



Ivo André Craveiro Simões Baptista

Desenvolvimento de um jogo controlado  
através de potenciais EEG estacionários evocados visualmente

Fevereiro de 2015



UNIVERSIDADE DE COIMBRA



Desenvolvimento de um jogo controlado  
através de potenciais EEG estacionários evocados  
visualmente

Orientadores: Prof. Dr. Urbano José Carreira Nunes  
Dr. Gabriel Pereira Pires

Júri:

Prof. Dr. Fernando Manuel dos Santos Perdigão  
Prof. Dr. Paulo José Monteiro Peixoto  
Prof. Dr. Urbano José Carreira Nunes



Ivo André Craveiro Simões Baptista

Departamento de Engenharia Eletrotécnica e de Computadores

Faculdade de Ciências e Tecnologia, Universidade de Coimbra

Fevereiro, 2015



# Agradecimentos

Gostaria de agradecer aos meus orientadores, Professor Doutor Urbano Nunes e Professor Doutor Gabriel Pires, pela oportunidade de trabalhar nesta área, bem como todos os conselhos e ajuda fornecidos durante a execução do trabalho.

Ao Ricardo Parafita gostaria de agradecer o seu trabalho no desenvolvimento da versão original do jogo utilizado no trabalho desenvolvido.

Ao João Perdiz gostaria de agradecer pela ajuda fornecida nas várias fases do trabalho.

Gostaria de agradecer aos participantes nas experiências efetuadas, pelo tempo que disponibilizaram para a realização das mesmas.

Um especial agradecimento à minha família, pelo apoio incondicional fornecido nesta fase da minha formação académica.

Este trabalho teve o apoio do Programa Operacional Fatores de Competitividade (POFC) do QREN e da Fundação para a Ciência e Tecnologia (FCT) através dos projetos AMSHMI12: RECI/EEI-AUT/0181/2012 e ProjB-DAM: Centro-07-ST24-FEDER-002028.



# Resumo

As interfaces cérebro-computador (BCI) apresentam-se como uma alternativa promissora às interfaces homem-máquina convencionais. Estas permitem a interação entre um utilizador e um sistema artificial através da interpretação dos seus estados mentais. Os sistemas BCI baseados em eletroencefalografia tem especial interesse, visto que esta é um técnica não invasiva com uma alta resolução temporal, baixo custo e fácil aquisição. No entanto, esta técnica de aquisição de sinal apresenta um baixo rácio sinal ruído (SNR).

Neste trabalho, foi desenvolvido um jogo de computador controlado por um BCI. O controlo é feito recorrendo à deteção de potenciais estacionários evocados visualmente (*steady state visually evoked potentials* - SSVEP), os quais são evocados como resposta a um estímulo visual repetitivo. O jogo desenvolvido consiste numa nave a percorrer uma pista, onde aparecem obstáculos. O objetivo do utilizador é desviar a nave dos obstáculos selecionando o estímulo correto. Os comandos considerados são: CIMA, BAIXO, DIREITA e ESQUERDA. Cada comando tem um estímulo visual associado, com uma frequência distinta.

Com vista a detetar o comando que o utilizador pretende enviar ao jogo, foram implementados métodos de extração de características e classificação. Foram testados e comparados vários métodos de extração de características, nomeadamente, os métodos de Goertzel, Welch e *canonical-correlation analysis* (CCA). A classificação é feita recorrendo ao classificador baseado na análise discriminante linear de Bayes ou pela escolha da característica de maior amplitude.

Nos testes *offline* foram comparados todos os métodos de extração de características. Foram efetuados testes *online* com cinco participantes. Para cada participante foi realizada uma sessão de jogo para o classificador baseado na análise discriminante linear de Bayes e uma sessão de jogo para a escolha da característica de maior amplitude. O sistema *online* obteve uma taxa média de acerto na classificação de 89.5% com a escolha da característica de maior amplitude, e 91% com o classificador baseado na análise discriminante linear de Bayes. Foram efetuados testes comparativos entre estímulos de padrões alternados e estímulos de gráficos singulares, para quatro dos participantes nos testes online, concluindo que os estímulos de padrões alternados são a melhor escolha para o controlo do jogo.

**Palavras-chave:** eletroencefalografia (EEG), potenciais estáticos evocados visualmente (SSVEP), interface cérebro-computador (BCI), rácio sinal ruído (SNR), análise da correlação canónica (CCA), extração de características, classificação



# *Abstract*

Brain-computer interfaces (BCI) are a promising alternative to conventional human-machine interfaces. Those allow direct interaction between a user and an artificial system, through the interpretation of the user's mental states. BCI systems based on electroencephalography (EEG) are of special interest, since EEG is a noninvasive technique with high temporal resolution, low cost and easy acquisition. However, the EEG signal acquisition has low signal to noise ratio (SNR), as well as low spatial resolution.

In this work, a computer game controlled via BCI was developed. The control is done through the detection of steady state visually evoked potentials (SSVEP), which are evoked as a response to a repetitive visual stimulus. The game consists in a spaceship traveling a track, where obstacles appear. The player's objective is to avoid those obstacles by looking at the correct stimulus. The considered commands were the following: UP, DOWN, LEFT, RIGHT. Each command has an associated visual stimulus with a distinct frequency.

With the objective of detecting the user's desired command, feature extraction and classification methods were implemented. Several feature extraction methods were tested and compared, namely, Welch's method, Goertzel's method and canonical-correlation analysis (CCA). The classification is done via the Bayes linear discriminant analysis classifier or by choosing the feature with highest value.

In the online tests, the feature extraction methods were compared. Online tests with five participants were performed. For each participant, a gaming session was done for each classification algorithm. The online system attained a mean classification success rate of 89.5% with the choice of the feature with the highest value, and 91% with the Bayes linear discriminant analysis classifier. Comparative tests between single graphic stimuli and pattern reversal stimuli were performed for four of the participants in the online tests, concluding that pattern reversal stimuli are a better choice for controlling the game.

**Key words:** electroencephalography (EEG), steady state visually evoked potentials (SSVEP), brain computer interface (BCI), signal to noise ratio (SNR), canonical correlation analysis (CCA), feature extraction, classification



# Conteúdo

<b>Agradecimentos</b>	<b>i</b>
<b>Resumo</b>	<b>ii</b>
<b>Abstract</b>	<b>iv</b>
<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>Abreviaturas</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação e contexto . . . . .	1
1.2 Trabalho realizado e contribuições . . . . .	2
<b>2 Princípios neurofisiológicos e estado da arte</b>	<b>5</b>
2.1 Métodos de geração dos estímulos visuais . . . . .	7
2.2 Indicadores de desempenho de classificação em sistemas BCI . . . . .	8
2.3 Aplicações BCI . . . . .	9
2.4 Aplicações SSVEP . . . . .	10
<b>3 Métodos</b>	<b>15</b>
3.1 Extração de características . . . . .	15
3.1.1 Transformada de Fourier . . . . .	15
3.1.2 Método de estimação da densidade espectral de Welch . . . . .	16
3.1.3 Algoritmo de Goertzel . . . . .	18
3.1.4 Análise de correlação canónica . . . . .	19
3.2 Métodos de classificação . . . . .	22
3.2.1 Análise discriminante linear bayesiana . . . . .	23
<b>4 Setup experimental e conceção gráfica do jogo</b>	<b>25</b>
4.1 Aquisição de dados EEG . . . . .	25
4.2 Jogo Spacecraft . . . . .	26
4.2.1 OpenGL . . . . .	29
4.2.2 Estímulos visuais . . . . .	31
4.2.2.1 Temporizador . . . . .	34
4.2.3 Comunicação UDP . . . . .	36
4.2.4 Obstáculos . . . . .	37

<b>5</b>	<b>Classificação de SSVEP</b>	<b>39</b>
5.1	Protocolo experimental de treino . . . . .	39
5.2	Pré processamento . . . . .	39
5.3	Extração de características . . . . .	39
5.3.1	Método de Goertzel . . . . .	39
5.3.2	Método de Welch . . . . .	40
5.3.3	Método CCA . . . . .	41
5.4	Classificação . . . . .	41
5.5	Classificação <i>online</i> . . . . .	42
<b>6</b>	<b>Resultados experimentais</b>	<b>45</b>
6.1	Resultados de dados de treino . . . . .	45
6.1.1	Validação dos estímulos . . . . .	45
6.1.2	Resultados dos dados de treino . . . . .	45
6.2	Resultados online . . . . .	50
6.3	Resultados qualitativos dos estímulos visuais . . . . .	52
<b>7</b>	<b>Conclusões e trabalho futuro</b>	<b>53</b>
	<b>Bibliografia</b>	<b>55</b>

# Lista de Figuras

1.1	Exemplo de ecrã do jogo Spacecraft. . . . .	3
1.2	Diagrama de blocos dos principais módulos desenvolvidos. . . . .	3
2.1	Sistema Internacional 10-20 . . . . .	7
2.2	Exemplo de resposta SSVEP a um estímulo de 8 Hz. No domínio do tempo e frequência. . . . .	7
2.3	Exemplos de estímulos (retirado de [Zhu et al., 2010]) . . . . .	8
3.1	Exemplos de funções janela. . . . .	17
3.2	Ilustração do método CCA. . . . .	20
4.1	Arquitetura do sistema. . . . .	26
4.2	Diagrama de “simulink” usado na aquisição. . . . .	27
4.3	Ambiente de aquisição de dados. 1) gEEGCap com elétrodos montados. 2) Amplificador g.USBamp. 3) Ecrã a correr o jogo. 4) Computador que corre o jogos e faz processamento de dados. . . . .	27
4.4	Capturas de ecrã do jogos Spacecraft. 1) Taxa de refrescamento do jogo. 2) Estímulo de padrões alternados. 3) Obstáculo. 4) Nave espacial. 5) Pontuação e quantidade de colisões do jogador. . . . .	28
4.5	Módulos de software do jogo Spacecraft. . . . .	29
4.6	Organização das <i>threads</i> do jogo. . . . .	30
4.7	Exemplos de estímulos implementados. . . . .	31
4.8	Diagrama de classes para os estímulos. . . . .	32
4.9	Diagrama de atividades da <i>thread</i> dedicada a um estímulo. . . . .	33
4.10	Exemplos de estímulos, a 10 Hz (em cima) e 13 Hz (em baixo), apresentados num ecrã de 60 Hz . . . . .	35
4.11	Situação de colisão da nave com um obstáculo. . . . .	38
5.1	Ilustração do protocolo de treino adotado para os estímulos. . . . .	40
5.2	Esquema da aplicação do classificador com um conjunto de dados de treino e de teste. . . . .	43
5.3	Processo de classificação <i>online</i> . . . . .	43
6.1	Exemplos de respostas SSVEP a diferentes frequências. . . . .	46
6.2	Comparação da resposta a um estímulo de $6.667Hz$ e $0.75Hz$ a resoluções diferentes. Espectros obtidos pelo método de Welch. . . . .	47



# Lista de Tabelas

2.1	Aplicações BCI controladas com P300 e imaginação motora (por vezes combinados com SSVEP). . . . .	11
2.2	Aplicações BCI controladas por SSVEP. . . . .	13
6.1	Resultados (% de acerto) de diferentes métodos de extração de características. Foram considerados o primeiro e segundo harmônicos da frequência de estímulo. A classificação é feita recorrendo ao método de escolha da característica máxima. . . . .	48
6.2	Resultados (% de acerto) de diferentes métodos de extração de características. Para diferentes sujeitos de teste. Foram considerados o primeiro e segundo harmônicos da frequência de estímulo. A classificação é feita recorrendo ao método de escolha da característica máxima. . . . .	49
6.3	Comparação dos resultados (% de acerto) dos métodos de classificação (escolha de característica de maior amplitude e classificador linear de Bayes) e o número de harmônicos escolhidos como características, para 2 classes e 4 classes. A extração de características é feita com o algoritmo de Goertzel. . . . .	49
6.4	Comparação dos resultados (% de acerto) dos métodos de classificação (escolha de característica de maior amplitude e classificador linear de Bayes) e o número de harmônicos escolhidos como características, para 4 classes. A extração de características é feita com o algoritmo de Goertzel. . . . .	50
6.5	Comparação da taxa de sucesso para estímulos de padrões alternados e estímulos simples. A extração de características é feita pelo algoritmo de Goertzel. A classificação é feita pela escolha da característica de maior amplitude. . . . .	51
6.6	Taxa de sucesso para os testes online. A extração de características é feita pelo método de Goertzel. Nas sessões de treino é usada a escolha da característica de maior amplitude. Nas sessões de jogo é usado o classificador linear de Bayes. . . . .	51
6.7	Taxa de sucesso para os testes online. A extração de características é feita pelo método de CCA. Nas sessões de treino é usada a escolha da característica de maior amplitude. Nas sessões de jogo é usado o classificador linear de Bayes. . . . .	51



# Abreviaturas

API	Interface para programação de aplicações
BCI	Interface cérebro computador
CCA	Análise da correlação canónica
EEG	Eletroencefalografia
FFT	Fast Fourier Transform
ISR	Instituto de Sistemas e Robótica
ITR	Taxa de transferência de informação
PSD	Densidade espectral de potência
SNR	Rácio sinal ruído
SSVEP	Potenciais estacionários evocados visualmente



# Capítulo 1

## Introdução

### 1.1 Motivação e contexto

O estudo das interfaces cérebro computador (BCI) tem despertado um interesse crescente na comunidade científica. Estas apresentam-se como uma alternativa às interfaces homem-máquina habituais, pelo facto de permitirem a interação direta entre o cérebro e um sistema artificial. Logo permitem a geração de um canal de comunicação, bem como uma fonte de controlo.

Este tipo de interfaces apresenta-se como uma ferramenta promissora para pessoas com limitações físicas e motoras, devido à sua forma de interação independente da atividade muscular e pelo seu potencial em diversas aplicações em áreas da reabilitação motora e reabilitação cognitiva/comportamental. Em certos casos extremos de paralisia, o BCI torna-se a única modalidade de comunicação disponível, devido à impossibilidade do controlo dos músculos pelo utilizador. Logo pacientes de doenças como a esclerose lateral amiotrófica (ALS) e paralisia cerebral podem beneficiar imenso deste tipo de tecnologia, permitindo adquirir funções perdidas previamente, melhorando a sua qualidade de vida. Podem também ser usadas com fins de reabilitação, para recuperação de funções motoras devido a lesões cervicais. Além de todo o interesse que esta área apresenta para estes pacientes, estas interfaces também podem ser usadas por pessoas sem limitações motoras, em alternativa a outras interfaces, para controlar as mais diversas aplicações. Neste sentido, estas interfaces podem ser aplicadas com funções lúdicas e até mesmo educativas, permitindo o controlo de jogos e outras formas de entretenimento. A implementação destas interfaces na área do entretenimento também pode ajudar a motivar pacientes na utilização desta tecnologia, aliando a componente de diversão à tarefa de reabilitação.

Um BCI funciona através da interpretação de sinais cerebrais recolhidos de forma não invasiva ou invasiva. Logo esta é uma área que necessita de uma forte componente de processamento de sinal, estando dependente de áreas como *machine learning*, neurologia, entre outras.

Apesar de todo o interesse despertado por esta área, a integração destas interfaces no dia à dia ainda necessita de um maior investimento, em termos de investigação. A recolha de sinal electroencefalográfico (EEG) é a técnica mais utilizada nas interfaces BCI por ser não invasiva, de fácil aquisição e de baixo custo. No entanto, o sinal EEG é muito suscetível a ruído, devido a fontes externas, bem como fontes internas provenientes de processos cognitivos. Muitas aplicações necessitam de treino e calibração devido à variação dos sinais cerebrais de pessoa para pessoa, ou na mesma pessoa em diferentes sessões de utilização. Pessoas sem prática na utilização de BCI podem apresentar baixas taxas de sucesso, o que pode contribuir para a sua desmotivação no uso destas interfaces. Estes são alguns dos desafios a superar, com vista a disponibilizar estas interfaces a um público mais alargado.

Do sinal EEG pode ser obtida informação proveniente de vários neuromecanismos. A interface desenvolvida neste trabalho é baseada num neuromecanismo designado por potenciais estacionários evocados visualmente (*steady state visually evoked potentials* - SSVEP). Os SSVEP representam po-

tenciais evocados como resposta a estímulos visuais repetitivos.

### 1.2 Trabalho realizado e contribuições

O trabalho consistiu no desenvolvimento de um jogo controlado através de SSVEP, e que teve por base uma versão desenvolvida anteriormente (jogo "Spacecraft" [Parafita et al., 2013]). Esta versão anterior utiliza o neuromecanismo SSVEP com marcação de fase, o qual necessita de um período de calibração antes de cada sessão de jogo. Logo foi necessário alterar o neuromecanismo de controlo, com vista a utilizar SSVEP com codificação em frequência, o qual permite utilizar o sistema sem efetuar treino, ou efetuar o treino somente uma vez. Os estímulos visuais implementados na versão anterior não funcionavam como esperado para frequências superiores a 6Hz, para superar este problema foram implementados estímulos visuais com um temporizador de elevada resolução. Estes permitem uma maior gama de frequências, o que aumenta o número de comandos possíveis no jogo. O jogo desenvolvido consiste no controlo de uma nave, que viaja sobre uma pista. O utilizador tem que desviar a nave para esta não colidir com os obstáculos que aparecem na pista, com vista a receber pontuação, sendo penalizado por cada colisão. As opções de movimento da nave são cima, baixo, direita e esquerda, existindo um tipo de obstáculo correspondente a cada movimento. Estes movimentos são evocados por estímulos visuais repetitivos, presentes no jogo. A figura 1.1 apresenta o jogo desenvolvido.

O trabalho realizado e principais contribuições foram as seguintes (ver figura 1.2):

- Implementação e teste de algoritmos de extração de características e de classificação, implementados num módulo de processamento de sinal independente do jogo.
- Implementação de comunicação UDP/IP em matlab e C++, com vista a permitir a comunicação entre o jogo e o módulo de processamento de sinal.
- Alteração dos módulos de software do jogo "Spacecraft". Nomeadamente, implementação de estímulos visuais com frequência constante, temporização e sincronização do desenho com a taxa de refrescamento do monitor. Os estímulos visuais são configuráveis, permitindo estímulos de padrões alternados e estímulos de gráficos singulares. Foram utilizados quatro estímulos, posicionados de acordo com os quatro comandos a enviar.
- Testes de controlo *online* do jogo com cinco participantes.

No capítulo 2 é feita uma revisão da literatura sobre aplicações BCI e aplicações baseadas no neuromecanismo SSVEP, onde se dá particular importância a artigos que descrevem jogos. Os módulos do jogo, comunicação e aquisição de dados são descritos no capítulo 4. A implementação do processamento de dados é descrita no capítulo 5. Os resultados do trabalho efetuado são analisados no capítulo 6.

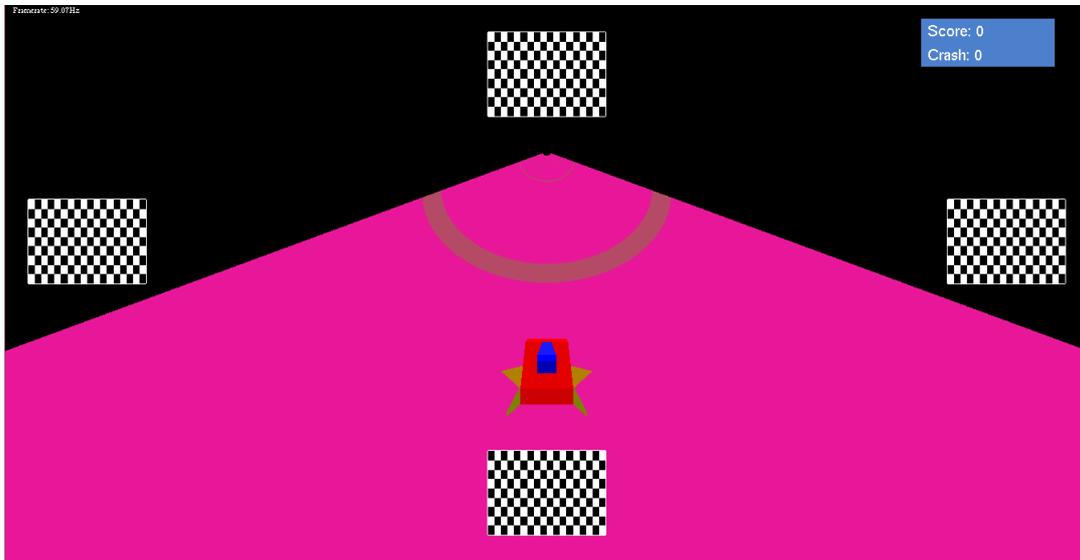


Figura 1.1: Exemplo de ecrã do jogo Spacecraft.

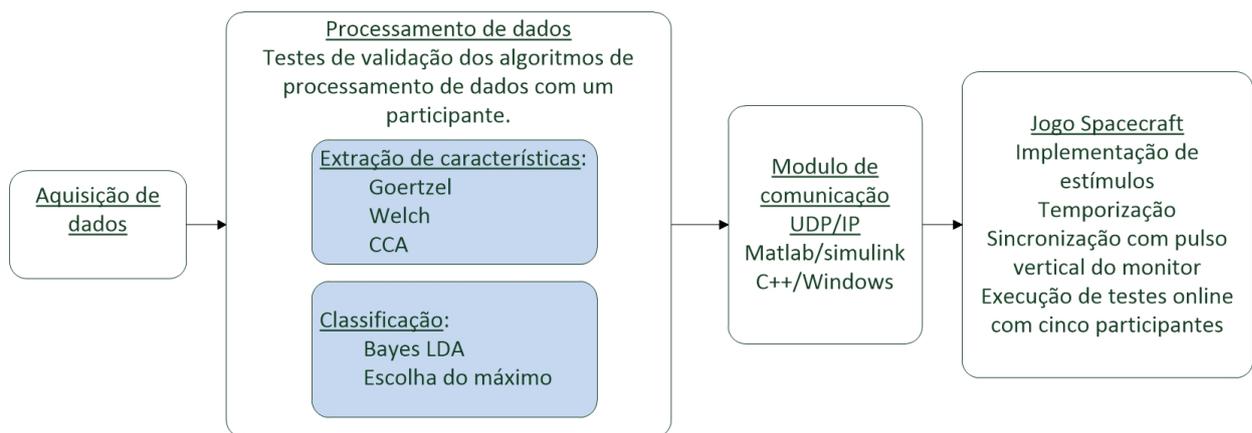


Figura 1.2: Diagrama de blocos dos principais módulos desenvolvidos.



## Capítulo 2

# Princípios neurofisiológicos e estado da arte

As interfaces cérebro computador (*brain-computer interfaces* - BCI) permitem a interação direta com sistemas, contornando os caminhos de saída habituais do cérebro, nervos periféricos e músculos [Wolpaw et al., 2002]. Isto é possível obtendo sinais cerebrais e interpretando as intenções dos utilizadores a partir dos mesmos. Este modo de interação permite o controlo de uma variada gama de aplicações, bem como um modo de comunicação. São particularmente úteis para pessoas com problemas motores.

