الجمهورية الجائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Mohamed Khider - Biskra
Faculté des Sciences et de la Technologie
Département : Génie électrique

جامعة محمد خيضر بسكرة
كلية العلوم والتكنولوجيا
قسم الهنسة الكهربائية

المرجع: .....

Thèse présentée en vue de l'obtention
De diplôme de
Doctorat LMD en **: Automatique**

Option **: Modélisation et commande des systèmes dynamiques**

Conception et Réalisation D'une Interface Cerveau Ordinateur

présentée par :

BENDIB Hafed-Eddine

Soutenue publiquement le : 01/07/2024.

Devant le jury composé de :

Mostefa Mohamed TOUBA	Maître de Conférence A	Président	Université de Biskra
Mebarek BAHRI	Professeur	Rapporteur	Université de Biskra
Mohamed BOUMEHRAZ	Professeur	Co-rapporteur	Université de Biskra
Abdelhamid MESSAOUDI	Maître de Conférence A	Examinateur	Université de Biskra
Fouzi DOUAK	Maître de Conférence A	Examinateur	Université de Khenchela

REMERCIEMENTS

Je tiens à remercier du fond du cœur ma femme et ma fille pour leur patience et leur soutien tout au long de ma recherche. Leur amour et leur compréhension ont été essentiels pour que je puisse mener à bien ce projet.

Je souhaite exprimer ma profonde reconnaissance envers mes directeurs de thèse, Pr. Mebarek Bahri et Pr. Mohamed Boumehraz, pour m'avoir donné l'opportunité de poursuivre mon doctorat au sein du laboratoire de Modélisation des Systèmes énergétiques. Leur expertise inégalée, leur dévouement sans faille, leur sens de l'humour rafraîchissant et leur confiance indéfectible en mes compétences ont métamorphosé mon passage au laboratoire LMSE en une expérience à la fois exaltante sur le plan professionnel et enrichissante sur le plan personnel. De plus, je tiens à exprimer ma sincère gratitude envers tous les membres du laboratoire LMSE, en particulier l'équipe exceptionnellement brillante dédiée au contrôle intelligent. Non seulement ai-je bénéficié de précieuses interactions scientifiques avec eux, mais leur jovialité constante et l'atmosphère exceptionnelle qui a baigné ces années de doctorat ont indéniablement embelli mon parcours.

Je tiens également à exprimer ma profonde gratitude au Professeur Mohamed Mostefa TOUBA, Président du jury, ainsi qu'aux Maîtres de Conférence Dr. Abedlehamid MESSAOUDI de l'Université de Biskra et Dr. Fouzi DOUAK de l'Université de Khenchela, pour avoir accepté de juger mon travail et d'être membres de la commission d'examen.

Je tiens à exprimer ma profonde gratitude envers Madame F. Betka, ingénieure au sein du laboratoire LMSE, pour son précieux soutien, sa générosité et son professionnalisme exceptionnels tout au long de ma thèse de doctorat. Je tiens également à exprimer ma sincère gratitude envers le Docteur Benaziza Walid de l'Université de Batna, le Docteur Bendib Abdelhalim de l'Université d'Oran et le Docteur Mallem Ali de l'Université de Khenchela pour leur précieux soutien moral et leur expertise scientifique. Leur guidance et leurs conseils ont été d'une importance capitale tout au long de mon parcours académique. Je souhaite exprimer ma reconnaissance envers mes collègues de l'école doctorale, notamment Dr. Khaled T., Dr. Soufiane B., Dr. Jamal T., Dr. Seif-el-islam H. et Dr. Maroua H., pour leur sympathie, leur aide précieuse et leur bonne humeur.

En outre, je souhaite chaleureusement remercier tous mes amis (Ing. Saâd H., Mr. Atef B., Mr. Fateh K., Sr. Moundher G., Dr. Sami B., Dr. Abdelhey Z., …) qui m'ont apporté un soutien moral inestimable et ont partagé mon espoir de succès. Leur encouragement et leur amitié ont été des sources de motivation essentielles qui ont illuminé mon chemin vers l'accomplissement de ce doctorat.

Abstract

Brain-computer interfaces (BCIs) are much more than just a technology; they are a bridge between our mind and the machine. Imagine being able to control a computer, write an email, or even drive a wheelchair simply by thinking. This is precisely what BCIs enable, and their importance spans several essential fields. They hold significant importance in various domains, including medicine, neuroscience, and human-machine interaction, due to their ability to establish a direct connection between the human brain and a computer. In recent years, the field of BCIs has experienced rapid development, driven by significant advancements in signal processing and artificial intelligence. In this thesis, we began with an in-depth theoretical study to lay the necessary foundations for designing complex and multidisciplinary systems, addressing fundamental concepts and cutting-edge technologies. Subsequently, we developed an EEG acquisition system for a BCI, integrating high-quality hardware and software solutions at a reduced cost, which allowed us to develop an economical prototype aimed at the academic community, thus facilitating research and innovation in this field. The associated firmware, which orchestrates the hardware, developed in C/C++, is designed to meet real-time requirements. Furthermore, we conducted a comprehensive process to rigorously evaluate its performance and efficiency, taking into account various determining parameters. To enrich this ecosystem, we designed an innovative toolbox called Step2BCI, which is fully compatible with the hardware platform and capable of implementing various experimental paradigms and protocols. Step2BCI integrates a wide range of advanced signal processing, artificial intelligence, and machine learning techniques, thereby offering increased flexibility and analytical power. It has been tested on several third-party data sets, demonstrating its ability to execute a complete pipeline for brain activity classification, while supporting various data formats, making the process smoother and more efficient. This toolbox enables the execution of complex experiments without requiring advanced programming skills, which democratizes access to BCI technology and facilitates its adoption by a greater number of researchers and developers. By facilitating the integration of diverse techniques and simplifying the experimentation process, Step2BCI helps accelerate progress in the field of BCIs. Finally, this thesis proposes a comprehensive and affordable solution for the design and implementation of a brain-computer interface, contributing to democratizing access to this promising technology and promoting new advances in the field, thereby paving the way for innovative and beneficial applications for society.

Keywords: Brain-Computer Interfaces (BCI), ADS1299-EFE, EEG Acquisition, Toolbox Step2BCI, Machine and Deep Learning, Feature Extraction, Step2BCI.

Résumé

Les interfaces cerveau-ordinateur (BCI) sont bien plus qu'une simple technologie ; elles sont une passerelle entre notre esprit et la machine. Imaginez pouvoir contrôler un ordinateur, écrire un e-mail ou même piloter un fauteuil roulant simplement en pensant. C'est précisément ce que les BCI permettent, et leur importance s'étend à plusieurs domaines essentiels. Elles revêtent une importance capitale dans divers domaines, notamment la médecine, les neurosciences et l'interaction homme-machine, grâce à leur capacité à établir une connexion directe entre le cerveau humain et un ordinateur. Ces dernières années, le domaine des BCI a connu une évolution rapide, stimulée par des avancées significatives en traitement de signal et en intelligence artificielle. Dans cette thèse, nous avons commencé par une étude théorique approfondie pour poser les bases nécessaires à la conception de systèmes complexes et multidisciplinaires, en abordant des concepts fondamentaux et des technologies de pointe. Par la suite, nous avons élaboré un système d'acquisition EEG pour une BCI, en intégrant des solutions matérielles et logicielles de haute qualité à un coût réduit, ce qui nous a permis de développer un prototype économique destiné à la communauté universitaire, facilitant ainsi la recherche et l'innovation dans ce domaine. Le micrologiciel associé, qui orchestre le matériel, développé en langage C/C++, est conçu pour répondre aux exigences en temps réel. D'autre part, nous avons effectué un processus intégral pour évaluer rigoureusement sa performance et son efficacité en tenant compte de divers paramètres déterminants. Pour enrichir cet écosystème, nous avons conçu une boîte à outils innovante appelée Step2BCI, qui est parfaitement compatible avec la plateforme matérielle et capable de mettre en œuvre divers paradigmes et protocoles expérimentaux. Step2BCI intègre une large gamme de techniques avancées de traitement de signal, d'intelligence artificielle et d'apprentissage automatique, offrant ainsi une flexibilité et une puissance d'analyse accrues. Elle a été testée sur plusieurs ensembles de données tierces, démontrant sa capacité à exécuter un pipeline complet pour la classification de l'activité cérébrale, tout en prenant en charge divers formats de données, rendant le processus plus fluide et efficace. Cette boîte à outils permet l'exécution d'expériences complexes sans nécessiter des compétences de programmation avancées, ce qui démocratise l'accès à la technologie BCI et facilite son adoption par un plus grand nombre de chercheurs et de développeurs. En facilitant l'intégration de diverses techniques et en simplifiant le processus d'expérimentation, Step2BCI contribue à accélérer les progrès dans le domaine des BCI. Finalement, cette thèse propose une solution complète et abordable pour la conception et la réalisation d'une interface cerveau-ordinateur, contribuant à démocratiser l'accès à cette technologie prometteuse et à favoriser de nouvelles avancées dans le domaine, ouvrant ainsi la voie à des applications innovantes et bénéfiques pour la société.

Mots-clés : Interfaces cerveau-ordinateur (BCI), ADS1299-EFE, Acquisition EEG, Toolbox Step2BCI, Apprentissage Automatique et profond, Extraction des caractéristiques, Step2BCI.

ملخص

تعد واجهات الدماغ-الحاسوب (BCI) أكثر من مجرد تقنية بسيطة؛ إنها جسر.بين عقلنا والآلة. تخيل أن تكون قادرًا على التحكم في جهاز الكمبيوتر، كتابة بريد إلكتروني أو حتى قيادة كرسي متحرك بمجرد التفكير. هذا بالضبط ما تتيحهBCI ، وأهميتها تمتد إلى عدة مجالات أساسية. إنها تكتسب أهمية كبيرة في مجالات متعددة مثل الطب وعلوم الأعصاب والتفاعل بين الإنسان والآلة، بفضل قدرتها على إنشاء اتصال مباشر بين الدماغ البشري وجهاز الكمبيوتر. في السنوات الأخيرة، شهد مجال BCI تطورًا سريعًا بفضل التقدم الكبير في معالجة الإشارة والذكاء الاصطناعي. في هذه الأطروحة، بدأنا بدراسة نظرية معمقة لوضع الأسس اللازمة لتصميم أنظمة معقدة ومتعددة التخصصات، من خلال تناول المفاهيم الأساسية والتقنيات المتقدمة. بعد ذلك، قمنا بتطوير نظام لاقتناص إشارات EEG لواجهة الدماغ-الحاسوب، من خلال دمج حلول مادية وبرمجية عالية الجودة بتكلفة منخفضة، مما سمح لنا بتطوير نموذج أولي اقتصادي موجه للمجتمع الجامعي، مما يسهل البحث والابتكار في هذا المجال. تم تطوير البرنامج الثابت (Firmware) المرتبط الذي يدير الأجهزة بلغة C/C++، ليلبي متطلبات الوقت الفعلي. من ناحية أخرى، قمنا بإجراء عملية شاملة لتقييم أدائه وكفاءته بدقة، مع مراعاة مختلف العوامل المحددة. لتعزيز هذا النظام البيئي، قمنا بتصميم مجموعة أدوات مبتكرة تسمى « Step2BCI» ، وهي متوافقة تمامًا مع النظام المادي وقادرة على تنفيذ نماذج وبروتوكولات تجريبية متنوعة. تدمج Step2BCI مجموعة واسعة من التقنيات المتقدمة في معالجة الإشارة، والذكاء الاصطناعي، والتعلم الآلي، مما يوفر مرونة وقوة تحليلية متزايدة. تم اختبارها على عدة مجموعات من البيانات التابعة لأطراف ثالثة، مما أثبت قدرتها على تنفيذ خط معالجة كامل لتصنيف النشاط الدماغي، مع دعم تنسيقات بيانات متنوعة، مما يجعل العملية أكثر سلاسة وفعالية. تتيح هذه المجموعة من الأدوات تنفيذ تجارب معقدة دون الحاجة إلى مهارات برمجية متقدمة، مما يتيح الوصول إلى تقنية BCI ويسهل تبنيها من قبل عدد أكبر من الباحثين والمطورين. من خلال تسهيل دمج تقنيات متنوعة وتبسيط عملية التجربة، تساهم Step2BCI في تسريع التقدم في مجال .BCI في الختام، تقترح هذه الأطروحة حلاً شاملاً وبتكلفة معقولة لتصميم وتنفيذ واجهة الدماغ-الحاسوب، مما يسهم في دمقرطة الوصول إلى هذه التكنولوجيا الواعدة وتشجيع تقدمات جديدة في هذا المجال، مما يفتح الطريق أمام تطبيقات مبتكرة ومفيدة للمجتمع.

كلمات مفتاحية: الواجهات الدماغية-الحاسوبية (BCI) ، ADS1299-EFE، اقتناءEEG ، صندوق أدوات Step2BCI، التعلم الآلي والعميق، استخراج السمات ، Step2BCI.

LISTE DES ABREVIATIONS

Abréviations	Anglais	Français
AFE	Analog Front End	Frontal Analogique
ANN	Artificial Neural Network	Réseau de neurones artificiel
BCI	Brain-Computer Interface	Interface Cerveau-Ordinateur
BOLD	Blood-Oxygen-Level Dependent	dépendant du niveau d'oxygène sanguin
CAN	Analog Digital Converter	Convertisseur Analogique Numérique
CAR	Common Average Reference	Moyenne commune de référence
CC	Direct Current	Courant Continu
CMOS	complementary metal-oxide semiconductor	semi-conducteur à oxyde métallique complémentaire
CMOS	Complementary Metal-Oxide-Semiconductor	Oxyde métallique semi-conducteur complémentaire
CNA	Digital Analog Converter	Connvertisseur Numérique Analogique
COV	Covariance	Covariance
CPU	Central Processing Unit	Unité Centrale de Calcul
CSP	Common Spatial Pattern	Motif Spatial Commun
CUDA	Compute Unified Device Architecture	Architecture de Périphérique de Calcul Unifiée
CV	Cross-Validation	Validation croisée
DFA	Detrended Fluctuation Analysis	Analyse des fluctuations sans tendance
DFT	Discrete Fourier Transform	Transformée de Fourier discrète
DPSS	Discrete Prolate Spheroidal Sequences	Séquences sphéroïdales prolates discrètes
DR	Data Rate	Débit de données
DRDY	data ready	Données prêtes
DSP	power spectral density	Densité Spectrale de Puissance
DT	Decision-Tree	Arbre de décision
DTS	Digital Temperature Sensor	capteur de température numérique
ECoG	ElectroCorticoGraphy	ElectroCorticoGraphie
EEG	ElectroEncephaloGraphy	ElectroEncéphaloGraphie
EMG	ElectroMyoGraphy	ElectroMyoGraphie
ENOB	Effective Number Of Bits	Nombre effectif de bits
EOG	ElectroOculoGraphy	ElectroOculoGraphie
ERD	Event Related Desynchronization	désynchronisation liée aux événements
ERP	Event Related Potentials	Potentiels Liés aux Événements
ERS	Event Related Synchronization	synchronisation liée aux événements
ESD	Electrostatic Discharge	Décharge Électrostatique
ESP-IDF	Espressif IoT Development Framework	Cadre de développement Espressif IoT
ETI	Electrode Tissu Impedance	Impédance Électrode Tissu
FDG	Fluo-Deoxy-Glucose	Fluo-Déoxy-Glucose
FFT	Fast Fourier Transform	Transformée de Fourier Rapide
fNIRS	Functional Near InfraRed Spectroscopy	Spectroscopie Proche de l'Infrarouge fonctionnelle
FPGA	Field-Programmable Gate Arrays	Réseaux de portes programmables

FreeRTOS	Free Real-Time Operating System	Système d'exploitation libre en temps réel
GD	Gradient Descente	Déscente du Gradient
GNB	Gaussian Naïve Bayes	Bayes naïf gaussien
GPU	Graphical Processing Unit	Unité de traitement graphique
GUI	Graphical User Interface	Interface utilisateur graphique
HF	High Frequency	Haute fréquence
HFD	Higushi's Fractal Dimension	Dimension fractale de Higuchi
I²C	Inter-Integrated Circuit	-
I²S	Inter-IC Sound	-
ICA	Independent Component Analysis	Analyse en Composantes Indépendantes
IDE	Integrated Development Environment	l'environnement de développement intégré
IM	Motor Imagery	Imagination Motrice
INR	Intracortical Neuron Recording	Enregistrement Neuronal Intracortical
IoT	Internet of Things	Internet des objets
IRMf	Functional Magnetic Resonance Imaging	Imagerie par Résonance Magnétique Fonctionnelle
KDE	Kernel Density Estimate	Estimation de la densité du noyau
KNN	K-Nearest Neighbors	K-Voisins les plus proches
LCD	Liquid Crystal Display	Écran à cristaux liquides
LDA	Linear Discriminant Analysis	Analyse discriminante linéaire
LED	Light Emitting Diode	Diode à Émission Lumineuse
LFP	Local Field Potentials	Potentiels des Champs Locaux
MEG	Magnetoencephalography	Magnétoencéphalographie
micro SD	Micro Secure Digital	Micro Secure Digital
MISC	Miscellaneous	Divers
MLP	Multi-Layer Perceptron	Perceptron multicouche
MNB	Multinomial Naïve Bayes	Bayes naïfs multinomiaux
MPR	Myoelectric Pattern Recognition	Reconnaissance des formes myoélectriques
MSE	Mean Squared Error	Erreur quadratique moyenne
MUA	multiunit activity	Activité Multi-Unitaire
PCA	Principal Component Analysis	Analyse en Composantes Principales
PE	Evoked Potential	Potentiel Évoqué
PEA	Auditory Evoked Potentials	Potentiels Évoqués Auditifs
PES	Somesthetic Evoked Potentials	Potentiels Évoqués Somesthésiques
PET	Positron Emission Tomography	Tomographie par Emission de Positrons
PEV	Visual Evoked Potentials	Potentiels Évoqués Visuels
PGA	Programmable Gain Amplifier	Amplificateur à Gain programmbale
PM	Premotor	Pré-Moteur
PPSE	Excitatory Post-Synaptic Potential	Potentiel Post-Synaptique Excitateur
PPSI	Inhibitory Post-Synaptic Potential	Potentiel Post-Synaptique Inhibiteur
RDATAC	Read Data Continuous	
ReLU	Rectified Linear Unit	Unité Linéaire Rectifiée
RF	Radio-Frequency	Radio-fréquence

RLD	Right Leg Drive	-
SampEn	sampled entropy	entropie échantillonnée
SCP	Slow Cortical Potentials	Potentiels corticaux lents
SDATAC	Stop Read Data Continuous	Arrêter la lecture des données en continu
SFS	Sequential Features Selection	Sélection Séquentielle des Caractéristiques
SMR	SensoriMotor Rhythms	Rythmes Sensori-Moteurs
SNC	Central nervous system	Système Nerveux Central
SNP	Peripheral nervous system	Système Nerveux Périphérique
SNR	Signal Noise Ratio	Rapport Signal Bruit
SNR	Signal to Noise Ratio	rapport signal/bruit
SoC	System-on-a-Chip	Système Sur Puce
SPI	Serial Peripheral Interface	Interface périphérique série
SPS	Sample Per Second	Échantillon Par Seconde
SRB	Stimulus Reference Bias	Stimulus, Référence, Bias
SSAEP	Steady-State Auditory Evoked Potentials	Potentiels Évoqués auditifs à l'État d'Équilibre
SSEP	Steady-State Evoked Potentials	Potentiels Évoqués à l'État d'Équilibre
SSSEP	Steady-State somatosensory Evoked Potentials	Potentiels Évoqués somesthésiques à l'État d'équilibre
SSVEP	Steady-State Visual Evoked Potentials	Potentiels Évoqués Visuels à l'État d'Équilibre
STFT	Short Time Fourier Transform	Transformée de Fourier à court terme
SUA	Single Unit Activity	Activité d'une Seule Unité
SVM	Support Vector Machine	Machine à vecteurs de support
TCP/IP	Transmission Control Protocol/Internet Protocol	Protocole de contrôle de transmission/Protocole Internet
TEDAC	Telemetry Data Acquisition and Control	Acquisition et contrôle des données de télémétrie
Ti	Texas Instruments	-
TRMC	Common Mode Rejection Ratio	Taux de Réjection en Mode Commun
TVS	Transient Voltage Suppressor	Suppresseur de tension transitoire
UART	Universal Asynchronous Receiver-Transmitter	Récepteur-émetteur asynchrone universel
UCC		Unité de Contrôle et de Communication
UDP	User Datagram Protocol	Protocole de datagramme utilisateur
USB	Universal Serial Bus	Bus universel en série
VLSI	Very-Large-Scale Integration	Intégration à très grande échelle
VPP	Voltage Peak-to-Peak	Tension Crête-à-Crête

TABLE DES FIGURES

CHAPITRE I

Figure I.1. Organisation des nerfs rachidiens dans le système nerveux périphérique.

Figure I.2. Vue d'ensemble des 12 paires de nerfs crâniens.

Figure I.3. Coupe latérale de l'encéphale montrant les différents éléments qui composent sa structure profonde.

Figure I.4. (A) Schéma légendé de la structure du neurone. (B) Illustration de la zone synaptique et de la transmission électrochimique.

Figure I.5. Représentation temporelle du potentiel d'action neuronal.

Figure I.6. Vue générale de la surface du cortex, des principaux lobes, les sillons et les aires fonctionnelles.

Figure I.7. Cortex cérébral avec matière grise et blanche et les ventricules.

Figure I.8. Vue latérale du lobe insulaire.

Figure I.9. (A) Noyaux gris centraux. (B) physiologie du système sous cortical limbique.

Figure I.10. Homoncule de Penfield

Figure I.11. Décussation des voies nerveuses optiques

Figure I.12. Dynamique électrophysiologique des neurones pyramidaux corticaux : générateurs des potentiels de champ et leur impact sur l'électrogenèse de surface.

Figure I.13. (A) motifs des signaux cérébraux P300 pour le fonctionnement BCI, (B) Motif de rythme sensorimoteur, (C) Motif de potentiel évoqué visuel en état d'équilibre, (D) Motif de potentiel cortical lent.

Figure I.14. Les composantes principales du potentiel évoqué P300.

Figure I.15. Représentation schématique des trois paradigmes : stimulus unique (en haut), oddball (au milieu), et trois stimulus (en bas). L'ERP déclenché est présenté à droite.

Figure I.16. Le modèle temporo-spatial des ERPs dans la BCI auditive active : (A) Grande forme d'onde moyenne à l'électrode P3. (B) Topographie de l'amplitude moyenne à 155 ms (valeur N2) et 485 ms (pic LPC).

Figure I.17. (a) Un potentiel (SSVEP) mesuré sur le lobe occipital (Oz) par rapport à une référence (Cz) située sur le Vertex, (b), Le spectre de puissance montre clairement la composante fondamentale du SSVEP à 15 Hz, les autres composantes 30 Hz, 45 Hz sont des pics aux fréquences harmoniques supérieures.

Figure I.18. Un échantillon d'une activité μ sur une période d'une seconde.

Figure I.19. Exemples de schémas ERD et ERS (obtenus à partir de données EEG et IRMf) lors de mouvements réels et imaginaires du pied (gauche) ou de la main (droite). Le schéma antagoniste d'activation (ERD, signal BOLD positif) et de désactivation (ERS, signal BOLD

négatif) est clairement visible. On note également la similarité des cartes ERD pour les mouvements réels et imaginaires.

Figure I.20. Rythme sensorimoteur.

Figure 21. Potentiel cortical lent.

CHAPITRE II

Figure II.1. Structure fonctionnelle d'une interface cerveau-ordinateur illustrant la série de composantes utilisées pour traduire l'activité cérébrale en signaux de contrôle.

Figure II.2. Catégorisation des interfaces cerveau-ordinateur.

Figure II.3. Disponibilité du système pendant différents paradigmes de contrôle temporel.

Figure II.4. Mode de fonctionnement des Interfaces Cerveau-Ordinateur basées sur le paradigme SSVEP.

Figure II.5. Illustration du fonctionnement des interfaces cerveau-ordinateur combinant des protocoles synchrones et au rythme autonome.

Figure II.6. Le Système International 10-20 pour le positionnement des électrodes EEG.

Figure II.7. Le premier EEG enregistré par Hans Berger en 1924 et publié en 1929.

Figure II.8. (A) Dispositif MEG (B) illustration du détecteur MEG et des électrodes de stimulation (C) Enregistrement sur la carte cérébrale des zones d'activation.

Figure II.9. (A) Principe de Spectroscopie Proche de l'Infrarouge fonctionnelle. (B) Illustration du dispositif fNIRS.

Figure II.10. Principe de fonctionnement d'un IRM.

Figure II.11. (A) Image d'activation obtenue par imagerie par résonance magnétique fonctionnelle (IRMf) lors de la stimulation électrique (4 Hz) du doigt de la main gauche. Chaque stimulation dure 20 secondes, alternant avec des périodes de repos. (B Le signal BOLD présente des modifications toutes les 20 secondes, correspondant à la stimulation, avec un retour rapide à l'état de repos lorsque la stimulation est arrêtée.

Figure II.12. Étude du cerveau humain en imagerie multimodale, utilisant la TEP pour évaluer le débit sanguin cérébral.

Figure II.13. Structure du dispositif implantable Wimagine.

Figure II.14. Exemple d'un acte chirurgical pour implanter une interface MUA d'une BCI de type INR. **(*Illustration tirée de l'article en ligne : https://tpemainbionique.weebly.com/le-bci.html*)**

Figure II.15. Positionnement des électrodes suivant les différentes techniques d'acquisition BCI (Invasives, semi-invasives et non-invasives).

CHAPITRE III

Figure III.1. Schéma fonctionnel du système d'acquisition proposé.

Figure III.2. Kit d'électrodes en or Ag/AgCl et pâte conductrice Ten20 utilisés.

Figure III.3. Schéma fonctionnel simplifié de l'ADS1299-AFE.

Figure III.4. Configuration des broches de l'ADS1299-AFE à 8 Canaux

Figure III.5. Exécution de la commande RREG.

Figure III.6. Exécution de la commande WREG.

Figure III.7. Utilisation de la commande RDATAC.

Figure III.8. (A) Réponse en fréquence (jusqu'à 1 KHz) du filtre Sinc à décimation, (B) sa fonction de transfert jusqu'à 4 f_{mod}.

Figure III.9. Schéma interne du suppresseur ESD SN65240P.

Figure III.10. Schéma fonctionnel de l'ESP32-WROOM-32.

Figure III.11. Plateforme logicielle du système d'acquisition EEG sur le front-end et sur le back-end.

Figure III.12. Organisation de la bibliothèque compilable.

Figure III.13. Séquence d'initialisation du système d'acquisition EEG basé sur l'ESP32.

Figure III.14. Organigramme du flux de données EEG en modes de conversion continue et de lecture continue.

CHAPITRE IV

Figure IV.1. Prototype proposé.

Figure IV.2. Distribution du bruit référé à l'entrée.

Figure IV.3. Comportement du bruit référé à l'entrée en fonction du gain.

Figure IV.4. Données de séries temporelles du bruit référé à l'entrée capturé à partir du canal 8.

Figure IV.5. Occurrence de différents niveaux de bruit.

Figure IV.6. Mesure de la température de la puce sur 8 canaux.

Figure IV.7. Comportement du DC-Offset en fonction du réglage du gain.

Figure IV.8. Calibrage à l'aide d'un signal de test interne.

Figure IV.9. Rapport de réjection du mode commun dans la bande [1-120 Hz].

Figure IV.10. Positionnement des électrodes ECG sur le poignet, le muscle Brachioradialis, et sur le lobe de l'oreille A2.

Figure IV.11. (a) Série temporelle d'ECG et (b) périodogramme de Welch correspondant en dB.

Figure IV.12. (a) EEG brut prélevé respectivement sur FP1, FP2, C3, et O2, (b) Emplacement des électrodes EEG.

Figure IV.13. (a) Dérivation O2-A1 de l'EEG enregistré par CH7, (b) Spectrogramme O2-A1 montrant clairement les rythmes alpha (zones encerclées), (c) Densité spectrale O2-A1 avec périodogramme multicouches.

Figure IV.14. (a) EMG filtré par filtre passe-bande, (b) EMG brut rectifié, (c) Enveloppe EMG et (d) Périodogramme EMG.

CHAPITRE V

Figure V.1. Réponse fréquentielle d'un filtre Butterworth passe-bas d'ordre 1, 2, ...6.

Figure V.2. Extraction de 2000 échantillons des rythmes Alpha (8, 12 Hz) et Beta (13, 28 Hz) au niveau des électrodes C3-Cz-C4 en utilisant un filtre Butterworth passe-bande de 4ème ordre.

Figure V.3. Prétraitement du signal EEG mesuré au niveau du positionnement C3 suivant le système international 10-20 : (A) Signal EEG-C3 brut à 1000 Hz, (B) Signal EEG-C3 sous-échantillonné jusqu'à 100 Hz, (C) Signal EEG-C3 normalisé à l'aide de l'algorithme Robust-Scaler, la ligne de base (en marron) représente la dérive orignal du signal, (D) Signal EEG-C3 final normalisé et rectifié.

Figure V.4. Implémentation des fonctions de calculs analytiques et statistiques.

Figure V.5. Périodogrammes traditionnel, de Welch et Multitaper.

Figure V.6. Comparaison de la puissance de bande (Rythme bêta) : Welch vs. Multitaper.

Figure V.7. (A) Spectre par la transformée de Fourier à Court Terme STFT, (B) Spectrogramme associé du signal EEG au niveau de C3.

Figure V.8. Convolution du signal EEG avec une ondelette de Morlet : 1) Mise en échelle pour la détection des fréquences élevées : (A) Localisation centrée, (B) Ondelette Morlet décalée dans le temps. 2) Mise en échelle pour la détection des basses fréquences : (C) Localisation centrée, (D) Ondelette de Morlet décalée dans le temps.

Figure V.9. (A) Série temporelle du signal EEG-C3 dans l'intervalle (0, 40 s) comportant les marqueurs associés aux essais (trials), (B) Extraction du 1er essai (trial 0).

Figure V.10. Schématisation d'un mélange de l'activité cérébrale.

Figure V.11. Formation du plan de décision de KNN pour différentes valeurs de k : 1, 2 et 4 voisins.

Figure V.12. Données de caractéristiques à 3 classes.

Figure V.13. Vecteurs de Support et la Marge SVM.

Figure V.14. Cœur d'un réseau MLP pour l'apprentissage profond

CHAPITRE VI

Figure VI.1. Estimation de la densité spectrale de puissance Welch des canaux C3, Cz et C4.

Figure VI.2. Estimation de la densité spectrale de puissance Welch dans la plage (8, 15 Hz) des canaux C3, Cz et C4.

Figure VI.3. Histogramme illustrant la discrimination réduite en raison de la faible variance logarithmique des données brutes.

Figure VI.4. Histogramme illustrant la discrimination des caractéristiques par la maximisation de la variance au moyen de la technique du CSP.

Figure VI.5. Périodogramme de Welch pour les trois composantes discriminantes : 1, 29 et 59.

Figure VI.6. Résultats de la classification par la matrice de confusion (A) KNN, (B) LDA, (C)SVM.

Figure VI.7. Évaluation des performances du modèle Pytorch au cours de la phase d'apprentissage.

Figure VI.8. Montage des électrodes des trois canaux EEG bipolaires (C3, C4 et Cz).

Figure VI.9. Les coefficients de décomposition de 6 niveaux pour la classe 1 (Left) des canaux C3 et C4.

Figure VI.10. Les coefficients de décomposition de 6 niveaux pour la classe 1 (Left) des canaux C3 et C4.

Figure VI.11. Taux de la variance expliquée cumulée en fonction du nombre de composantes, et le seuil de variance est un critère de choix.

Figure VI.12. Illustration 2D et 3D des trois premières composantes principales de l'ensemble d'entrainement *pca_data_T* calculées par PCA.

Figure VI.13. Évaluation des performances du modèle d'apprentissage profond en terme de la fonction de coût (perte) durant le processus d'apprentissage.

LISTE DES TABLEAUX

CHAPITRE III

Tableau III.1. Etude comparative des modules d'acquisition EEG trouvés dans la littérature.

Tableau III.2. Types d'électrodes préférés pour l'EEG, l'ECG et l'EMG.

Tableau III.3. Spécifications et conditions opérationnelles de l'ADS1299-AFE.

Tableau III.4. Définitions des commandes associées à l'ADS1299-AFE.

Tableau III.5. Conditions de fonctionnement des régulateurs de tension utilisés.

Tableau III.6. Comparaison de l'ESP 32 avec les modules concurrents les plus connus.

CHAPITRE IV

Tableau IV.1. Conditions d'évaluation expérimentale de l'ADS1299-AFE.

Tableau IV.2. Analyse statistique du bruit référé à l'entrée

Tableau IV.3. Variation de la valeur crête-à-crête du bruit référé à l'entrée en fonction du gain.

Tableau IV.4. Mesure des critères sous-jacents en fonction du gain, pour un taux de transfert de 250 éch. /s.

Tableau IV.5. Rapport de spécifications par rapport à l'EVM ADS1299EEG-FE

CHAPITRE VI

Tableau VI.1. Récapitulatif des performances de la classification par les estimateurs de step2bci.

TABLE DES MATIERES

Remerciements ii

Abstract iii

Résumé iv

ملخص v

Abréviations vi

Tables des Figures ix

Liste des Tableaux xiv

Introduction générale 1

Chapitre I

I. Anatomie Fonctionnelle Et Electrogenèse : Clés De Compréhension Des Interfaces Cerveau-Ordinateur (BCIs). 8

I.1. Introduction 8

I.2. Bases neuro-anatomiques et anatomiques fonctionnelles 8

I.3. Système nerveux 8

I.3.1.Système Nerveux Périphérique – SNP 8

I.3.2.Système Nerveux Central – SNC 10

I.3.2.1. Structure du neurone et le message nerveux 12

I.3.2.2. Anatomie fonctionnelle du cerveau 13

I.3.2.3. Les aires motrices 17

I.3.2.4. Les aires sensorielles somesthésiques 18

I.3.2.5. Les aires sensorielles visuelles 19

I.3.2.6. Les aires sensorielles auditives 19

I.4. Dynamique neuronale et Électrogenèse corticale 20

I.4.1.Production de l'EEG de surface 21

I.5. Motifs d'intérêt pour les interfaces cerveau-ordinateur 23

I.5.1.Les composantes d'intérêt 24

I.5.1.1. Activité cérébrale spontanée 24

I.5.1.2. Potentiels évoqués – PE 24

I.5.1.3. Potentiels liés aux évènements - ERP 25

I.5.1.4. Potentiel P300 endogène 28
I.5.1.5. Potentiels évoqués à l'état d'équilibre 31
I.5.1.6. Rythmes sensori-moteurs 36
I.5.1.7. Potentiels corticaux lents 38
I.6. Conclusion 39
Bibliographie du Chapitre I 40

Chapitre II

II. Interfaces Cerveau-Ordinateur (BCIs). 47
II.1. Introduction 47
II.2. Présentation des interfaces cerveau-ordinateur 48
II.3. Système Acquisition des signaux EEG 48
II.3.1. Prétraitement 48
II.3.2. Posttraitement 48
II.4. Catégorisation des interfaces Cerveau-Ordinateur 48
II.5. Interfaces Cerveau ordinateur catégorisées selon le protocole de fonctionnement 50
II.5.1. Interfaces Cerveau-Ordinateur avec des protocoles synchrones (Cue-based) 51
II.5.2. Interfaces Cerveau-Ordinateur basées dur des protocoles au rythme autonome (Self-paced) 52
II.5.3. Interfaces Cerveau-Ordinateur basées sur des protocoles combinés synchrones et au rythme autonome 53
II.5.4. Interfaces Cerveau-Ordinateur basées sur des protocoles non-adaptatifs 54
II.5.5. Interfaces Cerveau-Ordinateur basées sur des protocoles adaptatifs 54
II.6. Interfaces Cerveau ordinateur catégorisées selon le motif cérébral 55
II.7. Interfaces Cerveau-ordinateur catégorisées suivant le mode d'acquisition 57
II.7.1. Mode d'acquisition non-invasif 55
II.7.1.1. Électroencéphalographie – EEG 57
II.7.1.2. Magnétoencéphalographie – MEG 60
II.7.1.3. Spectroscopie Proche de l'Infrarouge fonctionnelle – fNIRS 62
II.7.1.4. Imagerie par Résonance Magnétique fonctionnelle – IRMf 63
II.7.1.5. Tomographie par Emission de Positrons – TEP 65
II.7.2. Mode d'acquisition semi-invasif 66
II.7.2.1. Électrocorticographie – EcoG 66
II.7.3. Mode d'acquisition invasif 68

II.7.3.1. Enregistrement Neuronal Intracortical – INR 68
II.8. Interfaces Cerveau-ordinateur catégorisées suivant la stratégie de fonctionnement 70
II.9. Conclusion 71
Bibliographie du Chapitre II 72

Chapitre III

III. Système D'acquisition De L'activité EEG Pour Une Interface Cerveau-Ordinateur. 82
III.1. Introduction 82
III.2. État de l'art et travaux similaires 85
III.3. Architecture du système d'acquisition EEG 89
III.3.1. Conception du matériel 89
III.3.1.1. Électrodes 90
III.3.1.2. Frontal analogique ADS1299-AFE 92
III.3.1.3. Filtre passe-bas anti repliement 99
III.3.1.4. Dispositifs de protection 100
III.3.1.5. Alimentation électrique analogique et numérique 101
III.3.1.6. Unité de contrôle et de communication – UCC 102
III.3.2. Présentation du logiciel 106
III.3.2.1. Bibliothèque compilable 107
III.3.2.2. Firmware 108
III.3.2.3. Pilote 110
III.3.2.4. Cadre d'acquisition, de visualisation et de contrôle-AVCF 111
III.4. Conclusion 112
Bibliographie du Chapitre III 113

Chapitre IV

IV. Évaluation Global Du Système D'acquisition EEG. 118
IV.1.Introduction 118
IV.2. Analyse des performances du système d'acquisition 119
IV.2.1. Bruit référé à l'entrée 119
IV.2.2. Mesure de la température inhérente 123
IV.2.3. Mesure du décalage de la composante continue (Offset-CC) 124

IV.2.4. Étalonnage à l'aide du signal de test interne 125
IV.2.5. Taux de réjection en mode commun (TRMC) 126
IV.2.6. Rapport de performance comparé à un modèle de référence 127
IV.3.Mesure du signal ECG 128
IV.4.Mesure du signal EEG 129
IV.5.Mesure du signal EMG 130
IV.6.Conclusion 132
Bibliographie du Chapitre III 134

Chapitre V

V. Développement De la Boite À Outils Step2bci. 135
V.1. Introduction 135
V.1.1. Pourquoi Python ? 135
V.2. État de l'art est travaux similaires 136
V.3. Présentation de la boite à outils step2bci 139
V.4. Prétraitement du signal EEG 140
V.4.1. Structure des données EEG 140
V.4.2. Filtrage numérique 141
V.4.3. Sous-échantillonnage 142
V.4.4. Normalisation et échelonnage des données EEG 144
V.4.5. Correction de la ligne de base 145
V.4.6. Fonctions de calculs analytiques et statistiques 146
V.4.7. Fonctions de l'analyse spectrale et fréquentielle 147
V.4.7.1. Densité spectrale de puissance (DSP) 148
V.4.7.2. Spectrogramme via la transformée de Fourier à Court Terme (STFT) 148
V.4.7.3. Périodogrammes 149
V.4.7.4. Analyse de la puissance de bande 151
V.4.8. Décomposition en ondelettes et analyse multi-résolution 153
V.4.9. Segmentation (Epoching) des données et l'extraction des essais 157
V.5. Post-traitement du signal EEG 158
V.5.1. Extraction des caractéristiques 159
V.5.1.1. Extraction multi-résolution par décomposition en ondelettes 159
V.5.1.2. Extraction par les motifs spatiaux communs (filtrage spatiale) 159

V.5.1.3. Extraction de la variance logarithmique 161
V.5.1.4. Extraction de la puissance des sous-bandes fréquentielles 162
V.5.1.5. Extraction des caractéristiques par l'entropie échantillonnée (SampEn) 162
V.5.2. Sélection et réduction de l'espace des caractéristiques 164
V.5.2.1. Sélection séquentielle des caractéristiques (SFS) 164
V.5.2.2. Sélection par l'analyse des composantes principales (PCA) 165
V.5.2.3. Sélection par analyse en composantes indépendantes (ICA) 167
V.5.3. Classification par Apprentissage automatique 170
V.5.3.1. Classifieur K-plus proches voisins (KNN) 171
V.5.3.2. Classifieur d'analyse discriminante linéaire (LDA) 172
V.5.3.3. Classifieur Machine à vecteurs de support (SVM) 175
V.5.4. Classification par apprentissage profond 180
V.5.4.1. Module de gestion des dispositifs (deepDevice) 186
V.5.4.2. Module de personnalisation et de transformation des données EEG (neuroDataset) 187
V.5.4.3. Module d'apprentissage profond (deepBCI) 187
V.6. Conclusion 188
Bibliographie du Chapitre V 191

Chapitre VI

VI. Protocole Expérimentale Hors-Ligne D'une BCI Via La Boite À Outils Step2bci 198
VI.1. Introduction 198
VI.2. Protocoles réels de la BCI hors ligne utilisant step2bci 198
VI.3. Protocole d'imagination motrice avec une rétroaction asynchrone 199
VI.3.1. Présentation des données (Dataset 1) 199
VI.3.2. Préparation des données 200
VI.3.3. Création des ensembles d'entraînement et de test 201
VI.3.4. Calcul des caractéristiques spectrales par le périodogramme de Welch 202
VI.3.5. Application de la technique CSP (Common Spatial Pattern) 206
VI.3.6. Classification 208
VI.3.6.1. Sélection et réduction de la dimensionnalité des caractéristiques 208
VI.3.6.2. Sélection d'estimateurs 209

VI.3.6.3. Création des modèles d'apprentissage automatique types à l'aide de la boite à outils Step2bci 210
VI.3.6.4. Création du modèle d'apprentissage profond à l'aide de la boîte à outils Step2bci 212
VI.3.7. Interprétation et discussion des résultats 215
VI.4. Protocole d'imagination motrice avec et sans rétroaction (feedback) 216
VI.4.1. Présentation des données 216
VI.4.2. Préparation des données 217
VI.4.3. Création des ensembles d'apprentissage et de test 218
VI.4.4. Extraction des caractéristiques par décomposition en ondelettes 218
VI.4.5. Réduction de dimensionalité 222
VI.4.5.1. Approche PCA 222
VI.4.5.2. Approche SFS 224
VI.4.6. Classification par apprentissage profond à l'aide de la boîte à outils Step2bci 225
VI.4.7. Interprétation et discussion des résultats 228
VI.5. Conclusion 229
Bibliographie du Chapitre VI 231

Conclusion générale 232

Annexes 234

INTRODUCTION GENERALE

Le cerveau humain, organe remarquable encapsulé dans la boîte crânienne, a longtemps captivé l'imagination des scientifiques, des philosophes et des érudits à travers l'histoire [Penfield, W. 2015, Ackerman, D. 2004]. Ses subtilités, la complexité de ses fonctions et les mystères qui entourent encore son fonctionnement interne continuent d'être des sujets de fascination intense.

Dans le domaine des neurosciences et de la technologie, notre voyage commence par une exploration des aspects fondamentaux de l'anatomie fonctionnelle du cerveau, de l'orchestration de son activité électrique et des origines énigmatiques de son électrogenèse, d'où la fusion de l'exploration des fonctions cérébrales et de l'ingénierie de pointe a donné naissance à un domaine au potentiel et à l'innovation immenses : Les interfaces cerveau-ordinateur BCI (*Brain-Computer Interface*). Au carrefour des neurosciences et de la technologie, en constante évolution, la danse énigmatique des fonctions cérébrales rencontre la précision des prouesses techniques. C'est à cette intersection dynamique que se situe les interfaces cerveau-ordinateur. Alors que nous nous plongeons dans ce monde fascinant, il est essentiel de comprendre d'abord les concepts fondamentaux qui sous-tendent ce champ multidisciplinaire [Li, C., & Zhao, W. 2020].

Terminologie et histoire des interfaces cerveau-ordinateur :

Avant de plonger dans les profondeurs de notre quête, il est essentiel d'établir un lexique commun et de se passer brièvement dans le contexte historique des interfaces cerveau-ordinateur. Les BCIs, parfois appelées de manière interchangeable interfaces cerveau-machine (ICM), ne sont pas seulement des merveilles technologiques de notre époque ; elles ont un passé chargé d'histoire qui remonte à la confluence de la curiosité scientifique et de l'innovation technologique. Cette section démêle les subtilités terminologiques et retrace la trajectoire historique des BCIs, depuis leur conceptualisation jusqu'à leur évolution en outils puissants dotés d'un vaste potentiel dans les domaines des soins de santé, de la communication et au-delà.

Des expériences précoces, telles que les expériences d'électroencéphalographie (EEG) pionnières de Hans Berger dans les années 1920 [İnce, R., Adanır, S. S., & Sevmez, F. 2021] aux techniques innovantes de Wilder Penfield et Herbert Jasper [Penfield, W., 1937 ; Penfield, W., & Jasper, H. 1954], nous montrent les étapes initiales de l'enregistrement des signaux neuronaux. Cette période a vu la naissance de l'EEG en tant qu'outil puissant pour l'analyse des ondes cérébrales, un moment épique de l'histoire des BCIs.

La période allant des années 1940 aux années 1960 a été marquée par des avancées significatives dans la compréhension de l'activité cérébrale, grâce à des pionniers tels que Grey Walter [Walter, W. G. 1950], Rasmussen et Penfield [Penfield, W. et al., 1949]. Ces développements ont jeté les bases de l'émergence des BCIs en tant que domaine de recherche et de technologie évolutionnaire.

Au cours des décennies des années 1960 et 1970, des chercheurs pionniers dans le domaine des Interfaces Cerveau-Ordinateur (BCI), notamment G. Walter, J.J. Vidal et M.B. Sterman, ont entrepris des explorations significatives dans le domaine des BCIs non invasives. Cette période a coïncidé avec l'introduction du terme "cyborg" par Manfred Clynes et Nathan S. Kline en 1960, dans leur article influent intitulé "Cyborgs and Space" [Clynes, M. E., et al., 1960]. Cette publication visionnaire a traité de la fusion de l'homme avec la technologie et a ouvert la voie à des discussions plus étendues sur les interfaces homme-machine. Bien que leurs travaux initiaux ne fussent pas directement associés aux BCIs, les concepts développés par Clynes et Kline ont grandement contribué à la conceptualisation ultérieure des BCIs en tant que moyen de combler le fossé entre les humains et les machines.

En 1973, J.J. Vidal a marqué un jalon important en entreprenant les premières expérimentations concrètes visant à utiliser l'activité cérébrale, mesurée par électroencéphalographie (EEG), pour le contrôle de systèmes électroniques [Vidal, J.J. 1973]. Parallèlement, les travaux de M. B. Sterman sur le contrôle du rythme sensorimoteur SMR (*Sensori-Motor Rhythm*) ont anticipé le développement du neurofeedback [Sterman, M. B., 1974], jetant ainsi les bases des paradigmes contemporains des BCIs.

