



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ROBSON GUIMARÃES DA CRUZ

**COMANDO DE CADEIRA DE RODAS MOTORIZADA POR
FUSÃO DE INFORMAÇÕES DE EYE TRACKER E BCI**

MANAUS-AM

2019

ROBSON GUIMARÃES DA CRUZ

**COMANDO DE CADEIRA DE RODAS MOTORIZADA POR
FUSÃO DE INFORMAÇÕES DE EYE TRACKER E BCI**

Dissertação apresentada ao curso
de Mestrado em Engenharia Elétrica, área
de concentração Controle e Automação de
Sistemas, do Programa de Pós-Graduação
em Engenharia Elétrica da Universidade
Federal do Amazonas.

BANCA EXAMINADORA

Prof. Dr. –Ing. Vicente Ferreira de Lucena Junior
Universidade Federal do Amazonas – UFAM

Prof. Dr. Teodiano Freire Bastos Filho
Universidade Federal do Espírito Santo – UFES

Prof. Dr. Eduardo Lázaro Martins Naves
Universidade Federal do Amazonas – UFU

Ficha Catalográfica

C957c Cruz, Robson Guimarães da
Comando de cadeira de rodas motorizada por fusão de informações de eye tracker e BCI / Robson Guimarães da Cruz. 2019
73 f.: il. color; 31 cm.

Orientador: Vicente Ferreira de Lucena Junior
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Amazonas.

1. rastreamento ocular. 2. interface cérebro-computador. 3. cadeira de rodas motorizada. 4. fusão de dados. 5. healthcare. I. Lucena Junior, Vicente Ferreira de II. Universidade Federal do Amazonas III. Título

ROBSON GUIMARÃES DA CRUZ

COMANDO DE CADEIRA DE RODAS MOTORIZADA POR FUSÃO DE
INFORMAÇÕES DE EYE TRACKER E BCI

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovado em 13 de dezembro de 2019.

BANCA EXAMINADORA



Prof. Dr. Vicente Ferreira de Lucena Junior, Presidente
Universidade Federal do Amazonas



Prof. Dr. Teodiano Freire Bastos Filho, Membro
Universidade Federal do Espírito Santo



Prof. Dr. Eduardo Lázaro Martins Naves, Membro
Universidade Federal de Uberlândia

Agradecimentos

- Primeiramente à minha família e amigos que me apoiam desde que retomei minha carreira acadêmica.
- Ao Professor Dr.-Ing. Vicente Lucena, por seu apoio desde a graduação até essa fase do mestrado, disponibilizando sua *expertise* e os recursos necessários.
- Aos demais membros da banca por se disponibilizarem a acompanhar e avaliar este trabalho. Também pelo apoio no decorrer do convênio UFAM – UFES – UFU para o desenvolvimento das atividades de pesquisa, como ao disponibilizar a cadeira de rodas, a BCI e o *Eye Tracker*, além do suporte para receber autorização para a visita técnica na *Ryerson University* (Toronto – Canadá).
- Agradeço também aos demais professores do PPGEE-UFAM que contribuíram com a minha formação no mestrado, seja técnica, pessoal ou profissional.
- Ao CETELI por proporcionar a infraestrutura e laboratório para o desenvolvimento desta pesquisa.

Também gostaria de agradecer a FAPEAM pelo suporte financeiro para a minha permanência no curso e a oportunidade de visita técnica ao campus da *Ryerson University*, em Toronto. Foi uma grande experiência e contribuiu bastante com minha formação.

“ A tarefa não é tanto ver
aquilo que ninguém viu, mas
pensar o que ninguém ainda
pensou sobre aquilo que todo
mundo vê. ”

(Arthur Schopenhauer)

Resumo

Na literatura hoje estão presentes inúmeras variações de modos de comandar uma cadeira de rodas motorizada (CRM). Algumas soluções prezam pela especialização, outras caminham no sentido de viabilizar versões genéricas que pudessem se tornar populares o suficiente a ponto de ser produzidas em massa. De fato, ainda estamos um pouco longe de chegar a uma solução padrão, pois há uma larga variedade de condições físicas que fazem uma solução ótima para uma pessoa, ser inadequada e traga desconfortos a outra. Essa variedade de necessidades dos usuários é ainda mais complexa de contornar em um país como o nosso, em desenvolvimento, onde cadeiras motorizadas não são tecnologias tão populares e os preços inviabilizam a aquisição por grande parte dos potenciais usuários.

Este trabalho se insere entre as soluções especializadas, propondo uma opção que pode ser utilizada por pessoas com deficiências severas. Com uma interface gráfica intuitiva e feedback visual em cores significantes no cotidiano, é oferecido um sistema que possibilita o comando de uma CRM usando um dispositivo de rastreamento ocular (conhecido em inglês como *Eye Tracker*) como entrada dominante e confirmações pontuais via Interface Cérebro-Computador (ICC, ou do inglês, *Brain-Computer Interface* – BCI), ambos dispositivos comerciais, no qual é possível alcançar confiabilidade próxima de 95%.

Palavras-chave: rastreamento ocular, interface cérebro-computador, cadeira de rodas motorizada, cadeira de rodas elétrica, fusão de dados, *healthcare*, automação.

Abstract

Currently the literature presents a lot of ways to command a motorized wheelchair. Some of them aim to specialization, others prefer working in generic solutions such that become popular and enable the mass production. In fact, we are still a little far from a standard solution because there are a wide variety of physical conditions that some good solutions to one person is unsuited and cause discomfort to another. Those variety of requirements is more complex to overcome in a country like ours in development, where the motorized wheelchair is not a popular technology and its cost turn the acquisition difficult for most of those potential users.

This work is a part of the specialized solutions, which is an option that can be used by people with severe disabilities. With an intuitive graphical interface and meaningful color visual feedback on a daily basis, it is provided system that enables the command of a motorized wheelchair using Eye Tracker as the dominant input and simple confirmations via Brain-Computer Interface (BCI), both commercial devices, in which is possible to achieve the reliability rate close to 95%.

Key words: eye tracking, brain-computer interface, motorized wheelchair electrical wheelchair, data fusion, healthcare, automation.

LISTA DE SIGLAS

AD	Analógico-Digital
ADC	<i>Analog-to-Digital Converter</i>
ALS	<i>Amyotrophic Lateral Sclerosis</i> / Esclerose Lateral Amiotrófica
API	<i>Application Programming Interface</i>
BCI	<i>Brain-Computer Interface</i>
BCM	Broadcom SOC <i>channel</i>
CRM	Cadeira de Rodas Motorizada
CT	Centro de Treinamento
DA	Digital-Analógico
DAC	<i>Digital-to-Analog Converter</i>
EEG	Eletroencefalografia
EMG	Eletromiografia
EOG	Eletrooculografia
ERD	<i>Event-Related Desynchronization</i>
ERS	<i>Event-Related Synchronization</i>
ET	<i>Eye Tracker</i> / <i>Eye Tracking</i>
GUI	<i>Graphical User Interface</i>
HMI	<i>Human-Machine Interface</i>
I2C	<i>Inter-Integrated Circuit</i>
IHC	Interação Humano-Computador
IHM	Interface Humano-Máquina
JSON	<i>JavaScript Object Notation</i>
KF	<i>Kalman Filter</i>

MI	<i>Motor Imagery</i>
MM	Média Móvel (simples)
MME	Média Móvel Exponencial
MMP	Média Móvel Ponderada
MVC	<i>Model-View-Controller</i>
SW	<i>Smart Wheelchair</i>
SSVEP	<i>Steady-State Visually Evoked Potential</i>
UFAM	Universidade Federal do Amazonas
UFES	Universidade Federal do Espírito Santo
UFU	Universidade Federal de Uberlândia

LISTA DE FIGURAS

Figura 1 - <i>Eye Tracker</i> comercial da Tobii Studio (Fonte da imagem do dispositivo: https://help.tobii.com/hc/en-us/articles/212824889-Frequently-asked-questions) e exemplos de aplicações usando ET [10].	19
Figura 2 - BCI comercial Emotiv Epop+ (Fonte: https://www.emotiv.com/epoc/), sinais EEG e estímulos relacionados [11].	21
Figura 3 - Etapas comuns em sistemas comandados por sinais EEG.	21
Figura 4 - Classificação de Whyte e respectivas relações entre as fontes de dados [18].	22
Figura 5 - Tipos de arquitetura centralizada (a), descentralizada (b) e distribuída (c) [17][19].	23
Figura 6 - Modelo básico JDL simplificado (tradução própria) [17].	24
Figura 7 - Modelo de Fusão Omnibus (tradução própria) [19].	25
Figura 8 - Ilustração da distribuição normal $N(\mu, \sigma^2)$.	26
Figura 9 - Diagrama resumido do processo de estimativa pelo KF.	29
Figura 10 - Estrutura geral da proposta de sistema.	35
Figura 11 - Diagrama de interações de componentes do sistema.	36
Figura 12 - Arquitetura do sistema relativa a fusões.	37
Figura 13 - Concepção inicial da GUI e sequência de operação.	38
Figura 14 - Diagrama de estados das interações possíveis até a seleção do comando ‘girar para esquerda’.	39
Figura 15 - Principais elementos do software do sistema na visão do modelo MVC. ...	41
Figura 16 - Diagrama de classes do sistema de software com suas fronteiras.	43
Figura 17 - Interface gráfica do sistema com um indicador de estado para testes.	43
Figura 18 - Máquina de estado do sistema na operação de selecionar comando.	44
Figura 19 - Distribuição normal x precisão na posição do cursor.	45
Figura 20 - Comportamento da GUI e descrição das transições.	46
Figura 21 - Etapas de cálculo da posição do cursor via algoritmo KF.	48
Figura 22 - Fluxograma com alternância de algoritmos para cálculo da posição do cursor.	49
Figura 23 - Arquitetura de fusão atualizada.	50
Figura 24 - Ordenação dos eventos de entrada.	51

Figura 25 – Ilustração do acesso concorrente a recursos.	52
Figura 26 - Passos na preparação da interface com BCI.....	54
Figura 27 - Fluxograma dos modos de treinamento e operação.....	56
Figura 28 - Diagrama de interações entre os elementos de hardware chaves do sistema.	57
Figura 29 - Circuito para integração dos componentes do sistema embarcado à cadeira	58
Figura 30 - Foto de partes montadas no sistema embarcado (1: Dois conversores D/A (Digital-Analógico); 2: Conversor A/D (Analógico/Digital); 3: Chave de três estados; 4: Raspberry)	59
Figura 31 - Fluxo dos dados entre o socket e as portas I2C.....	60
Figura 32 - Fonte <i>step down</i> LM2596.....	60
Figura 33 - Comunicação entre conversores e Rasperry via I2C	61
Figura 34 - Diagrama de classes do sistema embarcado da CRM.....	61
Figura 35 - Cadeira de rodas motorizada com sistema embarcado	62
Figura 36 - Percurso realizado durante experimentos	64

LISTA DE TABELAS

Tabela 1 - CRM comerciais versões Smart e características [8][9].....	16
Tabela 2 - Equações do Filtro de Kalman.	28
Tabela 3 - Equações do KF para uma dimensão.	28
Tabela 4 - Aplicações diversas envolvendo o comando de CRM com ET ou BCI, com entradas uni ou multimodais.....	30
Tabela 5 - Estados de interações com o sistema.....	38
Tabela 6 - Confiabilidade do evento B versus confiabilidade geral.	51
Tabela 7 - Dados do BCI disponíveis via API Emotiv	55
Tabela 8 - Equivalência entre comandos recebidos, escala dos conversores e sinais de controle da cadeira.	56
Tabela 9 - Resumo dos dados obtidos na experimentação.....	64

Sumário

Capítulo 1 - Introdução	14
1.1 Motivação	16
1.2 Objetivos	17
1.2.1 Objetivo Geral.....	17
1.2.2 Objetivos Específicos.....	17
Capítulo 2 – Fundamentação Teórica	19
2.1 Conceitos e Tecnologias	19
2.1.1 <i>Eye Tracking</i>	19
2.1.2 <i>Brain-Computer Interface</i>	20
2.1.3 Fusão de Dados ou Informações.....	22
2.1.4 Distribuição Normal.....	25
2.1.5 Média Móvel.....	27
2.1.6 Filtro de Kalman	28
2.2 Trabalhos Relacionados.....	29
2.2.1 Modos de atuação.....	31
2.2.2 Interfaces de entrada e relação entre sinais	32
Capítulo 3 – Concepção Inicial da Solução	35
Capítulo 4 – Desenvolvimento	41
4.1 Software.....	41
4.2 Integração e Adaptações na CRM.....	56
Capítulo 5 – Resultados experimentais	63
Capítulo 6 – Conclusão	66
Referências Bibliográficas	69
Publicações	73

Capítulo 1 - Introdução

As cadeiras de rodas são dispositivos que vêm sendo empregados há várias décadas na sociedade para ajudar pessoas com limitações físicas. A primeira versão motorizada (CRM – cadeira de rodas motorizada) remonta a época pós Segunda Guerra Mundial e foi desenvolvida no Conselho Nacional de Pesquisas do Canadá, enquanto que em 1956 começou a produção em massa pelas empresas *Everest & Jennings* e *American Wheelchair Company* [1]. Entretanto, a CRM convencional, controlada por *joystick*, não é capaz de atender as necessidades de pessoas com deficiências físicas mais severas como tetraplegia, esclerose múltipla ou esclerose lateral amiotrófica (ALS, na sigla em inglês), permitindo-lhes cumprir sua rotina diária sem assistência de terceiros. Para essas pessoas, são feitos arranjos mais sofisticados, o que pode incluir desde métodos de entrada alternativos como rastreamento ocular, interface cérebro-computador (BCI, na sigla em inglês), eletromiografia (EMG), voz, movimentos da cabeça, da língua, multimodais, além de sistemas de navegação assistida, semi ou completamente autônoma [1][2][3].

Segundo dados obtidos pelo Instituto Brasileiro de Geografia e Estatística (IBGE), no Censo de 2010, é estimado em cerca de 7% o percentual de pessoas com algum grau de deficiência física, das quais 1,62% não conseguem se locomover [4]. De acordo com o Relatório Mundial sobre Deficiências de 2011, elaborado pela Organização Mundial da Saúde (OMS), das cerca de 75 milhões de pessoas no mundo que precisam de cadeira de rodas para se locomover, apenas de 5 a 15% conseguem obtê-la [5].

CRM mais sofisticadas, chamadas de *Smart Wheelchair*, recebem hoje muita atenção no meio científico, mas o preço é um impeditivo à sua adoção principalmente em países em desenvolvimento como no caso do Brasil, onde mesmo versões mais simples podem chegar a R\$ 7.000,00 reais, segundo informação da agência de fomento FAPESP [6]. Uma CRM adaptada para tetraplégicos custa em média R\$ 2.450,00 reais, porém, o SUS oferece pagar a fornecedores apenas R\$ 1.170,00 reais, segundo a Revista Nacional de Reabilitação [7]. Como a maior parte da população brasileira depende do sistema público de saúde, soluções menos generalistas e de menor custo, podem favorecer a popularização dessa tecnologia.

Esta dissertação de mestrado se propõe a desenvolver uma interface de comando alternativa para uma cadeira de rodas motorizada, onde a intenção do usuário é estabelecida por meio de fusão de informações de dispositivos *Eye Tracker* (ET) e *Brain-*

Computer Interface (BCI) comerciais. Por meio do ET, o usuário pode mover um cursor na interface gráfica (no inglês, *Graphical Unit Interface* - GUI) e selecionar uma entre as opções de controle disponíveis para a cadeira. Para minimizar as chances de comandos não-intencionais, a GUI produz estímulos visuais em alguns dos passos do processo de seleção, os quais devem ser percebidos nos sinais EEG capturados pela BCI. A fusão entre a posição do cursor, controlado por ET, e o padrão classificado dos sinais EEG, estima a ação selecionada pelo usuário, a qual é enviada à cadeira para reproduzir o movimento. Espera-se obter um sistema de fácil adaptação para o usuário, permitindo-o locomover a CRM local de forma segura, em interações simples e sem exigir grande esforço. Ou seja, que a interface com ET e BCI combinados, permita a realização dos percursos da cadeira de rodas por a pessoas impossibilitadas de usar o *joystick*.

Ressalta-se que este trabalho apresenta uma proposta alternativa de processamento de comando de uma cadeira de rodas motorizada para controle local. Entre as possíveis contribuições, estão:

- Aplicação de ET e BCI, combinados por fusão em nível de decisão, para comando de CRM local;
- Utilização de dispositivos de entrada e comerciais;
- Abordagem para tratar o “*Midas Touch*” em dois níveis: desbloqueio e confirmação de ação, ambos por intermédio da GUI.

Ao final do desenvolvimento, foram medidos e analisados:

- a) Tempo de execução de percursos com a interface de entrada proposta a ser comparado com *joystick*;
- b) Comparativo de percursos ao usar *joystick* ou interface de entrada proposta.

Esses dados servem para avaliar a facilidade de uso do sistema, tanto em relação ao esforço demandado do usuário quanto à manobrabilidade alcançada pela combinação das interfaces de entrada. A comparação com o *joystick* é comumente usada na literatura como referência e permite uma noção quantitativa mais clara do que o sistema desenvolvido proporciona ao usuário.

1.1 Motivação

Os dados estatísticos do IBGE afirmam que cerca de 1,62% da população brasileira possui incapacidade de locomoção, ou seja, cerca de 3,4 milhões de pessoas necessitam de cadeira de rodas, da configuração mais simples à mais sofisticada [4]. Essa demanda tende a aumentar se considerarmos o fenômeno do contínuo aumento da proporção de idosos na população de todo o mundo. Ao lidar com um largo espectro de necessidades, o meio científico tem se dirigido a soluções mais sofisticadas chamadas convencionalmente de *Smart Wheelchairs* (SW) [1], para atender o máximo de condições e restrições com a mesma cadeira. Porém, baseado no preço atual das cadeiras motorizadas simples, é natural que essas versões mais complexas possuam preços bem mais elevados, dificultando a transição dessas aplicações da academia para a produção industrial e à população, o que é evidenciado pelo fato de haver relativamente poucas aplicações comerciais. Além disso, algumas cadeiras vendidas como *Smart* não trazem tantos recursos quanto o apresentado pela literatura como SW.

A Tabela 1 contém diversos dispositivos relacionados a SW e suas características, onde maioria inclui a CRM.

Tabela 1 - CRM comerciais versões *Smart* e características [8][9].

Distribuidor	Modelo	Preço	Sensores	Modos de operação	EPW Incluída?
Smart Rehab, Ltd.	Smart Wheelchair	US\$ 14200	Sonar, Detector de colisão, detector de linha	Tratamento de colisão, seguidor linha	Sim
Smart Rehab, Ltd.	Smart Box	US\$ 5000	Detector de colisão, detector de linha	Tratamento de colisão, seguidor linha	Não
Applied AI Systems, Inc.	TAO-7	US\$ 37400	Sonar, sensores de distância, visão computacional	Passeio aleatório, controle compartilhado, navegação autônoma	Sim
Nurion Industries	Wheelchair Pathfinder	US\$ 4500	Sonar, sensores de distância	Vibração quando obstáculo ou descida brusca são detectadas	Não
ActivMedia	Robotic Chariot	US\$ 36490	Sensores de distância, deslocamento e colisão; GPS, visão computacional	Passeio aleatório, controle compartilhado, navegação autônoma	Sim
Karman Healthcare	Karman	US\$ 8400	(Não mencionado)	Permite deslocamento do usuário em pé.	Sim
Wheelchair 88	Wheelchair 88	US\$ 12500	Suspensão automática	Permite deslocamento do usuário em pé; Suavizador de trajetórias; Sobe e desce	Sim

Para esta dissertação serão empregados CRM, *Eye Tracker* e BCI comerciais já disponíveis que são integrados em uma solução com custo mais conveniente à realidade brasileira, mas com desempenho apropriado tanto em relação à segurança quanto na

facilidade de operação. Esses fatores impactam diretamente na escolha do sinal EEG e técnica de obtenção do mesmo, pois algumas abordagens provocam maior fadiga do usuário. Também é tratado o problema de comandos não intencionais, o qual é mais recorrente nas interfaces de entrada baseada em ET. Cada forma de entrada tem desvantagens associadas, mas que podem ser superadas pelo uso combinado e um fluxo adequado no processo de seleção de comandos.

A necessidade por meios de entrada alternativos conduz à demanda de que os sistemas de treinamentos disponibilizem essas novas modalidades de comando para novos usuários, permitindo-lhes iniciar a adaptação para uso da CRM e da modalidade de comando adequada. Por outro lado, os Centros de Treinamento (CT) podem permitir novos usuários de cadeiras de rodas praticarem a locomoção, de forma segura, seja em ambiente virtual ou real, para facilitar e acelerar a adaptação aos recursos disponíveis como sensores e interface humano-máquina (IHM ou HMI – *Human-Machine Interface*, do inglês). A interface multimodal aqui apresentada poderá integrada, em trabalhos futuros, para o uso tanto em CRM local quanto remota, permitindo conduzir, à distância, cadeiras presentes em CT de universidades parceiras como a Universidade Federal do Espírito Santo (UFES) ou a Universidade Federal de Uberlândia (UFU).