Existem várias modalidades de aquisição de sinais cerebrais, podendo estas ser invasivas e não invasivas. A eletroencefalografia (EEG) é uma modalidade não invasiva, sendo popularmente utilizada devido à sua alta resolução temporal, custo-eficácia e facilidade de aquisição [Zhu et al., 2010]. Os sinais EEG medem a atividade elétrica na superfície do escalpe, provenientes dos processos neuro fisiológicos do cérebro, captada através de eléctrodos. Estes sinais apresentam um rácio sinal ruído reduzido (SNR), estando também sujeitos a artefactos de diversas fontes, bem como baixa resolução espacial. A cada sessão de aquisição, é necessário montar os eléctrodos no escalpe, sendo um processo moroso, visto que é necessário garantir uma boa condutividade, recorrendo a gel condutivo e limpeza do escalpe. Os eléctrodos a seco não necessitam de gel, mas fornecem um sinal de pior qualidade.

O posicionamento dos eléctrodos no escalpe é habitualmente definido pelo sistema internacional 10-20 (ver figura 2.1), que visa definir um padrão para a aquisição de sinais no escalpe, com o objetivo de obter consistência e comparabilidade nos dados adquiridos. Neste sistema, as distâncias entre eléctrodos são de 10% e 20% da distância entre a zona frontal e posterior, bem como entre lado direito e esquerdo do crânio. As letras indicam os lobos cerebrais englobadas pelas localizações dos eléctrodos, parietal (P), occipital (O), frontal (F), pré frontal (Fp) e temporal (T). A letra C indica pontos na linha central, entre o inião (Iz) e nasião (Nz). Estes pontos bem como a zona pré auricular esquerda (A1) e a direita (A2), são usados como referência na montagem dos eléctrodos. O nasião encontra-se na zona superior ao nariz, ao nível dos olhos. O inião situa-se na zona anterior do crânio, ligeiramente abaixo da zona occipital. As zonas pré-auriculares situam-se ao nível da orelha, do lado esquerdo e direito. A cada localização está associado um número; números ímpares indicam localizações no hemisfério esquerdo, já os números pares estão associados ao hemisfério direito, zero (z) está associado à linha central do cérebro, na linha que liga o nasião e inião, passando por Cz. Existem outros padrões para a localização dos eléctrodos, como por exemplo o sistema 10-5 e sistema 10-20 estendido, que apresentam uma maior resolução.

Diversos sinais de controlo podem ser extraídas do sinal EEG, provenientes de fenómenos presentes no sinal sob condições específicas. São detetadas características presentes no sinal, que indicam o estado mental do utilizador, o qual é convertido num sinal de controlo. O fenómeno responsável por estas características pode ser proveniente de um estímulo externo, ou proveniente de processos cognitivos independentes de estímulos. Interfaces baseadas em fenómenos dependentes de estímulos

---

são categorizadas como reativas (BCI exógena), enquanto que interfaces independentes de estímulos são categorizadas como ativas (BCI endógena) [Chae et al., 2012].

Os *event-related potentials* (ERP) são um neuromecanismo possível para aplicação em BCI. Estes são potenciais evocados por estímulos externos ou eventos. O P300, em particular, representa uma deflexão positiva no sinal EEG que ocorre entre os 200 e 700 ms, como resposta a um estímulo raro [Angeloni et al., 2012]. Devido à necessidade de estímulos externos, as interfaces baseadas no fenómeno P300 são reativas.

Os ritmos sensório motores (*sensorimotor rhythms* - SMR), correspondentes às bandas de frequência  $\mu$  (8 - 12 Hz) e  $\beta$  (18 - 24 Hz), são um possível sinal de controlo. Estes ritmos apresentam decréscimos de potência durante períodos de imaginação motora e aumentos durante momentos de relaxamento. Este fenómeno pode ser denominado de *event-related synchronization* nos períodos de maior potência e *event-related desynchronization* nos períodos de menor potência [Pires et al., 2011]. Visto que este fenómeno depende apenas de atividades cerebrais do utilizador, as interfaces criadas com base no mesmo são ativas.

O trabalho desenvolvido focou-se nos potenciais estacionários evocados visualmente (*steady-state visually evoked potentials* - SSVEP), os quais representam potenciais evocados como resposta a um estímulo visual repetitivo, a uma determinada frequência. Este fenómeno deve-se aos mecanismos neuro fisiológicos, que processam informação visual [Wang et al., 2006]. Ao processar um estímulo repetitivo, a ativação dos neurónios sincroniza com a frequência do mesmo [Nakanishi et al., 2014], gerando picos à frequência mencionada, bem como os seus harmónicos, no espectro do sinal EEG. A resposta SSVEP pode ser observada no sinal EEG sob o córtex visual, com maior amplitude na zona occipital. A figura 2.2 mostra a resposta a um estímulo visual repetitivo de 8 Hz. É possível observar picos no espectro do sinal à frequência alvo e segundo harmónico (16 Hz).

As interfaces que utilizam este fenómeno são reativas, visto que o sistema está dependente de estímulos visuais externos. A amplitude da resposta SSVEP a um estímulo visual é tanto maior quanto mais próximo este estiver do centro do campo de visão, este facto torna a interface BCI dependente de movimentos oculares necessários ao controlo do foco visual; sendo normalmente, por este motivo, considerada uma BCI dependente.

Codificação em frequência é uma técnica muito usada na implementação de interfaces recorrendo a SSVEP. Consiste em associar cada comando a um estímulo a piscar a uma determinada frequência. No sinal EEG do utilizador, é detetada a frequência dominante, a qual é convertida no comando desejado. Este método permite uma elevada taxa de transferência de informação (*information transfer rate* - ITR), bem como baixos tempos de treino. Devido a estas características, este método tem sido muito usado nos campos da neurociência visual e engenharia neuronal [Nakanishi et al., 2014]. A cada estímulo pode estar associada uma fase, que é posteriormente detetada, aumentando o número de comandos possíveis, sem a necessidade de aumentar o número de frequências utilizadas. A este método dá-se o nome de marcação de fase (*phase tagging*) [Parafita et al., 2013, Kuo-Kai Shyu et al., 2010].

Com vista à deteção da frequência dominante, o sinal EEG passa por várias fases de processamento. A fase de pré-processamento engloba a filtragem do sinal, onde são eliminadas as componentes do espectro desnecessárias para a deteção da frequência dominante. De seguida, são extraídas as características do sinal que permitem fazer a deteção. Nesta fase podem ser usadas técnicas de processamento de sinal, como por exemplo a transformada rápida de Fourier (*fast Fourier transform* - FFT) com vista a

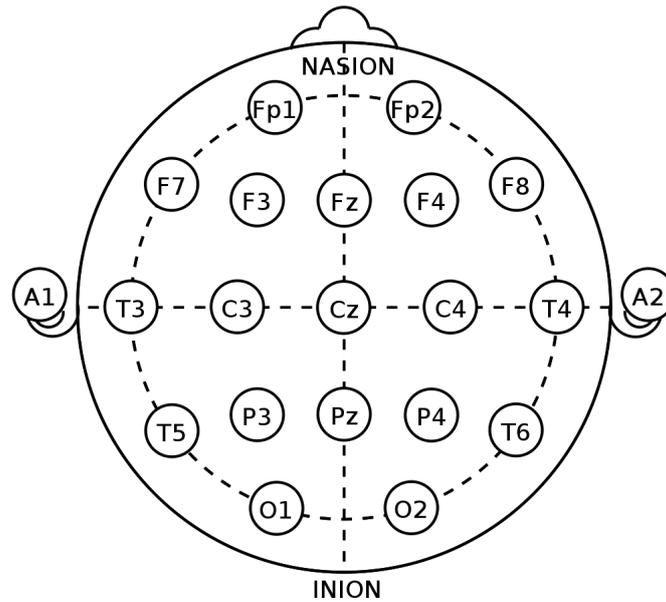


Figura 2.1: Sistema Internacional 10-20

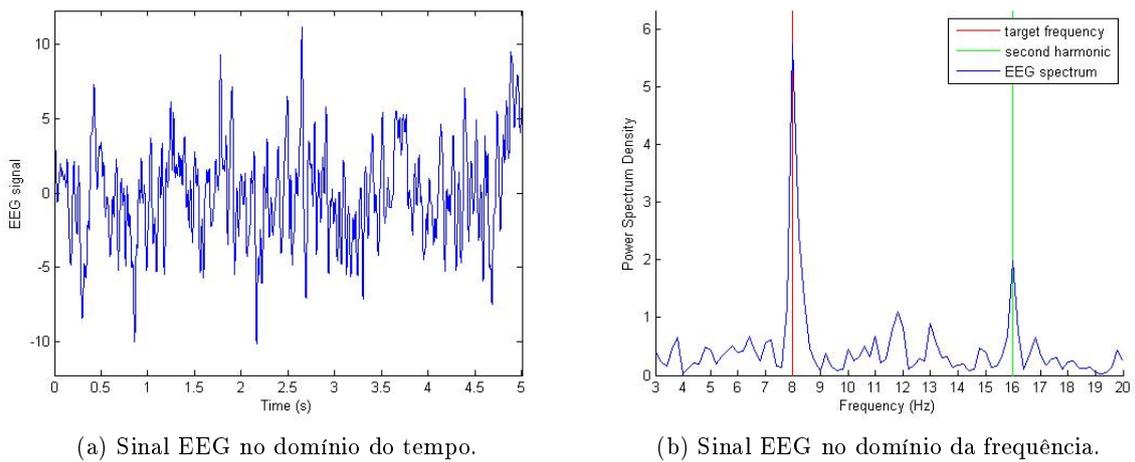


Figura 2.2: Exemplo de resposta SSVEP a um estímulo de 8 Hz. No domínio do tempo e frequência.

estimar a densidade espectral de potência do sinal EEG, ou a análise da correlação canônica (*canonical correlation analysis* - CCA). As características extraídas são usadas na classificação.

## 2.1 Métodos de geração dos estímulos visuais

Os estímulos são uma componente importante de um sistema BCI baseado em SSVEP. É necessário que estes evoquem uma resposta discriminável no sinal EEG do utilizador, quando este foca a sua atenção no estímulo visual em causa, de preferência sem causar desconforto ou fadiga. Para tal, é necessário cuidado na escolha das frequências a utilizar, bem como na apresentação dos estímulos. É importante garantir que não ocorrem erros na frequência dos estímulos, como os que resultam da taxa de refrescamento de monitores. Isto requer que, antes da sua utilização na interface, estes sejam

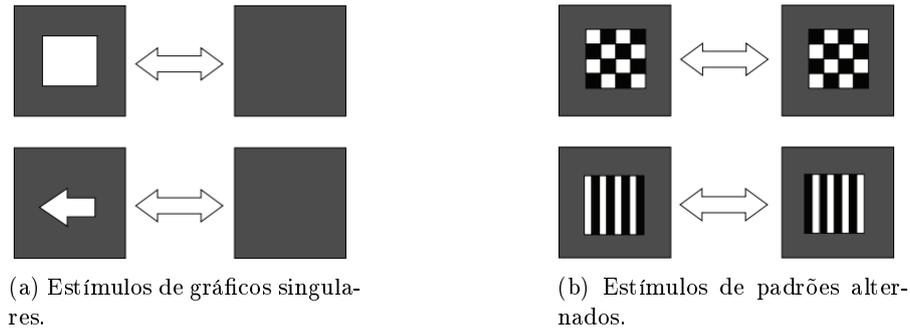


Figura 2.3: Exemplos de estímulos (retirado de [Zhu et al., 2010])

devidamente validados.

Nas BCI são utilizados três tipos de estímulos visuais: estímulos luminosos (provenientes de LEDs ou luzes fluorescentes), gráficos singulares que comutam entre dois estados a uma determinada frequência e estímulos de padrões alternados [Zhu et al., 2010]. Os estímulos luminosos podem ser gerados por circuitos dedicados, o que permite que os mesmos tenham uma elevada exatidão na frequência, no entanto não são ideais para aplicações como jogos, visto que não permitem a sua integração no ambiente do jogo. Os gráficos singulares consistem em formas desenhadas num ecrã, com dois estados (ver figura 2.3a). A alteração entre estes estados gera o efeito que produz o estímulo. Os estímulos de padrões alternados consistem na alternância entre dois padrões distintos. Os gráficos singulares podem ser considerados como um caso particular dos estímulos de padrões alternados [Zhu et al., 2010]. No entanto, estímulos de gráficos singulares geram uma resposta à frequência correspondente a um ciclo (duas comutações de estado), porém os estímulos de padrões alternados geram uma resposta à frequência correspondente a uma comutação de estado. Um padrão muito usado neste tipo de estímulos é o tabuleiro de xadrez, como mostra a figura 2.3b. Estímulos baseados em de padrões alternados e gráficos singulares são visualizados em ecrãs com taxas de refrescamento definidas. Neste caso, os ecrãs têm uma exatidão limitada ao apresentar estímulos com frequências não sub-múltiplas desta taxa de refrescamento. No caso dos estímulos com frequências sub-múltiplas da taxa de refrescamento, os instantes de transição coincidem com a atualização das *frames* do monitor. Estímulos com frequências não sub-múltiplas podem apresentar artefactos, os quais podem gerar respostas a frequências erradas.

## 2.2 Indicadores de desempenho de classificação em sistemas BCI

Os indicadores usuais de desempenho de classificação num sistema BCI são a taxa de sucesso da classificação (*accuracy*), velocidade de classificação e o número de alvos. O número de alvos é um indicador das tarefas mentais distintas que o utilizador teve que efetuar. No contexto do neuromecanismo SS-VEP, corresponde ao número de comandos disponibilizados. Estes indicadores podem ser agregados num único, definido como a taxa de transferência de informação (ITR) [Zhu et al., 2010]. Esta taxa é medida em bits/min e é definida por

$$ITR = \left( \log_2(N) + P \log_2(P) + (1 - P) \log \left( \frac{1 - P}{N - 1} \right) \right) \frac{60}{T} \quad (2.1)$$

onde  $T$  representa o tempo que o classificador demora a tomar uma decisão, desde o início de cada tarefa mental, este parâmetro apresenta-se como um indicador da velocidade de classificação. O valor  $P$  representa a taxa de sucesso da classificação, apresenta valores no intervalo  $[0, 1]$ .  $N$  representa o número de escolhas possíveis. É possível aumentar o ITR do sistema aumentando  $N$  ou diminuindo  $T$ , no entanto é necessário considerar o compromisso gerado. O aumento de escolhas aumenta o número de classes que o classificador terá que discriminar, podendo baixar a taxa de acerto de classificação e consequentemente produzir comandos errados do utilizador. Já o tempo de decisão representa o tempo necessário para que a resposta SSVEP tenha amplitude suficiente para a classificação, logo baixar este valor pode baixar a taxa de sucesso de classificação.

## 2.3 Aplicações BCI

Múltiplas aplicações podem ser desenvolvidas usando como componente sistemas BCI baseados em EEG. A tabela 2.1 lista exemplos deste tipo de aplicações. Muitas destas visam melhorar a qualidade de vida de pessoas com limitações motoras, proporcionando mobilidade, ou capacidade de comunicar. Outras servem como opções de lazer. Muitas aplicações configuram sistemas híbridos por utilizarem mais que um neuromecanismo.

Uma aplicação comum baseada em BCI é o *Speller*. Este permite a escrita em computador sem a necessidade de teclado, sendo de grande utilidade para pessoas com limitações motoras. Em [Yin et al., 2014] é apresentada uma abordagem de um sistema *speller* híbrido, utilizando P300 e SSVEP, com vista a aumentar a velocidade de escrita.

Em [Fan et al., 2014] é proposta a utilização de um sistema BCI para a escolha de destinos num veículo autónomo. São usados os neuromecanismos P300 e SSVEP, o primeiro para fazer a escolha do destino, e o segundo para a confirmação da escolha. Este sistema visa proporcionar mobilidade a utilizadores. Esta finalidade pode também ser obtida aplicando um sistema BCI a cadeiras de rodas, o que permite a utilização por pacientes sem a capacidade motora para operar a mesma. Um sistema deste género é descrito em [Punsawad and Wongsawat, 2013], onde o controlo é feito através de SSVEP e *motion visual stimulus*. Neste artigo é tida em especial consideração a segurança do paciente, fornecendo três modos de funcionamento: normal, fadiga e emergência. Em [Lopes et al., 2011] é proposta uma abordagem de controlo partilhado para robôs móveis, a qual foi testada experimentalmente com a plataforma “Robchair” desenvolvida no instituto de sistemas e robótica (ISR) da universidade de Coimbra. A BCI utilizada é baseada no neuromecanismo P300. Os sistemas BCI podem também ser usados para controlar humanoides, como o descrito em [Chae et al., 2012], baseado em imaginação motora.

A implementação de sistemas BCI na área dos jogos tem apresentado uma crescente popularidade. Estes proporcionam uma plataforma de teste para neuromecanismos. Permitem também o treino na utilização de BCI [Marshall et al., 2013], bem como proporcionar atividades de lazer a pacientes sem as capacidades motoras necessárias para operar mecanismos de controlo tradicionais, como por exemplo *joysticks*, ratos e teclados. Os jogos apresentam ao jogador objetivos a alcançar, os quais são atingidos depois de superados um conjunto de desafios. Estes desafios podem estar ligados à atenção, rapidez e estratégia, logo permitem o treino destas qualidades. A implementação de sistemas BCI em aplicações de entretenimento, como os jogos, permite também uma difusão destas interfaces pelo público em

geral. Exemplos da aplicação de BCI a diferentes jogos são apresentados na tabela 2.1. Em [Scherer et al., 2008] foi implementado um sistema de navegação num mundo virtual, embora não apresente os objetivos e desafios associados ao jogos, a exploração de um mundo virtual é uma característica muito comum em diversos tipos de jogos. Um jogo competitivo mutijogador é descrito em [Bonnet et al., 2013], onde é explorada a utilização de um sistema BCI num ambiente competitivo entre dois jogadores, com a finalidade de motivar, e contribuir à diversão associada ao jogo. É feita a comparação entre o ambiente multijogador e jogador único. Os autores concluem que a situação multijogador é preferível. A possibilidade de utilizar um sistema BCI como fonte de controlo adicional é explorada em [Laar et al., 2013]. A potência das ondas  $\alpha$  é usada como fonte de controlo extra para um jogo de computador (*world of warcraft*). Esta potência é usada como medida do relaxamento e atenção do utilizador. O modo de jogo é alterado consoante o valor da potência das ondas  $\alpha$ . Esta abordagem mostra uma possibilidade para integração de sistemas BCI na área dos videojogos comerciais. O neuromecanismo P300 é usado em [Angeloni et al., 2012] para controlar um jogo de memória.

## 2.4 Aplicações SSVEP

O SSVEP tem como vantagens boa imunidade a ruído, necessidade de um menor número de canais EEG e o facto de não necessitarem de períodos longos de treino. Permite também um alto ITR, quando comparado com outros neuromecanismos. No entanto, os estímulos utilizados podem causar fadiga visual em sessões de uso prolongado, sendo estes sistemas impróprios para pessoas com epilepsia ou fotosensibilidade. Existe também uma limitação associada ao número de estímulos que podem estar presentes numa aplicação, limitando o número de comandos possíveis. Isto deve-se às frequências disponíveis para a implementação. Habitualmente são usadas frequências entre os  $4Hz$  e os  $50Hz$  [Zhu et al., 2010]. A proximidade espacial dos estímulos pode gerar respostas indesejadas, ou mesmo distrair o utilizador do seu alvo. A tabela 2.2 lista exemplos de sistemas desenvolvidos, utilizando SSVEP como neuromecanismo.

Este neuromecanismo foi usada em [Meattini et al., 2014] para controlar remotamente uma mão robótica. Foi usado um sistema de aquisição comercial de baixo custo. Embora a taxa de sucesso seja elevada, são necessários  $26.6s$  de aquisição para gerar um comando, tornando a interação em tempo real com o sistema difícil. Os autores concluem que a elevada latência deve-se, principalmente, à baixa qualidade do sinal EEG fornecido pelo sistema comercial. Neste caso é proposto o algoritmo *common spatial patterns* (CSP) para a extração de características e *Support vector machine* (SVM) para a classificação. Também na área da robótica foi implementada uma órtese controlada através de LEDs [Ortner et al., 2011]. Este sistema tem como objetivo ajudar pessoas que sofram de tetraplegia, oferecendo-lhes algum controlo sobre a mão. O comando é enviado assincronamente, dando a possibilidade ao utilizador de só enviar comando quando deseja. Apesar de apresentar um taxa de verdadeiros positivos média de 78%, a taxa de falsos positivos apresenta valores elevados. O facto de os estímulos estarem no campo visual é apresentado como uma possível causa para o valor elevado de falsos positivos. Segundo os autores, o sistema pode ser melhorado implementando um mecanismo *brain-switch*, ou seja a utilização de outro neuromecanismo para ativar/desativar o envio de comandos originados por SSVEP. A extração de características é baseada na densidade espectral de potência (PSD), a classificação é feita escolhendo a frequência correspondente ao valor de maior amplitude.

Artigo	Neuromecanismo	Descrição
A speedy hybrid BCI spelling approach combining P300 and SSVEP. [Yin et al., 2014]	P300 e SSVEP	Abordagem de um sistema de escrita por BCI, que combina SSVEP e P300 com vista a aumentar a velocidade de escrita.
A brain-computer interface-based vehicle destination selection system using P300 and SSVEP signals [Fan et al., 2014]	P300 e SSVEP	Sistema de escolha de destinos recorrendo a um BCI. O destino é escolhido através do neuro-mecanismo P300, enquanto que a confirmação da seleção é feito por SSVEP.
Hybrid SSVEP-motion visual stimulus based BCI system for intelligent wheelchair. [Punsawad and Wongsawat, 2013]	SSVEP e Motion Visual Stimulus	Controlo de uma cadeira de rodas com três modos de funcionamento: modo normal onde é usado SSVEP e ritmos das ondas alfa; modo de fadiga, onde são usados Motion Visual Stimulus; modo de emergência, onde o sistema é controlado remotamente.
Wheelchair navigation assisted by Human-Machine shared-control and a P300-based Brain Computer Interface [Lopes et al., 2011]	P300	Abordagem de controlo partilhado para robôs móveis. É proposto um paradigma baseado em P300 na implementação da BCI.
Toward Brain-Actuated Humanoid Robots : Asynchronous Direct Control Using an EEG-BasedBCI [Chae et al., 2012]	Imaginação motora	Controlo de um robô humanoide através de imaginação motora. O robô navega um labirinto enviando imagens de uma câmara montada na cabeça.
Towards self-paced (asynchronous) Brain-Computer Communication : Navigation through virtual worlds [Scherer et al., 2008]	Imaginação Motora	Controlo de um avatar num mundo virtual. O sistema é self-paced, logo considera o caso de o utilizador não enviar um comando. Contém três comandos de navegação: rodar à direita, rodar à esquerda e mover em frente.
Two Brains, One Game: Design and Evaluation of a Multiuser BCI Video Game Based on Motor Imagery [Bonnet et al., 2013]	Imaginação motora	Desenvolvimento de um jogo multi jogador controlado por imaginação motora. Este é um jogo de futebol onde o utilizador marca golos, do lado direito ou esquerdo imaginando movimentos na mão direita ou esquerda respetivamente.
Experiencing BCI control in a popular computer game [Laar et al., 2013]	Ritmos alfa	Teste da utilização da potência das ondas alfa como canal adicional de controlo no jogo <i>world of warcraft</i> .
P300-based Brain-Computer Interface Memory Game to Improve Motivation and Performance [Angeloni et al., 2012]	P300	Jogo de memória, onde o utilizador tem que descobrir uma imagem alvo num conjunto de imagens, através do neuro-mecanismo P300.