L'impact de ces pionniers est solidement ancré dans les annales de la littérature scientifique, où leurs travaux fondateurs continuent de servir de pierres angulaires à notre compréhension des BCIs et de leur évolution au fil du temps.

La fin du XXe siècle a ouvert une nouvelle ère de recherche en BCI avec le développement des potentiels liés à des événements ERPs (*Event Related Potentials*) et des BCI basées sur les potentiels évoqués P300 [Dino, N. ,1980]. La découverte et l'utilisation de ces phénomènes électrophysiologiques ont illuminé les possibilités des BCIs non invasives pour la communication et le contrôle. Le voyage s'est poursuivi avec le raffinement des Interfaces Cerveau-Machine (ICM) et l'émergence des BCIs électrocorticographiques (ECoG), qui ont établi un pont entre les modalités invasives et non invasives [Hill, J. et al., 1980]. Les BCIs sont passés de la science-fiction et du domaine des expériences contrôlées à des applications réelles, notamment des aides à la communication pour les personnes gravement handicapées, des jeux et la rééducation cognitive.

Aujourd'hui, les BCIs sont à l'aube d'un potentiel de transformation qui englobe des domaines aussi divers que la neuro-réhabilitation, la neuro-ergonomie et le neuro-jeu. Comprendre le parcours qui nous a menés jusqu'ici n'est pas seulement une curiosité historique, mais une base essentielle sur laquelle nous construirons l'avenir de la technologie des BCIs.

Problématique : Accessibilité et exhaustivité

Notre exploration des BCIs nous amène à un point critique : la reconnaissance des défis et des lacunes dans le paysage actuel des systèmes BCIs. À une extrémité du spectre se trouvent les systèmes BCI de qualité professionnelle, formidables dans leurs capacités mais souvent d'un prix exorbitant, encombrants et inaccessibles aux chercheurs novices ou au grand public. À l'autre extrémité se trouvent des alternatives abordables sur le marché mais sont souvent dépourvues du logiciel d'accompagnement essentiel et des boîtes à outils pour la mise en œuvre des différents protocoles et paradigmes expérimentaux ou de compatibilité avec les structures de données standard qui sont indispensables pour une exploration et une expérimentation significatives dans la recherche sur les BCIs, ce qui les rend moins idéales pour des recherches ou des développements complets au sein des laboratoires de recherche.

Les défis ne se limitent pas au prix et à l'accessibilité ; ils s'étendent au manque d'orientation globale dans la sphère universitaire. Une profusion d'études sur les BCIs peuplent le paysage académique, mais nombre d'entre elles restent parfois fragmentées, ne fournissant pas d'informations opérationnelles approfondies ni de lignes directrices complètes en matière de développement. Les chercheurs se retrouvent avec des pièces de puzzle, mais pas d'image définitive de la manière de construire, de modifier ou de personnaliser les systèmes BCI. C'est dans ce creuset de défis qu'émerge l'essence de notre recherche. L'objectif de cette thèse est double : créer un système BCI complet et accessible et documenter cette création en tant que guide complet pour les développeurs, la transformant ainsi en un véritable guide pratique.

Objectif de la thèse : Combler les lacunes

À la lumière de ces défis, cette thèse émerge avec un objectif résolu : concevoir un système BCI complet et open-source qui comble ces lacunes. Notre solution vise à démocratiser la technologie BCI, en la rendant accessible, adaptable et abordable. Elle porte la marque de la performance, de l'efficacité et de la précision, dans le but premier d'encourager l'innovation et d'accélérer le rythme de la recherche et du développement en matière des BCIs.

Au fond, cette thèse est un témoignage de la fusion des idées théoriques et des applications pratiques. Son objectif principal est de concevoir un écosystème BCI complet qui englobe le matériel, les logiciels et une autre panoplie d'outils. Cet écosystème comprendra un

système d'acquisition EEG construit sur le frontal analogique ADS1299-AFE fabriqué par Ti (*Texas Instruments*) et le fameux microcontrôleur Espressif- ESP-WROOM-32. Ce système doit être accompagné d'un micrologiciel et d'un puissant Framework pour l'acquisition, la visualisation et le contrôle (AVCF) fonctionnant sur une plateforme RaspberryPi 4. Le cœur de cet écosystème est la boîte à outils "Step2BCI", méticuleusement conçue pour faciliter les paradigmes de conception BCI grâce à une pléthore d'outils. *Step2BCI* est un arsenal basé sur Python qui regorge de progiciels de prétraitement et de post-traitement, peuplés par des techniques d'apprentissage automatique et d'apprentissage profond. Ensemble, ces éléments sont conçus pour permettre aux développeurs de mener à bien leurs projets BCI en répondant à leurs besoins précis.

Organisation de la thèse

Cette thèse se déroule en plusieurs chapitres méticuleusement élaborés, chacun s'appuyant sur le précédent pour créer un récit complet.

Le chapitre I intitulé " ***Anatomie fonctionnelle et électrogenèse : Clés de compréhension des interfaces cerveau-ordinateur***", se penche sur les fondements théoriques, il explore les concepts de la neuroanatomie et de l'anatomie fonctionnelle, en mettant l'accent sur le système nerveux et la structure des neurones. Il examine également la dynamique neuronale et l'électrogenèse corticale, ainsi que la production de l'EEG de surface. Enfin, il identifie les motifs d'intérêt pour les interfaces cerveau-ordinateur, en soulignant l'importance de comprendre l'activité cérébrale pour le développement des BCIs.

Le chapitre II intitulé "***Interfaces Cerveau-Ordinateur***", se concentre sur les interfaces cerveau-ordinateur (BCIs) en tant que méthode de communication directe entre l'activité cérébrale et des dispositifs externes. Il explore les applications des BCIs, notamment dans la rééducation, la robotique, le divertissement, la réalité virtuelle et plus encore. Le chapitre présente également les systèmes BCI, leur état de l'art, et aborde des aspects tels que l'acquisition des signaux EEG et la catégorisation des BCI en fonction de différents critères, offrant ainsi un aperçu complet de ce domaine en pleine expansion.

Le chapitre III intitulé "***Système d'Acquisition de l'activité EEG pour une Interface Cerveau-Ordinateur*** ", constitue le cœur de la mise en œuvre pratique, où nous détaillons la configuration matérielle de notre système d'acquisition EEG. Le chapitre souligne l'objectif principal du système d'acquisition EEG basé sur l'ADS1299-AFE à 8 canaux, à savoir fournir un modèle portable, économique et accessible pour la recherche en neurosciences. Il explore en détail l'architecture matérielle du système, notamment la conception des électrodes, le

frontal analogique ADS1299-AFE, le filtre passe-bas anti-repliement, les dispositifs de protection, l'alimentation électrique analogique et numérique, ainsi que l'unité de contrôle et de communication (UCC) basée sur le microcontrôleur ESP32-D0WDQ6. Ce chapitre aborde également la composante logicielle du système, décrivant la bibliothèque compilable, le micrologiciel (*Firmware*), le pilote, et le cadre d'acquisition, de visualisation, et de contrôle (AVCF). Dans l'ensemble, ce chapitre fournit un aperçu complet du système d'acquisition EEG développé dans le cadre de cette thèse, mettant en avant sa portabilité, son coût réduit, et son accessibilité pour la communauté de recherche en neurosciences.

Une fois le système en place, le chapitre IV intitulé ***"Évaluation Globale des Performances du Système d'Acquisition"***, met l'accent sur l'évaluation rigoureuse de ses performances. Ce processus est essentiel pour évaluer l'efficacité et la précision du dispositif d'acquisition EEG. Dans ce chapitre, un protocole expérimental détaillé est proposé pour évaluer le fonctionnement du système dans un environnement non protégé, ce qui est crucial pour déterminer son adéquation à des applications pratiques. L'objectif principal de ce chapitre est de fournir une analyse approfondie des performances du système, en mettant l'accent sur la qualité des signaux acquis. Plusieurs paramètres critiques, tels que le bruit référé à l'entrée, la température inhérente, le décalage de la composante continue (Offset-CC), l'étalonnage à l'aide du signal de test interne, le taux de réjection en mode commun (TRMC), ainsi que la comparaison des performances par rapport à un modèle de référence EVM ADS1299EEG-FE23, ont été rigoureusement évalués. La deuxième partie du chapitre se concentre sur la mesure des biopotentiels, comprenant l'analyse des signaux ECG, EEG, et EMG. Ces mesures biopotentielles sont cruciales pour évaluer la capacité du système à capturer efficacement les différents signaux électrophysiologiques.

Le chapitre V intitulé "***Développement de la Boite à Outils BCI en Python (step2bci)"***, se focalise sur la création d'une boîte à outils BCI open-source en Python, nommée step2bci. Il explique pourquoi Python a été choisi comme langage de programmation, passe en revue les travaux similaires existants, et présente en détail les fonctionnalités de la boîte à outils. Step2bci offre des outils pour le prétraitement et le post-traitement des signaux EEG, y compris des fonctions d'extraction de caractéristiques et de classification. Cette boîte à outils est conçue pour être accessible et fonctionner sur divers systèmes d'exploitation. Elle joue un rôle essentiel dans le développement de systèmes BCI personnalisés et constitue une ressource précieuse pour la communauté de recherche en neurosciences.

Le chapitre VI, intitulé ***"Protocole Expérimental Hors-ligne d'une BCI via la Boite à Outils Step2bci"***, met en œuvre la boîte à outils step2bci pour démontrer son efficacité

dans des protocoles BCI hors ligne. Il explique en détail comment cette boîte à outils est utilisée pour préparer, prétraiter et classifier les données EEG, en mettant l'accent sur l'implémentation du code. L'objectif est de montrer comment step2bci peut être appliqué dans des cas réels en utilisant des ensembles de données de tierces parties provenant de laboratoires spécialisés, notamment ceux de l'Institut de technologie de Berlin (ALL) et l'Université de technologie de Graz (AUT). Le chapitre se concentre sur un protocole d'imagination motrice avec rétroaction asynchrone, détaillant chaque étape du processus, de la préparation des données à l'interprétation des résultats. De plus, il aborde un protocole d'imagination motrice avec et sans rétroaction, explorant la réduction de la dimensionnalité et l'utilisation de l'apprentissage profond avec step2bci pour classifier les données. Finalement, il discute des résultats obtenus et de leur signification.

Enfin, la thèse conclut en résumant le travail monumental entrepris et trace la voie pour les améliorations et applications futures de cet écosystème BCI. Il ne s'agit pas seulement d'une thèse, mais d'un guide pour les développeurs et les chercheurs, garantissant que le monde des interfaces cerveau-ordinateur reste ouvert, accessible et plein d'opportunités d'innovation.

BIBLIOGRAPHIE DE L'INTRODUCTION GENERALE :

Penfield, W. (2015). Mystery of the mind: A critical study of consciousness and the human brain. Princeton University Press.

Ackerman, D. (2004). An alchemy of mind: The marvel and mystery of the brain. Simon and Schuster.

Li, C., & Zhao, W. (2020). Progress in the brain–computer interface: an interview with Bin He. National Science Review, 7(2), 480-483.

İnce, R., Adanır, S. S., & Sevmez, F. (2021). The inventor of Electroencephalography (EEG): Hans Berger (1873–1941). Child's Nervous System, 37, 2723-2724.

Penfield, W., & Boldrey, E. (1937). Somatic motor and sensory representation in the cerebral cortex of man as studied by electrical stimulation. Brain, 60(4), 389-443.

Penfield, W., & Jasper, H. (1954). Epilepsy and the functional anatomy of the human brain.

Walter, W. G., & Walter, V. J. (1949). The electrical activity of the brain. Annual review of physiology, 11(1), 199-230.

Penfield, W., & Rasmussen, T. (1949). Vocalization and arrest of speech. Archives of Neurology & Psychiatry, 61(1), 21-27.

Clynes, M. E., & Kline, N. S. (1960). Cyborgs and space. Astronautics, 14(9), 26-27.

Vidal, J. J. (1973). Toward direct brain-computer communication. Annual review of Biophysics and Bioengineering, 2(1), 157-180.

Sterman, M. B., Macdonald, L. R., & Stone, R. K. (1974). Biofeedback training of the sensorimotor electroencephalogram rhythm in man: effects on epilepsy. Epilepsia, 15(3), 395-416.

Dino, N. (1980). A design development of building system for mass housing (Master's thesis, Middle East Technical University.

Hill, J., Farquhar, J., Grosse-Wentrup, M., Martens, S., & Schölkopf, B. (1988). Development of Brain-Computer Interface Systems. Neurophysiology, 70(6), 510-523.

CHAPITRE I

ANATOMIE FONCTIONNELLE ET ELECTROGENESE : CLES DE COMPREHENSION DES INTERFACES CERVEAU-ORDINATEUR (BCI)

CONTENU

I.1. Introduction
I.2. Bases neuro-anatomiques et anatomiques fonctionnelles
I.3. Système nerveux
I.3.1. Système Nerveux Périphérique - SNP
I.3.2. Système Nerveux Central – SNC
I.3.2.1. Structure du neurone et le message nerveux
I.3.2.2. Anatomie fonctionnelle du cerveau
I.3.2.3. Les aires motrices
I.3.2.4. Les aires sensorielles somesthésiques
I.3.2.5. Les aires sensorielles visuelles
I.3.2.6. Les aires sensorielles auditives
I.4. Dynamique neuronale et Électrogenèse corticale
I.4.1. La production de l'EEG de surface
I.5. Motifs d'intérêt pour les interfaces cerveau-ordinateur
I.5.1. Les composantes d'intérêt
I.5.1.1. Activité cérébrale spontanée
I.5.1.2. Potentiels évoqués - PE
I.5.1.3. Potentiels liés aux évènements
I.5.1.4. Potentiel P300 endogène
I.5.1.5. Potentiels évoqués à l'état d'équilibre
I.5.1.6. Rythmes sensori-moteurs
I.5.1.7. Potentiels corticaux lents
I.6. Conclusion
Bibliographie du Chapitre I

I.1. Introduction

Le cerveau est l'organe central du système nerveux humain, il est le siège d'une activité neuronale très complexe qui s'y déroule en permanence [F. L. Da Silva, 2010]. Cette activité est à l'origine une dépolarisation synchronisée [Sanei S. et al., 2007] d'une population de neurones suite à leurs excitations qui se traduit métaboliquement par des flux ioniques transmembranaires. Cependant, plusieurs pathologies et maladies neurodégénératives peuvent être responsables d'un dysfonctionnement neurologique ou cognitif que l'on peut citer : l'épilepsie, le stress, la perte d'attention et de mémoire, la sclérose en plaques, les troubles du comportement, la démence, l'encéphalite, etc. qui entraîne par la suite une altération de l'activité cérébrale spontanée [Sanei S. et al., 2007].

I.2. Bases neuro-anatomiques et anatomiques fonctionnelles

L'objectif de cette section est de décrire l'anatomie du système nerveux humain dans le contexte de réalisation des interfaces cerveau-ordinateur. Du fait que l'interprétation de l'activité électrique à la surface du cerveau n'est possible qu'à la condition de comprendre l'architecture du cortex cérébral.

Nous allons décrire les bases de la neuro-anatomie et l'anatomie fonctionnelle cérébrale. L'attention est portée sur la description du système nerveux central (SNC) et le système nerveux périphérique (SNP).

I.3. Système nerveux

Le système nerveux est le système qui permet à l'organisme humain de s'informer des changements de son environnement par des organes sensoriels et d'y réagir par des réponses effectrices adaptées. La structure du système nerveux est constituée d'un système nerveux central (SNC) et un système nerveux périphérique (SNP).

I.3.1. Système Nerveux Périphérique - SNP

Le SNP est un vaste réseau de nerfs qui commence par les racines provenant du système nerveux central (la moëlle épinière et l'encéphale), sa fonction principale est de relier le système nerveux central (SNC) aux membres et aux organes, par les nerfs. Le SNP est composé de 31 paires de nerfs rachidiens et 12 pairs de nerfs crâniens.

Les 31 nerfs rachidiens sont immergés de la moëlle épinière au niveau de la colonne vertébrale qui ont à la fois des racines dorsales afférentes responsables de la conduite de l'influx nerveux sensitif vers la moëlle épinière (SNC) et des racines ventrales efférentes responsables de

la conduite de l'influx nerveux moteur du SNC vers les organes effecteurs [Martin, J. H., 2012]. Comme illustrée par la figure I.1 on peut distinguer :

- 8 nerfs cervicaux de C1 à C8.
- 12 nerfs thoraciques de T1 à T12.
- 5 nerfs lombaires de L1 à L5.
- 5 nerfs sacrés de S1 à S5.
- 1 nerf coccygien - Co.

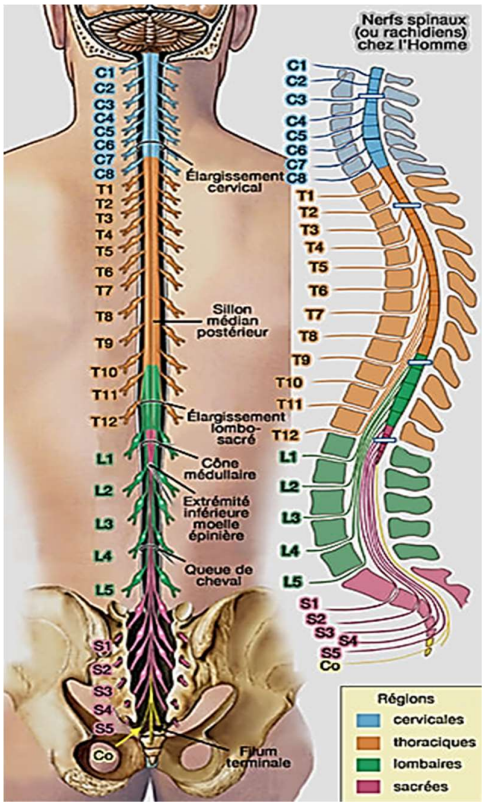


Figure I.1. Organisation des nerfs rachidiens dans le système nerveux périphérique.

Les 12 paires de nerfs crâniens, qui prennent naissance au niveau du cerveau et du tronc cérébral (figure I.2), sont constitués de :

- 5 nerfs moteurs : le nerf pathétique, le nerf oculaire externe, le nerf oculomoteur, le nerf spinal et le nerf hypoglosse.
- 3 nerfs sensoriels : le nerf optique, le nerf auditif et le nerf olfactif.
- 4 nerfs mixtes : le nerf facial, le nerf trijumeau, le nerf pneumogastrique et le nerf glossopharyngien.

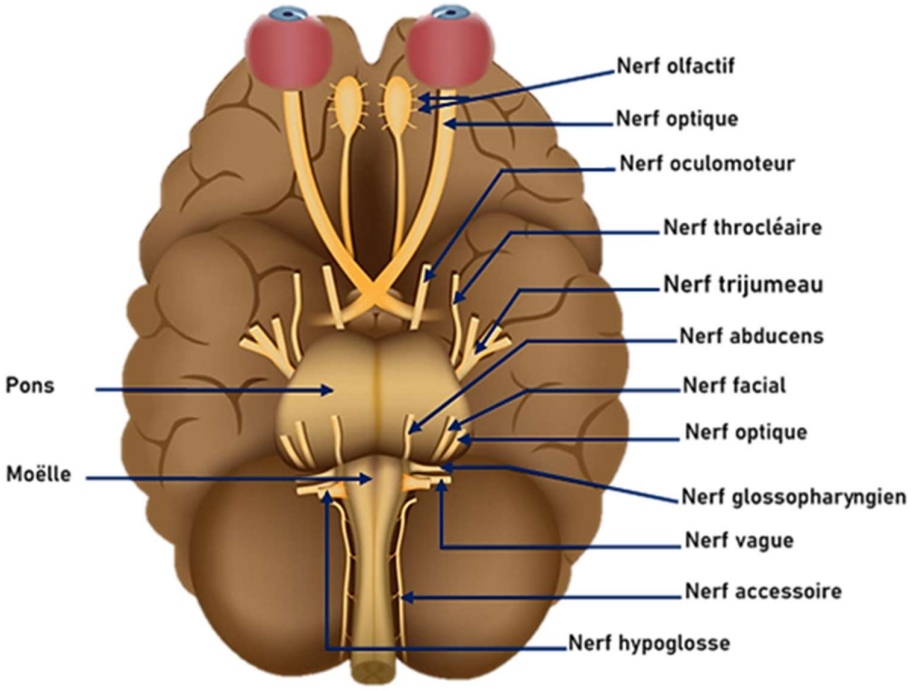


Figure I.2. Vue d'ensemble des 12 paires de nerfs crâniens.

Il existe un système nerveux autonome (ou végétatif) qui est une partie du SNP, sa fonction principale est de réguler les processus physiologiques involontaires, notamment le la respiration, le rythme cardiaque, la digestion et l'excitation, la pression sanguine, etc. le système nerveux végétatif est constitué de trois branches anatomiques distinctes : sympathique, parasympathique et entérique.

I.3.2. Système Nerveux Central - SNC

Le système nerveux central comprend l'encéphale (cerveau, cervelet et tronc cérébral), qui est logé dans la boite crânienne, et la moelle épinière contenue dans le canal rachidien. L'encéphale est responsable de la plupart des fonctions telles que : le mouvement, la pensée, la planification, la perception, la parole, la mémoire, etc., sa partie inférieure est rattachée à la moelle épinière au niveau du tronc cérébral [Nieuwenhuys, R. et al., 2007].

L'encéphale est composé d'une structure très complexe appelée le cerveau qui est composé d'environ 100 milliards de cellules nerveuses (neurones), il est le chef d'orchestre de l'organisme du corps humain, en agissant sur les muscles ou les glandes, et constitue le siège des fonctions cognitives. Cependant, les comportements plus sophistiqués nécessitent que le cerveau intègre les informations provenant des systèmes sensoriels et renvoie une réponse appropriée. En

conséquence, le cerveau joue un rôle central dans la création des réponses à l'environnement et la régulation hormonale [Bear, M. et al., 2020].

D'autre part, l'encéphale contient un cervelet, situé à l'arrière du crâne et en dessous du lobe occipital du cerveau, il est composé de deux hémisphères et un vermis. Cet organe traite, d'une façon continue, les informations reçues du cerveau et de la moëlle épinière pour effectuer l'organisation motrice et temporo-spatiale [Schmahmann J. D., 2004], par exemple la coordination, la régulation et la synchronisation des gestes musculaires. Le cervelet implique indirectement dans le processus cognitif : le langage, l'attention, la mémorisation, etc.

L'encéphale comprend également le tronc cérébral, situé dans la fosse crânienne postérieure à la base du cerveau, qui est composé du mésencéphale, la protubérance annulaire et le bulbe rachidien. Il est responsable des fonctions de survie, telles que : la régulation du rythme cardiaque et de respiration, la tension artérielle, la conscience, etc.

La voie de communication entre l'encéphale et le reste de l'organisme consiste à la moelle épinière (spinale), qui est une structure cylindrique aplatie et fragile qui s'étend à partir du bulbe rachidien situé au niveau du tronc cérébral vers le cône médullaire au niveau de la région lombaire de la colonne vertébrale qui la protège, sa fonction principale et de transporter l'influx nerveux entre le cerveau et le corps [Kandel E. R. et al., 2000]. Certaines informations liées à la perception sont traitées par la moëlle épinière et ne sont pas transmises au cerveau, dans ce cas on parle de la fonction de réflexe.

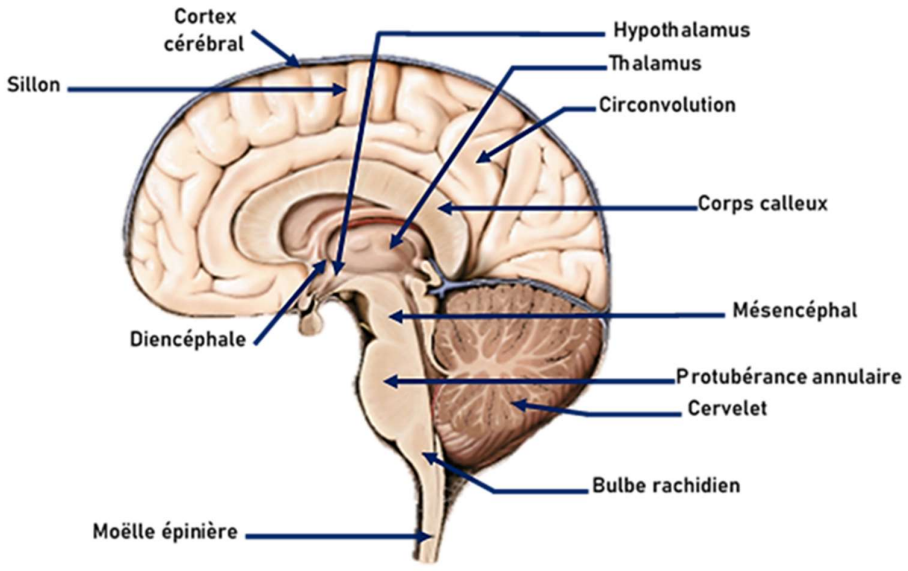
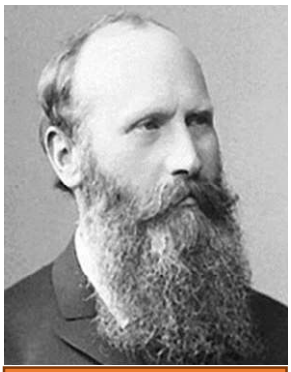


Figure I.3. Coupe latérale de l'encéphale montrant les différents éléments qui composent sa structure profonde.

I.3.2.1. Structure du neurone et le message nerveux


Heinrich Wilhelm Waldeyer
1836-1921

Le mot « neurone » est introduit la première fois en 1881 par l'anatomiste allemand *Heinrich Wilhelm Waldeyer* (1836 - 1921) pour renommer l'unité fondamentale du système nerveux : cellule nerveuse, ensuite *Santiago Ramon y Cajal* formula la description histologique du système nerveux à la fin des années 1880 [Pestronk, A., 1997].

Un neurone est l'unité fonctionnelle de base du système nerveux, il est composé d'un corps cellulaire (le soma), des dendrites et d'un axone, qui se termine par une zone synaptique (figure I.4). Le neurone transmet et module un signal électrique (influx nerveux) qui correspond à une dépolarisation de la membrane axonale : le potentiel d'action. Ainsi, l'axone joue le rôle d'un émetteur de l'influx nerveux jusqu'à la synapse, la zone de communication entre les neurones. Les molécules émises au niveau des synapses sous l'influence des potentiels d'action sont appelés neurotransmetteurs qui sont excitateurs ou inhibiteurs et déterminent ainsi la réponse obtenue [Kandel, E. R., 2013].

Les neurones sont organisés en réseaux très précis dont les connexions déterminent leurs rôles. Traditionnellement, une distinction est faite entre le SNC et le SNP ; il est courant de parler de neurones efférents, qui transmettent des informations de commande du SNC au SNP, et de neurones afférents, qui transmettent des informations sensorielles du SNP au SNC.

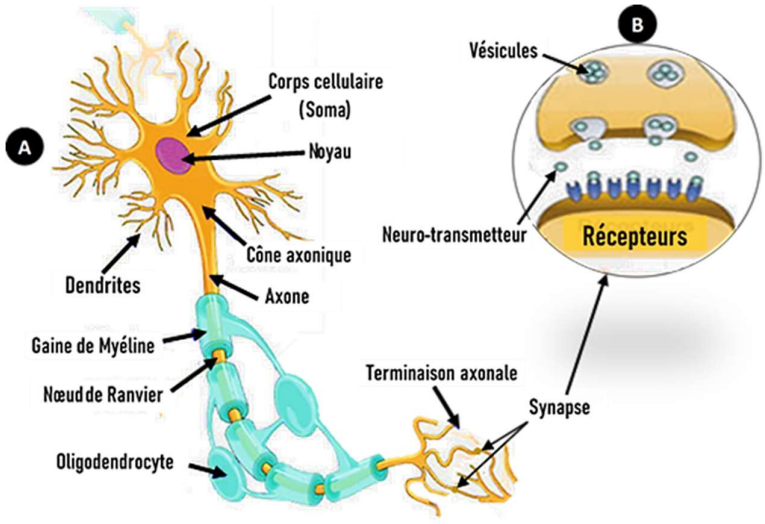


Figure I.4. (A) Schéma légendé de la structure du neurone. (B) Illustration de la zone synaptique et de la transmission électrochimique.

Lorsque plusieurs neurones présynaptiques transmettent leur message simultanément au même neurone postsynaptique par des synapses excitatrices (Potentiel PPSE) et inhibitrices (Potentiel PPSI), le potentiel postsynaptique résultant a une valeur d'amplitude égale à la somme algébrique spatiale et/ou temporelle des PPS unitaires (figure I.5). On parle de la sommation spatiale lorsque la somme des PPS au niveau de la zone gâchette (cône axonique), atteint une valeur de seuil qui peut déclencher un potentiel d'action le long de l'axone [Gescheider G. A. et al., 2010]. D'autre part, la sommation temporelle consiste à la somme des PPS successifs provenant de la même connexion synaptique d'un neurone, si cette somme atteint la valeur de seuil, alors un potentiels d'action sera déclenché le long de l'axone.

En conclusion, le neurone effectue une somme algébrique de tous les potentiels membranaires propagés par les entrées présynaptiques, en soustrayant la somme des hyperpolarisations de la somme des dépolarisations au niveau de la zone gâchette du neurone. Tous les potentiels locaux, dendritiques et somatiques, sont additionnés par sommations spatiale et temporelle pour produire un potentiel synaptique global.

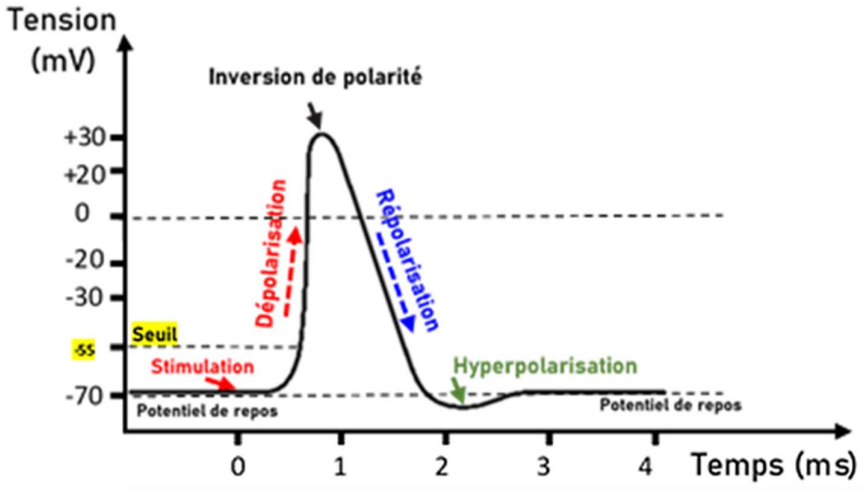


Figure I.5. Représentation temporelle du potentiel d'action neuronal.

I.3.2.2. Anatomie fonctionnelle du cerveau

Dans la plupart des définitions, le terme 'cerveau' désigne uniquement une partie de l'encéphale. Sur le plan anatomique fonctionnelle, le cerveau est composé de nombreuses parties, chacune ayant une fonction spécifique et importante. Il contrôle et coordonne la motricité, la sensibilité, la production hormonale et l'homéostasie des fonctions internes, telles que la température, la pression artérielle du corps et le rythme cardiaque, etc. Le cerveau est subdivisé en deux hémisphère cérébraux droit et gauche reliés entre eux par des fibres nerveuses

qui forment le corps calleux. Chaque hémisphère traite les informations sensorielles et émet les commandes motrices d'une façon controlatérale ; c'est-à-dire de/vers la partie du corps opposée. Attendu que la plupart des dispositifs d'enregistrement non-invasif de l'activité électrique du cerveau, récupèrent seulement les signaux apparaissant sur la surface du cortex cérébral, alors l'accent sera mis sur cet organe qui est composé de six couches de neurones et de nombreuses aires fonctionnelles (voir la figure I.6). Étant donné que le cortex cérébral est impliqué dans la plupart des tâches motrices, sensitives et cognitives et qu'il est relativement accessible expérimentalement, il s'agit de la zone cérébrale prioritaire dans la plupart des recherches sur les BCIs [Miller L. & Hatsopoulos N., 2012].

Comme l'illustre la figure I.7, à la surface des hémisphères cérébraux se trouve un tissu appelé cortex cérébral, dont la majorité de sa surface est pliée en circonvolutions (chacune est appelée gyrus) : la surface de la partie exposée du cerveau est 600 cm^2, tandis que la surface totale du cortex est d'environ 1800 cm^2 [Purves, D. et al., 2019]. Ainsi, l'un des avantages du grand nombre de ces circonvolutions est d'augmenter la surface totale du cortex et le nombre de neurones impliqués dans le traitement simultané de l'information. Le cortex cérébral est également appelé la substance grise en raison de la forte concentration de corps cellulaires neuronaux qu'il contient. La fonction de la substance grise est de recevoir les messages nerveux sensoriels, analyser et traiter ces informations complexes et finalement élaborer des réponses aux stimuli. Du point de vue fonctionnel, la substance grise (le cortex cérébral) est impliquée dans plusieurs fonctions telles que la motricité, l'organisation, la planification, le langage, les sensations, etc.

Au-dessous du cortex cérébral, on trouve la substance blanche qui est un tissu nerveux d'aspect blanchâtre composé principalement d'axones myélinisés très fins à longue portée. La substance blanche assure la conduction rapide de l'influx nerveux vers la substance grise du cortex ou vers d'autres zones du système nerveux.

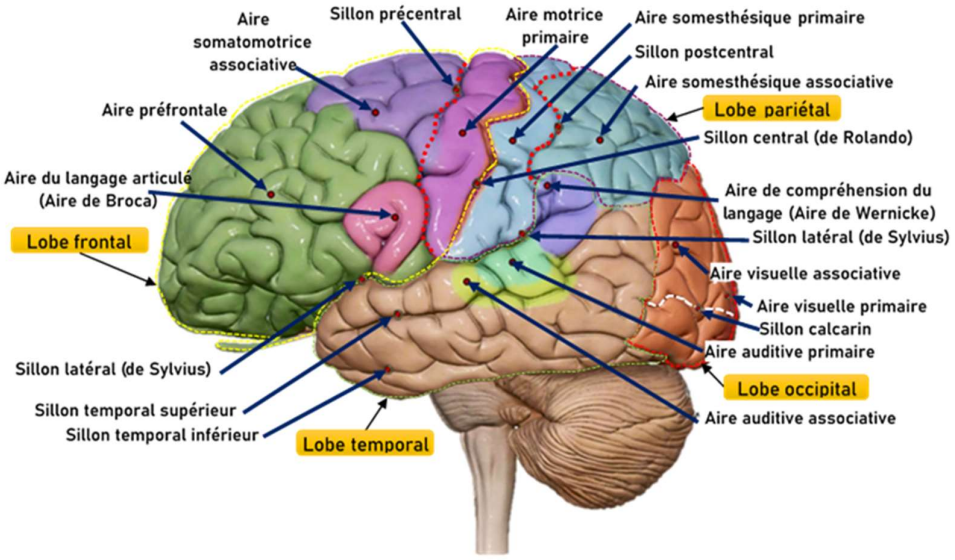


Figure I.6. Vue générale de la surface du cortex, des principaux lobes, les sillons et les aires fonctionnelles.

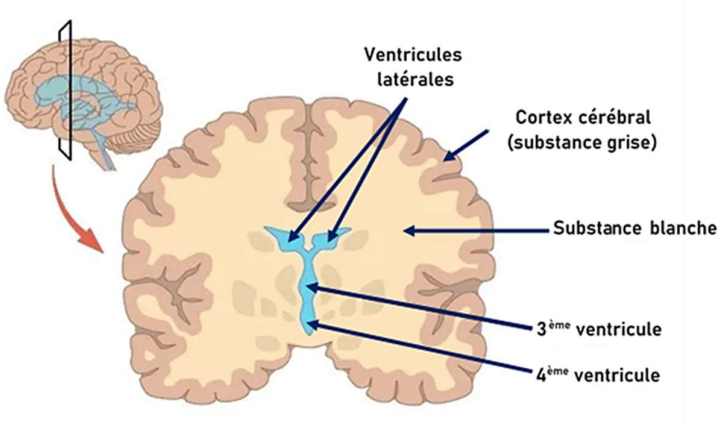


Figure I.7. Cortex cérébral avec matière grise et blanche et les ventricules.

À l'exception du *lobe insulaire* (aussi appelé *insula*) situé à l'intérieur (voir la figure I.8), dont le rôle est impliqué dans les émotions et le dégoût, chaque hémisphère du cortex cérébral est composé du *lobe frontal*, lieu d'initiation et de coordination des mouvements volontaires, de comportement, des tâches cognitives supérieures comme la pensée, la résolution des problèmes, les fonctions du langage, etc. ; du *lobe pariétal*, siège des sensations, d'attention, de l'orientation spatiale, etc. ; du *lobe occipital*, permettant l'interprétation des informations visuelles, la création des souvenirs visuels et la reconnaissance des formes et des couleurs ; du *lobe temporal*, centre de l'audition, des émotions et de la mémoire à court terme. Ces lobes sont séparés par des sillons : sillon central de Rolando, sillon latéral de Sylvius, sillon antérieur occipital, etc. (voir la figure I.6).

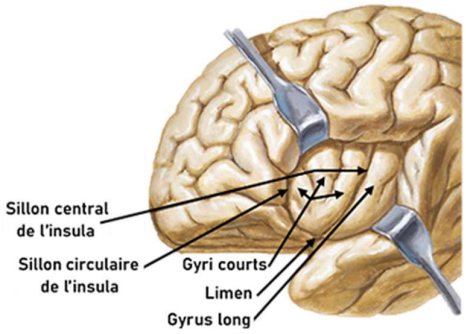


Figure I.8. Vue latérale du lobe insulaire.

L'organisation fonctionnelle du cerveau n'est pas tout à fait stricte, parce que différentes aires corticales peuvent impliquer simultanément dans un processus cognitif assez complexe (par exemple : la mémoire, le langage, les émotions, etc.), dans ce cas les zones du cerveau interagissent en coopération en fonctions des tâches à réaliser. D'autre part les systèmes sous-cortical et limbique comme illustré par la figure I.9 composés des ganglions de la base (globus pallidus, putamen, noyau caudé, noyau sous-thalamique, le noyau accumbens), du thalamus (impliqué dans la motricité et les émotions), de l'hypothalamus (le moteur endocrinien), de l'amygdale (impliqué dans la peur et la détection du plaisir), de l'hippocampe (impliqué dans les émotions), de l'épiphyse (contrôle la régulation du cycle nycthémérale: horloge biologique interne), du tronc cérébral (impliqué dans la régulation du rythme cardiaque et la respiration) et du cervelet (contrôle la coordination motrice et la posture) sont en interaction constante avec le cortex cérébral.

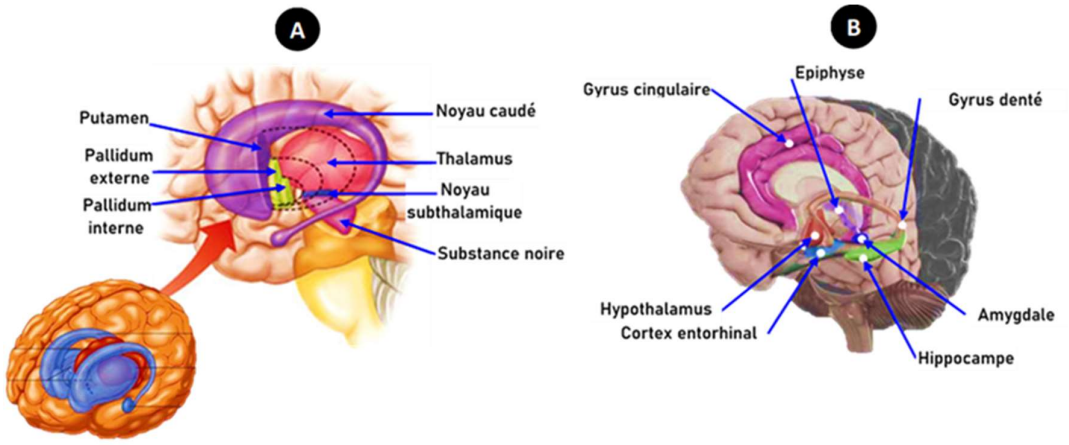


Figure I.9. (A) Noyaux gris centraux. (B) physiologie du système sous cortical limbique.

Avec l'évolution des techniques de l'imagerie cérébrale, les chercheurs neuroanatomistes ont pu dresser une cartographie fonctionnelle du cortex cérébral où ils localisent trois catégories des aires fonctionnelles en déterminant leurs fonctions.

I.3.2.3. Les aires motrices

L'activation de ces aires provoque des mouvements volontaires des muscles spécifiques du corps humain. Ce cortex implique dans la planification, l'organisation et l'exécution de la motricité volontaire. Il combine les trois aires suivantes :

- **L'aire motrice primaire (M1)**

Comme indiqué par la figure I.6, situé dans le lobe frontal, est une région cérébrale d'une grande importance dans le domaine des BCIs en raison de sa relation étroite avec le contrôle des mouvements.

L'aire motrice primaire est organisée somatotopiquement, c'est-à-dire que des régions spécifiques de M1 sont dédiées principalement au contrôle de zones corporelles spécifiques. Cette organisation illustrée par la figure I.10 se reflète dans l'homoncule de Penfield [Penfield W. & Boldrey E., 1937]. Les représentations des jambes et des pieds se trouvent dans la paroi médiale du cerveau ; les représentations du tronc, du haut du bras et de la main se produisent progressivement plus latéralement dans l'hémisphère ; et le visage est le plus latéral.

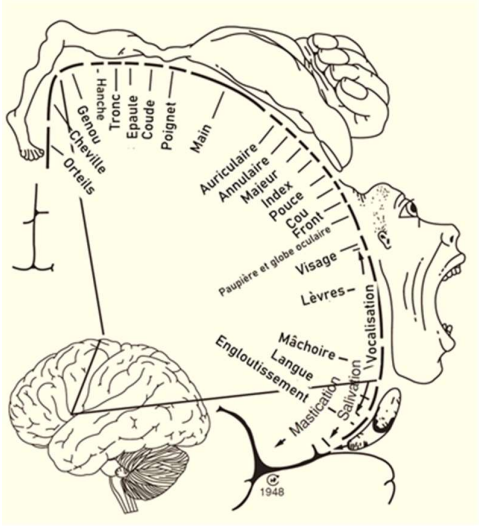


Figure I.10. Homoncule de Penfield

La fonction de cette aire est de commander directement l'exécution du mouvement volontaire.

- **L'aire prémotrice (PM)**

La PM est également située dans le lobe frontal, elle se trouve dans la zone antérieure de l'aire motrice primaire M1 (voir la figure I.6), elle est appelée aussi l'aire somatomotrice

associative. Cette aire interagit avec l'aire motrice primaire, les aires somesthésiques associatives, les ganglions de la base et le thalamus dans le l'objectif de traiter les activités motrices apprises et répétitives de nature complexe [Amunts, K. et al., 1997], par exemple : l'écriture. Cette aire a une grande influence sur la qualité des tâches motrices qui demandent beaucoup de dextérité.

- **L'aire de Broca**

Parfois appelée l'aire motrice du langage, qui est découverte en 1861 par le médecin français Paul Broca. D'après la figure I.6, l'aire de Broca est située dans la partie ventrale du 3ème gyrus frontal et au-dessus du sillon central de l'hémisphère dominant, c'est-à-dire : l'hémisphère gauche des droitiers ou l'hémisphère droit des gauchers. Elle établit une communication avec l'aire prémotrice pour coordonner les mouvements du larynx, du pharynx et de la bouche afin de générer la synthèse des mots [Dronkers NF, 1996]. En même temps des influx nerveux sont émis au système de respiration afin d'adapter la quantité d'air nécessaires pour vibrer les cordes vocales. Sa fonction principale est de permettre de traduire les pensées en langage parlé et écrit et vice-versa.

I.3.2.4. Les aires sensorielles somesthésiques

Situées dans le lobe pariétal, immédiatement derrière le sillon de Rolando, leur fonction est de traiter les informations sensorielles afférentes : le toucher, la pression artérielle, la douleur, etc. Elle est constituée des aires suivantes :

- **L'aire somesthésique primaire**

Cette aire se trouve derrière le sillon de Rolando dans la circonvolution postcentral de chaque hémisphère cérébral, elle s'étend jusqu'au sillon latéral. Cette aire reçoit des types particuliers des sensations en provenance des récepteurs somatiques du toucher, de la douleur, de la température et des articulations. Chaque point de cette aire reconnait une information sensorielle issue d'une partie précise du corps. Son rôle principal est de localiser, avec une grande précision et en coopération avec le thalamus, la source de la sensation [Kaas, J. H., 2004].

- **L'aire somesthésique associative**

Située immédiatement derrière l'aire somesthésique primaire et délimité par le lobe occipital (comme illustré par la figure I.6). Elle a pour rôle d'interpréter l'influx nerveux reçu des couches sous-corticales comme le thalamus, où elle détermine avec précision la forme, la

nature et l'orientation des objets touchant une partie du corps sans avoir à les voir, ce qui suggère qu'elle puisse affiner en quelques sorte l'information reçue par l'aire somesthésique primaire [Hsiao, S. S. et al., 1993]. De plus, l'aire somesthésique associative implique également dans la mémorisation des expériences sensorielles passées et les comparer aux nouvelles.

I.3.2.5. Les aires sensorielles visuelles

Elles occupent le lobe occipital du cortex cérébral et délimitées par le lobe temporal et pariétal, leur rôle est de traiter les informations provenant des voies visuelles telles que : la couleur, la texture, la forme, etc. [Felleman D.J. & Van Essen D.C., 1991] Comme illustré par la figure I.11, Les deux nerfs optiques se croisent au niveau du *chiasma*, ce qui signifie que chaque hémi-champ visuel est analysé par l'hémisphère cérébral opposé.

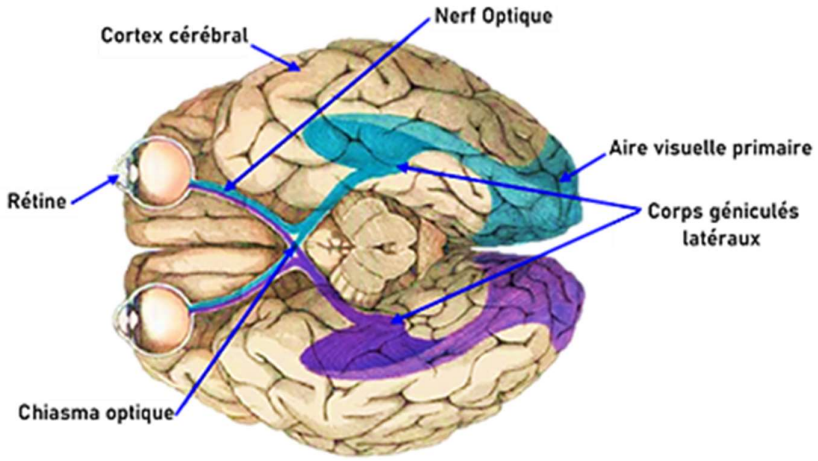


Figure I.11. Décussation des voies nerveuses optiques.