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver uma interface de entrada multimodal, com comandos definidos por fusão de informações de *Eye Tracker* e BCI comerciais, que possibilite ao usuário controlar uma cadeira de rodas local com segurança e sem grande esforço mental.

1.2.2 Objetivos Específicos

- Construir programa para obtenção dos dados do *Eye Tracker* e BCI.
- Identificar e implementar uma técnica que seja apropriada para a fusão de informações do *Eye Tracker* e BCI.

- Desenvolver programa de gerenciamento da IHM adequada ao processo de seleção de comandos e interface de entrada combinada.
- Integrar a IHM à cadeira de rodas local por meio de um sistema computacional que possibilite a realização da prova de conceito.

Nos capítulos seguintes, é apresentado o processo de desenvolvimento e pesquisa objetos desta dissertação. No capítulo 2, é contido a base teórica e tecnológica sobre as quais é construída a solução. O capítulo 3 compõem-se da concepção inicial, seguido pelo capítulo 4, que descreve o desenvolvimento propriamente dito, tanto em relação ao *software* quanto à adaptações no *hardware*. O capítulo 5 apresenta os resultados obtidos a partir dos experimentos realizados para a validação do conceito, enquanto o capítulo 6 elenca as conclusões obtidas e sugestões de trabalhos futuros.

Capítulo 2 – Fundamentação Teórica

2.1 Conceitos e Tecnologias

Diversas soluções têm sido propostas para propiciar mais independência para usuários de cadeiras de rodas motorizadas. Casos de deficiências mais severas demandam interfaces de entrada mais sofisticadas, como *Eye Tracking* e BCI, e têm se tornado mais populares com a melhoria dos sistemas computacionais, técnicas de processamento e menor custo dos dispositivos. Esses avanços permitiram sistemas mais confiáveis tanto em relação ao tempo de processamento quanto à acurácia, resultando em operações cada vez mais seguras.

2.1.1 *Eye Tracking*

Os métodos de entrada por rastreamento ocular utilizam o movimento dos olhos ou a posição do olhar, o que pode ser feito por imagem (*Computer-vision-based Eye Tracking* ou simplesmente *Eye Tracking*) ou a partir de sinais de sensores na região ocular (eletrooculografia ou EOG). Os dados obtidos do movimento ou posição dos olhos são associados às ações do usuário disponíveis. Entre possíveis aplicações usando dispositivos de entrada por rastreamento ocular, podem ser encontradas interfaces de comunicação, projetos relacionados a *e-learning*, vigilância, assistência a motoristas (como um indicador de atenção) e tecnologias assistivas [10]. Nessa última, permite pessoas com deficiência severa controlar sistemas que lhes possibilitem maior autonomia em atividades cotidianas. Outras aplicações podem ser observadas na Figura 1.



Linha de Pesquisa	Exemplos de Aplicação
Mobilidade e acessibilidade em Healthcare	IHM em cadeira de rodas e comunicação de pessoas com deficiência
Sistemas de assistência a motoristas	Detecção de fadiga
Autenticação de usuário em sistemas	Autenticação biométrica associada ao reconhecimento pela íris
E-Learning	Percepção de estado emocional, dificuldade de aprendizado e fadiga

Figura 1 - *Eye Tracker* comercial da Tobii Studio (Fonte da imagem do dispositivo: <https://help.tobii.com/hc/en-us/articles/212824889-Frequently-asked-questions>) e exemplos de aplicações usando ET [10].

O controle de CRM baseado em ET frequentemente depende da definição do parâmetro *dwell time*, o qual representa o tempo de permanência do olhar do usuário sobre uma certa opção para que esta seja selecionada. Mas o ajuste desse tempo, para mais ou para menos, pode originar mais comandos não-intencionais ou provocar maior desconforto ao usuário [11]. Para lidar com o problema de comandos não-intencionais, uma opção é usar um botão para o usuário pressionar quando quiser interagir com o sistema. Ele pode ser físico [11], ou virtual, onde uma região do campo visual é definida como um botão de controle, tal como feito por Fedorova [12].

O dispositivo *Eye Tracker* comercial, disponível para esta pesquisa, é fabricado e vendido pela empresa Tobii Studio, possuindo diversas opções de controle e alcançando resultados confiáveis [10]. O fabricante fornece uma API (*Application Programming Interface*) por meio da qual é possível capturar do usuário diversas características, tais quais posições dos olhos, cabeça e olhar, além de permitir realizar a calibração e alterar configurações do dispositivo. A abordagem a ser utilizada é mover o cursor da GUI usando o olhar e fornecer, na tela, o acesso às opções de comando da CRM.

2.1.2 Brain-Computer Interface

A captura de sinais de ondas cerebrais é utilizada há várias décadas na medicina. Porém, nos últimos anos, surgiram diversas aplicações em outras áreas, como a engenharia, impulsionadas pelo desenvolvimento de dispositivos chamados BCI. Diferente de alguns equipamentos médicos comuns, eles são não invasivos e podem ser posicionados sobre o couro cabeludo sem exigir implante de eletrodos. Além disso, o custo deles é muito menor que os equipamentos convencionais de eletroencefalografia hospitalar. BCIs se referem a sistemas que capturam sinais gerados pela atividade cerebral (EEG), transmitindo e convertendo-os em intenções do usuário, os quais são previamente associadas a certos padrões dos sinais. Pode-se encontrar aplicações onde os sinais EEG são originados por estímulo externo (exógenas) ou provocados pelo próprio usuário (endógenas). A forma como são extraídos os sinais EEG dependem do padrão de sinal a ser empregado na aplicação, o que está relacionado ao tipo de estímulo gerador. Dentre os mais conhecidos e utilizados, apresentados na Figura 2, estão o P300, o SSVEP (*Steady-State Visually Evoked Potential*) e o ERD/S (*Event-Related Desynchronization/Synchronization*). As estratégias de controle de uma CRM baseadas no

o sinal P300 requer menor esforço de concentração e atenção, apresenta uma acurácia relativamente alta e um tempo de processamento [13][14]. Porém, a necessidade de repetição do estímulo para melhor elucidar o sinal EEG pode tornar mais lento o tempo de resposta e favorecer a aplicação do SSVEP. Tanto o P300 quanto o SSVEP são potenciais que dependem de um estímulo visual externo.

	Sinal EEG	Estímulo
	P300	Evento visual; pico de EEG tem aparecimento 300 ms após o início do estímulo
	SSVEP	Evento visual; associado à frequência do estímulo
	ERD / ERS	Tarefa mental: imaginar movimento de membros, palavra, cálculo, entre outros

Figura 2 - BCI comercial Emotiv Epoc+ (Fonte: <https://www.emotiv.com/epoc/>), sinais EEG e estímulos relacionados [11].

Em geral, sistemas que aplicam BCIs possuem processos para aquisição dos sinais, extração de características, classificação e tradução (em intenções do usuário). É comum também haver uma etapa de pré-processamento após a aquisição do sinal de EEG para melhorar a qualidade dos sinais obtidos.

A aquisição ocorre por meio de eletrodos colocados em certas posições do escalpo, depois os sinais são digitalizados e recebidos por alguma unidade computacional, onde são pré-processados para melhorar a qualidade do sinal, filtrando ruídos e destacando os padrões desejáveis. Sobre esses dados são aplicados algoritmos de extração de características para avaliar a presença de padrões relacionados a algum estímulo, e esses padrões são vinculados a intenções do usuário para acionar, por exemplo, uma CRM [13][15].

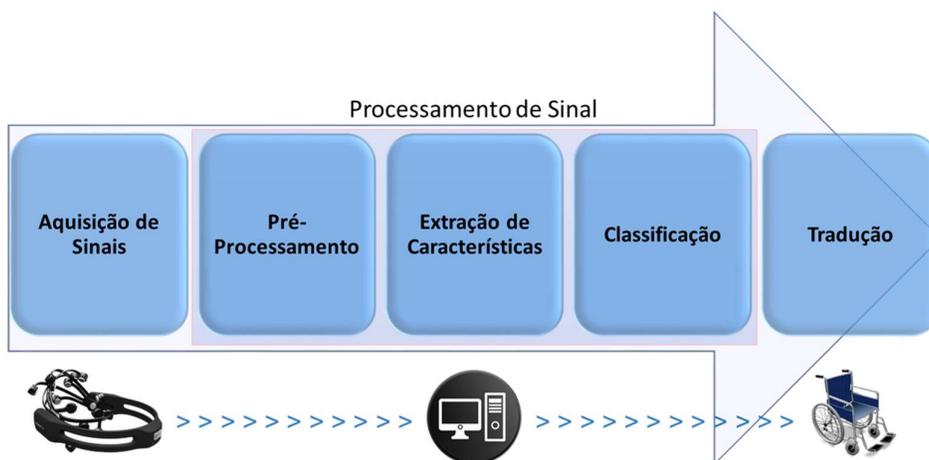


Figura 3 - Etapas comuns em sistemas comandados por sinais EEG.

O dispositivo BCI comercial aplicado nesta pesquisa é o Emotiv Epoc+, fabricado e vendido pela Emotiv, que possui API para que sejam desenvolvidas aplicações em

linguagens como C++, C#, Java e Python. Ele possui 14 eletrodos que são distribuídos no escalpo em posições especificadas pelo padrão internacional 10-20, e se conecta ao computador por *Bluetooth* ou USB, alcançando entre 6 horas e 12 horas de autonomia, dependendo da interface de comunicação utilizada. O programa a ser desenvolvido deve usar as funções disponibilizadas pela API para adquirir os sinais e decodificá-los em ações, caso seja identificado algum dos padrões previamente treinados. A etapa de treinamento também possui suporte via API e é possível salvar até quatro padrões por perfil de usuário.

2.1.3 Fusão de Dados ou Informações

Fusão de dados é a aplicação de técnicas para associar, correlacionar ou combinar dados, direta ou indiretamente, oriundos de sensores ou base de dados, em uma estrutura comum, ainda que tenham origens diferentes. O objetivo da fusão é melhorar a qualidade dos dados e, conseqüentemente, a acurácia e os processos de tomada de decisões associados a eles [16][17].

Apesar de relacionados, dados e informações representam elementos distintos. Enquanto os dados são vistos como uma simbologia para representar um estímulo, sinal ou fenômeno, a informação agrega alguma inferência sobre o fenômeno a partir de dados ou outras informações [17]. Por exemplo, um sensor de temperatura, ao fornecer valores numéricos relativos ao fenômeno, posicionaria as medições em uma escala; por outro lado uma possível inferência sobre esses dados poderia ser, por exemplo, de a temperatura estar acima ou abaixo de alguma referência definida.

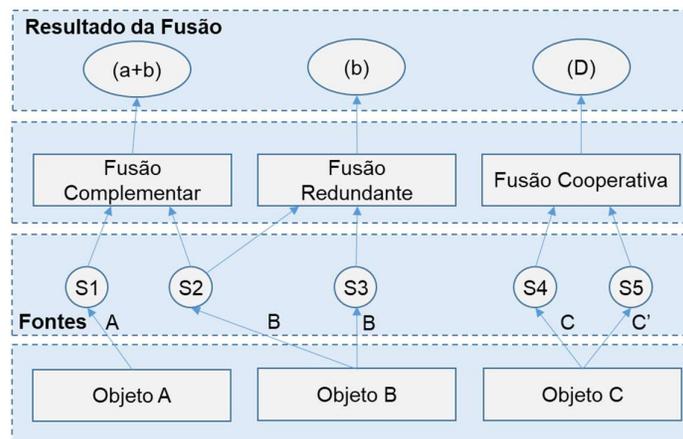


Figura 4 - Classificação de Whyte e respectivas relações entre as fontes de dados [18].

Diversas arquiteturas, também referenciadas como modelos ou frameworks, têm sido propostas para a fusão de dados. E, com isso, são observadas formas diferentes de classificar um sistema de fusão. Conforme citado por Castanedo [18], Whyte classifica a fusão como complementar, redundante ou cooperativa, e estas refletem possíveis relações entre as fontes de dados, conforme ilustrado na Figura 4.

Na fusão complementar, dados ou informações de fontes diferentes participam como amostras em pontos de vista variados de um mesmo objeto ou cena. Tais amostras são combinadas para obtenção de uma informação mais completa da fonte. Na fusão redundante, as fontes são as mesmas, e podem ser utilizados sensores similares monitorando o mesmo alvo a partir de posições diferentes. Os dados são tratados para gerar outros do mesmo tipo com o valor mais acurado. O processo de fusão cooperativa recebe dados de diferentes modalidades ou fontes e apresenta como resultado dados ou informações mais complexas que as originais [18][19].

Quanto à disposição estrutural dos elementos, as arquiteturas de sistemas de fusão podem ser centralizadas, distribuídas ou híbridas. Nas arquiteturas centralizadas, as fontes de dados são reunidas e combinadas na etapa responsável pela fusão. As descentralizadas possuem módulos de fusão específicos para cada sensor ou fonte de dados. Ao unir as características das duas primeiras, é obtida a arquitetura híbrida, a qual propicia a fusão entre dados já fusionados. A Figura 5 ilustra cada um desses tipos.

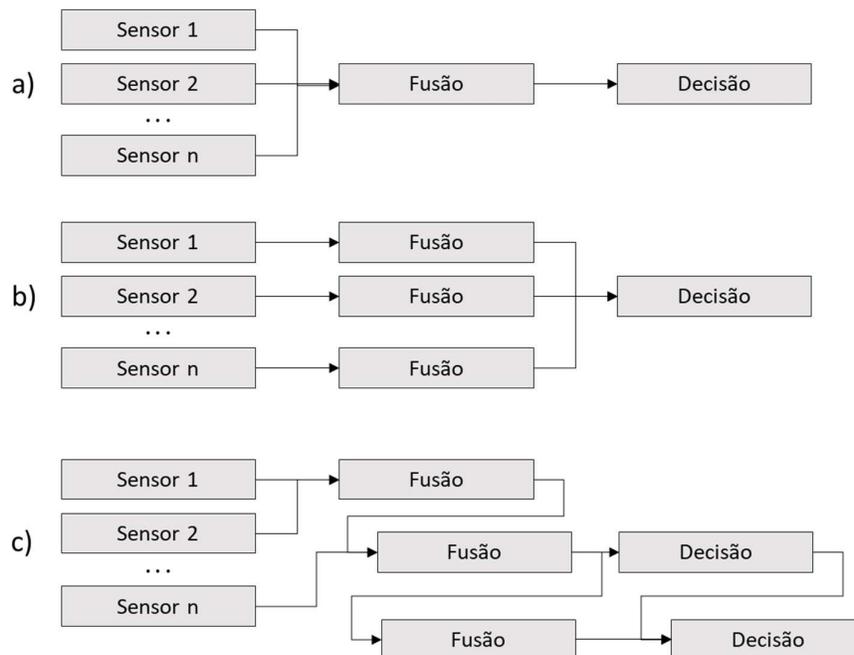


Figura 5 - Tipos de arquitetura centralizada (a), descentralizada (b) e distribuída (c) [17][19].

O departamento de defesa americano, por meio da *Joint Working Group Directors of Laboratories* (JDL), que inclusive foi responsável pela definição mais aceita de fusão de dados, em 1986, propôs uma arquitetura hierárquica mostrada na Figura 6. O modelo JDL possui 5 níveis hierárquicos, onde:

- Nível 0: captura e representação simbólica do sinal ou fenômeno;
- Nível 1: oferece a combinação dos dados recebidos dos sensores para identificar um objeto de interesse, seja um atributo específico, estado ou entidade;
- Nível 2: proporciona uma visão relacional de entidades e eventos;
- Nível 3: pela avaliação do estado, nesse nível são feitas deduções a respeito de estados futuros, inclusive vantagens e desvantagens de ações relacionadas;
- Nível 4: relaciona-se a identificação de possíveis melhorias e otimizações do processo, especialmente para sistemas de tempo real.

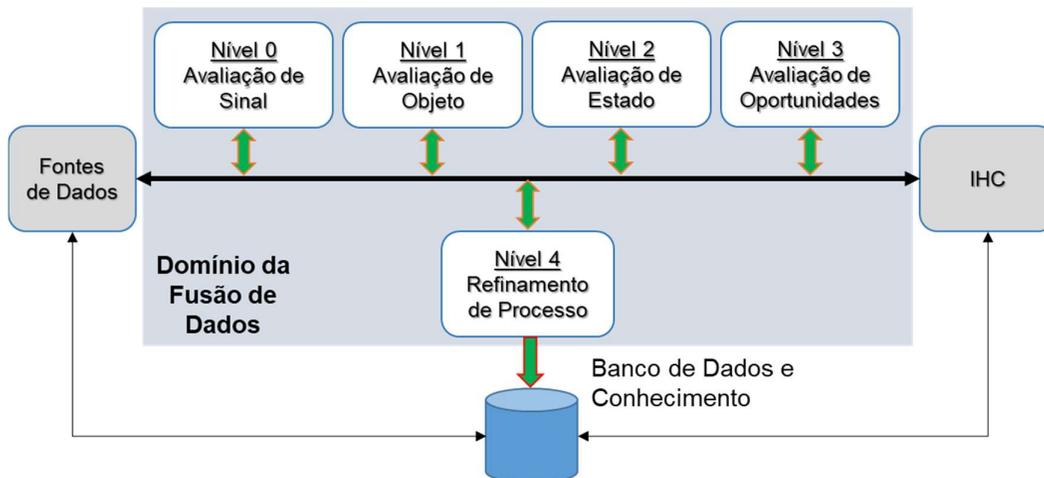


Figura 6 - Modelo básico JDL simplificado (tradução própria) [17].

Nesta arquitetura, destaca-se ainda o suporte à base de dados ou conhecimento que orienta os processos de tomada de decisão e melhoria, além da possibilidade de operar como um provedor dos dados de entrada. Porém, os sensores são os elementos mais encontrados no fornecimento de dados.

Uma outra arquitetura importante, descrita por Bedworth e O'Brien, é chamada de Omnibus [19] e pode ser observada na Figura 7. Trata-se de uma arquitetura iterativa e realimentada que relaciona as etapas do processo às atividades de observar, orientar, decidir e agir.

O ciclo começa com a captura e processamento de sinais ou dados, dos quais são extraídas informações que orientam o processo decisório. A tomada de decisão,

considerado o contexto, provoca uma ação de controle para executar alguma operação. A mudança no estado do ambiente de operação poderá ser percebida pelos sensores e influenciar estados e tarefas futuras [19].

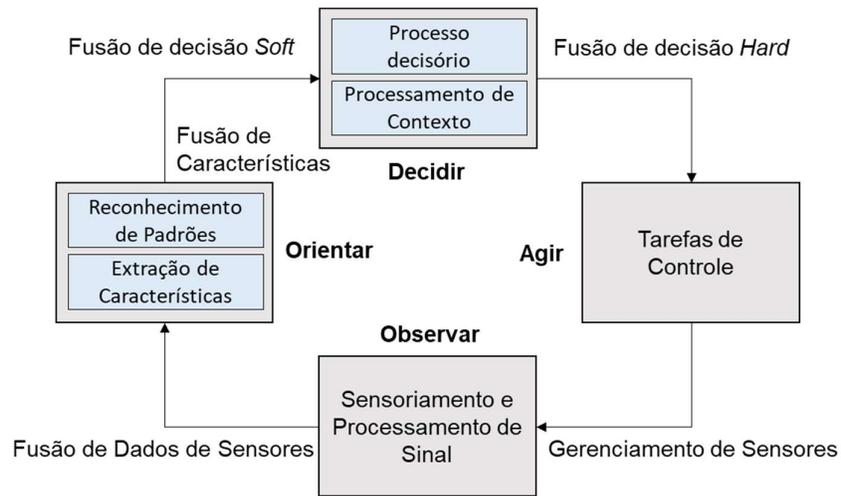


Figura 7 - Modelo de Fusão Omnibus (tradução própria) [19].

Neste trabalho, as APIs do BCI e *eye tracker* associam os dados capturados para melhorar a precisão. No processo de treinamento de sinais do BCI, por exemplo, são capturadas amostras dos 14 eletrodos, que são combinadas para gerar padrões de referência para uso durante a operação. O dispositivo de rastreamento ocular possibilita obter a posição do olhar, mas o cálculo pode usar outras características obtidas pelas câmeras do dispositivo, como a posição da cabeça e do ombro.

A utilização do ET e BCI apresentam separadamente limitações que são superadas pela atuação conjunta das interfaces, tal como demonstrado na pesquisa de Hyder [20]. Embora o primeiro seja mais fácil de usar que o segundo [2] e tenha potencial para mapear muitos comandos [11], está mais sujeito a entradas não-intencionais (conhecido na literatura como “Toque de Midas”) [14]. No decorrer deste trabalho são exploradas as características mais promissoras de cada interface e, por meio de fusão, estas são combinadas para oferecer uma experiência melhor ao usuário e uma alternativa ao uso do *joystick*.