Tabela 2.1: Aplicações BCI controladas com P300 e imaginação motora (por vezes combinados com SSVEP).

Os sistemas *speller* também podem ser implementados recorrendo a SSVEP, como é exemplo o trabalho de [Vilic et al., 2013], que apresenta um sistema com esta finalidade. Este sistema *speller* tem a particularidade de apresentar suporte de dicionário, ou seja o utilizador pode inserir alguns caracteres da palavra desejada, e de seguida escolher a palavra de uma lista. As características são extraídas no domínio da frequência, através da PSD. O comando a enviar é escolhido através de critérios, com o objetivo de garantir que a frequência de valor máximo detetada é predominante nesse instante, bem como nas épocas anteriores. A taxa de sucesso com este método é alta atingindo os 90.81%, com um tempo médio de seleção de 6.62s, correspondendo a um ITR médio de 21.94<sup>bits</sup>/min.

Em [Kuo-Kai Shyu et al., 2010] é apresentado um sistema de controlo de aplicações multimédia recorrendo a SSVEP com marcação de fase. A particularidade deste sistema é que o processamento de sinal é implementado utilizando uma *field programmable gate array* (FPGA), apresentando-se como um sistema de baixo custo. São utilizados quatro estímulos à frequência de 21Hz, com um desfasamento de 90°. Este sistema apresenta uma ITR média de 24.67<sup>bits</sup>/min.

Na área de entretenimento, existem várias possibilidades para a aplicação de SSVEP, desde jogos de exploração a jogos de estratégia. No entanto, os estímulos podem revelar-se desconfortáveis em sessões de jogos prolongadas. Em [Lalor et al., 2005] é feita uma primeira abordagem à utilização deste neuromecanismo num jogo de computador. O jogo consiste num avatar que se desequilibra para um dos lados aleatoriamente. Cabe ao jogador equilibrar o avatar através da resposta gerada por um de dois estímulos presentes. O jogo passa-se num ambiente imersivo 3D, mostrando que a resposta SSVEP pode ser usada, apesar deste fator de distração.

Em [Po-Lei et al., 2014] é proposto um jogo de uma garra virtual, no qual são usados elétrodos a seco. Os autores concluem que o nível de concentração está associado à taxa de sucesso do jogador.

O trabalho realizado em [Chumerin and Manyakov, 2013] apresenta-se como um passo na direção dos jogos comerciais, visto que utiliza um sistema de aquisição comercial para fazer a aquisição de dados. Neste trabalho é feita uma comparação entre o sistema comercial e um sistema de aquisição usado em investigação e para uso clínico, concluindo que os resultados em ambos os sistemas são semelhantes. Mais uma vez é investigada a relação entre a atenção e a resposta SSVEP, onde é encontrada uma correlação entre ambas.

Um jogo de damas controlado através de LEDs é proposto em [Akhtar et al., 2014]. As peças são movidas por uma braço robótico. É feito o teste de vários fatores na resposta taxa de sucesso do sistema. Estes fatores são a distancia entre o utilizador e os estímulos, distância entre estímulos, ângulo entre estímulos e quantidade de estímulos no campo visual do jogador. Apesar de não ser encontrada nenhuma ligação entre a taxa de sucesso e a maior parte destes fatores, conclui-se que o número de estímulos no campo visual do utilizador baixa a taxa de sucesso.

A versão original do jogo "Spacecraft" desenvolvida por [Parafita et al., 2013], é controlada através de SSVEP com marcação de fase. Apesar da maior parte dos trabalhos nesta área considerarem que a resposta SSVEP é somente discriminável a partir do 6Hz, neste trabalho são usadas frequências entre 3Hz e 5Hz com elevada taxa de sucesso, da ordem dos 97%. Esta configuração permite baixo tempo de calibração, bem como uma elevada taxa de sucesso utilizando um canal.

Artigo	Descrição	Desempenho
Gestural art: A Steady State Visual Evoked Potential (SSVEP) based Brain Computer Interface to express intentions through a robotic hand [Meattini et al., 2014]	Controlo de uma mão robótica usando um sistema de aquisição comercial (Emotiv EPOC).	Taxa de sucesso média: 88.9% Latência média: 26.6s
An SSVEP BCI to control a hand orthosis for persons with tetraplegia. [Ortner et al., 2011]	Controlo de uma Órtese através de um sistema BCI self-paced (assíncrono) baseado em SSVEP. Órtese com 4 posições possíveis controladas através de 2 LEDs.	Valor preditivo positivo médio: $78 \pm 10\%$
DTU BCI speller: an SSVEP-based spelling system with dictionary support. [Vilic et al., 2013]	Sistema de escrita por BCI, baseado em SSVEP, com suporte de dicionário.	Taxa de sucesso média: $90.81 \pm 4.11\%$ Tempo médio de seleção: $6.62s \pm 1.03$ Taxa de transmissão: $21.94 \pm 15.63^{\text{bits}}/\text{min}$
Development of a Low-Cost FPGA-Based SSVEP BCI Multimedia Control System. [Kuo-Kai Shyu et al., 2010]	Sistema de controlo multimédia de baixo custo. O processamento de sinal é feito com uma FPGA. Estímulos são implementados com LEDs.	Taxa de sucesso média: 89.29% Tempo médio de seleção: $3.77s \pm 1.03$ Taxa de transmissão média: $24.67^{\text{bits}}/\text{min}$
Steady-state visual evoked potential-based computer gaming on a consumer-grade eeg device. [Chumerin and Manyakov, 2013]	Navegação de um avatar num labirinto, utilizando um sistema de aquisição comercial (Emotiv Epoch).	Taxa de sucesso média: 80.37%
A spacecraft game controlled with a brain-computer interface using SSVEP with phase tagging. [Parafita et al., 2013]	Controlo de uma versão anterior do jogo Spacecraft. Tem como objetivo controlar uma nave para a direita e para a esquerda, com vista a desviar a mesma de obstáculos, através de SSVEP com marcação de fase.	Taxa de sucesso média: 97% Taxa de transmissão média: $15^{\text{bits}}/\text{min}$
Playing checkers with your mind: An interactive multiplayer hardware game platform for brain-computer interfaces. [Akhtar et al., 2014]	Jogo das damas em hardware, controlado com SSVEP. A resposta SSVEP é obtida com LEDs nas casas do tabuleiro. Estudo dos efeitos da distancia e ângulo do utilizador aos estímulos, bem como número de estímulos, distância entre estímulos e outros aspetos afetam a resposta SSVEP.	Taxa de sucesso média: 82.7%
Design a brain computer interface gaming system using steady-state visual evoked potential. [Po-Lei et al., 2014]	Jogo de controlo de uma garra virtual, com o objetivo de agarrar objetos apresentados. São utilizados dois estímulos, um move a garra, o outro ordena a que a garra agarra o objeto.	Taxa de sucesso média: 96% Taxa de transmissão média: $18.23^{\text{bits}}/\text{min}$

Tabela 2.2: Aplicações BCI controladas por SSVEP.



# Capítulo 3

## Métodos

### 3.1 Extração de características

Com vista a identificar o comando que o utilizador pretende enviar ao sistema, é necessário extrair características do sinal EEG. Diversos métodos podem ser utilizados nesta fase do processamento do sinal. A resposta SSVEP é um padrão EEG caracterizado essencialmente no domínio da frequência, logo as características usadas para classificação são essencialmente extraídas neste domínio. Técnicas de extração baseadas em medidas de correlação do sinal com sinusóides de determinadas frequências são também muito utilizadas. As secções seguintes descrevem as várias técnicas que foram aplicadas para deteção de SSVEP.

#### 3.1.1 Transformada de Fourier

A conversão do sinal no domínio do tempo para o domínio da frequência é feita recorrendo à transformada de Fourier

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad (3.1)$$

onde  $x(t)$  representa o sinal no domínio do tempo e  $\omega$  representa a frequência angular, em  $rad/s$ . Esta é uma generalização da série de Fourier, que descreve qualquer sinal periódico por uma soma de exponenciais complexas. A transformada inversa de Fourier é dada por

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{j\omega t} d\omega. \quad (3.2)$$

O sinal  $x(t)$  pode apresentar um determinado tipo de simetrias, que se manifestam no domínio da frequência. Estas simetrias podem ser aproveitadas para simplificar a análise do sinal, bem como melhorar a eficiência computacional do cálculo da transformada. Sendo o sinal EEG,  $x(t)$ , um sinal real, então a propriedade da simetria da transformada de Fourier,  $X(-\omega) = X^*(\omega)$ , pode ser aplicada. Logo, as características a classificar podem ser extraídas das frequências positivas do espectro, já que as correspondentes frequências negativas terão o conjugado do valor.

A potência presente num intervalo de frequências é calculada recorrendo ao teorema de Parseval

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 d\omega. \quad (3.3)$$

Para sinais reais a potência associada a cada frequência  $\omega$  é dada por

$$P(\omega) = 2|X(\omega)|^2 \text{ onde } 0 \leq \omega < +\infty. \quad (3.4)$$

No domínio discreto, a transformada de Fourier é definida como Transformada Discreta de Fourier, dada por

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi\frac{k}{N}n} \text{ onde } k = 0, \dots, N-1. \quad (3.5)$$

Onde são utilizadas as amostras do sinal  $x(t)$ , ou seja  $x(n) \equiv x(nh)$ . O valor  $n$  representa o índice da amostra,  $h$  representa o período de amostragem do sinal e  $N$  é o número de amostras. Esta transformada converte  $N$  amostras do sinal  $x(t)$  em  $N$  valores complexos. A relação entre estes e a Transformada de Fourier é por

$$X(k) = \frac{X(\omega_k)}{h} \text{ onde } \omega_k = k\frac{2\pi}{Nh} \text{ e } k = 0, \dots, N-1, \quad (3.6)$$

onde é possível observar que cada valor  $X(k)$  representa uma amostra da Transformada de Fourier do sinal  $x(t)$ , para o valor de frequência angular  $\omega_k$ , a menos de um fator escala  $\frac{1}{h}$ . Os valores de  $\omega_k$  variam entre  $-\omega_c$  e  $\omega_c$ , onde  $\omega_c = 2\pi\frac{1}{2h}$  é a frequência de Nyquist.

É possível observar que  $X(k)$  é periódica, com período  $N$ , logo  $X(-k) = X(N-k)$ , sendo possível alterar o domínio de  $k$  para o intervalo  $[0, N-1]$ . Os valores de  $k$  entre  $[0, \frac{N}{2}[$  correspondem às frequências positivas no intervalo  $[0, \omega_c[$ , já os valores de  $k$  entre  $]\frac{N}{2}, N-1]$  correspondem às frequências no intervalo  $]-\omega_c, 0]$ . O índice  $k = \frac{N}{2}$  corresponde, simultaneamente, aos valores de  $\omega_c$  e  $-\omega_c$ .

Tal como no caso contínuo, a Transformada Discreta de Fourier contém relações entre as simetrias do sinal no domínio das amostras e correspondente transformada. Devido a  $x(n)$  ser real e recorrendo à periodicidade da transformada, é possível observar  $X(N-k) = X^*(k)$ .

O teorema de Parseval pode, também, ser aplicado ao domínio discreto, sendo dado por

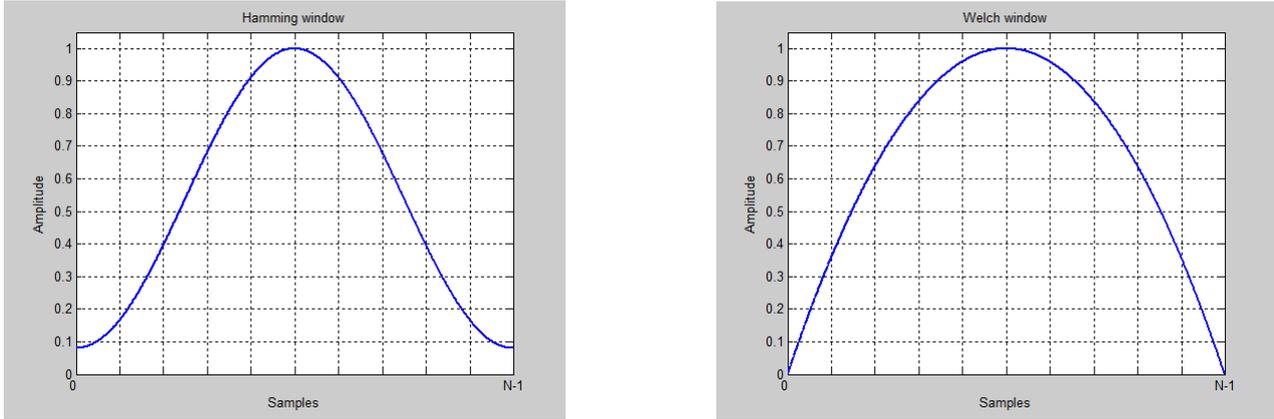
$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2. \quad (3.7)$$

Os valores da Transformada Discreta de Fourier são normalmente obtidos através de um algoritmo designado por FFT (*fast Fourier transform*) que reduz o número de operações necessárias relativamente ao cálculo direto de (3.5). Este método apresenta um grau de complexidade de  $(N) \log_2(N)$  em contraste com a aplicação de (3.5) que apresenta uma complexidade  $N^2$  [Press et al., 1992].

#### 3.1.2 Método de estimação da densidade espectral de Welch

O método de Welch [Welch, 1967] deve o seu nome a Peter D. Welch e permite determinar uma estimativa da potência associada a cada frequência de um espectro, para um sinal  $x(n)$ . Este método baseia-se na segmentação de uma sequência de dados de tamanho  $N$ , em sequências mais pequenas de tamanho  $L$ , onde para cada uma é calculado um periodograma modificado e é feita a média dos vários periodogramas. Uma vantagem importante é o facto de reduzir a variância da densidade de potência estimada, com o custo da redução resolução no domínio da frequência, i.e. o intervalo entre frequências passa a ser maior.

A sequência de operações do método é apresentada de seguida, como é descrita em [Welch, 1967]. De início é necessário segmentar os dados em  $K$  sequências de tamanho  $L$ , podendo haver sobreposição de amostras entre os segmentos. Logo, considerando  $D$  amostras sobrepostas e  $x(n)$  como o sinal original, cada segmento  $x_k(n)$  é definido por



(a) Janela de Hamming

(b) Janela de Welch

Figura 3.1: Exemplos de funções janela.

$$x_k(n) = x(n + (k - 1)D) \text{ onde } n = 0, 1, \dots, L - 1. \quad (3.8)$$

Para cada secção  $x_k(n)$  é calculado um periodograma modificado, para tal é necessário definir uma janela  $W(n)$  de tamanho  $L$  e determinar a Transformada Discreta de Fourier de  $x_k(n)W(n)$ :

$$A_k(n) = \frac{1}{L} \sum_{l=0}^{L-1} x_k(l)W(l)e^{-2\pi j \frac{l}{L}n} \quad (3.9)$$

Por fim, para cada  $A_k(n)$  determina-se um periodograma modificado  $I_k(f_n)$  através de

$$I_k(f_n) = \frac{L}{U} |A_k(n)|^2 \text{ onde } f_n = \frac{n}{L}; n = 0, \dots, \frac{L}{2} \text{ e } U = \frac{1}{L} \sum_{l=0}^{L-1} W^2(l) \quad (3.10)$$

e determina-se a estimativa da densidade espectral de potência  $\hat{P}(f_n)$ , calculando a media dos  $K$  periodogramas modificados:

$$\hat{P}(f_n) = \frac{1}{K} \sum_{k=1}^K I_k(f_n) \quad (3.11)$$

No cálculo dos periodogramas podem ser usados diversos tipos de janela. Um exemplo possível é a janela de Hamming (figura 3.1a):

$$W_h(n) = \alpha - \beta \cos\left(\frac{2\pi n}{L-1}\right) \text{ onde } \alpha = 0.54 \text{ e } \beta = 0.46. \quad (3.12)$$

Outro exemplo é a janela de Welch (figura 3.1b) :

$$W_w(n) = 1 - \left(\frac{n - \frac{L-1}{2}}{\frac{L-1}{2}}\right)^2. \quad (3.13)$$

A utilização de amostras sobrepostas em cada segmento permite atenuar o efeito de perda de informação gerado pelas janelas [Mathworks, 2012c].

### 3.1.3 Algoritmo de Goertzel

Em situações que necessitem do cálculo de apenas  $M$  valores da DFT, onde  $M \leq \log_2(N)$ , o algoritmo de Goertzel [Proakis and Monolakis, 1996] revela-se uma escolha mais eficiente que o algoritmo FFT, visto que este último determina obrigatoriamente todos os valores da DFT [Proakis and Monolakis, 1996, Sysel and Rajmic, 2012]. Este algoritmo permite calcular os valores de  $X(k)$  para índices  $k$  à escolha, utilizando (3.5) como um banco de filtros em paralelo, onde cada filtro determina o valor da DFT para cada índice  $k$  [Proakis and Monolakis, 1996].

A dedução do algoritmo é possível tomando  $W_N^{mk} = e^{-j\frac{2\pi}{N}mk}$ , é possível observar que  $W_N^{mk}$  é periódica, de período  $N$  e que  $W_N^{-Nk} = 1$ . Possibilitando que (3.5) possa ser reescrita, substituindo  $W_N^{mk}$  e multiplicando por  $W_N^{-Nk}$ , sob a forma:

$$X(k) = \sum_{m=0}^{N-1} x(m)W_N^{mk} = \sum_{m=0}^{N-1} x(m)W_N^{-k(N-m)}, \quad (3.14)$$

A convolução discreta para uma sequência  $x(n)$  finita, com  $N$  amostras e início em  $n = 0$ , e um sinal  $h_k(n)$  é dada por

$$y_k(n) = x(n) * h_k(n) = \sum_{m=0}^{N-1} x(m)h_k(n-m). \quad (3.15)$$

Considerando  $y_k(N) = X(k)$ , (3.14) pode ser reescrita por

$$y_k(n) = \sum_{m=0}^{N-1} x(m)W_N^{-k(n-m)}, \quad (3.16)$$

É possível observar semelhanças entre (3.15) e (3.16), considerando

$$h_k(n) = W_N^{-kn}, \quad (3.17)$$

logo  $y_k(n)$  pode ser obtido através da convolução de  $x(n)$  com o sinal  $h_k(n)$ .

O sinal  $h_k(n)$  pode ser visto como a resposta a impulso de um sistema linear e invariante no tempo discreto. Aplicando a transformada de  $Z$  a (3.15) obtém-se:

$$Y_k(z) = X(z) \cdot H_k(z) \quad (3.18)$$

onde

$$H_k(z) = \frac{1}{1 - W_N^{-k}z^{-1}} \quad (3.19)$$

representa a transformada de  $Z$  de (3.17). Os valores de  $X(k)$  podem ser obtidos passando o sinal  $x(n)$  por um conjunto de filtros em paralelo, onde cada filtro terá uma função de transferência  $H_k(Z)$ , com um pólo em  $e^{-j\frac{2\pi}{N}k}$ , ou seja no círculo de raio unitário à frequência  $\omega_k = \frac{2\pi k}{N}$ . Cada filtro calcula o valor da DFT para o índice  $k$ .

É possível reduzir o número de multiplicações complexas reescrevendo (3.19) como uma função de

transferência de segundo grau:

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \frac{1 - W_N^k z^{-1}}{1 - W_N^k z^{-1}} = \frac{1 - W_N^k z^{-1}}{1 - (2\cos(\frac{2\pi k}{N})) z^{-1} + z^{-2}} \quad (3.20)$$

ou como um conjunto de equações às diferenças:

$$v_k(n) = 2\cos\left(\frac{2\pi k}{N}\right) v_k(n-1) - v_k(n-2) + x(n) \quad (3.21)$$

$$y_k(n) = v_k(n) - W_N^k v_k(n-1). \quad (3.22)$$

Para cada valor  $k$ , (3.21) é aplicada recursivamente para os valores de  $n = 0, 1, \dots, N$ , com condições iniciais  $v_k(-1) = 0$  e  $v_k(-2) = 0$ . De seguida é determinado  $X(k) = y_k(N)$ , através de (3.22). Neste caso, cada iteração da parte recursiva do algoritmo necessita de uma multiplicação pelo fator real  $2\cos(\frac{2\pi k}{N})$ , que pode ser calculado na primeira iteração, visto que é constante para cada  $k$ . O fator complexo  $W_N^k$  é somente utilizado no cálculo do resultado do algoritmo, para cada  $k$ . Esta forma do algoritmo diminui o número de operações necessárias com números complexos.

### 3.1.4 Análise de correlação canónica

É possível extrair características dos sinais EEG através da correlação entre vários canais EEG e um conjunto de sinais de referência associados às frequências que evocam os SSVEP. Uma das técnicas de correlação é o coeficiente de correlação canónica (CCA - *canonical correlation analysis*). O sinal de referência pode ser o sinal evocado SSVEP à frequência fundamental do estímulo, ou o conjunto das  $N$  componentes harmónicas (frequência fundamental e suas frequências múltiplas). Neste caso, toma-se como hipótese que a atividade elétrica cerebral, que gera o SSVEP, funciona como a saída de um sistema linear, tendo como entrada o sinal de estímulo [Lin et al., 2007]. Logo, como o cérebro humano tem uma dinâmica semelhante a um filtro passa baixo [Lin et al., 2007], é possível testar a existência de resposta SSVEP, analisando a correlação canónica entre o sinal EEG e estes sinais de referência. Este processo é aplicado a cada frequência de interesse, gerando um coeficiente de correlação. Embora este método gere vários coeficientes de correlação canónica, é escolhido o valor máximo como característica de cada frequência [Lin et al., 2007].

A análise da correlação canónica é um método estatístico, que permite encontrar um coeficiente de correlação, entre dois conjuntos de dados. Este representa uma generalização do coeficiente de correlação, que é aplicável a duas variáveis. Cada conjunto de dados é composto por um determinado número de variáveis com o mesmo número de amostras. Aplicando ao caso de estudo neste trabalho, então cada canal do sinal EEG representa uma variável do primeiro conjunto de dados, onde as amostras representam as amostras temporais do mesmo. Os sinais de referência compõem o segundo conjunto de dados. O número de coeficientes, que se pode obter destes dados, é igual ao número de variáveis do menor conjunto de dados. Cada coeficiente é determinando, a partir de cada conjunto de dados, através de uma variável definida por uma combinação linear das variáveis desse conjunto. Esta variável tem o nome de variável canónica, e é determinada de modo a maximizar o coeficiente de correlação canónica obtido. De seguida é determinada a correlação entre as variáveis canónicas para cada conjunto de dados. Este processo é mostrado na figura 3.2. A derivação do método, apresentada de seguida foi obtida de [Weenink, 2003].