- **L'aire visuelle primaire (V1)**

Elle se trouve dans la zone médiale du lobe occipital. L'aire V1 détecte les points sombres et lumineux ainsi que le contour de la scène visuelle. Les neurones issus des corps géniculés latéraux, transmettant les informations visuelles, se projettent dans cette aire.

- **L'aire visuelle associative**

Au niveau du lobe occipital, l'aire visuelle associative a la fonction d'interpréter l'information visuelle reçue de l'aire visuelle primaire et du thalamus en la comparant aux informations mémorisées par les expériences précédentes.

I.3.2.6. Les aires sensorielles auditives

Ces aires sont situées dans le lobe temporal au-dessous du sillon latéral de Sylvius [Hackett T.A., 2011].

- **L'aire auditive primaire**

L'aire auditive primaire se trouve dans la partie supérieure du lobe temporal délimitée par le sillon latéral, elle reçoit les informations principales de la sonorité telles que : l'amplitude du son, le rythme sonore, la hauteur du son, etc.

- **L'aire auditive associative**

Située immédiatement sous l'aire primaire, elle est impliquée dans la fonction de distinction les natures sonores tels que : la musique, la parole, le bruit, etc. En collaboration avec l'aire de Wernicke (l'aire de compréhension du langage), l'aire auditive associative permet également d'interpréter la signification linguistique, en d'autre terme traduire le sens des mots en pensées.

Cependant, il existe d'autres aires fonctionnelles qui ne sont pas présentées en détail dans ce contexte, car elles ne sont pas très courantes dans le domaine du développement des BCIs, il s'agit de :

- L'aire gnosique : qui intègre les informations issues de plusieurs aires associatives pour former une unité de pensée commune. Ensuite le cerveau exploite ces réponses pour déclencher les réactions appropriées.
- L'aire gustative primaire qui reçoit l'influx nerveux issue de la gustation.
- L'aire olfactive primaire qui reçoit l'influx nerveux issue de l'olfaction.
- L'aire oculomotrice qui sert en tant que régulateur des mouvements des yeux.

I.4. Dynamique neuronale et Électrogenèse corticale

Au cours des dernières décennies, l'électroencéphalographie a devenu un outil d'aide précieuse pour l'exploration et l'analyse de l'activité cérébrale liée au processus cognitif dans lequel l'étude des potentiels évoqués et l'activité cérébrale spontanée devient de plus en plus d'une importance majeure, en particulier dans le domaine de la recherche des BCIs [Treder M. S., 2011].

Toutes les méthodes mentionnées dans la littérature : l'EEG [Abiri R. et al., 2020] et l'ECoG [Reza A., 2015] sont basées sur l'exploration des potentiels de champs qui se propagent partout sur la surface crânienne. Sachant qu'un conteneur d'oscillations de rythmes différents représente une synchronisation simultanée de plusieurs polarisation et dépolarisation neuronales produisant ainsi une activité macroscopique à une large échelle [Gray C. M., 1994]. Cette forme de l'activité opère suivant un processus coopératif neuronale.

Malgré la contribution de l'activité d'un seul neurone (au niveau microscopique) à un processus cognitifs global, les interprétations ne sont effectuées qu'au niveau mésoscopique (regroupement local des neurones) ou macroscopique (réseau de neurones global), car l'obtention d'une information microscopique intra neuronale n'a jamais permis d'avoir une interprétation pertinente sur le plan de cognition [Gazzaniga M. S. 2004].

Plusieurs études de recherches [Laufs H. et al., 2003 ; He B. J., 2013] ont démontré que l'analyse des potentiels de champs macroscopique peuvent donner un aperçu de la nature des processus cérébraux.

En plus d'une structure fonctionnelle que l'on a décrit dans les sections précédentes, le cortex cérébral possède également une structure anatomique laminaire (6 couches cellulaires de **I** à **VI**), qui est en relation directe avec la diffusion des activités électriques normales ou même pathologiques.

I.4.1. Production de l'EEG de surface

Les générateurs de l'électrogenèse ou des potentiels de champ sont généralement les neurones pyramidaux situés dans les couches **III** (couche pyramidale externe), **IV** (couche granulaire interne) et **V** (couche pyramidale interne) du cortex. Ces neurones sont orientés orthogonalement à la surface corticale et leur axone s'étend en profondeur vers la couche **VI** (couche multiforme) et la substance blanche [Tjia M., 2017]. Les neurones pyramidaux possèdent des arbres dendritiques très développé et peuvent établir un grand nombre de connexions synaptiques, chaque cellule pyramidale pouvant recevoir de 10^4 à 10^5 synapses. Leurs dendrites apicales sont très longues et se dirigent perpendiculairement à la première couche corticale (appelée la couche moléculaire), cependant leur arborisation dendritique basilaire part de la base de leurs corps cellulaires.

Un neurone pyramidal excité devient négatif à la surface de ses dendrites corticales, de sorte que l'électrode EEG voit les charges positives s'échapper, ce qui lui donne le comportement d'un puits des charges négatives, et par conséquent un potentiel négatif est mesuré au point de contact de l'électrode. D'autre part, les dendrites apicales des neurones inhibés deviennent plus positives et il y aura donc création d'un potentiel positif (le principe est illustré par la figure I.12). À l'exception des neurones pyramidaux, la plupart des cellules corticales ne sont pas parallèles, et la différence de potentiel enregistrée à la surface sera donc presque nulle. Une autre condition pour que l'enregistrement cortical soit réussi est que toutes les cellules aient une excitation synchrone, c'est-à-dire que l'origine de l'excitation doit provenir de la même couche

ou du même niveau (exemple couche **IV**) [Chagnac-Amitai Y. & Connors B. W., 1989]. En effet, les neurones pyramidaux réunissant les conditions de parallélisme et de synchronisme sont l'origine de l'électrogenèse de surface qui est une très forte activité simultanée de milliers de cellules pyramidales.

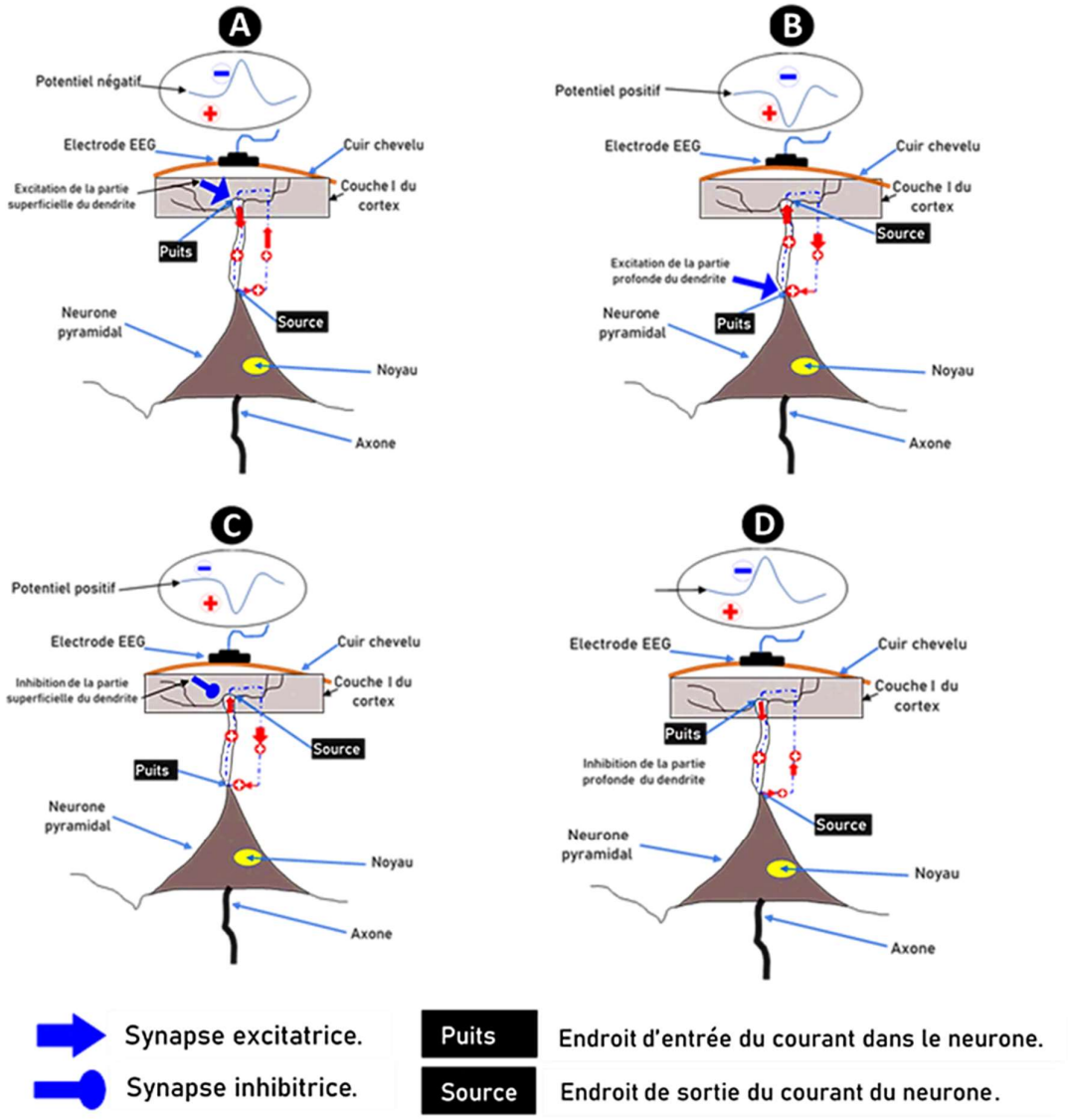


Figure I.12. Dynamique électrophysiologique des neurones pyramidaux corticaux : générateurs des potentiels de champ et leur impact sur l'électrogenèse de surface.

La dépolarisation et l'hyperpolarisation des neurones pyramidaux sous-jacents sous l'effet des afférentes excitatrices ou inhibitrices produisent des courants ioniques se bouclant dans l'espace cellulaire et extracellulaire comme illustré par la figure I.12. Ces courants extracellulaires entrainent une variation des champs de potentiels dont leur sommation algébrique produit l'activité EEG mesurée par une électrode spécifique posée sur le cuir chevelu.

I.5. Motifs d'intérêt pour les interfaces cerveau-ordinateur

Chaque interface cerveau-ordinateur est conçue pour opérer par rapport à un ou plusieurs motifs cérébraux. Dans cette section, nous examinerons uniquement les signaux cérébraux les plus courants dans la conception des BCIs en raison de leur mise en œuvre facile et simplifiée. Cette étude repose sur les bases physiologiques de ces motifs, l'apprentissage initiale requise, et la vitesse dans laquelle les informations sont détectées et transmises à l'application BCI. Ces composantes présentent un intérêt particulier pour les BCIs et sont divisées en plusieurs catégories.

La figure I.13 représente les motifs cérébraux les plus courants dans les applications des BCIs [Wolpaw J. R. et al., 2002 ; Gruss L. F. et al., 2012].

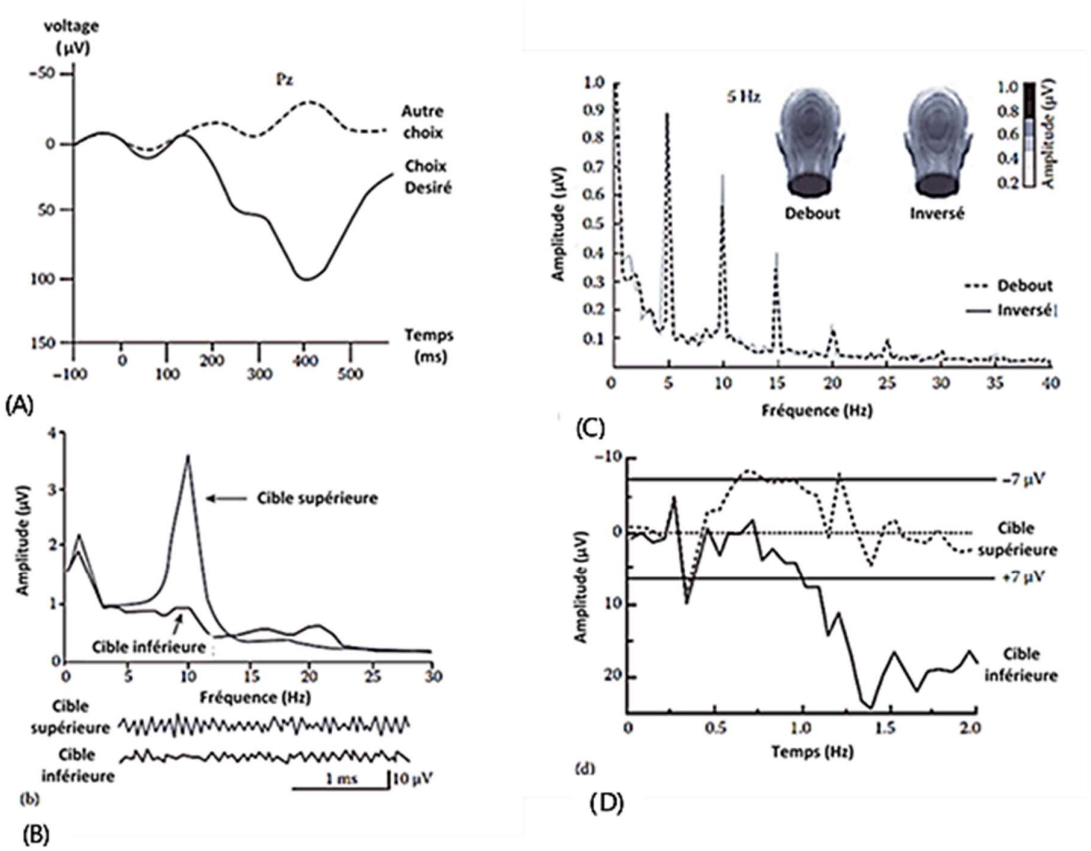


Figure I.13. (A) motifs des signaux cérébraux P300 pour le fonctionnement BCI, (B) Motif de rythme sensorimoteur, (C) Motif de potentiel évoqué visuel en état d'équilibre, (D) Motif de potentiel cortical lent.

I.5.1. Les composantes d'intérêt

I.5.1.1. Activité cérébrale spontanée

En général, lorsque nous parlons du terme "EEG", nous nous référons à l'EEG spontanée, pour cette raison l'activité spontanée est également appelée l'activité oscillatoire de l'EEG. Même en l'absence de toute entrée, le cerveau génère des modèles complexes d'activité spontanée. En fait, elle est causée par une activité synchrone d'un réseau complexe de neurones chez une personne éveillée et détendue dans un état d'inactivité et de repos [He B. et al., 2020].

Dans une activité normale on observe un tracé sous forme des ondes rythmiques (oscillations) lentes qui sont observables même en l'absence de toute forme de stimulus externe. L'activité spontanée inclut les motifs cérébraux pendant que l'utilisateur dort, engage dans une tâche cognitive ou une imagination motrice, souffre de troubles cérébraux, etc.

Les fluctuations de cette activité influent sur la façon dont le cerveau réagit au monde extérieur, c'est-à-dire lorsqu'on présente au sujet une même entrée sensorielle on verra une gamme de réponse différente pour chaque essai, cela vaut dire que la réponse du cerveau à une entrée sensorielle dépend aussi de l'état interne du cerveau au moment où le stimulus est traité.

Toutefois, Les BCIs basés sur l'activité spontanée sont plus difficiles à réaliser car les motifs ont un rapport SNR très faible et ne sont pas similaires d'un sujet à l'autre [Pfurtscheller G. & Neuper C., 2001]. Il existe deux types de motifs spontanés :

- Le rythme mu (Rolandique) dans la gamme 10 – 12 Hz.
- Le rythme Beta central dans la gamme 14 – 18 Hz.

I.5.1.2. Potentiels évoqués - PE

Un potentiel évoqué PE est un motif cérébral spécifique détecté suite à la présentation d'un évènement sensoriel (auditif, visuel, somesthésiques, etc.), cognitif ou moteur, il est peut-être discret ou continu rythmique. Les potentiels évoqués sont obtenus à partir des paradigmes où le sujet est confortablement installé en lui demandant de suivre des instructions spécifiques lors de l'enregistrement de son activité cérébrale et la durée du protocole varie selon le type du paradigme et le nombre des échantillons à extraire. Les potentiels évoqués consistent en une activité très dense et synchrone d'un grand nombre de neurones pyramidaux. L'analyse des PE est importante pour la détection des dysfonctionnements sensoriels liés à certaines pathologies des voies sensorielles [Sugerman R. A., 2006].

Toutefois, la plupart des potentiels évoqués possèdent une amplitude inférieure à 5 µV [Yamada, T. et al., 2004] et sont partiellement ou totalement obscurcis par l'activité EEG spontanée et par d'autres signaux non pertinents (bruits, artefacts EMG et EOG, etc.), en d'autre terme, le rapport signal/bruit SNR (*Signal-to-Noise ratio*) du signal lié aux PE étant très faible. En conséquence, la moyenne d'un grand nombre d'échantillons est une technique courante pour annuler la contribution des signaux de bruit et des artefacts liés au sujet et donc améliorer le SNR. En plus, les PE sont verrouillés en phase avec les stimuli et peuvent être déterminés avec un haut degré de précision. Cette propriété permet de les extraire facilement en répétant la stimulation un grand nombre de fois, puis en effectuant la technique de la moyenne [Yamada T. et al., 2004]. D'autre part, la classification d'un seul échantillon d'entrée nécessite une formation à priori avec un ensemble d'apprentissage. Dans les applications BCI, on se limite généralement à un nombre limité des stimulations afin d'obtenir une communication rapide et fiable.

Selon la méthode de stimulation, il existe deux catégories des PE : les potentiels liés aux évènements PLE et les potentiels évoqués à l'état d'équilibre (*Steady-State Evoked Potentials, SSEP*).

Dans la pratique, peu de recherches qui s'intéressent des applications BCI basée sur ce type des potentiels évoqués [Muller-Putz G. R. et al., 2006].

I.5.1.3. Potentiels liés aux évènements - ERP

Les potentiels liés aux évènements ERP (*Event Related Potentials*) sont des potentiels évoqués en réponse à un événement (stimulus) particulier d'un intérêt expérimental (auditif, visuel, somesthésique, etc.), discret (discontinu) et isolé (non rythmique). L'isolement est obtenu en séparant plusieurs stimuli avec de longs intervalles de temps, cela permet l'estimation d'une référence de base indépendante du stimulus [Norcia A. M. et al., 2015]. La fréquence des stimuli des ERP est généralement inférieure à 2 Hz.

Les potentiels liés aux événements ERP sont synchronisés avec le stimulus externe et possèdent une faible fréquence et une grande amplitude [Luck S. J., 2014]. Comme pour les potentiels évoqués, l'extraction de courts segments ERP est réalisée en faisant la moyenne des données de nombreux essais afin de réduire l'influence des signaux de fond non pertinents.

Les évènements ERP peuvent être classés comme suit :

- **Synchronisation/(Dé) synchronisation liée à un événement**

Un type particulier d'ERP est caractérisé par l'apparition d'une désynchronisation liée aux événements ERD (*Event-Related Desynchronization*) et d'une synchronisation liée aux événements ERS (*Event-Related Synchronization*). Une diminution de la synchronisation des neurones entraîne une diminution de la puissance dans des bandes de fréquences spécifiques. Ce phénomène est défini comme une ERD et peut être identifié par une diminution de l'amplitude du signal. L'ERS se caractérise par une augmentation de la puissance dans des bandes de fréquence spécifiques, générée par une augmentation de la synchronisation des neurones et/ou de l'amplitude du signal.

En général, auditive, visuelle et tactile sont les trois principales sources de stimulation de l'ERP [Cecotti H. & Ries A. J., 2017].

- **Les potentiels évoqués auditifs – PEA :**

Les PEA sont une sous-classe spécifique des ERP dans laquelle les réponses aux stimuli acoustiques sont enregistrées. Ceux sont des réponses cérébrales cadencées dans le temps à un évènement auditif tel que : le son parole, les différentes tonalités, etc. Les PEA sont générés dans la cochlée et sont transmis par le nerf cochléaire, puis traversent le tronc cérébral, le corps géniculé médian et atteignent finalement le cortex cérébral où ils peuvent être détectés par EEG.

Initialement, les signaux PEA sont utilisés dans la détection précise et rapide de la perte unilatérale [Chiappa K. H., 1997] et du seuil d'audition à un stade précoce. Cependant, les réponses PEA ont récemment été introduites dans la technologie des interfaces cerveau-ordinateur (BCI) [Islam M. N. et al., 2021].

Malgré la grande utilité des BCIs à base de PEA pour les malvoyants ou pour les patients en syndrome d'enfermement LIS (*Locked-In Syndrome*) ; certains travaux montrent que les interfaces cerveau-ordinateur basées sur les potentiels évoqués auditifs (PEA) ne sont pas aussi robustes que celles basées sur les potentiels évoqués visuels (PEV) [Brunner C. et al., 2011], ce qui semble qu'il serait plus difficile de récupérer des sons que des images sur la base des potentiels liés aux événements.

Il existe un type spécifique des PEA appelé la présentation auditive en série rapide RSAP (*Rapid Serial Auditory Presentation*) qui fait référence à des expériences de présentation en série

rapide de stimuli acoustiques à la fréquence de 10Hz, la tâche de l'utilisateur étant de reconnaître le son cible parmi les sons perturbateurs [Franco A., 2015].

- **Les potentiels évoqués visuels – PEV :**

Les PEV sont une sorte des potentiels liés aux évènements ERP qui se produit en réponse à un stimulus visuel. Les motifs PEV sont cachés dans l'EEG spontané normal, en conséquence, pour les séparer des lectures d'EEG de fond ; des techniques de stimulation répétitive et de la moyenne temporelle des signaux verrouillés dans le temps sont généralement utilisées.

À l'origine, les PEV sont utilisés dans les tâches cliniques pour analyser le fonctionnement maculaire/périmaculaire et les voies visuelles ; lorsqu'il y a une diminution notable de l'acuité visuelle, un accident vasculaire cérébral (AVC) et même lors d'un diagnostic de sclérose en plaques, etc. [York D. H. et al., 1981].

Après la présentation d'un stimulus visuel à l'aide d'une diode électroluminescente LED (*Light Emitting Diode*) scintillante ; on obtiendra des PEV de latences différentes directement au-dessus de l'aire visuelle primaire (voir la figure I.13), les plus marqués sont les potentiels N2 après 90 ms et P2 après 120 ms.

Les principales composantes des PEV sont détectées au niveau du lobe occipital, mais celles qui ont des amplitudes les plus élevées sont recueillies au niveau du sillon calcarin.

Une autre technique de référence actuellement utilisée pour collecter les VEP est le damier alternant, les PEV d'intérêt sont détectés au niveau de l'aire visuelle primaire et parfois dans le cortex pariétal [Clark V. P., & Hillyard, S. A., 1996]. Lorsque la luminosité est maintenue constante, mais que le contraste est soudainement modifié (Exemple : un damier en noir et blanc alternant) chez un sujet normal, on observe 3 composantes [Vialatte F. B. et al., 2010] :

- La composante N75 qui apparait sous forme d'un pic négatif avec une latence de 75 ms.
- La composante P100 qui correspond à un pic positif avec 100 ms de latence est de forme inchangée chez tous les utilisateurs et trouve son origine dans l'aire visuelle primaire V1.
- La composante N135 qui apparait sous forme d'un pic négatif avec une latence de 135 ms.

Il convient de noter que l'extraction de ces composantes susmentionnées à partir d'une seule tentative de stimulation est extrêmement difficile, de sorte que leur utilisation comme

marqueurs de BCI est peu probable [Zhu D., et al. 2010]. En revanche, lorsque les stimulations sont répétitives, un processus de fonctionnement permanent se déclenche au niveau de la chaine de perception visuelle qui entrainera l'émergence de potentiels évoqués visuels en état d'équilibre SSVEP (*Steady-State Visual Evoked Potentials*) [Regan D., 1966].

En outre, la présentation visuelle rapide en série RSVP (*Rapid Serial Visual Presentation*) [Lees, S. et al., 2018] est considéré comme un type spécifique des PEV. Le protocole expérimental de cette technique consiste à placer l'utilisateur devant un écran sur lequel s'affiche une série d'images les unes après les autres. La fréquence de passage des éléments est souvent 10 Hz (la durée d'affichage de chaque élément est 0.1 seconde). Ainsi, le sujet devra être capable de distinguer l'élément cible parmi un ensemble d'éléments distrayants (perturbateurs) ; étant donné qu'il a déjà appris à connaître l'élément cible avant de commencer le paradigme RSVP. En général, les distracteurs peuvent être soit un changement de couleurs, de formes, des lettres entre les chiffres, etc.

- **Les potentiels évoqués somesthésiques – PES**

Les PES sont une sous-catégorie des potentiels liés aux évènements ERP. Ils sont déclenchés par des stimuli électriques, mécaniques, le toucher, les vibrations, la douleur, etc. pour exciter les nerfs périphériques. Initialement les PES sont utilisés dans le diagnostic clinique pour faire le test des voies sensitives chez les patients en mesurant la sensibilité tactile ce qui permet de localiser des anomalies de sens telle que la paresthésie [Buonocore M., & Demartini L., 2016], etc. Les signaux PES comprennent une série de déviations d'amplitude qui peuvent être provoquées par pratiquement tous les stimuli sensoriels.

Les PES de faibles amplitudes qui apparaissent dans des périodes de latence courtes (les composantes P9/N9, P11/N11, P13/P14, N18, P20/N20, P22) sont rationnellement distinctes de l'état de conscience, tandis que celles de fortes amplitudes qui apparaissent dans des périodes de latence moyennes ou longues (les composantes P26, N30, P40, et N60) sont liées aux fonctions cognitives et au niveau de vigilance de l'utilisateur [Luck S. J., 2014].

I.5.1.4. Potentiel P300 endogène

Il a été décrit pour la première fois par Sutton et al. (1965) [Sutton S. et al., 1965], le potentiel P300 (parfois appelé P3) est une fluctuation positive qui se produit principalement dans l'activité cérébrale au niveau du centre du lobe pariétal lorsque l'utilisateur en état de vigilance reçoit un stimulus sensoriel inattendu ; en réponse, les PES standards sont suivis d'une série de potentiels évoqués très spécifiques déclenchés suite à la réaction cognitive du sujet [Polich J.,

2007]. Comme l'indique la figure I.14, Le motif contient principalement 5 composantes, P1, N1, P2, N2 et P3, où P3 (P300) désigne le potentiel positif à une latence d'environ 300 ms après la présentation du stimulus déclencheur ; cependant, sa latence peut varier de 250 à 750 ms [Haider A. & Fazel-Rezai R., 2017] selon la difficulté ou la nature de la décision de l'utilisateur (conscient ou inconscient) [Kutas M. et al., 1977]. Le P300 peut être analysé dans la plupart des signaux ERP tels que le PEV, le PEA et le PES.

Le P300 est caractérisé par sa facilité de détection grâce à sa grande amplitude qui s'atténue progressivement lorsque le sujet s'habitue à ce stimulus inattendu ou lorsqu'on commence à s'éloigner de la zone centrale du lobe pariétal.

Le P300 endogène (lié à une réaction cognitive) est une composante importante des potentiels évoqués qui est largement utilisé pour les systèmes BCI [Guger C. et al., 2009], L'histoire de son introduction dans les BCIs remonte à 1988 [Farwell L. A., & Donchin E., 1988]. Depuis lors, un grand nombre d'études concernant les BCIs basées sur le P300 ont été proposé [Allison B. Z., 2010]. Leurs applications (même en dehors du laboratoire) s'accroissent sans cesse, notons par exemple les BCIs basées sur le P300 qui permettent aux utilisateurs de sélectionner des éléments affichés sur un écran d'ordinateur sont de plus en plus performants, portables, non invasive et peu coûteux [Lotte F., 2018]. En outre, de nombreux laboratoires différents explorent les possibilités d'accroître encore les capacités et l'utilité des BCIS basés sur le P300.

Variantes des paradigmes basés sur le motif P300 endogène

En pratique, le P300 peut être déclenché par des événements rares et pertinents selon trois différents types de paradigmes :

- Paradigme à stimulus unique [Donchin E., & Coles M. G., 1988] : Dans ce cas, un stimulus cible se produit de manière aléatoire dans le temps, sans aucun autre type de stimulus. Bien que le sujet doive réagir à la cible, ce paradigme est plus facile à réaliser que les autres types car le sujet doit uniquement répondre au stimulus cible.
- Paradigme oddball typique [Picton, T. W., 1992] : Dans ce type, Des stimuli auditifs ou visuels sont les plus couramment utilisés ; les stimuli non-cibles sont très probables (répétitifs) présentant les mêmes caractéristiques physiques et rarement interrompues par un stimulus cible (pertinent et rare). En effet, l'enregistrement est effectué sur les réactions cognitives du sujet uniquement au stimulus cible, c'est-à-dire l'événement irrégulier et rare qui fait apparaître le pic P300 dans l'EEG environ 300 ms après le début du stimulus (voir la figure I.15).

- Paradigme oddball modifié (oddball à trois stimuli) [Squires, K. C. et al., 1976] : Au contraire du paradigme oddball, cette variante introduit un troisième stimulus distracteur non ciblé mais peu probable [Polich, J., 1991]. Le distracteur provoque un nouveau potentiel appelé le P3a, avec une latence relativement courte (entre 250 et 280 ms) réparti sur la région frontale centrale du cortex cérébral, cette composante est associée à l'effet de surprise. Cependant, le stimulus cible suscite un potentiel P3b (également appelé P3 cible) qui est prédominant sur le lobe pariétal (plus postérieur que P3a). Le stimulus cible doit être connu à l'avance par le sujet qui impliquera dans une tâche cognitive en réagissant lorsqu'il perçoit le stimulus cible déclencheur. Le P3b est utilisé pour les BCIs afin de déterminer la cible particulière sur laquelle l'utilisateur porte son attention et il s'apparait entre 250 et 500 ms ou plus selon la complexité de la tâche cognitive à effectuer.

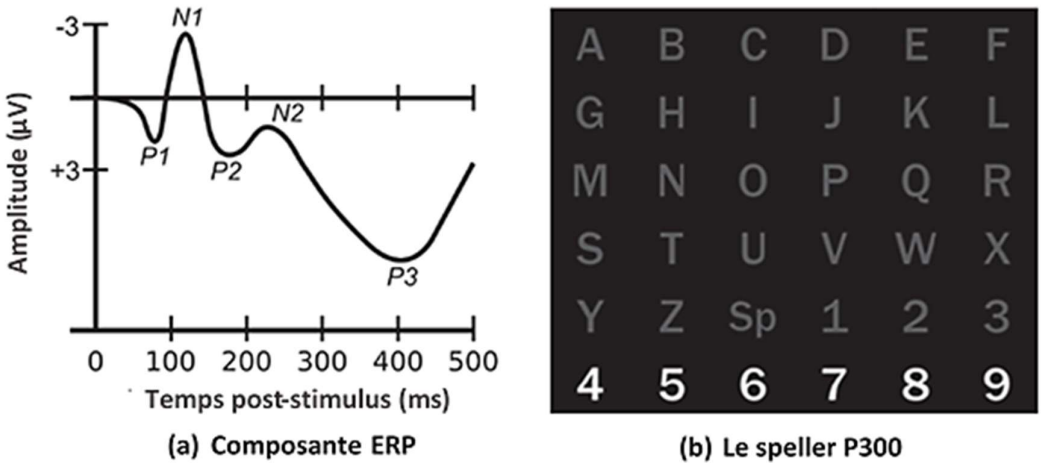


Figure I.14. Les composantes principales du potentiel évoqué P300.

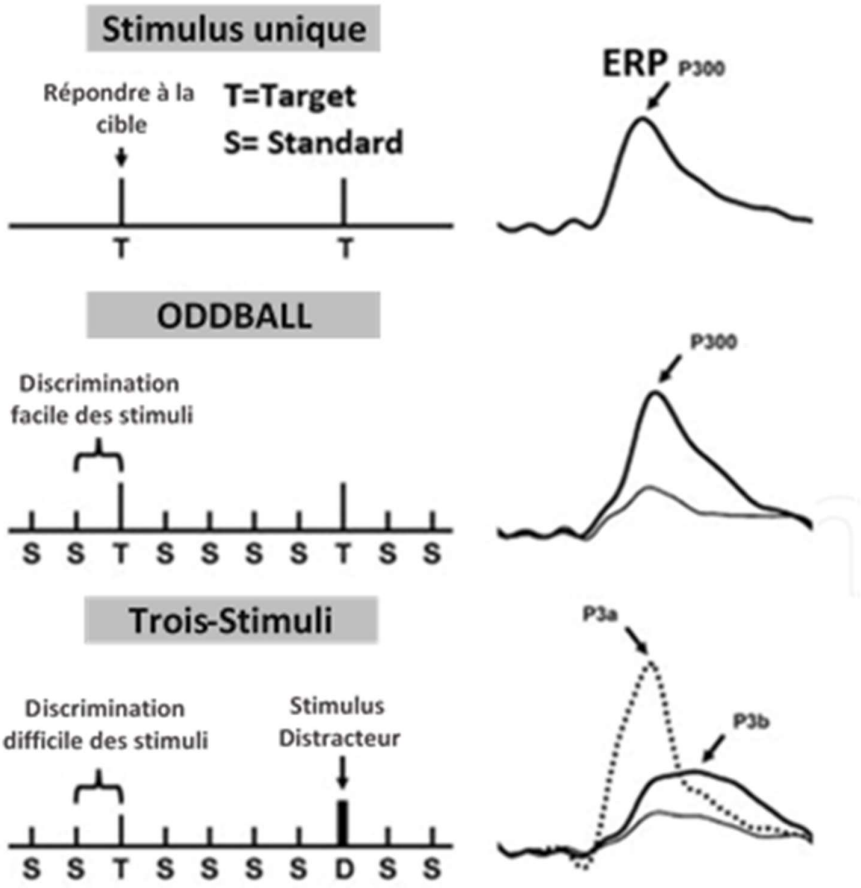


Figure I.15. Représentation schématique des trois paradigmes : stimulus unique (en haut), oddball (au milieu), et trois stimulus (en bas) ; l'ERP déclenché est présenté à droite.

Scénario expérimental du Speller P300

Ce scénario est basé sur les stimuli visuels permettant d'écrire en épelant des lettres. On utilise un écran ou un autre support d'affichage pour présenter au sujet une matrice 6 x 6 à l'écran composée de l'alphabet et des chiffres. Le sujet porte son attention sur une lettre spécifique. Les lignes et les colonnes de la matrice sont alors successivement surlignées. Lorsque la ligne ou la colonne contient la lettre choisie, un potentiel évoqué apparaît en théorie de type P300. On utilise alors un classifieur pour déterminer si ce signal correspond à une réponse positive à la stimulation ou non.

I.5.1.5. Potentiels évoqués à l'état d'équilibre

Les Potentiels évoqués à l'état d'équilibre SSEP (Steady-State Evoked Potentials) sont une autre sous-catégorie des potentiels évoqués, qui sont générés en réponse à un stimulus périodique à fréquence constante appelé un stimulus à l'état d'équilibre [Kawashima, Y. et al,

2021]. La fréquence des stimuli SSEP varie généralement entre 3,5 et 75 Hz et l'activité cérébrale générée dans la zone corticale associée imite le rythme du stimulus.

Il a été démontré que les oscillations cérébrales maintiennent généralement un niveau constant dans le temps, alors que les potentiels sont évoqués par des stimuli à l'état stable (par exemple, une lumière scintillante à fréquence fixe). Techniquement, le SSEP est défini comme une forme de réponse à une stimulation sensorielle répétitive dans laquelle les composantes fréquentielles constitutives de la réponse restent constantes dans le temps en amplitude et en phase [Galloway NR., 1990]. Le stimulus peut être visuel, auditif ou somesthésique, alors les SSEP peuvent être divisés en trois sous-catégories :

- **Les potentiels évoqués auditifs à l'état d'équilibre**

Les potentiels évoqués auditifs à l'état d'équilibre SSAEP (Steady state Auditory Evoked Potentials) sont enregistrés à partir de l'aire auditive du lobe temporal, mais proviennent du niveau du tronc cérébral [Hu, H. et al., 2021], ceux sont un type des potentiels évoqués oscillatoires en réponse à des stimuli auditifs rapides ; dont le motif le plus connu est le potentiel évoqué à 40 Hz qui est décrit en 1981 par Galambos [Galambos R. et al., 1981]. En 1984, il a été découvert que les SSAEP peuvent être évoqués par des stimuli uniques modulés en amplitude tonale, dans une large gamme de fréquences [Rickards F. W., & Clark G. M., 1982].

Le SSAEP est une modalité BCI prometteuse pour atténuer le problème de la fatigue oculaire qui survient souvent lors de l'utilisation de systèmes BCI basées sur le SSVEP.

Le scénario du paradigme basé sur le SSAEP consiste souvent à présenter au sujet une séquence audio contenant des chiffres cibles et non cibles, sachant que le sujet est toujours informé du chiffre qui était la cible. Chaque chiffre avait une durée de 200 ms et une même intensité vocale que les autres. Leur fréquence d'apparition a été randomisée de 250 à 450 ms. La tâche du sujet est d'identifier les chiffres cible dans la séquence et de discriminer leur latéralité tout en ignorant les voix non cibles. En effet, la reconnaissance volontaire par le sujet de la propriété d'une voix humaine cible, permet d'améliorer la discrimination entre les réponses cérébrales aux voix cibles et non cibles [Mosquera-Sanchez, J. A. et al., 2018].

Comme indiqué par la figure I.16. Les marqueurs neurologiques utilisés dans l'identifications de la voix cible parmi plusieurs voix différentes sont la composante tardive LPC (Late Positive Component) et le motif N2 qui sont des potentiels liés aux évènements (ERP).

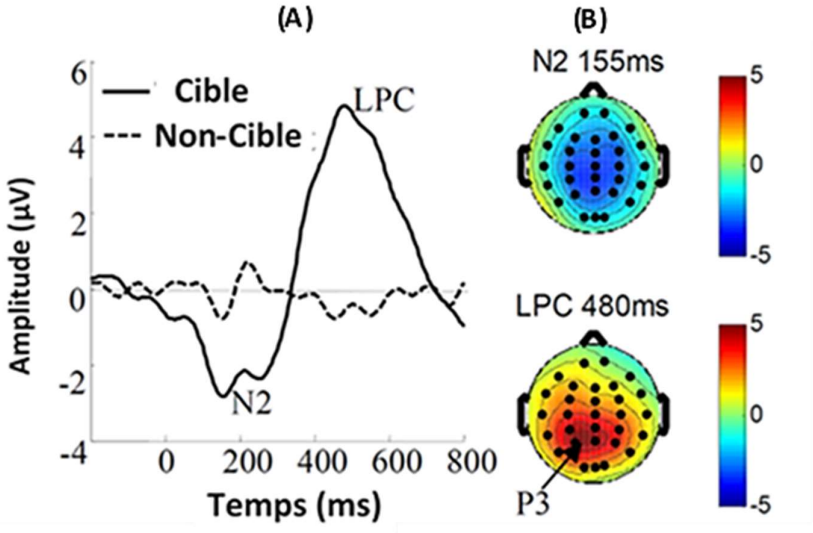


Figure I.16. Le modèle temporo-spatial des ERPs dans la BCI auditive active : (A) Grande forme d'onde moyenne à l'électrode P3. (B) Topographie de l'amplitude moyenne à 155 ms (valeur N2) et 485 ms (pic LPC). [Wolpaw, J. R. et al., 2020]

Dans les BCIs auditives, un plus grand nombre d'essais moyennés est nécessaire en raison du faible rapport signal/bruit de la réponse EEG auditive. De plus, la présentation d'un stimulus auditif prend généralement plus de temps par rapport à la modalité basée sur le SSVEP. Pour ces raisons, il est plus crucial de choisir un nombre optimal d'essais moyennés.

Autrefois, la technique SSAEP manquait de précision, mais en 2017, Yunyong et Yodchanan ont réussi à mettre en œuvre un paradigme expérimental dans lequel quatre commandes différentes peuvent être détectées avec une précision qui atteint 81% [Punsawad, Y., & Wongsawat, Y., 2017].

- **Les potentiels évoqués somesthésiques à l'état d'équilibre (SSSEP)** [Severens M. et al. 2010]

Les potentiels évoqués somesthésiques à l'état d'équilibre SSSEP (*Steady-State Somatosensory Evoked Potentials*) sont un type des potentiels évoqués qui peuvent être détectés en réponse à une stimulation tactile rapide et répétitive à fréquence constante. Leur caractérisation a été réalisée pour la première fois par Snyder et al. (1992) où la stimulation a été appliquée aux doigts et aux paumes [Snyder A.Z., 1992]. Les électrodes sont positionnées sur des emplacements spécifiques de la partie controlatérale de l'aire somesthésique primaire (ex. C3, Cz, C4, etc.) suivant le système international de positionnement d'électrodes 10-20.

Dans la littérature plusieurs paradigmes BCI sont basés sur les potentiels SSSEP notons : la stimulation des deux index aux différentes fréquences de 17 à 35 Hz [Müller-Putz G. R. et al., 2006], la stimulation des deux pouces aux fréquences 25, 33, 41.7, 50, 66.7 Hz et la stimulation des deux poignets à 27 Hz [Yao et al., 2014]. En effet, des dizaines de publications ont démontré la faisabilité de concevoir des interfaces cerveau-ordinateur basées sur les marqueurs SSSEP car ils sont suffisamment stables pour être exploitées pour de tels systèmes ; à savoir la discrimination des tâches [Chen X., 2015], le contrôle d'un fauteuil roulant [Kaufmann T. et al., 2014], etc.

Dans certains contextes d'utilisation tels que les malvoyants et les patients qui ne peuvent pas regarder les lumières scintillantes de manière cohérente, les BCIs sont conçues pour être opérées avec les SSSEP en demandant à l'utilisateur de se concentrer sur un rythme vibratoire à une fréquence particulière parmi plusieurs afin de réduire la fatigue causée par l'attention visuelle requise par le paradigme SSVEP. Dans certains protocoles, lorsque le bruit des vibrations préoccupe le sujet, il peut être isolé acoustiquement au moyen d'écouteurs distrayants. [Pokorny C. et al., 2013].

- **Les potentiels évoqués visuels à l'état d'équilibre (SSVEP)**

Les potentiels évoqués visuels à l'état d'équilibre SSVEP (*Steady-State Visual Evoked Potential*) étudiés pour la première fois par Regan en 1966 [Regan, D., & Spekreijse, H., 1986] sont des oscillations caractérisées par une amplitude élevée et une fréquence constante, elles sont obtenues sur le cortex visuel suite à la présentation d'un stimulus rapide et répétitif à rythme régulier par le moyen d'une diode électroluminescente (LED), un stroboscope ou avec un mécanisme d'inversement rapide des couleurs d'un motif présenté face au sujet. Il a été démontré par Vialatte F. et al. (2010) que la plage de fréquences des stimuli qui permettent de générer des potentiels SSVEP est comprise entre 3 et 40 Hz. En effet, les stimuli de fréquences inférieures à 3 Hz ne peuvent pas établir des réponses évoquées à l'état d'équilibre, mais au-delà de 40 Hz, l'amplitude du SSVEP devient trop faible pour être détectée et traitée. D'autres études ont prouvées qu'un paradigme SSVEP effectué dans la plage fréquentielle de 15 à 25 Hz souffre de deux difficultés majeures : les yeux de la personne deviennent plus fatigués et le risque d'avoir une crise d'épilepsie augmente également [Pastor, M. A. et al., 2003].

La figure I.17 représente un motif SSVEP détecté lors d'une stimulation visuelle à une fréquence de 15 Hz. Selon Zhu et al. 2010 le signal a été moyenné sur 10 essais synchronisés au

début du stimulus. La phase transitoire (portion d'une activité EEG spontanée) est clairement visible avant l'émergence des réponses à l'état d'équilibre en régime permanent.

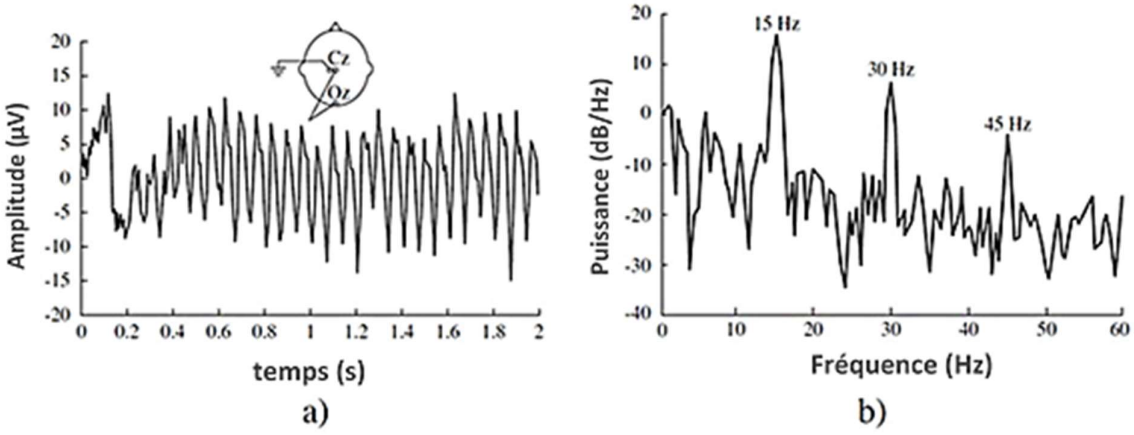


Figure I.17. (a) Un potentiel (SSVEP) mesuré sur le lobe occipital (Oz) par rapport à une référence (Cz) située sur le Vertex, (b), Le spectre de puissance montre clairement la composante fondamentale du SSVEP à 15 Hz, les autres composantes 30 Hz, 45 Hz sont des pics aux fréquences harmoniques supérieures.

Il est démontré que les SSVEP sont les motifs les désirés pour les BCIs nécessitant une communication rapide et fiable [Volosyak I., 2011 ; Wang, Y. et al., 2006 ; Bakardjian, H.et al., 2010] parce que (a) les SSVEP sont les moins susceptibles d'être affectés par les différents artefacts (EMG, EOG, etc.) [Perlstein W.M. et al., 2003] ; (b) l'utilisation d'une simple technique de filtrage dans l'espace fréquentiel permet facilement de séparer la composante qui correspond à la réponse au stimulus déclencheur [Delorme A., Makeig S., 2004] ; (c) le protocole expérimental nécessite un nombre total réduit des canaux dans le système d'acquisition EEG [Azinfar L. et al., 2013] ; (d) la bande de fréquences SSVEP large permet d'accroitre le degré de liberté sur le paradigme de contrôle [Diez P. F. et al., 2011] ; et (e) les caractéristiques des SSVEP sont relativement similaires par rapport à la disparité des sujets impliqués dans le paradigme SSVEP [Norcia A. M. et al., 2015]. Cependant, l'inconvénient majeur des applications basées sur les SSVEP est que les utilisateurs doivent rester longtemps concentrés sur une source clignotante, ce qui n'est pas agréable aux utilisateurs souffrant de troubles moteurs sévères. [Nicolas-Alonso, L. F., & Gomez-Gil, J., 2012]. En effet, les BCIs basées sur les SSVEP nécessitent un mécanisme de stimulation visuelle qui clignotent à des fréquences différentes, donc chaque motif est lié à une fréquence spécifique et par conséquent on pourra associer des options de commandes différentes ; par exemple : dans une application BCI on peut utiliser les motifs liés aux fréquences 5 Hz, 8 Hz, 11Hz et 14 Hz pour commander les

directions avant, arrière, droite et gauche. La sélection est effectuée par le sujet en concentrant la source de stimulation associée à l'action qu'il souhaite réaliser.