2.1.4 Distribuição Normal

A distribuição normal é uma das distribuições de probabilidade utilizadas na modelagem, análise e inferência estatística de variáveis aleatórias contínuas. Em notação

formal, é geralmente apresentada como $N\sim(\mu, \sigma^2)$, que denota uma distribuição normal parametrizada com média μ e variância σ^2 [21].

Para determinar a probabilidade de uma variável aleatória X adquirir valores dentro de um intervalo $[x_1, x_2]$, pode ser aplicada a expressão [21]:

$$P(x_1 \leq x \leq x_2) = \int_{x_1}^{x_2} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(x-\mu)^2/\sigma^2} dx, \text{ onde } \begin{cases} \mu \rightarrow \text{m\u00e9dia} \\ \sigma \rightarrow \text{desvio padr\u00e3o} \end{cases}$$

Comumente, s\u00e3o adotadas tabelas que permitem mais facilmente obter essa probabilidade ou ainda s\u00e3o usados resultados emp\u00edricos. Em um conjunto de dados distribu\u00eddos normalmente [21]:

- 68% dos dados est\u00e3o no intervalo $[\mu - \sigma, \mu + \sigma]$;
- Cerca de 95% dos dados se concentram entre $\mu - 2\sigma$ e $\mu + 2\sigma$;
- Aproximadamente 99,7% dos dados se encontram em $[\mu - 3\sigma, \mu + 3\sigma]$;

A Figura 8 apresenta o gr\u00e1fico da distribui\u00e7\u00e3o normal com os resultados emp\u00edricos mencionados acima. Quanto menor a vari\u00e2ncia, maior a concentra\u00e7\u00e3o de valores em um intervalo menor e, portanto, mais estreito \u00e9 o gr\u00e1fico. Al\u00e9m disso, o gr\u00e1fico \u00e9 sim\u00e9trico em rela\u00e7\u00e3o \u00e0 m\u00e9dia.

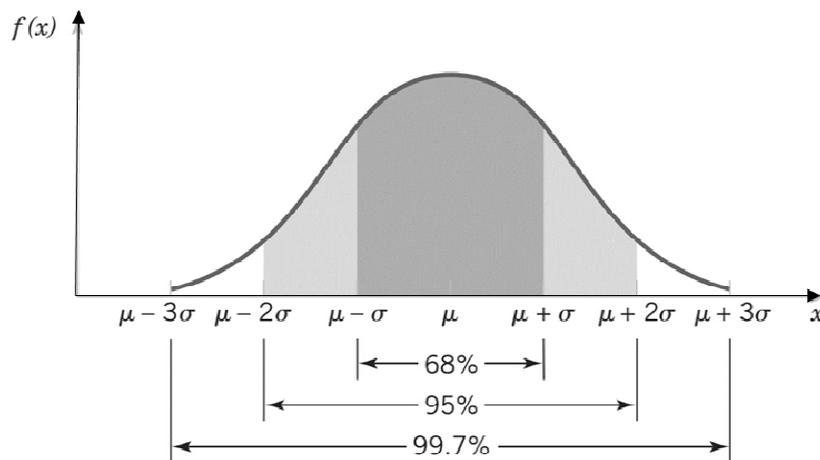


Figura 8 - Ilustra\u00e7\u00e3o da distribui\u00e7\u00e3o normal $N\sim(\mu, \sigma^2)$.

Por meio dessas probabilidades, obtidas tanto por tabela quanto por c\u00e1lculo, s\u00e3o estabelecidos intervalos de confian\u00e7a dos valores conforme o grau de precis\u00e3o necess\u00e1rio na aplica\u00e7\u00e3o.

2.1.5 Média Móvel

Média móvel é uma nomenclatura aplicada quando são calculadas médias em diferentes intervalos de dados amostrais, ou seja, como se o intervalo fosse deslocado ao longo da série de medidas sobre as quais se deseja operar. É um tipo de medida de tendência utilizado no mercado financeiro e no processamento de sinais, sendo categorizado como um filtro passa-baixas. É usada para filtrar variações de curto prazo e facilitar a visualização na tendência de alcance temporal maior; ou ainda, suavizar as variações dos dados [22].

Os tipos mais comuns são: média móvel simples (MM), média móvel exponencial (MME) e média móvel ponderada (MMP). A MM é calculada como uma média aritmética, onde todos os valores da amostra possuem o mesmo peso sobre a média. Já a MMP é similar à média ponderada convencional, atribuindo pesos a alguns valores segundo a necessidade. A MME possui uma expressão que favorece valores mais recentes atribuindo um peso maior a eles.

$$MM_{n+1} = \frac{x_1 + x_2 + \dots + x_n}{n} + \frac{x_{n+1}}{n} - \frac{x_1}{n}$$

$$MMP_{n+1} = \frac{k_1x_1 + k_2x_2 + \dots + k_nx_n}{n} + \frac{k_{n+1}x_{n+1}}{n} - \frac{k_1x_1}{n}$$

$$MME_n = \left[V \cdot \left(\frac{s}{1+d} \right) \right] + MME_{n-1} \cdot \left[1 - \left(\frac{s}{1+d} \right) \right]$$

onde:

- x_1, x_2, \dots, x_n : amostras
- k_1, k_2, \dots, k_n : ponderações das amostras
- V : parâmetro de ponderação dos valores atuais
- s : fator de suavização
- d : número de amostras
- MME_n : estimativa atual
- MME_{n-1} : estimativa anterior

A média móvel simples foi aplicada combinada às probabilidades segundo parâmetros da distribuição normal. Assim, a posição do cursor da IHM foi estimada de acordo com a precisão necessária.

2.1.6 Filtro de Kalman

O filtro de Kalman ou Kalman Filter (KF) é uma formulação relacionada a sistemas dinâmicos que permite realizar filtragem de sinais a partir de soluções numéricas recursivas, e pode ser aplicado tanto a modelos estacionários quanto não estacionários. O processamento da solução é computado a partir de cada nova entrada de dados e o último estado estimado, portanto exigindo apenas o armazenamento do último estado ao invés de todo o histórico de entradas [23]. Para a filtragem em uma dimensão, são usadas as seguintes equações:

Tabela 2 - Equações do Filtro de Kalman.

Equação	Descrição
$\hat{x}_n = \hat{x}_{n-1} + G_n \cdot (y_n - H_n \cdot \hat{x}_{n-1})$	Equação para cálculo de cada iteração
$\hat{x}_{n-1} = F_{n,n-1} \cdot \hat{x}_{n-1,n-1}$	Equação de transição para prever próximo estado
$G_n = P_{n-1} \cdot H_n^T [H_n \cdot P_{n-1} \cdot H_n^T + R_n]^{-1}$	Ganho do filtro de Kalman
$P_n = (I - G_n \cdot H_n) \cdot P_{n-1}$	Equação para correção de erro na covariância
$P_{n-1} = F_{n,n-1} \cdot P_{n-1,n-1} \cdot F_{n,n-1}^T + Q_{n-1}$	Equação para extrapolação de covariância

Ao realizar manipulações algébricas e considerar a taxa de variação do sistema dinâmico constante, pode-se chegar as seguintes equações para o cálculo em uma dimensão:

Tabela 3 - Equações do KF para uma dimensão.

Equações
$\hat{x}_n = \hat{x}_{n-1} + G_n \cdot (y_n - \hat{x}_{n-1})$
$\hat{x}_n = \hat{x}_{n-1,n-1} \text{ (constante)}$
$G_n = \frac{p_{n-1}}{p_{n-1} + r_n}$
$p_n = (1 - G_n) \cdot p_{n-1}$
$p_{n-1} = p_{n-1,n-1} + q_n$

As iterações que ocorrem durante o processo de filtragem consideram os erros de estimação por meio da covariância, a qual também é corrigida. Dessa forma, o filtro vai reduzindo a influência dos ruídos e converge, a cada iteração, para valores

estatisticamente mais confiáveis. A Figura 9 apresenta um fluxo simplificado do processo.

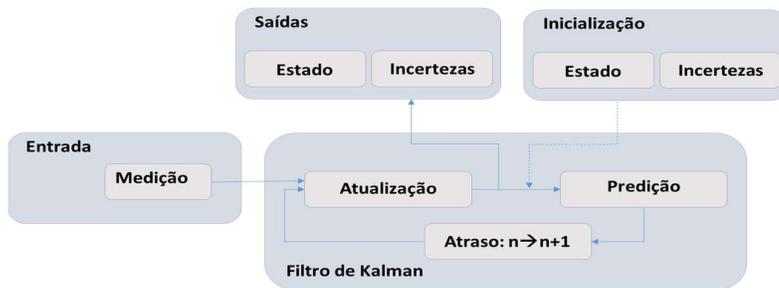


Figura 9 - Diagrama resumido do processo de estimativa pelo KF

O filtro de Kalman é aplicado sobre as amostras do *eye tracker* com a consideração de que, a cada janela de amostragem, temos um sistema constante com uma saída a ser estimada pelas amostras obtidas. Vale ressaltar que com a aplicação do KF, além da redução na influência dos ruídos, foi alcançado um tempo de resposta adequado à atuação em tempo real do usuário com o cursor do sistema.

2.2 Trabalhos Relacionados

A literatura apresenta aplicações de CRM onde o *eye tracking* e o BCI atuam como métodos de entrada independentes, combinados entre si ou com outros. Como interfaces independentes, podem ser usadas como única interface de entrada no sistema (alternadamente) ou como uma das opções disponibilizadas ao mesmo tempo, caso em que o método de entrada é chamado de multimodal [1][2][3].

As interfaces de entrada da cadeira de rodas podem acessar comandos de alto ou baixo nível. Nas aplicações com navegação em alto nível, a locomoção da cadeira ocorre autonomamente ao selecionar algum ponto de destino no ambiente de operação, ou seja, a cadeira possui um sistema computacional e sensorial capaz de estimar e verificar rotas seguras até o local escolhido pelo usuário. A navegação em baixo nível ocorre pela seleção de comandos simples como ir para frente, para trás ou girar. Ambos os casos podem ter alguma assistência, por exemplo, de um controlador que auxilie na detecção ou desvio de obstáculos, passagem por portas ou ainda na suavização de trajetória [1][3][13]. Algumas aplicações com algumas dessas características são apresentadas na

Tabela 4.

Tabela 4 - Aplicações diversas envolvendo o comando de CRM com ET ou BCI, com entradas uni ou multimodais.

Ref	Tipo de Publicação, Ano	Instituição	Interface de Entrada Alternativa	Sinal	Relação entre as entradas	Modo de Atuação	Comandos
[2]	Journal, 2017	De Gruyter	Teclado físico	Acionamento físico de botões	NA	Baixo nível e acionamento discreto	Frente, Trás, Giro 45 graus (Esquerda ou Direita)
			BCI	EF+MI, MC+MI, MC+EF	Fusão de Dados		
[6]	Conferência, 2017	IEEE	ET+ Touchpad, ET+Soft Switch	Acionamento físico de Soft-Switch, contato com Touchpad ou Dweel Time; Posição do olhar do usuário na GUI	Cooperação	Baixo nível e acionamento discreto	Frente, Trás, Giro 45 graus (Esquerda ou Direita)
[15]	Journal, 2013	Elsevier	BCI	P300	NA	Baixo nível, discreto	Frente, Trás, Giro (45 ou 90 graus; Esquerda ou Direita)
[16]	Conferência, 2015	IEEE	BCI + ET	(Direção dos olhos, EF, concentração), Posição dos olhos	Fusão de Dados	Baixo nível	Frente, Trás, Giro (Esquerda ou Direita)
[17]	Journal, 2014	Elsevier	BCI	MI e SSVEP	Independentes	Baixo nível	Frente, Trás, Giro, Acelerar, Desacelerar, Manter velocidade, Switch
[18]	Journal, 2013	Elsevier	BCI	SSVEP	NA	Baixo nível, discreto	Frente, Parar, Giro (Esquerda ou Direita)
[19]	Conferência, 2013	IEEE	BCI	SSVEP, Motion Visual Stimulus	NA	Baixo nível	Frente, Trás, Giro (Esquerda ou Direita)
[20]	Journal, 2017	Springer	BCI+ET	P300, Eye Blinking	Cooperação (Switch)	Alto nível ou baixo nível	Escolher Ponto de destinação; Frente, Trás, Giro (Esquerda ou Direita)
[21]	Conferência, 2013	IEEE	ET+BCI	MI e Posição do olhar do usuário na GUI	Cooperação	Discreto	NA
[29]	Conferência, 2011	IEEE	BCI	SSVEP	NA	Baixo nível	Frente, Trás, Giro (Esquerda ou Direita)
[30]	Dissertação, 2007	UFES	ET	EOG, Video-oculografia	Independentes	Baixo nível	Frente, Trás, Giro (Esquerda ou Direita)
[31]	Dissertação, 2017	UFU	ET	EOG, Video-oculografia	Independentes	Baixo nível	Frente, Trás, Giro (Esquerda ou Direita)

[32]	Conferência, 2018	IEEE	ET	Video-oculografia	NA	Baixo nível, discreto	Frente, Trás, Giro (Esquerda ou Direita)
[33]	Dissertação, 2017	UFES	BCI	Concentração	NA	Baixo nível	Frente, Trás, Giro (Esquerda ou Direita)
[34]	Tese, 2017	UFES	BCI+ET	MI, Concentração, EMG	NA	-	-
[35]	Dissertação, 2016	UFU	ET	Video-oculografia	NA	Baixo nível	Frente, Trás, Giro (Esquerda ou Direita)
[36]	Dissertação, 2018	UFU	BCI	SSVEP	NA	Baixo nível	Frente, Trás, Giro (Esquerda ou Direita)

*EF: Expressões
Faciais

*MC:
Movimentos da
Cabeça

*MI: Motor
Imagery

*NA: Não
Aplicável

O livro *Devices for Mobility and Manipulation for People with Reduced Abilities (Rehabilitation Science in Practice Series)*, publicado em 2014 por Bastos Filho [3], trata dos mais variados métodos alternativos para o comando de CRM, portanto também é uma fonte importante para conhecer o estado da arte.

Ressalta-se que foram observados diversos trabalhos com características que serão reproduzidas de forma similar nesta dissertação. Os tópicos seguintes apresentam as características principais e destacam as abordagens próprias.

2.2.1 Modos de atuação

Em alguns dos trabalhos não fica claro se os comandos executam uma ação discreta (exemplo: “ir para frente” um metro) ou contínua, com o usuário mantendo o sinal de entrada. Enquanto os grupos de Shahin [2], Meena [11], e Lopes [24] executam um passo discreto linear ou angular fixo por acionamento, os de Ben Taher [25], Cao [26], e Diez [27] usam acionamentos discretos que promovem um movimento continuado até o recebimento de um comando de parada. O grupo de Ben Taher usou uma janela de tempo de um segundo na qual o estímulo deve ser mantido para que o movimento seja efetuado [25], enquanto o de Diez et al. demanda que seja enviado, a cada segundo, um novo comando para o sistema de controle da cadeira [27]. Escolher entre uma ou outra modalidade é uma decisão entre ser mais importante ou a maior flexibilidade em traçar

trajetórias, ou menor esforço para executar a tarefa, o que pode mudar, dependendo da condição física do operador. Se estiver fadigado, tende a priorizar um menor esforço.

Uma outra alternativa verificada foi a implementada por Punsawat e Wongsawat [28], na qual são proporcionados perfis que o usuário pode escolher. Há três modos possíveis de operação associados às condições normal, fadigado e emergência. Nesta dissertação foi desenvolvido o modo chamado de normal, mas com apoio dinâmico que facilita o controle do cursor da GUI pelo usuário. O modo normal implica também que os estímulos de entrada devem ser mantidos para a continuação dos movimentos.

2.2.2 Interfaces de entrada e relação entre sinais

Neste trabalho foi desenvolvido uma interface multimodal de ET e BCI, modalidade que é utilizada pelos grupos de Ben Taher [25], Puanhvuan [29], e Huang [30]. Ben Taher e seu grupo definem o comando de entrada em uma CRM a partir das expressões faciais identificadas a partir dos sinais EEG, combinadas com as posições da pupila extraídas de imagens oriundas de uma webcam [25]. Tal combinação ocorre por uma técnica de fusão inspirada na teoria da votação. O BCI é usado para escolher a opção de comando pelo grupo de Puanhvuan e, por meio de piscadas capturadas por *eye tracker*, o usuário confirma a seleção [29]. ET e BCI também são aplicados por Huang, Lo e Shi para atuar cooperativamente ao executar tarefas de controle contínuo de um cursor, onde a posição do cursor é identificada por dados oriundos do ET e a intenção do usuário é confirmada por potencial EEG baseado em imagética motora (ou do inglês, *Motor Imagery* - MI) [30]. As interações com o usuário na GUI devem possuir um feedback visual para o usuário, pois conforme observado nos experimentos de Müller [31], isso é importante para melhorar o desempenho da BCI.

No trabalho de Cavalieri [32] foram realizados estudos, simulações, experimentos relacionados ao sinal mioelétrico da piscada e análise do rastreamento videoocular. A interação ocorria por meio de um PDA, porém as interfaces de entrada não atuavam de forma cooperativa, mas separadamente. Já na pesquisa de Martins [33] e Silva [34], o enfoque era desenvolver um ambiente de treinamento para que novos usuários de CRM pudessem realizar treinamentos de forma segura; enquanto o primeiro direcionou para um ambiente de simulação, o segundo contribuiu com uma aplicação real, porém ambos usaram comandos fornecidos por sinais visuais como uma das opções de entrada. A

pesquisa de Couto [35] visava um ambiente de treinamento e foi aplicada inicialmente em um robô móvel, depois em uma cadeira de rodas, com a entrada de comando fornecida de acordo com a atenção do usuário percebida via BCI. Enquanto isso, na tese de doutoramento de Bueno [36], a pesquisa foi voltada ao desenvolvimento de interface cérebro-computador (ICC), com aplicação de redes neurais na classificação dos padrões em sinais de eletroencefalografia de uma base de dados.

Visando comparar de forma mais concisa soluções de comando ocular em relação às necessidades das pessoas a quem se destinava o sistema, Borges [37] realizou uma avaliação de usabilidade e identificou divergências entre as soluções e as demandas, o que levou alguns usuários a não conseguirem utilizar o sistema proposto. Ressalta-se que os sistemas devem efetivamente ajudar os usuários a contornar as dificuldades impostas pela sua condição física, como por exemplo, reduzindo o esforço em manter o sinal em interfaces mais cansativas ou manter o cursor da GUI nas regiões de interesse do usuário.

O trabalho de investigação de Zambalde [38] indica a possibilidade de usar um monitor LCD para gerar os estímulos visuais no caso de adotar o SSVEP como potencial de referência.

A abordagem adotada nesta pesquisa possui características mais próxima aos trabalhos dos grupos de Ben Taher [25] e Huang [30]. A combinação de ET e BCI ocorre no nível de informação, o que implica em processar previamente os dados de cada interface. A informação obtida pelos dados do *eye tracker* se reflete na posição do cursor da GUI. O papel do BCI é prover um meio ao usuário reforçar a informação do posicionamento do cursor já informado pelo ET. Em seguida, uma técnica de fusão das informações geradas pelas interfaces de entrada é implementada para, estatisticamente, dar mais certeza quanto ao comando intencionado pelo usuário.

Os objetos da fusão do sistema proposto são resultados de associações anteriores que se convertem nas informações de posição do cursor e recebimento de potenciais de EEG treinados pelo usuário. Assim, a fusão pode ser vista como no nível de decisão. As informações geradas por cada interface devem passar por uma nova etapa decisória para definir o comando de entrada e, portanto, a arquitetura estrutural apresenta-se como híbrida ou distribuída. O comportamento do sistema vinculado à fusão, com etapas de observar, orientar, decidir e agir, é semelhante ao descrito no modelo de fusão Omnibus [19]. Já o relacionamento entre as informações e o resultado da fusão, proporcionando informações mais complexas que as originais, remete a um processo de fusão cooperativa.

A fusão de informação também pode ser tratada como fusão de alto nível. Dentre os métodos, em nível de decisão, os mais comuns são o Bayesiano e o *Dempster Shafer* [39]. O primeiro se baseia nas teorias probabilísticas de Bayes, onde se definem probabilidades de eventos condicionados à ocorrência de outros. O segundo define matematicamente crença em hipóteses em termos de máxima e mínima probabilidade. Apesar do método *Dempster Shafer* apresentar mais robustez a ruídos e imprecisões de informações a priori, versões modificadas de métodos bayesianos demonstraram maior eficiência e extensibilidade [39].

Um outro método probabilístico encontrado na literatura é baseado em cadeias de Markov, o qual permite modelar e avaliar sistemas dinâmicos com um conjunto de estados finitos, suas relações e transições, gerando mínima complexidade computacional. Além de ser escalável, é um método capaz de incorporar incertezas [39].