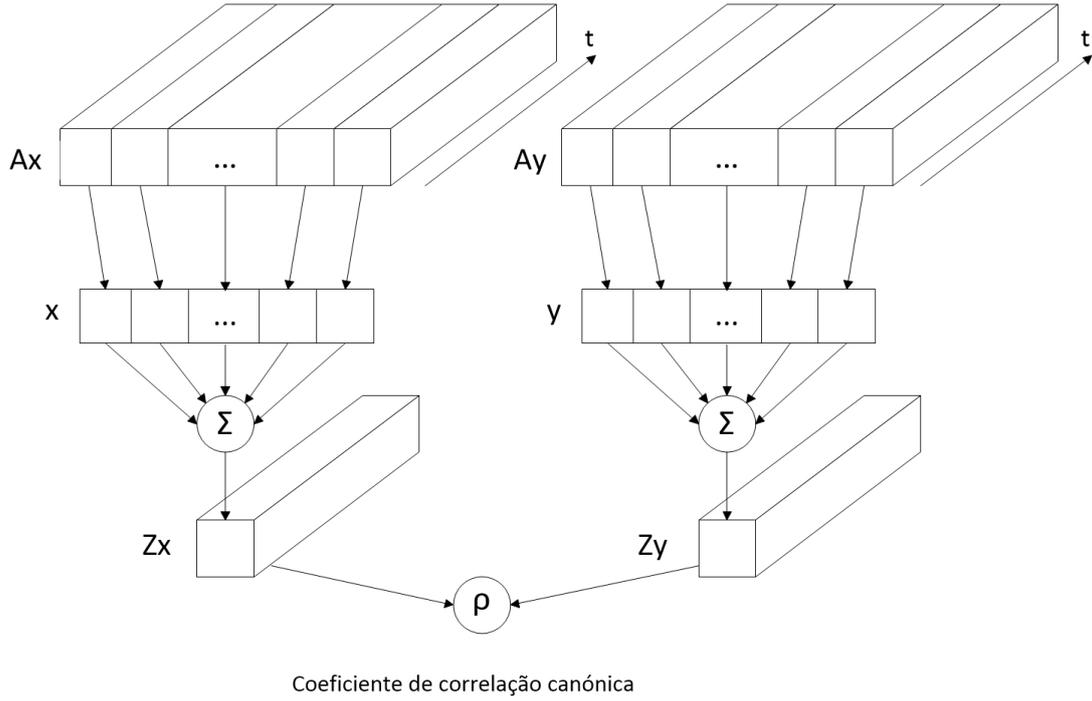


Figura 3.2: Ilustração do método CCA.

Considera-se os conjuntos de dados  $X$  e  $Y$ . O conjunto de dados  $X$  representa o sinal EEG e tem dimensão  $m \times n$ , onde  $n$  é o número de canais.  $Y$  representa o conjunto de sinais de referência, tendo dimensão  $m \times p$ , onde  $p$  é o número de sinais de referência. O valor  $m$  representa o número de amostras temporais de ambos os conjuntos de dados.

$$X = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix} \text{ e } Y = \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1p} \\ Y_{21} & Y_{22} & \cdots & Y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{m1} & Y_{m2} & \cdots & Y_{mp} \end{bmatrix} \quad (3.23)$$

É possível determinar as matrizes  $A_x$  subtraindo a cada coluna de  $X$  a sua média, ou seja  $A_{ij} = X_{ij} - \mu_j$ , onde  $\mu_j$  é a média das amostras da coluna  $j$  de  $X$ .  $A_y$  é obtida aplicando o mesmo processo a matriz  $Y$ . Cada elemento da matriz de covariância de  $X$ ,  $\Sigma$ , é dado por

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1n} \\ \Sigma_{21} & \Sigma_{22} & \cdots & \Sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{n1} & \Sigma_{n2} & \cdots & \Sigma_{nn} \end{bmatrix} \text{ onde } \Sigma_{ij} = \sum_{k=1}^m (X_{ki} - \mu_i)(X_{kj} - \mu_j), \quad (3.24)$$

$\mu_j$  é a média da coluna  $j$  da matriz  $X$  e  $\mu_i$  é a média da coluna  $i$ . Logo, a matriz de covariância pode ser obtida pelo produto

$$\Sigma = \frac{1}{m-1} A_x^T A_x. \quad (3.25)$$

O coeficiente de correlação entre a colunas  $j$  e coluna  $i$  de  $X$  é obtido através dos elementos da matriz de covariância,  $\Sigma$ , por

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}\Sigma_{jj}}}. \quad (3.26)$$

As matrizes de dados  $A_x$  e  $A_y$  podem ser denominados como dados independentes e dependentes respetivamente [Weenink, 2003]. É possível definir as variáveis canónicas  $Z_x$  e  $Z_y$ , de dimensão  $m \times 1$ , projetando  $A_x$  e  $A_y$  nos vetores  $x$  e  $y$  respetivamente. Os vetores  $x$ , de dimensão  $n \times 1$ , e  $y$ , de dimensão  $p \times 1$ , indicam os fatores escala a multiplicar a cada variável dos conjuntos de dados de modo a maximizar a correlação entre  $Z_x$  e  $Z_y$ :

$$Z_x = A_x x \quad (3.27)$$

$$Z_y = A_y y. \quad (3.28)$$

A correlação entre as variáveis canónicas é determinada por

$$\rho = \frac{Z_y^T Z_x}{\sqrt{Z_y^T Z_y} \sqrt{Z_x^T Z_x}}. \quad (3.29)$$

É possível observar que a  $\rho$  não é afetado pela multiplicação de uma constante a  $Z_x$  ou  $Z_y$ , logo é possível omitir o termo  $\frac{1}{m-1}$  no seu cálculo.

É possível também forçar as restrições

$$\begin{aligned} Z_x^T Z_x &= x^T A_x^T A_x x = x^T \Sigma_{xx} x = 1 \\ Z_y^T Z_y &= y^T A_y^T A_y y = y^T \Sigma_{yy} y = 1 \end{aligned} \quad (3.30)$$

onde  $\Sigma_{xx} = A_x^T A_x$  é a matriz de covariância de  $X$  e  $\Sigma_{yy} = A_y^T A_y$  é a matriz de covariância de  $Y$ , ambas multiplicadas pelo fator  $(m-1)$ . Sob estas condições, é necessário determinar os vetores  $x$  e  $y$  que maximizem o valor  $\rho$ .

Este problema de maximização pode ser escrito sob a forma de lagrangiano [Weenink, 2003], através de

$$L(\rho_x, \rho_y, x, y) = y^T \Sigma_{yx} x - \frac{\rho_x}{2} (x^T \Sigma_{xx} x - 1) - \frac{\rho_y}{2} (y^T \Sigma_{yy} y - 1). \quad (3.31)$$

A matriz  $\Sigma_{xy} = \Sigma_{yx}^T = A_x^T A_y$  representa a matriz de covariância entre  $A_x$  e  $A_y$ . (3.31) pode ser resolvida derivando em ordem a  $x$  e  $y$ , obtendo

$$\frac{\partial L}{\partial x} = \Sigma_{xy} y - \rho_x \Sigma_{xx} x = 0 \quad (3.32)$$

$$\frac{\partial L}{\partial y} = \Sigma_{yx} x - \rho_y \Sigma_{yy} y = 0 \quad (3.33)$$

Através de (3.32) e (3.33), é possível obter

$$x^T \frac{\partial L}{\partial x} - y^T \frac{\partial L}{\partial y} = \rho_x x^T \Sigma_{xx} x - \rho_y y^T \Sigma_{yy} y = 0. \quad (3.34)$$

De (3.34) conjuntamente com (3.30) é possível concluir que  $\rho_x = \rho_y = \rho$ .

Substituindo  $\rho$  em (3.32) e (3.33), para o caso em que  $\Sigma_{xx}$  e  $\Sigma_{yy}$  são invertíveis, obtém-se:

$$x = \frac{\Sigma_{xx}^{-1} \Sigma_{xy} y}{\rho} \quad (3.35)$$

$$y = \frac{\Sigma_{yy}^{-1} \Sigma_{yx} x}{\rho} \quad (3.36)$$

Substituindo  $x$  em (3.33) por (3.35) e  $y$  em (3.32) por (3.36) obtém-se:

$$(\Sigma_{xy}^T \Sigma_{xx}^{-1} \Sigma_{xy} - \rho^2 \Sigma_{yy}) y = 0 \quad (3.37)$$

$$(\Sigma_{xy}^T \Sigma_{yy}^{-1} \Sigma_{xy}^T - \rho^2 \Sigma_{xx}) x = 0 \quad (3.38)$$

que governam o cálculo dos coeficientes de correlação canónica.

Multiplicando (3.37) por  $\Sigma_{yy}^{-1}$  pela esquerda, obtém-se:

$$\Sigma_{yy}^{-1} \Sigma_{xy}^T \Sigma_{xx}^{-1} \Sigma_{xy} y = \rho^2 y, \quad (3.39)$$

onde é possível observar que os vetores  $y$  correspondem aos vetores próprios da matriz  $\Sigma_{yy}^{-1} \Sigma_{xy}^T \Sigma_{xx}^{-1} \Sigma_{xy}$ , associados aos valores próprios  $\rho^2$ . Multiplicando a (3.37) por  $\Sigma_{xy} \Sigma_{yy}^{-1}$  à esquerda e inserindo a matriz identidade,  $\Sigma_{xx}^{-1} \Sigma_{xx}$ , obtém-se:

$$(\Sigma_{xy}^T \Sigma_{yy}^{-1} \Sigma_{xy}^T - \rho^2 \Sigma_{xx}) \Sigma_{xx}^{-1} \Sigma_{xy} y = 0, \quad (3.40)$$

depois de alguma manipulação matemática. É possível observar, que (3.40) é semelhante a (3.38), onde  $x = \Sigma_{xx}^{-1} \Sigma_{xy} y$ . Logo, é possível obter o vetor  $x$ , depois de determinado o vetor  $y$ . Os coeficientes de correlação canónica correspondem aos valores próprios  $\rho^2$ . Conhecendo  $x$  e  $y$  é possível determinar as variáveis canónicas  $Z_x$  e  $Z_y$  através de (3.27) e (3.28) respetivamente.

## 3.2 Métodos de classificação

As características extraídas são utilizadas para classificar o sinal EEG. É possível utilizar algoritmos de classificação supervisionados, com vista atribuir uma classe baseada em dados de treino, obtidos previamente. É possível também, escolher um comando baseado na característica do sinal com maior amplitude, sem recorrer a um classificador. Esta escolha pode ser feita com características extraídas do domínio da frequência, escolhendo a frequência que possua uma maior amplitude no seu valor do espectro. O mesmo critério pode ser aplicado para os coeficientes de correlação canónica, onde pode ser escolhida a frequência correspondente ao maior coeficiente. Estes métodos de escolha possuem a vantagem de não necessitarem de uma fase prévia de treino.

### 3.2.1 Análise discriminante linear bayesiana

A análise discriminante linear bayesiana é uma abordagem estatística para o problema de classificação de padrões [Duda et al., 1999]. Pode ser usado no contexto SSVEP, para atribuir uma classe a cada frequência alvo. A utilização deste classificador pressupõe a existência de um conjunto de dados de treino, que indicam o comportamento das características para cada classe considerada, logo é um método de classificação supervisionado. Os dados de treino geram um sub-espço, no espaço das características, associado a cada classe de treino. As características são identificadas pelo subespço onde se encontram.

Este método baseia-se na minimização do risco associado à tomada de uma decisão de classificação. Considerando as ações de classificação  $\alpha_n$ , onde  $n = 1, \dots, N$ , o objetivo do classificador é escolher uma ação utilizando como referência um vetor coluna de características  $\mathbf{x}$ . Considera-se um conjunto de classes  $\omega_k$ , onde  $k = 1, \dots, K$ . As ações de classificação podem representar a escolha de uma classe pelo classificador, no entanto, permitem incluir ações como não efetuar a classificação. De uma forma simplificada podem-se considerar  $K$  ações de classificação, onde  $\alpha_k$  representa a escolha da classe  $\omega_k$  como resultado da classificação. É necessário definir uma função de custo,  $\lambda(\alpha_n|\omega_k)$ , que caracteriza a perda associada a escolher a ação  $\alpha_n$  quando a classe correta é  $\omega_k$ . Uma função de custo possível é

$$\lambda(\alpha_n|\omega_k) = \begin{cases} 0 & \text{se } n = k \\ 1 & \text{se } n \neq k \end{cases}, \quad (3.41)$$

onde é possível observar que o custo de escolher a classe correta é nulo.

Para tomar uma decisão, é necessário inferir a classe a partir do vetor de características, logo é necessário conhecer a probabilidade da classe correta ser  $\omega_k$ , sabendo que a característica é  $\mathbf{x}$ , (probabilidade *a posteriori*). Esta probabilidade é dada pelo teorema de Bayes:

$$\hat{P}(\omega_k|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_k)P(\omega_k)}{p(\mathbf{x})} \quad \text{onde } p(\mathbf{x}) = \sum_{k=1}^K p(\mathbf{x}|\omega_k)P(\omega_k), \quad (3.42)$$

onde  $p(\mathbf{x})$  é um fator constante para todas as classes, que normaliza a soma dos  $\hat{P}(\omega_k|\mathbf{x})$  para todas as classes. Este fator é também chamado *evidence* e representa a função de densidade de probabilidade do vetor  $\mathbf{x}$ . O termo  $P(\omega_k)$ , também conhecido como conhecimento à priori, representa o conhecimento sobre o processo, sendo a probabilidade de  $\omega_k$  ser a classe correta sem ter em conta as características. Este termo pode ser considerado constante para todas as classes, ou pode ser obtido pela contagem das ocorrências de cada classe nos dados de treino. A soma das probabilidades à priori para todas as classes é unitária. A função de distribuição de probabilidade do vetor  $\mathbf{x}$  para a classe  $\omega_k$ ,  $p(\mathbf{x}|\omega_k)$  representa o quão provável é que o vetor de características tome o valor  $\mathbf{x}$ , se este pertencer à classe  $\omega_k$ , também conhecido como *likelihood*. Uma possibilidade para esta função, a qual foi usada no trabalho desenvolvido, é uma distribuição gaussiana:

$$p(\mathbf{x}|\omega_k) = \frac{e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \Sigma^{-1}(\mathbf{x}-\mu_k)}}{\sqrt{2\pi} |\Sigma|}. \quad (3.43)$$

O termo  $p(\mathbf{x}|\omega_k)$  é obtido dos dados de treino, de onde é determinada a média para cada classe  $\mu_k$ ,

### 3.2. MÉTODOS DE CLASSIFICAÇÃO

---

bem como a matriz de covariância para os dados de treino  $\Sigma$ . Logo é possível observar que a matriz de covariância é a mesma para todas as classes, o que torna as fronteiras de decisão lineares.

Conhecendo a probabilidade de a classe  $\omega_k$  ser a correta, tomando em conta o vetor de características  $\mathbf{x}$ , e o custo de tomar a ação  $\alpha_n$  quando a classe correta é  $\omega_k$ , é possível determinar o risco associado à escolha da ação  $\alpha_n$  como sendo a soma dos custos  $\lambda(\alpha_n|\omega_k)$  pesados pela probabilidade  $\hat{P}(\omega_k|\mathbf{x})$ . Logo, conhecendo o vetor  $\mathbf{x}$ , o risco associado à ação  $\alpha_n$  é obtido por

$$R(\alpha_n|\mathbf{x}) = \sum_{k=1}^K \lambda(\alpha_n|\omega_k) \hat{P}(\omega_k|\mathbf{x}). \quad (3.44)$$

A ação a escolher,  $\alpha$ , será aquela que minimize este risco, sendo esta dada por

$$\alpha = \arg \min_{n=1, \dots, N} \sum_{k=1}^K \hat{P}(\omega_k|\mathbf{x}) \lambda(\alpha_n|\omega_k). \quad (3.45)$$

## Capítulo 4

# Setup experimental e conceção gráfica do jogo

Este capítulo descreve o *framework* de desenvolvimento bem como os principais módulos desenvolvidos (ver figura 4.1). O sistema pode ser dividido em 3 componentes principais: Aquisição de biosinais, processamento de dados e jogo (estímulos visuais e gráficos).

### 4.1 Aquisição de dados EEG

A aquisição de dados EEG foi feita em ambiente de Simulink em funcionamento de tempo real, assente no driver e bibliotecas Highspeed Online Processing blockset, desenvolvido pela “gtec” [Gtec, ]. Esta biblioteca de blocos “Simulink” permite a recolha de dados e a configuração do amplificador, utilizado nas experiências realizadas.

A figura 4.2 mostra o diagrama de blocos do modelo “Simulink”. O bloco de aquisição “g.USBamp” tem como saída os sinais EEG, permitindo o seu processamento pelos outros blocos no diagrama. Este bloco fornece uma interrupção de tempo (instantes de amostragem), que garante o funcionamento em tempo real [Guger et al., 2009].

O bloco “onlineClassification” tem dois modos de funcionamento, treino e jogo. Faz o processamento e classificação de dados e transmite a deteção do comando via UDP/IP para o jogo desenvolvido numa aplicação externa. No modo de treino, é enviado o código relativo ao estímulo para o qual a pessoa deve olhar. Ao mesmo tempo, os dados EEG são etiquetados com esse código (conjuntos de dados usados para posteriormente construir os modelos de classificação). No modo de jogo, o código enviado é o detetado *online* pelo classificador. A classificação é efetuada no modo de jogo no fim de cada época com estímulo alvo.

Os dados adquiridos são guardados num ficheiro e ficam disponíveis para processamento posterior *offline*.

A figura 4.3 mostra o *setup* experimental. Os sinais são medidos recorrendo a eléctrodos passivos, montados na touca g.EEGcap de acordo com o sistema internacional 10-20 estendido, conforme pode ser observado na figura 4.3. Foi usado gel condutor para garantir uma baixa impedância no contacto dos eléctrodos com o escalpe. A referencia foi colocada na orelha, ponto A1 ou A2, já o eléctrodo terra foi montado na posição POz. O eléctrodo terra foi escolhido de modo a ser próximo dos canais EEG, com o objetivo de estar sob o efeito de ruído proveniente de fontes semelhantes, com vista a reduzir o mesmo, sendo escolhido a posição POz, visto que se encontra na fronteira da zona occipital e parietal. Foram usados os canais na zona occipital, para a medida do sinal EEG. Sendo estes canais Oz, O1 e O2. Os sinais adquiridos são amplificados pelo amplificador g.USBamp da “gtec”. Este permite a

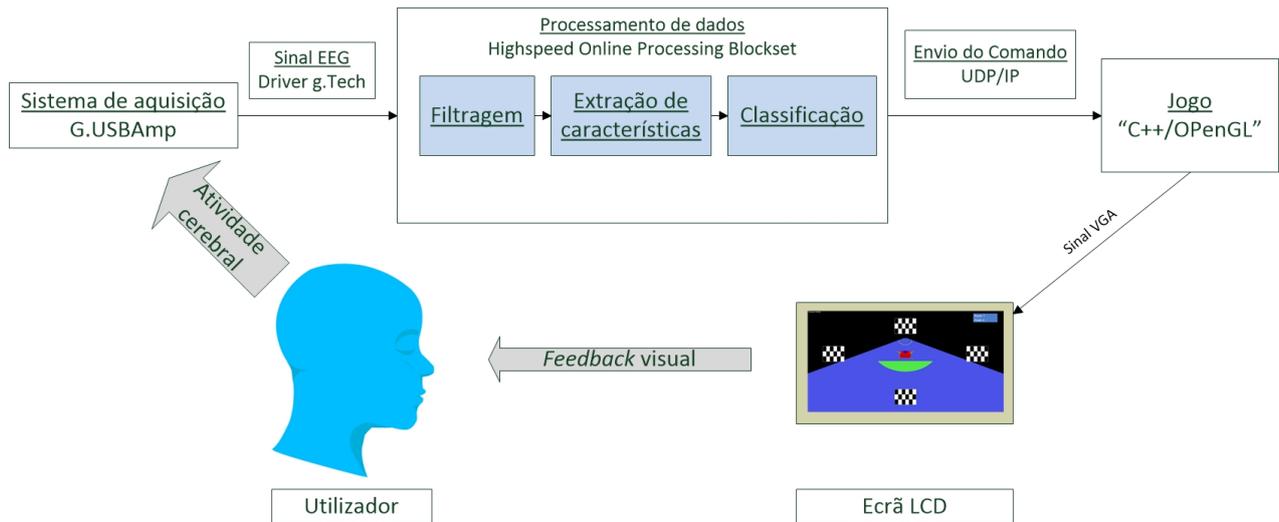


Figura 4.1: Arquitetura do sistema.

amplificação de biosinais de 16 canais distintos amostrados a  $256Hz$ . Permite também a aplicação de um filtro Notch, à frequência de  $50Hz$ , com vista a eliminar componentes de ruído geradas pela rede elétrica, bem como filtros passa banda.

## 4.2 Jogo Spacecraft

O jogo “Spacecraft” original [Parafita et al., 2013] foi utilizado para testar o controlo com SSVEP. Consiste numa nave a percorrer uma pista, onde aparecem obstáculos, dos quais o jogador se tem que desviar. A figura 4.4 mostra o jogo em funcionamento. São considerados quatro comandos distintos: mover a nave para a esquerda (figura 4.4a), mover a nave para a direita (figura 4.4b), seguir em frente (figura 4.4c) e saltar (figura 4.4d). A cada comando está associado um obstáculo, caso a intenção do utilizador (comando) seja corretamente detetada pelo classificador, a nave supera esse obstáculo. O contador de pontos é incrementado neste caso. Caso o comando seja errado, a nave colide com o obstáculo, incrementando o contador de colisões. O comando é escolhido pelo utilizador olhando para um dos quatro estímulos visuais presentes. Os estímulos estão dispostos no ecrã de acordo com o comando que representam, por exemplo, estímulo à direita da pista representa o comando "desvia nave para a direita", aplicando-se a mesma lógica para os estímulos de cima e à esquerda. O estímulo de baixo representa o comando: seguir em frente. A conceção dos elementos do jogo foi feita recorrendo ao “OpenGL” descrito na secção 4.2.1.

O código do jogo encontra-se estruturado em diferentes módulos distintos pelo seu contexto. A figura 4.5 mostra a estrutura hierárquica destes módulos.

