIVAÇÃO 11
1.2 OBJETIVO GERAL 11
1.3 OBJETIVOS ESPECÍFICOS 11
2 FUNDAMENTAÇÃO TEÓRICA 12
2.1 ELETROMIOGRAFIA E ENCEFALOGRAFIA 12
2.2 INTERFACE HOMEM-COMPUTADOR 12
2.3 MOUSE EMULATION 13
2.4 DESENVOLVIMENTO DE SOFTWARE 14
3 METODOLOGIA 16
3.1 COLETA DE DADOS E VOLUNTÁRIOS 16
3.2 INTERFACE DE APRENDIZADO E DADOS GERADOS 16
3.3 SISTEMA PROPOSTO 20
4 RESULTADOS E DISCUSSÃO 23
5 CONCLUSÃO 28
6 REFERÊNCIAS 29

1 INTRODUÇÃO

1.1 Motivação

O crescimento tecnológico das útlimas décadas afetou positivamente diversas áreas da vida em sociedade, incluindo a área da saúde. Com o objetivo de promover ou facilitar a comunicação de pessoas com limitações físicas, surgem, a cada dia, diversos tipos de equipamentos. Esses equipamentos usam como parâmetros de entrada as mais variadas formas de comando: voz, visão, movimentos e até mesmo sinais eletrofisiológicos, como é o caso descrito neste presente trabalho. A partir da compreensão das capacidades motoras remanescentes dos futuros usuários, por exemplo, é possível elaborar sistemas de comunicação que sejam adequados à tais capacidades e permitam uma melhora na qualidade de vida do usuário, trazendo conforto e comodidade a este. Para que o sistema seja colocado em uso, é necessária a criação de ferramentas de software que permitam uma melhor análise dos dados coletados, a fim de determinar a acurácia do sistema.

1.2 Objetivo Geral

A partir da necessidade de criar uma ferramenta que seja útil para a análise de dados coletados em um determinado experimento de eletromiografia, esse trabalho visa discorrer sobre os processos e etapas da construção do aplicativo de software desenvolvido para satisfazer tal demanda.

1.3 Objetivos específicos

Esse trabalho tem busca facilitar a análise dos dados coletados em um sistema de Interface Homem-Computador que utiliza sinais eletromiográficos como parâmetros de entrada para manipular o *mouse* e executar determinadas tarefas, por meio da criação de uma interface gráfica que exibe tais dados ao longo do tempo. A partir dos dados coletados, espera-se sincronizar o percurso do cursor do mouse controlado pelo voluntário com os correspondentes sinais eletrofisiológicos que servem de comando, a fim de facilitar a visualização do sistema, permitindo futuras análises.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Eletromiografia e Encefalografia

Eletromiografia (EMG) é o estudo dos sinais provenientes de músculos envolvendo a sua detecção, análise e variadas aplicações. Embora o termo “eletromiográfico” faça referência à geração de gráficos - uma vez que no passado os estudos dos sinais mioelétricos só podiam ser representados pelos gráficos gerados - ele continua sendo fortemente usado, ainda que muitos médicos prefiram utilizar o termo “sinal mioelétrico”. (DE LUCA, 2006). A eletromiografia de superfície são biopotenciais que representam a atividade elétrica gerada pela musculatura e coletada sobre a superfície da pele. Um eletrodo de EMG é um transdutor da corrente iônica, que flui no tecido biológico, em corrente elétrica, a qual flui nos condutores metálicos (HERMENS et al., 2010).

A eletroencefalografia é a técnica médica que lê a atividade elétrica gerada pelas estruturas cerebrais, a partir do escalpo. O eletroencefalograma (EEG) pode ser definido como a atividade elétrica em si, coletada diretamente do escalpo, por meio de eletrodos de metal e um meio condutor (NIEDERMEYER; SILVA, 1993).

2.2 Interface Homem-Computador

Nos últimos anos, um considerável esforço tem sido empregado para o desenvolvimento de interfaces que seja *user-friendly* (amigável ao usuário). Tais projetos têm, em geral, como entrada, parâmetros como voz, visão, gestos etc. Um dos grandes desafios é estabelecer um vínculo entre o computador e os sinais eletrofisiológicos do usuário, ao explorar as características elétricas do sistema nervoso. Dessa forma, o interesse em explorar sinais bioelétricos, como EMG, EEG e EOG (eletroculografia), tem crescido expressivamente, tendo com finalidade o desenvolvimento de interfaces homem-máquina. (MOON, I.; LEE, M.; MUN, 2004)

Segundo Barreto et al. (2000, p. 53)

> “Um vasto número de pessoas sofre de deficiências motoras severas, como lesão medular, esclerose lateral amiotrófica (ELA) etc [...]. A qualidade de vida desses indvíduos poderia ser

> melhorada ao ser provido a eles um meio confiável e prático de se utilizar o computador (PC). [...] Essa melhora pode vir de ao menos duas formas:
>
> - Aumento na integração à sociedade e à produtividade ao comunicar e trabalhar, utilizando computador e suas aplicações de software padrão.
> - Aumento de controle não-assistido sobre o ambiente de tais pacientes, usando aplicações de software dedicadas e dispositivos para ligar e desligar aparelhos, alarmes etc (ON/OFF)."