Capítulo 3 – Concepção Inicial da Solução

Por meio da avaliação de literatura, recursos disponíveis no laboratório e financiamento proveniente de cooperação acadêmica entre a Universidade Federal do Amazonas (UFAM), UFES e UFU, foi definido o uso do *eye tracking* e do BCI. A Figura 10 apresenta a estrutura geral do sistema, com os principais componentes e interfaces de comunicação.

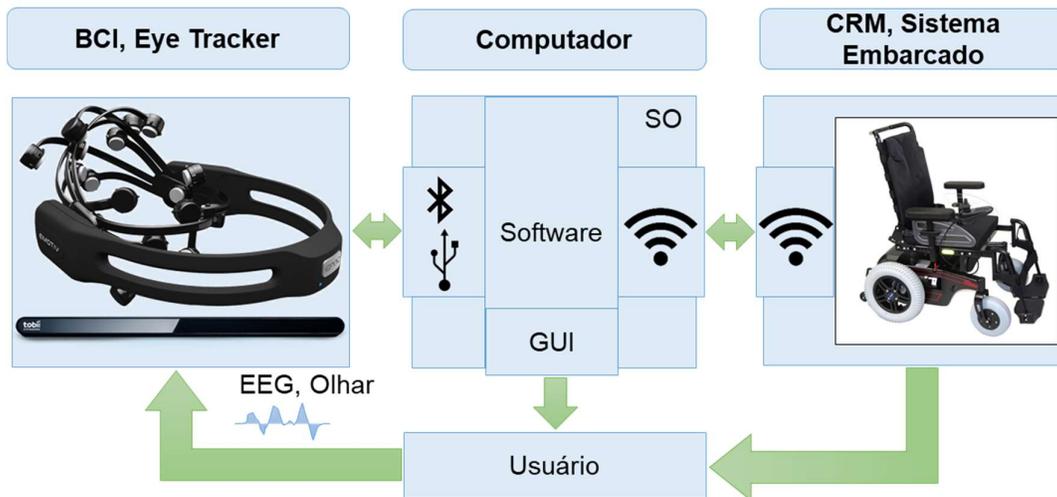


Figura 10 - Estrutura geral da proposta de sistema.

Neste trabalho, para o controle da cadeira de rodas local, a comunicação pode ocorrer via roteador, enquanto que no caso da cadeira remota é necessário estabelecer até ela um caminho pela internet. Em ambos os casos deve ser estabelecida uma arquitetura cliente-servidor, onde a CRM, utilizando um sistema embarcado (servidor), fornece ao Computador (cliente) o serviço de locomoção via comandos codificados e padronizados. Este último gerencia os processos, estados e a Interação Humano-Computador (IHC).

O usuário é disposto no sistema com autonomia para sua própria locomoção e deve perceber a movimentação da cadeira de rodas e fornecer os devidos comandos para que ela alcance a localização desejada. Os comandos de entrada são obtidos pelo uso simultâneo do *Eye Tracker* e BCI, portanto, as restrições do usuário não podem ser tão severas que impeçam a utilização de algum dos dispositivos.

Na GUI estão presentes um cursor, controlado por ET, e as opções de comando da cadeira. Uma vez combinados por fusão, os dados processados do ET e BCI são enviados ao controlador da CRM previamente adaptado para atuar, não somente por

joystick, mas também por *software*. O sistema computacional é adequado para responder rapidamente à interação do usuário, permitindo este opere a cadeira de forma segura.

Os programas para permitir o usuário interagir com o *eye tracker* e o BCI foram implementados usando as APIs fornecidas pelos fabricantes. Os dispositivos disponíveis são o Tobii Eye Tracker 4C e Emotiv Epoc+. Os pacotes de desenvolvimento permitem a criação de aplicações em diversas linguagens como Python, C, C++, C#, Java e .NET.

A Figura 9 apresenta outros detalhes dos elementos que constituem o sistema, suas interações e respectivas saídas. Os sinais EEG do usuário são amostrados pela BCI. Por meio do ET, o usuário é capaz de movimentar o cursor na GUI e navegar entre as opções de comando disponíveis. O P300 e SSVEP são opções de potencial EEG que serão avaliadas. A literatura apresenta o primeiro como o que exige menor esforço de concentração e atenção, atingindo desempenho relativamente alto e com baixo tempo de processamento.

Para a implementação de captura de sinal EEG baseado em P300 ou SSVEP, é necessário um estímulo luminoso, que pode ser gerado pela GUI ou mecanismo externo sincronizado à operação. O processamento das duas entradas, o olhar e o sinal EEG do usuário, geram respectivamente a posição do cursor e uma informação sobre identificação ou não do estímulo visual. Essas informações passam por uma etapa de fusão, onde são combinadas para identificar o comando percebido. Tal comando deve passar a seguir por uma infraestrutura de rede, que o transmite ao controlador da cadeira de rodas motorizada local ou remota, e este gera os sinais para a execução do movimento.

Na Figura 11 consta um diagrama que mostra o fluxo com que deve ocorrer a interação entre os elementos do sistema. Os comandos identificados são usados para apresentar um *feedback* visual ao usuário conforme opera o sistema.

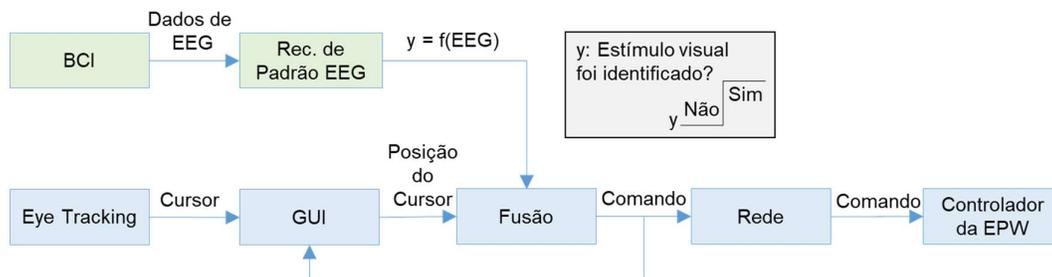


Figura 11 - Diagrama de interações de componentes do sistema.

O desempenho de entradas baseadas em ET podem ser similares aos de *mouses* e *joysticks*, a depender do tamanho das regiões de interação com a GUI; quanto maiores os

elementos, melhores os resultados [40]. Para lidar com o problema de comandos não-intencionais, também conhecido na literatura como “Midas Touch”, a seleção de comandos demanda confirmação que também depende da BCI.

A associação das informações geradas pelo *eye tracker* e a BCI ocorrem em uma etapa de processamento de alto nível. O contexto da aplicação também é utilizado no cálculo que estima a intenção do usuário a partir da qual, na sequência, realiza-se o comando a ela associada. A arquitetura completa de fusão no sistema é apresentada na Figura 12.

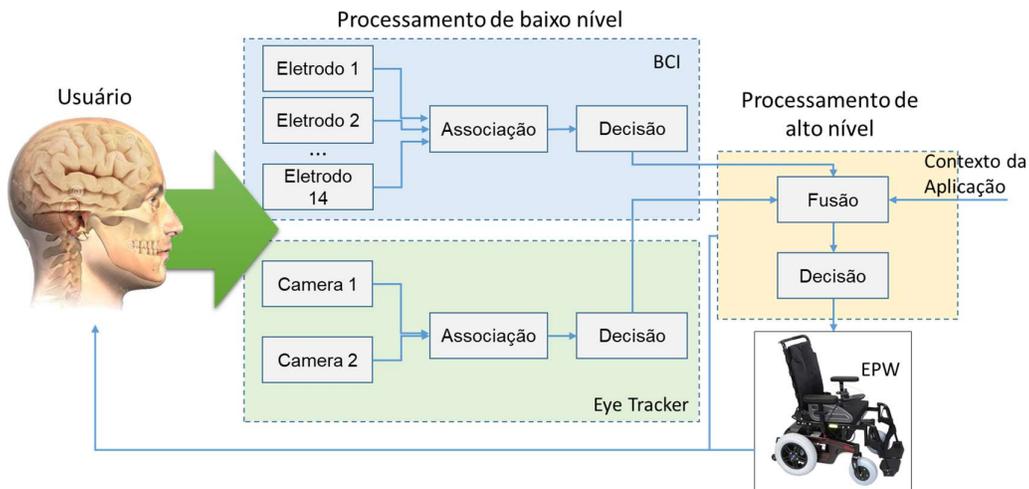


Figura 12 - Arquitetura do sistema relativa a fusões.

As Figuras 12 e 13 apresentam o processo de seleção de comandos. O posicionamento dos elementos na interface gráfica reproduz o formato com melhor desempenho verificado ao comparar disposições encontradas na literatura para uso com ET [41]. Esta abordagem de comando da cadeira é chamada de baixo nível, caracterizada por controlar a cadeira com movimentos simples: ir para frente, para trás ou girar. Inicialmente o cursor deve ser posicionado na região inferior esquerda, com o ícone de cadeado, o que o torna verde enquanto que a região circular central muda para a cor amarela. A seguir, o cursor deve ser posicionado nessa área amarela central, que deve cintilar e assim provocar um estímulo visual. O reconhecimento deste estímulo nos sinais obtidos pela BCI é usado na primeira etapa de fusão, juntamente com a informação do cursor, que confirma a intenção de comandar a cadeira. Isso torna a região central verde e libera para serem selecionadas as opções de movimento. Ao selecionar uma das alternativas, a região correspondente a ela provoca um novo estímulo visual, cujo reconhecimento conduz a uma nova etapa de fusão, que confirmará a seleção. A cadeira

começará a se mover e, para manter o movimento, o cursor deve ser mantido na região do comando selecionado até que ela alcance a posição desejada.

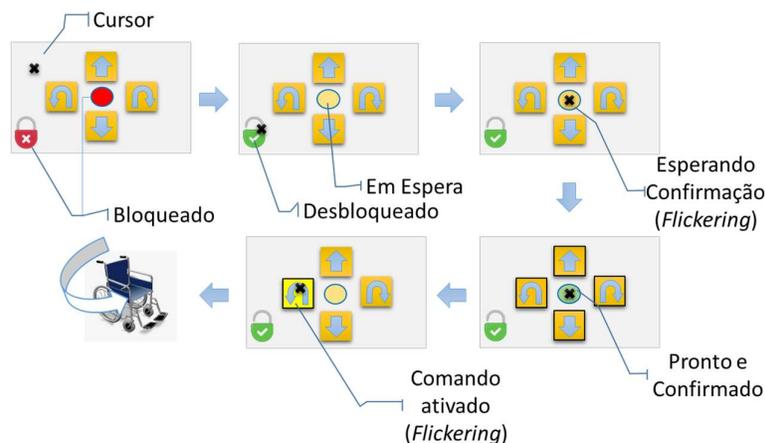


Figura 13 - Concepção inicial da GUI e sequência de operação.

A região central se torna amarela novamente após seleção de movimento, o que demandará uma nova confirmação de interação para selecionar um outro comando. Depois de um tempo de inatividade, sem comandos selecionados, a GUI retorna para o estado inicial de bloqueio.

A seleção de comando é descrita em sete estados de interações, conforme ilustrado no diagrama de estados da Figura 14. Uma característica de cada estado é listada na Tabela 5.

Tabela 5 - Estados de interações com o sistema.

Estado	Característica
A	Interação Bloqueada
B	Desbloqueio de Interação
C	Transição com limite de tempo
D	Cintilação da área circular
E	Confirmação – estímulo visual identificado em sinal EEG
F	Transição com limite de tempo
G	Seleção e Confirmação de Comando

As opções de movimento da cadeira somente estão disponíveis entre os estados E e G, o que ocorre após o reconhecimento do estímulo visual no sinal EEG do usuário

provocado no estado D. No estado G a única opção disponível é a selecionada, então, escolher outro comando é possível somente após outra confirmação, o que significa necessariamente retornar ao estado C. Durante o desenvolvimento, devem ser estudados quais os limites de tempo de inatividade adequados que provoquem a transição automática do estado F para C ou do C para o A.

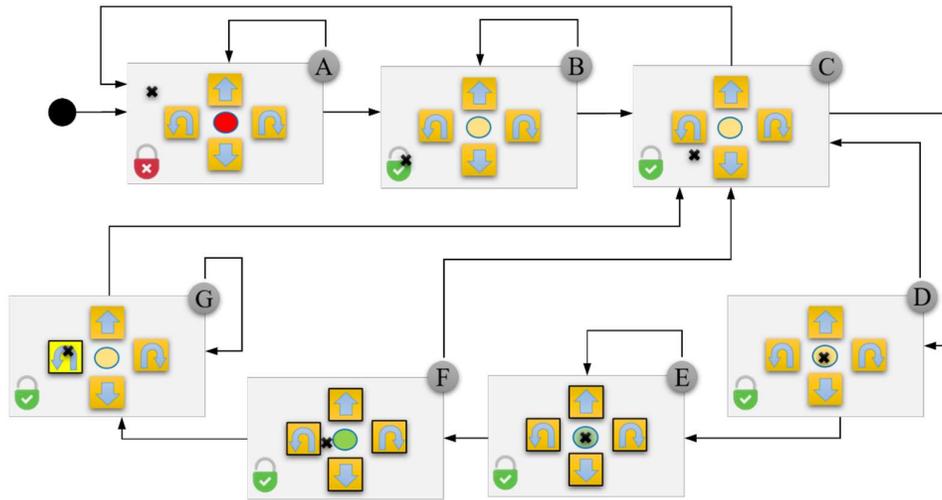


Figura 14 - Diagrama de estados das interações possíveis até a seleção do comando 'girar para esquerda'.

O estado A permanece se o cursor não selecionar o ícone do cadeado do canto inferior esquerdo. Uma vez desbloqueado, o sistema alcança o estado B, que é mantido enquanto o cursor continuar sobre o ícone indicado. No estado C, o sistema permanece enquanto o cadeado ou a região central não são selecionados, situações onde retornaria para o estado A ou avançaria para o D, respectivamente. No estado D um estímulo visual ocorre para confirmar a intenção de mover a cadeira que, uma vez identificado, conduz o sistema ao estado seguinte onde as opções de movimento tornam-se disponíveis. O sistema continua no estado E se o usuário mantiver o cursor na região circular verde, do contrário o estado F é alcançado. A seleção de uma opção de comando inicia o estado G, onde ocorre outro estímulo visual, cujo reconhecimento faz com que o movimento da CRM comece. A cadeira continuará o movimento escolhido se o cursor se mantiver na região do comando escolhido. O diagrama acima ilustra os estados interativos da GUI para selecionar o comando “girar para a esquerda”. Porém, os demais comandos são análogos, com a diferença ocorrendo no estado designado pela letra G, onde o usuário seleciona a opção de movimento que a cadeira deve iniciar.

Não haverá um comando de parada explícito. Após confirmação de comando, uma *thread* deve ser executada ciclicamente a certo intervalo de tempo para que o sinal de controle seja mantido e o movimento continue. Porém, isso ocorrerá se e somente se o cursor for mantido na opção selecionada. Para comandar a CRM local, o ciclo de execução da *thread* será definido de acordo com a taxa de transmissão da entrada até o sistema embarcado à cadeira e deste ao controlador de movimento. No caso da cadeira remota, deve ser considerado, além do sistema de controle da cadeira, também o tempo de transmissão e recepção de mensagens entre Manaus e o CT. Espera-se com isso reproduzir, ainda que limitadamente, o comportamento proporcionado ao comandar a CRM via *joystick*.

Capítulo 4 – Desenvolvimento

A interface de comando alternativa é composta por dispositivos *eye tracker* e BCI comerciais, o que demandou o desenvolvimento tanto em software quanto em hardware.

O sistema de software é composto de programas de alto e médio nível, incluindo o sistema embarcado instalado na cadeira de rodas que permite a integração do conjunto. Baseado em uma arquitetura *Model-View-Controller* (MVC) e orientada a objetos, foram desenvolvidos *scripts* nas linguagens Python, C# e C, alguns dos quais se comunicam intermediados por uma infraestrutura de rede e *socket*. A IHM é composta pelos dispositivos de entrada alternativos e GUI, e é responsável pela interação do usuário com o controle da CRM. Um módulo coordenador direciona as funcionalidades das demais entidades para prover ao usuário o serviço de comandar a cadeira de rodas via dispositivos de entrada alternativos.

Em relação a hardware, realizou-se adaptações na cadeira de rodas com a inclusão de um sistema embarcado. Enquanto a IHM envia os comandos via *socket* e rede sem fio para este sistema, ele é responsável por decodificá-los em sinais de tensão para efetivamente executá-los.

O código fonte completo é mantido em repositório no Github no endereço <https://github.com/robsongcruz/Mestrado>.

4.1 Software

O sistema foi desenvolvido sob a arquitetura MVC, na qual foram mapeadas e agrupadas as unidades conforme representadas na Figura 15.

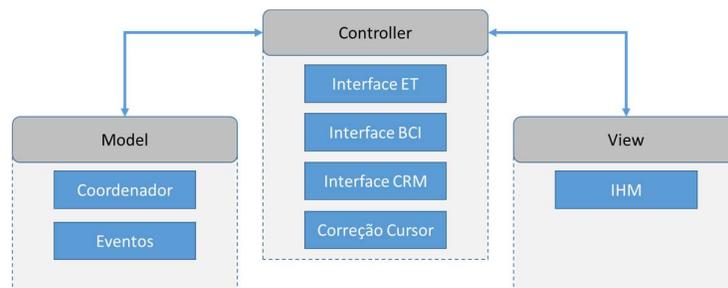


Figura 15 - Principais elementos do software do sistema na visão do modelo MVC.

O sistema de *software* é responsável por interagir com o usuário e atuar como cliente de um sistema embarcado responsável por enviar os comandos ao controlador da CRM. As unidades foram projetadas orientadas a objetos e são constituídas por uma ou mais classes.

- Interface ET : composto por um *script* em Python e outro em C#. São responsáveis por receber e processar a posição do olhar do usuário na tela, permitindo dessa forma que ele controle um cursor na GUI e assim selecione comandos de movimentação da cadeira. O *script* em C# interage diretamente com a API do fabricante do *eye tracker* e, por meio de *socket*, transmite os dados à instância em Python, sendo responsável por realizar o processamento adequado, seja na condução dos estados do sistema como também para posicionar o cursor corretamente.
- Interface BCI: conjunto de *scripts* que fornecem as funcionalidades do BCI necessárias ao sistema, tais quais: a captura dos dados, treinamento com EEG do usuário e uma unidade de comunicação que envia os dados para o Coordenador (que se detalha mais adiante).
- Interface CRM: códigos desenvolvidos no sistema embarcado instalado na CRM que emula os sinais do *joystick* de acordo com os comandos de movimentação selecionados pelo usuário. A comunicação com o módulo de comandos ocorre via *socket*, por meio do qual a aplicação recebe pacotes JSON, interpretados e realizados em porta digital, e em seguida convertidos em sinal analógico de forma a emular o que seria gerado diretamente via *joystick*.
- Correção de cursor: *script* para realizar o tratamento dos dados do *eye tracker*. Os algoritmos resultam em suavização do trajeto realizado pelo cursor da IHM, de acordo com o estado do sistema.
- IHM: interface humano-máquina que atua como um canal de interação do usuário com o sistema. É caracterizado por componentes de *software* e *hardware*.
- Eventos: unidade funcional onde são mapeados os eventos necessários à interação do usuário com a IHM. Esses eventos são embutidos nos componentes presentes na GUI para permitir a transição entre estados do sistema segundo os eventos ativados durante a interação com o usuário.
- Coordenador: responsável por coordenar as funcionalidades proporcionadas pelas demais entidades para o prover o serviço de comando da CRM. Para que seja possível o comando, o usuário deve seguir certos passos mapeados em uma máquina de estados.

As classes que compõem o sistema e representam as unidades funcionais apresentadas anteriormente são mostradas na Figura 16.