Certos módulos implementam tarefas utilizadas por outros módulos. O módulo “rp\_glut” encarrega-se do desenho de diferentes primitivas em “OpenGL” bem como outras funcionalidades tais como a criação de uma janela, limpeza do *framebuffer* e gestão de eventos de teclado e rato. É neste módulo que se encontram as funções que desenharam a nave espacial e a pista. O módulo “rp\_sound” contém

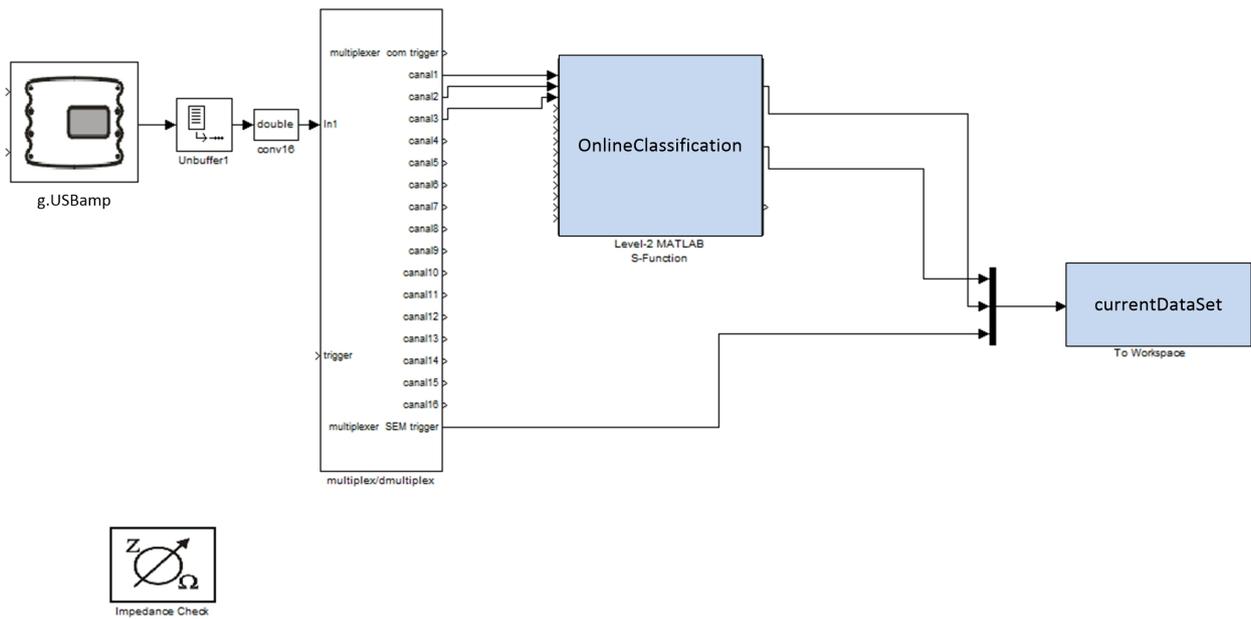


Figura 4.2: Diagrama de “simulink” usado na aquisição.

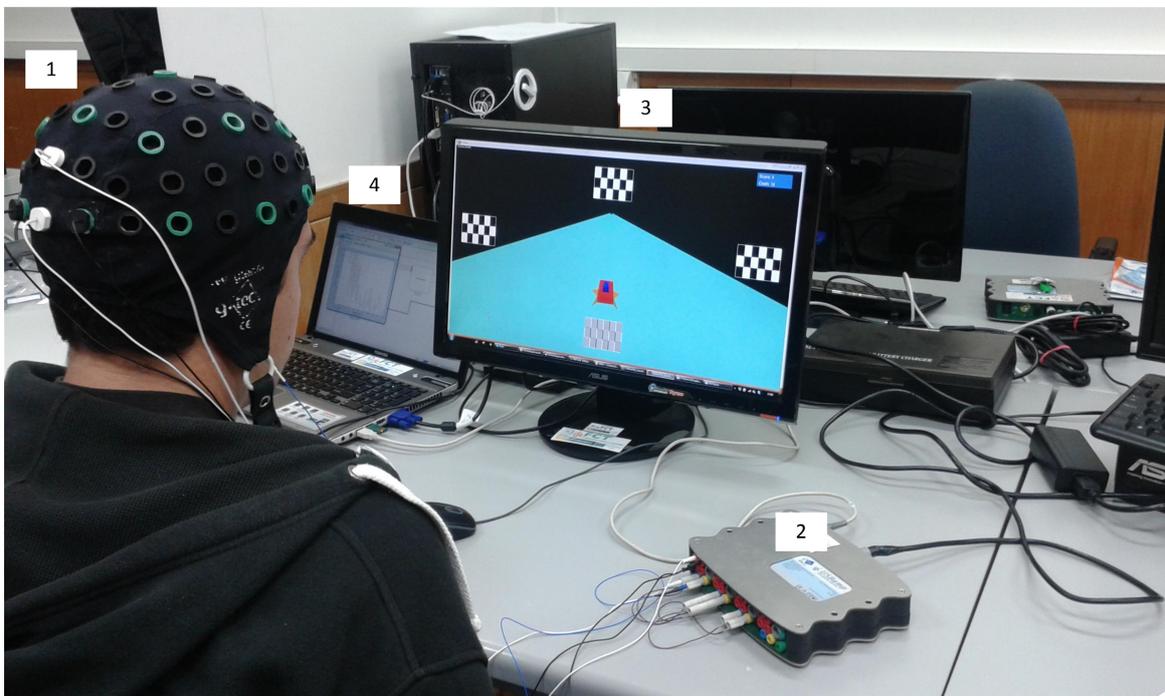


Figura 4.3: Ambiente de aquisição de dados. 1) gEEGCap com elétrodos montados. 2) Amplificador g.USBamp. 3) Ecrã a correr o jogo. 4) Computador que corre o jogos e faz processamento de dados.

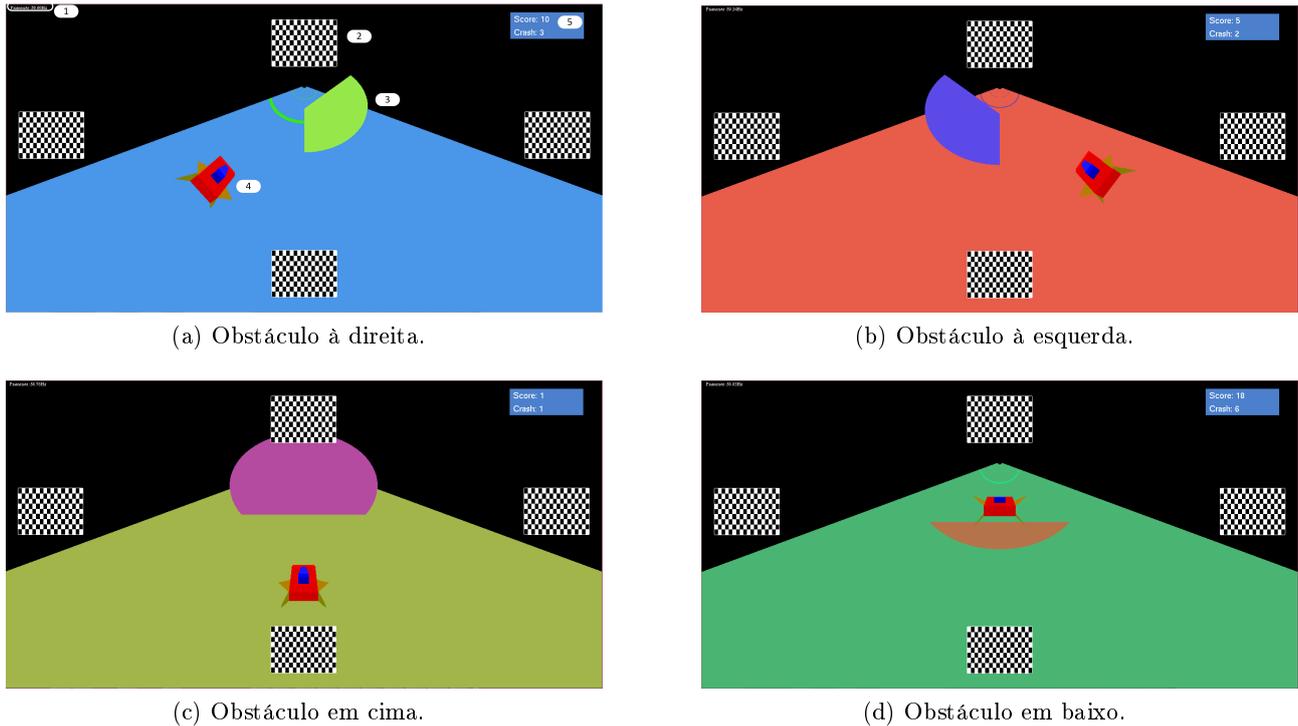


Figura 4.4: Capturas de ecrã do jogos Spacecraft. 1) Taxa de refrescamento do jogo. 2) Estímulo de padrões alternados. 3) Obstáculo. 4) Nave espacial. 5) Pontuação e quantidade de colisões do jogador.

funções para iniciar o som do jogo, carregar ficheiros de som e reproduzir os mesmos durante o jogo. Utiliza a biblioteca “OpenAL”, que é uma biblioteca aberta para interface com dispositivos de áudio. O módulo “rp\_tasks” contém funções analíticas de cálculo, nomeadamente, algoritmos de classificação e cálculo de normas e distâncias. Outros módulos de interface com *buffers* de dados, filtragem e aquisição foram removidos do jogo, visto que estas tarefas são feitas em “simulink”.

O módulo “Core” define memória partilhada entre todas as *threads*, bem como constantes úteis à implementação.

Outros módulos implementam *threads* com diferentes responsabilidades. Entre estas responsabilidades encontram-se aquisição de sinal e estimação espectral, no entanto estas foram excluídas no âmbito deste trabalho. A figura 4.6 mostra a organização das *threads* do jogo.

A *thread* que implementa o jogo utiliza os módulos descritos anteriormente para desenhar a nave e pista, desenhar os obstáculos e estímulos visuais e fazer cálculos necessários para o movimento da nave. O módulo de *software* responsável por esta função é o “GameTask”. A execução das primitivas gráficas do jogo é sincronizada com o pulso vertical do monitor. Isto é feito dando indicação ao OpenGL para suspender a execução desta *thread*, até à receção do pulso seguinte. Deste modo, a *thread* responsável pelo desenho do jogo tem uma frequência de execução semelhante à taxa de refrescamento do monitor, o que permite evitar erros devido a sobreposição de *frames*.

O módulo “SimulinkControlThread” implementa uma *thread* de comunicação recorrendo ao módulo “SocketMannagerUDP”. Estes módulos são descritos na secção 4.2.3. A *thread* de comunicação comunica com a *thread* do jogo através de memória partilhada definida no módulo “Core”.

O módulo “DualStateStimulus” implementa os estímulos visuais descritos. A cada estímulo corresponde um objeto e uma *thread* dedicada a qual é responsável pelas operações de atualização do mesmo.

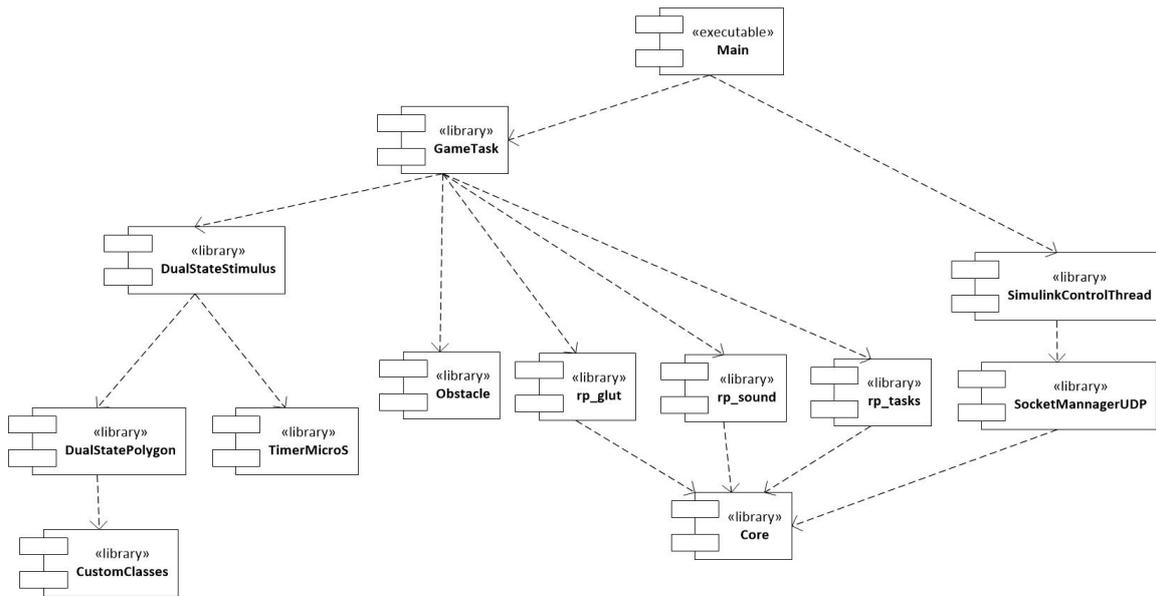


Figura 4.5: Módulos de software do jogo Spacecraft.

Os objetos correspondentes a cada estímulo são instanciados na *thread* do jogo, com vista a garantir a visibilidade. As *threads* dedicadas de cada estímulos são lançadas a partir desta e partilham o objeto associado a cada estímulo, ou seja atualizações feitas ao estímulo na *thread* dedicada são visíveis no jogo.

### 4.2.1 OpenGL

O OpenGL é uma interface para programação de aplicações (API), cujo objetivo é fazer a interface com o *hardware* gráfico (GPU). Consiste numa biblioteca de funções e tipos de variáveis que permitem o desenho de primitivas gráficas 3D, com recurso a aceleração por *hardware*. É uma API multiplataforma e independente da linguagem de programação que a utiliza. Com vista a manter esta versatilidade, não existem funções de interface com o sistema operativo, sendo estas fornecidas por bibliotecas distintas como o GLUT e freeGLUT.

O funcionamento do OpenGL baseia-se numa máquina de estados, onde antes de serem chamadas as primitivas são definidos os seus parâmetros. A sequência de operações efetuadas pela API tem o nome de pipeline gráfica. As versões modernas permitem a programação de componentes desta pipeline, no entanto deixaram de suportar modo imediato, ou seja o desenho gráfico passou a estar a cargo de objetos presentes em memória gráfica, deixando de ser possível a passagem direta de vértices e cores, entre outros parâmetros, para o pipeline gráfico. Embora esta abordagem torne o desenho mais eficiente, torna a utilização da API mais complexa, logo a versão anterior ainda é muito usada. No contexto do trabalho realizado, foi usada a versão anterior do OpenGL, também conhecida como *legacy* OpenGL.

O desenho de uma primitiva gráfica deve ser precedido pela definição dos seus parâmetros, na máquina de estados do OpenGL. Entre estes parâmetros, destacam-se o parâmetro de preenchimento da primitiva, cor ou textura, os vértices que a constituem, o tipo e as transformações a que está sujeita.

São permitidos vários tipos de primitivas, entres os quais estão triângulos, quadrados, pontos e linhas. A forma da primitiva é diretamente definida pelos vértices fornecidos, desde o início da

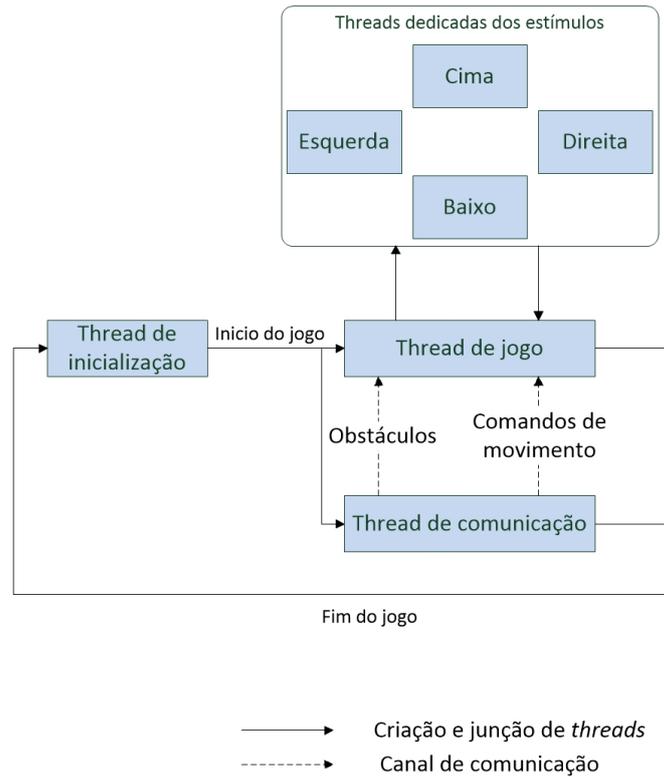


Figura 4.6: Organização das *threads* do jogo.

configuração da primitiva, marcado pela chamada de uma função própria, até ao fim. Quando finalizada a configuração da primitiva a API encarrega-se do seu desenho.

O preenchimento pode ser definido de duas maneiras, com um parâmetro de cor para cada vértice ou a partir de uma textura. Os valores de cor, ou pixeis da textura podem também conter informação de transparência. As texturas são armazenadas na memória gráfica e são utilizadas, associando os seus pontos aos vértices.

O OpenGL permite a definição de matrizes de transformação. Estas são armazenadas em pilhas de matrizes, cujas transformações são aplicadas aos vértices, quando estes são enviados para a máquina de estados. As transformações possíveis dividem-se em transformações de projecção e geometria. As transformações de geometria definem deslocamento, rotação e escala dos vértices aos quais são aplicadas. As transformações de projecção definem os parâmetros da câmara virtual, associada a um plano de projecção.

As primitivas desenhadas são projetadas para o plano de projecção definido pela câmara virtual. Sendo desenhadas para um espaço de armazenamento na memória gráfica, definido como *framebuffer*. A cada refrescamento do ecrã, o *framebuffer* é transferido para o monitor. O OpenGL permite, também, a utilização de um duplo *framebuffer* com vista a permitir o paralelismo entre o desenho de primitivas gráficas e a atualização do monitor.

Embora a API não permita a interface com o sistema operativo, para criar janelas e obter comandos de rato e teclados, estas operações são garantidas por bibliotecas externas. No entanto certas operações são demasiado dependentes da plataforma onde são efetuadas, portanto nem mesmo bibliotecas como freeGLUT as disponibilizam. Um exemplo de uma operação deste género é a sincronização da execução do código com o pulso *VSync* do monitor. Para efetuar estas operações, é necessário recorrer a exten-

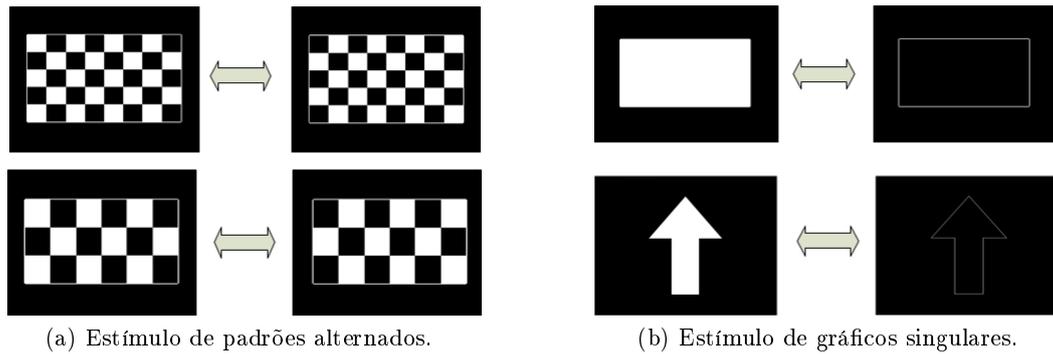


Figura 4.7: Exemplos de estímulos implementados.

sões implementadas pela plataforma. O carregamento destas extensões é simplificado pela biblioteca OpenGL Extension Wrangler Library GLEW.

#### 4.2.2 Estímulos visuais

Foram implementados estímulos visuais repetitivos configuráveis com vista a evocar a resposta SSVEP no utilizador. É possível definir estímulos de dois tipos: estímulos de gráficos singulares (figura 4.7b) e estímulos de padrões alternados (figura 4.7a). Os estímulos de gráficos singulares correspondem a um polígono preenchido por uma de duas cores. A cor de preenchimento reflete o estado em que o estímulo se encontra. É permitida a definição de ambas as cores de preenchimento, no entanto no trabalho realizado foram usados estímulos que comutam entre a cor branca e preta. Os estímulos de padrões alternados apresentam um padrão em xadrez preto e branco, onde é possível escolher o número de casas presentes. As casas a branco comutam para preto quando ocorre uma alteração de estado do estímulo. Ambos os tipos de estímulos visuais necessitam da definição de parâmetros geométricos como a sua posição, tamanho e rotação. É também necessário definir a frequência de comutação dos estímulos, porém, no caso dos estímulos de padrões alternados, esta frequência deve ser metade da frequência que se pretende obter como resposta, visto que a resposta evocada por estes corresponde a uma comutação de estado, como descrito na secção 2.1.

Os estímulos foram implementados em C++, com vista a utilizar as ferramentas desta linguagem no que toca à programação orientada a objeto. Isto permite uma mais fácil integração destes estímulos no jogo “Spacecraft”, bem como um bom desacoplamento dos restantes módulos de software. A componente gráfica foi implementada recorrendo às bibliotecas “OpenGL”.

Cada estímulo é representado por uma classe (ver figura 4.8) com atributos que indicam o seu período de comutação, estado e outras propriedades relevantes, bem como métodos que desenham os estímulos em “OpenGL” e mudam o seu estado. O efeito de piscar é conseguido alternando entre o estado *on* e *off*. O estado do estímulo é definido pelo atributo “isOn”. Em cada um destes estados o estímulo é desenhado de maneira diferente. Cada estímulo pode ser ativado e desativado. Caso o estímulo se encontre desativado a atualização do estado não é efetuada. Estes estímulos tem a opção de correr numa *thread* dedicada, logo têm parâmetros de identificação da mesma.

O período de comutação, parâmetro “blinkPeriodMS”, define o tempo em milissegundos entre comutações do estado do estímulo. Este é inicializado no construtor, através da frequência desejada, pela equação 4.1.

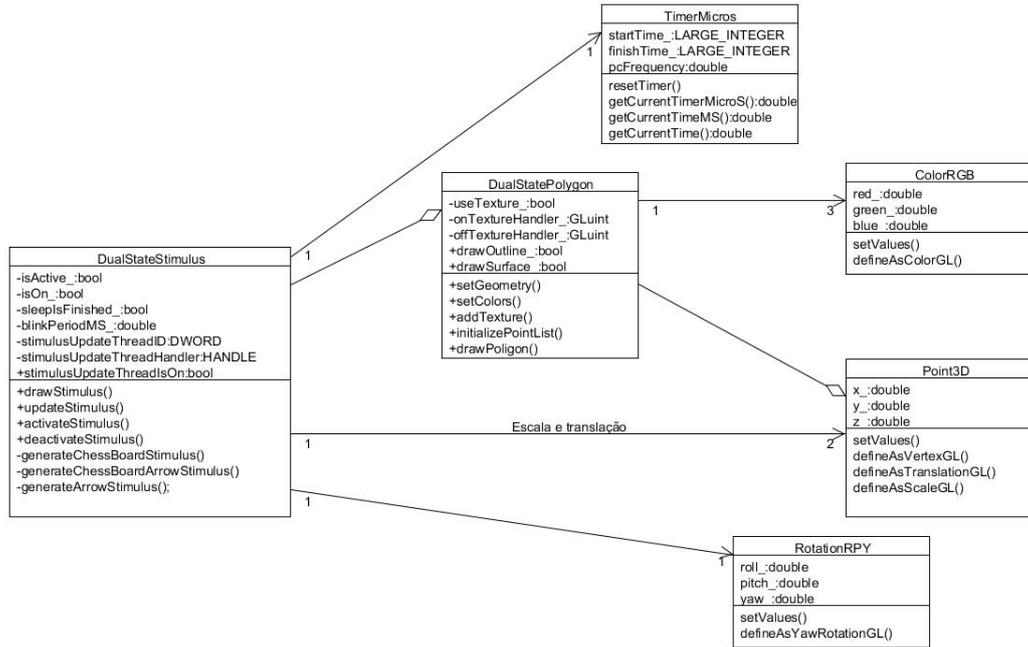


Figura 4.8: Diagrama de classes para os estímulos.

$$periodMS = \frac{500.0}{frequencyHz} \quad (4.1)$$

Cada instanciação desta classe contém um temporizador, que é usado na comutação de estado. Esta instanciação também armazena o instante seguinte de comutação. Cada vez que o temporizador atinge o instante de comutação, o estado é comutado, originando o efeito de piscar à frequência desejada. De seguida, é determinado o próximo instante de comutação, somando-lhe o período do estímulo, e o processo repete-se. Este processo é feito pelo método “updateStimulus” numa *thread* dedicada, com a prioridade máxima possível no sistema “Windows”, para cada estímulo. Esta *thread* suspende a sua execução durante 80% do tempo que falta para a comutação seguinte, o tempo restante é passado em espera atarefada (*busy-waiting*). A suspensão é efetuada com vista a que a *thread* não sobrecarregue o sistema, no entanto para minimizar erros de temporização causados por períodos de suspensão mais longos que o desejado, foi introduzida a fase de espera atarefada. Caso o tempo de espera seja inferior a 5ms a suspensão da *thread* é contornada, pelos motivos mencionados. A figura 4.9 ilustra o processo descrito.