I.5.1.6. Rythmes sensori-moteurs

Les rythmes sensorimoteurs ou SMR (*Sensorimotor Rhythms*) sont des oscillations cérébrales enregistrés sur l'aire motrice primaire et l'aire somesthésique primaire (voir la figure I.18). Ces motifs peuvent être changés en raison d'un mouvement effectif ou d'un mouvement imaginé qui créent ce qu'on appelle la désynchronisation liée aux évènements (ERD) et la synchronisation liée aux évènements (ERS). Les SMR se répartissent généralement en trois grandes bandes de fréquences : mu (7-13 Hz), bêta (14-30 Hz) et gamma (30-200 Hz), sachant que la bande Mu (7-13 Hz) représente le rythme alpha présent dans les aires motrice et somesthésique primaires.

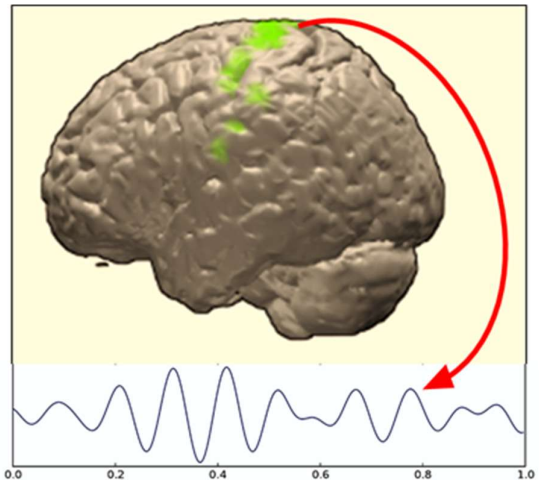


Figure I.18. Un échantillon d'une activité μ sur une période d'une seconde.

En 1959, il a été décrit par Chatrien que pendant le mouvement il y avait une présence désynchronisation du rythme en arceau (ERD) [Pfurtscheller G., & Aranibar A., 1977], cette diminution consiste au blocage rythmique causé par des corrélats de réseaux corticaux activés suite à un évènement interne (imagination de mouvement) ou externe (mouvement volontaire). Cependant, l'augmentation du rythme SMR due à un événement sensorimoteur survenant immédiatement après l'exécution du mouvement initial est appelée la synchronisation liée aux évènement ERS [Pfurtscheller G., 1992]. L'ERS peut être, du moins dans certaines circonstances, le corrélat d'un réseau cortical désactivé ou inhibé. La figure I.19 illustre les motifs ERD et ERS associés à des mouvements réels ou imaginaires du pied ou de la main.

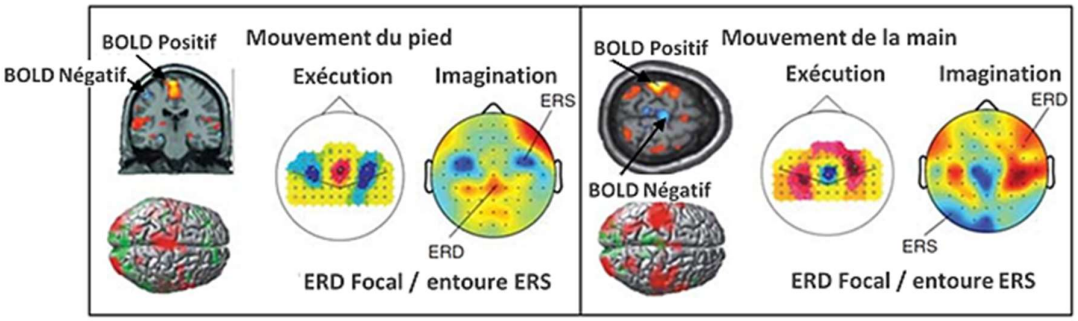


Figure I.19. Exemples de schémas ERD et ERS (obtenus à partir de données EEG et IRMf) lors de mouvements réels et imaginaires du pied (gauche) ou de la main (droite). Le schéma antagoniste d'activation (ERD, signal BOLD positif) et de désactivation (ERS, signal BOLD négatif) est clairement visible. On note également la similarité des cartes ERD pour les mouvements réels et imaginaires. D'après [Pfurtscheller, G., & Lopes da Silva, F. H., 2011]

L'ERD de la bande Mu commence juste avant le début du mouvement et atteint son maximum juste après le début du mouvement et récupère son niveau initial en quelques secondes. En revanche, la bande Bêta montre un ERD court pendant l'initiation du mouvement, suivi d'un ERS qui atteint le maximum après l'exécution du mouvement (voir la figure I.20).

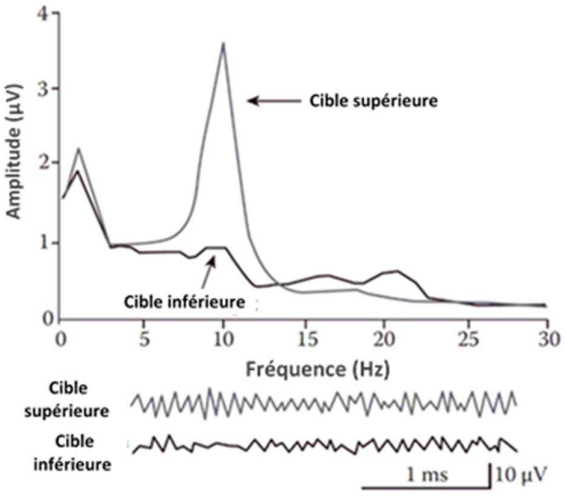


Figure I.20. Rythme sensorimoteur. D'après (Wolpaw, J. R. et al., 2002)

Dans le cas du mouvement, l'ERD du rythme Mu se produit dans le cortex moteur et l'ERS se produit dans le noyau sous-thalamique, alors que l'ERD du rythme Bêta se produit de manière uniforme dans le cortex moteur, le thalamus et le noyau sous thalamique [Klostermann F. et al., 2007].

I.5.1.7. Potentiels corticaux lents

Les potentiels corticaux lents SCP (*Slow Cortical Potentials*) sont des changements des niveaux de polarisations négatives ou positives de l'EEG (figure I.21) qui durent de 300 ms à plusieurs secondes. Ils proviennent de dépolarisations de l'arbre dendritique apical dans les couches corticales supérieures (couche I et II du néocortex) qui sont causées par l'activité synchrone des afférences intracorticales ou thalamocorticales [Zoefel, B. et al., 2011].

Un changement dans la direction de la polarité négative est le résultat de potentiels postsynaptiques excitateurs synchronisés très lents émis par les dendrites apicales des neurones pyramidaux, ceci est associé à une augmentation de l'activité corticale ou du mouvement, tandis qu'un changement dans la direction de la polarité positive est le résultat d'une diminution de ces mêmes potentiels, d'une activité inhibitrice des interneurones, ou d'un influx excitateur provenant des corps cellulaires des couches IV et V et ceci est associé à une diminution de l'activité corticale dont l'utilisateur est en état de repos [Birbaumer et al. 2000].

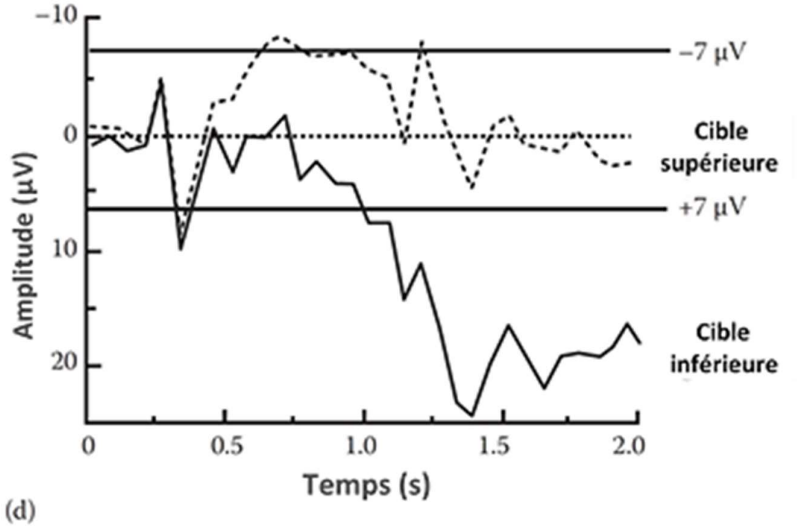


Figure I.21. Potentiel cortical lent. (Tiré de Wolpaw et al. 2002)

Les SCP requièrent souvent un entraînement extensif et approfondie en matière de stratégies cognitives et comportementales individuelles [Studer et al. 2014]. Ils prennent entre 1 s et plusieurs secondes pour se développer, et le taux de transfert d'informations est donc assez lent par rapport au SSVEP et au P300 visuel, ce qui réduit son efficacité lors de l'utilisation des applications BCI. Comme pour les BCIs basées sur les paradigmes sensori-moteurs, les BCIs basées sur les SCP ne s'appuient pas sur des stimuli externes tels que les stimuli visuels de la SSVEP afin d'obtenir des motifs cérébraux pour commander la BCI. Au lieu de cela, les utilisateurs contrôlent leurs processus de pensée afin d'interagir avec la BCI.

I.6. Conclusion

Ce chapitre met en évidence l'importance fondamentale du cerveau en tant qu'organe central du système nerveux humain. Son activité neuronale complexe est essentielle au bon fonctionnement du corps. Cependant, diverses pathologies et maladies neurodégénératives peuvent entraîner des dysfonctionnements neurologiques et cognitifs. Pour comprendre les interfaces cerveau-ordinateur, il est crucial d'avoir une compréhension approfondie de l'anatomie et du fonctionnement du système nerveux, qui se compose du système nerveux central (SNC) et du système nerveux périphérique (SNP). Le SNP est un réseau de nerfs reliant le SNC aux membres et aux organes, tandis que le SNC comprend l'encéphale et la moelle épinière. L'encéphale, qui comprend le cerveau, le cervelet et le tronc cérébral, joue un rôle central dans des fonctions telles que le mouvement, la pensée, la perception et la mémoire. Le cortex cérébral, en particulier, est d'un intérêt particulier dans la recherche sur les interfaces cerveau-ordinateur en raison de son implication dans les tâches motrices, sensorielles et cognitives. Le cortex cérébral, avec ses six couches de neurones et ses aires fonctionnelles, est impliqué dans des fonctions clés telles que la motricité, l'organisation, la planification, le langage et les sensations. Différentes aires du cortex jouent des rôles spécifiques dans le contrôle des mouvements, le traitement visuel et d'autres fonctions cognitives.

L'étude de l'activité cérébrale repose sur des techniques telles que l'électroencéphalographie (EEG) et l'électrocorticographie (ECoG), qui permettent l'exploration des potentiels de champs cérébraux. Ces techniques sont cruciales pour la recherche sur les interfaces cerveau-ordinateur, en particulier pour l'étude des potentiels évoqués et de l'activité spontanée du cerveau. Ce chapitre pose les bases théoriques nécessaires à la compréhension des interfaces cerveau-ordinateur. Il souligne l'importance du cerveau, de son anatomie et de son fonctionnement, ainsi que des techniques d'exploration de son activité. Ces connaissances fondamentales serviront de fondement pour approfondir l'étude des modèles neurologiques des interfaces cerveau-ordinateur dans les prochains chapitres.

BIBLIOGRAPHIE DU CHAPITRE I :

Abiri, R., Borhani, S., Sellers, E. W., Jiang, Y., & Zhao, X. (2019). A comprehensive review of EEG-based brain–computer interface paradigms. *Journal of neural engineering*, *16*(1), 011001.

Allison, B. Z., Brunner, C., Kaiser, V., Müller-Putz, G. R., Neuper, C., & Pfurtscheller, G. (2010). Toward a hybrid brain–computer interface based on imagined movement and visual attention. *Journal of neural engineering*, *7*(2), 026007.

Amunts, K., Schlaug, G., Jäncke, L., Steinmetz, H., Schleicher, A., Dabringhaus, A., & Zilles, K. (1997). Motor cortex and hand motor skills: structural compliance in the human brain. *Human brain mapping*, *5*(3), 206-215.

Azinfar, L., Amiri, S., Rabbi, A., & Fazel-Rezai, R. (2013). A review of P300, SSVEP, and hybrid P300/SSVEP brain-computer interface systems. *Braincomputer interface systems-recent progress and future prospects, ed. R. Fazel-Rezai (London: In Tech). doi*, *10*(2013), 187024.

Bakardjian, H., Tanaka, T., & Cichocki, A. (2010). Optimization of SSVEP brain responses with application to eight-command brain–computer interface. *Neuroscience letters*, *469*(1), 34-38.

Bear, M., Connors, B., & Paradiso, M. A. (2020). *Neuroscience: exploring the brain, enhanced edition: exploring the brain*. Jones & Bartlett Learning.

Birbaumer, N., Elbert, T., Canavan, A. G., & Rockstroh, B. (1990). Slow potentials of the cerebral cortex and behavior. *Physiological reviews*, *70*(1), 1-41.

Brunner, C., Allison, B. Z., Altstätter, C., & Neuper, C. (2011). A comparison of three brain–computer interfaces based on event-related desynchronization, steady state visual evoked potentials, or a hybrid approach using both signals. *Journal of neural engineering*, *8*(2), 025010.

Buonocore, M., & Demartini, L. (2016). Inhibition of somatosensory evoked potentials during different modalities of spinal cord stimulation: a case report. *Neuromodulation: Technology at the Neural Interface*, *19*(8), 882-884.

Cecotti, H., & Ries, A. J. (2017). Best practice for single-trial detection of event-related potentials: Application to brain-computer interfaces. *International Journal of Psychophysiology*, *111*, 156-169.

Chagnac-Amitai, Y. A. E. L., & Connors, B. W. (1989). Synchronized excitation and inhibition driven by intrinsically bursting neurons in neocortex. *Journal of Neurophysiology*, *62*(5), 1149-1162.

Chen, X., Wang, Y., Nakanishi, M., Gao, X., Jung, T. P., & Gao, S. (2015). High-speed spelling with a noninvasive brain–computer interface. *Proceedings of the national academy of sciences*, *112*(44), E6058-E6067.

Chiappa, K. H. (1997). Evoked Potentials in Clinical Medicine. Lippincott Williams & Wilkins.

Clark, V. P., & Hillyard, S. A. (1996). Spatial selective attention affects early extrastriate but not striate components of the visual evoked potential. Journal of cognitive neuroscience, 8(5), 387-402.

Da Silva, F. L. (2010). Electrical Neuroimaging of the Human Brain: Overview and Future Directions. In C. Mulert & L. Lemieux (Eds.), *EEG - fMRI: Physiological Basis, Technique, and Applications* (pp. 3-11). Springer.

Delorme, A., & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of neuroscience methods*, *134*(1), 9-21.

Diez, P. F., Mut, V. A., Avila Perona, E. M., & Laciar Leber, E. (2011). Asynchronous BCI control using high-frequency SSVEP. *Journal of neuroengineering and rehabilitation*, *8*(1), 1-9.

Donchin, E., & Coles, M. G. (1988). Is the P300 component a manifestation of context updating? *Behavioral and brain sciences*, *11*(3), 357-374.

Dronkers, N. F. (1996). A new brain region for coordinating speech articulation. *Nature*, *384*(6605), 159-161.

Farwell, L. A., & Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, *70*(6), 510-523.

Felleman, D.J., and Van Essen, D.C. (1991). Distributed hierarchical processing in the primate cerebral cortex. Cerebral cortex (New York, N.Y.: 1991), 1(1), 1-47.

Franco, A., Eberlen, J., Destrebecqz, A., Cleeremans, A., & Bertels, J. (2015). Rapid serial auditory presentation. *Experimental psychology*.

Galambos, R., Makeig, S., & Talmachoff, PJ. (1981). "A 40-Hz auditory potential recorded from the human scalp." Proceedings of the National Academy of Sciences, 78(4), 2643-2647.

Galloway NR. Human Brain Electrophysiology: Evoked Potentials and Evoked Magnetic Fields in Science and Medicine. Br J Ophthalmol. 1990 Apr ;74(4) :255. PMCID: PMC1042082.

Gazzaniga, M. S. (2004). *The cognitive neurosciences*. MIT press.

Gescheider, G. A., Wright, J. H., & Verrillo, R. T. (2010). *Information-processing channels in the tactile sensory system: A psychophysical and physiological analysis*. Psychology press.

Gray, C. M. (1994). Synchronous oscillations in neuronal systems: mechanisms and functions. *Journal of computational neuroscience*, *1*(1-2), 11-38.

Gruss, L. F., Wieser, M. J., Schweinberger, S. R., & Keil, A. (2012). Face-evoked steady-state visual potentials: effects of presentation rate and face inversion. *Frontiers in Human Neuroscience*, *6*, 316.

Guger, C., Daban, S., Sellers, E., Holzner, C., Krausz, G., Carabalona, R., ... & Edlinger, G. (2009). How many people are able to control a P300-based brain–computer interface (BCI)?. *Neuroscience letters*, *462*(1), 94-98.

Hackett, T.A. (2011). Information flow in the auditory cortical network. Hearing research, 271(1-2), 133-146.

Haider, A., & Fazel-Rezai, R. (2017). Application of P300 event-related potential in brain-computer interface. *Event-Related Potentials and Evoked Potentials*, *1*, 19-36.

He, B., Yuan, H., Meng, J., & Gao, S. (2020). Brain–computer interfaces. *Neural engineering*, 131-183.

He, B. J. (2013). Spontaneous and task-evoked brain activity negatively interact. *Journal of Neuroscience*, *33*(11), 4672-4682.

Hsiao, S. S., Johnson, K. O., & Twombly, I. A. (1993). Roughness coding in the somatosensory system. *Acta psychologica*, *84*(1), 53-67.

Hu, H., & Ewert, S. D. (2021, December). Exploring artifact rejection for high-pulse rate electrically evoked auditory steady state responses in cochlear implants users. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (pp. 1311-1316). IEEE.

Islam, M. N., Sulaiman, N., Bari, B. S., Rashid, M., & Mustafa, M. (2021). Auditory Evoked Potential (AEP) Based Brain-Computer Interface (BCI) Technology: A Short Review. *Advances in Robotics, Automation and Data Analytics: Selected Papers from iCITES 2020*, 272-284.

Kaas, J. H. (2004). Evolution of somatosensory and motor cortex in primates. *The Anatomical Record Part A: Discoveries in Molecular, Cellular, and Evolutionary Biology: An Official Publication of the American Association of Anatomists*, *281*(1), 1148-1156.

Kandel, E. R., Markram, H., Matthews, P. M., Yuste, R., & Koch, C. (2013). Neuroscience thinks big (and collaboratively). *Nature Reviews Neuroscience*, *14*(9), 659-664.

Kandel, E. R., Schwartz, J. H., Jessell, T. M., Siegelbaum, S., Hudspeth, A. J., & Mack, S. (Eds.). (2000). *Principles of neural science* (Vol. 4, pp. 1227-1246). New York: McGraw-hill.

Kaufmann, T., Herweg, A., & Kübler, A. (2014). Toward brain-computer interface based wheelchair control utilizing tactually-evoked event-related potentials. *Journal of neuroengineering and rehabilitation*, *11*, 1-17.

Kawashima, Y., Li, R., Chen, S. C. Y., Vickery, R. M., Morley, J. W., & Tsuchiya, N. (2021). Steady state evoked potential (SSEP) responses in the primary and secondary somatosensory cortices of anesthetized cats: Nonlinearity characterized by harmonic and intermodulation frequencies. *Plos one*, *16*(3), e0240147.

Klostermann, F., Nikulin, V. V., Kühn, A. A., Marzinzik, F., Wahl, M., Pogosyan, A., ... & Curio, G. (2007). Task-related differential dynamics of EEG alpha-and beta-band synchronization in cortico-basal motor structures. *European journal of Neuroscience*, *25*(5), 1604-1615.

Kutas, M., McCarthy, G., & Donchin, E. (1977). Augmenting mental chronometry: The P300 as a measure of stimulus evaluation time. *Science*, *197*(4305), 792-795.

Laufs, H., Krakow, K., Sterzer, P., Eger, E., Beyerle, A., Salek-Haddadi, A., & Kleinschmidt, A. (2003). Electroencephalographic signatures of attentional and cognitive default modes in spontaneous brain activity fluctuations at rest. Proceedings of the national academy of sciences, 100(19), 11053-11058.

Lees, S., Dayan, N., Cecotti, H., McCullagh, P., Maguire, L., Lotte, F., & Coyle, D. (2018). A review of rapid serial visual presentation-based brain–computer interfaces. *Journal of neural engineering*, *15*(2), 021001.

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: a 10-year update. *Journal of neural engineering*, *15*(3), 031005.

Luck, S. J. (2014). *An introduction to the event-related potential technique*. MIT press.

Martin, J. H. (2012). *Neuroanatomy text and atlas*. McGraw-Hill.

Miller, L., & Hatsopoulos, N. I. C. H. O. L. A. S. (2012). Neuronal Activity in Motor Cortex and Related Areas. *Brain-Computer Interfaces : Principles and Practice*, 15-44.

Mosquera-Sanchez, J. A., Sarrazin, M., Janssens, K., De Oliveira, L. P. R., & Desmet, W. (2018). Multiple target sound quality balance for hybrid electric powertrain noise. *Mechanical Systems and Signal Processing*, *99*, 478-503.

Muller-Putz, G. R., Scherer, R., Neuper, C., & Pfurtscheller, G. (2006). Steady-state somatosensory evoked potentials: suitable brain signals for brain-computer interfaces? *IEEE transactions on neural systems and rehabilitation engineering*, *14*(1), 30-37.

Nicolas-Alonso, L. F., & Gomez-Gil, J. (2012). Brain computer interfaces, a review. *sensors*, *12*(2), 1211-1279.

Nieuwenhuys, R., Voogd, J., & Van Huijzen, C. (2007). *The human central nervous system: a synopsis and atlas*. Springer Science & Business Media

Norcia, A. M., Appelbaum, L. G., Ales, J. M., Cottereau, B. R., & Rossion, B. (2015). The steady-state visual evoked potential in vision research: A review. *Journal of vision*, *15*(6), 4-4.

Pastor, M. A., Artieda, J., Arbizu, J., Valencia, M., & Masdeu, J. C. (2003). Human cerebral activation during steady-state visual-evoked responses. *Journal of neuroscience*, *23*(37), 11621-11627.

Penfield, W., & Boldrey, E. (1937). Somatic motor and sensory representation in the cerebral cortex of man as studied by electrical stimulation. *Brain*, *60*(4), 389-443.

Perlstein, W. M., Cole, M. A., Larson, M., Kelly, K., Seignourel, P., & Keil, A. (2003). Steady-state visual evoked potentials reveal frontally-mediated working memory activity in humans. *Neuroscience letters*, *342*(3), 191-195.

Pestronk, A. (1997). Histology of the nervous system of man and vertebrates. *Neurology*, *48*(1), 299-300.

Pfurtscheller, G., & Neuper, C. (2001). Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, *89*(7), 1123-1134.

Pfurtscheller, G., & Aranibar, A. (1977). Event-related cortical desynchronization detected by power measurements of scalp EEG. *Electroencephalography and clinical neurophysiology*, *42*(6), 817-826.

Pfurtscheller, G., & Lopes da Silva, F. H. (2011). Event-related desynchronization (ERD) and event-related synchronization (ERS). In Event-related dynamics of brain oscillations (pp. 37-49). Elsevier.

Pfurtscheller, G. (1992). Event-related synchronization (ERS): an electrophysiological correlate of cortical areas at rest. *Electroencephalography and clinical neurophysiology*, *83*(1), 62-69.

Picton, T. W. (1992). The P300 wave of the human event-related potential. *Journal of clinical neurophysiology*, *9*(4), 456-479.

Pokorny, C., Klobassa, D. S., Pichler, G., Erlbeck, H., Real, R. G., Kübler, A., ... & Müller-Putz, G. R. (2013). The auditory P300-based single-switch brain–computer interface: paradigm transition from healthy subjects to minimally conscious patients. *Artificial intelligence in medicine*, *59*(2), 81-90.

Polich, J. (1991). P300 in clinical applications: meaning, method, and measurement. *American Journal of EEG Technology*, *31*(3), 201-231.

Polich, J. (2007). Updating P300: an integrative theory of P3a and P3b. *Clinical neurophysiology*, *118*(10), 2128-2148.

Punsawad, Y., & Wongsawat, Y. (2017). Multi-command SSAEP-based BCI system with training sessions for SSVEP during an eye fatigue state†. Technology and Economics of Smart Grids and Sustainable Energy, 1(1), 123-136. https://doi.org/10.1002/tee.22441.

Purves, D., Augustine, G. J., Fitzpatrick, D., Hall, W., LaMantia, A. S., & White, L. (2019). *Neurosciences*. De Boeck Supérieur.

Regan, D., & Spekreijse, H. (1986). Evoked potentials in vision research 1961–1986. *Vision research*, *26*(9), 1461-1480.

Regan, D. (1966). Some characteristics of average steady-state and transient responses evoked by modulated light. *Electroencephalography and clinical neurophysiology*, *20*(3), 238-248.

Reza, A. (2015). ELECTROCORTICOGRAPHY-BASED BRAIN COMPUTER INTERFACE FOR PEDIATRIC PATIENT: CHALLENGE ON THE HORIZON. *Jurnal Kedokteran p-ISSN*, *2460*, 9749.

Rickards, F. W., & Clark, G. M. (1982). Steady-state evoked potentials in humans to continuous amplitude modulated tones. *The Journal of the Acoustical Society of America*, *72*(S1), S54-S54.

Sanei, S., & Chambers, J. A. (2013). *EEG signal processing*. John Wiley & Sons.

Schmahmann, J. D. (2004). Disorders of the cerebellum: Ataxia, dysmetria of thought, and the cerebellar cognitive affective syndrome. Journal of Neuropsychiatry and Clinical Neurosciences, 16(3), 367-378.

Severens, M., Farquhar, J., Desain, P., Duysens, J., & Gielen, C. (2010). Transient and steady-state responses to mechanical stimulation of different fingers reveal interactions based on lateral inhibition. Clinical Neurophysiology, 121(12), 2090-2096.

Snyder, A. Z. (1992). Steady-state vibration evoked potentials: descriptions of technique and characterization of responses. *Electroencephalogr. Clin. Neurophysiol.* 84, 257–268.

Squires, K. C., Wickens, C., Squires, N. K., & Donchin, E. (1976). The effect of stimulus sequence on the waveform of the cortical event-related potential. *Science*, *193*(4258), 1142-1146.

Sugerman, R. A. (2006). Structure and function of the neurologic system. *Pathophysiology. The Biologic Basis for Disease in Adults and Children. 5th ed. St. Louis, Missouri: Elsevier Mosby*, 411-435.

Sutton, S., Braren, M., Zubin, J., & John, E. R. (1965). Evoked-potential correlates of stimulus uncertainty. *Science*, *150*(3700), 1187-1188.

Tjia, M., Yu, X., Jammu, L. S., Lu, J., & Zuo, Y. (2017). Pyramidal neurons in different cortical layers exhibit distinct dynamics and plasticity of apical dendritic spines. *Frontiers in neural circuits*, *11*, 43.

Treder, M. S., Schmidt, N. M., & Blankertz, B. (2011). Gaze-independent brain–computer interfaces based on covert attention and feature attention. *Journal of neural engineering*, *8*(6), 066003.

Vialatte, F. B., Maurice, M., Dauwels, J., & Cichocki, A. (2010). Steady-state visually evoked potentials: focus on essential paradigms and future perspectives. *Progress in neurobiology*, *90*(4), 418-438.

Volosyak, I. (2011). SSVEP-based Bremen–BCI interface—boosting information transfer rates. *Journal of neural engineering*, *8*(3), 036020.

Wang, Y., Wang, R., Gao, X., Hong, B., & Gao, S. (2006). A practical VEP-based brain-computer interface. *IEEE Transactions on neural systems and rehabilitation engineering*, *14*(2), 234-240.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. Clinical Neurophysiology, 113(6), 767-791.

Wolpaw, J. R., Millán, J. D. R., & Ramsey, N. F. (2020). Brain-computer interfaces: Definitions and principles. *Handbook of clinical neurology*, *168*, 15-23.

Yamada, T., Yeh, M., & Kimura, J. (2004). Fundamental principles of somatosensory evoked potentials. *Physical Medicine and Rehabilitation Clinics*, *15*(1), 19-42.

Yao, L., Meng, J., Zhang, D., Sheng, X., & Zhu, X. (2013). Combining motor imagery with selective sensation toward a hybrid-modality BCI. *IEEE Transactions on Biomedical Engineering*, *61*(8), 2304-2312.

York, D. H., Pulliam, M. W., Rosenfeld, J. G., & Watts, C. (1981). Relationship between visual evoked potentials and intracranial pressure. *Journal of neurosurgery*, *55*(6), 909-916.

Zhu, D., Bieger, J., Molina, G. G., & Aarts, R. M. (2010). A survey of stimulation methods used in SSVEP-based BCIs. *Computational intelligence and neuroscience*, *2010*, 1-12.

Zoefel, B., Huster, R. J., & Herrmann, C. S. (2011). Neurofeedback training of the upper alpha frequency band in EEG improves cognitive performance. *Neuroimage*, *54*(2), 1427-1431.

CHAPITRE II

INTERFACES CERVEAU-ORDINATEUR (BCI)

CONTENU

II.1. Introduction
II.2. Présentation des interfaces cerveau-ordinateur
II.3. Système Acquisition des signaux EEG
  II.3.1. Prétraitement
  II.3.2. Posttraitement
II.4. Catégorisation des interfaces Cerveau-Ordinateur
II.5. Interfaces Cerveau ordinateur catégorisées selon le protocole de fonctionnement
  II.5.1. Interfaces Cerveau-Ordinateur avec des protocoles synchrones (Cue-based)
  II.5.2. Interfaces Cerveau-Ordinateur basées dur des protocoles au rythme autonome (Self-paced)
  II.5.3. Interfaces Cerveau-Ordinateur basées sur des protocoles combinés synchrones et au rythme autonome
  II.5.4. Interfaces Cerveau-Ordinateur basées sur des protocoles non-adaptatifs
  II.5.5. Interfaces Cerveau-Ordinateur basées sur des protocoles adaptatifs
II.6. Interfaces Cerveau ordinateur catégorisées selon le motif cérébral
II.7. Interfaces Cerveau-ordinateur catégorisées suivant le mode d'acquisition
  II.7.1. Mode d'acquisition non-invasif
    II.7.1.1. Électroencéphalographie – EEG
    II.7.1.2. Magnétoencéphalographie - MEG
    II.7.1.3. Spectroscopie Proche de l'Infrarouge fonctionnelle – fNIRS
    II.7.1.4. Imagerie par Résonance Magnétique fonctionnelle - IRMf
    II.7.1.5. Tomographie par Emission de Positrons - TEP
  II.7.2. Mode d'acquisition semi-invasif
    II.7.2.1. Électrocorticographie - ECoG
  II.7.3. Mode d'acquisition invasif
    II.7.3.1. Enregistrement Neuronal Intracortical - INR
II.8. Interfaces Cerveau-ordinateur catégorisées suivant la stratégie de fonctionnement
II.9. Conclusion
Bibliographie du Chapitre II

II.1. Introduction

Comme nous le savons, le cerveau fonctionne en transmettant un signal pour induire le mouvement de différentes parties du corps à travers les nerfs et les muscles. Apparu dans les années 1950, une interface cerveau-ordinateur (ICO ou BCI) est une méthode de communication directe entre l'activité électrique du cerveau et un dispositif externe (ordinateur, organe prothétique, etc.) [Dornhege et al., 2007], qui permet aux utilisateurs d'envoyer les commandes en produisant différent types de signaux cérébraux.

Depuis, des applications prometteuses des BCI ont vu le jour notamment dans les domaines d'assistances thérapeutiques, parmi lesquelles celles liées à la rééducation et réhabilitation fonctionnelle, ce qui a permis de surmonter les handicaps moteurs des personnes souffrant des lésions du système nerveux périphérique. C'est le cas où la personne est partiellement ou totalement incapable d'exécuter les commandes émises par le système nerveux central en conséquence d'un traumatisme cérébral neuronal, comme le syndrome d'enfermement 'Locked-in Syndrome', la myopathie de Duchenne, etc. [Birbaumer N., & Cohen L. G., 2007]. Des membres prothétiques robotisés pouvant immigrer leur mobilité aux personnes handicapées.

Cependant, on peut rencontrer l'utilisation des interfaces cerveau-ordinateur dans plusieurs applications, la robotique collaborative [Dmytriyev Y. et al., 2021] et interactive [Vidaurre C., 2010], le divertissement [Allison B. Z. et al., 2010a], la réalité virtuelle [Lécuyer A. et al., 2008], le multimédia [Gao X. et al., 2003], etc.

Dans ce chapitre nous présentons les systèmes BCI et proposons leur état-de-l 'art.

II.2. Présentation des interfaces cerveau-ordinateur

La conception d'un système BCI est relativement une tâche complexe qui nécessite des connaissances pluridisciplinaires en matière de : Génie électrique, programmation, apprentissage automatique, neurosciences, etc. Cependant ce type de systèmes est composé de différents blocs fonctionnels de base, formant ainsi une structure en boucle fermée, comme illustré par la figure II.1.

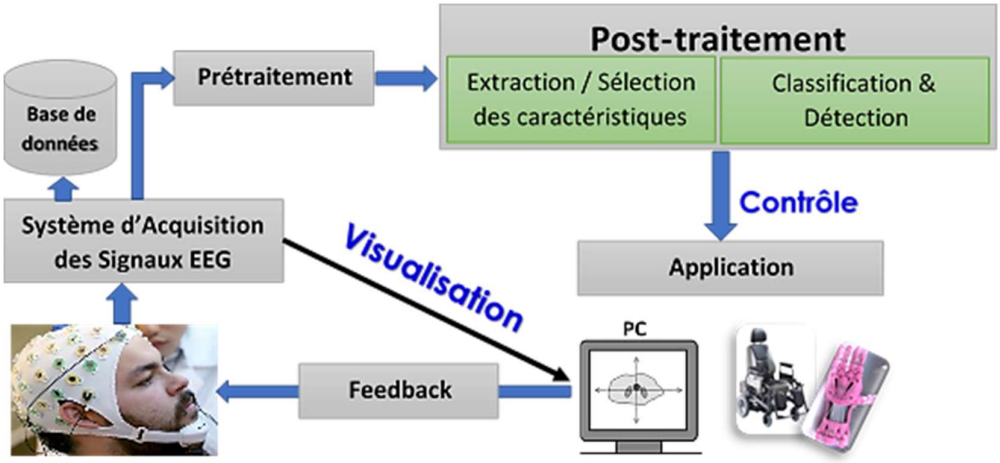


Figure II.1. Structure fonctionnelle d'une interface cerveau-ordinateur illustrant la série de composantes utilisées pour traduire l'activité cérébrale en signaux de contrôle.

II.3. Système Acquisition des signaux EEG

Cette étape consiste à enregistrer l'activité cérébrale du sujet, plusieurs techniques sont utilisées pour effectuer cette étape telles que : la Magnétoencéphalographie (MEG) [Baillet S., 2017], l'Imagerie par Résonance Magnétique fonctionnelle (IRM*f*) [Logothetis N. K., 2008] ou encore l'Electrocorticographie (ECoG) [Schalk G. et al., 2011]. Cependant, la méthode la plus utilisée fréquemment dans nombreux champs d'applications c'est bien que l'Electroencéphalographie (EEG) [Niedermeyer E., & da Silva F. L., 2005].

II.3.1. Prétraitement

Le prétraitement des signaux EEG est une étape importante avant l'extraction de toute caractéristique. Elle se fait après l'acquisition des données. Le prétraitement améliore la qualité du signal mesuré et donc permet de mettre en valeur l'information pertinente qui est présente dans le flux brut. Une étape typique du prétraitement est le filtrage. Les filtres sont conçus pour supprimer les composantes fréquentielles indésirables (la composante continue l'Offset CC de 0 Hz, les bruits hautes-fréquences HF, etc.). Dans le prétraitement, la sélection de canal par rapport aux décimations de données est déterminée de manière à améliorer les performances de classification. Des segments de données sont collectés et un filtre de moyenne mobile est appliqué pour de meilleures performances [Krusienski D. J. et al., 2008].

II.3.2. Posttraitement

Le post-traitement des signaux EEG dans les interfaces cerveau-ordinateur (BCI) est une étape cruciale pour extraire les caractéristiques pertinentes des données mesurées. Cette étape comprend l'extraction de caractéristiques fréquentielles, spatiales, temporelles et statistiques à

partir des signaux EEG recueillis dans un paradigme multi-essais [Lemm S. et al., 2005 ; Grosse-Wentrup M., & Buss M., 2008].

Parmi ces caractéristiques, on trouve les caractéristiques fréquentielles, telles que la puissance spectrale dans des bandes de fréquence spécifiques [Ang K.K. et al., 2012]. Les caractéristiques spatiales peuvent être représentées par des méthodes telles que les CSP (*Common Spatial Pattern*), qui identifient les projections spatiales discriminantes pour la classification des signaux EEG [Ramoser H. et al., 2000]. Pour les caractéristiques statistiques, des mesures comme l'écart type et la variance peuvent être calculées sur les coefficients des ondelettes extraites des signaux EEG [Yang, R. et al., 2010]. En ce qui concerne les caractéristiques temporelles, on peut mesurer les latences des réponses cérébrales, qui représentent le délai entre un stimulus et la réponse du cerveau [Aggarwal S., & Chugh N., 2019]. En utilisant ces différentes caractéristiques, il est possible d'obtenir une représentation complète des signaux EEG et d'améliorer la performance des classifieurs dans les BCIs.

Ensuite, la réduction de la dimensionalité des caractéristiques est effectuée à l'aide d'algorithmes tels que l'analyse en composantes principales PCA (*Principal Component Analysis*) et l'analyse en composantes indépendantes ICA (*Independent Component Analysis*) [Samek et al., 2013]. Pour améliorer les performances des classifieurs, la sélection des caractéristiques pertinentes est réalisée en utilisant des algorithmes tels que la sélection séquentielle des caractéristiques SFS (*Sequential Features Selection*) et le calcul de la variance entre les caractéristiques, suivis de la suppression des composantes fortement corrélées [Lal et al., 2005 ; Krusienski et al., 2008]. Cette étape permet de réduire les coûts de calcul et d'augmenter la précision des estimateurs dans les BCIs. Les tableaux de caractéristiques résultants sont ensuite utilisés comme entrées pour des classifieurs linéaires ou non linéaires, en fonction du niveau de complexité des données et du degré de séparabilité obtenue [Pfurtscheller, G. et al., 1998]. La génération des commandes permettant de piloter le système BCI dépend de la performance du modèle validé lors de l'apprentissage [Lemm, S. et al., 2005].

II.4. Catégorisation des interfaces Cerveau-Ordinateur

Les interfaces cérébrales sont généralement classées en fonction de certains critères :

- La nature du motif cérébral (SSVEP, P300, ERD, ERS, etc.).
- Le type de stimulus appliqué (Visuel, auditif, tactile, etc.).
- Le protocole de fonctionnement (asynchrone, synchrone, etc.).
- Le Mode d'enregistrement utilisé (invasif, semi-invasif non-invasif).

La figure II.2 illustre les différents critères selon lesquels les interfaces cerveau-ordinateur sont catégorisées.

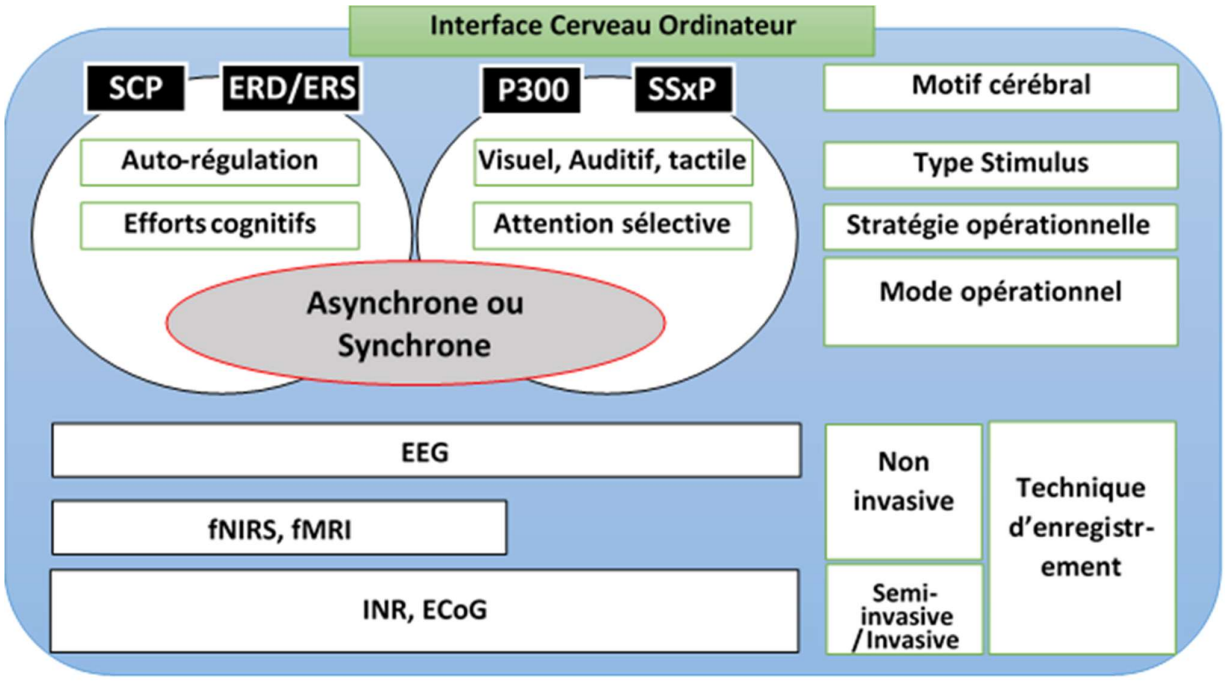


Figure II.2. Catégorisation des interfaces cerveau-ordinateur.

Il existe de nombreux motifs cérébraux qui peuvent être utilisés pour communiquer avec une BCI [Ince, N. F. et al., 2005 ; Frolov, A. et al., 2020].

II.5. Interfaces cerveau-ordinateur catégorisées selon le protocole de fonctionnement

L'Interface cerveau-ordinateur ne fonctionne pas de manière isolée mais dans son propre environnement avec la disponibilité des trois éléments principaux : l'utilisateur, l'application et le support externe de la BCI [Perdikis S. et al., 2018].

Avant tout, on doit fournir un concept très important lié à la façon dont la BCI doit être initié :

Pendant l'utilisation d'une BCI, il y a deux possibilités pour l'utilisateur : soit il intervient activement sur l'interface (exemple : le contrôle délibéré par imagination), soit il n'utilise du tout l'interface BCI (exemple : la pensée, l'écriture, la lecture, le sommeil, etc.) :

- L'état d'utilisation active de la BCI est appelé *contrôle intentionnel*.
- L'état de non-utilisation de la BCI est dite *absence de contrôle* NC (*No Control*).

À la lumière de ce qu'on a mentionné, on distingue quatre conceptions différentes des BCI selon les protocoles d'exploitation :

II.5.1. Interfaces Cerveau-Ordinateur avec des protocoles synchrones (Cue-based)

Ce type de BCI est basé sur des indices (Cue-based), où l'on suppose que les utilisateurs ne peuvent le contrôler qu'à des moments précis, par exemple ; produire un signal de commande après un nombre déterminé des indicateurs visuels. En effet, un protocole synchrone limite les temps pendant lesquels la BCI va effectivement convertir l'activité cérébrale de l'utilisateur en action [Mason S. G., & Birch G. E., 2003 ; Müller-Putz, G. R. et al., 2005].

Dans ce type de protocoles, la BCI commence immédiatement à analyser la réponse cérébrale produite après la présence d'un stimulus sensoriel spécifique (visuel, auditif, tactile, etc.), en d'autre terme, la BCI génère les commandes pendant la période post-stimulus, comme illustré par la figure II.3.

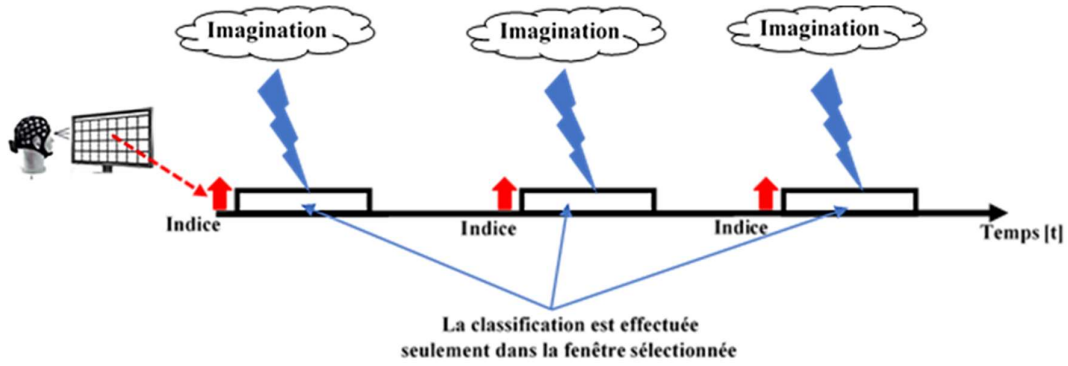


Figure II.3. Disponibilité du système pendant différents paradigmes de contrôle temporel.

Un type courant des BCI synchrones est celui dite le P300 speller [Farwell L. & Donchin E., 1988]. Cette BCI est basée sur le paradigme des potentiels évoqués ERP (*Event-Related Potentials*). Le potentiel évoqué P300 (parfois P3) atteigne un pic positif environ 300 ms après le début de stimulus. Une matrice 6*6 s'affiche à l'écran contenant un alphabet, des chiffres et d'autres caractères spéciaux. L'utilisateur doit sélectionner un caractère et en porte son attention pendant que chaque ligne ou colonne de la matrice clignote successivement. L'utilisateur observe les lignes et les colonnes contenant le caractère choisi, ensuite un p300 est généré à chaque note de la lettre sélectionnée, puis le classificateur commence à déterminer si la réponse du sujet correspond à la lettre choisie ou non. Ce paradigme permet ainsi à l'utilisateur d'épeler grâce à l'activité cérébrale [Sellers E. W. et al. 2006 ; Vaughan T.M. et al. 2006 ; Blankertz, B. et al., 2001]. Cette BCI est synchrone, car la sortie est basée sur les réponses de l'utilisateur à des stimuli présentés selon un calendrier déterminé par la BCI.