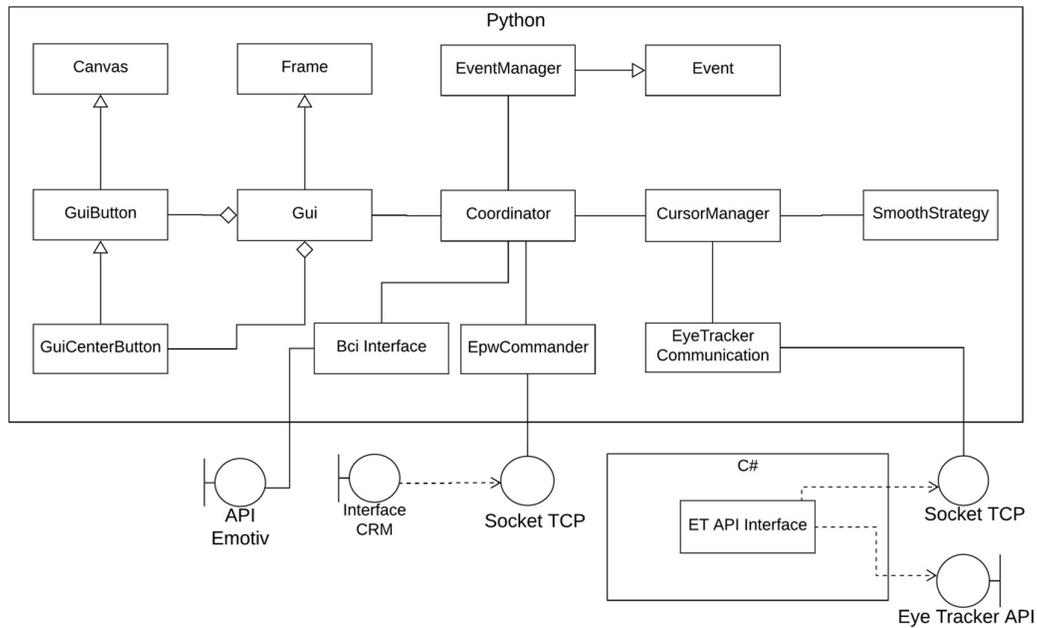


Figura 16 - Diagrama de classes do sistema de software com suas fronteiras.

O modelo da interface gráfica com o usuário (GUI) foi projetado de forma similar ao mostrado por Ktena, cujo trabalho identificou que essa distribuição de componentes gráficos, comparado às demais estudadas, alcança melhores resultados em relação à usabilidade proporcionada ao usuário [41]. A GUI é apresentada na Figura 17.

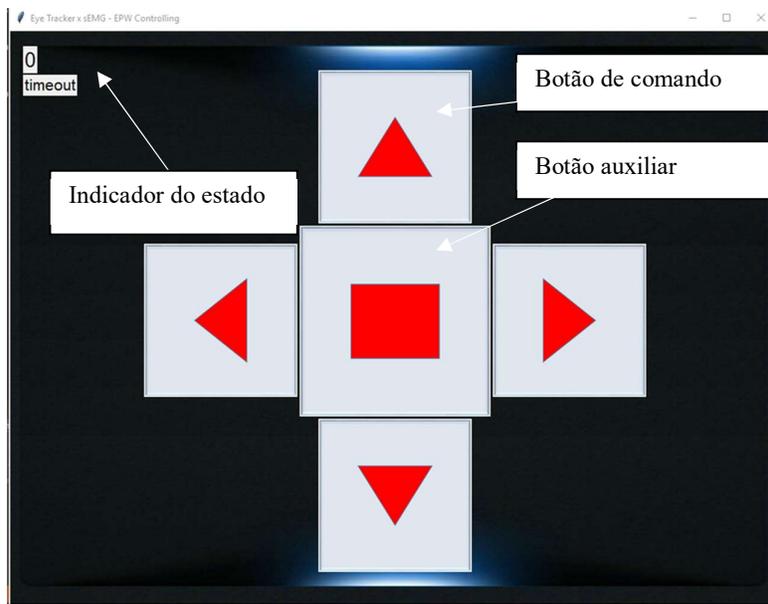


Figura 17 - Interface gráfica do sistema com um indicador de estado para testes.

A classe Gui é a que fornece a interface gráfica, por meio da qual ocorrem as interações do usuário com o sistema. Foi desenvolvida como uma extensão da classe Frame e agrega as instâncias das classes de GuiButton e GuiButtonCenter, que representam os botões pelos quais são realizados os comandos de movimento da cadeira de rodas. As classes Frame e Canvas pertencem ao toolkit tkinter da linguagem Python.

Conforme mencionado anteriormente, a interface de entrada do sistema é composta por ET e BCI, com intermédio da GUI e um cursor alternativo. Porém, diferente do indicado inicialmente, onde o usuário deveria desbloquear a interação antes de desbloquear os comandos, não haverá a ação de desbloqueio de interação, pois isso adicionava uma operação a mais sem que fosse possível estabelecer, pela pesquisa, que isso oferecia mais segurança ao usuário.

A classe que realiza o cursor alternativo da GUI é a CursorManager, associada às instâncias de EyeTrackerCommunication e SmoothStrategy; a primeira é uma das partes da interface com dispositivo ET, enquanto a segunda encapsula funções para suavização do trajeto do cursor. Complementar à classe EyeTrackerCommunication, foi desenvolvido um *script* em C# que interage diretamente com a API do *eye tracker*. A opção pela linguagem C# ocorreu pelo fato da API também ser implementada nessa linguagem, o que aumenta a compatibilidade e simplifica o processo de desenvolvimento.

A evolução da máquina de estados ocorre segundo os eventos apresentados na Figura 18.

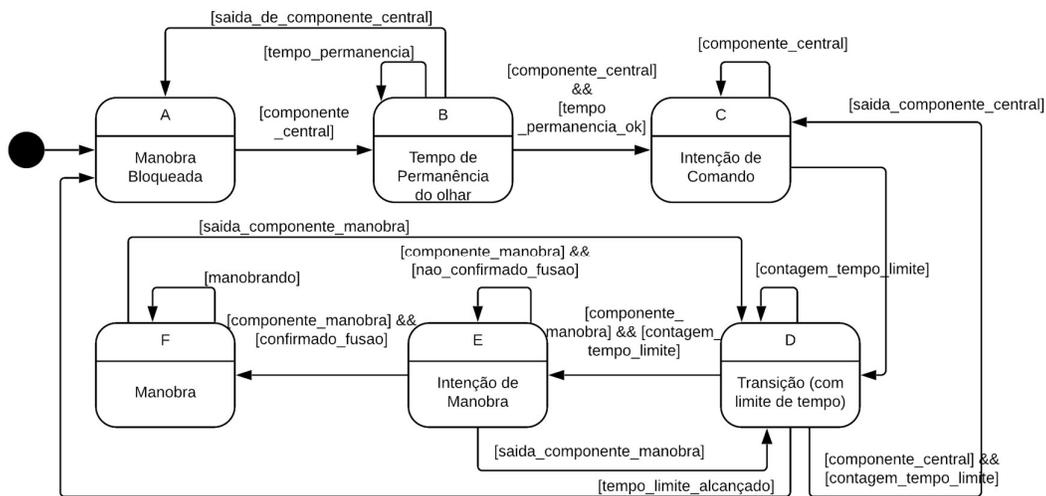


Figura 18 - Máquina de estado do sistema na operação de selecionar comando.

A classe Coordenador combina uma máquina de estados interna com as funções implementadas nas classes EventManager, CursorManager e Gui. A class EventManager

é uma extensão da classe Event, que é fornecida também pelo toolkit tkinter disponível na linguagem Python. Essa classe implementa ações de resposta a eventos de controle e sua instância é fornecida à GUI para vincular os eventos nativos dos componentes gráficos a essas ações.

A interação com usuário inicia no estado A, no qual os comandos estão bloqueados e isso é indicado pela cor vermelha nos componentes. Ao posicionar o cursor alternativo no componente central, a máquina de estados avança do estado A para um estado de transição B, cuja transição para o estado C seguinte é satisfeita quando a média das amostras (x, y) de posição do olhar indicam cada coordenada entre os valores $(x_c \pm 66, y_c \pm 66)$. O número de amostras utilizado é de $n=100$, o que implica em uma janela de tempo em cerca de 1s.

A Figura 19 apresenta de que forma é estimada a posição do cursor em relação às amostras do ET e a precisão com a qual isso ocorre. De forma similar é avaliada a posição do cursor nos demais componentes gráficos de comando, como na transição do estado E para o estado F.

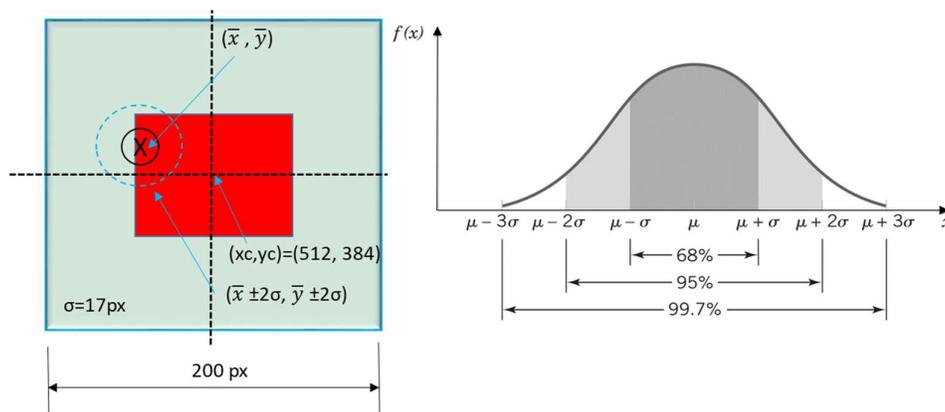


Figura 19 - Distribuição normal x precisão na posição do cursor.

Considerando o componente quadrado com lado definido em 200px, e as amostras normalmente distribuídas com parâmetros $N_x \sim (\mu_x = \bar{x}, \sigma_x^2 = 17^2)$ e $N_y \sim (\mu_y = \bar{y}, \sigma_y^2 = 17^2)$, onde $\bar{x} < |x_c - 66|$ e $\bar{y} < |y_c - 66|$, é possível afirmar que a posição do olhar do usuário está sobre o componente de comando com intervalo de confiança maior ou igual a 95%.

O estado C representa que há uma intenção do usuário de movimentar a CRM e é sinalizado com a cor verde no componente central. Deslocar o cursor para fora do componente antes de alcançar o estado C, retorna o sistema ao estado de bloqueio A. A técnica de *dwell time* é usada para confirmar a ação e, assim, o alcance do estado C.

A transição do estado C para o estado D ocorre ao sair da região do componente central após a confirmação, o que ativa um *timer* de 3s. Um comando de movimentação deve ser selecionado nesse intervalo de tempo ou o sistema retorna para o estado de bloqueio A. Se o cursor retornar ao componente central, o sistema retorna para o estado C. Ou se um dos comandos for selecionado, ocorre a transição para o estado E.

A transição do estado E para F, como envolve a movimentação da cadeira de rodas, exige um tratamento com menor possibilidade de erros ou falhas. Além da abordagem probabilística aplicada na transição B-C, para garantir, até com 95% de confiança que o cursor está sobre o componente, o sistema também deve receber da BCI a informação de que o padrão de sinal EEG treinado foi recebido. A transição caracteriza-se pela confirmação multimodal (ET + BCI) do comando. Essa confirmação, por duas fontes diferentes, aumenta a probabilidade de que o usuário realmente o esteja ativando. Os potenciais devem ser estimulados de acordo com o método escolhido e treinado.

Por fim, o estado F da máquina de estados do sistema é mantido enquanto o cursor permanecer sobre o componente de comando selecionado e já confirmado na transição E-F. Quando o cursor sai da área do componente, retorna-se ao estado de transição D, de onde o usuário pode optar por outro comando ou selecionar o mesmo comando novamente.

O comportamento da GUI, conforme evolução dos estados, é apresentada na Figura 20.

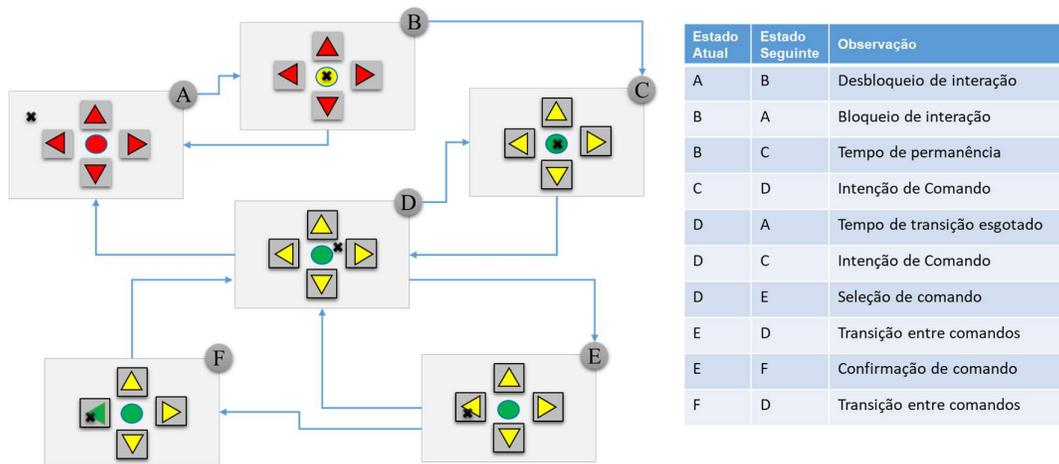


Figura 20 - Comportamento da GUI e descrição das transições.

A comunicação entre as camadas em Python e C# da interface do ET é possível por meio de *sockets* sob o protocolo TCP (*Transmission Control Protocol*). A aplicação

em C# atua como um cliente TCP, coletando os dados do Eye Tracker fornecidos pela API e transmitindo uma *stream* de dados em formato JSON com o padrão

{“x”: [posição x do olhar], “y”: [posição y do olhar]}

a qual é recebida como um objeto da classe String, de até 1024 bytes (*buffer*), pela instância da classe EyeTrackerCommunication, com o papel de servidor TCP. A *string* é processada para gerar um vetor bidimensional com os valores do tipo inteiro contendo as posições informadas. A partir desse vetor são aplicados os cálculos que estimam a posição real do olhar do usuário. Apesar do protocolo TCP ser um pouco mais lento que o UDP (*User Datagram Protocol*), ele possui mecanismos que oferecem maior garantia de entrega das mensagens, então esse foi adotado. Isso foi feito a partir da importação da biblioteca socket no Python e System.Net.Sockets no C#.

Foi observado durante os estudos de uso a ocorrência de pequenas oscilações rápidas no cursor durante o uso do eye tracker que parecem ser resultado tanto da própria volatilidade no olhar dos usuários quanto a ruídos. Essas oscilações, além do desconforto, também induzem a mais variações na posição do olhar do usuário, pois o usuário é compelido a olhar para o cursor. Levando em conta esse problema, a posição do olhar é calculada na instância da classe SmoothStrategy utilizando alternativamente dois algoritmos de fusão redundante e assim definir a posição do cursor: uma média móvel simples (MM) e outro baseado em filtro de Kalman (KF) em versão unidimensional (para cada coordenada). Variações bruscas, em torno de valores próximos, que representam componentes de alta frequência, são filtrados com os algoritmos e, com isso, observa-se um trajeto mais suave do cursor do que utilizar os dados brutos (x, y) informados pela API do dispositivo *eye tracker*.

A arquitetura de fusão do sistema é distribuída e compõe-se de duas unidades – uma redundante e outra cooperativa, fora as unidades internas às API dos fabricantes dos dispositivos de entrada. A fusão redundante é caracterizada pela estimativa de posição do cursor da GUI controlado via *eye tracker*, onde são aplicadas duas formas de cálculo alternantes de acordo com o estado do sistema. As amostras de posição do olhar informados pelo ET são combinadas em um valor único que representa a posição real mais provável do olhar do usuário.

A aplicação das duas interfaces de entrada de forma complementar varia essa confiança conforme se estiver utilizando uma (menor confiança) ou ambas (maior confiança). Como há interseção entre os eventos de entrada dos dispositivos,

matematicamente, pode ser verificado que a probabilidade da interseção associada à confiança é no máximo igual à maior entre os eventos. Por outro lado, a probabilidade da interseção dos erros também é reduzida e será menor do que as probabilidades individuais, pois os valores são menores que 1. Isso justifica a notação de menor ou maior confiança. O cálculo da probabilidade da interseção de dois eventos independentes é realizado pela expressão $P(A \cap B) = P(A).P(B)$.

A estratégia adotada para melhorar a experiência do usuário, ao mover o cursor com o *eye tracker*, foi utilizar a média móvel nos estados cuja confiabilidade na posição do cursor deve ser maior. Por outro lado, o filtro baseado em KF é aplicado nos estados onde a confiabilidade pode ser menor, mas deve ser rápida a resposta às mudanças de posição do olhar do usuário.

A média móvel converge mais lentamente que o filtro baseado em KF, conseqüentemente, reage mais lentamente às mudanças de posição do cursor. O efeito obtido com a MM é que nos estágios de maior confiança o cursor apresenta um leve atraso na atualização da posição e menor susceptibilidade aos ruídos que geram pequenas oscilações em torno da posição real do olhar do usuário. Isso ajuda a manter o cursor na região de comando e evita o desconforto causado pelas pequenas variações na posição dele decorrentes dos ruídos e de pequenos deslocamentos involuntários do olhar. Os passos para estimar a posição pelo filtro são apresentados na Figura 21.

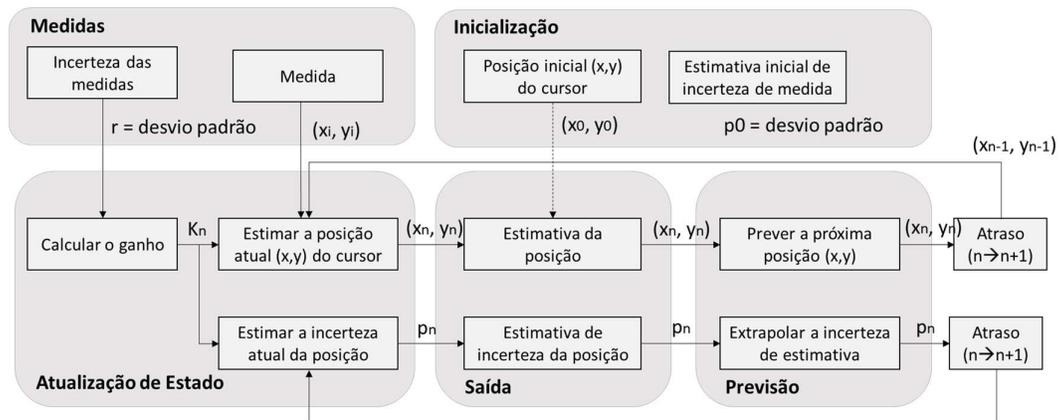


Figura 21 - Etapas de cálculo da posição do cursor via algoritmo KF.

A cada chamada do sistema para calcular a posição (x, y) do cursor, todas as amostras do vetor de posições são usadas para estimá-la. O ciclo apresentado na Figura 21 é repetido até que sejam usadas todas as amostras. Entre um ciclo e outro, as amostras usadas não são descartadas - quando novos valores são recebidos, a mesma quantidade é descartada das amostras mais antigas, mantendo o vetor com 100 elementos. A taxa de

amostragem do *eye tracker* é cerca de 90Hz segundo o fabricante, portanto a janela de amostragem possui em torno de 1 s (100 amostras * 1 / 90 ~ 1s). O final das iterações resulta na posição estimada do olhar do usuário, cujas coordenadas são aplicadas ao cursor alternativo da GUI.

A execução de cada ciclo no cálculo da posição do cursor ocorre a partir da aplicação das equações de (I) a (IV) sobre o vetor com as amostras:

$$\left\{ \begin{array}{l} \hat{x}_n = \hat{x}_{n-1} + G_n(y_n - \hat{x}_{n-1}) \text{ (I)} \\ G_n = \frac{p_{n-1}}{p_{n-1} + r} \text{ (II)} \\ p_n = (1 - G_n) \cdot p_{n-1} \text{ (III)} \\ p_{n+1} = p_n \text{ (IV)} \end{array} \right. , \text{ onde } \left\{ \begin{array}{l} \hat{x}_n: \text{estimativa atual da posição do olhar} \\ \hat{x}_{n-1}: \text{estimativa anterior da posição} \\ y_n: \text{amostra } n \text{ (vetor)} \\ r: \text{incerteza na medida} \\ G_n: \text{Ganho de Kalman} \\ p_n: \text{Covariância} \end{array} \right.$$

O modelo de fusão de informações do ET e da BCI é melhor categorizado como cooperativo, pois são dados heterogêneos cuja combinação resulta em uma nova informação mais complexa que as iniciais – a seleção de comando do usuário. A interface de entrada dominante é o *eye tracker*, que é usado para mover o cursor ao longo da tela. A Figura 22 estabelece o fluxograma de como ocorre a alternância nas expressões de estimação da posição que são aplicadas ao cursor alternativo.