Com o objetivo de implementar soluções que sejam eficazes para pessoas com deficiências ou certas dificuldades, diversas tentativas utilizam sinais biomédicos. Tais dispositivos criados são baseados na capacidade motora remanecente do usuário. (AHSAN; IBRAHIMY, 2009)

Entre os diversos sinais bioelétricos utilizados atualmente, o sinal EMG pode ser estimado como a fonte de novas técnicas de IHM, atuando este como parâmetro de entrada. A atividade elétrica induzida pelo movimento do músculo do braço humano pode ser interpretada e transformada como *input* para um dado controle do computador. Além disso, o sinal EMG pode ser facilmente coletado diretamente sobre pele, por meio de eletrodos de superfície. (AHSAN; IBRAHIMY; KHALIFA, 2009)

2.3 Mouse Emulation

Os computadores de hoje são normalmente do tipo interface gráfica para o usuário. Assim sendo, a grande maioria das operações consiste em selecionar comandos, ícones etc. Selecionar requer duas capacidades básicas de seu usuário: apontar, que é posicionar o cursor sobre o alvo desejado na tela, e clicar, que é a função de *mouse down/up*, completando a seleção (BARRETO; SCARGLE; ADJOUADI, 2000).

Um sistema que promove um aumento na comunicação ou uma comunicação alternativa é chamado de *AAC System* (*Aumentative and Alternative Communication System)*. Aplicações de IHC mais comuns se baseiam no controle de Sistemas AAC. Pode-se considerar que tais sistemas funcionam como ferramenta para estender a capacidade de comunicação de indivíduos que possuem limitações nessa área. Sistemas AAC promovem, muitas vezes, comunicação por meio de um computador. (BARRETO; SCARGLE; ADJOUADI, 2000)

Para esse tipo de sistema, uma funcionalidade imprenscindível seria o controle de um cursor para manipular a interface gráfica do computador. A título de exemplo, pacientes amputados, tetraplégicos ou que sofrem de doenças de ordem neuromuscular podem utilizar os sinais EMG de músculos como parâmetro de inserção de dados em interfaces que convertem tais sinais em ações do *mouse*, como o *click*. Assim, a interface pode permitir que o paciente escreva, selecione tarefas etc. (BARRETO; SCARGLE; ADJOUADI, 2000)

2.4 Desenvolvimento de Software

A Engenharia de Software é a aplicação dos conceitos de engenharia, objetivando a criação de um software, que seja feito de maneira econômica, resultando em um produto confiável e que possa ser usado de forma eficiente nos computadores reais. (PRESMAN, 2002)

O primeiro modelo de desenvolvimento de software a ser publicado (ROYCE, 1970) está exemplificado na Figura 1 abaixo. Por ter eventos encadeados, é chamado de Modelo em Cascata ou Ciclo de Vida do Software (CVDS). (SOMMERVILLE, 2007)

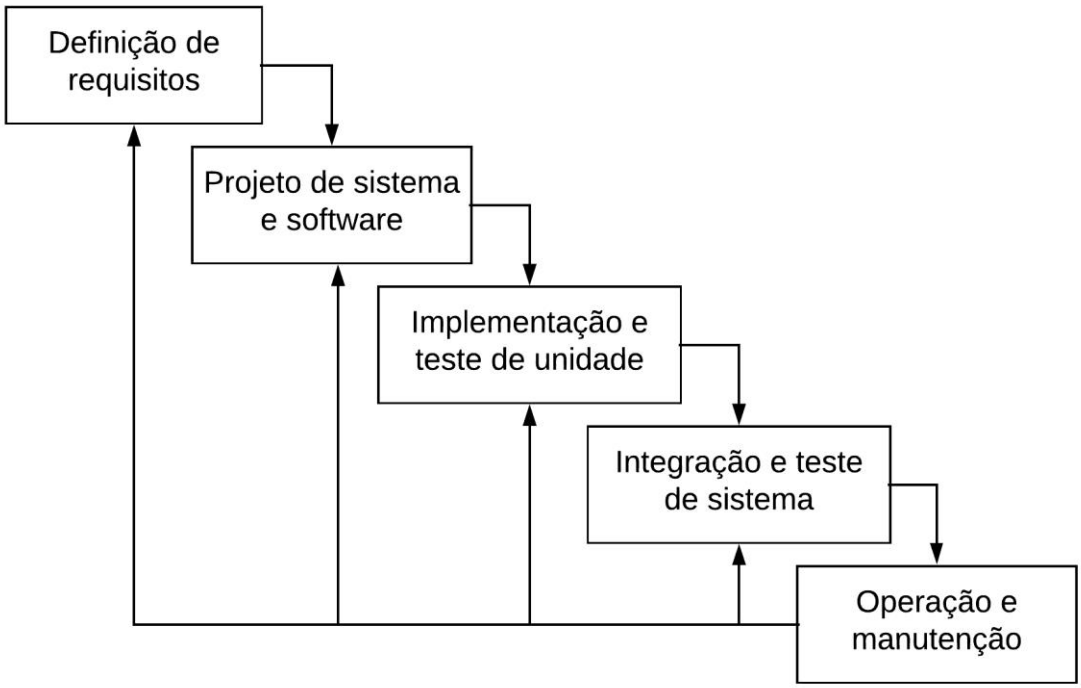

Figura 1: Modelo em Cascata

O Ciclo de Vida de Desenvolvimento de Software (CVDS) é definido pela a norma NBR ISO/IEC 12207/1998 como

> “Estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema, desde a definição de seus requisitos até o término de seu uso.”

Segundo a literatura (SOMMERVILLE, 2007), os principais estágios do modelo são:

1 - **Análise de definição de requisitos:** Os serviços, restrições e objetivos do sistema são definidos por meio consultas a usuários do sistema [...].