Ce type des BCI est *périodiquement* disponible pour l'utilisateur lorsqu'il est allumé/éveillé, en plus il nécessite un mécanisme d'indexation (générateur d'indices) et il ne supporte pas le NC (No-Control).

II.5.2. Interfaces Cerveau-Ordinateur basées sur des protocoles au rythme autonome (Self-paced)

Egalement, appelées des BCI à protocole asynchrone. Dans ce type, les utilisateurs sont autorisés à intervenir à tout moment, c'est-à-dire, que la personne est autorisée à utiliser la BCI quand elle le souhaite [Daly J. J. et al., 2009]. Le système BCI est disponible en permanence pour l'utilisateur lorsqu'il est allumé/éveillé et il supporte le NC (*No-Control*).

La réalité virtuelle [Leeb R. et al., 2007 ; Scherer R. et al., 2008] est l'application la plus fréquente utilisant le protocole au rythme autonome, parce qu'elle exploite le paradigme d'imagination motrice basé sur les motifs cérébraux généré par le cortex moteur du lobe pariétal. En imaginant un sens d'un mouvement particulier, on pourra naviguer dans un environnement virtuel.

Aussi, un autre paradigme qui supporte ce protocole est appelé SSVEP, car il exploite les potentiels évoqués visuels à l'état d'équilibre (*Steady-State Visual Evoked Potentials*) qui sont détecté généralement dans le cortex occipital pour générer la commande. Le principe est basé sur la mise en place de différentes sources lumineuses clignotant à différentes fréquences. L'utilisateur se concentre sur l'une des sources, son activité EEG augmente d'une manière significative à la même fréquence de la source lumineuse cible, ce qui entrainera l'activation de la réponse associée à cette cible.

Le mode de fonctionnement de ces BCI est illustré dans la figure II.4 :

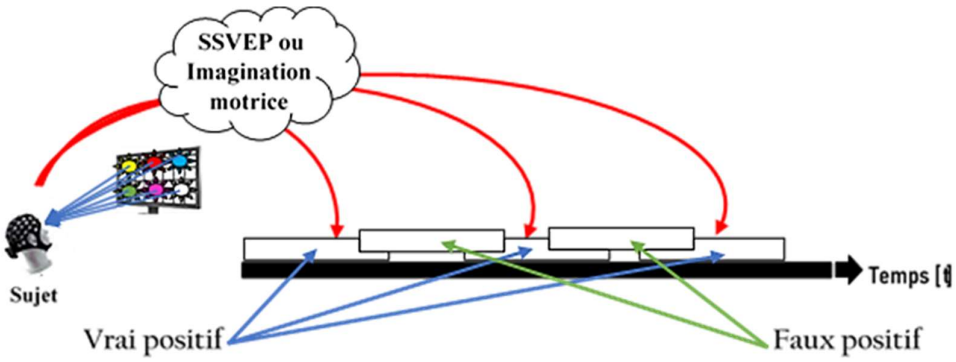


Figure II.4. Mode de fonctionnement des Interfaces Cerveau-Ordinateur basées sur le paradigme SSVEP.

Si l'utilisateur n'utilise pas la BCI, c'est-à-dire, il arrête sa concentration sur l'une des sources clignotantes, il n'y aura pas d'augmentation de rythme de type SSVEP, donc le système ne produit aucune réponse. C'est pour cette raison que le protocole au rythme autonome fasse la distinction entre les états sans contrôle et les états avec contrôle intentionnel.

Les protocoles au rythme autonome peuvent produire des interactions plus naturelles et dynamiques entre l'utilisateur et le système BCI, mais ils sont beaucoup plus difficiles à développer.

II.5.3. Interfaces Cerveau-Ordinateur basées sur des protocoles combinés synchrones et au rythme autonome

Dans ce type de protocoles, la BCI se met initialement en veille (c'est le mode d'attente inactif), ensuite l'utilisateur peut produire un signal de commande de type SSVEP ou par imagination motrice (IM) qui ne se produit pas au hasard, pour activer la BCI puis le mettre sous l'autorité du protocole synchrone afin de supporter le contrôle intentionnel, jusqu'à ce qu'il produise une autre activité IM suivant le protocole au rythme autonome pour remettre la BCI en mode de veille (figure II.5).

Pfurtscheller et al. 2010, ont développé une BCI qui permet aux utilisateurs d'activer/désactiver le système juste en imaginant un mouvement du pied gauche ou droit.

De ce fait, selon l'état des éléments de l'environnement virtuel, l'utilisateur peut effectuer plusieurs actions par un seul état mental (par ex. l'imagination du mouvement de la main droite ou gauche). De là est née une sorte de système qui intègre l'intelligence pour que les commandes soient orientées en fonction du contexte d'exploitation. [Allison B. et al., 2010b ; Millán J. D. R. t al., 2010]

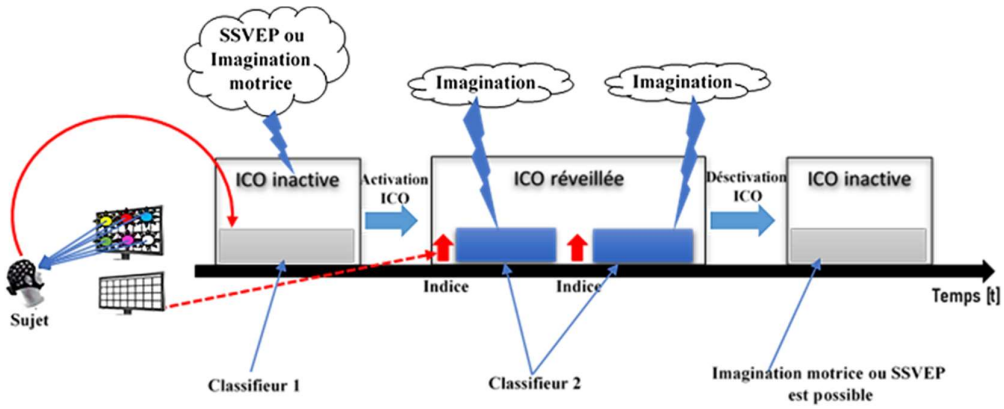


Figure II.5. Illustration du fonctionnement des interfaces cerveau-ordinateur combinant des protocoles synchrones et au rythme autonome.

II.5.4. Interfaces Cerveau-Ordinateur basées sur des protocoles non-adaptatifs

Généralement, avant d'utiliser une application basée sur un système BCI, on procède à l'étalonnage des paramètres initiaux de la BCI (paramètres des filtres spatiaux, poids des classifieurs, etc.) via une session dans laquelle l'utilisateur est invité à effectuer un ensemble des tâches mentales prédéfinies. La durée nécessaire pour effectuer le pré-étalonnage dépend fortement de la qualité des signaux acquis, de la précision espérée du classifieur, etc.

Les réponses produites sous forme des données étiquetées sont alors regroupées puis mises dans un pipeline d'apprentissage pour ajuster les paramètres de la BCI. En utilisant ces paramètres optimaux dans un scénario en ligne, l'activité cérébrale doit être interprétée correctement à la réponse attendue. [Pfurtscheller G., & Lazarou, I. et al., 2018 ; Müller K. R. et al. 2008]. Selon Lotte, F. et al., 2018, il est recommandé de réajuster les paramètres avant chaque session.

Ce protocole est compatible aux applications BCI limitées dans lesquelles un seul pré-étalonnage est suffisant [Nijboer F. et al. 2008]. Certaines systèmes BCI peuvent fonctionner pour une langue durée avec les mêmes paramètres [Sellers E. W., 2006] et parfois un réétalonnage périodique est nécessaire afin d'améliorer la précision du classifieur du système.

Cependant, au fur et à mesure que la recherche dans le domaine des BCI ce progressait, la nécessité d'éliminer les sessions de pré-étalonnage augmentait également. Pour que les utilisateurs de la BCI n'aient pas à subir le fastidieux processus d'étalonnage chaque fois qu'ils veulent utiliser une BCI, l'une des solutions à ce problème est l'application des protocoles adaptatifs qui proposent l'ajustement en ligne des paramètres de la BCI, par ex. l'apprentissage par transfert [Barachant A. et al., 2011 ; Barachant A., & Congedo M., 2014 ; Shenoy P. et al., 2006 ; Wan, Z. et al., 2021].

II.5.5. Interfaces cerveau-ordinateur basées sur des protocoles adaptatifs

Les BCIs basées sur des protocoles adaptatifs (également appelés BCI auto-étalonnées) intègrent des procédures automatiques de la mise à jour des paramètres pour que ce type des BCIs peuvent s'adapter aux changements spontanés et adaptatifs des signaux cérébraux de l'utilisateur, ils sont généralement préférables dans l'usage quotidien et multi-utilisateur. Elles deviendront plus courantes [Cecotti H., & Graser A., 2010 ; Lotte F. et al., 2007] à mesure que la recherche et le développement évolueront. Pendant cette période de transition, la plupart des protocoles continueront d'être partiellement auto-paramétrés en ce sens qu'ils sélectionnent et/ou

mettent à jour certains paramètres par eux-mêmes et reçoivent le reste par le personnel de support technique.

Des études ont été menées sur les méthodes d'étalonnage coadaptatif basé sur l'apprentissage automatique pour les interfaces cerveau-ordinateur afin d'éliminer les procédures de réétalonnage et pour offrir un schéma d'adaptation qui guide individuellement l'utilisateur [Vidaurre C. et al., 2011].

Présentons à titre d'exemple, une BCI appelée le P300 speller adaptative (pour épeler les lettre d'un texte) qui devrait effectuer périodiquement une brève session d'orthographe automatisée au cours de laquelle l'utilisateur reçoit un certain nombre de lettres cibles à sélectionner dans une matrice. L'analyse des réponses évoquées des lettres cibles et celles non cibles donnerait étalonnage en ligne des paramètres du système en vue d'interpréter correctement l'activité cérébrale de l'utilisateur. [Kaplan A. Y. et al., 2013, Fazel-Rezai R. et al., 2012]. Une autre étude propose une BCI basée sur le P300 et le rythme sensorimoteur SMR, les paramètres clés ont été mis à jour automatiquement et, en outre, certains paramètres ont été étalonnés selon les besoins de l'utilisateur [Kübler, A. et al., 2009].

II.6. Interfaces cerveau-ordinateur catégorisées selon le motif cérébral

D'une façon générale, une activité cérébrale est composée d'impulsions électriques qui provoquent l'activation des neurones cérébraux. Les BCIs qui sont catégorisées selon le motif cérébral s'appuient sur différents arrangements d'impulsions pour communiquer avec l'application spécifique. Un exemple des motifs cérébraux est celui des potentiels liés aux événements ERP P300 (Event-Related Potentials P300). Le P300 atteigne un pic de tension positif maximal environ 300 ms après le début du stimulus (par exemple, dans le cadre du paradigme dit "*oddball*").

Dans la littérature, plusieurs recherches [Donchin E. et al., 2000 ; Polich J., 2007 ; Sellers E. W., & Donchin E., 2006] considèrent que les ERP P300 nécessitent peu d'entraînement initial, ceci est un avantage considérable par rapport à d'autres types de motifs cérébraux. Les P300 peuvent être évoqués par des paradigmes visuels, auditifs, tactiles, et même olfactifs ou gustatifs. D'autre part, il existe un autre motif cérébral largement utilisé est le potentiel évoqué à l'état d'équilibre SSEP (*Steady-State Evoked Potential*), qui sont l'activité électrique du cerveau en réponse à la stimulation de voies nerveuses sensorielles spécifiques, par opposition aux potentiels spontanés.

Il existe trois types de motifs cérébraux SSEP :

1- Les potentiels évoqués auditifs à l'équilibre **SSAEPs** (*Steady-State Auditory Evoked Potentials*) généralement enregistrés au niveau du cuir chevelu mais provenant du tronc cérébral [Dobie R. A., & Wilson, M. J., 1998].

2- Les potentiels évoqués visuels à l'équilibre **SSVEP**s (*Steady-State Visually Evoked Potentials*), trouvés dans le lobe occipital du cerveau et provoqués par la concentration sur un modèle stable de stimuli visuels, comme le clignotement régulier d'une source lumineuse [Sakurada, T. et al., 2015].

3- Les potentiels évoqués somatosensoriels à l'équilibre **SSSEP**s (*Steady-State Somatosensory Evoked Potentials*), qui présentent une réponse électrophysiologique cérébrale sinusoïdale provoquée par une stimulation vibro-tactile mécanique délivrée sur la peau glabre (par ex, bout du doigt). Ce motif est modulé par l'attention spatiale sélective [Vialatte, F. B. et al., 2010].

Parmi les nombreux SSEPs, les BCI basées sur le SSVEP qui utilise de source lumineuse scintillante de différentes fréquences provenant d'écrans LED ou d'écrans à cristaux liquides LCD (*Liquid Crystal Display*) est bien étudié car il pourra fournir le chemin de communication le plus rapide et le plus fiable [Volosyak I. 2011 ; Wang M. et al. 2018 ; Panicker R. C. et al., 2011] ; Par exemple, lorsque l'utilisateur se concentre sur un stimulus cible avec une certaine fréquence, cette même fréquence est synchronisée à une certaine zone selon la modalité (par exemple, le cortex visuel pour les stimuli visuels), et l'amplitude à la fréquence du stimulus cible est supérieure à celles des stimuli non cibles.

Le système BCI utilise ces motifs cérébraux distincts pour effectuer différentes actions (par exemple, directions, marche/arrêt et frappe) en fonction de l'attention sélective de l'utilisateur [Müller-Putz, G. R., & Pfurtscheller, G., 2007 ; Wang, H. et al., 2019 ; Zhang, Y. et al., 2012]. Une méthode de signal cérébral moins utilisée est celle des potentiels corticaux lents SCPs (*Slow Cortical Potentials*), qui sont des décalages de l'activité électrique corticale d'une durée de plusieurs centaines de millisecondes à plusieurs secondes [Gevensleben H. et al., 2009]. Enfin, il y a le rythme sensorimoteur SMR (*Sensori-Motor Rhythm*), qui implique la désynchronisation/synchronisation liée aux événements (ERD/ERS), qui est une activité cérébrale associée à l'imagination motrice.

II.7. Interfaces cerveau-ordinateur catégorisées suivant le mode d'acquisition

Il existe trois modes utilisés dans l'acquisition de l'activité cérébrale :

II.7.1. Mode d'acquisition non-invasif

Ce mode consiste à placer des capteurs à des endroits précis du cuir chevelu du crâne (EEG, etc.), ou à entourer tout le volume de la tête avec un dispositif de capture, ce dernier couvrant tous les points de la surface du crâne (Magnétoencéphalographie - MEG, l'Imagerie par résonnance magnétique - IRMf, Spectroscopie proche de l'infrarouge, etc.). Ce mode ne nécessite pas de chirurgie, d'implantation de produits chimiques ou de machines internes, ni d'insertion d'aiguille pour recevoir et enregistrer l'activité neuronale [Veena, N. et al., 2020].



Dans cette technique, des capteurs sont installés sur ou très près de la tête afin de mesurer les signaux du cerveau. La chirurgie n'est pas nécessaire dans cette technique et elle est très simple à mettre en œuvre sur les patients. Les EEG et les MEG sont utilisés dans la BCI non invasive.

- Les mesures directes qui détectent l'activité électrique (par exemple, EEG) ou magnétique (par exemple, MEG) du cerveau.
- Les mesures indirectes de la fonction cérébrale reflétant le métabolisme ou l'hémodynamique du cerveau par exemple : l'IRMf, la spectroscopie proche de l'infrarouge et la tomographie par émission de positrons) qui ne caractérisent pas directement l'activité neuronale. Contrairement aux méthodes d'enregistrement invasives, cette technique est connue pour sa simplicité, son moindre coût, son innocuité, et elle est particulièrement appréciée dans le domaine des interfaces cerveau-ordinateur [Ayaz, H. et al., 2020].

II.7.1.1. Électroencéphalographie - EEG

La conception et l'implémentation de l'interface cerveau-ordinateur présenté dans ce manuscrit est basée sur la technique d'enregistrement EEG.

L'électroencéphalographie est une technique non-invasive pour l'exploration de l'activité cérébrale. Elle est fondamentalement effectuée en plaçant des électrodes spécifiques (humides ou sèches, actives ou passives) sur le cuir chevelu sur toute la surface crânienne du cortex cérébral selon une norme standardisée appelée le Système International 10-20 agrée et adoptée

par la Fédération internationale de neurophysiologie clinique IFCN (*International Federation of Clinical Neurophysiology*) [Jasper H. H., 1958], ce standard est illustré par la figure II.6.

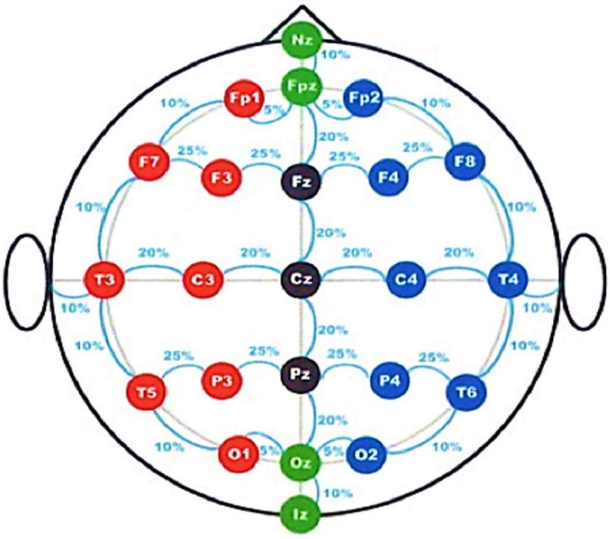


Figure II.6. Le Système International 10-20 pour le positionnement des électrodes EEG.

La résolution spatiale est limitée et la durée d'enregistrement ne dépasse guère la journée. Toutefois ce système est peu cher, facile d'utilisation et permet d'envisager de nombreuses applications, y compris pour le grand public. De fait, c'est aujourd'hui le mode d'enregistrement le plus utilisé.

Aujourd'hui cette technique devient un outil très utile dans plusieurs domaines tels que :

- La pratique neurologique ou psychiatrique [Hochberg L. R., 2012].
- La recherche en neurosciences cognitives [Laufs H., 2012].
- La réparation des fonctions neuronales du cerveau [da Silva F. L., 2013].
- La mise en place d'interfaces cerveau-ordinateur [Schalk, G., & Mellinger, J., 2010].

Chronologie historique

- À la fin des années 1770, Galvani s'est intéressé à l'influence de l'électricité sur les actions musculaires des animaux. En 1786, il a finalement élaboré la théorie de l'électricité animale [Piccolino M., 1997].
- Le physiologiste britannique Richard Caton a démontré [da Silva F. L., 2013], en 1875, que l'activité fonctionnelle correspond à l'apparition d'une activité électrique dans une zone circonscrite du cortex cérébral.

- 49 ans plus tard, le neurologue allemand Hans Berger a jeté les bases d'un domaine totalement nouveau en appliquant la technique EEG à un être humain et en enregistrant pour la première fois l'activité cérébrale représentée sur la figure II.7 [Berger H., 1929]. Il a ainsi réussi à caractériser les rythmes alpha et bêta [Haas, L. F., 2003], il est considéré comme l'inventeur de l'électroencéphalographie, et il a été le premier à proposer ce terme.

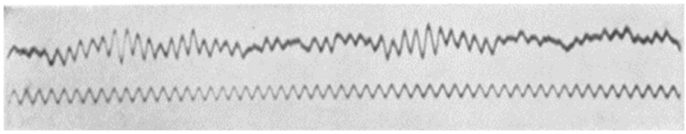


Figure II.7. Le premier EEG enregistré par Hans Berger en 1924 et publié en 1929.
D'après [H. Berger, 1929].

- L'ingénieur en électronique J. F. Toennies (1902-1970), en collaboration avec l'Institut Kaiser Wilhelm des sciences du cerveau de Berlin (*Hirnforschungs-Institut*), sous la direction du neuroanatomiste Oskar Vogt, a réussi à concevoir le NEUROGRAPHE [Eccles J. C., 2013], un appareil considéré comme très perfectionné par rapport à ceux utilisés à cette époque. Puis en 1936, il achève la mise en œuvre d'un amplificateur différentiel biologique [Schomer, D. L. & Da Silva, F. L., 2012].
- En 1934, en Grande-Bretagne, le neurophysiologiste Lord Adrian (1889-1977) avec son collaborateur Matthews confirme et démontre les observations de Hans Berger dans sa publication [E. D. Adrian et al., 1934]. De même, W. Gray Walter (1910-1977) a découvert à la même époque (en 1936) un foyer de rythmes cérébraux lents appelés ondes Delta et Thêta [Schomer, D. L. & Da Silva, F. L., 2012].
- Aux États-Unis, le Dr. Gibbs (1903-1992) a été invité par Albert Grass (*Massachusetts Institute of Technology*) à concevoir un amplificateur biologique à impression d'encre à 3 canaux.
- Après la Seconde Guerre mondiale, la technique de l'EEG a connu un grand essor avec l'ouverture de plusieurs laboratoires spécialisés en EEG, notamment aux États-Unis, en Grande-Bretagne, au Canada et en France, alors que l'activité en Allemagne a sensiblement diminué.
- L'implantation intracrânienne des électrodes a été introduite pour la première fois chez l'homme par les travaux de Meyers et Hayne en 1948 et de Knott et al., en 1950 [Schomer, D. L. & Da Silva, F. L., 2012], ce qui a ouvert les perspectives sur les études cliniques et le traitement de l'épilepsie.

- En 1947, Dawson a été le premier à décrire le potentiel évoqué de la stimulation électrique du nerf cubital [Dawson G. D., 1954].
- En 1958, Van der Tweel a introduit la technique SML (*Sine wave Modulated Light*) [Kamp A. et al., 1960].
- Dans les années 1970, les travaux de F.H. Lopes da Silva et de ses collègues ont montré que le rythme alpha est généré principalement par une couche dipolaire dans le cortex visuel [Da Silva F. L., & Van Leeuwen W. S. 1977].

II.7.1.2. Magnétoencéphalographie - MEG

La MEG est une autre technique de neuro-imagerie fonctionnelle permettant d'enregistrer d'une manière non invasive les variations faibles (de l'ordre de quelques femtoteslas ~10^{-15} Tesla) du champ magnétique à la surface crânienne généré par les courant électriques associés à l'activité cérébral d'un être vivant. Le scanner MEG est équipé des magnétomètres SQUID (*Superconducting Quantum Interference Device*) inventé en 1968 par David Cohen [Cohen D., 1972], ou des magnétomètres SERF (*Spin Exchange Relaxation-Free*) [Hämäläinen M. et al., 1993 ; Boto E. et al., 2018]. Ces magnétomètres sont conçus d'être très sensibles pour détecter des minuscules champs magnétiques induits par l'activité électriques des neurones pyramidaux actifs [Hämäläinen M. et al., 1993]. La détection de la variation du champ magnétique est seulement possible lors de l'activation synchrone de quelques milliers de neurones, et cela dû au fait que le champ induit par un seul neurone ou d'un nombre limité de neurones est extrêmement faible et donc ne peut pas atteindre le seuil de détection de l'appareil MEG.

En vue d'obtenir une cartographie claire, la technique MEG nécessite l'introduction d'une IRM pour combiner les informations fonctionnelles de la MEG avec les images anatomo-structurelles provenant de l'appareil IRM.

Les principaux avantages de la MEG sont les suivants :

- La MEG est facile à utiliser, du fait qu'il ne nécessite pas le placement et le collage des électrodes.
- Une très haute résolution temporelle de l'ordre de la milliseconde.
- Avec un nombre de 256 à 512 canaux SQUID, on pourra avoir une excellente résolution spatiale avec une précision millimétrique.
- La MEG non invasive qui ne nécessite pas l'injection d'isotopes ou l'exposition à des rayons X ou à des champs magnétiques [Tan H. R. et al., 2015].

- Une mesure directe des fonctions cérébrales.
- Un rapport signal-bruit SNR (*Signal-Noise Ratio*) est élevé par rapport à l'EEG.

Les principaux inconvénients de la MEG sont les suivants :

- Dans un contexte BCI où la portabilité et la facilité d'utilisation sont une exigence, le principal inconvénient de la MEG est son coût prohibitif et sa taille encombrante.
- Les protocoles expérimentaux doivent être effectués dans une pièce blindée magnétiquement en raison de la sensibilité du dispositif MEG aux champs magnétiques environnants.

Les applications de la MEG comprennent :

- La recherche fondamentale sur les processus cérébraux perceptifs et cognitifs.
- La réalisation des BCI basées sur l'imagination motrice [Mellinger, J. et al. 2004].
- La localisation des régions affectées par une pathologie avant une ablation chirurgicale.
- La détermination de la fonction de diverses parties du cerveau et le neurofeedback.
- Elle peut être appliquée dans un cadre clinique pour trouver les emplacements des anomalies ainsi que dans un cadre expérimental pour simplement mesurer l'activité cérébrale.

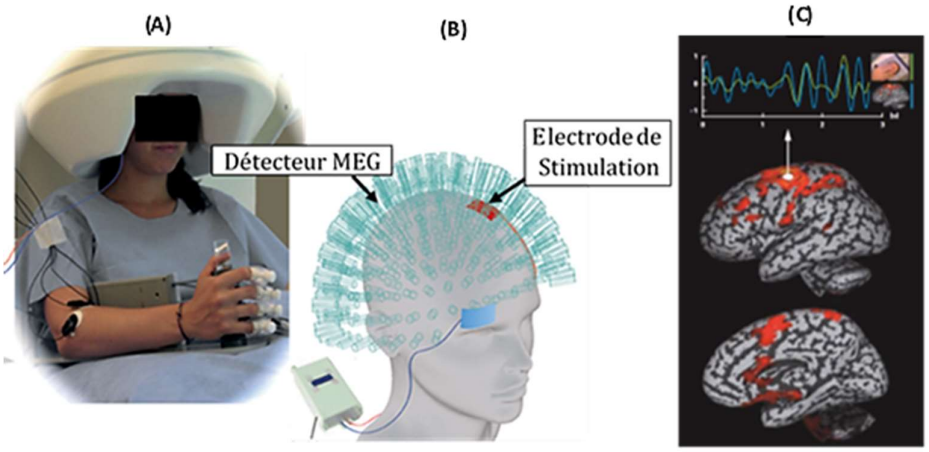


Figure II.8. (A) Dispositif MEG (B) illustration du détecteur MEG et des électrodes de stimulation (C) Enregistrement sur la carte cérébrale des zones d'activation.

II.7.1.3. Spectroscopie Proche de l'Infrarouge fonctionnelle – fNIRS

Le terme fNIRS signifie "*functional Near-Infrared Spectroscopy*", est un moyen de d'évaluer d'une façon indirecte l'activité cérébrale en enregistrant la saturation en oxygène de l'hémoglobine, ceci est effectué en mesurant la variation dans l'oxyhémoglobine et la désoxyhémoglobine, qui reflètent l'activité cérébrale locale.

La lettre « *f* » dans le terme fNIRS désigne le concept 'fonctionnelle' qui vient du fait que cette technique est capable d'évaluer l'activité cérébrale. Les dispositifs fNIRS sont aussi appelés les oxymètres cérébraux - SRCO2, le terme qui signifie la Saturation Régionale Cérébrale en Oxygène.

Le principe est basé sur la mesure des changements d'action d'un faible faisceau de lumière non-ionisant [Ferrari M. et al., 2004] avec une longueur d'onde entre 650 nm et 850 nm [Bittante G. et al, 2022]. Ce dernier lorsqu'il traversera le tissu cortical, une partie de la lumière sera absorbée par celui-ci et une autre partie sera réfléchie vers les photodétecteurs placés sur la tête du sujet. En comparant l'intensité de la lumière avant et après son passage dans le cortex cérébral, il sera possible de déterminer la capacité d'absorption de la lumière par les molécules d'hémoglobines contenues dans le cortex cérébral et donc calculer la teneur en oxygène de la zone cérébrale explorée.

Ensuite, ces signaux sont transportés via des fibres optiques flexibles vers un système d'acquisition et de conditionnement afin de cartographier les informations fonctionnelles cérébrales. En raison de la nature latente de la réponse du flux sanguin aux sites activés par les neurones dans le cortex cérébral, la résolution temporelle est plus lente dans le fNIRS que dans l'EEG. En outre, les signaux fNIRS sont vulnérables aux artefacts de mouvement et d'impulsion causés par les mouvements physiques et les battements de cœur pendant la mesure.

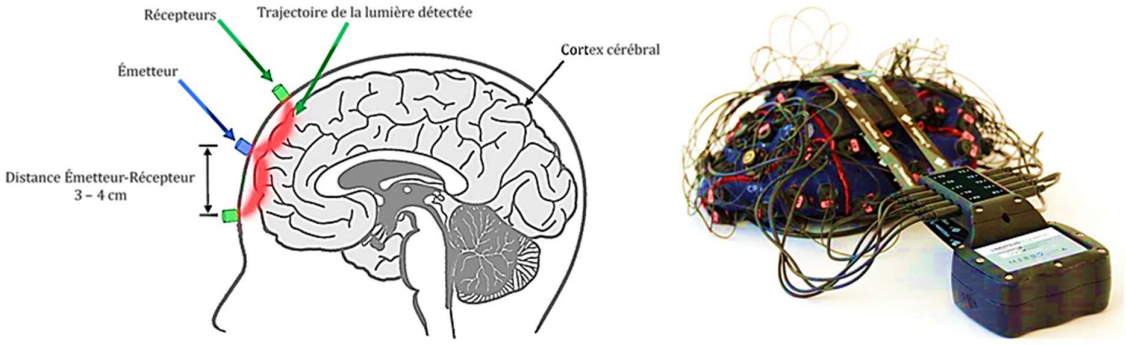


Figure II.9. (A) Principe de Spectroscopie Proche de l'Infrarouge fonctionnelle. (B) Illustration du dispositif fNIRS.

II.7.1.4. Imagerie par Résonance Magnétique fonctionnelle - IRMf

L'imagerie par résonance magnétique fonctionnelle ou IRM fonctionnelle (IRMf) est une technique de neuro-imagerie fonctionnelle apparue dans les années 1990 [Ogawa S. et al., 1992] et utilisée pour l'analyse du fonctionnement du cerveau. Elle consiste à mesurer indirectement l'activité neuronale du cerveau en déterminant la réponse hémodynamique (variation locale des caractéristiques du débit sanguin) des aires corticales actives. Le déclenchement de l'activité électrique des cellules nerveuses entraîne une consommation supplémentaire d'oxygène transporté par l'hémoglobine dans les globules rouges, transformant ainsi cette dernière en désoxyhémoglobine locale, ce mécanisme évoque une augmentation d'afflux en oxygène dans les régions cérébrales activées afin de compenser les besoins énergétiques en oxygène et en glucose. La désoxyhémoglobine reflète la transmission des champs magnétiques intenses qui interfèrent fortement avec le champ magnétique de l'IRMf, mais d'autre part, l'oxyhémoglobine fait transmettre des champs magnétiques moins intenses, ce qui réduit les interférences avec le champ IRM.

Cette variation locale et transitoire de l'oxygène transporté par l'hémoglobine dans les globules rouges des zones corticales activées est connue sous le nom de l'effet BOLD (*Blood Oxygen Level Dependant*), qui est basé sur le soi-disant couplage neurovasculaire, créant une corrélation entre le déclenchement neuronal et le taux métabolique cérébral de l'oxygène [Dunn, A. K. et al., 2005]. Finalement l'IRMf permet de d'exploiter le signal BOLD pour l'imagerie et rend les régions corticales actives visibles. [Weiskopf N. et al., 2007].

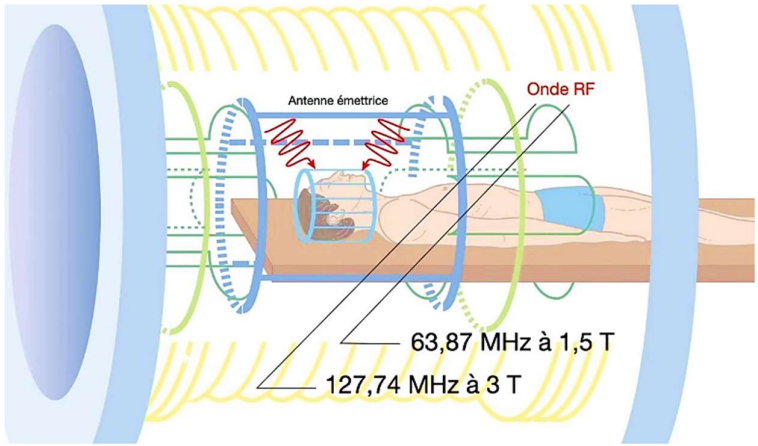


Figure II.10. Principe de fonctionnement d'un IRM.

En terme des avantages, l'IRMf se caractérise par son innocuité, sa courte durée de préparation du sujet, sa résolution temporelle de l'ordre de 40 ms (excellente par rapport celle de la Tomographie par Emission de Positrons [Lindquist M. A. et al, 2009], mais lente par rapport

l'EEG et le MEG [Mulert C., et al., 2004] et sa bonne résolution spatiale de l'ordre du millimètre [Olman C. A., 2012], d'où il est possible d'identifier les aires cérébrales impliquées dans une tâche mentale particulière [Buckner R. L. et al., 2008] ce qui pourra être utile pour les applications des BCIs.

En outre, et contrairement à l'EEG, cette technique permet de mesurer facilement l'activité cérébrale dans les couches corticales sous-jacentes situées en profondeur difficilement mesurables. Cette méthode offre la possibilité d'acquérir plus de 4 images anatomo-fonctionnelles par secondes, permettant le suivi de l'activité neuronale durant des tâches complexes.

D'autre côté, l'un des inconvénients de l'IRMf est qu'elle ne mesure pas directement l'activité neuronale, mais permet de déduire cette activité à partir du volume et du flux sanguin mesurés, et ceci ne permet pas de distinguer les activités neuronales excitatrices de celles inhibitrices. La latence lente de la réponse hémodynamique de l'ordre de 4 à 7 secondes (la durée entre l'activation neuronale et la modification des concentrations en désoxyhémoglobine) ajoute un autre empêchement pour l'utilisation de l'IRMf dans des applications BCI temps-réel. Finalement, la taille et le coût d'utilisation rend l'IRMf inefficace et peu pratique pour les usages quotidiens.

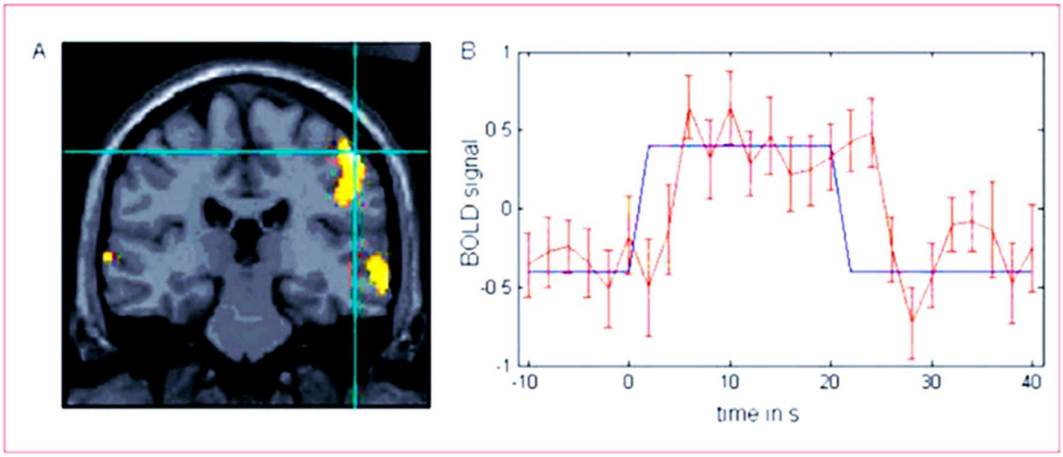


Figure II.11. (A) Image d'activation obtenue par imagerie par résonance magnétique fonctionnelle (IRMf) lors de la stimulation électrique (4 Hz) du doigt de la main gauche. Chaque stimulation dure 20 secondes, alternant avec des périodes de repos. (B) Le signal BOLD présente des modifications toutes les 20 secondes, correspondant à la stimulation, avec un retour rapide à l'état de repos lorsque la stimulation est arrêtée. D'après [Hutton C., et al., 2011]

II.7.1.5. Tomographie par Emission de Positrons - TEP

La TEP est une technique d'imagerie non invasive chirurgicale [Coyle, J. T. et al., 1983 ; Krepki R. et al., 2007 ; Fox, P. T. et al., 1989], qui permet de mesurer, en 3 dimensions par médecine nucléaire ou par rayonnement, l'activité métabolique cérébrale dans le contexte des processus fonctionnels d'un organe humain [Herholz, K., & Heiss, W. D., 2004].

La technique est basée sur scintigraphie qui nécessite l'injection préalable d'un atome traceur émetteur de positrons dans la circulation sanguine (produit radioactif) dont l'annihilation produit elle-même deux photons. Le lieu d'émission et la concentration de ces derniers est détecté par une caméra TEP. Le radiotraceur administré est le Fluo-Déoxy-Glucose (FDG) marqué au fluor-18 qui est consommé de façon physiologique par les cellules cérébrales. Ce traceur radioactif va se concentrer dans des régions ayant un besoin métabolique augmenté dû à leur activation.

Par conséquent, sur la base des positrons émis on pourra alors capturer le tracé spatial de l'activité cérébrale et cela permet à reconstruire une image 3D des aires cérébrales les plus actives et qui montrent une croissance dans leur besoin métabolique [Townsend D. W., 2008].

L'utilisation de la TEP dans le domaine des BCI reste encore limité du fait de son coût élevé, sujette à controverse en raison du processus d'injection du radiotraceur, et la demi-vie des radionucléides [Simanovics, J. et al., 2023 ; Boecker H. et al. 1999 ; Martini M. L. et al., 2020].

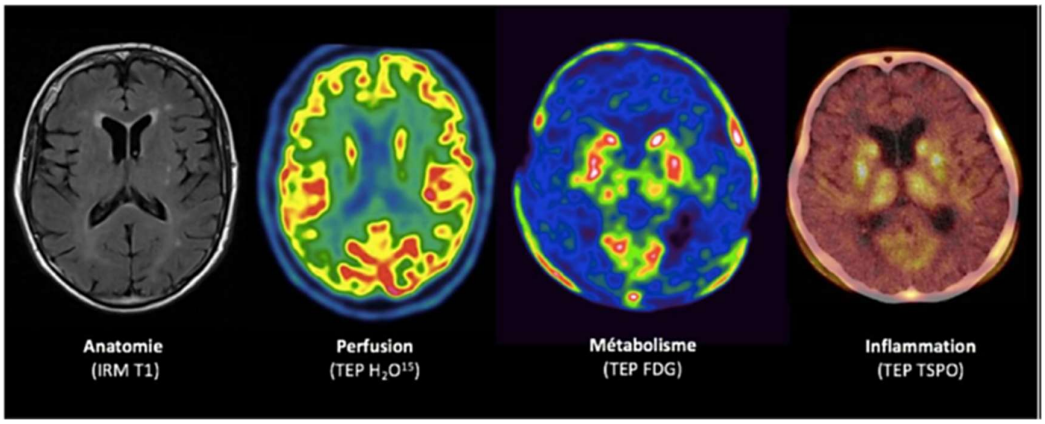
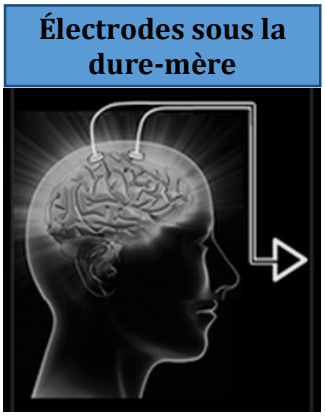


Figure II.12. Étude du cerveau humain en imagerie multimodale, utilisant la TEP pour évaluer le débit sanguin cérébral.

II.7.2. Mode d'acquisition semi-invasif



Ce mode d'acquisition repose sur l'idée que les électrodes positionnées ne pénètrent pas le cerveau c'est-à-dire elles sont implémentées à l'intérieur du crâne mais en dehors de la matière grise et par conséquent les risques de complication et de formation de tissu cicatriciel sont moindres et des applications médicales sont rapidement envisageables, alors il s'agit d'une matrice d'électrodes très minuscules (de quelques millimètres de diamètre) placée en sous-dural ou en épidural (sous ou à la surface de la dure-mère), sachant que la dure-mère est la membrane fibreuse qui entoure le cerveau juste sous la boîte crânienne.

L'électrocorticographie (ECoG) est souvent la technique la plus utilisée dans ce mode d'acquisition.

Les avantages de la technique BCI partiellement invasive comprennent une faible complexité, un faible risque clinique, une stabilité supérieure à long terme, etc.

II.7.2.1. Électrocorticographie - ECoG

L'ECoG est une technique très proche de l'EEG qui permet d'explorer l'activité cérébrale plus locale comme illustrée par la figure II.13, cette technique consiste à effectuer une craniotomie qui permet d'exposer une partie du cerveau afin de placer une grille d'électrodes corticales planaires soit à la surface ou en dessous de la dure-mère pour enregistrer seulement des potentiels de champs locaux.

L'ECoG mesure l'activité cérébrale avec des résolutions spatiales et fréquentielles (de 1 à 300 Hz) plus élevées que l'EEG.

Les avantages :

- Moins invasive que les grilles de micro-électrodes pénétrantes MEA (*Micro-Electrode Arrays*), donc moins d'effets secondaires que les MEA.
- Elle fournit une excellente qualité des signaux mesurés par rapport à l'EEG.

Pour relever le défi pour éviter l'utilisation d'une fil conducteur transcutané, qui est à l'origine des infections locales ; l'équipe de recherche biomédicale française CLINATEC-CEA-LETI [Chavret G. et al., 2011 ; 2012] a mené un projet visant à développer une neuroprothèse, basée sur la mesure l'ECoG, afin de restituer les capacités motrices des patients tétraplégiques. L'acquisition de l'activité neuronale s'est effectuée à l'aide du dispositif

Wimagine® entièrement implanté d'une manière semi-invasive à l'intérieur de la boite crânienne qui est composé de 64 électrodes placées en épidural (à la surface de la dure-mère). Cette plateforme intègre un module émetteur/récepteur sans fil dans la bande 402-405 MHz, avec une consommation d'énergie ultra-faible de 5mA (pendant la transmission RX/TX), l'alimentation (jusqu'à 100 mW) est fournie sans fil à la carte *Wimagine*® via une liaison inductive à une fréquence de 13.56 MHz [Charvet, G. et al., 2011]. Ce system a permis, dorénavant, de réaliser des enregistrements de haute qualité en offrant une meilleure résolution spatiale et un rapport signal/bruit SNR plus élevé que les EEG classiques. Cette avancée technologique est destinée pour concevoir des BCI implantables au profit des patients nécessitant une surveillance chronique tout en garantissant leur sécurité.

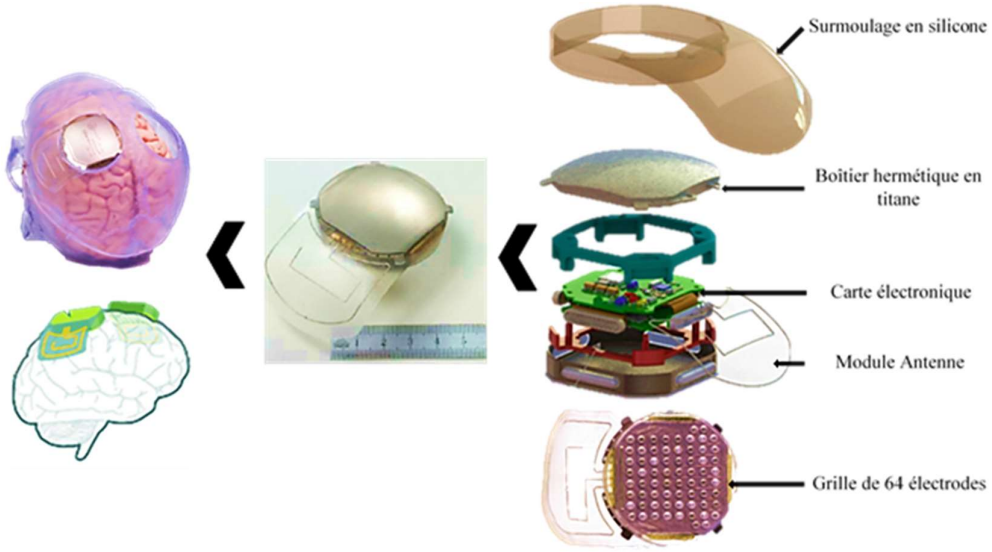


Figure II.13. Structure du dispositif implantable Wimagine.

Mestais, C. S. et al., 2014 ont publié une étude des performances fonctionnelles la biocompatibilité à long terme du dispositif *Wimagine*®.

Cependant, d'autres travaux de recherche ont été effectués en utilisant les signaux ECoG en terme d'analyse des fonctions motrices et somatosensorielles de d'un sujet humain [Ryun, S. et al., 2018], l'estimation de la trajectoire d'un mouvement 3D. La classification des signaux ECoG avant le début d'un mouvement volontaire [Lee S. H., 2013], Décodage de l'activation des doigts à partir de données ECoG [Jubien G. et al., 2019], Eliseyev A. et al. 2014 ont publié des résultats précliniques d'une plateforme BCI destinée au sujets tétraplégiques basée sur le dispositif ECoG WIMAGINE®.

Dougherty M. E. et al., 2019 ont développé une nouvelle approche d'enregistrement simultané des potentiels électriques de surface corticale µECoG (CSEP, *Cortical Surface Electrical Potentials*) et l'activité multi-unitaire laminaire LMUA (*Laminar Multi-Unit Activity*) et ont conclu que la couche corticale III est un contributeur majeur à la surface Hγ (*High gamma*).

Desai S. A. et al., 2019 ont présenté une approche d'apprentissage en utilisant le système NeuroPace® RNS®, avec laquelle ils arrivent de différentier les patients qui répondent aux traitements d'épilepsie avec une précision de classification qui atteint 76%. Récemment, Yao L. et al., 2022 ont présenté une classification améliorée des mouvements individuels des doigts en basant sur l'ECoG, dans laquelle ils ont obtenu une précision moyenne de 71.3% ± 7.1%.

II.7.3. Mode d'acquisition invasif

Le terme "invasif" vient du fait que le dispositif d'enregistrement peut être envoyé profondément dans le cerveau et y persiste pendant une longue période [Waldert, S., 2016].



Dans cette technique, des électrodes ou des capteurs sont implantés à l'intérieur du crâne après une intervention chirurgicale pour explorer l'activité neuronale. Cette technique fournit des résultats de haute qualité car les capteurs se trouvent dans la matière grise.

Les inconvénients de cette technique :

- L'accumulation de tissu cicatriciel dans le cortex cérébral ou l'encapsulation fibreuse autour des électrodes que le corps perçoit comme des intrus.
- En raison de certaines conditions médicales, le corps n'accepte pas les électrodes ou les capteurs.