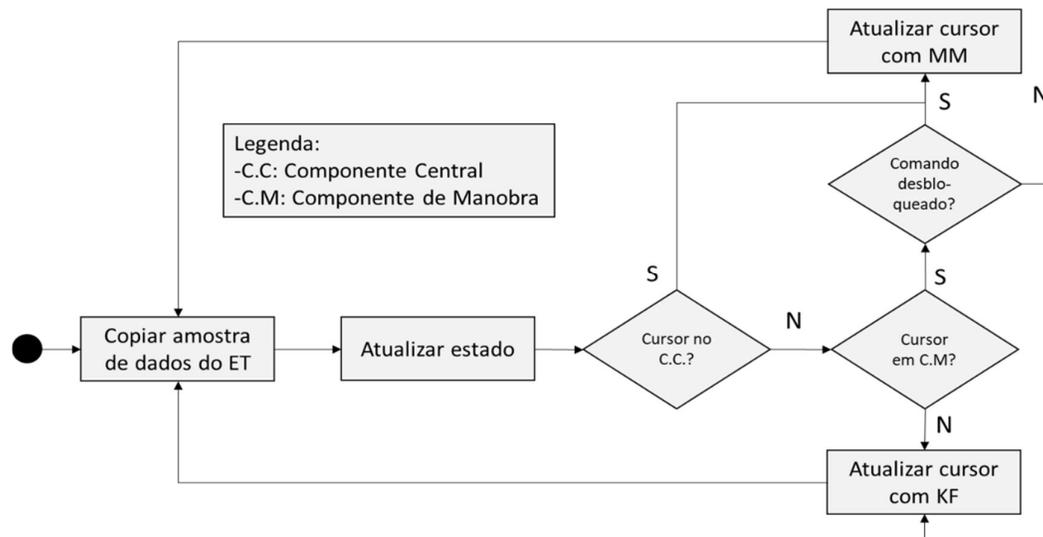


Figura 22 - Fluxograma com alternância de algoritmos para cálculo da posição do cursor.

A interação demandada via BCI é pontual e ocorre no estado E do sistema. Após o usuário seguir os estados preliminares e chegar ao E, o sistema entende que há a intenção de manobrar a CRM, o que é confirmado por uma notificação originada na instância da classe de operação do BCI. Se as demais condições do estado E são atendidas,

a máquina de estados avança para o estado F, onde a movimentação da cadeira é iniciada. Durante a execução do movimento selecionado, não é necessário manter o estímulo do potencial EEG, apenas o cursor sobre o comando no botão da GUI. A minimização da interação via BCI torna menos cansativo ao usuário operar o sistema.

A Figura 23 apresenta a arquitetura de fusão do sistema, atualizada com as novas necessidades identificadas no decorrer da pesquisa.

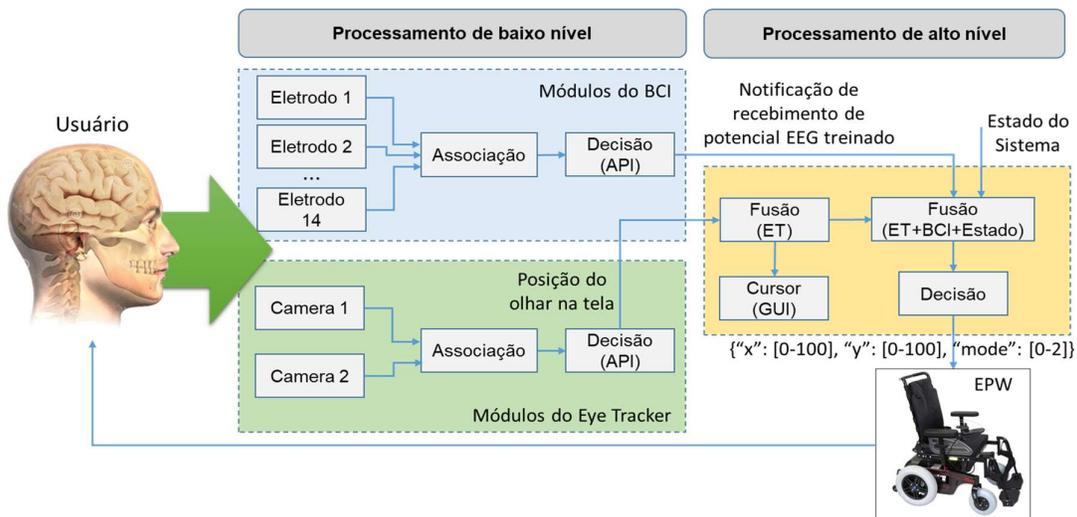


Figura 23 - Arquitetura de fusão atualizada.

Conforme pode ser observado na Figura 23, a fusão depende da notificação de que o potencial EEG treinado foi recebido, do estado do sistema e informação do ET. A decisão para ativação ou não ocorre a partir do valor estimado das amostras de rastreamento ocular e de um indicador quantitativo recebido da API do BCI.

O evento de confirmação do comando que caracteriza a fusão cooperativa entre ET e BCI foi modelado como um evento equivalente da combinação série-paralelo de eventos – o primeiro é o posicionamento do cursor no componente de comando, seguido pelos eventos paralelos de posicionamento (mantido) e confirmação via BCI. Isso é apresentado na Figura 24. $P(A)$ corresponde à confiabilidade na estimação da posição do olhar, $P(B//C)$ é a confiança alcançada com a interação em paralelo do ET e BCI, enquanto que $P[A \rightarrow (B//C)]$ se refere ao processo completo, com a ocorrência do evento A seguido pelos eventos B e C de forma paralela. A expressão obtida de $P[A \rightarrow (B//C)]$ representa a fusão das informações do ET e BCI e varia entre 0 e 1, conforme for maior ou menor a probabilidade da correta estimação das informações associadas aos eventos.

A confiança da correta estimação na posição do cursor, relacionada a ocorrência dos eventos A e C, foi definida e calculada em 95%, permitiu construir as expressões apresentadas a seguir.



Figura 24 - Ordenação dos eventos de entrada

$$P[A \rightarrow (B//C)] = P(A) \cdot P(B//C) = 0,9025 + 0,0475 \cdot P(B) \text{ (II)}$$

O evento P(B) na expressão (II) representa o evento de reconhecimento do comando treinado na BCI e é quantificado pelo valor da variável *power of signal* obtido via API, o qual representa o quão similar o sinal identificado é em relação ao que foi treinado. A API permite o treinamento de até 5 sinais (neutro + 4 comandos) por perfil, porém, uma maior quantidade de comandos treinados implica também em maior dificuldade do algoritmo na classificação. Para esse sistema são utilizados 2 sinais: o neutro, mais um sinal associado à confirmação de comando.

Pela expressão (II), foi verificado que a maior confiança possível de obter é limitada a 0,95 ou 95%, que é o percentual alcançado pelo ET individualmente, porém, isso demandaria uma confiança de 100% no segundo dispositivo de entrada que, no caso, é a BCI. Outros possíveis níveis de confiança podem ser observados na Tabela 6.

Tabela 6 - Confiabilidade do evento B versus confiabilidade geral.

P(B)	P[A->(B//C)]
0,7	0,93575
0,8	0,9405
0,9	0,94525
0,95	0,947625
0,96	0,9481

O algoritmo de classificação da API é preparado para identificar potenciais de EEG associados à MI, expressões faciais, concentração e alguns outros chamados de métricas de desempenho. Segundo informações obtidas do fabricante, podem ser aplicadas outras técnicas, desde que implementado um algoritmo próprio para a análise dos potenciais de EEG que esteja de acordo com a estratégia adotada. No caso do uso do SSVEP ou P300, é recomendado que o dispositivo BCI seja usado na posição contrária,

pois proporciona melhor sensoriamento do cortex visual. A aplicação foi preparada para atuar com sinais EEG gerados por imagética motora ou por expressões faciais, que são nativos na API da Emotiv.

O limiar definido para a confirmação do comando de manobra a partir da expressão (II), de $P[A \rightarrow (B//C)]$, é de 94%, o que exige a média das amostras da variável *power of signal* maiores que aproximadamente 0,8. Após confirmado, uma mensagem é enviada, via socket, ao sistema embarcado da cadeira, que por meio do módulo DAC (*Digital-Analog Converter*), faz a conversão para sinais analógicos e envia ao controlador. Esse último, por sua vez, efetiva a ação ao atuar nos motores da cadeira e realizar a manobra.

A mensagem enviada ao sistema embarcado possui o formato JSON com a seguinte estrutura: `{“x”: [0-100%], “y”: [0-100%]}`.

Na mensagem, os percentuais indicam a proporção do sinal que será aplicado às direções x e y, e são combinados vetorialmente no controlador da CRM para aplicar o sinal de comando para frente, trás ou giro. Para este sistema, o percentual a ser aplicado ao selecionar cada comando, é de 100%. A velocidade máxima, que corresponde aos 100%, deve ser regulada diretamente no painel de configuração da cadeira de rodas.

O *software* apresenta como recursos críticos o vetor de amostras do ET e a variável que registra as notificações associadas aos eventos de transição entre estados, conforme ilustrado na Figura 25. Os recursos são acessados de forma concorrentes por *threads* ou contextos diferentes durante as operações. Para proteger as regiões críticas, foram usadas instâncias do objeto Lock, o que impede que mais de uma *thread* realize alterações no recurso simultaneamente ou que a leitura seja feita em paralelo com a escrita sobre a mesma região da memória.

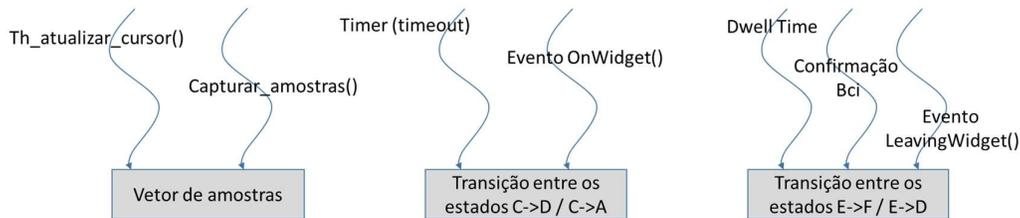


Figura 25 – Ilustração do acesso concorrente a recursos.

O vetor de amostras é acessado de forma concorrente pelo processo responsável por atualizar o cursor na tela e pelo processo que recebe as amostras do ET. A *thread* que atualiza o cursor usa o vetor para estimar a posição do olhar do usuário, seja com a média ou KF, conforme o estado do sistema. Enquanto isso, uma *thread* captura as amostras do

ET via socket e as insere no vetor: caso a quantidade de amostras presentes no vetor for menor que o padrão (100 amostras), novos valores ocupam as posições do vetor em fila; caso o vetor estiver completo, uma quantidade de novas amostras n é inserida ao final após a remoção das n amostras mais antigas. Entre uma execução e outra, as *threads* são suspensas para evitar que o mesmo processo ocupe o recurso de forma recorrente, impedindo que os outros também acessem.

O segundo caso ilustrado se refere ao acesso concorrente sobre a variável onde são percebidos os eventos de transição entre os estados do sistema. Isso pode ocorrer entre os estados C e D ou C e A, onde envolveria uma *thread* que realiza um *timer* e um evento assíncrono gerado ao entrar ou sair da região do componente central. Após o usuário desbloquear o acesso aos comandos e sair da região do componente central da GUI, é iniciado um contador de tempo, instância da classe *threading.Timer*, que pode retornar o sistema para o estado A ou avançar para o estado D, dependendo de ter sido selecionado ou não um dos comandos de manobra da cadeira dentro do tempo limite.

O terceiro item se refere aos eventos de cumprimento de *dwell time*, confirmação de EEG pelo BCI ou o evento gerado pela saída do cursor do interior do componente. Esses eventos poderiam alterar de forma concorrente a variável de transição. No estado E, espera-se que o usuário confirme a seleção do comando de manobra ou saia do componente associado, o que levaria o sistema ao estado D. Para confirmar o comando, o cursor deve ser mantido no componente até que ocorram as notificações de que foi cumprido o *dwell time* e recebida a sinalização via BCI. O BCI sinaliza o recebimento de EEG treinado quando a média da variável *power of signal* alcança o valor 0,7, o que ativa um evento que emula um clique do *mouse*. O clique do *mouse* sobre um componente da GUI, no estado correto (E), ativa um evento *onClick* que é capturado pelo sistema e confirma o comando a ser executado pela cadeira de rodas. Isso conduz a máquina de estados para o estado F, no qual uma função envia o comando e o percentual da escala para o sistema embarcado para que inicie o movimento escolhido.

Um último caso de acesso compartilhado a recurso é da variável *power of signal* da classe de interface ao BCI. Ela é necessária na decisão de comandos, mas também deve ser atualizada durante o recebimento de dados pela interface do BCI. O acesso também é controlado por um objeto *Lock*.

O tempo de permanência do olhar nos componentes (*dwell time*) usado nas transições de estados A-B e E-F, não foi estipulado de forma direta. A estratégia adotada foi perceber e aguardar que as amostras obtidas do ET convergissem para valores que,

com cerca de 95% de confiança, indicassem que o componente foi selecionado. Dentro do componente, a posição do cursor é calculada usando a média das amostras, o que causa um atraso de tempo durante a convergência dos valores estimados de x e y . Essa convergência é mais rápida quanto menor for a variância das amostras. Isso torna o sistema flexível ao tempo do usuário, que está relacionado à habilidade deste em realizar a movimentação do cursor pelo ET. Outro benefício do uso da média no cálculo da posição do cursor no interior dos componentes é que na ocasião de manter o cursor nessa região, os ruídos de valor ou pequenas variações na posição do olhar não causam desconforto ao usuário devido às oscilações do cursor em torno do ponto de olhar real.

Na classe que realiza a interface com a API do BCI, foi implementada a funcionalidade de treinamento offline. Para que seja possível a identificação do potencial EEG via BCI, necessário para a confirmação de comando no estado E, deve ser realizado um treinamento prévio da aplicação com os potenciais de EEG do usuário.

Para operar o BCI, deve estar instalada a API Emotiv, de onde são acessadas as funções necessárias. A versão baixada no site do fabricante e instalada foi o Emotiv-Installer-v2.1.0-x64. A API permite o acesso às *streams* de dados, funções e configurações, de acordo com a licença associada ao dispositivo. Além disso, deve ser gerado uma chave de acesso no site da Emotiv, a qual é usada para identificar e autorizar o acesso da aplicação ao BCI e à API.

Antes de usar o BCI, seja em modo de operação ou para realizar treinamento, devem ser executadas sequencialmente algumas funções da API, o que é mostrado na Figura 26. Esses passos foram agrupados em uma função chamada `preparar_interface_bci()`.

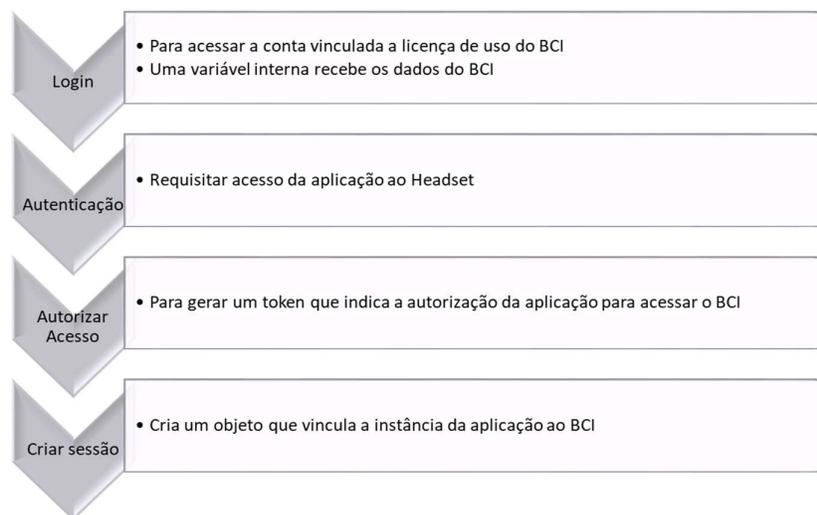


Figura 26 - Passos na preparação da interface com BCI.

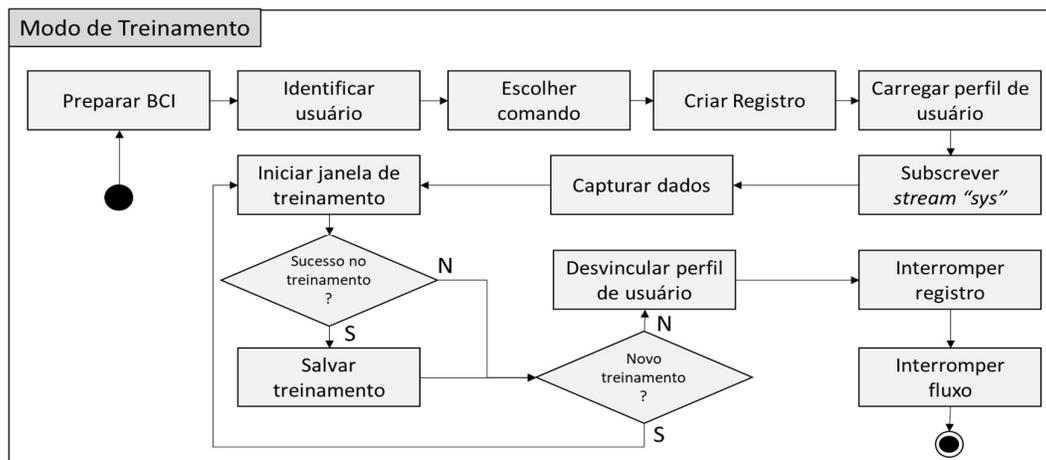
Após criar a sessão, a sequência de passos seguintes depende da operação a ser executada. Além do treinamento, modo de operação e outras funções de suporte, foi implementado uma sessão de captura livre para realizar testes unitários, com o BCI isoladamente, e assim verificar o fluxo de dados em tempo real, o que permite validar o correto funcionamento.

A API da BCI permite a escolha de quais dados serão retornados pela *stream*. Na função de captura de dados, informa-se um parâmetro que discrimina quais entre os 8 serão retornados. A Tabela 7 apresenta as *streams* de dados disponíveis.

Tabela 7 - Dados do BCI disponíveis via API Emotiv

Id da Stream	Descrição
“eeg”	Dados brutos de EEG.
“mot”	Dados gerados pelo acelerômetro do Headset.
“dev”	Informações de estado do gorro de eletrodos como bateria e qualidade dos contatos.
“pow”	Potência de bandas padrão – alfa, beta baixo, beta alto, gama, entre outras.
“met”	Dados da detecção de métricas de desempenho: stress, concentração, entre outros.
“com”	Dados da detecção de sinais de comandos mentais.
“fac”	Resultado da detecção de expressões faciais.
“sys”	Eventos de Sistema como carregar perfil ou eventos no processo de treinamento.

A *stream* “sys” foi usada no modo de treinamento, enquanto que a “com” no modo de operação e nas capturas livres. A Figura 27 apresenta o fluxo de operação da BCI completo, de acordo com o modo de atuação, se treinamento ou operação.



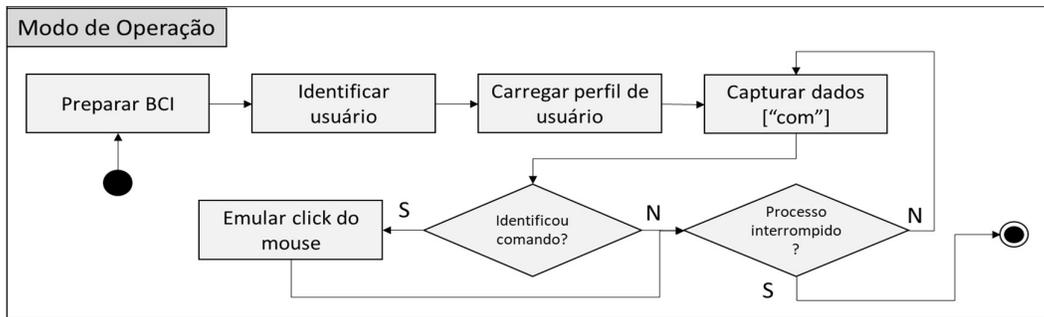


Figura 27 - Fluxograma dos modos de treinamento e operação.

A decisão sobre a identificação do comando treinado na BCI é realizada com o valor de 0,7 na média de *power of signal*, o que ativa o evento de clique do *mouse*. Um vetor armazena as últimas 5 amostras, então, o recebimento de mais valores leva ao descarte dos valores mais antigos. Porém, a média só leva em consideração amostras de comando recebidas consecutivamente, assim, o recebimento de sinal neutro limpa o vetor de amostras.

Além de gerar o clique ao identificar o comando, a interface BCI mantém atualizado o valor da variável *power of signal* para que este fique disponível para o processo de decisão de acionamento ou não do comando de manobra, que ocorre no Coordenador do sistema.

4.2 Integração e Adaptações na CRM

A integração entre a CRM e a parte do sistema que inclui a IHM ocorre via *software* em Python embarcado em uma Raspberry Pi 3. Os comandos são recebidos em pacotes JSON e convertidos para o formato apropriado conforme a Tabela 8.

Tabela 8 - Equivalência entre comandos recebidos, escala dos conversores e sinais de controle da cadeira.