2 - **Projeto de sistema e software:** O processo de projeto de sistema divide os requisitos em sistemas de hardware ou de software. Ele estabelece a arquitetura geral do processo [...].

3 - **Implementação e teste de unidade:** Durante esse estágio, o projeto de software é realizado com um conjunto de programas ou unidades de programa [...].

4 - **Integração e teste de sistema:** As unidades individuais de programa ou os programas são integrados e testados como um sistema completo para garantir que os requisitos de software foram atendidos [...].

5 - **Operação e manutenção:** Geralmente (embora não necessáriamente) esta é a fase mais longa do ciclo de vida. O sistema é instalado e colocado em operação. A manutenção envolve a correção de erros não detectados nos estágios anteriores do ciclo de vida, no aprimoramento da implementação das unidades de sistema e na amplificação dos serviços de sistema à medida que novos requisitos são identificados.

3 METODOLOGIA

Embora a coleta de dados e a interface de aprendizado não tenham sido feitas pelo presente autor, é de fundamental importância explicar detalhadamente ambas, uma vez de que estas são a base para o trabalho desenvolvido.

3.1 Coleta de Dados e Voluntários

A coleta de dados foi feita por uma equipe de três integrantes do laboratório NIATS, da Universidade Federal de Uberlândia, que contou com onze voluntários saudáveis, que não possuíam nenhum tipo de limitação neuromuscular, os quais se dispuseram livremente a participar das coletas.

Em cada coleta, um eletrodo de EMG é posicionado acima do músculo temporal, na cabeça, para detectar a sua atividade mioelétrica. A detecção do sinal e a identificação da contração são feitas em tempo real, por meio de um dispositivo chamado Itan. O objetivo ao detectar a contração é de determinar o tempo em que o músculo se contrai.

Colocam-se também eletrodos de EEG no voluntário para captar as atividades neurais durante a execução das tarefas. A diferença é que os sinais EEG são apenas armazenados e não servem como parâmetro de entrada para comandos, como os sinais EMG, e somente serão usados para análises futuras, que não farão parte deste presente trabalho.

3.2 Interface de Aprendizado e Dados Gerados

O voluntário, durante a coleta, posiciona-se sentado ereto diante de uma interface de aprendizado apresentada numa tela de computador. Pede-se a ele que execute determinadas tarefas. Uma máquina de estado, criada dentro do emulador de *mouse*, tem o tempo de contração muscular como parâmetro de entrada. Portanto, o sinal eletromiográfico é processado em tempo real, a fim de se obter a envoltória da contração, já que a sua duração está relacionada aos comandos existentes.

Conforme a Tabela 1 abaixo, uma contração menor que 300 milisegundos representa o comando de rotacionar o cursor do *mouse* em 90º no sentido horário.

Uma contração que for maior que 300 ms e menor que 1 segundo representa o comando de movimento do *mouse* na direção em que a ponta do cursor aponta. E por fim, uma contração maior que 1s representa o comando de clique do mouse. Como o comando para movimentar o *mouse* (contração de 300ms a 1s) é apenas no início do movimento, o usuário deverá fazer uma contração de duração qualquer quando desejar parar de movimentar o cursor. Dessa forma, o voluntário pode controlar o cursor do mouse e desempenhar as tarefas que lhe são solicitadas.

Tabela 1: Comandos de mouse de acordo e suas respectivas durações de contração

Duração da contração	Comando
0 – 300 ms	Rotação
300 – 1000 ms	Movimento
> 1000 ms	Click

Para cada voluntário, a coleta acontece durante cinco dias seguidos, sendo cada dia chamado de *Seção*. Assim, existe a *Seção 1*, *Seção 2, Seção 3, Seção 4* e *Seção 5*, cada uma delas representando um dia, em ordem numérica. Em uma sessão, o usuário deve atingir os *Targets* (alvos, do inglês), que podem ser de três tamanhos distintos (*big, medium, small* – grande, médio e pequeno). Cada tamanho representa o *Protocol* (protocolo, do inglês) da coleta, sendo o alvo grande o *Protocol 1*, o médio seria o *Protocol 2*, e o pequeno seria o *Protocol 3*. E por último, em cada uma das sessões, para cada um dos protocolos acima, são feitas cinco repetições, nomeadas *Repetition 1, Repetition 2, Repetition 3, Repetition 4, Repetition 5*. Dessa forma, para um voluntário, são geradas 75 matrizes dados, resultado este obtido simplesmente pelo produto de 5 sessões por 3 protocolos e por 5 repetições. O Quadro 1 representa as subdivisões da coleta de um único voluntário.

Section	Protocol	Repetition
1 2 3 4 5	1 - BIG 2 - MEDIUM 3 - SMALL	1 2 3 4 5

Quadro 1: Subdivisões da coleta completa de um voluntário. Fonte: próprio autor.

Para facilitar a notação, utiliza-se, por exemplo, o termo S1 P2 R5, para se referir ao experimento resultante da Sessão 1, Protocolo 2, Repetição 5 de um determinado voluntário. Na interface gráfica, os alvos são posicionados de forma equidistante do centro. O *Target_1*, ou seja, o Alvo 1 é sempre posicionado à cima, o *Target*_2 (Alvo 2) à direita, o *Target*_3 (Alvo 3) abaixo e o *Target*_4 (Alvo 4) à esquerda, conforme a Fig 2. As dimensões da interface são 1366 pixels de largura por 768 pixels de altura. E a referência da tela, ou seja, o ponto de coordenadas x, y iguais a (0,0) é o canto superior esquerdo.