La technique la plus connue dans ce mode d'acquisition est bien adoptée : enregistrement de neurones intra corticaux INR (*Intracortical Neuron Recording*).

II.7.3.1. Enregistrement Neuronal Intracortical - INR

L'INR (*Intracortical Neuron Recording*) est une technique d'enregistrement avec une grande précision de l'activité cérébrale unitaire c'est-à-dire l'activité d'une seule unité SUA (*Single Unit Activity*) ou multi-unitaire MEA (*Multi-Unit Activity*) qui consiste à placer une

micro-électrode ou une grille de micro-électrodes pénétrantes (en tungstène ou en platine) à l'intérieur des corps cellulaires des neurones (figure II.14).

Une autre approche pour capturer l'activité cérébrale consiste à des micro-électrodes aux proximités corps cellulaires des neurones pour enregistrer les potentiels des champs locaux, cette approche est baptisée LFP (*Local Field Potentials*) [Perge, J. A. et al., 2014].

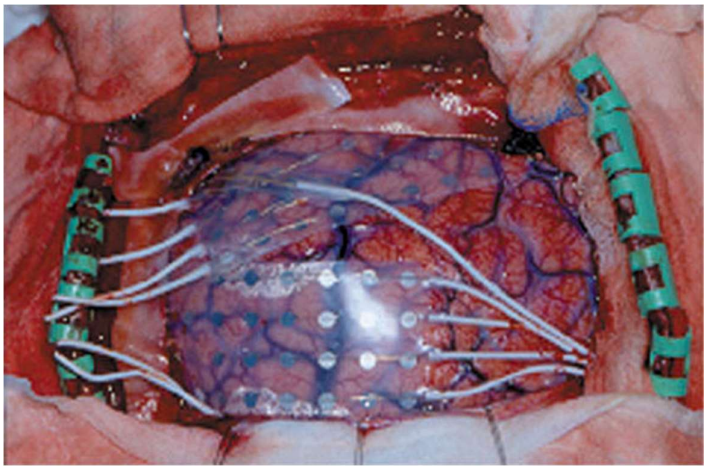


Figure II.14. Exemple d'un acte chirurgical pour implanter une interface MUA d'une BCI de type INR.
(*Illustration tirée de l'article en ligne : https://tpemainbionique.weebly.com/le-bci.html*)

Cette technique offre une excellente résolution spatiale, qu'il s'agisse avec l'utilisation de l'interface SUA, MUA ou avec LFP, tandis que la résolution temporelle est similaire à celle de l'ECoG.

Le mode invasif est encore peu utilisé, parce qu'il comporte toujours des risques de complications, de rejet du corps étranger, de la désorientation profonde, des lésions tissulaires et la dégradation de la qualité du signal au fil du temps ; il n'a été jusqu'à ce jour testé qu'aux sujets volontaires [Gunasekera B. et al. 2015] ou des patients souffrant de troubles neuro-moteurs graves [Jarosiewicz, B. et al., 2015].

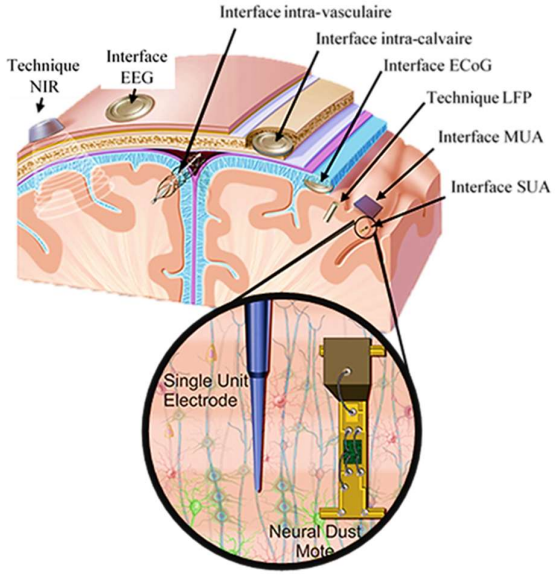


Figure II.15. Positionnement des électrodes suivant les différentes techniques d'acquisition BCI (Invasives, semi-invasives et non-invasives).

II.8. Interfaces Cerveau-ordinateur catégorisées suivant la stratégie de fonctionnement

Une autre façon de catégoriser les BCI est leur stratégie de fonctionnement. Il s'agit de la façon dont une BCI suscite des signaux cérébraux pendant son utilisation.

- Une BCI qui utilise l'attention sélective présente aux utilisateurs une stimulation auditive, visuelle ou tactile qui peut susciter des réponses aux signaux cérébraux [Allison, B. Z. et al., 2010a].
- Les efforts cognitifs, une autre stratégie de fonctionnement, s'appuient sur le biofeedback pour entraîner les utilisateurs à maintenir un niveau souhaitable d'amplitude de fréquence des ondes cérébrales [Chatterjee, A. et al., 2007].
- Certaines BCI utilisent la stratégie de fonctionnement unique de l'imagerie motrice, qui permet à l'utilisateur d'imaginer des mouvements musculaires, afin de provoquer un pic d'activité neuronale dans le cortex moteur [Pfurtscheller G. & Neuper C., 2006].

II.9. Conclusion

Dans ce chapitre, nous avons exploré les interfaces cerveau-ordinateur (BCI), qui permettent la communication directe entre l'activité électrique du cerveau et des dispositifs externes tels que les ordinateurs ou les organes prothétiques. Les BCI ont connu un développement significatif au fil des années et ont trouvé des applications prometteuses dans divers domaines, notamment l'assistance thérapeutique, la rééducation fonctionnelle, la robotique collaborative, la réalité virtuelle, le divertissement, etc.

La conception d'un système BCI est complexe et nécessite une expertise multidisciplinaire en génie électrique, programmation, apprentissage automatique et neurosciences. La première étape du processus BCI est l'acquisition des signaux EEG, qui enregistrent l'activité cérébrale du sujet. L'EEG est la méthode la plus couramment utilisée pour cette étape. Les signaux EEG subissent ensuite un prétraitement, qui comprend des étapes telles que le filtrage pour supprimer les composantes indésirables et la sélection des canaux pertinents. Le post-traitement des signaux EEG est une étape cruciale où des caractéristiques fréquentielles, spatiales, temporelles et statistiques sont extraites pour améliorer la performance des classifieurs.

Les BCIs sont catégorisées en fonction de plusieurs critères tels que le motif cérébral utilisé, le type de stimulus appliqué, le protocole de fonctionnement et le mode d'enregistrement utilisé. Différents motifs cérébraux peuvent être utilisés pour communiquer avec une BCI, tels que l’imagination motrice, les potentiels évoqués P300, les réponses de synchronisation visuelle SSVEP, etc.

En fonction du protocole de fonctionnement, les BCIs peuvent être classées en cinq catégories principales : synchrones (*cue-based*), autonomes (*self-paced*), combinant synchrones et autonomes, adaptatives et non adaptatives. Chaque catégorie présente des avantages et des limitations spécifiques.

En conclusion, les interfaces cerveau-ordinateur sont des outils fascinants qui ouvrent de nouvelles perspectives pour l'interaction homme-machine. Elles offrent des solutions potentielles pour la réhabilitation fonctionnelle, la robotique, la réalité virtuelle et d'autres domaines.

BIBLIOGRAPHIE DU CHAPITRE II :

Aggarwal, S., & Chugh, N. (2019). Signal processing techniques for motor imagery brain computer interface: A review. *Array*, *1*, 100003.

Ahmadi, A., Dehzangi, O., & Jafari, R. (2012, May). Brain-computer interface signal processing algorithms: A computational cost vs. accuracy analysis for wearable computers. In *2012 Ninth International Conference on Wearable and Implantable Body Sensor Networks* (pp. 40-45). IEEE.

Allison, B., Luth, T., Valbuena, D., Teymourian, A., Volosyak, I., & Graser, A. (2010). BCI demographics: How many (and what kinds of) people can use an SSVEP BCI? *IEEE transactions on neural systems and rehabilitation engineering*, *18*(2), 107-116.

Allison, B. Z., Brunner, C., Kaiser, V., Müller-Putz, G. R., Neuper, C., & Pfurtscheller, G. (2010). Toward a hybrid brain–computer interface based on imagined movement and visual attention. *Journal of neural engineering*, *7*(2), 026007.

Ang, K. K., Chin, Z. Y., Wang, C., Guan, C., & Zhang, H. (2012). Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Frontiers in neuroscience*, *6*, 39.

Ayaz, H., Izzetoglu, M., Bunce, S., Heiman-Patterson, T., & Onaral, B. (2007, May). Detecting cognitive activity related hemodynamic signal for brain computer interface using functional near infrared spectroscopy. In *2007 3rd international IEEE/EMBS conference on neural engineering* (pp. 342-345). IEEE.

Baillet, S. (2017). Magnetoencephalography for brain electrophysiology and imaging. Nature Neuroscience, 20(3), 327-339.

Barachant, A., Bonnet, S., Congedo, M., & Jutten, C. (2011). Multiclass brain–computer interface classification by Riemannian geometry. *IEEE Transactions on Biomedical Engineering*, *59*(4), 920-928.

Barachant, A., & Congedo, M. (2014). A plug&play P300 BCI using information geometry. *arXiv preprint arXiv:1409.0107*.

H. Berger (1929). Über das Elektroenkephalogramm des Menschen. Archiv für Psychiatrie und Nervenkrankheiten, 87(1), 527-570.

Birbaumer, N., & Cohen, L. G. (2007). Brain–computer interfaces: communication and restoration of movement in paralysis. *The Journal of physiology*, *579*(3), 621-636.

Bittante, G., Patel, N., Cecchinato, A., & Berzaghi, P. (2022). Invited review: A comprehensive review of visible and near-infrared spectroscopy for predicting the chemical composition of cheese. *Journal of Dairy Science*.

Blankertz, B., Curio, G., & Müller, K. R. (2001). Classifying single trial EEG: Towards brain computer interfacing. *Advances in neural information processing systems*, *14*.

Boecker, H., Ceballos-Baumann, A., Bartenstein, P., Weindl, A., Siebner, H. R., Fassbender, T., ... & Conrad, B. (1999). Sensory processing in Parkinson's and Huntington's disease: Investigations with 3D H215O-PET. *Brain*, *122*(9), 1651-1665.

Boto, E., Holmes, N., Leggett, J., Roberts, G., Shah, V., Meyer, S. S., ... & Brookes, M. J. (2018). Moving magnetoencephalography towards real-world applications with a wearable system. *Nature*, *555*(7698), 657-661.

Buckner, R. L., Andrews-Hanna, J. R., & Schacter, D. L. (2008). The brain's default network: anatomy, function, and relevance to disease. *Annals of the New York Academy of Sciences*, *1124*(1), 1-38.

Cecotti, H., & Graser, A. (2010). Convolutional neural networks for P300 detection with application to brain-computer interfaces. *IEEE transactions on pattern analysis and machine intelligence*, *33*(3), 433-445.

Charvet, G., Foerster, M., Filipe, S., Porcherot, J., Beche, J. F., Guillemaud, R., ... & Benabid, A. L. (2011, April). WIMAGINE: A wireless, low power, 64-channel ECoG recording platform for implantable BCI applications. In *2011 5th International IEEE/EMBS Conference on Neural Engineering* (pp. 356-359). IEEE.

Charvet, G., Foerster, M., Chatalic, G., Michea, A., Porcherot, J., Bonnet, S., ... & Benabid, A. L. (2012, August). A wireless 64-channel ECoG recording electronic for implantable monitoring and BCI applications: WIMAGINE. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 783-786). IEEE.

Chatterjee, A., Aggarwal, V., Ramos, A., Acharya, S., & Thakor, N. V. (2007). A brain-computer interface with vibrotactile biofeedback for haptic information. *Journal of neuroengineering and rehabilitation*, *4*(1), 1-12.

Cohen, D. (1972). Magnetoencephalography: detection of the brain's electrical activity with a superconducting magnetometer. *Science*, *175*(4022), 664-666.

Coyle, J. T., Price, D. L., & DeLong, M. R. (1983). Alzheimer's disease: a disorder of cortical cholinergic innervation. *Science*, *219*(4589), 1184-1190.

Da Silva, F. L., & Van Leeuwen, W. S. (1977). The cortical source of the alpha rhythm. *Neuroscience letters*, *6*(2-3), 237-241.

Da Silva, F. L. (2013). EEG and MEG: relevance to neuroscience. *Neuron*, *80*(5), 1112-1128.

Daly, J. J., Cheng, R., Rogers, J., Litinas, K., Hrovat, K., & Dohring, M. (2009). Feasibility of a new application of noninvasive brain computer interface (BCI): a case study of training for

recovery of volitional motor control after stroke. *Journal of neurologic physical therapy*, *33*(4), 203-211.

Dawson, G. D. (1954). A summation technique for the detection of small evoked potentials. *Electroencephalography & Clinical Neurophysiology, 6,* 65–84.

Desai, S. A., Tcheng, T. K., & Morrell, M. J. (2019). Quantitative electrocorticographic biomarkers of clinical outcomes in mesial temporal lobe epileptic patients treated with the RNS® system. *Clinical Neurophysiology*, *130*(8), 1364-1374.

Dmytriyev, Y., Zaki, A. M. A., Carnevale, M., Insero, F., & Giberti, H. (2021, June). Brain computer interface for human-cobot interaction in industrial applications. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)* (pp. 1-6). IEEE.

Dobie, R. A., & Wilson, M. J. (1998). Low-level steady-state auditory evoked potentials: effects of rate and sedation on detectability. *The Journal of the Acoustical Society of America*, *104*(6), 3482-3488.

Donchin, E., Spencer, K. M., & Wijesinghe, R. (2000). The mental prosthesis: assessing the speed of a P300-based brain-computer interface. *IEEE transactions on rehabilitation engineering*, *8*(2), 174-179.

Dornhege, G., Millán, J. R., Hinterberger, T., McFarland, D. J., & Müller, K.-R. (2007). Toward brain-computer interfacing. The MIT Press.

Dougherty, M. E., Nguyen, A. P., Baratham, V. L., & Bouchard, K. E. (2019, July). Laminar origin of evoked ECoG high-gamma activity. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 4391-4394). IEEE.

Dunn, A. K., Devor, A., Dale, A. M., & Boas, D. A. (2005). Spatial extent of oxygen metabolism and hemodynamic changes during functional activation of the rat somatosensory cortex. *Neuroimage*, *27*(2), 279-290.

Eccles, J. C. (2013). Memories of the Institute for Brain Research: 1930-1950. In: Eccles J.C. (eds) Eccles, J. C. (2013). *Facing Reality: Philosophical adventures by a brain scientist*. Springer Science & Business Media.

Eliseyev, A., Mestais, C., Charvet, G., Sauter, F., Abroug, N., Arizumi, N., ... & Benabid, A. L. (2014, August). CLINATEC® BCI platform based on the ECoG-recording implant WIMAGINE® and the innovative signal-processing: preclinical results. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 1222-1225). IEEE.

Farwell, L. A., & Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, *70*(6), 510-523.

Fazel-Rezai, R., Allison, B. Z., Guger, C., Sellers, E. W., Kleih, S. C., & Kübler, A. (2012). P300 brain computer interface: current challenges and emerging trends. *Frontiers in neuroengineering*, 14.

Ferrari, M., Mottola, L., & Quaresima, V. (2004). Principles, techniques, and limitations of near infrared spectroscopy. *Canadian journal of applied physiology*, *29*(4), 463-487.

Fox, P. T., & Mintun, M. A. (1989). Noninvasive functional brain mapping by change-distribution analysis of averaged PET images of H215O tissue activity. *Journal of Nuclear Medicine*, *30*(2), 141-149.

Frolov, A., Bobrov, P., Biryukova, E., Isaev, M., Kerechanin, Y., Bobrov, D., & Lekin, A. (2020). Using multiple decomposition methods and cluster analysis to find and categorize typical patterns of EEG activity in motor imagery brain–computer interface experiments. *Frontiers in Robotics and AI*, *7*, 88.

Gao, X., Xu, D., Cheng, M., & Gao, S. (2003). A BCI-based environmental controller for the motion-disabled. *IEEE Transactions on neural systems and rehabilitation engineering*, *11*(2), 137-140.

Gevensleben, H., Holl, B., Albrecht, B., Vogel, C., Schlamp, D., Kratz, O., ... & Heinrich, H. (2009). Is neurofeedback an efficacious treatment for ADHD? A randomised controlled clinical trial. *Journal of Child Psychology and Psychiatry*, *50*(7), 780-789.

Grosse-Wentrup, M., & Buss, M. (2008). Multiclass common spatial patterns and information theoretic feature extraction. *IEEE transactions on Biomedical Engineering*, *55*(8), 1991-2000.

Gunasekera, B., Saxena, T., Bellamkonda, R., & Karumbaiah, L. (2015). Intracortical recording interfaces: current challenges to chronic recording function. *ACS chemical neuroscience*, *6*(1), 68-83.

Haas, L. F. (2003). Hans berger (1873–1941), richard caton (1842–1926), and electroencephalography. *Journal of Neurology, Neurosurgery & Psychiatry*, *74*(1), 9-9.

Hämäläinen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., & Lounasmaa, O. V. (1993). Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics*, *65*(2), 413.

Herholz, K., & Heiss, W. D. (2004). Positron emission tomography in clinical neurology. *Molecular Imaging & Biology*, *6*(4), 239-269.

Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., ... & Donoghue, J. P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, *485*(7398), 372-375.

Hutton, C., Josephs, O., Stadler, J., Featherstone, E., Reid, A., Speck, O., ... & Weiskopf, N. (2011). The impact of physiological noise correction on fMRI at 7 T. *Neuroimage*, *57*(1), 101-112.

Ince, N. F., Goksu, F., Tewfik, A. H., & Arica, S. (2009). Adapting subject specific motor imagery EEG patterns in space–time–frequency for a brain computer interface. *Biomedical signal processing and control*, *4*(3), 236-246.

Jarosiewicz, B., Sarma, A. A., Bacher, D., Masse, N. Y., Simeral, J. D., Sorice, B., ... & Hochberg, L. R. (2015). Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Science translational medicine*, *7*(313), 313ra179-313ra179.

Jasper, H.H. (1958) The Ten-Twenty Electrode System of the International Federation. Electroencephalography and Clinical Neurophysiology, 10, 371-375.

Jubien, G., Schaeffer, M. C., Bonnet, S., & Aksenova, T. (2019, July). Decoding of finger activation from ECoG data: a comparative study. In *2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

Kamp, A., Sem-Jacobsen, C. W., Leeuwen, W. S. V., & T-Weel, L. V. D. (1960). Cortical responses to modulated light in the human subject. *Acta physiologica scandinavica*, *48*(1), 1-12.

Kaplan, A. Y., Shishkin, S. L., Ganin, I. P., Basyul, I. A., & Zhigalov, A. Y. (2013). Adapting the P300-based brain–computer interface for gaming: a review. IEEE Transactions on Computational Intelligence and AI in Games, 5(2), 141-149.

Kirar, J. S., & Agrawal, R. K. (2018, November). Relevant frequency band selection using Sequential forward feature selection for motor imagery brain computer interfaces. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 52-59). IEEE.

Krepki, R., Blankertz, B., Curio, G., & Müller, K. R. (2007). The Berlin Brain-Computer Interface (BBCI)–towards a new communication channel for online control in gaming applications. *Multimedia Tools and Applications*, *33*(1), 73-90.

Krusienski, D. J., Sellers, E. W., McFarland, D. J., Vaughan, T. M., & Wolpaw, J. R. (2008). Toward enhanced P300 speller performance. *Journal of neuroscience methods*, *167*(1), 15-21.

Kübler, A., Furdea, A., Halder, S., Hammer, E. M., Nijboer, F., & Kotchoubey, B. (2009). A brain–computer interface controlled auditory event-related potential (P300) spelling system for locked-in patients. *Annals of the New York Academy of Sciences*, *1157*(1), 90-100.

Laufs, H. (2012). A personalized history of EEG–fMRI integration. *Neuroimage*, *62*(2), 1056-1067.

Lazarou, I., Nikolopoulos, S., Petrantonakis, P. C., Kompatsiaris, I., & Tsolaki, M. (2018). EEG-based brain–computer interfaces for communication and rehabilitation of people with motor impairment: a novel approach of the 21st Century. *Frontiers in human neuroscience*, *12*, 14.

Lécuyer, A., Lotte, F., Reilly, R. B., Leeb, R., Hirose, M., & Slater, M. (2008). Brain-computer interfaces, virtual reality, and videogames. *Computer*, *41*(10), 66-72.

Lee, S. H., Choi, K., Jeong, S., Kim, J. S., & Chung, C. K. (2013, February). Classifying ECoG signals prior to voluntary movement onset. In *2013 International Winter Workshop on Brain-Computer Interface (BCI)* (pp. 36-38). IEEE.

Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., & Pfurtscheller, G. (2007). Brain–computer communication: motivation, aim, and impact of exploring a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *15*(4), 473-482.

Lemm, S., Blankertz, B., Curio, G., & Muller, K. R. (2005). Spatio-spectral filters for improving the classification of single trial EEG. *IEEE transactions on biomedical engineering*, *52*(9), 1541-1548.

Lindquist, M. A., Loh, J. M., Atlas, L. Y., & Wager, T. D. (2009). Modeling the hemodynamic response function in fMRI: efficiency, bias and mis-modeling. *Neuroimage*, *45*(1), S187-S198.

Logothetis, N. K. (2008). What we can do and what we cannot do with fMRI. Nature, 453(7197), 869-878.

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., & Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of neural engineering*, *4*(2), R1.

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2018). A review of classification algorithms for EEG-based brain–computer interfaces: a 10-year update. *Journal of neural engineering*, *15*(3), 031005.

Martini, M. L., Oermann, E. K., Opie, N. L., Panov, F., Oxley, T., & Yaeger, K. (2020). Sensor modalities for brain-computer interface technology: a comprehensive literature review. *Neurosurgery*, *86*(2), E108-E117.

Mason, S. G., & Birch, G. E. (2003). A general framework for brain-computer interface design. *IEEE transactions on neural systems and rehabilitation engineering*, *11*(1), 70-85.

Mellinger, J., Schalk, G., Braun, C., Preissl, H., Rosenstiel, W., Birbaumer, N., & Kübler, A. (2007). An MEG-based brain–computer interface (BCI). *Neuroimage*, *36*(3), 581-593.

Mestais, C. S., Charvet, G., Sauter-Starace, F., Foerster, M., Ratel, D., & Benabid, A. L. (2014). WIMAGINE: wireless 64-channel ECoG recording implant for long term clinical applications. *IEEE transactions on neural systems and rehabilitation engineering*, *23*(1), 10-21.

Millán, J. D. R., Rupp, R., Mueller-Putz, G., Murray-Smith, R., Giugliemma, C., Tangermann, M., ... & Mattia, D. (2010). Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges. *Frontiers in neuroscience*, 161.

Mulert, C., Jäger, L., Schmitt, R., Bussfeld, P., Pogarell, O., Möller, H. J., ... & Hegerl, U. (2004). Integration of fMRI and simultaneous EEG: towards a comprehensive understanding of localization and time-course of brain activity in target detection. *Neuroimage*, *22*(1), 83-94.

Müller, K. R., Tangermann, M., Dornhege, G., Krauledat, M., Curio, G., & Blankertz, B. (2008). Machine learning for real-time single-trial EEG-analysis: from brain–computer interfacing to mental state monitoring. *Journal of neuroscience methods*, *167*(1), 82-90.

Müller-Putz, G. R., Scherer, R., Brauneis, C., & Pfurtscheller, G. (2005). Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components. *Journal of neural engineering*, *2*(4), 123.

Muller-Putz, G. R., & Pfurtscheller, G. (2007). Control of an electrical prosthesis with an SSVEP-based BCI. *IEEE Transactions on biomedical engineering*, *55*(1), 361-364.

Niedermeyer, E., & da Silva, F. L. (Eds.). (2005). *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins.

Nijboer, F., Furdea, A., Gunst, I., Mellinger, J., McFarland, D. J., Birbaumer, N., & Kübler, A. (2008). An auditory brain–computer interface (BCI). *Journal of neuroscience methods*, *167*(1), 43-50.

Ogawa, S., Tank, D. W., Menon, R., Ellermann, J. M., Kim, S. G., Merkle, H., & Ugurbil, K. (1992). Intrinsic signal changes accompanying sensory stimulation: functional brain mapping with magnetic resonance imaging. *Proceedings of the National Academy of Sciences*, *89*(13), 5951-5955.

Olman, C. A., Harel, N., Feinberg, D. A., He, S., Zhang, P., Ugurbil, K., & Yacoub, E. (2012). Layer-specific fMRI reflects different neuronal computations at different depths in human V1. *PloS one*, *7*(3), e32536.

Panicker, R. C., Puthusserypady, S., & Sun, Y. (2011). An asynchronous P300 BCI with SSVEP-based control state detection. *IEEE Transactions on Biomedical Engineering*, *58*(6), 1781-1788.

Perdikis, S., Tonin, L., Saeedi, S., Schneider, C., & Millán, J. D. R. (2018). The Cybathlon BCI race: Successful longitudinal mutual learning with two tetraplegic users. *PLoS biology*, *16*(5), e2003787.

Perge, J. A., Zhang, S., Malik, W. Q., Homer, M. L., Cash, S., Friehs, G., ... & Hochberg, L. R. (2014). Reliability of directional information in unsorted spikes and local field potentials recorded in human motor cortex. *Journal of neural engineering*, *11*(4), 046007.

Pfurtscheller, G., Neuper, C., Schlogl, A., & Lugger, K. (1998). Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. *IEEE transactions on Rehabilitation Engineering*, *6*(3), 316-325.

Pfurtscheller, G., Allison, B. Z., Bauernfeind, G., Brunner, C., Solis Escalante, T., Scherer, R., ... & Birbaumer, N. (2010). The hybrid BCI. *Frontiers in neuroscience*, 3.

Pfurtscheller, G., & Neuper, C. (2006). Future prospects of ERD/ERS in the context of brain–computer interface (BCI) developments. *Progress in brain research*, *159*, 433-437.

Piccolino, M. (1997). Luigi Galvani and animal electricity: two centuries after the foundation of electrophysiology. *Trends in neurosciences*, *20*(10), 443-448.

Polich, J. (2007). Updating P300: an integrative theory of P3a and P3b. *Clinical neurophysiology*, *118*(10), 2128-2148.

Ramoser, H., Muller-Gerking, J., & Pfurtscheller, G. (2000). Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE transactions on rehabilitation engineering*, *8*(4), 441-446.

Ryun, S., Kim, J. S., Lee, D., & Chung, C. K. (2018, January). Prediction of motor and somatosensory function from human ECoG. In *2018 6th International Conference on Brain-Computer Interface (BCI)* (pp. 1-4). IEEE.

Sadiq, M. T., Yu, X., & Yuan, Z. (2021). Exploiting dimensionality reduction and neural network techniques for the development of expert brain–computer interfaces. *Expert Systems with Applications*, *164*, 114031.

Sakurada, T., Kawase, T., Komatsu, T., & Kansaku, K. (2015). Use of high-frequency visual stimuli above the critical flicker frequency in a SSVEP-based BMI. *Clinical neurophysiology*, *126*(10), 1972-1978.

Schalk, G., & Leuthardt, E. C. (2011). Brain-computer interfaces using electrocorticographic signals. *IEEE reviews in biomedical engineering*, *4*, 140-154.

Schalk, G., & Mellinger, J. (2010). *A practical guide to brain–computer interfacing with BCI2000: General-purpose software for brain-computer interface research, data acquisition, stimulus presentation, and brain monitoring*. Springer Science & Business Media.

Scherer, R., Lee, F., Schlogl, A., Leeb, R., Bischof, H., & Pfurtscheller, G. (2008). Toward self-paced brain–computer communication: navigation through virtual worlds. *IEEE Transactions on Biomedical Engineering*, *55*(2), 675-682.

Schomer, D. L., & Da Silva, F. L. (2012). *Niedermeyer's electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins.

Sellers, E. W., Vaughan, T. M., & Wolpaw, J. R. (2010). A brain-computer interface for long-term independent home use. *Amyotrophic lateral sclerosis*, *11*(5), 449-455.

Sellers, E. W., & Donchin, E. (2006). A P300-based brain–computer interface: initial tests by ALS patients. *Clinical neurophysiology*, *117*(3), 538-548.

Sellers, E. W., Krusienski, D. J., McFarland, D. J., Vaughan, T. M., & Wolpaw, J. R. (2006). A P300 event-related potential brain–computer interface (BCI): the effects of matrix size and inter stimulus interval on performance. *Biological psychology*, *73*(3), 242-252.

Shenoy, P., Krauledat, M., Blankertz, B., Rao, R. P., & Müller, K. R. (2006). Towards adaptive classification for BCI. *Journal of neural engineering*, *3*(1), R13.

Simanovics, J., Kolosovs, D., & Litvinenko, A. (2023, April). Brain-Computer Interfacing: Developments, Perspectives and Limitations. In *2023 IEEE 10th Jubilee Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* (pp. 1-6). IEEE.

Tan, H. R., Gross, J., & Uhlhaas, P. J. (2015). MEG—measured auditory steady-state oscillations show high test–retest reliability: A sensor and source-space analysis. *Neuroimage*, *122*, 417-426.

Townsend, D. W. (2008, May). Positron emission tomography/computed tomography. In *Seminars in nuclear medicine* (Vol. 38, No. 3, pp. 152-166). WB Saunders.

Vaughan, T. M., McFarland, D. J., Schalk, G., Sarnacki, W. A., Krusienski, D. J., Sellers, E. W., & Wolpaw, J. R. (2006). The wadsworth BCI research and development program: at home with BCI. *IEEE transactions on neural systems and rehabilitation engineering*, *14*(2), 229-233.

Veena, N., & Anitha, N. (2020). A review of non-invasive BCI devices. *Int. J. Biomed. Eng. Technol*, *34*(3), 205-233.

Vialatte, F. B., Maurice, M., Dauwels, J., & Cichocki, A. (2010). Steady-state visually evoked potentials: focus on essential paradigms and future perspectives. *Progress in neurobiology*, *90*(4), 418-438.

Vidaurre, C., Kawanabe, M., von Bünau, P., Blankertz, B., & Müller, K. R. (2010). Toward unsupervised adaptation of LDA for brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, *58*(3), 587-597.

Vidaurre, C., Sannelli, C., Müller, K. R., & Blankertz, B. (2011). Machine-learning-based coadaptive calibration for brain-computer interfaces. *Neural computation*, *23*(3), 791-816.

Volosyak, I. (2011). SSVEP-based Bremen–BCI interface—boosting information transfer rates. *Journal of neural engineering*, *8*(3), 036020.

Waldert, S. (2016). Invasive vs. non-invasive neuronal signals for brain-machine interfaces: will one prevail? *Frontiers in neuroscience*, *10*, 295.

Wan, Z., Yang, R., Huang, M., Zeng, N., & Liu, X. (2021). A review on transfer learning in EEG signal analysis. *Neurocomputing*, *421*, 1-14.

Wang, H., Li, T., Bezerianos, A., Huang, H., He, Y., & Chen, P. (2019). The control of a virtual automatic car based on multiple patterns of motor imagery BCI. *Medical & biological engineering & computing*, *57*, 299-309.

Wang, M., Li, R., Zhang, R., Li, G., & Zhang, D. (2018). A wearable SSVEP-based BCI system for quadcopter control using head-mounted device. *Ieee Access*, *6*, 26789-26798.

Weiskopf, N., Hutton, C., Josephs, O., Turner, R., & Deichmann, R. (2007). Optimized EPI for fMRI studies of the orbitofrontal cortex: compensation of susceptibility-induced gradients in the readout direction. *Magnetic Resonance Materials in Physics, Biology and Medicine*, *20*, 39-49.

Yang, R., Song, A., & Xu, B. (2010). Feature extraction of motor imagery EEG based on wavelet transform and higher-order statistics. *International Journal of Wavelets, Multiresolution and Information Processing*, *8*(03), 373-384.

Yao, L., Zhu, B., & Shoaran, M. (2022). Fast and accurate decoding of finger movements from ECoG through Riemannian features and modern machine learning techniques. *Journal of Neural Engineering*, *19*(1), 016037.

Zhang, Y., Xu, P., Liu, T., Hu, J., Zhang, R., & Yao, D. (2012). Multiple frequencies sequential coding for SSVEP-based brain-computer interface. *PloS one*, *7*(3), e29519.

CHAPITRE III

SYSTEME D'ACQUISITION DE L'ACTIVITE EEG POUR UNE INTERFACE CERVEAU-ORDINATEUR

CONTENU

III.1. Introduction

III.2. État de l'art et travaux similaires

III.3. Architecture du système d'acquisition EEG

III.3.1. Conception du matériel

III.3.1.1. Électrodes

III.3.1.2. Frontal analogique ADS1299-AFE

III.3.1.3. Filtre passe-bas anti repliement

III.3.1.4. Dispositifs de protection

III.3.1.5. Alimentation électrique analogique et numérique

III.3.1.6. Unité de contrôle et de communication - UCC

III.3.2. Présentation du logiciel

III.3.2.1. Bibliothèque compilable

III.3.2.2. Firmware

III.3.2.3. Pilote

III.3.2.4. Cadre d'acquisition, de visualisation et de contrôle-AVCF

III.4. Conclusion

Bibliographie du Chapitre III

III.1. Introduction

Depuis la découverte de l'électroencéphalographie dans les années 1920 jusqu'à nos jours et en raison de son importance, ce domaine de recherche a motivé la fabrication d'appareils commerciaux, à l'échelle professionnelle, destinés à mesurer l'activité cérébrale ; plusieurs fabricants ont développé des modèles [medicalexpo, 2020] hautement qualifiés pour être passés dans le domaine médical et professionnel et qui se caractérisent par leur robustesse et leur précision, mais malheureusement ils étaient loin d'être abordables par le grand public en raison de leur coût très élevé et de leur encombrement, parmi plusieurs types nous en citerons quelques-uns : Nicolet Monitor EEG (*Natus - USA*), Neurofax EEG (*Nihon Kohden - Japon*), TruScan EEG (*DeyMed - France*), ICU Monitoring (*EBNeuro - Italie*), Desktop EEG Machine Kit (*Clarity - Inde*), Visor2 (*ant neuro - UK*), cette gamme de produits n'est plus financièrement accessible aux étudiants chercheurs qui souhaiteraient réaliser leurs expériences avec l'EEG. La raison qui a encouragé d'autres équipes de recherche et de passionnés de la technologie d'interface cerveau-ordinateur à essayer de concevoir des dispositifs d'acquisition EEG plus miniaturisés, moins coûteux et avec le même niveau de fiabilité, de tels systèmes sont disponibles pour la recherche scientifique et même pour des études cliniques, à cet égard il faut mentionner certains projets promoteurs tels que : NeuroScan, Biosemi, TMSi, ActivCHamp (*BrainProducts*), EGI, EMOTIV, NeuroSky, g.tec, ANT Neuro, Muse, OpenBCI et Neuroelectrics, etc. [imotions, 2017]. Cependant, même avec ce type de dispositifs, un étudiant chercheur d'un pays en voie de développement sera privé de s'en procurer en raison de leur coût relativement élevé (le tableau III.1 établit une comparaison entre certains de ces modèles). La réalisation de tels systèmes reste une tâche compliquée et exigeante. Heureusement, l'évolution énorme et rapide de la technologie des circuits intégrés donne naissance à des composants VLSI (*Very-Large-Scale Integration*) entièrement programmables, souvent appelés FPGA (*Field-Programmable Gate Arrays*) [Kumar M. A., & Chari K. M., 2018] qui sont des réseaux logiques programmables. Ces techniques avancées conduisent à la fabrication de composants frontaux analogiques (par exemple ADS1299AFE) de sorte qu'ils incluent de nombreuses fonctions (filtrage, conversion, amplification, interfaçage, contrôle, etc.) sur la même puce. L'introduction de ce type d'amplificateurs et de convertisseurs programmables, caractérisés par un plancher de bruit très bas et un rapport de rejet en mode commun suffisamment élevé, permet de réaliser des BCI et des dispositifs d'acquisition EEG malgré la nature bruyante et non stationnaire des signaux EEG [Liyanage SR et al., 2013]. C'est la raison qui nous a motivés à concevoir un système d'acquisition EEG basé sur le frontal analogique ADS1299-AFE à 8 canaux qui comprend un convertisseur

analogique à haute résolution de 24 bits [Instruments T., 2017a]. L'objectif principal est de fournir un modèle avec lequel on peut mesurer les signaux EEG, ou même l'utiliser comme une interface cerveau-ordinateur en combinaison avec une boîte à outils de post-traitement dont on peut, dans la littérature, trouver quelques exemples [Venthur, B., 2015 ; Tayeb, Z. et al., 2018]. Tout ceci doit assurer la portabilité du système, sa faible consommation, son coût global réduit et son accessibilité par les personnes intéressées par la recherche en neurosciences.

Dispositif **Paramètres**	**Mitsar-EEG-202**	**Q21**	**Discovery 24E**	**FlexComp Infiniti**	**Optima+ 4BT**	**SAGA 64+ for EEG**	**g.Nautilus PRO**	**ActiveTwo Mk2**	**OpenBCI**
Compagnie / Pays	MITSAR/RUS	NeuroField inc./USA	BrainMaster/USA	Thought Tech. / Canada	Neurobit / Pologne	TMSi / Pays-Bas	g.tec / Autriche	Biosemi / Pays-Bas	OpenBCI / USA
Utilisation	Recherche Clinique	Recherche Thérapie-neuronale	Recherche Biofeedback Thérapie	Recherche Clinique Biofeedback	Recherche BCI	Recherche Médecine	Recherche Clinique BCI	Recherche BCI	Recherche BCI Biofeedback
# Canaux	Up to 41	20	24	10	4	64+	32	152+	16
Compatibilité logicielle	WinEEG Psytask	NF 64 NG	BrainAvatar QEEG Pro	BioGraph	BioEra Bioexplorer BrainBay Neurobit rec.	Polybench SAGA SDK Matlab	g. HISYS g. Recorder g.BSAnalyse	ActiView	OpenBCI GUI MATLAB, OpenViBE
TRMC	NA	>120 dB	>110 dB	NA	>130 dB	>100 dB	>100 dB	>90 dB	>110 dB
Interface	USB	USB-CAN	USB	USB Bluetooth	USB Bluetooth	USB Ethernet Wi-Fi	Sans-fil 2.4GHz	USB	Wi-Fi Bluetooth
Résolution du CAN	24-bit	24-bit	24-bit	14-bit	16-bit	24-bit	24-bit	24-bit	24-bit
Bruit référé à l'entrée	<1,5 μV_{pp}	<0,25 μV_{pp}	<2,82 μV_{pp}	NA	<1 μV_{pp}	<1 μV_{pp}	<1,69 μV_{pp}	<1 μV_{pp}	<1 μV_{pp}
Bande passante	CC à 500Hz	CC à 80Hz	CC à 80Hz	CC à 512Hz	CC à 800Hz	CC à 800Hz	NA	CC à 3,2KHz	CC à 8KHz
Débit de transfert	0.5 or 2 Kéch./S	Max 256 éch./S	Max 256 éch./S	Max 2,048 Kéch./S	Max 2,0 Kéch./S	Max 4,096 Kéch/S	500 éch./S	16384 éch./S	Max 16 Kéch./S
Prix ($) (2023)	10500	5995	7490	6500	1250,36	NA	NA	15770	949,99

Tableau III.1. Comparaison des modules d'acquisition EEG trouvés dans la littérature.

III.2. État de l'art et travaux similaires

Ives J. R. et al., 1973 ont mis en place un système d'enregistrement EEG à 4 canaux, basé sur une carte de télémétrie TEDAC (*Telemetry Data Acquisition and Control*) qui comporte 4 émetteurs montés sur le cuir chevelu, un récepteur, deux dispositifs d'enregistrement, le premier est graphique avec un terminal Tektronix et le second sur une bande magnétique (PEC), puis ils ont utilisé un calculateur PDP-12, équipé d'une mémoire de 8 Ko et d'un disque RK8. Cette configuration a été réalisée pour effectuer l'analyse des données EEG et étudier les crises spontanées des patients épileptiques.

Quelques années plus tard, Dumermuth, G. & Dinkelmann, R.,1979 ont conçu un système d'acquisition EEG à 8 canaux parallèles basé sur une machine EEG conventionnelle connectée à l'unité Satellite-DEC LSI-11 pour effectuer les opérations d'acquisition et de prétraitement (calcul de la transformée de Fourier rapide - TFR, la moyenne temporelle des données, etc.), il est équipé d'un convertisseur analogique-numérique CAN (*ADAC modèle 1020*) avec une résolution de 12 bits, les données sont transférées à 102,4 Hz à un mini-ordinateur central DEC PDP-11 qui a pour rôle de post-traiter les données reçues du LSI-11. Il ne fait aucun doute que l'utilisation de deux gros ordinateurs constituait le principal problème en raison de leur faible capacité de stockage à l'époque.

Avec l'évolution de la technologie des circuits intégrés vers l'intégration à grande échelle LSI (*Large Scale Integration*), les ordinateurs deviennent de plus en plus puissants et rentables, aussi Vo, T. D. H. et al., 1986 ont proposé un système d'acquisition EEG basé sur un enregistreur analogique Gould 2800S et un enregistreur à cassette 7 canaux TEAC-X300 FM, la numérisation des signaux analogiques est effectuée avec un convertisseur analogique/numérique A/N à microprocesseur (*Cyborg ISAAC 2000*), l'ensemble du système est connecté à un IBM-PC équipé d'un microprocesseur 8086 et d'un coprocesseur 8087, ce qui leur a permis d'accélérer les calculs et d'effectuer des fonctions transcendantales.

Dunseath W. R., & Kelly E. F., 1995 ont mis en œuvre un prototype pour l'acquisition d'EEG extensibles à 32 canaux, ils l'ont considéré comme plus efficace que les autres modèles commerciaux existants, ce système comprenait des électrodes EEG actives (un préamplificateur avec TLC2201ACD), des amplificateurs d'instrumentation (*Burr-Brown INA101U*) pour ajuster/améliorer le taux de réjection en mode commun TRMC, un circuit de calibration construit par un oscillateur sinusoïdal contrôlé en tension. La conversion A/N du signal analogique échantillonné à la sortie est réalisée par un convertisseur AD578 à 12 bits. L'interfaçage avec un

ordinateur se fait à l'aide d'une carte CYDIO 24H I/O de CyberResearch qui est assez rapide pour collecter des données à des fréquences allant jusqu'à 2000 échantillons par seconde. Les résultats de leurs tests montrent d'excellentes performances, le bruit référé à l'entrée est en moyenne de 1,31 ± 0,11 µVpp (bande passante de 1 KHz), le rapport de réjection en mode commun varie entre (85 - 90 dB) à la fréquence de 60 Hz.

Depuis lors, une nouvelle ère est arrivée, dans laquelle de nouvelles méthodes ont été développées pour intégrer les différentes fonctions d'un système EEG dans un circuit intégré. Parmi les nombreux travaux, on a distingué celui de Akin T. et al., 1998 qui est le plus influent, lorsqu'ils ont présenté une autre configuration invasive et sans fil basée sur un dispositif d'acquisition de signaux neuronaux mis en œuvre dans une seule puce monolithique, par le biais du processus bipolaire CMOS (*Complementary Metal-Oxide-Semiconductor*), qui combine toutes les fonctions (filtrage, multiplexage, amplification, conversion, etc.). De plus, ce système est entièrement implantable, c'est-à-dire qu'il est physiquement isolé du monde extérieur. L'alimentation, la réception des données, l'envoi des commandes sont assurés par la télémétrie Radiofréquence RF, la numérisation du signal est assurée par un convertisseur A/N de 8 bits. Cette conception est considérée optimale pour les applications d'enregistrement à long terme.

En réponse à la demande croissante des systèmes d'acquisition EEG portables et à faible consommation pour les patients ambulatoires, Yazicioglu R.F. et al., 2005 ont publié une lettre présentant le premier circuit intégré spécialisé sous forme un frontal CMOS multicanaux de 12 bits de résolution, capable de détecter les signaux EEG provenant d'électrodes Ag/AgCl conventionnelles sans avoir à introduire de composants de réglage. Ce modèle performant est caractérisé par un bruit référé à l'entrée inférieur à 1µV. De plus, les développeurs du modèle ont amélioré l'amplificateur d'instrumentation intégré afin que le TRMC soit immunisé contre la tension de décalage des électrodes (*Offset*), et ont obtenu un TRMC de 95 dB à une tension de décalage 0V, et un TRMC de 107 dB à une tension de décalage 50 mV.

Dans le même contexte lié au besoin de modules plus miniaturisés et moins chers, Verma, N. et al., 2010 ont développé un système sur puce (*SoC*) non invasif et de faible puissance, qui est utilisé comme outil de détection des crises chez les patients épileptiques. Chaque puce correspond à un seul canal EEG et comprend un filtre passe-bas, un amplificateur d'instrumentation, un convertisseur A/N à 12 bits et un extracteur de caractéristiques. Le classificateur SVM (*Support Vector Machine*) est implémenté dans un ordinateur distant pour lire en continu la sortie frontale afin de détecter le début des crises d'épilepsie. Les résultats des tests ont montré des performances

telles que : le taux de réjection en mode commun (TRMC) est de ~60 dB, le bruit référé à l'entrée est de ~3,67 µVpp dans la bande [0 - 100 Hz].

Même si les équipements disponibles ont atteint des performances élevées, et compte tenu de la nature bruyante des signaux EEG très faibles et de leur extrême sensibilité aux interférences, l'isolation de ces systèmes est devenue un défi en soi. Cette lacune a incité Minglong, Y. et al., 2012 à proposer une approche alternative d'acquisition EEG haute performance pour les environnements non protégés. Un circuit de conditionnement EEG intégrant un banc de filtres est mis en œuvre, la fonction de conversion A/N à 12 bits est immédiatement exécutée par un microcontrôleur MSP430F169, et les données de sortie sont capturées via le logiciel d'acquisition de données LabVIEW.

Toujours dans le but de réduire les artefacts et la taille du système, Xu, J.et al., 2014 ont présenté un dispositif de mesure EEG à 8 canaux mis en œuvre avec le processus CMOS 0,18 µm, son frontal recommande l'utilisation d'électrodes actives équipées de préamplificateurs pour réduire les artefacts causés soit par les différentes interférences de ligne, soit par l'impédance électrode-tissu variable, la conversion A/N de 12 bits et le traitement analogique sont réalisés par un back-end de faible puissance et hautement configurable, les résultats montrent que le système est caractérisé par un bruit référé à l'entrée de 4,95 µVpp soit 1,75 µVrms dans la bande [0,5 - 100Hz] et un TRMC qui atteint 84 dB.