Mensagem em JSON	Sinal de tensão (V)		Ação
	Porta X	Porta Y	
{"x": 0, "y": 0}	2,38	2,38	Parar
{"x": 100, "y": 0}	2,38	3,38	Ir para frente
{"x": -100, "y": 0}	2,38	1,38	Ira para trás
{"x": 0, "y": -100}	3,38	2,38	Giro no sentido horário
{"x": 0, "y": 100}	1,38	2,38	Giro no sentido antihorário
	Mínimo	Referência	Máximo
DAC (12 bits)	1182	2039	2998
ADC (16 bits)	11500	20000	28500
Tensão (V)	1,38	2,38	3,38

Para transformar os comandos recebidos da interface alternativa em movimentação da cadeira, são emuladas as saídas analógicas do *joystick*. Cada coordenada, x e y, do pacote JSON representa um percentual da tensão de operação, conforme equivalência mostrada na Tabela 8. Esses sinais são enviados à placa de controle da cadeira juntamente com a tensão de referência da escala. Essa estrutura fornece a escalabilidade do sistema para outras formas de entrada de comandos desde que mantidas as mensagens no formato JSON especificado.

A Figura 28 ilustra as interações entre os elementos de *hardware* chaves do sistema.

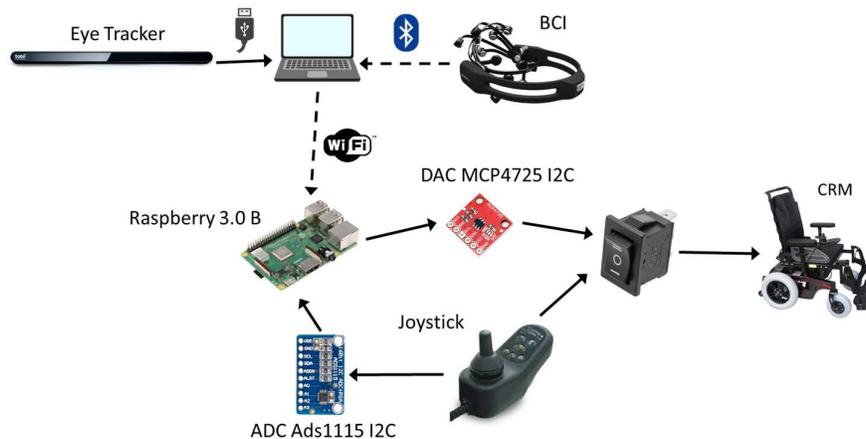


Figura 28 - Diagrama de interações entre os elementos de hardware chaves do sistema.

O sistema embarcado foi adicionado para condicionar o comando de alto nível, recebido na forma de pacote JSON, em sinais que possibilitem controlar o movimento da cadeira. Conversores analógico-digitais e digitais-analógicos são usados para adequar os sinais entre a Raspberry e a CRM, também entre o *joystick* e a Raspberry.

A cadeira de rodas motorizadas possui seu próprio sistema de controle e diagnóstico de *hardware*, que são compostos por dois módulos: o painel do *joystick*, que integra as funcionalidades de interação com o usuário (alavanca analógica, ajuste de velocidade, frenagem, entre outros), e controle de potência, que é responsável pelo gerenciamento dos motores, freios e baterias. Os módulos comunicam-se através de um protocolo proprietário baseado no CAN Bus.

A adaptação na cadeira motorizada resultou nos modos de operação manual, no qual o usuário controla a cadeira diretamente pela alavanca, e o digital, que utiliza a interface alternativa ET+BCI. Além disso, é possível controlar a cadeira de rodas pelo *joystick* sem a utilização do sistema embarcado, desde que selecionada a posição correta

em uma chave de três estados.

Uma das posições da chave de três estados é usada para conectar os sinais do *joystick* com o ADC, de onde é enviado a um barramento I2C da Raspberry (portas BCM 2 e 3). Com esse modo selecionado, os sinais enviados ao DAC, via *joystick* ou interface alternativa, são salvos em arquivos, o que permite a coleta de dados dos experimentos. Outra posição da chave faz com que os sinais do joystick sejam ligados diretamente ao controlador da cadeira conforme o modo original de funcionamento.

Em relação à alternância entre as duas modalidades de comando, esta acontece por meio de uma outra chave mecânica do tipo “*push button*” que, ao ser acionada, permite ao *software* realizar a mudança de um modo de operação para outro.

O *joystick* era conectado originalmente a terminais da placa de controle e diagnóstico da cadeira, por meio de um conector de 5 pinos, com as entradas VCC, Y, X, GND e REF. Os sinais de VCC e GND são fornecidos pela própria placa, enquanto a alavanca retornava os sinais de referência (REF) e sinais das posições de deslocamento X e Y. Com a adaptação realizada, mostrada nas Figuras 29 e 30, a cadeira pode ser comandada tanto por sinais do *joystick* quanto da Raspberry com os DACs.

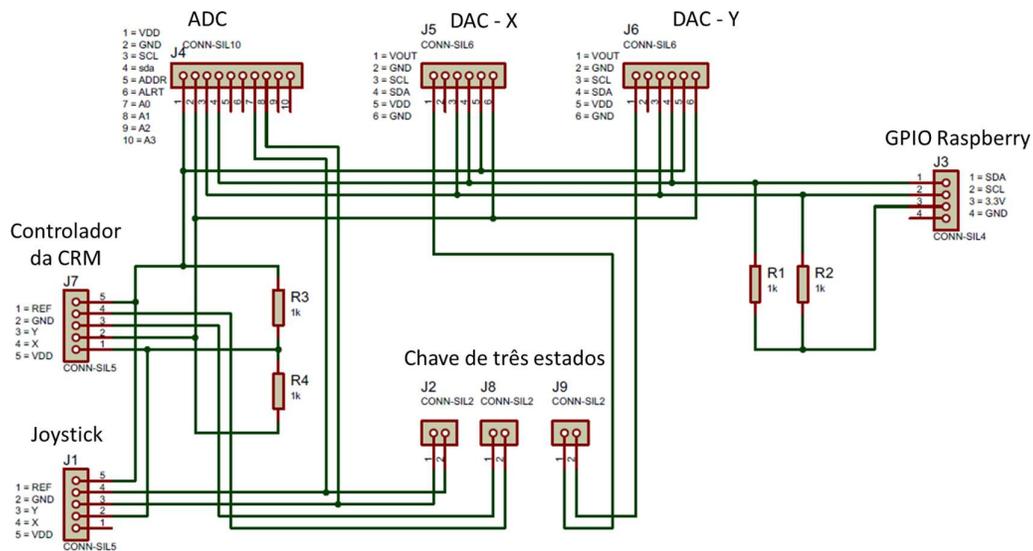


Figura 29 - Circuito para integração dos componentes do sistema embarcado à cadeira

O sinal de referência (REF) é o ponto médio entre VCC e GND. Em estado de repouso, os sinais de X e Y possuem o valor do sinal de referência em cerca de 2,38V, e isso determina que a cadeira deve ficar parada. O *joystick* possui um desgaste mecânico que deixa as coordenadas um pouco fora do referencial de repouso. Uma função foi criada para ajustar a referência inicialmente e reduzir a influência dessa distorção.

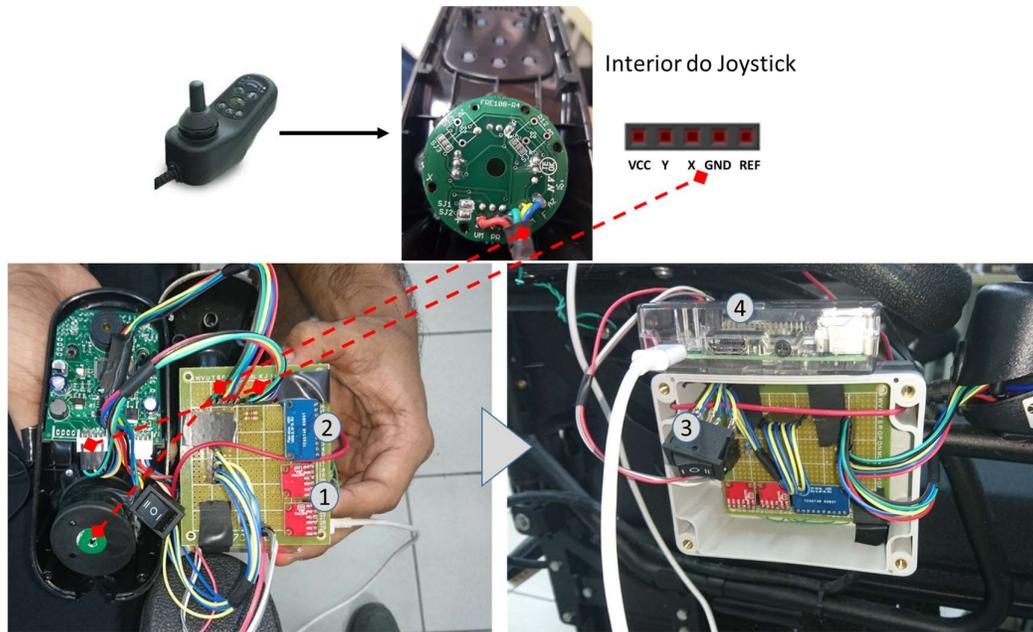


Figura 30 - Foto de partes montadas no sistema embarcado (1: Dois conversores D/A (Digital-Analógico); 2: Conversor A/D (Analogico/Digital); 3: Chave de três estados; 4: Raspberry)

Na Raspberry se encontram os programas para gerenciar a conexão com a IHM, converter os pacotes de comando JSON, emitir o comando ao conversor digital-analógico, para controlar a cadeira, mais as funções que salvam em arquivo os dados relativos às manobras executadas. O sistema operacional instalado no Raspberry é o Raspbian baseado no Debian Buster (10), com Kernel 4.19. Já a versão do Python usada no desenvolvimento foi a 3.

A conexão entre o módulo do programa IHM com o sistema embarcado é feito por *socket*, no qual os dados trafegam serializados em bytes. Para evitar que a leitura do *buffer* de entrada do *socket* não resulte em um pacote JSON incompleto, foi criado uma variável de onde os bytes são reunidos para recuperar os pacotes JSON ao padrão e descartar aqueles irrecuperáveis por perda de pacotes. Também foi usada a linguagem Python para implementar os *scripts* do sistema embarcado.

Os programas responsáveis por extrair o comando dos pacotes JSON e executá-los na cadeira atuam de forma assíncrona usando a estratégia de produtor-consumidor. Uma *thread* extrai os dados do *socket* e, ao completar a captura de um pacote completo e converter para o formato adequado, insere em um vetor de comandos a executar. Outra verifica se há comandos a executar e, caso sim, os executa. Se não há, é enviado um comando que mantém a cadeira parada. O tratamento do acesso simultâneo ao vetor de comandos é realizado por estrutura análoga ao Mutex, a qual é representada pela classe Lock no Python.

Para melhorar desempenho entre o tempo de recebimento de comando e a execução, a cada leitura do *buffer* de entrada são lidos 64 bytes dos 1024 de comprimento definido ao *socket*. O objeto Lock que controla o acesso ao *buffer* de controle durante a leitura da entrada é não-bloqueante, isto é, caso o Lock já tenha sido ativado por outro processo, o atual segue a execução. Nesse caso, uma variável temporária guarda a mensagem para posterior atualização do *buffer* de controle. O fluxo dos dados até a sinalização via porta I2C é ilustrado na Figura 31.

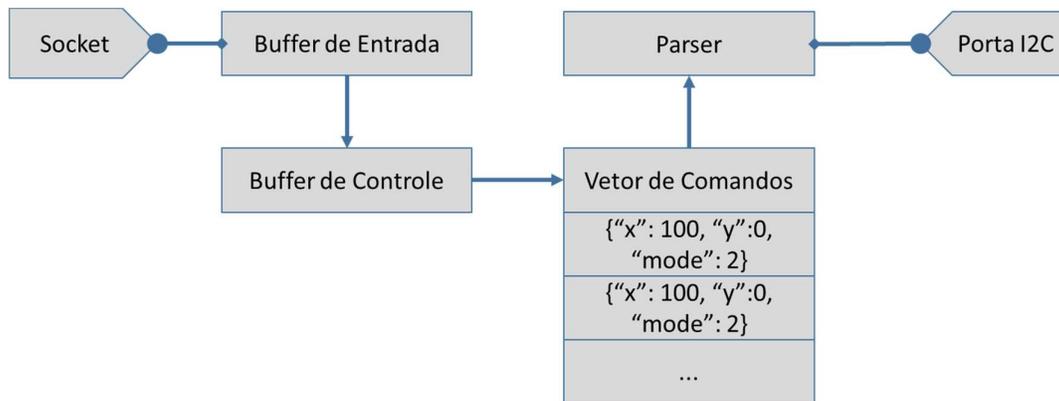


Figura 31 - Fluxo dos dados entre o socket e as portas I2C

A alimentação da Raspberry é realizada por uma ligação a partir das baterias da cadeira de rodas. Uma fonte ajustável step down LM2596 (Figura 32) foi usada para regular e estabilizar à tensão necessária para alimentar o sistema.



Figura 32 - Fonte step down LM2596

A comunicação entre os conversores A/D e D/A com a Raspberry acontece via portas I2C, com o primeiro cumprindo o papel o mestre do barramento e os conversores como escravos. Foram adaptados códigos da documentação oficial dos dispositivos conversores para classes que são instanciadas em uma terceira classe que atua como interface entre a IHM e a cadeira. Um objeto Lock protege o barramento I2C de escrita

simultânea por mais de um dispositivo. A integração entre A/D e D/A com a Raspberry é ilustrada na Figura 33.

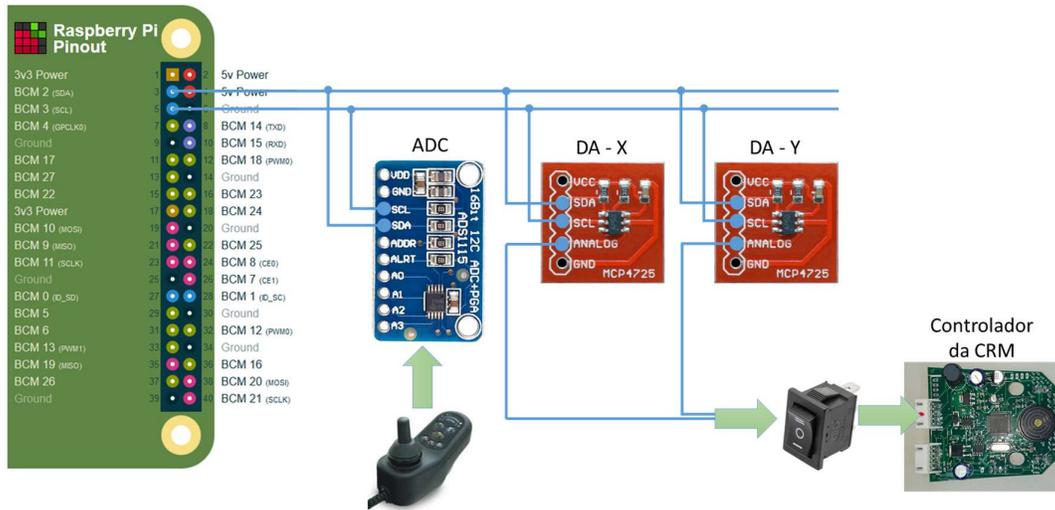


Figura 33 - Comunicação entre conversores e Raspberry via I2C

Os comandos que são recebidos pelo sistema embarcado são lidos do *socket*, transferidos a um *buffer* de controle, para a seguir serem transferidos para uma pilha de comandos a executar. Da pilha, os sinais que representam esses comandos são escritos no barramento I2C, endereço do conversor A/D, para emissão ao controlador da cadeira. O controlador mantém o último sinal recebido até a chegada de um novo. Em condição de repouso, a IHM envia sinais de parada para que o sistema embarcado pare ou mantenha a cadeira parada. A Figura 34 apresenta a associação entre as classes que compõem o sistema embarcado.

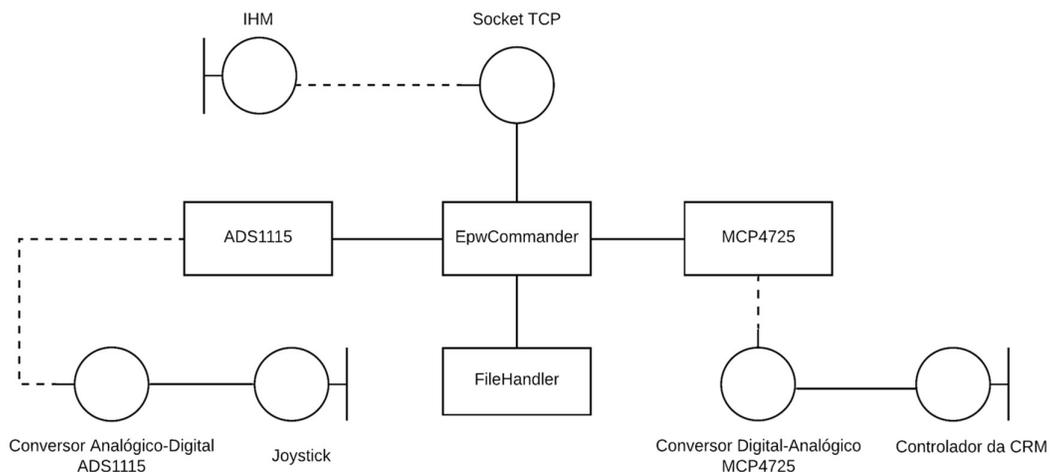


Figura 34 - Diagrama de classes do sistema embarcado da CRM

Enquanto a classe `EpwCommander` é responsável por coordenar as operações, internamente, as demais classes realizam seus papéis de acessar o barramento I2C, enviar sinais de controle e salvar os arquivos de dados capturados.

A versão final da CRM com o sistema embarcado é exibida na Figura 35. A placa que organiza as conexões entre o *joystick*, Raspberry, conversores e chaves, foi fixada no interior de uma caixa plástica.



Figura 35 - Cadeira de rodas motorizada com sistema embarcado

Em suma, a adaptação da CRM permite três formas de operação, das quais duas delas são manuais. A chave de três estados permite alternar entre enviar os sinais do *joystick* diretamente para a cadeira ou interfacear a operação pela Raspberry. No segundo caso, o controle da cadeira pode ser feito tanto com o *joystick* quanto pela entrada alternativa ET+BCI, ambos com registro dos comandos em arquivo de dados.

Capítulo 5 – Resultados experimentais

A avaliação do sistema ocorreu pela comparação dos percursos realizados pelas interfaces (*joystick* x interface ET/BCI). Essa comparação usou como métricas o tempo de realização de percurso e a diferença entre os sinais aplicados à CRM para realizar as operações nas diferentes opções de entrada.

- Descrição do experimento
 - a) Participaram 5 pessoas sem deficiência física e sem experiência prévia no uso de *eye tracker* ou BCI.
 - b) Cada pessoa realizou o percurso uma vez usando o *joystick* e três vezes com a interface alternativa.
 - c) São registrados o tempo e os sinais de controle emitidos para a CRM na realização de cada percurso.
 - d) A calibração do ET é realizada na aplicação do fabricante.
 - e) O treinamento dos potenciais de EEG é realizado na aplicação implementada.
 - f) Antes de cada experimento, o sistema é preparado ao usuário específico, portanto é calibrado o ET e treinados os potenciais de EEG.
 - g) O treinamento do sinal neutro no EEG é realizado sem estímulos ao usuário.
 - h) O sinal de comando a ser treinado utiliza as sobrancelhas elevadas como estímulo endógeno para o EEG.
 - i) No caso de dificuldade do usuário em manter o estímulo padrão, é aplicado outro que seja verificado como viável.
 - j) A BCI é usada com a qualidade geral dos contatos dos sensores maior que 80%. A solução salina deve ser reposta conforme a necessidade.
 - k) O *dispositivo* de EEG será usado na posição usual, com maior sensoriamento do cortex motor.
 - l) O trajeto é realizado conforme apresentado em Figura 36, o qual será indicado por fita no chão.