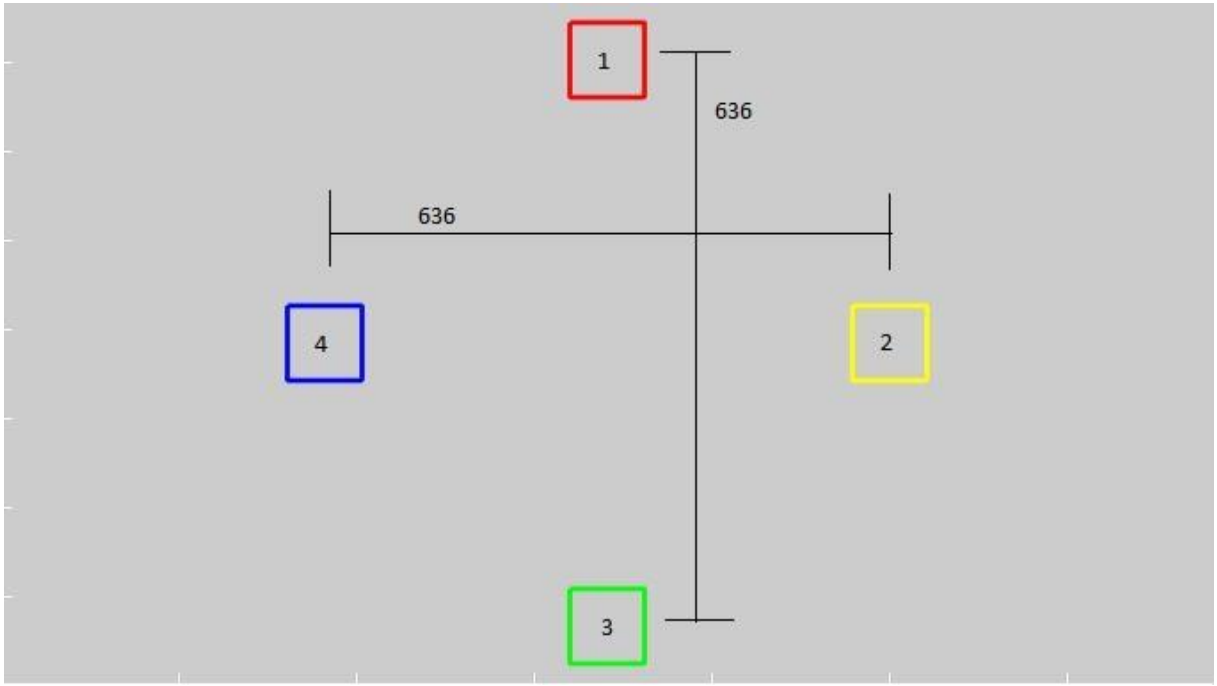

Figura 2: Representação da interface de aprendizado - medida em pixels. Fonte: próprio autor

Apenas um alvo aparece por vez e cabe ao voluntário clicar sobre este alvo utilizando o cursor. A ordem em que os alvos aparecem é aleatória, de forma que o

voluntário não pode prever o próximo alvo. Porém, o último alvo a aparecer é o mesmo que o primeiro, uma vez que o voluntário deve executar cinco *CLICK_IN*, isto é, cinco cliques corretos, cada um sobre um alvo. Por exemplo, os alvos poderiam aparecer na seguinte ordem: 2, 3, 1, 4, 2. Isto significa que o primeiro a aparecer seria Alvo 2, seguido do Alvo 3, depois Alvo 1, Alvo 4 e por último seria o Alvo 2 novamente. Quando o alvo é clicado, ele desaparece. Após um tempo aleatório, o próximo alvo na sequência aparece.

Cada clique ou aparecimento de alvo é considerado uma ação (*Action)*. Além desses dois, existem o *CLICK_OUT*, que é quando o usuário clica fora do alvo atual, e o *MOVE*, que representa o movimento do cursor, seja em linha reta ou uma rotação. Os dados gerados em uma repetição correlacionam o tempo com cada um dos eventos, como exemplificado na Tabela 2 abaixo.

Tabela 2: Exemplos de ações extraídas de um arquivo gerado

TIME	ACTION	X_POS	Y_POS	WIDTH	HEIGHT
20/06/2017 11:33:51.435	TARGET_1	673	56	21	21
20/06/2017 11:33:55.919	CLICK_OUT	62	110	0	0
20/06/2017 11:33:58.903	MOVE	105	110	0	0
20/06/2017 11:34:14.029	CLICK_IN	689	57	0	0
20/06/2017 11:34:17.120	TARGET_3	673	692	21	21

Na Tabela 2 se encontram apenas algumas ações que ocorreram em uma dada repetição, de um certo voluntário (V11 S2 P3 R4). Na coluna *Time* (tempo), por exemplo, encontra-se o momento exato em que ocorreu cada *Action*. Já *X_POS* e *Y_POS* representam a posição em pixels da ação nos eixos X e Y, respectivamente, da tela. *WIDTH* (largura) e *HEIGHT* (altura) representam o tamanho do alvo, em pixels, sendo diferentes de zero somente para ações de aparecimento de alvos (*Target_1, Target_3*). Como dito anteriormente, diferentes protocolos possuem alvos de diferentes tamanhos. A Tabela 3 relaciona os protocolos e seus respectivos tamanhos.