Afin d'améliorer la rentabilité des outils de diagnostic pour les patients nécessitant une surveillance ambulatoire permanente, certaines solutions ont été présentées, notamment Pinho F. et al., 2014 qui ont proposé un système d'acquisition EEG sans fil (Wi-Fi 802.11 b/g/n) à 32 canaux, basé sur le puissant et petit module de contrôle IGEP COM (CPU 1.0GHz), 4 frontaux analogiques ADS1299 avec un CAN de 24 bits ont été utilisés. Cette plateforme semble être optimale pour le suivi de patients ambulatoires souffrant de certaines maladies paroxystiques, compte tenu des spécifications à respecter dans un tel système, 32 électrodes sèches peuvent être connectées, l'autonomie en énergie dépasse 24 heures, la vitesse de transmission atteint 1 kilo- échantillons par seconde, etc. Les résultats obtenus ont montré un TRMC qui atteint ~115 dB et un bruit référé à l'entrée est inférieur à 1,0 µVpp pour un gain de 24.

L'intérêt croissant des chercheurs et des concepteurs enthousiastes pour l'interface cerveau-ordinateur et le neurofeedback a donné naissance à plusieurs techniques comme l'imagerie motrice sur laquelle Lo C. C. et al., 2016 ont proposé un dispositif EEG portable spécifique pour une interface cerveau-ordinateur basée sur l'imagerie motrice. La partie frontale du système intègre

des électrodes sèches actives et rétractables, un multiplexeur analogique (8x1), un filtre analogique spatial CAR (*Common Average Reference*), un amplificateur d'instrumentation, un filtre passe-bande [5 - 30Hz], un module de transmission sans fil via Bluetooth v2.0 et un microcontrôleur MSP430 doté d'un convertisseur A/N à 12 bits. Le MSP430 MCU contient le micrologiciel (*Firmware*) et l'algorithme de sélection des canaux pour ne sélectionner que les deux canaux qui contiennent les caractéristiques pertinentes d'imagerie motrice gauche et droite. Le back-end est situé sur un ordinateur distant avec la plateforme de visualisation et de classification des événements détectés.

Plus tard, certains travaux ont été réalisés pour tenter de résoudre le problème de modularité, l'attention étant attirée sur Uktveris T. et al., 2018 qui ont présenté un système d'acquisition EEG empilable, compact et portable basé sur le frontal analogique ADS1298. Le dispositif intègre deux front-end analogiques ADS1298 qui ont permis d'avoir un total de 16 canaux pour chaque carte, un microcontrôleur 8 bits Atmega2560 (CPU 16MHz, 256KB ISP Flash), un module Bluetooth avec HM-11 BLE4.0, un module Wi-Fi avec ESP8266-12E, un gyroscope MPU6050, un support de stockage micro SD, la connexion entre le microcontrôleur et le front-end est réalisée via le protocole SPI. La transmission des signaux EEG numérisés vers le back-end est sans fil (Bluetooth, Wi-Fi) avec une vitesse de transmission allant de 250 éch. /s à 1 Kéch. /s. Les résultats des tests montrent un CMRR stable de ~108 dB dans la bande [5 - 30 Hz] avec un gain de 4 et un bruit référencé à l'entrée égal à 2,87 µVpp pour un gain de 12 et un taux d'échantillonnage de 500 éch. /s.

Pour comprendre l'influence des interactions sociales sur la cognition, l'enregistrement simultané de l'activité cérébrale de deux utilisateurs ou plus est nécessaire au développement de l'approche hyperscanner. C'est sur cette base que Lee S. et al., 2019 ont conçu un système multi-utilisateur à 8 canaux, pour l'acquisition simultanée de signaux biopotentiels, chaque dispositif mono-utilisateur est équipé d'un frontal analogique ADS1299, d'un RaspberryPi Pi Zero W pour la gestion et la communication Wi-Fi via le protocole TCP/IP, puis le dispositif est connecté à un routeur commutateur pour lui attribuer une adresse IP. L'objectif de ce travail était de fournir une plateforme multicanaux et multi-utilisateurs qui peut être utilisée principalement dans des applications collaboratives de BCI et d'hyperscanning.

III.3. Architecture du système d'acquisition EEG

Dans cette section, nous allons décrire en détail comment le système proposé a été conçu du point de vue matériel et logiciel. Avant de choisir les différents composants, en particulier les frontaux analogiques (Analog Front-End, AFE), il est nécessaire de connaître tous leurs avantages et inconvénients, afin de rendre la mise en œuvre du système aussi simple que possible.

Des lacunes dans la conception physique peuvent apparaître à la suite d'une certaine décision matérielle incohérente. En outre, l'acquisition de données EEG, en temps réel, n'est pas une tâche assez simple, car le problème ne réside pas seulement dans la lecture continue de l'activité cérébrale à partir des électrodes placées sur le cuir chevelu humain, mais plutôt dans la capacité d'accéder aux parties configurables afin d'exploiter toute la flexibilité qui caractérise le frontal utilisé, car toutes les fonctions intrinsèques fournies par les frontaux analogiques doivent être manipulées de manière efficace. Plusieurs dispositifs actuellement disponibles se limitent à la lecture continue des signaux EEG et ne permettent pas leur reconfiguration en ligne, par exemple : ajustement du gain et de la fréquence d'échantillonnage, mesure de l'impédance électrode-tissu ETI (*Electrode-Tissue Impedance*), génération de signaux de test internes, etc.

III.3.1. Conception du matériel

Cette partie est consacrée à l'implémentation matérielle et à la description des différents modules du système. Le diagramme de la figure III.1 représente l'organisation fonctionnelle sous forme de blocs. En outre, certains aspects importants du système ont été illustrés graphiquement ou analytiquement.

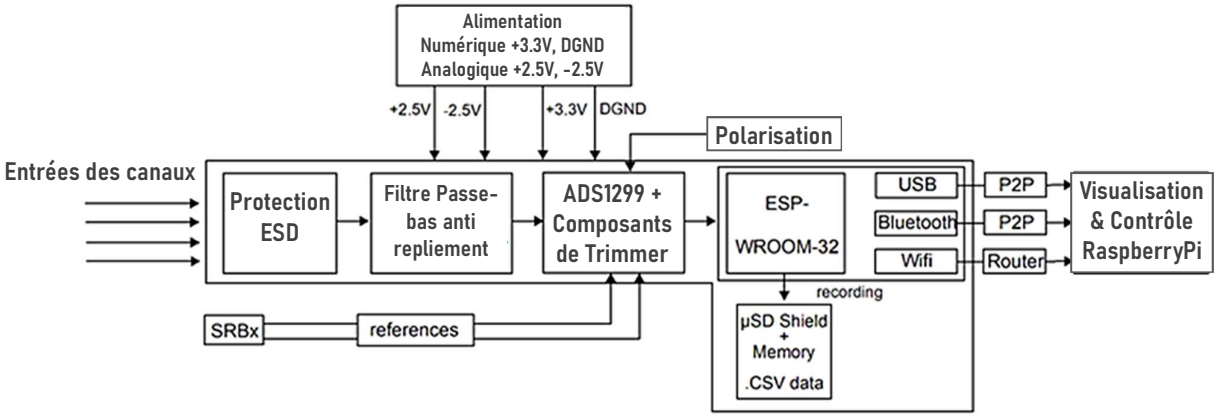


Figure III.1. Schéma fonctionnel du système d'acquisition proposé.

Ce dispositif d'acquisition EEG est constitué d'une structure hiérarchique matérielle et logicielle qui vise à transmettre les signaux physiologiques induits par des électrodes, placées sur la peau, vers un back-end de post-traitement (PC, RaspberryPi, etc.). En effet, la partie matérielle

est construite autour de plusieurs étages, chacun d'entre eux effectuant une tâche spécifique, comme la protection contre les décharges électrostatiques ESD (*Electrostatic Discharges*), et le filtrage passe-bas (f_c=7,2 KHz) pour éliminer les interférences Radiofréquence RF. De plus, les opérations d'amplification, d'échantillonnage et de conversion A/N sont gérées par le frontal analogique ADS1299. Enfin, les données acquises sont stockées localement sur une carte mémoire de type µSD ; simultanément, l'unité de contrôle et de communication les pousse vers le back-end. Ce dernier est équipé d'un pilote (*driver*) et d'un logiciel de contrôle qui ont pour rôle de gérer le processus de structuration des données reçues ainsi que de gérer les opérations de visualisation et de stockage. À noter que l'entrée BIAS sert de RLD (*Right Leg Drive*) dont le but est d'améliorer le taux de réjection en ode commun TRMC. La partie logicielle est décrite dans les sections suivantes et a pour rôle d'orchestrer le processus d'acquisition.

III.3.1.1. Électrodes

Une électrode est un composant capable de détecter la migration des ions à la surface des tissus et de la convertir en un courant électrique qui circule dans un fil conducteur vers un dispositif de mesure. Ce sont des organes de capture et sont extrêmement sensibles à leur environnement, il en existe de nombreux types qui diffèrent selon : leur technologie de fabrication [Salvo P. et al., 2012 ; Chen Y. H. et al., 2014], leur catégorie (sèche, humide, passive, active, invasive, non invasive, etc.) et donc leur application d'utilisation (EEG, EMG, ECG, etc.) [Ma R. et al., 2010]. Le choix des électrodes appropriées pour une application particulière est très important, car la qualité du signal dépend fortement du type d'électrodes utilisées [Zhao Z.et al., 2018]. Deux catégories principales divisent les électrodes et peuvent être utilisées dans n'importe quel système :

a) Les électrodes passives qui nécessitent une préparation préalable de la peau par l'application d'une pâte conductrice (par exemple, Ten20).
b) Les électrodes actives qui intègrent un préamplificateur à proximité de l'électrode afin de réduire considérablement l'impédance de sortie et, par conséquent, d'empêcher le bruit ambiant de contaminer l'activité cérébrale utile, ce type d'électrodes ne nécessite aucune préparation de la peau du patient.

Dans cette section, l'accent est mis sur un type d'électrode, à savoir les électrodes de surface (c'est-à-dire non invasives) et passives (c'est-à-dire nécessitant l'application d'une pâte conductrice d'électrolyte). Le tableau III.2 indique les électrodes préférées pour chaque type de biopotentiels.

Signal biologique	Source	Type d'électrodes préféré	Coût (2022)
EEG	Cérébrale	Ag/AgCl gold-cup, réutilisable	$ 44.99 (10 pièces)
ECG	Cardiaque	Ag/AgCl jetable	$ 14.99 (30 pièces)
EMG	Musculaire	CSA (Carbone/Sel/Adhésive)	Moins chers

Tableau III.2. Types d'électrodes préférés pour l'EEG, l'ECG et l'EMG.

Dans la plupart des applications, les électrodes sont choisies en fonction de certains critères tels que la nature et la source du biopotentiel, le coût des électrodes, le rapport signal/bruit SNR (*Signal to Noise Ratio*), etc. Par exemple, dans la mesure de l'EMG, les électrodes CSA (Carbone/Sel/Adhésive) sont préférables car elles ont montré un SNR considérablement élevé : 38,3 ± 10,6 dB comparé aux 32,7 ± 15,6 dB des électrodes Ag/AgCl, ainsi que leur faible coût [Posada-Quintero H. F. et al., 2016]. Cependant, sur la base de plusieurs recherches menées dans ce contexte [Cencen V. et al., 2016 ; Albulbul A., 2016], il a été démontré que les électrodes argent/chlorure d'argent (Ag/AgCl) sont considérées comme l'un des meilleurs types d'électrodes, en raison de leur faible potentiel de demi-cellule d'environ 220 mV, de leur stabilité et de leur faible bruit.

Ce travail se limite à l'utilisation d'électrodes passives Ag/AgCl à godets d'or, dans la mesure où elles garantissent précision et fiabilité [Cencen, V. et al., 2016]. Il s'agit d'électrodes commerciales réutilisables, étanches, dotées de coupelles en or Ag/AgCl de 10 mm avec un trou central de 2 mm. Elles contiennent un tube rétractable étanche et un moulage antioxydant entre l'électrode et le fil.

La figure III.2 montre un kit d'électrodes passives à cupule d'or et une pâte conductrice Ten20 utilisée pendant les tests.



Figure III.2. Kit d'électrodes en or Ag/AgCl et pâte conductrice Ten20 utilisés.

Les électrodes Ag/AgCl sont des électrodes passives non polarisées et sont préférables aux électrodes polarisées en termes de rejet des artefacts de mouvement [Lee S. & Kruse J., 2008] et de réponse aux courants de défibrillation. Cette caractéristique est importante aux basses fréquences proches du courant continu, où sont effectuées les mesures d'ECG et d'EEG. Les signaux EEG étant de très faible amplitude (dizaines de microvolts), nous nous sommes appuyés sur un type spécifique d'électrodes Ag/AgCl constituées d'une épaisse bande plaquée or (or pur à 99,9 %) qui permet un enregistrement sans artefact.

À la lumière de ce qui précède, nous avons été satisfaits de ce travail sur l'utilisation d'électrodes non polarisées réutilisables en Ag/AgCl de 10 mm avec un trou central de 2 mm pour mesurer les biopotentiels EEG, ECG et EMG. Ce type d'électrode passive nécessite une préparation préalable de la peau par application d'une pâte conductrice afin de limiter l'Impédance Electrode-Tissue (ETI) [Xu J. et al., 2014].

III.3.1.2. Frontal analogique ADS1299-AFE

L'utilisation de Frontaux Analogiques (AFEs) compacts pour l'acquisition de signaux bioélectriques est devenue une nécessité absolue en raison des nombreux avantages qu'ils offrent tels que : une grande précision [Toresano L. O. H. Z. et al., 2017], un faible bruit référé à l'entrée, une taille réduite, une faible consommation électrique, etc. Néanmoins, plusieurs alternatives d'AFEs existent sur le marché mais pour choisir le modèle approprié à l'acquisition de signaux EEG, certaines spécifications doivent être vérifiées pour faire un choix optimal.

Une analyse comparative a été faite par Mastinu E. et al., 2015 entre les deux frontaux qui sont couramment utilisés dans la recherche, il s'agit de l'ADS1299-AFE (*Texas Instruments*) et du RHA2216 (*Intan Technologies*), leurs résultats indiquent que les deux ont des rapports signal-bruit (SNR) similaires, tandis que l'ADS1299 présente de meilleures performances en termes de faible bruit référé à l'entrée et de haute précision de reconnaissance des formes myoélectriques MPR (*Myoelectric Pattern Recognition*).

L'utilisation du frontal analogique ADS1299-AFE est recommandée par plusieurs chercheurs [Uktveris T. et al., 2018 ; Mastinu E. et al., 2017] car il possède de nombreuses caractéristiques et fonctionnalités nécessaires aux applications EEG [Li Y. X. et al., 2014], ECG [Instruments T., 2017a ; Uktveris T. et al., 2018], et EMG sans qu'il soit nécessaire de concevoir des circuits supplémentaires. [Mastinu E. et al., 2017]

La raison pour laquelle nous avons choisi d'utiliser l'ADS1299-AFE est qu'il présente un réel intérêt et intègre toutes les fonctionnalités nécessaires aux applications EEG. De plus, une

étude expérimentale menée par Rashid U. et al., 2018 qui vise à évaluer les performances de l'ADS1299-AFE par rapport à un autre système de laboratoire d'une gamme professionnelle, les résultats ont montré qu'il n'y a pas de différence entre les deux appareils en termes de qualité du signal EEG aux basses fréquences (< 40 Hz), étant donné les excellentes performances évaluées, les auteurs ont conclu qu'un prototype basé sur l'ADS1299 est un bon choix dans le domaine des neurosciences ainsi que dans les applications BCI.

La réalisation d'un système évolutif est maintenant possible grâce à ses performances exceptionnelles [Rashid U. et al., 2018] telles que : faible bruit référé à l'entrée < 1 µVpp, TRMC élevé > 110 dB, haute densité d'intégration, une faible consommation d'énergie et un faible coût global, etc. Ce dispositif offre la possibilité de mesurer les signaux EEG dans deux configurations différentes ; référentielle ou séquentielle, dont les avantages de chacune peuvent être exploités en fonction de l'application réalisée. Techniquement, l'ADS1299 est opérationnel à une vitesse de transmission allant de 250 SPS (*Sample Per Second*) à 16 KSPS et se compose d'un bloc analogique et d'un bloc numérique [Instruments T., 2017a]. Le bloc analogique comprend 8 amplificateurs à gain programmable, 8 convertisseurs analogique-numérique (ADC avec architecture ΔΣ), un générateur de référence interne, un amplificateur BIAS et un multiplexeur d'entrée. De plus, le bloc numérique intègre une unité de contrôle qui possède 24 registres de configuration, un oscillateur intégré et un contrôleur d'interface périphérique série SPI (*Serial Peripheral Interface*).

Les multiplexeurs d'entrée sont très flexibles et hautement configurables offrant de nombreuses fonctions pour le routage des différents signaux, par exemple, il y a des broches SRB1 (*Stimulus Reference Bias 1*) et SRB2 (*Stimulus Reference Bias 2*) qui sont utilisées pour router un signal commun à une ou plusieurs entrées pour une configuration référentielle, c'est-à-dire, si le bit SRB1 dans le registre MISC (*Miscellaneous*) est activé, alors toutes les entrées inversées (INxN) seront connectées, en interne, au signal de référence SRB1. En revanche, le bit SRB2 dans le registre CHnSET est unique pour chaque canal, ce qui permet à un signal externe d'être acheminé via la broche SRB2 vers une entrée non-inverseuse (INxP) avec un bit SRB2 activé. Enfin, l'acquisition des flux de données échantillonnés est réalisée via l'interface SPI avec des débits de données allant de 250 SPS à 16 KSPS. La figure III.3 montre le schéma fonctionnel de l'ADS1299-AFE [Instruments T., 2017a].

Ce composant est disponible dans un boîtier TQFP-64 dont les conditions de fonctionnement sont résumées dans le tableau III.3.

Valeurs / Caractéristiques	Conditions	Min.	Nom.	Max.
Plage de température de fonctionnement (°C)	-	-40	-	85
Alimentation analogique (V)	*AVDD to AVSS*[1]	4,75	5	5,25
Alimentation numérique (V)	*DVDD to DGND*[2]	1,8	1,8	3,6
Tension de référence (V)	*VREFP to VREFN*[3]	-	4,5	-
Gains de l'amplificateur PGA	-	1, 2, 4, 6, 8, 12 and 24		
Débit des données (éch./s)	f_{CLK} = *2,048MHz*	250, 500, 1K, 2K, 4K, 8K, 16K		
Résolution du CAN (bit)	-	24		
Fréquence Horloge (MHz)	*CLKSEL=0*	1,5	2,048	2,25
Fréquence du modulateur intégré (MHz)	-	f_{CLK} / 2		

Tableau III.3. Spécifications et conditions opérationnelles de l'ADS1299-AFE.

Notations

[1] AVDD est la tension analogique positive égale à +2,5 V (en alimentation bipolaire) ou +5,0 V (en alimentation unipolaire), AVSS est la tension analogique négative égale à -2,5 V (en alimentation bipolaire) ou 0 V (en alimentation unipolaire).

[2] DVDD est la tension numérique positive, DGND est la masse numérique.

[3] VREFP est la tension de référence positive, VREFN est la tension de référence négative.

La figure III.3 illustre l'organisation interne simplifié du frontal ADS1299-AFE :

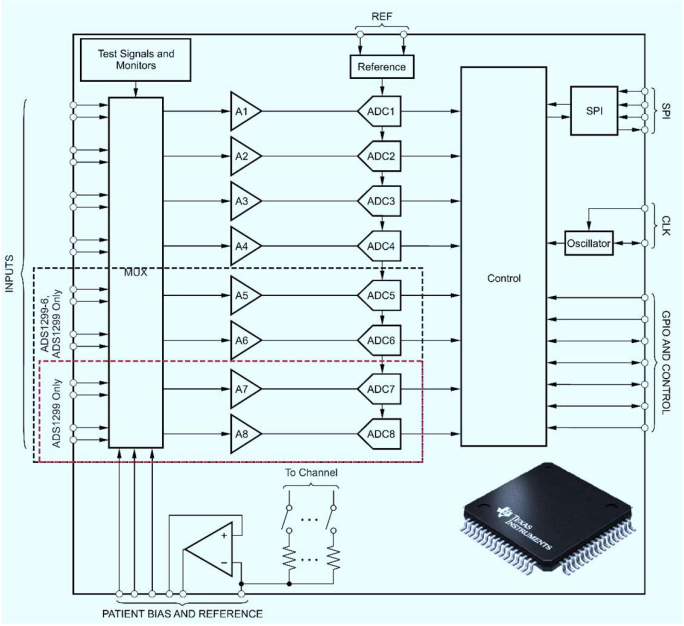


Figure III.3. Schéma fonctionnel simplifié de l'ADS1299-AFE. [Instruments T., 2023]

1) Fonctions des Pins

Certaines broches incluses dans le dispositif ont des fonctions très utiles pour concevoir l'application matérielle la plus basique (figure III.4.), ce sont les suivantes :

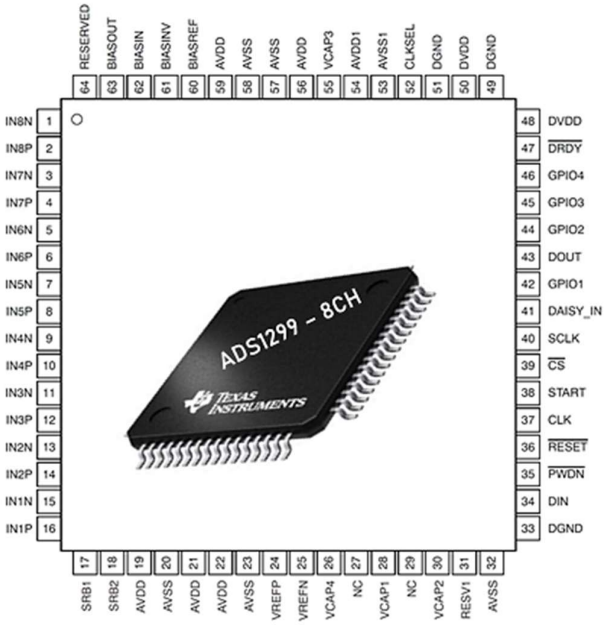


Figure III.4. Configuration des broches de l'ADS1299-AFE à 8 canaux.

- START : entrée du signal de synchronisation de (re)démarrage de la conversion.
- /RESET : Entrée de réinitialisation hard du système (active au front négatif).

- CLK : Entrée horloge maître.
- /CS (SPI) : Entrée de sélection (*Chip Select*).
- SCLK (SPI) : Entrée horloge série pour cadencer l'interface SPI.
- DIN (SPI) : entrée série des données.
- DOUT (SPI) : Sortie série des données.
- /DRDY (SPI) : Sortie d'état indiquant que les données sont prêtes (active au front négatif).
- INxN : Entrées différentielles analogiques négatives.
- INxP : Entrées différentielles analogiques positives.
- SRB1/2 : Entrée stimulus, Référence ou polarisation (Bias) 1/2.
- VREFN/VREFP : Entrée de référence analogique négative/positive.
- $V_{cap1}/V_{cap2}/V_{cap3}$: Broches des condensateurs de dérivation analogique.

2) Commande et configuration des registres internes de l'ADS1299-AFE

L'ADS1299-AFE à 8 canaux possède 24 registres internes polyvalents qui lui confèrent une flexibilité augmentée. Il est contrôlé via le protocole SPI (*Serial Peripheral Interface*) et supporte 10 commandes. Celles-ci sont données dans le tableau III.4, qui est dérivé de la page 40 de la fiche technique du dispositif [Instruments T., 2017a].

Commande	Premier Octet	Deuxième Octet	Description
Commandes du système			
WAKEUP	*0000 0010 (02H)*	-	Réveil à partir du mode veille.
STANDBY	*0000 0100 (04H)*	-	Entrer dans le mode veille.
RESET	*0000 0110 (06H)*	-	Réinitialiser le dispositif.
START	*0000 1000 (08H)*	-	démarrer et redémarrer (synchroniser) la conversion.
STOP	*0000 1010 (0AH)*	-	Arrêter la conversion.
Commandes de la lecture des données			
RDATAC	*0001 0000 (10H)*	-	Activer le mode de la lecture continue des données (Ce mode est par défaut à la mise sous tension).
SDATAC	*0001 0001 (11H)*	-	Arrêt de la lecture continue des données.
RDATA	*0001 0010 (12H)*	-	Lire les données par commande ; supporte la lecture multiple.
commandes de lecture/écriture des registres internes			
RREG	*001r rrrr (2xH)*	*000n nnnn*	Lecture de *n nnnn* registres à partir de l'adresse *r rrrr*
WREG	*010r rrrr (4xH)*	*000n nnnn*	Écriture de *n nnnn* registres à partir de l'adresse *r rrrr*

Tableau III.4. Définitions des commandes associées à l'ADS1299-AFE.

Notations

1- En mode RDATAC, la commande RREG est ignorée.

2- *n nnnn* = Nombre de registres à lire ou à écrire – 1.

3- *r rrrr* = Adresse du registre de départ pour les commandes de lecture ou d'écriture.

Les commandes (RREG/WREG) de Lecture/ Écriture des registres internes diffèrent des commandes du système et de la lecture des données. Ce sont des commandes de 2 octets ; le premier octet contient la commande lecture/écriture et l'adresse du registre, tandis que le deuxième octet de commande spécifie le nombre de registres à lire/écrire – 1. La commande WREG est toujours suivie d'au moins un octet supplémentaire, qui contient les données à écrire dans les registres sélectionnés. Lorsque le dispositif est en mode de lecture continue des données RDATAC (*Read Data Continuous*), une commande SDATAC (*Stop Read Data Continuous*) doit être émise avant que la commande RREG puisse être envoyée.

Considérons des exemples d'exécution des commandes RREG (figure III.5) et WREG (figure III.6), où :

- Pour lire 3 registres en commençant par le registre d'identité ID (00H), on envoie le 1er octet de commande 0010 000 puis le 2ème octet de commande 0000 0010, étant donné que 0000 0010 = (0000 00011 – 1) est le nombre des registres à lire – 1. Finalement, on lit les données via le pin DOUT.

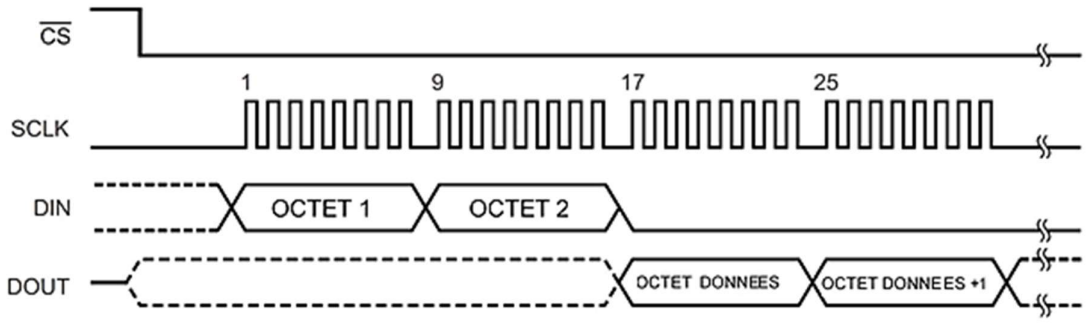


Figure III.5. Exécution de la commande RREG.

Le 17ème front montant SCLK de l'opération synchronise le bit de poids fort MSB (*Most Significant Bit*) du premier registre. Après SDATAC, la commande RREG peut être émise à tout moment et /CS doit être définie sur le niveau bas.

- Pour écrire les données binaires (0011 0000) dans les deux registres CH1SET d'adresse 05H et CH2SET d'adresse 06H, on envoie le 1er octet de commande 0100 0101 puis le 2ème octet de commande 0000 0001, étant donné que 0000 0001 = (0010 – 1) est le nombre des registres à écrire – 1. Finalement, on envoie les données à écrire au format MSB premier.

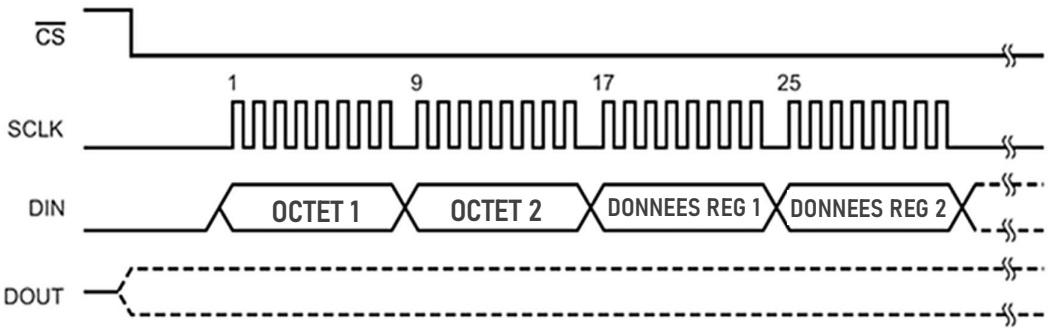


Figure III.6. Exécution de la commande WREG.

Notons que la commande WREG peut être émise à tout moment.

La commande RDATAC active la sortie de données converties sur chaque /DRDY sans qu'il soit nécessaire d'émettre d'autres commandes pour la lecture des données (RDATA ou RDATAC). Ce mode place les données de conversion dans le registre de décalage de sortie dans l'ADS1299 afin de pouvoir les acquérir par l'unité de contrôle et de commande. Le mode continu de lecture des données est le mode par défaut de l'appareil ; l'appareil utilise par défaut ce mode à la mise sous tension. Le mode RDATAC est annulé par la commande SDATAC.

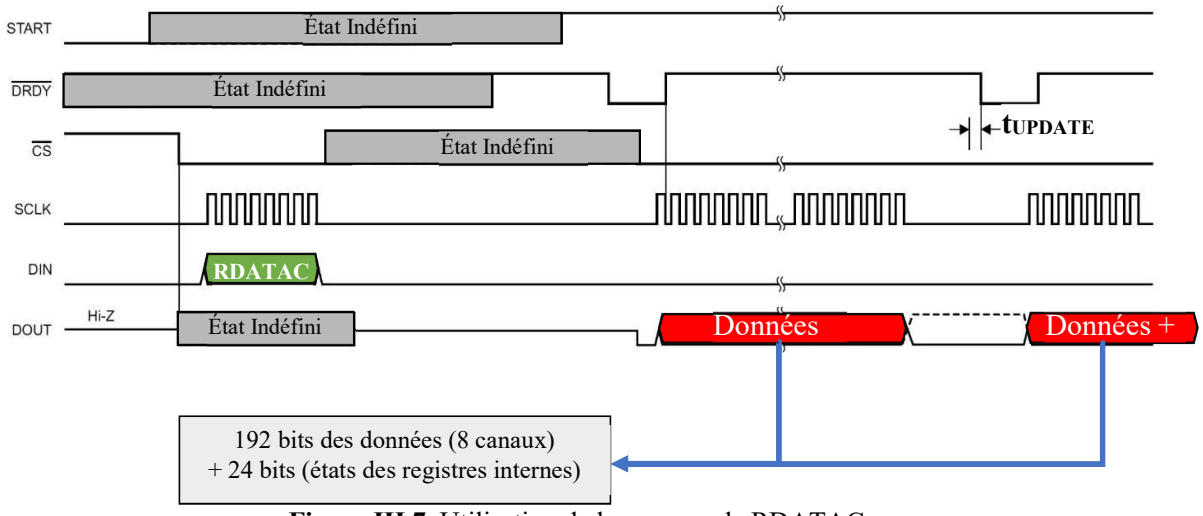


Figure III.7. Utilisation de la commande RDATAC.

Puisque l'ADS1299 est mis en mode de lecture continue, une seule commande START doit être envoyée. L'unité de contrôle reste en veille jusqu'à ce que l'ADS1299 active sa broche /DRDY (*data ready*), ce qui déclenche une interruption. Après avoir reçu cette interruption, l'unité de contrôle lit les 24 bits d'état des registres et les 192 bits des données reçues de 8 canaux de l'ADS1299, les stocke dans un tampon pour les mettre à la disposition du pilote, ensuite, elle se met simplement en veille et attend la prochaine assertion /DRDY.

III.3.1.3. Filtre passe-bas anti repliement

Le frontal analogique de l'ADS1299 contient un filtre de décimation numérique intégré, qui est un filtre Sinc passe-bas de 3ème ordre. Il prend le flux de données à la sortie du modulateur Delta-Sigma, à une fréquence très élevée f_{mod}=1,024 MHz, et l'ajuste au débit de données défini dans le registre CONFIG1 par le biais des bits [2 : 0] du registre DR à une vitesse de transmission de 250 SPS à 16 KSPS 250 SPS à 16 KSPS [Instruments T., 2017a]. Sa réponse fréquentielle est donnée par l'équation (III.1) :

$$T(f) = \left| \frac{\sin\left(\frac{f_D \pi f}{f_{\text{mod}}}\right)}{f_D \sin\left(\frac{\pi f}{f_{\text{mod}}}\right)} \right|^3 \qquad \text{(III.1)}$$

Où : f_D est le débit de données à la sortie du filtre de décimation.

Comme le montre la figure III.8, la fonction de transfert du filtre numérique se répète à chaque multiple de la fréquence d'échantillonnage. par conséquent la solution pour éviter le repliement consiste à limiter la bande des signaux d'entrée en dessous de la fréquence de Nyquist et ceci est fait en utilisant le filtre passe-bas anti repliement dont la fréquence de coupure f_c est calculée par l'équation (III.2), bien que ce filtre permet aussi d'éliminer le bruit hors bande qui se plie dans la bande d'intérêt et qu'il puisse également éliminer certaines interférences électromagnétiques EMI (*Electromagnetic Interference*) qui pourraient provenir de sources radiofréquences RF proches.

$$f_c = \frac{1}{2\pi R_{filt} C_{filt}} \qquad \text{(III.2)}$$

Où : R_{filt} est la résistance du filtre passe-bas.

C_{filt} est le condensateur du filtre passe-bas.

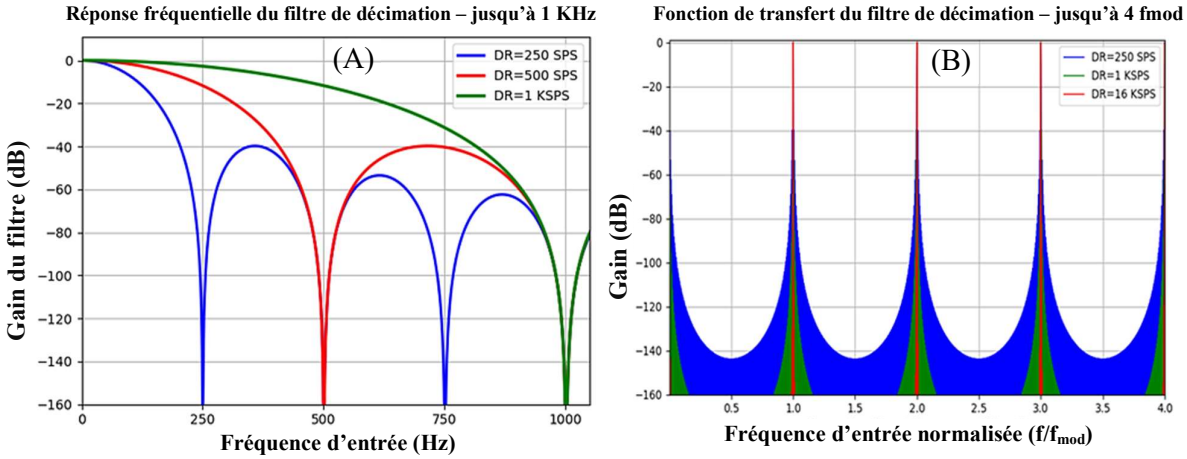


Figure III.8. (A) Réponse en fréquence (jusqu'à 1 KHz) du filtre Sinc à décimation, (B) sa fonction de transfert jusqu'à 4 f_{mod}.

III.3.1.4. Dispositifs de protection

La protection par les diodes de Suppression de tension transitoire TVS (*Transient Voltage Suppressor*) préserve l'ADS1299-AFE des décharges électrostatiques ESD (*Electro-Static Discharges*), car toutes les E/S câblées peuvent être soumises à des transitoires de haute tension provenant de différentes sources. Ces transitoires de durée suffisante peuvent facilement endommager tout circuit intégré s'il n'est pas protégé. Dans le système proposé, quatre suppresseurs SN65240P ont été mis en œuvre, ce sont des suppresseurs de tension transitoire (TVS) quadruples et unidirectionnels qui assurent une protection contre les interférences électriques, le courant de fuite des composants utilisés est très faible, jusqu'à 1 µA max à 6 V.

La figure III.9 montre un schéma de principe du suppresseur quadruple SN65240P :

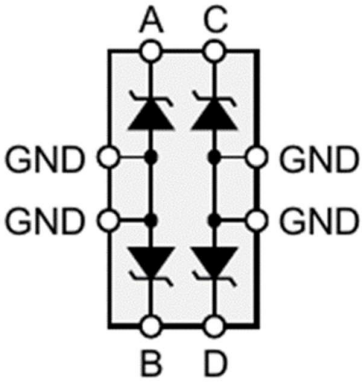


Figure III.9. Schéma interne du suppresseur ESD SN65240P.

Les diodes TVS internes du SN65240P fournissent une tension de claquage minimale de 6,5 V pour protéger les E/S, les émetteurs-récepteurs, typiquement implémentés en technologie

numérique CMOS 3 V ou 5 V. Le principe des suppresseurs ESD qui sont connectés en parallèle avec les E/S est de dériver l'énergie transitoire excessive vers la référence (rail d'alimentation, masse) lorsque la tension de claquage des diodes est dépassée en raison de la présence de transitoires haute tension. Certaines précautions doivent être prises lors de la conception de dispositifs à haute vitesse (qui opèrent à des hautes fréquences) car les suppresseurs peuvent avoir des effets dramatiques sur l'intégrité du signal.

III.3.1.5. Alimentation électrique analogique et numérique

Le rôle de cette partie est de fournir les tensions nécessaires aux autres blocs. Toutefois, afin d'assurer l'autonomie, une batterie de lithium (*Li-ion*) de 3,7 V, 3000 mAh a été utilisée comme source d'alimentation qui est le choix le plus préférable pour les appareils portables nécessitant une source électrique de coût réduit. Puisque le frontal analogique de l'ADS1299 nécessite deux types d'alimentation : numérique (DVDD) à 3,3 V qui est référencé à la masse (DGND) et analogique bipolaire (AVDD) de +2,5 V qui est référencé au retour analogique du dispositif (AVSS) de -2,5 V ; un ensemble de régulateurs linéaires à faible chute LDO (*Low Dropout*), caractérisés par leur faible courant de repos et leur excellente performance transitoire, ont été utilisés.

Cependant, le problème est qu' avec la batterie 3,7 V, il n'est pas possible d'alimenter directement les régulateurs TLV70333DBVR, TPS72325DBVR, LP5907MFX-2.5/NOPB TLV76733DGNR et l'inverseur LM2664M6/NOPB qui fournissent, respectivement, 3,3V, -2,5 V, +2,5 V, 3,3 V et -5 V, car les constructeurs de ces composants recommandent une tension d'alimentation égale à V_{out} + 1,5 V (~5 V) pour maintenir la tension de sortie bien régulée et que les performances dynamiques soient optimales [Instruments T., 2017b ; Instruments T., 2017c], pour faire face à cet obstacle nous avons introduit le MC34063, à très faible coût, comme convertisseur de tension DC/DC afin d'augmenter la tension de la batterie de 3,7 V à 5,5 V avec une capacité de délivrer un courant de sortie jusqu'à 500 mA.

Les régulateurs LDO sont choisis pour assurer la précision et la stabilité des tensions analogiques et numériques fournies aux autres blocs, car la précision des mesures effectuées par l'ADS1299-AFE est sensible aux petites fluctuations des valeurs des tensions analogiques (AVDD et AVSS) et numériques (DVDD et DGND) [Instruments T., 2016].

Le tableau III.5 résume les caractéristiques des régulateurs/inverseurs utilisés pour les différents rails d'alimentation à générer :

Spécifications / Régulateurs	V_{IN} (V)			V_{OUT} (V)			I_{OUT} (mA)		
	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX
TLV70333DBVR	2	-	5,5	3,3 - 2%	3,3 + 0,5%	3,3 + 2%	0	-	300
TLV76733DGNR	2,5	-	16	3,3 - 1%	-	3,3 + 1%	0	-	1000
TPS72325DBVR	-10	-	-2,7	-2,5 - 2%	-2,5 - 1%	-2,5 + 2%	0	-	200
LP5907MFX-2.5/NOPB	2,2	-	5,5	2,5 - 2%	-	2,5 + 2%	0	-	250
LM2664M6/NOPB	1,8	-	5,5	-5 * 99%	-5 * 99,9%	-	40	-	-

Tableau III.5. Conditions de fonctionnement des régulateurs de tension utilisés.

III.3.1.6. Unité de contrôle et de communication - UCC

L'unité de contrôle et de communication (UCC) est constituée d'un seul module contenant toutes les fonctions nécessaires au contrôle de l'ensemble du système. L'UCC tire parti du puissant microcontrôleur ESP32-D0WDQ6 doté d'un microprocesseur Dual-Core Xtensa LX6 à 32 bits cadencé jusqu'à 240 MHz [Espressif, 2020a].

Malgré sa très petite taille, l'UCC possède toutes les caractéristiques nécessaires qui assurent sa polyvalence, lui permettant de réaliser tout type d'application supportant plusieurs protocoles de communication tels que USB, Wi-Fi (avec connexions TCP/IP et/ou UDP sur IEE 802.11 b/g/n/e/i), Bluetooth (supporte BR/EDR 4.2 et Bluetooth LE), SPI et I2C, etc. [Maier A., 2017], depuis les petits réseaux de capteurs jusqu'à des tâches très complexes telles que : le codage de la voix, le traitement des photographies [Dokic K., 2020], les systèmes de réseaux intelligents [Viel F. et al., 2020], le système de mesure des vibrations embarqué et le traitement des données [Foltýnek P., 2019], qui nécessitent des ressources matérielles considérables. Cette unité est caractérisée par un haut niveau de robustesse, permettant un fonctionnement fiable dans un environnement industriel rigoureux [Espressif, 2020b].

La puissance de ce module réside dans le fait qu'il est capable de contrôler simultanément le système de mesure et de communiquer via Wi-Fi ou Bluetooth. Nous avons choisi ce composant sous la motivation de sa bonne portée sans fil, de sa faible consommation d'énergie, de son faible prix (~3,8$) comparé à certains concurrents : ChipKit (50$), Digi XBee (~40$) et Arduino-Due (~41,27$), et sa robustesse face à des conditions d'utilisation sévères [Espressif, 2020b].

De nombreux périphériques et interfaces ont été implémentés dans l'ESP32, tels que : capteur tactile, capteur de température, interface de carte SD, interface SPI haute vitesse, interfaces I^2C (*Inter-Integrated Circuit*), I^2S (*Inter-IC Sound*) et UART (*Universal Asynchronous Receiver-*

Transmitter). Étant donné que l'ESP32 peut être présenté comme un contrôleur maître qui gère d'autres dispositifs, il peut également agir comme un module esclave avec n'importe quel microcontrôleur ou carte de développement comme dispositif de communication. L'ESP32 est doté de capacités Wi-Fi et BLE4.0 intégrées, qui fournissent des solutions pour connecter de nombreux objets physiques, par exemple des capteurs, des cartes à puce à l'Internet, afin qu'ils puissent partager et collecter des données avec un minimum d'intervention humaine, ce que l'on appelle l'Internet des objets IoT (*Internet of Things*) [Maier A. et al., 2017].

En ce qui concerne sa programmation, l'ESP32 est équipé du système d'exploitation open-source FreeRTOS (*Free Real-Time Operating System*), qui est optimal pour les applications en temps réel ; en outre, il se caractérise par une grande flexibilité avec de multiples capacités logicielles :

- En Lua sous l'environnement LuaNode. [Shenzhen, 2016]
- En Micro-Python sous l'environnement de développement intégré (IDE) uPyCraft.
- En C/C++ sous Arduino IDE. [Arduino, 2020a]
- En C/C++ avec l'IoT Framework développé par la société ESPRESSIF. [Espressif, 2021a]

En concrétisant la démarche suivante : qualité - fiabilité - prix, l'ESP32 est le meilleur choix pour implémenter notre micro- logiciel. (voir la section III.3.2.2)

Le tableau III.6 montre la différence entre l'ESP32-WROOM-32 et ses quelques modules concurrents, ce qui résume la majorité des propriétés nécessaires qu'une UCC à usage général devrait avoir :

Caractéristiques / Module	*# Cores*	*Vitesse CPU*	*Mém. Flash*	*RAM/ SRAM*	*ROM*	*Blue-tooth*	*Ethernet*	*SPI*	*I²C*	*UART*	*Wi-Fi*	*USB*	*#CH DMA*	*Pins E/S*	*Tension D'alim.*	*Prix ($) 2020 Ebay.com*
ChipKIT Max32™	Single Core MX32-32bit	80MHz	512KB + 12KB Boot flash	128KB	-	-	10/100 Base-TX Mac	OUI	OUI	OUI	-	USB OTG2.0	6	83	3,3V	49,99
ESP32-WROOM-32	Dual Core Xtensa-LX6 32bit	Jusqu'à 240MHz	4MB (Flash Externe)	520KB+ 8KB RTC	448KB	V4.2 BR/ERD + BLE	OUI	OUI	OUI	OUI	802.11 b/g/n 2.4GHz ~ 2.5GHz	USB OTG2.0	13	48	3,0V à 3,6V	3,80
Arduino-Due	Single Core ARM Cortex-M3 32bit	84MHz	512KB	96KB	-	-	-	OUI	OUI	OUI	-	USB OTG2.0	12	54	3,3V	41,27
STM32F103C8T6	Single Core ARM Cortex-M3	72MHz	64KB jusqu'à 128KB	20KB	-	-	-	OUI	OUI	OUI	-	USB OTG2.0	7	80	2,0V à 3,6V	4,70

Tableau III.6. Comparaison de l'ESP 32 avec les modules concurrents les plus connus. [Digilent inc., 2020 ; Arduino, 2020b ; STMicroelectronics, 2020]

Certaines des tâches assignées l'UCC pour remplir sont :

- L'établissement des préréglages nécessaires pour le frontal analogique ADS1299 via le protocole SPI.
- L'acquisition, en temps réel, des données EEG mesurées-converties et les transférer vers le back-end à une vitesse de transmission allant jusqu'à 16000 échantillons par seconde.
- L'élaboration de la transmission filaire (USB) ou sans fil (Wi-Fi, Bluetooth) des données numérisées vers le back-end (par exemple RaspberryPi 4, PC hôte, …) pour le post-traitement attendu.
- La réception, pendant l'acquisition EEG, de commandes du back-end nécessitant la modification de certains paramètres (activation/désactivation de certains canaux, vérification de l'état des électrodes, modification du gain, etc.)