O temporizador é implementado na classe “TimerMicros”, explicada com maior detalhe mais à frente. Se o valor de tempo medido pelo temporizador for negativo, este é reiniciado e o próximo instante de ativação é definido com o período do estímulo. Este processo é realizado com vista a evitar erros devido a *overflow* do temporizador, ou seja erros causados por o temporizador atingir um valor superior ao que é possível guardar na arquitetura em questão, voltando ao valor zero.

É possível observar que esta classe depende de outras classes mais simples, “Point3D”, “ColorRGB” e “RotationRPY”. Estas simplificam a interface com o “OpenGL” e permitem guardar os atributos geométricos para o desenho de primitivas. A classe “Point3D” é usada para definir a escala e translações a fornecer ao pipeline do “OpenGL”, sob a forma de matrizes de transformação. Esta classe contém

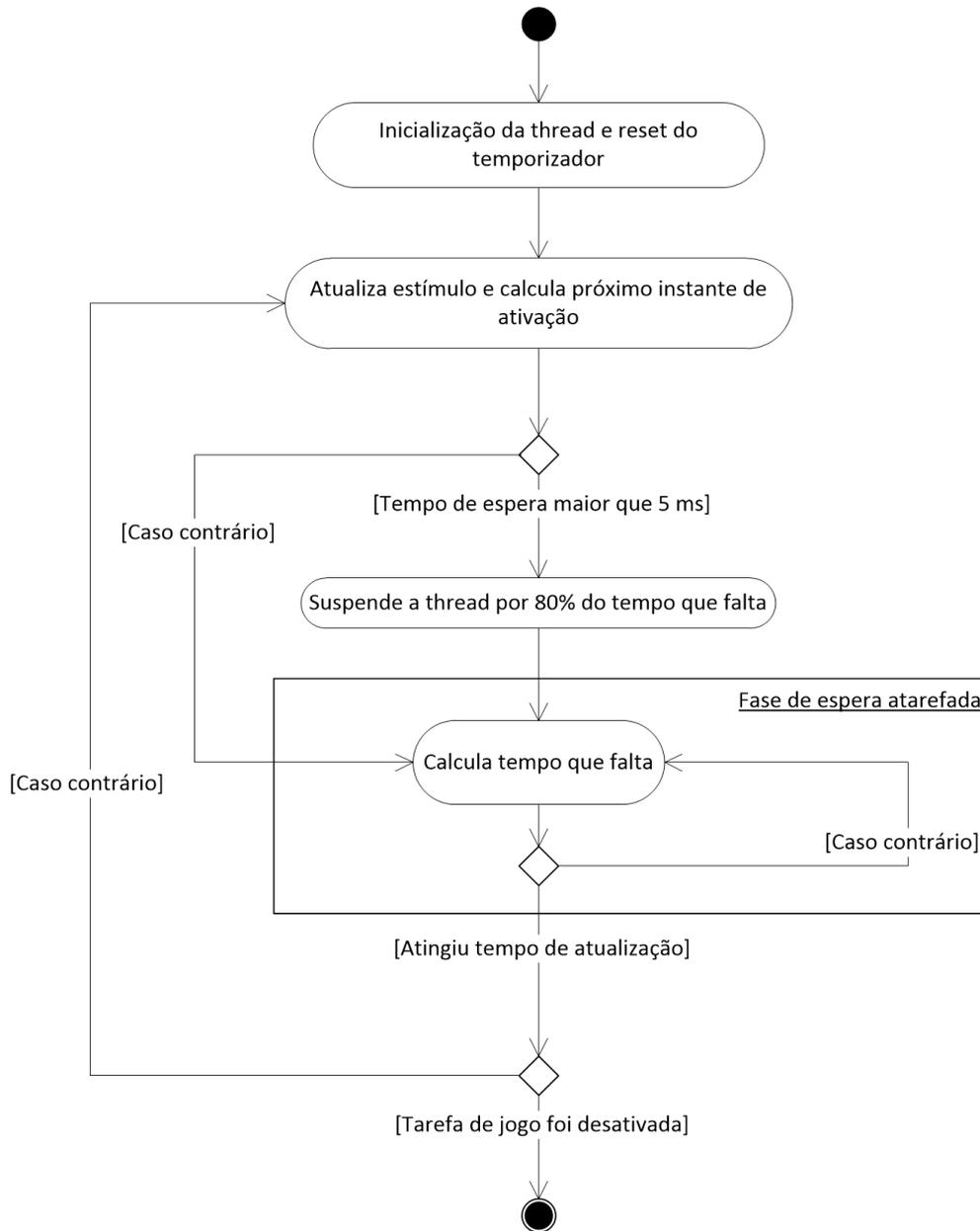


Figura 4.9: Diagrama de atividades da *thread* dedicada a um estímulo.

métodos que se encarregam destas tarefas, bem como para a definir como vértice de uma primitiva. Da mesma forma, a classe “RotationRPY” permite definir rotações. A classe “ColorRGB” permite definir a cor de primitivas.

A classe “DualStatePolygon” representa um polígono com duas opções de desenho. Estão-lhe associados 3 parâmetros de cor: cor *on*, cor *off* e cor das bordas do polígono. Armazena também duas texturas, uma para cada opção de desenho. A opção de desenho desejada é fornecida como parâmetro de entrada do método de desenho da classe. O preenchimento do polígono pode ser feito pelas texturas armazenadas ou pela cor, dependendo do modo de funcionamento escolhido no parâmetro “useTexture”.

Cada estímulo contém um conjunto de “DualStatePolygons”, que definem a sua forma. É possível definir um padrão em xadrez a partir de vários polígonos com cores diferentes, bem como um único polígono com texturas com o padrão correspondente. A segunda abordagem foi optada no trabalho realizado, com vista a reduzir o número de vértices a enviar à pipeline gráfica, melhorando o desempenho. Estímulos simples podem ser definidos a partir de um polígono com duas cores, uma para cada estado.

Estes estímulos foram implementados para serem apresentados em ecrãs de computador. Neste contexto, estão limitados à sua taxa de refrescamento, tipicamente  $60Hz$ . Logo é preciso cuidado na utilização de estímulos com frequências não sub-múltiplas desta taxa. A figura 4.10 mostra a amostragem de dois estímulos a frequências diferentes por um ecrã com a taxa de refrescamento típica. É possível observar que para o estímulo de  $10Hz$ , a quantidade *frames* para cada estado são regulares. No entanto, o estímulo de  $13Hz$  apresenta irregularidades no número de *frames* antes de cada comutação. Este fator pode originar artefactos visuais que geram respostas SSVEP à frequência errada. Na fase da análise dos resultados serão apresentadas respostas SSVEP para uma gama de frequências, obtidas de um utilizador, com vista a perceber a influência deste fator na resposta do utilizador.

#### 4.2.2.1 Temporizador

Para garantir o bom funcionamento dos estímulos, bem como das outras tarefas que requerem temporização, foi necessário implementar um temporizador. Este foi implementado recorrendo ao contador de performance do Windows. Este contador utiliza o registo TSC (Time Stamp Counter), presente em processadores Intel e AMD. Caso este registo não esteja disponível, ou seja atualizado a uma frequência variável, o sistema operativo encarrega-se de encontrar um contador alternativo [MSDN, a]. O valor do contador é obtido a partir de uma função presente na biblioteca de funções do “windows”. No entanto, para obter o valor em segundos é necessário obter a frequência do contador. A mesma biblioteca define uma função para este efeito.

Este temporizador foi implementado numa classe, com métodos para reiniciar a contagem do tempo, e obter o tempo que passou, em segundos, milissegundos ou microssegundos, desde a sua última reinicialização. Na Figura 4.8 é possível observar que a classe do temporizador tem como atributo a frequência do contador, “pcFrequency\_”, a qual é usada para obter o tempo em segundos. Esta frequência é obtida quando um temporizador é instanciado. O atributo “startTime\_” guarda o valor do contador no instante em que é chamado o método “resetTimer” ou quando é chamado o construtor da classe. Cada vez que é chamado o método “getCurrentTimeMicroS”, “getCurrentTimeMS” ou “getCurrentTime”, o

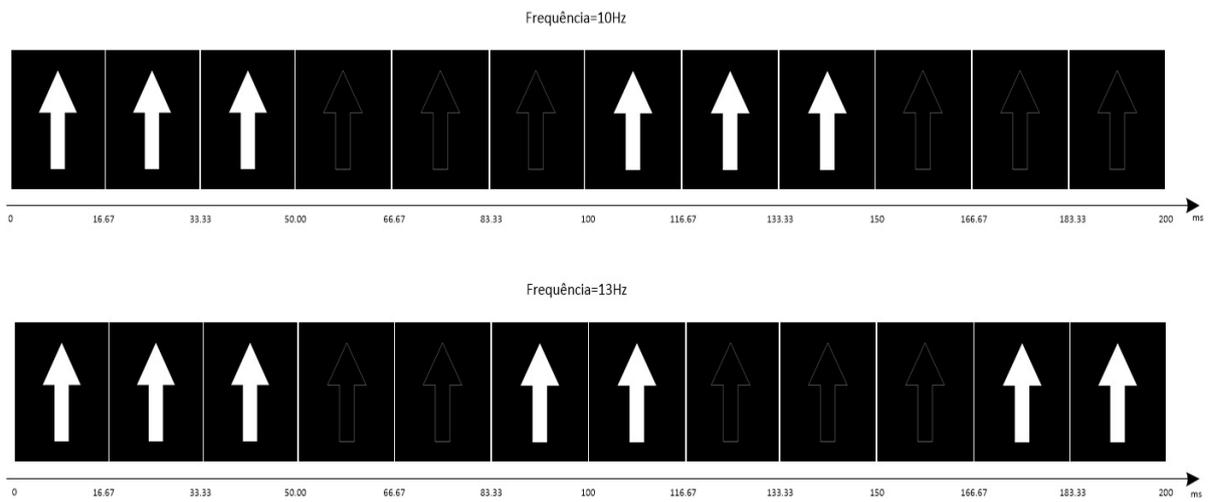


Figura 4.10: Exemplos de estímulos, a 10 Hz (em cima) e 13 Hz (em baixo), apresentados num ecrã de 60 Hz

valor do contador, nesse instante, é guardado no atributo “finishTime\_” e o tempo que passou é calculado pela equação 4.2 onde “timeConversion” é 1 para o calculo em segundos, 1000 para milissegundos e 1000000 no caso de micro-segundos.

$$elapsedTime = timeConversion \frac{(finishTime - startTime)}{pcFrequency} \quad (4.2)$$

Listagem 4.1: Funções da biblioteca windows para obtenção do valor do contador de performance e frequência do mesmo:

```
QueryPerformanceFrequency (LARGE_INTEGER * lpFrequency)
QueryPerformanceCounter (LARGE_INTEGER * lpPerformanceCount)
```

### 4.2.3 Comunicação UDP

A aquisição e classificação dos dados EEG é feita em “simulink”, logo é necessária comunicação entre o jogo e o modelo que faz a aquisição e classificação. Para tal, foram usadas *sockets* UDP.

Em C++ a implementação é feita recorrendo à API “Winsock”. Esta fornece funções e estruturas de dados para interface com *sockets* no sistema operativo Windows. Foi implementada uma classe com vista a simplificar a interface com a API mencionada. Esta classe está preparada para funcionar como *server*, permanecendo à espera de um pedido de comunicação externo.

Antes de usar as ferramentas da API “winsock”, é necessário inicializar as bibliotecas de ligação dinâmicas da mesma. Esta função é utilizada no método “initializeWinsock” da classe implementada. A função “finalizeWinsock” encarrega-se de libertar os recursos reservados para a utilização das *sockets*. Deve ser utilizada, sempre que a comunicação UDP deixe de ser necessária.

A classe mantém o registo do porto de comunicação bem como das estruturas de endereçamento. Contém um *buffer* de receção, onde ficam guardados os ultimo dados recebidos. O acesso ao *buffer* é feito recorrendo à função “getDataFromBuffer”. O envio de dados é feito a partir da função “sendData”, e requer a existência de um *buffer* externo à classe, do qual serão enviados os dados.

O método “initializeAdress” inicializa as estruturas de endereçamento armazenadas, a partir do porto e IP da socket. Já o método “startSocket” torna a *socket* apta para receção, fazendo o *bind* ao endereço a esta.

Foi implementada uma *thread* de comunicação com o “simulink” através de *sockets* UDP, que utiliza a classe descrita. Esta *thread* recebe comandos de movimento e obstáculos do “simulink” e transmite-os para o jogo. Cada vez que o jogo é iniciado é lançada a *thread* de comunicação que fica à espera de um pedido de sincronismo do “simulink”, ao chegar este pedido são enviadas as frequências dos estímulos e o jogo está pronto para receber o primeiro obstáculo. A informação recebida do “simulink” vem sob a forma de um inteiro, o qual pode indicar o término da recolha de dados ou pedido de sincronismo. O sinal que indica o tipo de obstáculo ou classe é ambíguo, sendo a interpretação do comando recebido feita através do estado de espera em que esta *thread* se encontra. Logo esta pode estar à espera de um obstáculo, comunicando ao jogo que deve gerar um novo obstáculo, ou, no caso da espera por um resultado de classificação, é comunicado ao jogo que deve mover a nave para a direção recebida.

Com vista a minimizar o esforço computacional foi implementada uma função que bloqueia a *thread* de comunicação, até que cheguem novos dados. Com este método poupa-se a carga computacional de fazer *busy-waiting* até ao momento de receção.

Em “matlab”, a implementação é feita utilizando um objeto do tipo “udp”. Este objeto é gerado utilizando a função “obj = udp(‘rhost’,rport)” [Mathworks, 2012a], onde “rhost” contém o endereço do anfitrião e “rport” contém o porto, para onde serão enviados os dados. O envio e recepção de dados é feito a partir dos mecanismos de escrita para ficheiro, presentes neste ambiente de programação.

Listagem 4.2: Lista dos métodos da classe que faz a interface com sockets UDP:

```
static int initializeWinsock ()
static int finalizeWinsock ()

int initializeAdress ()
int startSocket ()

int getReceivingBufferLength ()
int receiveDataToBuffer ()
int getDataFromBuffer(char* newBuffer , int newBufferSize)

int sendData(const char* sendingBuffer , int sendingBufferSize)
int setSocketBlockingMode(bool blockingIsActive)
int closeSocket ();
```

#### 4.2.4 Obstáculos

No início de cada época de classificação, é recebido do “simulink” o código do obstáculo, a colocar na pista ao longo do eixo perpendicular ao visor do utilizador. A posição do obstáculo gerado é determinada pela equação 4.3, onde *playerPosition* é a posição do jogador no instante atual, *playerVelocity* é a velocidade do jogador, *actionTime* é o tempo que a nave demora a superar o obstáculo partindo da zona de decisão e *timeToObstacle* é o tempo que a nave demora a chegar à zona de decisão do obstáculo, ligeiramente antes do mesmo. Durante a época de classificação a nave aproxima-se do obstáculo até chegar à zona de decisão. Nesta altura, o jogo recebe um comando de classificação e move a nave, de acordo com o comando recebido, superando ou chocando com o obstáculo.

$$obstaclePosition = playerPosition + playerVelocity(actionTime + timeToObstacle) \quad (4.3)$$

Na versão anterior do jogo, os obstáculos eram implementados através de vários polígonos. Quando ocorria uma colisão, os polígonos na área de colisão desciam até desaparecerem através da pista. Com vista a obter um melhor desempenho neste processo, foi implementada um nova versão para os obstáculos. Estes são baseados em quadrados, com a altura e o dobro da largura da pista, preenchido

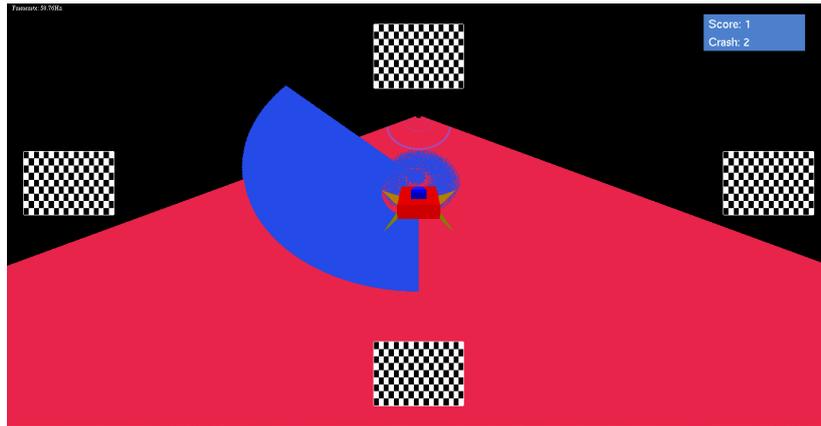


Figura 4.11: Situação de colisão da nave com um obstáculo.

por uma textura. Os canais de cor da textura correspondem à cor branca em todos os píxeis, no entanto é usado um canal de transparência que define a forma do obstáculo. Os quatro tipos de obstáculos possíveis são apresentados na figura 4.4. Os obstáculos são implementados numa classe, que guarda o seu tipo, a sua posição e um inteiro que permite aceder à sua textura na memória gráfica, através das funções do “OpenGL”. Contém métodos para desenhar o obstáculo e para testar a colisão com a nave. Caso seja detetada uma colisão, são determinados os píxeis com valores de transparência não nula afetados. A estes o valor da transparência é definido como nulo e é gerado um conjunto de pontos com uma determinada velocidade, de modo a dar a sensação que foram projetados do obstáculo. O efeito é representado na figura 4.11.

# Capítulo 5

## Classificação de SSVEP

### 5.1 Protocolo experimental de treino

Foi definido um protocolo experimental de treino para recolha de dados etiquetados. Este define a sequência de estímulos visuais alvo (estímulo para o qual o utilizado deve olhar), gerando a mesma quantidade de dados por cada estímulo. Este protocolo impede que o utilizador fique demasiado cansado. Para tal foram definidos períodos com estímulo de  $T_{on}$  segundos. Os períodos com estímulo são intercalados por etapas de descanso, de duração  $T_{off}$ , as quais permitem ao utilizador abstrair-se dos estímulos, bem como preparar-se para o próximo estímulo alvo. A Figura 5.1 faz uma ilustração do processo descrito. Este processo é repetido 10 vezes, com vista a obter dados referentes a 10 períodos para cada estímulo, obtendo assim 40 épocas (sequências de dados com uma determinada duração) com estímulo alvo e 40 épocas sem estímulo alvo.

Os códigos dos eventos (obstáculos, comando de classificação) são guardados numa matriz, onde a primeira linha representa o instante a que devem ser enviados, e segunda linha os eventos a enviar. Ao chegar o instante de envio, o código é enviado pelo bloco de processamento de dados da figura 4.2 à *thread* de comunicação do jogo. O envio é feito através do módulo de comunicação descrito na secção 4.2.3. Logo o jogo recebe o estímulo alvo através de um pacote UDP, enviado pelo bloco, o qual é convertido num obstáculo a apresentar ao utilizador. Estes eventos são também usados pelo bloco de processamento de dados para marcar os dados com o estímulo alvo de cada época.

### 5.2 Pré processamento

O sinal EEG é filtrado por um filtro *butterworth*, de grau 3, passa banda com frequência de corte inferior de 2Hz e frequência de corte superior de 30Hz. Para eliminação de ruído gerado pela rede elétrica, aplica-se adicionalmente um filtro Notch à frequência de 50Hz.

### 5.3 Extração de características

Foram testados três algoritmos de extração de características, com vista a comparar a sua eficácia. Nesta secção será descrita a sua utilização nos dados EEG.

#### 5.3.1 Método de Goertzel

O método de Goertzel determina os valores da transformada de Fourier de um sinal para frequências predeterminadas, em contraste com o espectro completo. A potência espectral é obtida, para cada valor  $F(k)$ , calculando  $P(k) = |F(k)|^2$ . No contexto do trabalho realizado, é necessário o cálculo do

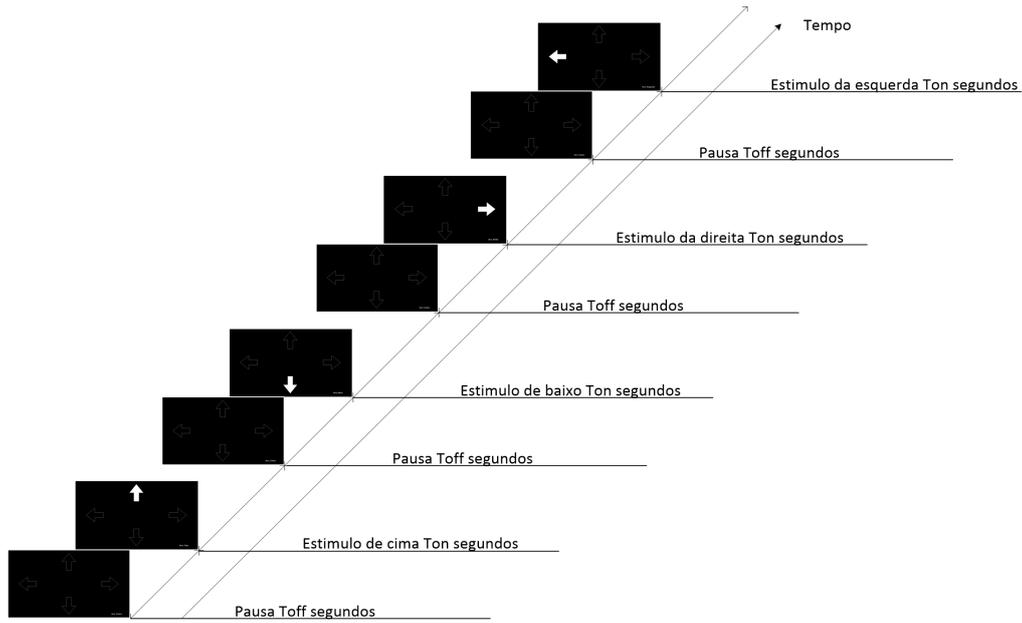


Figura 5.1: Ilustração do protocolo de treino adotado para os estímulos.

espectro para quatro frequências base, bem como o segundo harmónico de cada, logo este algoritmo apresenta-se como um bom candidato para extração de características. Foi utilizada a função “goertzel” do “matlab” para aplicar o algoritmo. Esta recebe como parâmetros o sinal e os índices das frequências, para as quais se deseja calcular o valor do espectro. Os índices  $k$  são determinados por

$$k = \frac{N}{F_s}, \quad (5.1)$$

onde  $N$  é o número de amostras temporais do sinal e  $F_s$  é a frequência de amostragem, ou seja  $256Hz$ .