Tabela 3: Protocolos e medidas dos alvos em pixels

PROTOCOL	WIDTH	HEIGHT
1	84	84
2	42	42
3	21	21

Dessa forma, a interface de aprendizado permite o controle do cursor na tela, por meio dos comandos feitos a partir da contração muscular, a fim de que o usuário possa clicar nos alvos na ordem em que estes aparecem. Os dados, como a posição do cursor, o EMG e o EEG, ao longo do tempo, são armazenados para futuras análises.

3.3 Sistema Proposto

Seguindo o modelo em cascata, a primeira etapa, que é a *Análise e definição de requisitos*, foi feita em conjunto com o futuro usuári o qual avaliará o desempenho dos voluntários. Foi definido que o software deverá exibir, por meio de uma interface amigável, o percurso feito pelo cursor do *mouse* controlado pelo voluntário. Além disso, a reprodução dos movimentos do cursor sincronizada ao gráfico da atividade eletromiográfica do músculo temporal do voluntário trará grande vantagem, caso seja possível fazê-la. O usuário, que manipulará o software, poderá escolher qual o voluntário, a sessão, o protocolo e a repetição que desejar observar.

A segunda etapa do modelo é *Projeto de sistema e software*. O projeto foi feito desenvolvido em MATLAB versão 2017b. O Sistema Operacional em que foi desenvolvido é o Windows 10. O *software* acessa, de acordo com os parâmetros de entrada escolhidos pelo usuário, o arquivo em formato texto (.txt) correspondente à exata repetição escolhida. Além disso, ele exibe graficamente o sinal eletromiográfico com e sem filtragem, juntamente com o envelope da contração. O tamanho do envelope de cada contração permite supor o tipo de comando que provavelmente o emulador de *mouse* identificou. Visualmente, espera-se comparar o provável comando de *mouse* com o que realmente aconteceu, isto é, observando a movimentação do *mouse*.

A matriz dos dados do emulador de *mouse*, como exemplificado na Tabela 2, e os dados coletados pelo EMG não estão, a princípio, sincronizados. Para sincronizá-los, é necessário determinar o começo e o fim da repetição que está sendo analisada. Para tanto, considera-se o *tempo inicial* o momento em que ocorre o primeiro *CLICK_IN* e o *tempo final*, o momento em que ocorre o último.

Algo importante a se considerar é a taxa de amostragem de ambos os dados (emulador de mouse e EMG). Como o EMG tem taxa de amostragem de 1000 Hz, sigifica que se tem 1000 pontos em um segundo. Já o emulador de mouse não possui uma taxa de amostragem fixa, isto é, ele não capta as ações a cada X milisegundos, como o EMG. Ele possui uma periodicidade relativa, uma vez que ele capta quando há mudança de ação. *Actions* como *TARGET*, *CLICK_IN* e *CLICK_OUT* são pontuais, ou seja, ocorrem em um único instante. Já o *MOVE* (movimentação) não é instantâneo, mas dura um determinado tempo. O emulador de *mouse* capta de tempos em tempos as ações e com isso é possível determinar a posição do cursor do *mouse* ao analisar os valores de *X_POS* e *Y_POS* quando ocorre a ação *MOVE*, ao longo do tempo.

Para exemplificar a falta de periodicidade fixa da amostragem do emulador de *mouse*, a Tabela 4 mostra os seis primeiros valores de tempo (em milisegundos) e as ações correspondentes de um dado experimento, sendo o *tempo inicial* o momento do primeiro *CLICK_IN*, como dito anteriormente.

Tabela 4: Tempo e Ação em um experimento

TIME	ACTION
0	CLICK_IN
3104	TARGET_3
5041	MOVE
5151	MOVE
5479	MOVE
5588	MOVE

Observa-se que quando não há *Action* do tipo *MOVE*, como nos dois primeiros frames, o intervalo entre os frames é maior. Porém, a partir do início do *MOVE*, os intervalos diminuem, porém não são constantes. Para esse exato experimento (V1 S1 P1 R1), a variação entre dois frames teve como valor médio 392.9 ms, com um desvio padrão de 615.58 ms. Logo, pode-se perceber que o tempo entre dois frames varia bastante, e portanto, exibir o percurso do mouse em “tempo real”, isto é, simulando o tempo real, pode ser complicado. A abordagem escolhida será de promover uma nova base temporal, isto é, um novo vetor contendo os valores de tempo, passando estes a ocorrerem à uma taxa de 100 Hz. Por meio da função *interpol1* do MATLAB, é possível fazer uma interpolação, criando assim novos vetores X e Y, com o objetivo de criar valores de X e de Y intermediários a dois valores consecutivos, relacionados ao novo vetor tempo.

Com isso, espera-se que o movimento do *mouse* seja suavizado, evitando assim uma alta descretização, e que a exibição do gráfico da atividade eletromiográfica ocorra de forma gradual, sem promover "saltos", como ocorreria caso o plot fosse feito a cada frame de posição.

4 RESULTADOS E DISCUSSÃO

Os parâmetros iniciais de inserção por parte do usuário são mostrados na Figura 3 abaixo.

Figura 3: Parâmetros de entrada

Ao escolher o voluntário, a sessão, o protocolo e a repetição que se deseja observar, o usuário deve clicar em LOAD (carregar, do inglês), conforme a Fig. 3. O comando LOAD, como o próprio nome diz, “carrega” os dados do experimento escolhido, isto é, ele encontra o arquivo correspondente e importa seus dados. Em seguida, o software faz os cálculos devidos e está apto a exibir o percurso feito pelo voluntário e o correspondente sinal EMG, ambos sincronizados.