3,15m

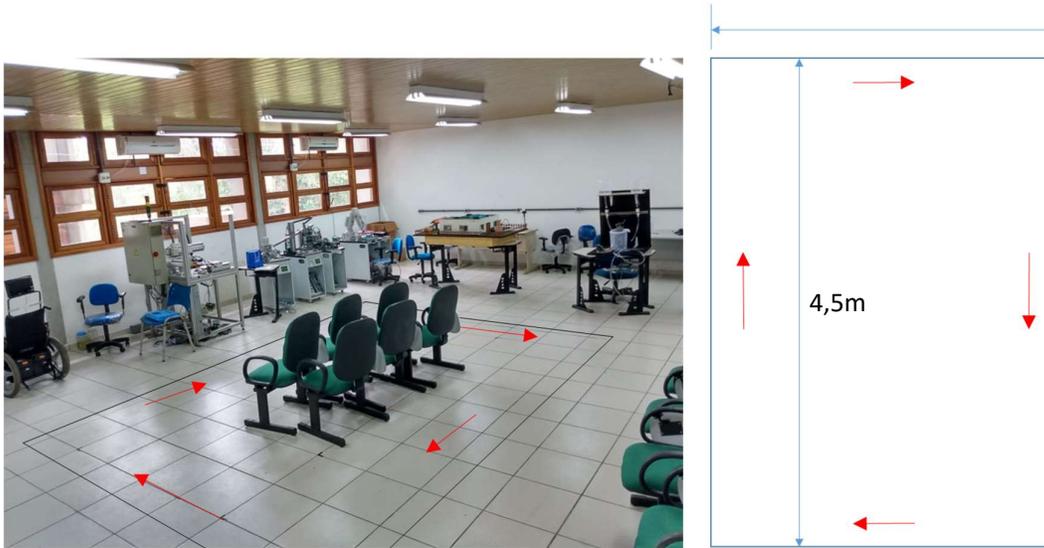


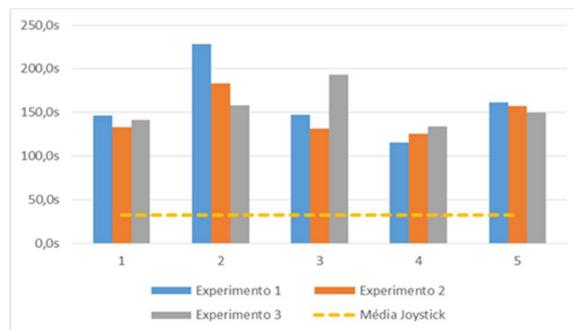
Figura 36 - Percurso realizado durante experimentos

- Experimentos

Na primeira parte do experimento, cada pessoa realizou o percurso uma vez usando o *joystick* como interface de entrada. A seguir, o mesmo percurso foi feito três vezes utilizando o *eye tracker* e o BCI combinados. A Tabela 9 apresenta um resumo dos dados obtidos.

Tabela 9 - Resumo dos dados obtidos na experimentação

	Joystick	Eye Tracker + BCI			
		1	2	3	Média
A	25,3s	146,2s	133,1s	141,7s	140,3s
B	56,2s	228,5s	183,2s	157,8s	189,8s
C	21,2s	147,3s	131,5s	193,0s	157,3s
D	33,4s	115,3s	125,7s	134,1s	125,0s
E	25,7s	161,4s	157,4s	149,6s	156,1s
Média	32,4s				153,7s



Como pode ser visto na tabela, o maior tempo para a realização do percurso com o *joystick* foi de 56,2s enquanto com ET+BCI chegou a 3min e 48s. Em uma volta de demonstração, chegou-se a quase 5 minutos o percurso com a interface alternativa.

Dentre as dificuldades para controlar a cadeira de rodas, teve destaque: a percepção do quanto manter acionado o comando para executar o movimento. O *joystick* responde mais rapidamente e a ação é quase instantânea. Um leve atraso entre o

acionamento e o movimento prejudicava a percepção e fazia com que os movimentos ultrapassassem a posição desejada.

Outro fator que interferiu negativamente foi o fato de que a versão Linux da Raspberry não foi preparada para garantir o tempo real. Para a continuidade do movimento, os comandos são reenviados a cada 100ms, mesmo tempo em que os pacotes JSON deviam ser consumidos na execução pelo sistema embarcado. Quando um comando é executado, este é mantido no registrador até a chegada do próximo. Com o atraso no processo, o comando era executado por um pouco mais de tempo do que deveria. Esse problema foi tratado de forma limitada ao reduzir o tempo de espera entre executar um comando e outro, mas uma solução mais robusta deve considerar um gerenciador mais complexo para lista de comandos ou a adaptação do escalonador de processos do sistema operacional, o que melhoraria a precisão da função de espera e aumentaria a garantia de tempo real.

Quanto à usabilidade da interface de entrada, não houve muita dificuldade. As pessoas conseguiram se acostumar ao uso e não houve acionamentos não intencionais. O *dwell time* dinâmico se comportou de maneira adequada, reduzindo conforme os usuários ficavam mais hábeis na interação. Todos os usuários conseguiram interagir com a BCI sem muito esforço, considerando adequado o tempo no qual obteve a resposta ao estímulo. Porém, foi relatado uma dificuldade na percepção do quanto acionar um comando na interface gráfica. Isso pode ser resultado do pouco costume com sistema ou pode indicar ser necessário mais estudos sobre como melhorar essa percepção e sensibilidade na GUI para tornar a execução dos percursos mais fluida.

O trabalho contribuiu com um método de fusão que considera a confiabilidade dos eventos e a interação série-paralela entre eles na decisão de ativação dos comandos. Outro destaque se refere ao posicionamento do cursor na GUI, que se adapta de forma natural à dinâmica própria do usuário – se é mais rápido ou devagar – e facilita o controle do cursor nos momentos mais críticos. Enquanto na literatura se observa um compromisso (*trade-offs*) na escolha de um tempo de permanência (*dwell time*), a abordagem implementada propiciou um comportamento mais dinâmico e adaptivo à desenvoltura com a qual o usuário interage com o cursor. Nesse caso, a definição do tempo de permanência ocorre indiretamente por meio da escolha do desvio padrão, ou seja, conforme o grau de confiança estatística a ser proporcionada na estimação da posição do olhar e o consequente tempo de convergência durante o cálculo.

Capítulo 6 – Conclusão

Este trabalho concebeu, baseado em estudos da literatura, um método alternativo para controlar localmente uma cadeira de rodas motorizada, onde informações de um *eye tracker* e uma BCI foram combinadas por uma técnica de fusão de alto nível. As interfaces de entrada atuam de forma cooperativa para confirmar as ações de seleção de comandos. Aplicar as interfaces combinadas reduz as probalidades de erros a valores inferiores que o uso das interfaces de forma individual, o que justifica o uso de um processo de seleção de comandos mais complexo.

A estrutura do sistema foi baseada em arquiteturas MVC e cliente-servidor, com a aplicação de orientação a objetos na modelagem e implementação dos componentes de *software*. O comportamento do sistema é descrito por uma máquina de estados e o ET possui a dominância na interface de entrada, mas podem ser integradas outras modalidades de entrada de comando tipo chaveado como interface complementar. No caso desta dissertação de mestrado, é utilizada a BCI emulando cliques do *mouse* ao identificar o recebimento de comando previamente treinado pelo usuário.

Cada interface de entrada possui sua própria API, as quais são disponibilizadas pelos respectivos fabricantes. No caso do *eye tracker*, são reunidas as amostras da posição do olhar, as quais foram utilizadas para estimar a posição. No caso da BCI, as funções disponibilizadas devem ser chamadas na ordem especificada na documentação para a correta execução do treinamento e monitoramento dos sinais de EEG.

A interface de entrada é destinada a pessoas com deficiências nos membros superiores e inferiores, mas como o usuário é responsável pelo sensoramento e manobra, ele não deve ter outras restrições que não o permita perceber corretamente o movimento da cadeira. A continuidade dos movimentos acontece com a manutenção da seleção de comando.

A seleção de comandos utiliza duas técnicas comuns nesse tipo de aplicação com ET: uma interface extra de entrada (multimodalidade) e a permanência do olhar durante um período de tempo na opção de comando. Essas são abordagens que permitem contornar a volatilidade na posição do olhar e evitar o problema conhecido como “Toque de Midas”, no qual o usuário aciona comandos de maneira involutária. A aplicação de estados do sistema e a necessidade de desbloqueio para selecionar um comando de movimento, forçam uma sequência de passos que também ajudam a contornar o

problema.

A sequência de operação ocorre de forma a contornar o problema de comandos não-intencionais, onde os comandos de manobra passam por etapas de seleção e confirmação controlados via máquina de estado. Com poucos passos para executar os comandos e interações via BCI pontuais, o sistema permite a interação do usuário de forma prática e com confiança que pode atingir até 95%, que é o valor definido na avaliação estatística das amostras do ET. Apesar da interação entre as interfaces reduzir a probabilidade conjunta na avaliação dos comandos, também pode ser verificado que os erros também são reduzidos. A fusão de informações obtidas do *eye tracker* e BCI garante que, estatisticamente, os erros na estimação dos comandos são menores do que seria com cada interface considerada individualmente, proporcionando uma operação mais segura da cadeira de rodas. O fato da interação com o sistema estar mapeado em estados bem definidos também contribui para a segurança.

A facilidade de uso e a segurança são fatores que tendem a contribuir para satisfação do usuário e, assim, motivar maior adesão e engajamento em relação a esse tipo de tecnologia assistiva que ainda é pouco difundida no Brasil. A sequência de estados bem definida e o *dwell time* dinâmico contribuíram para proporcionar tais fatores de forma equilibrada, sem que o aumento de um reduzisse fortemente o outro. Porém, o sistema embarcado precisa ser mais trabalhado para garantir o consumo dos comandos em tempo real. Durante os experimentos, não houve problemas em seguir o passo-a-passo para executar os comandos, mas houve um pouco de dificuldade na percepção de quanto acionar um comando para atingir determinada posição.

Os dispositivos BCI e ET apresentam algumas limitações. A API da BCI, na sua nova versão, depende da disponibilidade de internet para validar a licença e autorizar o acesso. No caso do *eye tracker*, a API é específica para plataforma *Microsoft Windows*.

Para os trabalhos futuros, podem ser realizados testes aplicando a interface desenvolvida em cenários de teleoperação, e assim para verificar a adequabilidade também para esse tipo de cenário. Além disso, poderia ser comparado o desempenho do sistema ao utilizar outros padrões de EEG no processo de decisão.

A preparação do sistema para utilização depende da calibração do *eye tracker* em *software* próprio do fabricante, o que enseja um estudo sobre calibração durante o uso ou mesmo a criação de um complemento que permita a calibração no próprio sistema, inclusive combinado com suporte da BCI. Também relacionado ao rastreamento ocular, outro potencial tópico para trabalho futuro é a detecção e remoção de artefatos (como a

piscada de olhos) nos dados obtidos do *eye tracker* utilizado, para que seja minimizada a influência deles em momentos inoportunos (durante a captura de sinais).

A limitação quanto a necessidade de Internet pela BCI pode ser superada com o uso de outros programas de acesso aos dados interface. Uma opção, poderia ser o OpenVibe, um software de uma empresa francesa que possui compatibilidade com alguns dispositivos desse tipo, inclusive com uma versão mais antiga do mesmo fabricante. Pode-se desenvolver um complemento que torne possível acessar os dados da BCI com o OpenVibe.

Para futuros trabalhos no grupo da UFAM de tecnologias assistivas, o sistema embarcado na cadeira de rodas pode ser mais explorado para garantir o tempo real de execução dos comandos. Aplicações com microcontroladores ou adaptação no escalonamento de processos do sistema operacional são opções viáveis.

O desenvolvimento de mais soluções cada vez mais apropriadas depende de mais informações quanto as necessidades dos potenciais usuários. Assim, pesquisas futuras podem explorar a aplicação deste sistema em cenário real, com usuários que precisam de soluções como esta. Isso contribuiria para refinar os requisitos em trabalhos futuros.

Referências Bibliográficas

- [1] J. Leaman and H. M. La, “A Comprehensive Review of Smart Wheelchairs : Past , Present , and Future,” *IEEE Trans. Human-Machine Syst.*, vol. 47, no. 4, pp. 486–499, 2017.
- [2] M. K. Shahin, Alaa Tharwat, T. Gaber, and A. E. Hassanien, “A Wheelchair control System using Human-Machine Interaction: Single-Modal and Multimodal Approaches,” *J. Intell. Syst.*, no. October, pp. 1–18, 2017.
- [3] S. Bastos Filho, T. F.; Kumar, Dinesh Kant ; Arjunan, *Devices for Mobility and Manipulation for People with Reduced Abilities (Rehabilitation Science in Practice Series)*, 1st ed. CRC Press, 2014.
- [4] SNPD, “Cartila do Censo 2010 - Pessoas com Deficiências,” 2012.
- [5] Organização Mundial da Saúde (OMS), “Relatório Mundial sobre Deficiência,” 2011.
- [6] F. de Amparo à Pesquisa do Estado de São Paulo (FAPESP), “Cadeira de rodas é controlada por expressões faciais,” 2016. [Online]. Available: <http://agencia.fapesp.br/cadeira-de-rodas-e-controlada-por-expressoes-faciais/23139/>. [Accessed: 23-Jul-2018].
- [7] R. Reação, “ESPECIAL – CADEIRAS DE RODAS – O novo momento das Cadeiras de Rodas no Brasil!,” 2018. [Online]. Available: <https://revistareacao.com.br/especial-cadeiras-de-rodas-o-novo-momento-das-cadeiras-de-rodas-no-brasil/>. [Accessed: 24-Jul-2018].
- [8] M. H. Al Sibai and S. Abdul Manap, “A Study on Smart Wheelchair Systems,” *Int. J. Eng. Technol. Sci.*, vol. 4, no. 1, pp. 25–35, 2015.
- [9] R. C. Simpson, “Smart wheelchairs: A literature review,” *J. Rehabil. Res. Dev.*, vol. 42, no. 4, pp. 423–436, 2005.
- [10] A. Al-Rahayfeh and M. Faezipour, “Eye Tracking and Head Movement Detection: A State-of-Art Survey,” *IEEE J. Transl. Eng. Heal. Med.*, vol. 1, no. July, pp. 2100212–2100212, 2013.

- [11] Y. K. Meena, H. Cecotti, K. Wong-Lin, and G. Prasad, "A multimodal interface to resolve the Midas-Touch problem in gaze controlled wheelchair," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 905–908, 2017.
- [12] A. A. Fedorova, S. L. Shishkin, Y. O. Nuzhdin, and B. M. Velichkovsky, "Gaze based robot control: The communicative approach," *Int. IEEE/EMBS Conf. Neural Eng. NER*, vol. July, pp. 751–754, 2015.
- [13] Fernández-Rodríguez, F. Velasco-Álvarez, and R. Ron-Angevin, "Review of real brain-controlled wheelchairs," *J. Neural Eng.*, vol. 13, no. 6, p. 61001, 2016.
- [14] A. Y. Kaplan, S. L. Shishkin, I. P. Ganin, I. A. Basyul, and A. Y. Zhigalov, "Adapting the P300-Based Brain – Computer Interface for Gaming : A Review," *IEEE Trans. Comput. Intell. AI Games*, vol. 5, no. 2, pp. 141–149, 2013.
- [15] S. Kalagi, J. Machado, V. Carvalho, F. Soares, and D. Matos, "Brain computer interface systems using non-invasive electroencephalogram signal: A literature review," *2017 Int. Conf. Eng. Technol. Innov.*, pp. 1578–1583, 2017.
- [16] F. Alam, R. Mehmood, I. Katib, N. N. Albogami, and A. Albeshri, "Data Fusion and IoT for Smart Ubiquitous Environments: A Survey," *IEEE Access*, vol. 5, pp. 9533–9554, 2017.
- [17] S. Ben Ayed, H. Trichilli, and A. M. Alimi, "Data fusion architectures: A survey and comparison," *Intell. Syst. Des. Appl.*, pp. 277–282, 2015.
- [18] F. Castanedo, "A review of data fusion techniques," *Sci. World J.*, vol. 2013, no. Article ID 704504, 19 pages, 2013.
- [19] Jaime Esteban, Andrew Starr, Robert Willetts, Paul Hannah, and Peter Bryanston-Cross, "A Review of Data Fusion Models and Architectures: towards Engineering Guidelines," *Neural Comput. Appl.*, vol. 14, no. 4, pp. 1–27, 2005.
- [20] R. Hyder, S. S. Chowdhury, and S. A. Fattah, "Real-Time non-intrusive eye-gaze tracking based wheelchair control for the physically challenged," *IECBES 2016 - IEEE-EMBS Conf. Biomed. Eng. Sci.*, pp. 784–787, 2017.
- [21] F. de A. A. Bastos, *Estatística e Probabilidade*, Versão 3.0. Fortaleza - CE: Editora da Universidade Federal do Ceará, 2015.

- [22] A. B. Downey, *Think DSP - Digital Signal Processing in Python*, Version 0. Needham, Massachusetts: Green Tea Press, 2014.
- [23] S. Haykin, *Kalman filtering and neural networks*. John Wiley & Sons, 2004.
- [24] A. C. Lopes, G. Pires, and U. Nunes, “Assisted Navigation for a Brain-Actuated Intelligent Wheelchair,” *Rob. Auton. Syst.*, vol. 61, no. 3, pp. 245–258, 2013.
- [25] F. Ben Taher, N. Ben Amor, and M. Jallouli, “A multimodal wheelchair control system based on EEG signals and Eye tracking fusion,” *INISTA 2015 - 2015 Int. Symp. Innov. Intell. Syst. Appl. Proc.*, pp. 1–8, 2015.
- [26] L. Cao, J. Li, H. Ji, and C. Jiang, “A Hybrid Brain Computer Interface System based on the Neurophysiological Protocol and Brain-actuated switch for Wheelchair Control,” *J. Neurosci. Methods*, vol. 229, pp. 33–43, 2014.
- [27] P. F. Diez *et al.*, “Commanding a robotic wheelchair with a high-frequency steady-state visual evoked potential based brain-computer interface,” *Med. Eng. Phys.*, vol. 35, no. 8, pp. 1155–1164, 2013.
- [28] Y. Punsawad and Y. Wongsawat, “Hybrid SSVEP-motion visual stimulus based BCI system for intelligent wheelchair,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 7416–7419, 2013.
- [29] D. Puanhvan, S. Khemmachotikun, P. Wechakarn, B. Wijarn, and Y. Wongsawat, “Navigation-synchronized multimodal control wheelchair from brain to alternative assistive technologies for persons with severe disabilities,” *Cogn. Neurodyn.*, vol. 11, no. 2, pp. 117–134, 2017.
- [30] B. Huang, A. H. P. Lo, and B. E. Shi, “Integrating EEG Information Improves Performance of Gaze Based Cursor Control,” *Int. IEEE/EMBS Conf. Neural Eng. NER*, pp. 415–418, 2013.
- [31] S. M. T. Müller, T. F. Bastos, and M. S. Filho, “Proposal of a SSVEP-BCI to command a robotic wheelchair,” *J. Control. Autom. Electr. Syst.*, vol. 24, no. 1–2, pp. 97–105, 2013.
- [32] D. C. Cavalieri, “Dissertação de Mestrado - Interface Homem-Máquina para uma Cadeira de Rodas Baseada no Movimento Ocular e Piscadas de Olhos,” Universidade Federal do Espírito Santo, 2007.

- [33] F. R. Martins, “Simulador para Treinamento de Cadeirantes em Ambiente Virtual por Comandos Musculares e/ou Visuais,” Universidade Federal de Uberlândia, 2017.
- [34] Y. Silva, W. Simões, M. Teófilo, E. Naves, and V. Lucena, “Training environment for electric powered wheelchairs using teleoperation through a head mounted display,” *2018 IEEE Int. Conf. Consum. Electron. ICCE 2018*, vol. 2018-Janua, pp. 1–2, 2018.
- [35] E. H. M. Couto, “Teleoperação de Cadeira de Rodas Robótica com COntrolador de Atraso de Tempo,” Universidade Federal do Espírito Santo, 2017.
- [36] L. Bueno, “Interface cérebro-computador baseada em EEG utilizando Redes Neurais auto-organizadas,” Universidade Federal do Espírito Santo, 2017.
- [37] L. R. Borges, “Avaliação de Usabilidade de Comando Ocular de Cadeira de Rodas Motorizada,” Universidade Federal de Uberlândia, 2016.
- [38] E. Pereira Zambalde, “Ssvep-Based Bci With Visual Stimuli From Lcd Screen Applied for Wheelchair Control: Offline and Online Investigations,” Universidade Federal de Uberlândia, 2018.
- [39] D. Blasch, Erik; Bossé, Élio; Lambert, *High-level information fusion management and systems design*, 1st ed. Boston: Artech House, 2012.
- [40] A. Kar, S. Member, and P. Corcoran, “A Review and Analysis of Eye-Gaze Estimation Systems , Algorithms and Performance Evaluation Methods in Consumer Platforms,” *IEEE Access*, vol. 5, pp. 16495–16519, 2017.
- [41] S. I. Ktena, W. Abbott, and A. A. Faisal, “A virtual reality platform for safe evaluation and training of natural gaze-based wheelchair driving,” *Int. IEEE/EMBS Conf. Neural Eng. NER*, vol. July, pp. 236–239, 2015.

Publicações

Autoria (Relacionada à dissertação):

- Título: *Electric Powered Wheelchair Command by Information Fusion from Eye Tracking and BCI*.
Evento: 2019 IEEE International Conference on Consumer Electronics (ICCE).
Link: <https://ieeexplore.ieee.org/document/8662090>

Outras (coautoria):

- Título: *A Digital Twin Architecture Based on the Industrial Internet of Things Technologies*.
Evento: 2019 IEEE International Conference on Consumer Electronics (ICCE).
Link: <https://ieeexplore.ieee.org/document/8662081>

- Título: *MobiHealth: a System to Improve Medication Adherence in Hypertensive Patients*.
Publicação: *Procedia Computer Science 141 (2018)*, páginas 366–373.
Evento: *The 8th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2018)*.
Link: <https://www.sciencedirect.com/science/article/pii/S1877050918318386>