Para cada índice  $k$ , (3.21) é aplicada recursivamente até determinar  $v_k(N)$ , por fim, este valor é usado para determinar  $y_k(N)$  através de (3.22), o qual corresponde ao valor da DFT para o índice  $k$ , ou seja  $X(k)$ .

### 5.3.2 Método de Welch

O método de Welch permite reduzir o ruído na estimação da potencia espectral do sinal, calculando a média dos espectros de vários segmentos deste. No entanto é sacrificada resolução no domínio da frequência, visto que cada segmento é menor que o sinal original. A segmentação do sinal é feita por uma janela de dimensão  $L$ . Foram testados diferentes valores para este parâmetro. A aplicação do algoritmo é feita a partir da função “pwelch” do “Matlab”, esta recebe o sinal, o tamanho da janela, o número de amostras sobrepostas o tamanho da FFT a calcular e a frequência de amostragem. Caso a dimensão da janela seja superior à dimensão da FFT ocorre o truncamento dos segmentos para o tamanho da FFT. Na situação inversa, são acrescentados zeros a cada segmento [Mathworks, 2012c]. Foi utilizada uma percentagem de amostras sobrepostas de 30% do tamanho da janela. A cada segmento é aplicada uma função janela de Hamming (figura 3.1a). Se o tamanho do sinal não permitir a divisão

num número inteiro de segmentos, o sinal é truncado, de modo a atingir o tamanho mais próximo, que seja sub-múltiplo de  $L$ .

Ambos os métodos descritos previamente baseiam-se na densidade de potência espectral, sendo aplicados sobre a média dos 3 canais utilizados. Estes geram um vetor de características, onde cada elemento corresponde ao valor  $P(f_a)$ , sendo  $f_a$  as frequências dos estímulos. Os segundos harmônicos das frequências alvo podem também ser incluídos no vetor, duplicando a dimensão do mesmo. Para estes algoritmos foi testado o desempenho com 1 harmónico e 2 harmónico, nestes vetores.

### 5.3.3 Método CCA

O método CCA tira proveito da correlação de vários canais nos sinais EEG. Foi usado para determinar um coeficiente de correlação canónica entre os canais EEG e um conjunto de sinais de referência. É extraído um coeficiente para cada frequência alvo. Os sinais de referência utilizados são dados por

$$y(t) = \begin{bmatrix} \sin(2\pi ft) \\ \cos(2\pi ft) \\ \vdots \\ \sin(2\pi Hft) \\ \cos(2\pi Hft) \end{bmatrix}^T, \quad (5.2)$$

onde  $f$  representa a frequência alvo. Estes sinais foram baseados em [Lin et al., 2007] e representam as primeiras componentes harmónicas da decomposição em série de Fourier da resposta SSVEP. Correspondem à matriz  $Y$  na secção 3.1.4. No trabalho desenvolvido foram testados valores de  $H$  de 1 e 2, visto que os harmónicos de ordem superior a este já são atenuados, tornando-se indiscrimináveis no espetro do sinal EEG.

O algoritmo CCA é implementado em “matlab” através da função “canoncorr”. Esta recebe como entradas o conjunto dos canais EEG,  $x(t)$ , bem como o sinal de referência  $y(t)$ . Devolve os coeficientes das combinações lineares que geram as variáveis canónicas bem como o coeficiente de correlação canónica associado a cada variável. O número de coeficientes obtidos é igual ao menor número de variáveis entre ambos os conjuntos, no entanto é escolhido como característica o coeficiente de valor mais elevado, visto que é o que tem uma maior capacidade descritiva [Lin et al., 2007].

Com o método CCA é gerado um vetor de características, onde cada elemento correspondente ao coeficiente de correlação canónica para cada frequência de alvo.

## 5.4 Classificação

As características extraídas pelos métodos da secção 5.3 são usadas na fase de decisão, com o objetivo de detetar o estímulo selecionado pelo utilizador (comando). A cada época está associado um vetor de características que será classificado. A figura 5.2 ilustra o processo de aplicação do classificador. O vetor de características tem o formato:

$$V = \begin{bmatrix} C_{cima} & C_{baixo} & C_{direita} & C_{esquerda} \end{bmatrix} \quad (5.3)$$

caso as características sejam extraídas pelos métodos de Goertzel e Welch, considerando apenas a frequência fundamental de cada estímulo. O valor  $C_{cima}$  corresponde à característica extraída para a frequência do estímulo de cima, acontecendo o mesmo para os restantes estímulos. Neste caso,  $V$  tem dimensões  $4 \times 1$ . A estrutura do vetor de características obtido com os métodos de Goertzel e Welch considerando o primeiro e segundo harmónico da frequência de cada estímulo é dada por

$$V = \begin{bmatrix} C_{cima} & C_{baixo} & C_{direita} & C_{esquerda} & C_{cima,2} & C_{baixo,2} & C_{direita,2} & C_{esquerda,2} \end{bmatrix}. \quad (5.4)$$

O valor  $C_{cima,2}$  representa a característica correspondente ao dobro da frequência do estímulo de cima, acontecendo o mesmo para os restantes estímulos. O vetor de características obtido através de (5.4) tem dimensões  $8 \times 1$ . Para o caso particular da classificação efetuada através do método da escolha da característica máxima considerando dois harmónicos, a característica escolhida é a soma da densidade espectral de potência para a frequência base do estímulo e para o segundo harmónico desta. Logo, o vetor de características é dados por

$$V = \begin{bmatrix} C_{cima} + C_{cima,2} & C_{baixo} + C_{baixo,2} & C_{direita} + C_{direita,2} & C_{esquerda} + C_{esquerda,2} \end{bmatrix}, \quad (5.5)$$

tendo como dimensão  $4 \times 1$ . O vetor de características gerado pelo método CCA é sempre dado por (5.3), independentemente do número de harmónicos considerados.

Foram testadas duas abordagens para a tomada de decisão. A primeira consiste em escolher a frequência correspondente à características de maior amplitude. A segunda consiste em aplicar o classificador linear de Bayes para determinar a frequência alvo.

O classificador linear de Bayes é um método supervisionado, logo necessita de treino, antes de estar apto a efetuar a classificação. As classes utilizadas correspondem às frequências alvo dos estímulos.

Em “Matlab” o classificador de Bayes é implementado pela função “classify”. Esta necessita de uma matriz com os vetores de treino e indicação da categoria das classes. Os vetores de treino são usados para construir a matriz, onde são dispostos de modo a que cada linha corresponda a uma observação, e cada coluna corresponda às características associadas a uma frequência. As classes são dispostas num vetor coluna, onde cada linha contém a classe do vetor presente na mesma linha da matriz de treino. Os vetores a classificar são dispostos da mesma forma que a matriz de treino, formando a matriz de amostras. A função devolve um vetor coluna, onde o elemento da linha  $i$  corresponde à classe do vetor presente na mesma linha da matriz de amostras. A probabilidade *à priori* de cada classe é de 25%, visto que é considerada uma distribuição uniforme para as classes possíveis. A função de custo é definida por (3.41).

## 5.5 Classificação *online*

Como foi mencionado na secção 4.1 o bloco que faz o processamento tem dois modos de funcionamento, treino e jogo. No modo de treino é construída a matriz de classificação, a partir dos dados obtidos. Esta matriz é usada no funcionamento em modo de jogo, para efetuar a classificação no final de cada época de decisão.

As amostras dos canais EEG são armazenados num *buffer* para cada canal. A cada intervalo de

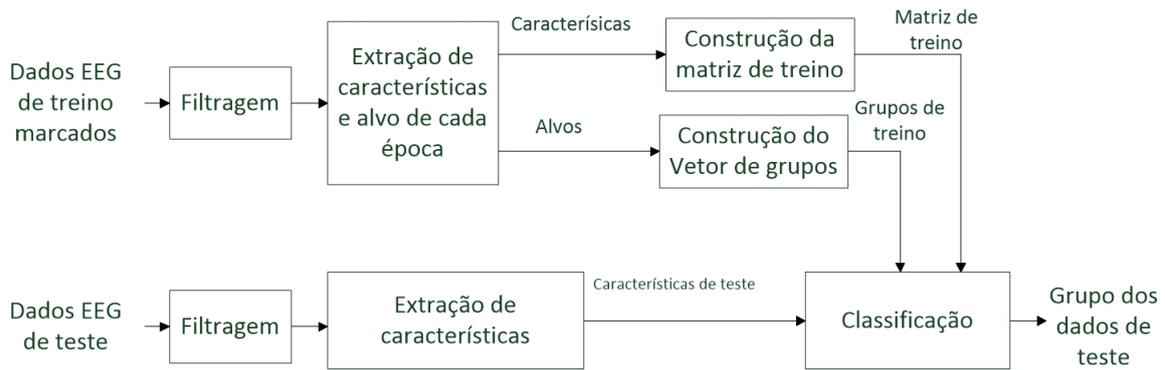


Figura 5.2: Esquema da aplicação do classificador com um conjunto de dados de treino e de teste.

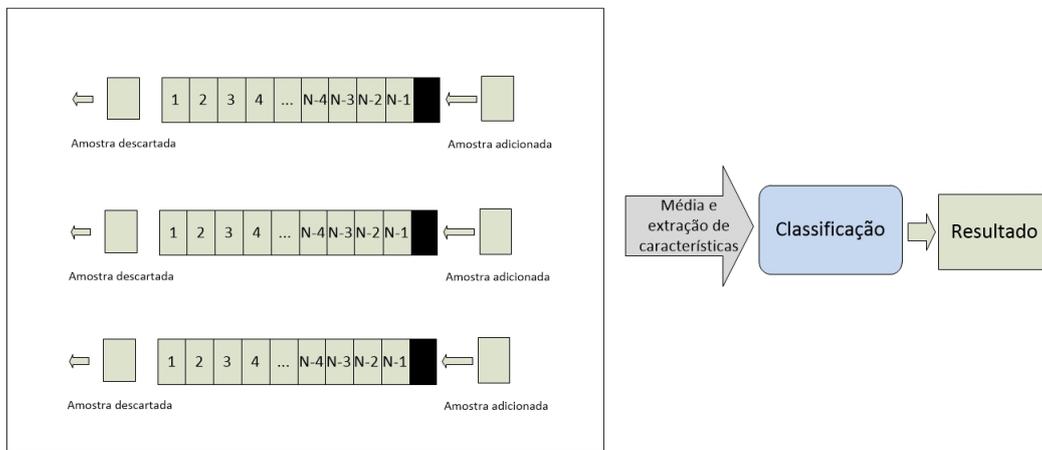


Figura 5.3: Processo de classificação *online*.

amostragem, a amostra mais antiga é descartada e o valor da nova amostra é adicionado ao *buffer*. O tamanho deste *buffer* é determinado por

$$N = f_s \cdot timeForProcessing, \tag{5.6}$$

onde  $f_s$  é a frequência de amostragem do sinal EEG, ou seja  $256Hz$ , e *timeForProcessing* representa o tempo que é dado ao utilizador, desde o momento em que aparece um obstáculo na pista até ao momento em que o classificador tem que tomar uma decisão. A cada instante de decisão é determinada a média dos *buffers*. Sobre esta média são aplicados os algoritmos de extração de características e de seguida é feita a classificação. A figura 5.3 apresenta o esquema da classificação *online*.



# Capítulo 6

## Resultados experimentais

### 6.1 Resultados de dados de treino

#### 6.1.1 Validação dos estímulos

Com vista a verificar o funcionamento dos estímulos para várias frequências foram efetuados testes experimentais, onde se pretendeu observar os sinais EEG de um utilizador a olhar para estímulos de diferentes frequências. A figura 6.1 mostra exemplos de respostas SSVEP a frequências sub-múltiplas da taxa de refrescamento do monitor (frequências  $5Hz$ ,  $6Hz$ ,  $7.5Hz$  e  $8.57Hz$ ), bem como para outras frequências (frequências  $7Hz$ ,  $8Hz$ ). Os dados apresentados foram extraídos durante épocas de 10 segundos e processados pelo método de Goertzel, para as frequências no intervalo  $[3, 20]$ , com uma resolução de  $0.1Hz$ .

É possível observar os picos característicos nas frequências alvo de cada época. No entanto, o segundo harmónico não é visível nos dados apresentados. Contrariamente ao esperado, a resposta SSVEP é discriminável para as frequências não sub-múltiplas da taxa de refrescamento do monitor, não sendo visíveis respostas a frequências erradas, concluindo-se que estas frequências também podem ser usadas. Em grande parte dos resultados apresentados, as frequências próximas dos  $10Hz$  apresentam uma forte atividade EEG, logo estas frequências foram excluídas dos testes, com vista a evitar falsos positivos. Foram escolhidas as frequências  $5Hz$ ,  $6Hz$ ,  $7Hz$  e  $8Hz$  para a implementação dos estímulos do jogo.

Utilizando épocas de 10 segundos é possível observar as respostas às frequências  $6.667Hz$  e  $8.57Hz$ , devido à resolução permitida na estimação espectral. No entanto, reduzindo a resolução do espectro, a densidade de potência espectral destas pode aparecer distribuída pelas frequências laterais à frequência de interesse. A figura 6.2 mostra o fenómeno descrito. Os espectros foram obtidos pelo método de Welch, visto que este permite obter resolução diferentes para os mesmos dados, variando o tamanho da janela. É possível observar que para a resolução  $0.1Hz$  ambas as respostas são discrimináveis. No entanto, ao baixar a resolução para  $0.5Hz$ , a resposta ao estímulo a  $6.667Hz$  torna-se difícil de discriminar, enquanto que a resposta a  $7.5Hz$ , continua a ter um pico discriminável.

#### 6.1.2 Resultados dos dados de treino

São apresentados os resultados de classificação usando os vários métodos de extração de características para o mesmo conjunto de dados. Foram testados os métodos de Goertzel, Welch e CCA, para a extração das características. A escolha do estímulo alvo foi obtida seguindo duas abordagens: classificador de Bayes e escolha da característica de maior amplitude. Os dados foram obtidos seguindo o protocolo descrito na secção 5.1. Nestes testes,  $T_{on}$  é de 12 segundo, com vista a testar vários intervalos de decisão, para usar na versão *online* do jogo. Os 2 segundos iniciais da época com estímulo servem

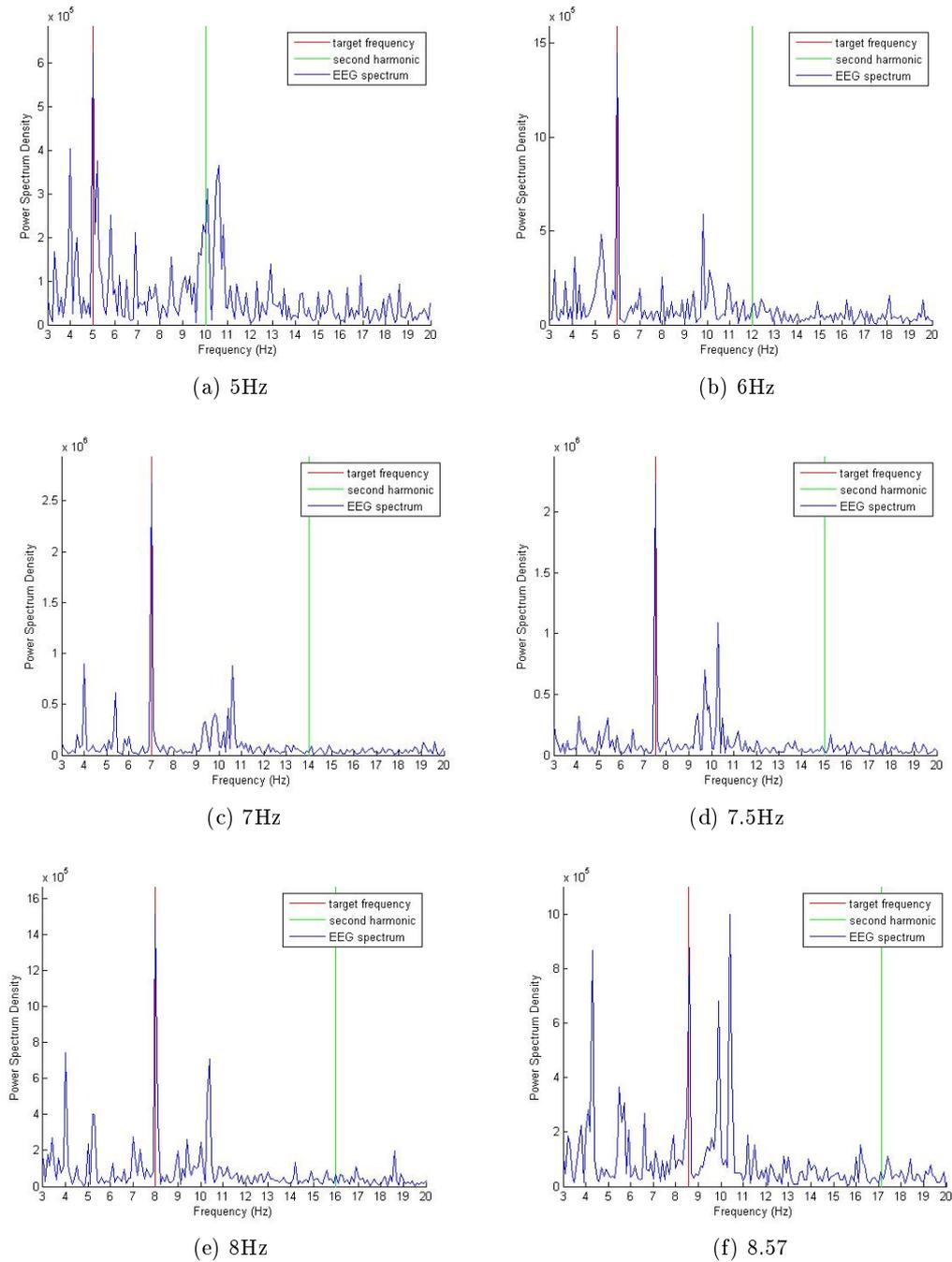
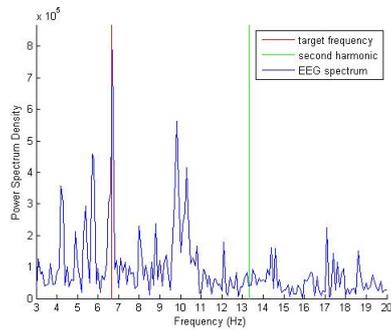
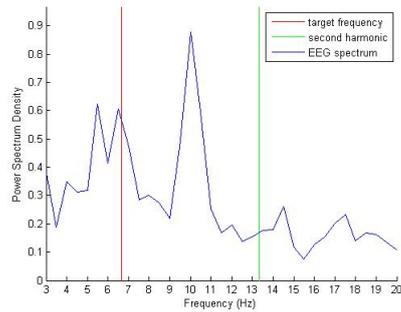


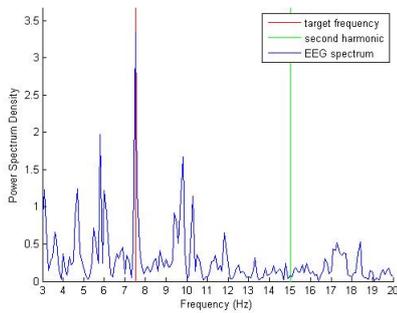
Figura 6.1: Exemplos de respostas SSVEP a diferentes frequências.



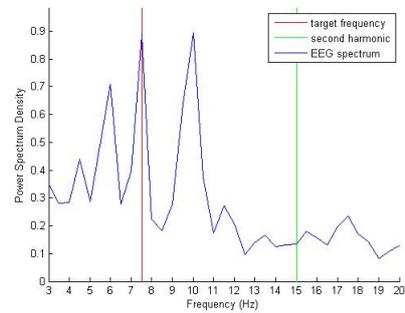
(a) Resposta a  $6.667Hz$  com resolução de  $0.1Hz$



(b) Resposta a  $6.667Hz$  com resolução de  $0.5Hz$



(c) Resposta a  $7.5Hz$  com resolução de  $0.1Hz$



(d) Resposta a  $7.5Hz$  com resolução de  $0.5Hz$

Figura 6.2: Comparação da resposta a um estímulo de  $6.667Hz$  e  $0.75Hz$  a resoluções diferentes. Espectros obtidos pelo método de Welch.

## 6.1. RESULTADOS DE DADOS DE TREINO

Sujeito A / 2 Classes	10 segundos	8 segundos	6 segundos	4 segundos	3 segundos
Goertzel	100	100	100	95	92,5
Welch (1 segundo)	100	97,5	92,5	95	87,5
Welch (2 segundos)	100	100	92,5	95	80
Welch (4 segundos)	100	100	95	92,5	NA
CCA	100	100	100	100	100

(a) Dados para duas classes.

Sujeito A / 4 Classes	10 segundos	8 segundos	6 segundos	4 segundos	3 segundos
Goertzel	100	100	100	95	88,75
Welch (1 segundo)	85	86.25	83.75	83.75	65
Welch (2 segundos)	95	95	96.25	91.25	77,5
Welch (4 segundos)	98.75	96.25	98.75	90	NA
CCA	100	100	100	100	97,5

(b) Dados para quatro classes.

Tabela 6.1: Resultados (% de acerto) de diferentes métodos de extração de características. Foram considerados o primeiro e segundo harmônicos da frequência de estímulo. A classificação é feita recorrendo ao método de escolha da característica máxima.

para o utilizador reconhecer o obstáculo, logo estes dados não são utilizados no processamento. Dos 10 segundos restantes de cada época são extraídas as amostras correspondentes ao intervalo de escolha desejado. A resposta SSVEP foi evocada através de estímulos de padrões alternados.

Nas tabelas 6.1, é apresentada a taxa de sucesso da escolha, em percentagem, para diferentes algoritmos de extração de características. Foi escolhido o comando correspondente à característica de maior amplitude. A tabela 6.1a apresenta os resultados para duas classes. Os resultados obtidos correspondem a 40 épocas (de 10, 8, 6, 4 e 3 segundos), 20 para cada estímulo. Na tabela 6.1b são apresentados os resultados para quatro classes, neste caso foram utilizadas 80 épocas (de 10, 8, 6, 4 e 3 segundos), 20 para cada estímulo. Para o algoritmo de Welch, foram testadas janelas de 1, 2 e 4 segundos, com 30% das amostras sobrepostas. Estes parâmetros foram testados para épocas de 10, 8, 6, 4 e 3 segundos, com vista a descobrir o mínimo de tempo necessário para a escolha do comando, de modo a minimizar o esforço do jogador.

É possível observar que o algoritmo de Goertzel é candidato para a extração de características, visto que só apresenta taxa de erro para épocas inferiores a 4 segundos. Porém o método CCA apresenta os melhores resultados, obtendo somente erros para épocas de 3 segundos, com quatro classes. O método de Welch apresenta resultados inferiores ao método de Goertzel, apesar da sua capacidade de reduzir ruído, através do cálculo da média da densidade espectral de potência. É possível observar, que o desempenho aumenta com o aumento do tamanho da janela, pelo que se infere que o aumento da resolução do espectro melhora o desempenho.