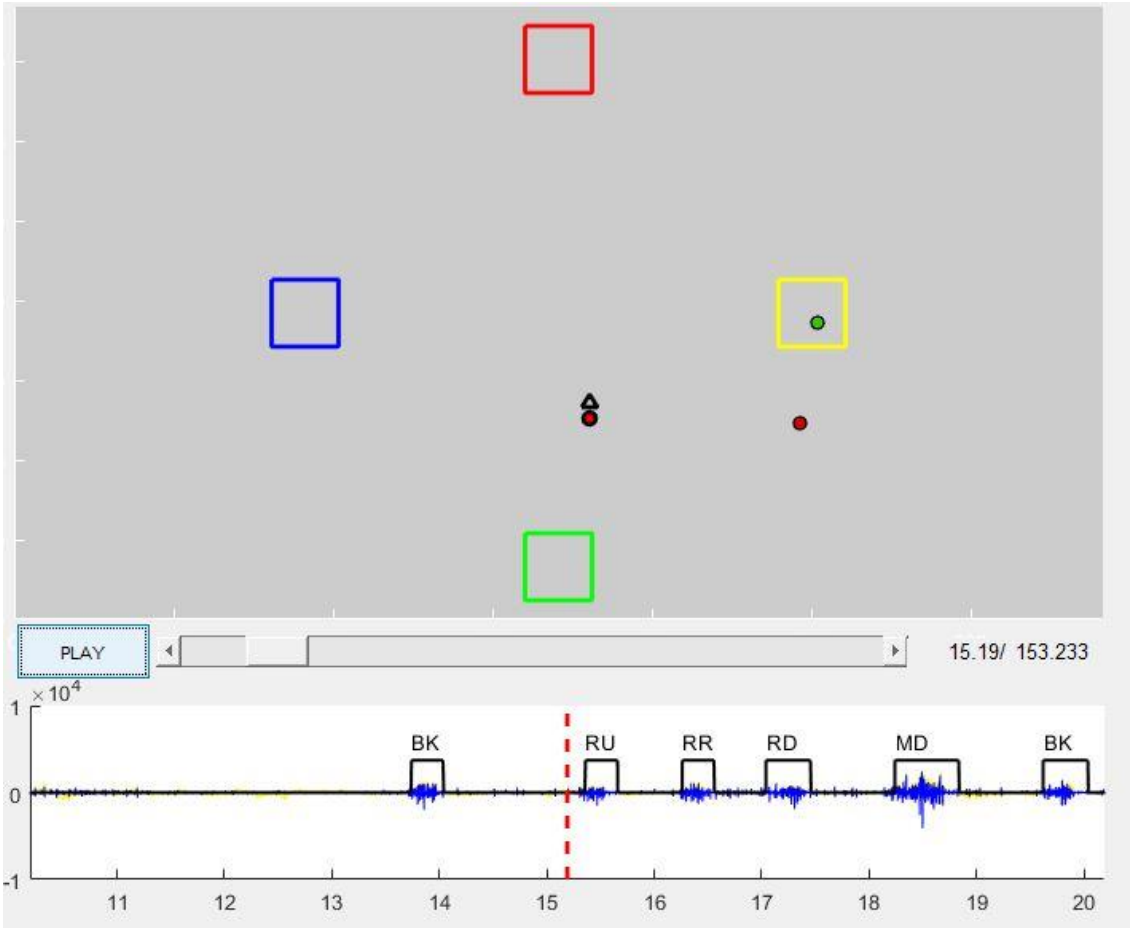

Figura 4: Representação gráfica do percurso do mouse (cima) e sinal EMG (embaixo)

Na Figura 4, existem dois gráficos. No gráfico superior, o quadrado vermelho representa o *Target_1*, o amarelo representa o *Target_2*, o verde representa o *Target_3* e o azul representa o *Target_4*. O círculo vermelho envolto por uma linha preta, próximo a um triângulo representa o cursor do *mouse*. O triângulo, por sua vez, aponta a direção na qual o cursor caminha em um dado instante. O círculo verde corresponde ao ponto da tela onde já ocorreu um *CLICK_IN*. O círculo vermelho, que não vem acompanhado do triângulo preto, representa o ponto na tela onde ocorreu o *CLICK_OUT*.

Já no gráfico inferior, também na Fig. 4, está a representação do sinal EMG. Em azul encontra-se o sinal EMG filtrado, em amarelo encontra-se o sinal EMG não-filtrado. Em preto está representada a detecção de contração. O eixo X corresponde ao valor temporal, em segundos, a partir do *tempo inicial*, que é a partir do primeiro *CLICK_IN*. O eixo Y é a amplitude do sinal, admensional. A linha vertical vermelha tracejada representa o instante atual, o qual está sincronizado com o que ocorre no gráfico acima em um dado instante. Acima de cada detecção de contração se encontra uma sigla, à qual se atribui uma ação, conforme a Tabela 5 abaixo.

Tabela 5: Siglas e suas correspondentes ações

SIGLA	AÇÃO
BK	*break*, isto é, parada
CK	click, seja CLICK_IN ou CLICK_OUT
MD, RD	*move down* (mover para baixo), *rotation down* (rotação para baixo)
MU, RU	*move up* (mover para cima), *rotation up* (rotação para cima)
MR, RR	*move right* (mover para direita), *rotation right* (rotação para direita)
ML, RL	*move left* (mover para esquerda), *rotation left* (rotação para esquerda)

O botão *PLAY* controla a movimentação dos gráficos, isto é, ele é o comando de início e pausa da exibição gráfica. Com o passar do tempo, a barra deslizante avança. É possível voltar ou adiantar o tempo da exibição, bastando apenas

deslocar o cursor da barra deslizante. Ao lado da barra deslizante está o tempo atual (15,19 segundos, na Figura 4, dividido pelo *tempo final*, que nesse experimento foi de 153,233 segundos).

Seguindo a terceira etapa do modelo em cascata, que é a *Implementação e teste de unidade*, foi testada cada uma das funcionalidades mostradas na Figura 5. As funcionalidades da interface buscam facilitar ainda mais a interpretação e a exibição dos dados.

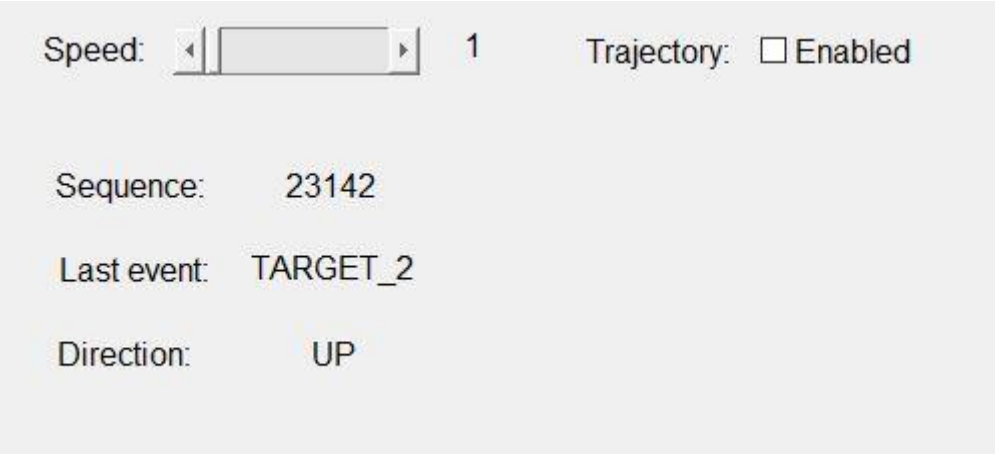

Figura 5: Funcionalidades da interface

A barra deslizante no canto superior esquerdo da Fig. 5 permite alterar a velocidade (*Speed*, em inglês) de exibição dos gráficos. Ao lado direito, encontra-se o *check box* chamado *Trajectory* (trajetória, em inglês), que uma vez habilitado, permite identificar o caminho feito pelo cursor do *mouse*, conforme a Fig. 6 abaixo.