A tabela 6.2 mostra os resultados dos algoritmos de extração de características para diferentes sujeitos. Os dados foram obtidos seguindo o protocolo descrito na secção 5.1, onde  $T_{on} = T_{off} = 5$  segundos. O primeiro segundo das épocas com estímulo não é usado no processamento dos dados. Os resultados obtidos correspondem a 80 épocas (de 4 ou 3 segundos), 20 para cada estímulo. É possível

4 classes	Sujeito B		Sujeito C		Sujeito D		Sujeito E		Sujeito F	
Épocas (segundos)	4	3	4	3	4	3	4	3	4	3
Goertzel	92,5	87,5	95	92,5	98,75	95	73,75	71,25	100	100
Welch (1 segundo)	77,5	77,5	87,5	86,25	90	85	52,5	36,25	95	91,25
Welch (2 segundos)	82,5	85	88,75	80	92,5	86,25	52,5	45	93,75	87,5
CCA	100	100	97,5	93,75	100	100	83,75	78,75	100	100

Tabela 6.2: Resultados (% de acerto) de diferentes métodos de extração de características. Para diferentes sujeitos de teste. Foram considerados o primeiro e segundo harmônicos da frequência de estímulo. A classificação é feita recorrendo ao método de escolha da característica máxima.

Sujeito A / 2 Classes	10 segundos	8 segundos	6 segundos	4 segundos
Característica máxima - 1 harmónico	100	100	100	97,5
Característica máxima - 2 harmónicos	100	100	100	95
LDA - 1 harmónico	100	97,5	97,5	97,5
LDA - 2 harmónicos	100	100	100	97,5

(a) Dados para duas classes.

Sujeito A / 4 Classes	10 segundos	8 segundos	6 segundos	4 segundos
Característica máxima - 1 harmónico	100	98.75	100	96.25
Característica máxima - 2 harmónicos	100	100	100	95
LDA - 1 harmónico	98.75	100	97,5	95
LDA - 2 harmónicos	98.75	98.75	96.25	95

(b) Dados para quatro classes.

Tabela 6.3: Comparação dos resultados (% de acerto) dos métodos de classificação (escolha de característica de maior amplitude e classificador linear de Bayes) e o número de harmónicos escolhidos como características, para 2 classes e 4 classes. A extração de características é feita com o algoritmo de Goertzel.

observar que o algoritmo CCA apresenta o melhor desempenho para todos os sujeitos.

As tabelas 6.3 apresentam os resultados obtidos variando o método de classificação (escolha de característica de maior amplitude e classificador discriminante linear de Bayes) e o número de harmónicos usado como características, para o sujeito A. É possível observar que o algoritmo LDA apresenta bons resultados para duas classes, quando usados dois harmónicos, o que aumenta a dimensão do espaço das características, no entanto o mesmo já não se verifica para o caso de quatro classes. O método de escolha da característica de valor máximo, apresenta os melhores resultados. As diferenças na taxa de sucesso de classificação, entre os algoritmos, são baixas, correspondendo a diferenças de um erro entre as hipóteses para a classificação. Logo os casos apresentados demonstram resultados semelhantes, apresentando-se como boas hipóteses para o controlo do jogo.

A tabela 6.4 mostra resultados comparativos dos algoritmos de classificação para diferentes sujeitos. Os dados foram obtidos seguindo o protocolo descrito na secção 5.1, onde  $T_{on} = T_{off} = 5$  segundos. O primeiro segundo das épocas com estímulo não é usado no processamento dos dados. Os resultados obtidos correspondem a 40 épocas (de 4 ou 3 segundos), 10 para cada estímulo. As diferenças de desempenho para os algoritmos de classificação são baixas para cada sujeito, não se destacando nenhum algoritmo dos apresentados.

4 classes	Sujeito B		Sujeito C		Sujeito D		Sujeito E		Sujeito F	
Épocas (segundos)	4	3	4	3	4	3	4	3	4	3
Caract. máx. - 1 harmónico	90	90	95	87,5	92,5	95	85	82,5	95	95
Caract. máx. - 2 harmónicos	92,5	92,5	87,5	85	100	97,5	80	72,5	97,5	95
LDA - 1 harmónico	90	87,5	95	87,5	90	85	82,5	85	97,5	95
LDA - 2 harmónicos	92,5	82,5	95	90	95	97,5	75	80	97,5	97,5

Tabela 6.4: Comparação dos resultados (% de acerto) dos métodos de classificação (escolha de característica de maior amplitude e classificador linear de Bayes) e o número de harmónicos escolhidos como características, para 4 classes. A extração de características é feita com o algoritmo de Goertzel.

## 6.2 Resultados online

Nesta secção são apresentados os resultados obtidos durante o controlo *online* do jogo. Os resultados apresentados correspondem à taxa de sucesso de cinco participantes. O decorrer do jogo segue o protocolo definido na secção 5.1, com  $T_{on} = T_{off} = 5s$ . Foram utilizados os quatro diferentes tipos de obstáculos alvo no jogo, o que implica que o método de decisão teve que discriminar entre quatro frequências diferentes, sendo estas as escolhidas na secção 6.1.1.

A extração de características é feita pelo método de Goertzel. É somente considerada a componente fundamental de cada frequência alvo. Para cada utilizador é efetuada uma sessão de jogo, onde a classificação é feita escolhendo a característica de maior amplitude. Durante esta sessão, é gerada uma matriz de treino, a partir dos dados obtidos. Esta matriz é utilizada numa sessão seguinte, onde é aplicado o classificador linear de Bayes. Este processo permitiu obter dados comparativos entre ambos os métodos de classificação, para vários utilizadores. De seguida, os testes foram repetidos utilizando o método CCA para a extração de características.

Para cada utilizador foram efetuadas duas sessões de jogo, com vista a comparar o desempenho dos estímulos de padrões alternados e estímulos simples. Nestas sessões, a classificação foi feita recorrendo ao método de escolha da característica máxima. Os resultados destas sessões são apresentados na tabela 6.5, para todos os participantes exceto o sujeito B. É possível notar um melhor desempenho por parte dos estímulos de padrões alternados.

Na tabela 6.6 são apresentados os resultados dos testes *online* efetuados com o método de Goertzel. Todos os participantes, à exceção do Sujeito D, apresentam uma ligeira melhoria no desempenho com o uso do classificador de Bayes. Estes resultados contrariam os obtidos na secção 6.1.2, onde é concluído que o método de escolha da característica de maior amplitude apresenta melhores resultados. O desempenho médio do sistema sem classificador foi de 89.5%, o que corresponde a um ITR de 16.19<sup>bits</sup>/min, determinado através da equação 2.1. O sistema com classificador obteve um desempenho médio de 91%, o que corresponde a um ITR de 17.05<sup>bits</sup>/min.

Na tabela 6.7 são apresentados os resultados dos testes *online* efetuados com o método de CCA. Como seria de esperar, de acordo com os resultados apresentados na secção 6.1.2, os resultados com o método CCA são melhores que os obtidos com o método de Goertzel. O desempenho médio do sistema sem classificador foi de 98.5%, o que corresponde a um ITR de 22.37<sup>bits</sup>/min. O sistema com classificador obteve um desempenho médio de 98%, o que corresponde a um ITR de 21.92<sup>bits</sup>/min.

	Estímulo padrões alternados	Estímulo simples
Sujeito C	90	82.5
Sujeito D	95	50
Sujeito E	85	55
Sujeito F	90	55

Tabela 6.5: Comparação da taxa de sucesso para estímulos de padrões alternados e estímulos simples. A extração de características é feita pelo algoritmo de Goertzel. A classificação é feita pela escolha da característica de maior amplitude.

Testes online	Treino	Classificação	Estímulos
Sujeito B	87.5	90	Padrões alternados
Sujeito C	90	92.5	Padrões alternados
Sujeito D	95	92.5	Padrões alternados
Sujeito E	85	87.5	Padrões alternados
Sujeito F	90	92.5	Padrões alternados
Média	89.5	91	

Tabela 6.6: Taxa de sucesso para os testes online. A extração de características é feita pelo método de Goertzel. Nas sessões de treino é usada a escolha da característica de maior amplitude. Nas sessões de jogo é usado o classificador linear de Bayes.

Testes online	Treino	Classificação	Estímulos
Sujeito B	95	100	Padrões alternados
Sujeito C	97.5	90	Padrões alternados
Sujeito D	100	100	Padrões alternados
Sujeito E	100	100	Padrões alternados
Sujeito F	100	100	Padrões alternados
Média	98.5	98	

Tabela 6.7: Taxa de sucesso para os testes online. A extração de características é feita pelo método de CCA. Nas sessões de treino é usada a escolha da característica de maior amplitude. Nas sessões de jogo é usado o classificador linear de Bayes.

### **6.3 Resultados qualitativos dos estímulos visuais**

Os participantes demonstraram preferência pelos estímulos de padrões alternados, reportando que estes geram menos fadiga e desconforto. Os estímulos de padrões alternados mostraram ser mais fáceis de captar a atenção, possivelmente contribuindo para o melhor desempenho obtido. Foi também reportado que os estímulos de gráficos singulares causam maior desconforto, devido à sua luminosidade. Estes aparentam gerar imensa fadiga visual, tendo sido necessário efetuar pausas para descanso, depois das sessões de jogo com este tipo de estímulos.

## Capítulo 7

# Conclusões e trabalho futuro

O controlo do jogo “Spacecraft” tem por base o neuromecanismo SSVEP. Para tal, foi necessária a implementação de módulos de *software*, bem como a alteração de alguns módulos já existentes neste jogo. As alterações realizadas tiveram como objetivo melhorar o desempenho do jogo, bem como implementar uma forma de controlo diferente da existente. Uma destas implementações consistiu em usar estímulos configuráveis, permitindo deste modo comparar o desempenho em testes *online* com recurso a de estímulos simples e estímulos de padrões alternados. A partir dos resultados obtidos pode concluir-se que os estímulos de padrões alternados oferecem uma melhor opção para a obtenção de resposta SSVEP. Os participantes nas experiências reportaram que os estímulos simples causavam maior desconforto e fadiga. Na fase de validação dos estímulos, foi possível observar que a resposta obtida apresentava a componente principal da frequência do estímulo, validando o funcionamento destes para as frequências testadas.

Foram efetuados testes para a escolha dos algoritmos a usar nas fases de extração de características e classificação. O melhor desempenho foi obtido pelo algoritmo CCA, para extração de características. Para o método de Welch foi verificado, que o seu desempenho aumenta com o aumento do tamanho da janela utilizada. Na fase de classificação online, verificou-se que a utilização do classificador discriminante linear de Bayes melhora os resultados, visto que a escolha da frequência dominante se adapta ao utilizador, no entanto este necessita de uma sessão de treino prévia. Isto foi verificado em testes *online* efetuados com um conjunto de participantes. Nestes testes, foi obtida uma ITR média de 16.19<sup>bits</sup>/min sem classificador, e 17.05<sup>bits</sup>/min com classificador recorrendo ao método de Goertzel para extração de características. Foram também efetuadas sessões *online* com o método CCA para extração das características. Nestas sessões foi obtida uma ITR média de 22.37<sup>bits</sup>/min sem classificador, e 21.92<sup>bits</sup>/min com classificador.

Como trabalho futuro é sugerida a diminuição da fadiga gerada pelos estímulos, visto que estes ainda causam um nível de fadiga considerável/elevado para sessões prolongadas. Uma possibilidade para atenuar a fadiga pode passar por identificar uma conjugação de cores e parâmetros dos estímulos mais favorável. A introdução de outros elementos de “jogabilidade”, ou tornar o jogo assíncrono pode ser uma opção desejável, contribuindo para a diversão associada ao jogo e usabilidade. Por fim, trabalho pode ser feito com vista a aplicar um equipamento de aquisição de biosinais comercial ao jogo, de forma a contribuir para a acessibilidade da tecnologia BCI.



# Bibliografia

- [Akhtar et al., 2014] Akhtar, A., Norton, J. J. S., Kasraie, M., and Bretl, T. (2014). Playing checkers with your mind: An interactive multiplayer hardware game platform for brain-computer interfaces. *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1650–1653. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6943922>.
- [Angeloni et al., 2012] Angeloni, C., Salter, D., Corbit, V., Lorence, T., Yu, Y.-c., and Gabel, L. A. (2012). P300-based Brain-Computer Interface Memory Game to Improve Motivation and Performance. *2012 38th Annual Northeast Bioengineering Conference, NEBEC 2012 (2012)*, pages 35–36.
- [Bonnet et al., 2013] Bonnet, L., Lotte, F., and Lécuyer, A. (2013). Two Brains, One Game: Design and Evaluation of a Multiuser BCI Video Game Based on Motor Imagery. *Computational Intelligence and ...*, 5(2):185–198. Available from: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6400237](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6400237).
- [Chae et al., 2012] Chae, Y., Jeong, J., and Jo, S. (2012). Toward Brain-Actuated Humanoid Robots : Asynchronous Direct Control Using an EEG-BasedBCI. *IEEE Transactions on Robotics (2012)*, 28(5):1131–1144.
- [Chumerin and Manyakov, 2013] Chumerin, N. and Manyakov, N. (2013). Steady-state visual evoked potential-based computer gaming on a consumer-grade eeg device. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(2):100–110. Available from: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6334432](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6334432).
- [Duda et al., 1999] Duda, R., Hart, P., and Stork, D. (1999). *Pattern classification*. 2 edition.
- [Fan et al., 2014] Fan, X., Bi, L., Teng, T., Ding, H., and Liu, Y. (2014). A brain-computer interface-based vehicle destination selection system using P300 and SS-VEP signals. *IEEE Intelligent Transportation Systems Magazine*, pages 1–10. Available from: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6484110](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6484110)[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6847229](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6847229).
- [Gtec,] Gtec. Highspeed Online Processing blockset. Available from: <http://www.gtec.at/Products/Software/High-Speed-Online-Processing-under-Simulink-Specs-Features>.
- [Guger et al., 2009] Guger, C., Daban, S., Sellers, E., Holzner, C., Krausz, G., Carabalona, R., Gramatica, F., and Edlinger, G. (2009). How many people are able to control a P300-based brain-computer interface (BCI)? *Neuroscience letters*, 462(1):94–8. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/19545601>.
- [Kuo-Kai Shyu et al., 2010] Kuo-Kai Shyu, Po-Lei Lee, Ming-Huan Lee, Ming-Hong Lin, Ren-Jie Lai, and Yun-Jen Chiu (2010). Development of a Low-Cost FPGA-Based SSVEP BCI Multimedia Control System. *IEEE transactions on biomedical circuits and systems*, 4(2):125–32. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/23853320>.

- [Kus et al., 2013] Kus, R., Duszyk, A., Milanowski, P., Labecki, M., Bierzynska, M., Radzikowska, Z., Michalska, M., Zygierevicz, J., Suffczynski, P., and Durka, P. J. (2013). On the quantification of SSVEP frequency responses in human EEG in realistic BCI conditions. *PLoS one*, 8(10):e77536. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3799619&tool=pmcentrez&rendertype=abstract>.
- [Laar et al., 2013] Laar, B. V. D., Gürkök, H., Bos, D. P.-O., Poel, M., and Nijholt, A. (2013). Experiencing BCI control in a popular computer game. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(2):176–184. Available from: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6484110](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6484110).
- [Lalor et al., 2005] Lalor, E. C., Kelly, S. P., Finucane, C., Burke, R., Smith, R., Reilly, R. B., and McDarby, G. (2005). Steady-State VEP-Based Brain-Computer Interface Control in an Immersive 3D Gaming Environment. *EURASIP Journal on Advances in Signal Processing*, 2005(19):3156–3164. Available from: <http://asp.eurasipjournals.com/content/2005/19/706906>.
- [Lin et al., 2007] Lin, Z., Zhang, C., Wu, W., and Gao, X. (2007). Frequency recognition based on canonical correlation analysis for SSVEP-based BCIs. *IEEE transactions on bio-medical engineering*, 54(6 Pt 2):1172–6. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/17549911>.
- [Lopes et al., 2011] Lopes, A. C., Pires, G., Vaz, L., and Nunes, U. (2011). Wheelchair navigation assisted by Human-Machine shared-control and a P300-based Brain Computer Interface. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2438–2444. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6094748>.
- [Marshall et al., 2013] Marshall, D., Coyle, D., Member, S., Wilson, S., and Callaghan, M. (2013). Games , Gameplay , and BCI : The State of the Art. 5(2):82–99.
- [Martinez et al., 2007] Martinez, P., Bakardjian, H., and Cichocki, A. (2007). Fully online multicommand brain-computer interface with visual neurofeedback using SSVEP paradigm. *Computational intelligence and neuroscience*, 2007(i):94561. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2266836&tool=pmcentrez&rendertype=abstract>.
- [Mathworks, 2012a] Mathworks (2012a). Create UDP object. Available from: <http://www.mathworks.com/help/instrument/udp.html>.
- [Mathworks, 2012b] Mathworks (2012b). Discriminant Analysis. Available from: <http://www.mathworks.com/help/stats/discriminant-analysis.html#bs31msq>.
- [Mathworks, 2012c] Mathworks (2012c). Welch’s power spectral density estimate. Available from: <http://www.mathworks.com/help/signal/ref/pwelch.html>.
- [Meattini et al., 2014] Meattini, R., Scarcia, U., Melchiorri, C., and Belpaeme, T. (2014). Gestural art: A Steady State Visual Evoked Potential (SSVEP) based Brain Computer Interface to express intentions through a robotic hand. *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 211–216. Available from: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6926255>.

- [MSDN, a] MSDN. Acquiring high-resolution time stamps. Available from: [https://msdn.microsoft.com/en-us/library/windows/desktop/dn553408\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dn553408(v=vs.85).aspx).
- [MSDN, b] MSDN. Creating a UDP Datagram Socket Application. Available from: <https://msdn.microsoft.com/en-us/library/ms881658.aspx>.
- [Nakanishi et al., 2014] Nakanishi, M., Wang, Y., Wang, Y.-T., Mitsukura, Y., and Jung, T.-P. (2014). Generating visual flickers for eliciting robust steady-state visual evoked potentials at flexible frequencies using monitor refresh rate. *PLoS one*, 9(6):e99235. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4053390&tool=pmcentrez&rendertype=abstract>.
- [Obermaier et al., 2001] Obermaier, B., Neuper, C., Guger, C., and Pfurtscheller, G. (2001). Information transfer rate in a five-classes brain-computer interface. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 9(3):283–8. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/11561664>.
- [Ortner et al., 2011] Ortner, R., Allison, B. Z., Korisek, G., Gaggli, H., and Pfurtscheller, G. (2011). An SSVEP BCI to control a hand orthosis for persons with tetraplegia. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 19(1):1–5. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/20875978>.
- [Parafita et al., 2013] Parafita, R., Pires, G., Nunes, U., and Castelo-branco, M. (2013). A spacecraft game controlled with a brain-computer interface using SSVEP with phase tagging. *SeGAH 2013 - IEEE 2nd International Conference on Serious Games and Applications for Health*.
- [Perdiz, 2014] Perdiz, J. a. (2014). Relatório Técnico classificação de SSVEP.
- [Pires et al., 2011] Pires, G., Torres, M., Casaleiro, N., Nunes, U., and Castelo-Branco, M. (2011). Playing Tetris with non-invasive BCI. *IEEE 1st Int. Conference on Serious Games and Applications for Health (SeGAH 2011), Portugal*.
- [Po-Lei et al., 2014] Po-Lei, L., Hao-Teng, S., and Hsiang-Chih, C. (2014). Design a brain computer interface gaming system using steady-state visual evoked potential. In *Consumer Electronics - Taiwan (ICCE-TW), 2014 IEEE International Conference on*, pages 5–6.
- [Press et al., 1992] Press, W. H., Teukolsky, S. a., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical recipes in C (2nd ed.): the art of scientific computing*, volume 29. Cambridge University Press. Available from: <http://www.jstor.org/stable/1269484?origin=crossref>.
- [Proakis and Monolakis, 1996] Proakis, J. G. and Monolakis, D. G. (1996). *Digital Signal Processing: principles, algorithms and applications*. Prentice-Hall International, Inc.
- [Punsawad and Wongsawat, 2013] Punsawad, Y. and Wongsawat, Y. (2013). Hybrid SSVEP-motion visual stimulus based BCI system for intelligent wheelchair. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference, 2013*:7416–9. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/24111459>.

- [Scherer et al., 2008] Scherer, R., Lee, F., Schl, A., Leeb, R., Member, S., and Member, G. P. (2008). Towards self-paced (asynchronous) Brain-Computer Communication : Navigation through virtual worlds. *Biomedical Engineering IEEE Transaction on (2008)*, pages 1–9.
- [Sysel and Rajmic, 2012] Sysel, P. and Rajmic, P. (2012). Goertzel algorithm generalized to non-integer multiples of fundamental frequency. *EURASIP Journal on Advances in Signal Processing*, 2012(1):56. Available from: <http://asp.urasipjournals.com/content/2012/1/56>.
- [Vilic et al., 2013] Vilic, A., Kjaer, T. W., Thomsen, C. E., Puthusserypady, S., and Sorensen, H. B. D. (2013). DTU BCI speller: an SSVEP-based spelling system with dictionary support. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference*, 2013:2212–5. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/24110162>.
- [Wang et al., 2006] Wang, Y., Wang, R., Gao, X., Hong, B., and Gao, S. (2006). A practical VEP-based brain-computer interface. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 14(2):234–9. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/16792302>.
- [Weenink, 2003] Weenink, D. (2003). Canonical correlation analysis. In *Institute of Phonetic Sciences, University of Amsterdam, Proceedings 25*, pages 81–99. Available from: <http://onlinelibrary.wiley.com/doi/10.1002/0470013192.bsa068/full>.
- [Welch, 1967] Welch, P. (1967). The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15:70–73. Available from: [http://web.iaincirebon.ac.id/ebook/luke/ieeeeexplore/Audio\\_and\\_Electroacoustics\\_IE/The\\_use\\_of\\_fast\\_Fourier\\_transform\\_for\\_the\\_estimation\\_of\\_power\\_spectra\\_A\\_method\\_based\\_on\\_time\\_averagi-P1T.pdf](http://web.iaincirebon.ac.id/ebook/luke/ieeeeexplore/Audio_and_Electroacoustics_IE/The_use_of_fast_Fourier_transform_for_the_estimation_of_power_spectra_A_method_based_on_time_averagi-P1T.pdf).
- [Wolpaw et al., 2002] Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 113(6):767–91. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/12048038>.
- [Yin et al., 2014] Yin, E., Zhou, Z., Jiang, J., Chen, F., Liu, Y., and Hu, D. (2014). A speedy hybrid BCI spelling approach combining P300 and SSVEP. *IEEE transactions on bio-medical engineering*, 61(2):473–83. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/24058009>.
- [Zhu et al., 2010] Zhu, D., Bieger, J., Garcia Molina, G., and Aarts, R. M. (2010). A survey of stimulation methods used in SSVEP-based BCIs. *Computational intelligence and neuroscience*, 2010:702357. Available from: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2833411&tool=pmcentrez&rendertype=abstract>.