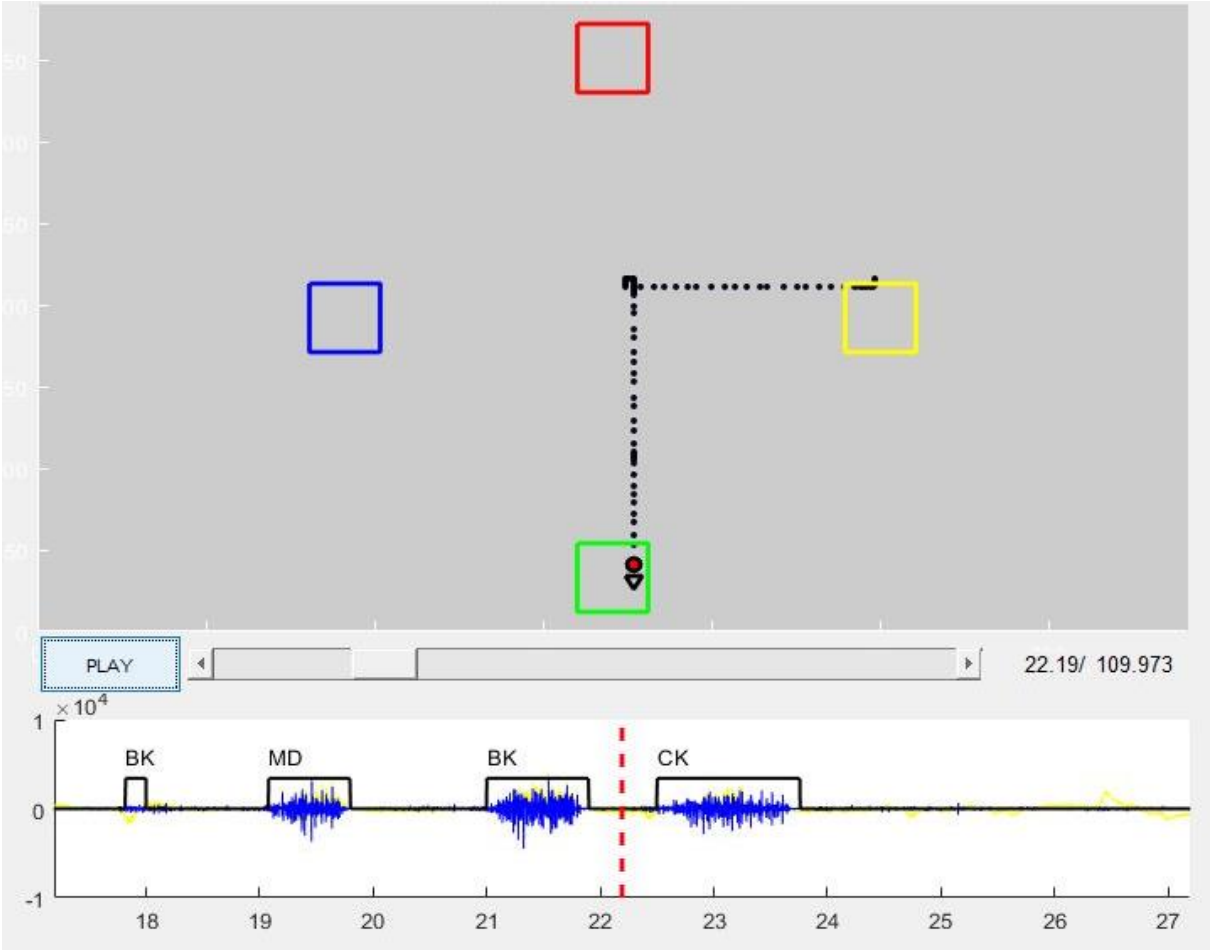

Figura 6: Percurso feito pelo usuário entre os Alvos 2 e 3

Além dessas duas funcionalidades, ainda na Fig. 5 se encontra um campo chamado *Sequence* (sequência, em inglês) que representa a sequência em que os alvos apareceram para o usuário nesse exato experimento. Já o campo *Last event* (último evento, em inglês) está relacionado ao último evento que ocorreu num dado instante, o qual na Tabela 4 foi chamado de *Action*, diferenciando apenas por não considerar *Action* do tipo *Move*. E por fim, mais abaixo, encontra-se o campo *Direction* (direção, em inglês), que corresponde à direção na qual o cursor do *mouse* está apontando.

A *Integração e teste de sistema*, que é a quarta etapa do modelo em cascata, foi feita, ao sistema ser testado com todas as suas funcionalidades ao mesmo tempo. Todas funcionaram perfeitamente, porém o tempo de execução aumentou, já que demandou um maior processamento de dados por parte do computador.

Por fim, a última etapa do modelo, que é a *Operação e manutenção*, ocorreu e ocorrerá enquanto o *software* estiver sendo usado pelo seu usuário, a fim de auxiliá-lo da melhor forma possível em suas análises de dado.

5 CONCLUSÃO

Conclui-se que o software cumpriu o objetivo proposto, que era criar uma interface para usuário que relaciona, de forma temporal, o percurso feito pelo cursor de *mouse* controlado por um voluntário, ao longo do tempo, com o sinal elétrico coletado no músculo, com suas devidas contrações. A interface é simples, bastante intuitiva e consegue exibir os sinais com veracidade. A interpolação dos valores X e Y da posição do cursor permitiu uma melhora na visualização, uma vez que os intervalos de tempo entre dois frames variam bastante. Com isso, o movimento do cursor foi suavizado, evitando "saltos" no gráfico, que ocorreria caso a exibição fosse feita apenas a cada frame.

Sendo assim, o software se mostrou uma ferramenta útil para a análise visual dos dados, já que este conseguiu correlacionar duas bases de dados (sinal EMG e matriz de dados do mouse), permitindo avaliar não só a atividade de um voluntário ao longo do tempo, mas também permite avaliar a eficácia de todo o sistema, isto é, permite enxergar se os comandos feitos pelo usuário tiveram o efeito esperado, e também, se o que ocorreu em termos de movimentação do *mouse* foi ou não desencadeado por uma contração por parte do voluntário.

6 REFERÊNCIAS

AHSAN, R.; IBRAHIMY, M. I. : > ɳ >ω . v. 33, n. 3, p. 41–45, 2009.

AHSAN, R.; IBRAHIMY, M. I.; KHALIFA, O. O. EMG Signal Classification for Human Computer Interaction: A Review. **EuroJournals**, v. 33, p. 480–501, 2009.

BARRETO, A. B.; SCARGLE, S. D.; ADJOUADI, M. A practical EMG-based human-computer interface for users with motor disatbilities. **Journal of rehabilitation research and development**, v. 37, n. 1, p. 53–63, 2000.

DE LUCA, C. J. ELECTROMAGNETIC. **Electromyography, C J Devices, Medical Webster, John G John, Ed Publisher, Wiley See, Flowmeter**, 2006.

HERMENS, H. J. et al. Development of recommendations for SEMG sensors and sensor placement procedures. **Journal of Electromyography and Kinesiology**, v. 10, n. 5, p. 361–374, 2010.

MOON, I.; LEE, M.; MUN, M. A novel EMG-based human-computer interface for persons with disability. **Proceedings of the IEEE International Conference**, p. 519 – 524., 2004.

NIEDERMEYER, E.; SILVA, F. H. L. D. **Electroencephalography: Basic principles, clinical applications and related fields**. 3. ed. Philadelphia: [s.n.].

PRESMAN, R. S. **Engenharia de Software**. 5. ed. Rio de Janeiro: Mc Graw Hill, 2002.

ROYCE, W. W. Managing the development of large software systems. **IEEE Wescon**, 1970.

SOMMERVILLE, I. **Engenharia de software**. 8. ed. São Paulo: Pearson, 2007.

NBR ISO/IEC 12207 – Tecnologia da Informação – **Processos de Ciclo de Vida**, ABNT 1998, Emenda 1: 2002, Emenda 2: 2004.