La figure III.10 décrit le schéma fonctionnel de l'ESP32-WROOM-32 :

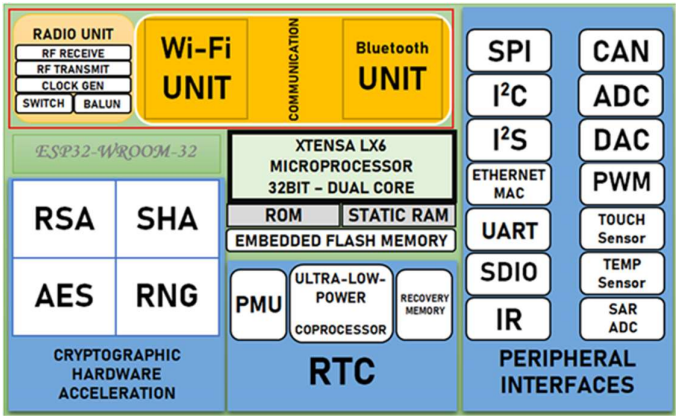


Figure III.10. Schéma fonctionnel de l'ESP32-WROOM-32.

III.3.2. Présentation du logiciel

Dans cette section, nous allons passer en revue tous les composants logiciels de la plateforme à la fois sur le front-end et le back-end (Figure III.11.).

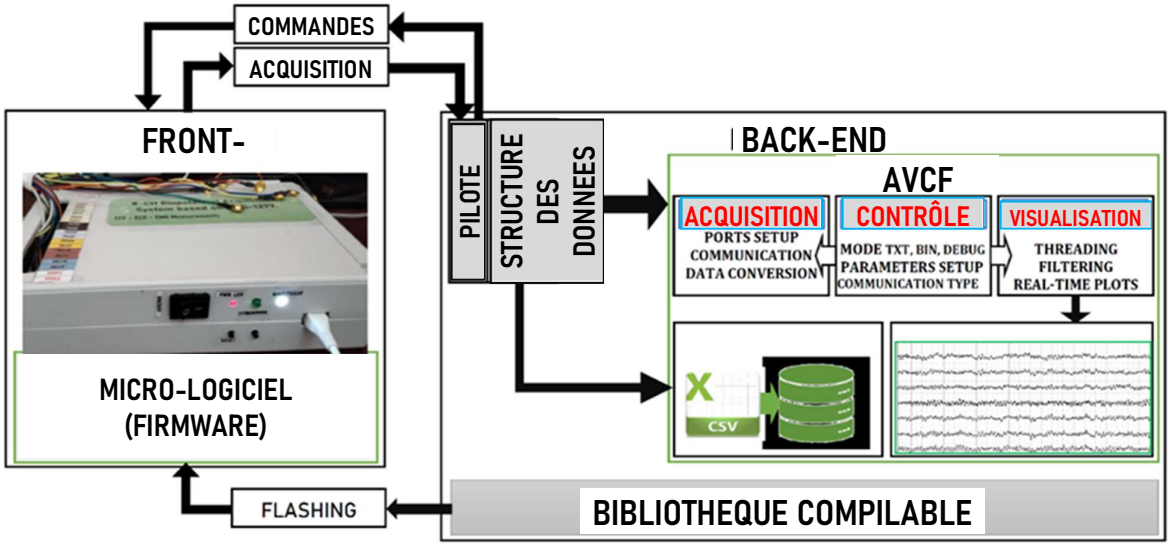


Figure III.11. Plateforme logicielle du système d'acquisition EEG sur le front-end et sur le back-end.

Cette section présente la conception de la bibliothèque, l'implémentation du Firmware et le cadre de streaming basé sur Python pour le système de mesure EEG basé sur l'ESP32. Le système global est caractérisé par la capacité de capturer et d'envoyer des données au back-end via l'interface USB (Universal Serial Bus) à une vitesse allant jusqu'à 16000 éch. /s, ou via l'interface sans fil (Bluetooth ou Wi-Fi) à une vitesse limitée à 250 éch. /s. La modélisation des solutions logicielles est une tâche importante dans la conception des systèmes embarqués et en temps réel. Il est donc important d'avoir une vision sur chaque étape du processus de modélisation. De plus, pour gérer la complexité du système, deux approches ont été considérées dans cette étude, l'une étant la séparation de la plate-forme en entités cohérentes qui peuvent être compilées séparément ; la seconde étant l'introduction de l'aspect objet (OA) qui rend l'extensibilité, la maintenance et la réutilisation des classes sous-jacentes une tâche relativement simple.

Malgré l'importance de concevoir une structure logicielle intégrée et optimale qui orchestre le bon fonctionnement du matériel (comportement synchronisé et en temps réel, signalement minimal des bogues, etc.), la plupart des études susmentionnées se sont contentées d'une brève présentation de leurs structures logicielles, ce qui, par conséquent, empêche les développeurs novices de concevoir leurs solutions en peu de temps.

Dans cette partie, l'accent a été mis sur la conception du logiciel. Le but de cette étude est de fournir un modèle exemplaire pour les développeurs de tels systèmes de mesure afin de

concevoir leurs propres plateformes d'une manière appropriée et optimale. Bien que nous ayons pris en compte le principe de portabilité, de flexibilité et de simplicité. Les sous sections suivantes traiteront la conception de la bibliothèque compilable, du micrologiciel (*Firmware*), et du Framework incluant le pilote (*Driver*).

III.3.2.1. Bibliothèque compilable

En raison du fait que le Framework ESP-IDF (*Espressif IoT Development Framework*) de l'ESP-32 est devenu compatible avec l'environnement Arduino IDE via le Core ESP32-Arduino [Arduino, 2020c], et afin de faciliter le développement du micrologiciel, une bibliothèque appelée BCI_LMSE_ESP a été créée afin qu'elle soit compatible avec l'IDE Arduino et qu'elle implémente les différentes routines nécessaires pour profiter des fonctionnalités incorporées dans l'ADS1299-AFE. Cette bibliothèque est développée en langage C/C++ par l'approche orientée objet OOP (OOP) [Kormanyos C., 2013] afin de donner certains aspects de modularité, réutilisabilité, lisibilité, etc. tandis que le Firmware a été réalisé en langage Arduino (un langage proche de C/C ++) puisque le Framework ESP-IDF est disponible dans l'environnement de développement via Arduino-Core pour l'ESP32 [Espressif, 2021b].

Comme illustrée par la figure III.12, la bibliothèque développée contient trois fichiers :

- Un fichier d'en-tête "BCI_LMSE_ESP.hpp" qui contient les déclarations de toutes les classes, les méthodes et les attributs utilisés.
- Un autre fichier d'en-tête "DEFINITIONS.hpp" qui comprend la définition des constantes, des adresses mémoire, des masques binaires, des valeurs des différentes configurations de registres internes de l'ADS1299-AFE, etc.
- Un fichier source "BCI_LMSE_ESP.cpp" qui intègre l'implémentation de toutes les méthodes définies dans les en-têtes.

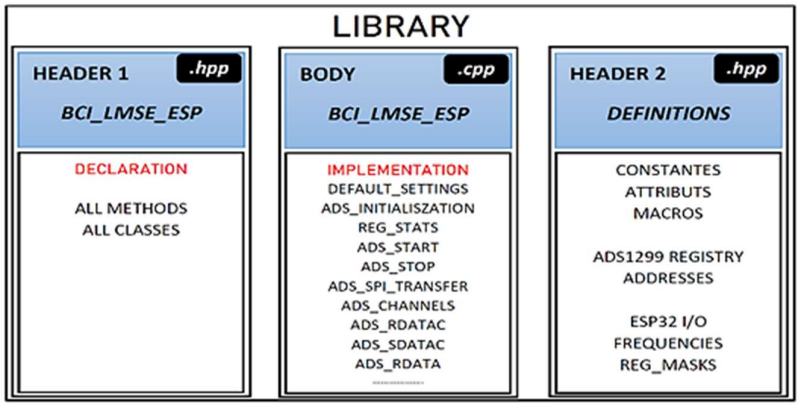


Figure III.12. Organisation de la bibliothèque compilable.

Afin d'assurer une configuration de base correcte, sans anomalies ni décalages, les instructions concernant les restrictions temporelles à respecter, qui sont fournies avec la fiche technique de l'ADS1299-AFE, ont été strictement suivies. La figure III.13 décrit la séquence d'initialisation du frontal contrôlé par le l'UCC basée sur ESP32.

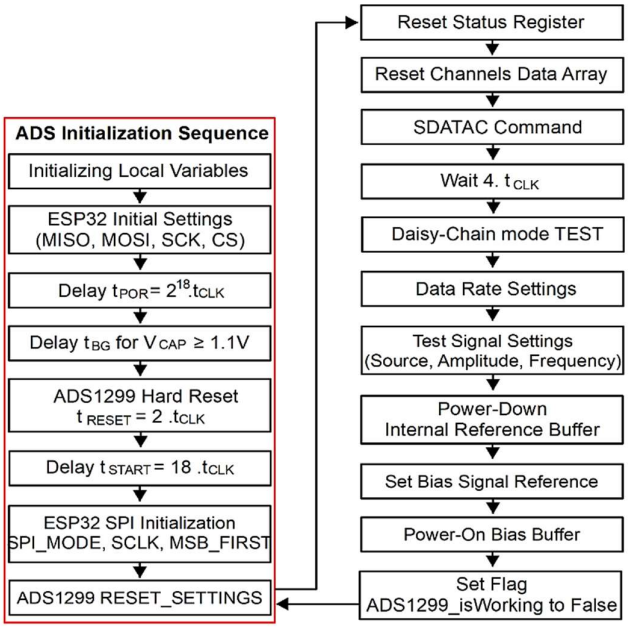


Figure III.13. Séquence d'initialisation du système d'acquisition EEG basé sur l'ESP32.

III.3.2.2. Firmware

Le micrologiciel implémenté dans l'UCC utilise l'interface de communication SPI en mode 1 (polarité d'horloge CPOL= 0, phase d'horloge CPHA= 1) [Instruments T., 2017a]. Les données sont capturées au niveau de DIN (*Master Out Slave In*) sur le front descendant de SCLK et sont transférées hors de DOUT (*Master In Slave Out*) sur le front montant de SCLK, où SCLK est le signal d'horloge série généré par l'UCC. La figure III.14 montre un organigramme du micrologiciel en cours d'exécution, en tenant compte du fait que trois modes de fonctionnement ont été pris en charge (*DEBUG*, *TEXT* et *BINARY*) et qu'ils sont discutés en détail dans la sous-section sur les pilotes.

Donc, La procédure minimale et suffisante pour la conversion et la lecture continue des données EEG est présentée dans l'organigramme de la figure III.14 où :

t_{BG} est le temps nécessaire pour que la tension à la sortie analogique (broche 28 du frontal) atteigne 1,1V (par rapport à V_{REFN}).

- t_{CLK} est la période du signal d'horloge.

- SDATAC est un mode inhérent qui signifie l'arrêt de la lecture continue (*Stop DATA Continuous*).
- RDATAC est l'opposé de SDATAC, est un mode frontal relatif qui signifie la lecture continue des données (*Read DATA Continuous*).

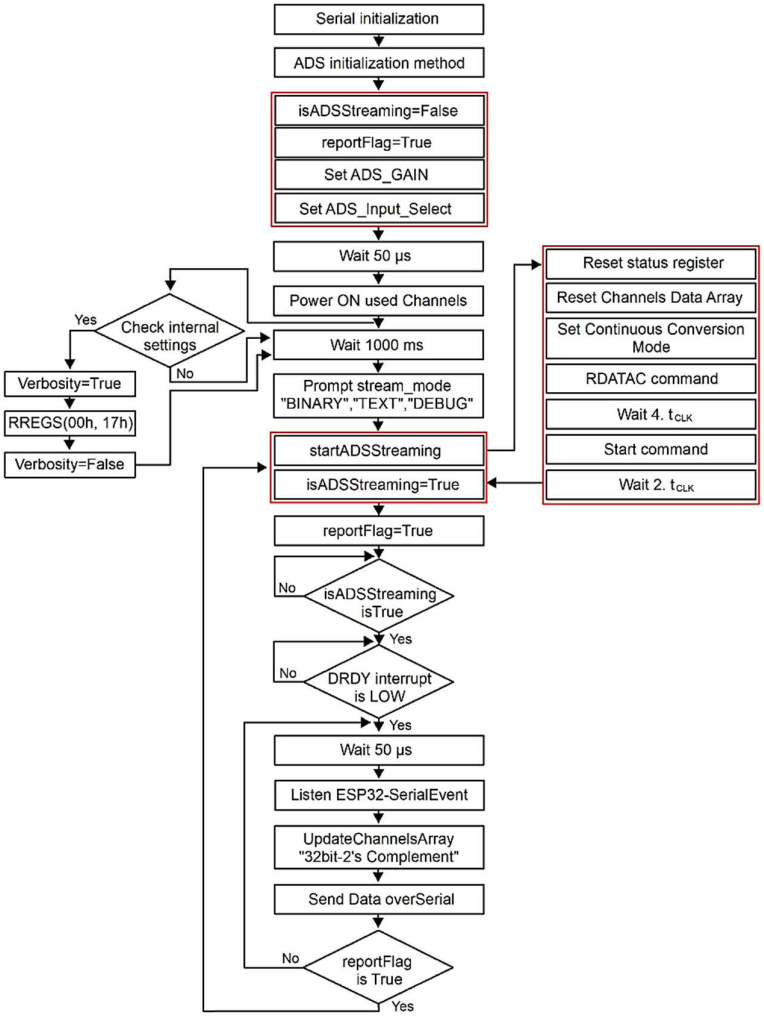


Figure III.14. Organigramme du flux de données EEG en modes de conversion continue et de lecture continue.

Cette séquence d'opérations place le dispositif en mode de flux continu. Les opérations commencent par la réservation d'un port série local, puis sa mise à disposition du dispositif. Ensuite, l'unité de contrôle envoie les commandes d'initialisation à l'ADS1299 tout en respectant les restrictions de temps recommandées par le fabricant [Instruments T., 2017a – pp. 38 à 42]. Lors de la phase initiale, il est nécessaire de déterminer les canaux qui seront utilisés, ainsi que le gain des amplificateurs intégrés.

On remarque simplement que des délais nécessaires sont introduits entre les opérations consécutives pour s'assurer que l'appareil est dans le bon état. Il convient de mentionner que les paramètres internes de l'appareil doivent être vérifiés avant de lancer le streaming afin d'éviter une mauvaise lecture des données. Enfin, la procédure accède à la boucle de streaming, sachant que, pour chaque impulsion d'entrée négative /DRDY, les données capturées sont converties et stockées dans les registres de sortie. L'UCC prend des mesures pour pousser les données tamponnées vers le back-end et libère ainsi les registres pour le flux suivant.

III.3.2.3. Pilote

Il s'agit de l'élément le plus important du logiciel, dont le rôle est de restructurer les données récupérées par le micrologiciel. Cette entité logicielle a été implémentée dans le back-end suivant 3 modes de fonctionnement tels que :

Le mode de débogage appelé "*DEBUG*" ; c'est un mode texte qui consiste à placer le dispositif dans un état qui affiche la configuration de tous ses registres internes afin d'assurer son bon fonctionnement avec les préréglages sélectionnés.

Le mode texte appelé "*TEXT*" est le mode le plus simple car il est basé sur l'inscription directe dans un tableau des données EEG mesurées, des indices et des séparateurs d'échantillons, de plus ce mode a l'avantage que les données sont clairement lisibles par l'utilisateur, alors que l'inconvénient majeur réside dans les erreurs qui peuvent se produire lors du streaming puisqu'il ne dépend pas des octets de contrôle, ce mode est essentiellement utilisé pour tester la cohérence du Firmware implémenté dans l'ESP32.

Le mode binaire appelé "*BINARY*" est un mode opérationnel qui permet de propager des données sur le port série au format binaire, c'est-à-dire en envoyant une série d'octets tamponnés. Ce mode est basé sur l'envoi d'un paquet de 35 octets pour chaque échantillon sur 8 canaux, le pilote à l'arrière peut facilement détecter l'en-tête (0xBB) suivi d'un compteur d'échantillons, il localise également les 32 octets suivants (ayant des valeurs signées avec le MSB en premier) afin d'extraire les données EEG des 8 canaux, enfin le pilote recherche un octet de postambule (0xEE) utilisé pour détecter la fin du paquet envoyé pour procéder à l'analyse de l'échantillon suivant. En termes généraux, 4N + 3 octets peuvent être reçus pour chaque échantillon mesuré, N étant le nombre de canaux supportés par le système.

La synchronisation du flux de données à la sortie du tampon dépend de l'état de l'interruption /DRDY émise par l'ADS1299-AFE, qui indique qu'un nouveau paquet est prêt à être transféré dont la période est déterminée par la fréquence d'échantillonnage.

III.3.2.4. Cadre d'acquisition, de visualisation et de contrôle-AVCF

Au niveau du back-end et comme illustré dans la Figure III.11, trois applications logicielles ont été développées en utilisant le langage Python 3, celles-ci sont utilisées pour acquérir, visualiser les données et ensuite contrôler toutes les opérations sous-jacentes, elles ont été testées sur un RaspberryPi 4 (CPU- ARMv7 32-bit, 4GB-RAM) [Raspberry Pi, 2021] exécutant un système d'exploitation Ubuntu MATE compatible avec l'architecture armhf et sur un PC (Dell Inspiron-15 CPU-Core i3, 8G-RAM, GPU-2GB-Nvidia GeForce 820M) exécutant Microsoft Windows 10, les deux plateformes étaient équipées de la version 3.8.5 du langage Python.

La première application, nommée *'liveCSVstreaming.py',* est équipée d'un pilote capable d'interagir avec le matériel pour localiser les données EEG brutes envoyées au port série sous forme de blocs d'octets contigus. Cette application prépare les interfaces (par exemple, le port série, les tampons, etc.), initialise les fichiers de stockage et, enfin, stocke les données acquises après avoir effectué la conversion en valeurs réelles (c'est-à-dire en microvolts). Elle permet également de dissocier ces séries d'octets dans des structures Python à partir du tampon, puis les données sont converties en valeurs décimales avant d'être finalement stockées sous forme de mesures réelles exprimées en microvolts. Le stockage est effectué dans un fichier initialisé et d'une taille allouée dynamiquement, la conversion établie pour obtenir les valeurs en microvolts est décrite par l'équation (III.3) :

$$V\left[uV\right] = 2\frac{V_{dec}\left(\frac{V_{ref}}{Gain}\right)}{2^{24}-1}10^{6} \qquad \text{(III.3)}$$

- V : est la valeur réelle de la tension exprimée en microvolts.
- V_{dec} : est la représentation décimale des données binaires reçues pour chaque trame.
- V_{ref} : est la tension de référence de l'ADS1299-AFE, typiquement égale à 4,5 V.
- Gain : égal à 1, 2, 4, 6, 8, 12 ou 24.

La deuxième application appelée, '*plotCSVStreaming.py',* est utilisée pour visualiser les données brutes converties à travers une interface graphique GUI (*Graphical User Interface*), le développement de cette structure est basé sur des modules python : PyQt5 et pyqtgraph [Fitzpatrick, M., 2019], cette application autant qu'elle intègre des filtres passe-bas, passe-haut, passe-bande, Notch, elle intègre également des scripts de calcul de densité spectrale de puissance et de périodogramme de Welch, etc. pour les données de séries temporelles.

La troisième application s'appelle '*AcquisitionSystemControl.py'*, celle-ci agit comme un noyau autour duquel différentes opérations de contrôle peuvent être effectuées dans les deux applications susmentionnées, on signale à titre d'exemple : l'envoi de commandes sur le port série à travers le driver implémenté dans '*liveCSVStreaming.py'*, la définition de nouveaux paramètres (arrêt de certains canaux, modification du gain, entrée en mode débogage, etc.), la bascule entre les filtres et la modification du type de spectres dans '*plotCSVStreaming.py'*, etc.

III.4. Conclusion

Dans ce chapitre, un système d'acquisition EEG basé sur l'ADS1299-AFE a été implémenté pour être utilisé dans une application Interface Cerveau-Ordinateur BCI. La partie d'acquisition et de conversion a été interfacée avec le module de contrôle et de communication basé sur l'ESP32-WROOM-32 via le protocole de communication SPI, la logique de conception des bibliothèques logicielles a été discutée, le micrologiciel (*Firmware*) du système est développé en utilisant le langage de programmation C/C++ qui intègre différentes fonctions qui pilotent l'acquisition en temps réel des signaux biopotentiels, les données capturées sont à la fois enregistrées au format (*.txt) sur un support de stockage local µSD et puis aux formats (*.txt ou *.csv) sur une mémoire distante (PC, RaspberryPi 4, etc.); ces deux formats sont standards et sont souvent adaptés pour un large éventail de plateformes logicielles tels que : les bibliothèques Python, la boite à outils MATLAB, le langage R, l'Octave GNU [GNU Octave, 2021], le logiciel Xplenty [Xplenty, 2021], etc. Désormais, grâce à la fonctionnalité "*Daisy Chain*" disponible sur le front-end ADS1299-AFE, la synchronisation de plusieurs dispositifs est possible.

Maintenant que nous avons introduit ces technologies dans cette conception, nous pensons qu'elle est qualifiée pour être modulaire, configurable et extensible, de sorte qu'elle peut facilement supporter des améliorations dans son architecture et surtout dans sa capacité d'empilement modulaire qui permet d'augmenter le nombre de ses canaux utilisables et par conséquent sa résolution spatiale. L'enregistrement et la visualisation en temps réel des signaux EEG ont été effectués via une plateforme spécialisée que nous avons développée en langage Python à cet effet. En outre, la portabilité du dispositif est assurée par la communication sans fil et par l'alimentation avec une batterie Li-ion de 3000 mAh de petite taille qui offre une excellente autonomie au système, de plus le choix de composants de surface et précis nous a permis de construire un dispositif relativement miniaturisé.

BIBLIOGRAPHIE DU CHAPITRE III :

Akin, T., Najafi, K., & Bradley, R. M. (1998). A wireless implantable multichannel digital neural recording system for a micromachined sieve electrode. IEEE Journal of Solid-State Circuits, 33(1), 109.

Albulbul, A. (2016). Evaluating major electrode types for idle biological signal measurements for modern medical technology. *Bioengineering*, *3*(3), 20.

Arduino, Learn Arduino (2020), https://www.arduino.cc/en/Guide. (accessed on 27 Feb 2023)

Arduino Due — Overview, 2020, https://store.arduino.cc/arduino-due. (accessed on 27 Feb 2023)

arduino-esp32 — Arduino core for the ESP32, 2020, https://github.com/espressif/arduino-esp32. (accessed on 05 Mar 2023)

Cencen, V., Hirotani, M., & Chan, A. D. (2016, May). Comparison of active and passive electrodes in their optimized electroencephalography amplifier system. In 2016 IEEE EMBS International Student Conference (ISC) (pp. 1-4). IEEE.

Chen, Y. H., De Beeck, M. O., Vanderheyden, L., Carrette, E., Mihajlović, V., Vanstreels, K., ... & Van Hoof, C. (2014). Soft, comfortable polymer dry electrodes for high quality ECG and EEG recording. Sensors, 14(12), 23758-23780.

, Digilent, chip KIT Max32 — Overview, 2020, https://reference.digilentinc.com /chipkit_max32/refmanual.(accessed on 04 Mar 2023)

Dokic, K. (2020). Microcontrollers on the edge–is esp32 with camera ready for machine learning? In *Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4–6, 2020, Proceedings 9* (pp. 213-220). Springer International Publishing.

Dumermuth, G., & Dinkelmann, R. (1979). EEG data acquisition and preprocessing by microcomputer satellite system. Computer Programs in Biomedicine, 10(3), 197-208. https://doi.org/10.1016/0010-468X(79)90069-2.

Dunseath, W. R., & Kelly, E. F. (1995). Multichannel PC-based data-acquisition system for high-resolution EEG. *IEEE Transactions on Biomedical Engineering*, *42*(12), 1212-1217.

Espressif, ESP32-WROOM-32 Datasheet (2020a), https://www.espressif.com /en/support/documents/technical-documents.(accessed on 25 Feb 2023)

Espressif ESP32 — Robust Design, 2020, https://www.espressif.com /en/products/socs/esp32. (accessed on 25 Feb 2023)

Espressif, Official IoT Development Framework (2021a), https://www.espressif.com /en/products/sdks/esp-idf. (accessed on 25 Feb 2023)

Espressif, Arduino Core for the ESP32 (2021), https://github.com/espressif/arduino-esp32.

Fitzpatrick, M. (2019). Create Simple GUI Applications with Python & Qt5. *Leanpub book,*

Foltýnek, P., Babiuch, M., & Šuránek, P. (2019). Measurement and data processing from Internet of Things modules by dual-core application using ESP32 board. *Measurement and Control*, *52*(7-8), 970-984.

GNU Octave Manual Version 6.3., 2021, https://www.gnu.org /software/octave/doc/v6.3.0/ (accessed on 12 Mar 2023)

https://imotions.com/blog/top-14-eeg-hardware-companies-ranked/, 2017.

Texas Instruments, ADS1299 Performance Demonstration Kit, User guide (2016), ttps://www.ti.com/tool/ADS1299EEGFE-PDK#tech-docs. (accessed on 25 Feb 2023)

Instruments, T. (Revised January 2017). Low-noise, 8-channel, 24-bit analog front-end for biopotential measurements ADS1299. *ADS1299 Datasheet*. (accessed on 04 Feb 2023)

Texas Instruments, TLV703 300-mA, Low-IQ, Low-Dropout Regulator (2017), https://www.ti.com/product/TLV703. (accessed on 08 Feb 2023)

Texas Instruments, TLV767 Adjustable- and fixed-output, 1-A, 16-V, positive-voltage low-dropout (LDO) linear regulator (2017), https://www.ti.com/product/TLV703. (accessed on 08 Feb 2023)

Texas Instruments, ADS1299 — ADS1299 Product details, Accessed 15 Jan 2023, https://www.ti.com/product/ADS1299. (accessed on 08 Feb 2023)

Ives, J. R., Thompson, C. J., & Woods, J. F. (1973). Acquisition by telemetry and computer analysis of 4-channel long-term EEG recordings from patients subject to "petit-mal" absence attacks. *Electroencephalography and clinical neurophysiology*, *34*(6), 665-668.

Kormanyos, C. (2013). *Real-time C++: efficient object-oriented and template microcontroller programming* (pp. I-XXIII). Berlin, Germany: Springer.

Kumar, M. A., & Chari, K. M. (2018). Efficient FPGA-based VLSI architecture for detecting R-peaks in electrocardiogram signal by combining Shannon energy with Hilbert transform. IET Signal Processing, 12(7), 879-888.

Lee, S., Cho, H., Kim, K., & Jun, S. C. (2019). Simultaneous EEG acquisition system for multiple users: development and related issues. Sensors, 19(20), 4592.

Lee, S., & Kruse, J. (2008). Biopotential electrode sensors in ECG/EEG/EMG systems. *Analog Devices*, *200*, 1-2.

Li, Y. X., Li, N. Q., & Liu, Y. (2014). A New Design of Portable EEG Signal Acquisition System. In *Advanced Materials Research* (Vol. 846, pp. 804-807). Trans Tech Publications Ltd.

Liyanage, S. R., Guan, C., Zhang, H., Ang K. K., Xu, J., Lee T. H. Dynamically weighted ensemble classification for non-stationary EEG processing. J Neural Eng. 2013 Jun ;10(3):036007.

Lo, C. C., Chien, T. Y., Chen, Y. C., Tsai, S. H., Fang, W. C., & Lin, B. S. (2016). A wearable channel selection-based brain-computer interface for motor imagery detection. Sensors, 16(2), 213.

Ma, R., Kim, D. H., McCormick, M., Coleman, T., & Rogers, J. (2010, August). A stretchable electrode array for non-invasive, skin-mounted measurement of electrocardiography (ECG), electromyography (EMG) and electroencephalography (EEG). In 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology (pp. 6405-6408). IEEE.

Maier, A., Sharp, A., & Vagapov, Y. (2017, September). Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. In *2017 Internet Technologies and Applications (ITA)* (pp. 143-148). IEEE.

Mastinu, E., Ortiz-Catalan, M., & Håkansson, B. (2015, August). Analog front-ends comparison in the way of a portable, low-power and low-cost EMG controller based on pattern recognition. In *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)* (pp. 2111-2114). IEEE.

Mastinu, E., Håkansson, B., & Ortiz-Catalan, M. (2017, May). Low-cost, open source bioelectric signal acquisition system. In *2017 IEEE 14th International Conference on Wearable and Implantable Body Sensor Networks (BSN)* (pp. 19-22). IEEE.

https://www.medicalexpo.com/medical-manufacturer/eeg-system-2716.html, 2020.

Minglong, Y., Qingsong, A., & Quan, L. (2012, July). Design of a high-performance EEG acquisition system for unshielded environment. In *Proceedings of 2012 IEEE/ASME 8th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications* (pp. 202-206). IEEE.

Pinho, F., Correia, J. H., Sousa, N. J., Cerqueira, J. J., & Dias, N. S. (2014, May). Wireless and wearable EEG acquisition platform for ambulatory monitoring. In 2014 IEEE 3nd International Conference on Serious Games and Applications for Health (SeGAH) (pp. 1-7). IEEE.

Posada-Quintero, H. F., Rood, R. T., Burnham, K., Pennace, J., & Chon, K. H. (2016). Assessment of carbon/salt/adhesive electrodes for surface electromyography measurements. *IEEE journal of translational engineering in health and medicine*, *4*, 1-9.

Rashid, U., Niazi, I. K., Signal, N., & Taylor, D. (2018). An EEG experimental study evaluating the performance of Texas instruments ADS1299. *Sensors*, *18*(11), 3721.

Raspberry Pi, Getting started, 2021, https://www.raspberrypi.com /products/raspberry-pi-4-model-b. (accessed on 12 Mar 2023)

Salvo, P., Raedt, R., Carrette, E., Schaubroeck, D., Vanfleteren, J., & Cardon, L. (2012). A 3D printed dry electrode for ECG/EEG recording. *Sensors and Actuators A: Physical*, *174*, 96-102.

Sanei, S., & Chambers, J. A. (2013). *EEG signal processing*. John Wiley & Sons.

Shenzhen Doctors of Intelligence & Technology DOIT, LuaNode ESP32 Document v1.0 (2016), https://cdn.hackada.io/files/18666798606144/LuaNode.pdf.(accessed on 03 Mar 2023)

STMicroelectronics, STM32F103C8 — Overview, 2020, https://www.st.com/en/microcontrollers-microprocessors/stm32f103c8.html. (accessed on 04 Mar 2023)

Tayeb, Z., Waniek, N., Fedjaev, J., Ghaboosi, N., Rychly, L., Widderich, C., Richter, C., Braun, J., Saveriano, M., & Cheng, G. (2018). Gumpy: A Python toolbox suitable for hybrid brain-computer interfaces. Journal of Neural Engineering, 15(6), 065003. https://doi.org/10.1088/1741-2552/aae186.

Toresano, L. O. H. Z., Wijaya, S. K., Prawito, Sudarmaji, A., & Badri, C. (2017, July). Data acquisition system of 16-channel EEG based on ATSAM3X8E ARM Cortex-M3 32-bit microcontroller and ADS1299. In *AIP Conference Proceedings* (Vol. 1862, No. 1, p. 030149). AIP Publishing LLC.

Uktveris, T., & Jusas, V. (2018). Development of a modular board for EEG signal acquisition. *Sensors*, *18*(7), 2140.

Venthur, B., Dähne, S., Höhne, J. et al. Wyrm: A Brain-Computer Interface Toolbox in Python. Neuroinform 13, 471–486 (2015). https://doi.org/10.1007/s12021-015-9271-8

Verma, N., Shoeb, A., Bohorquez, J., Dawson, J., Guttag, J., & Chandrakasan, A. P. (2010). A micro-power EEG acquisition SoC with integrated feature extraction processor for a chronic seizure detection system. IEEE journal of solid-state circuits, 45(4), 804-816.

Viel, F., Augusto Silva, L., Leithardt, V. R. Q., De Paz Santana, J. F., Celeste Ghizoni Teive, R., & Albenes Zeferino, C. (2020). An Efficient Interface for the Integration of IoT Devices with Smart Grids. *Sensors*, *20*(10), 2849.

Vo, T. D., Dwyer, G., & Szeto, H. H. (1986). A distributed microcomputer-controlled system for data acquisition and power spectral analysis of EEG. *Journal of neuroscience methods*, *16*(2), 141-148.

Xplenty: Data Integration Platform, 2021, https://www.xplenty.com/ (accessed on 12 Mar 2023)

Xu, J., Mitra, S., Matsumoto, A., Patki, S., Van Hoof, C., Makinwa, K. A., & Yazicioglu, R. F. (2014). A wearable 8-channel active-electrode EEG/ETI acquisition system for body area networks. *IEEE Journal of Solid-State Circuits*, *49*(9), 2005-2016.

Yazicioglu, R. F., Merken, P., & Van Hoof, C. (2005). Integrated low-power 24-channel EEG front-end. Electronics Letters, 41(8), 457-458.

Zhao, Z., Ivanov, K., Lubich, L., Omisore, O. M., Mei, Z., Fu, N., ... & Wang, L. (2018, July). Signal quality and electrode-skin impedance evaluation in the context of wearable electroencephalographic systems. In 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 4965-4968). IEEE.

CHAPITRE IV

ÉVALUATION GLOBALE DU SYSTEME D'ACQUISITION EEG

CONTENU

IV.1. Introduction

IV.2. Analyse des performances du système d'acquisition

IV.2.1. Bruit référé à l'entrée

IV.2.2. Mesure de la température inhérente

IV.2.3. Mesure du décalage de la composante continue (Offset-CC)

IV.2.4. Étalonnage à l'aide du signal de test interne

IV.2.5. Taux de réjection en mode commun (TRMC)

IV.2.6. Rapport de performance comparé à un modèle de référence

IV.3. Mesure du signal ECG

IV.4. Mesure du signal EEG

IV.5. Mesure du signal EMG

IV.6. Conclusion

Bibliographie du Chapitre IV

IV.1. Introduction

Dans ce chapitre, nous proposons un protocole expérimental pour évaluer les performances et le bon fonctionnement du dispositif dans un environnement non protégé. La figure IV.1 montre un prototype expérimental du système proposé.



Figure IV.1. Prototype proposé.

L'importance de cette partie réside dans le fait qu'elle doit pouvoir donner un résultat informatif sur le comportement du système, son efficacité, sa précision et en particulier les informations sur la qualité des signaux acquis. Plusieurs caractéristiques et paramètres qui déterminent les performances du système ont dû être inspectés, tels que : le taux de réjection en mode commun (TRMC), le bruit référé à l'entrée, l'étalonnage par rapport aux signaux de test, etc. Au lieu d'introduire des solutions commerciales relativement coûteuses qui sont utilisées dans l'évaluation des performances, tous les tests seront effectués en utilisant une approche software via un ensemble de scripts développés en Python sous l'environnement de développement Anaconda-Distribution [Anaconda, 2020]. Ce dernier autant qu'il est utilisé en programmation, il peut également être utilisé en sciences des données.

Deux types de tests ont été réalisés durant cette phase :

- Test des performances du système.
- Test de qualité des signaux EEG, ECG et EMG mesurés.

Le tableau IV.1 résume les conditions dans lesquelles les tests ont été effectués.

Paramètres	Conditions	Min.	Nom.	Max.	Mesurées
Plage de température de fonctionnement (°C)	-	-40	T_A=25	85	28
Alimentation analogique (V)	*AVDD to AVSS*[1]	4,75	5	5,25	5,03
Alimentation numérique (V)	*DVDD to DGND*[2]	1,80	1,80	3,6	3,34
Alimentation analogique à numérique	*AVDD to DVDD*	-2,1	-	3,6	-0,81
Tension de référence (V)	*VREFP to VREFN*[3]	-	4,50	-	4,53
Tension de référence positive (V)	*VREFP to DGND*	-	2	-	2
Tension de référence négative (V)	*VREFN to DGND*	-	-2,50	-	-2,53

Tableau IV.1. Conditions d'évaluation expérimentale de l'ADS1299-AFE.

IV.2. Analyse des performances du système d'acquisition

IV.2.1. Bruit référé à l'entrée

Le bruit référé à l'entrée est une caractéristique inhérente du frontal analogique qui identifie les composantes indésirables présentes dans le système indépendamment du signal d'entrée. En d'autres termes, il s'agit de la tension du bruit lorsqu'il est appliqué à l'entrée d'un circuit sans bruit produira la même chose qu'un circuit avec un bruit propre. Cette caractéristique est essentielle pour connaître la contribution du circuit au bruit global du système [Instruments T., 2005]. L'analyse du bruit référé à l'entrée a été réalisée en ajustant le registre CHnSET [2 : 0] à 001 (Par un court-circuit interne des entrées de l'ADS1299-AFE). Les signaux du bruit référé à l'entrée ont été échantillonnés pour toutes les valeurs possibles du gain (i.e. 1, 2, ... 24) et avec un taux de transfert des données fixe (*Data Rate*) DR=250 éch. /s. L'acquisition des séries temporelles est effectuée sur un intervalle de 12 secondes, nous avons donc collecté 3000 échantillons avec le taux de transfert défini.

Compte tenu de la nature non déterministe des signaux de bruit, nous avons adopté une méthode utile appelée KDE (*Kernel Density Estimate*) pour nous permettre de visualiser comment la densité du bruit varie en fonction de l'amplitude du signal. Il s'agit d'une estimation de la distribution des niveaux de bruit le long d'une courbe continue interprétable. Cette densité est souvent normalisée de manière à ce que l'intégrale sur tous les niveaux de potentiel soit égale à 1, mais elle peut dépasser 1 localement en tant que mesure relative de densité. En d'autre terme, une valeur de densité de probabilité supérieure à 1,0 ne signifie pas que la probabilité d'un événement individuel est supérieure à 1,0.

Nous avons remarqué sur la figure IV.2 que la densité de bruit en dessous de la valeur absolue 0,5 µV est élevée (plus de 50 % de la population totale), bien que les niveaux inférieurs à 1 µV représentent un pourcentage de plus de 93 %.

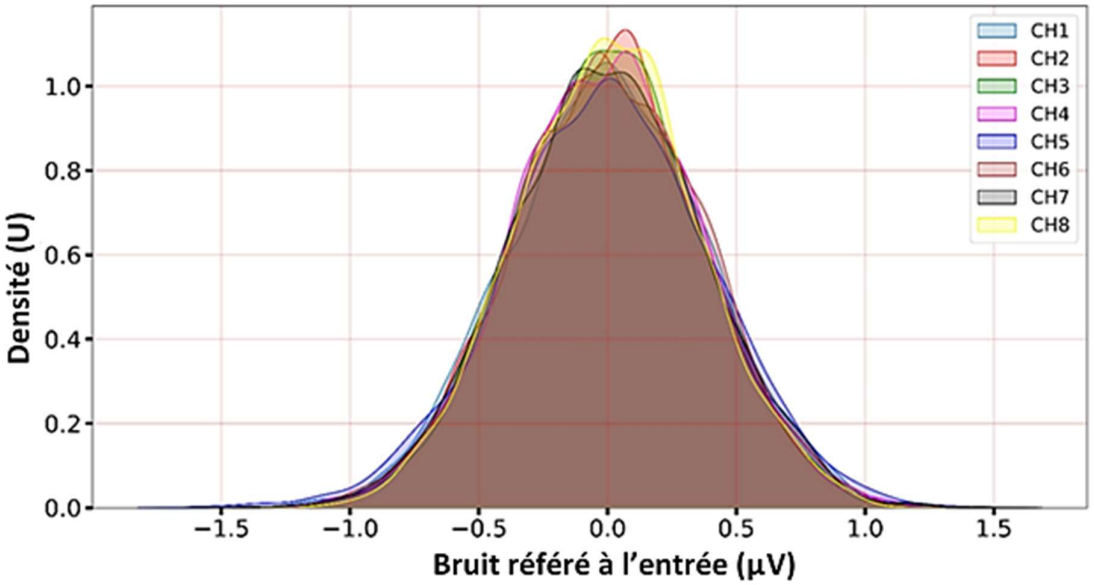


Figure IV.2. Estimation de la densité du bruit référé à l'entrée par KDE.

Le tableau IV.2 résume une analyse statistique qui définit le nombre moyen d'échantillons sur 8 canaux, distribués selon des plages potentielles spécifiques.

Plage de potentiel (µV)	**v < -1.0**	**[-1.0, -0.7]**	**[-0.7, -0.4]**	**[-0.4, -0.1]**	**[-0.1, 0.1]**	**[0.1, 0.4]**	**[0.4, 0.7]**	**[0.4, 0.7]**	**v>1.0**
Nombre moyen d'échantillons sur 8 canaux (#)	11	74	330	788	594	784	333	77	9
Pourcentage (%)	0,367	2,467	11	26,27	19,8	26,133	11,1	2,567	0,3

Tableau IV.2. Analyse statistique du bruit référé à l'entrée

On remarque que plus de 72% des échantillons sont dans la gamme [-0,4 - 0,4 µV].

Étant donné qu'il existe une distribution gaussienne du bruit, alors la relation entre les valeurs crête-à-crête V_{pp} et efficaces V_{rms} est décrite par l'équation (IV.1) [Baker B., 2005] :

$$V_{pp} = 2C_f V_{rms} \quad \text{(IV.1)}$$

C_f : Facteur d'écrêtage, souvent égal à 3 ou 3,3.

V_{rms} : est la valeur efficace du signal de bruit.

La tableau IV.3 montre la variation de la valeur V_{pp} en fonction du gain sachant la valeur V_{rms} :

Gain / Critères	1	2	4	6	8	12	24
V_{rms}(µV)	1,06	0,56	0,32	0,24	0,20	0,16	0,14
V_{pp}(µV)	7,04	3,75	2,13	1,61	1,31	1,10	0,93

Tableau IV.3. Variation de la valeur crête-à-crête du bruit référé à l'entrée en fonction du gain.

Les résultats obtenus ont montré que plus le gain est élevé, plus la valeur crête à crête du bruit référé à l'entrée est faible, ce qui est utile dans le cas de la mesure de signaux EEG de très faible amplitude. La figure IV.3 montre une très faible amplitude du bruit crête à crête de 0,93 µV pour un gain de 24.

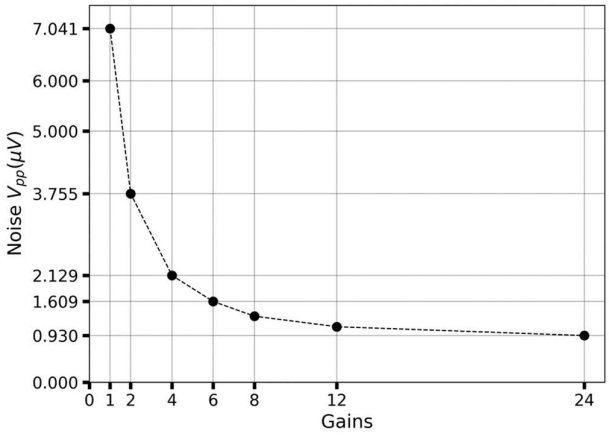


Figure IV.3. Comportement du bruit référé à l'entrée en fonction du gain.

La figure IV.4 montre un signal de bruit enregistré en 12 secondes ; on peut voir ici que la valeur absolue de l'amplitude maximale du bruit ne dépasse pas 0,5 µV.

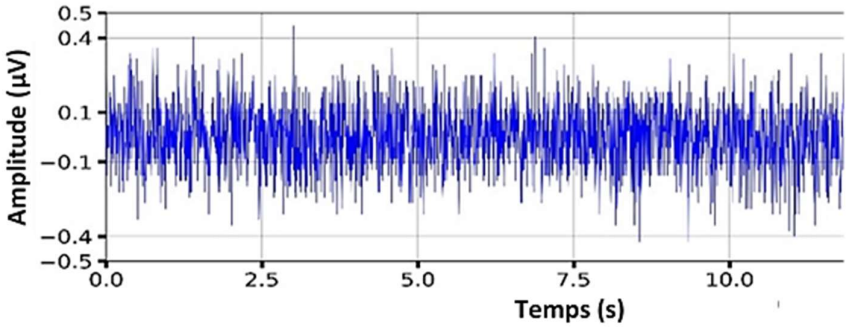


Figure IV.4. Données de séries temporelles du bruit référé à l'entrée capturé à partir du canal 8.

Dans la figure IV.5 et avec le même modèle de bruit pris (3000 échantillons), on a calculé les occurrences des différents niveaux en µV, il a été constaté que presque 50% des échantillons avaient un niveau inférieur à 0,1 µV, donc il peut être conclu que le ADS1299-AFE est caractérisé par des amplificateurs à très faible bruit et une excellente performance contre le bruit référé à l'entrée.

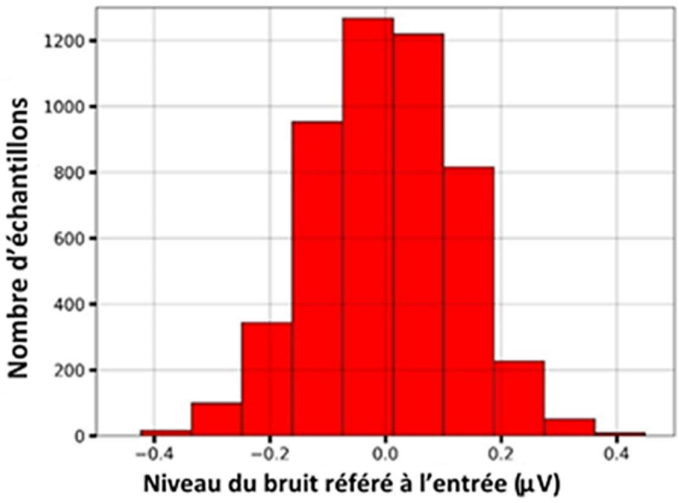


Figure IV.5. Occurrence de différents niveaux de bruit.

Au vu des résultats obtenus, nous concluons que le système proposé a une grande immunité contre le bruit référé à l'entrée.

D'autres critères techniques caractérisant le système sont importants dans l'évaluations du système, ceux sont :

- Le nombre effectif de bits ENOB (*Effective Number Of Bits*) du convertisseur analogique-numérique intégré qui est exprimé par l'équation (IV.2) [Instruments T., 2017], le CAN de l'ADS1299 est de 24 bits mais en pratique le bruit dans le signal introduit des imperfections et des distorsions supplémentaires, c'est pour cette raison que des valeurs ENOB inférieures à 24 bits ont été trouvées.

$$ENOB = log_2\left(\frac{V_{REF}}{\sqrt{2}A_D V_{rms}}\right) \quad \text{(IV.2)}$$

- La gamme dynamique, qui décrit le rapport entre l'amplitude maximale d'un signal non déformé et le bruit référencé en entrée, est exprimée par l'équation (IV.3) [Instruments T., 2017], où A_D est le gain du PGA.

$$Dynamic_{Range} = 20.log_{10}\left(\frac{V_{REF}}{\sqrt{2}A_D V_{rms}}\right) \quad \text{(IV.3)}$$

Le tableau IV.4 résume les valeurs des caractéristiques ENOB (bit) et la plage dynamique (dB) en fonction des différentes valeurs de gain du frontal analogique ADS1299-AFE :

Gain / Critères sous-jacents	1	2	4	6	8	12	24
V_{rms}(µV)	1,06	0,56	0,32	0,24	0,20	0,16	0,14
ENOB (bit)	21,52	21,44	21,25	21,08	20,93	20,67	19,86
Plage dynamique (dB)	129,60	129,12	127,96	126,02	126.02	124,44	119,86

Tableau IV.4. Mesure des critères sous-jacents en fonction du gain, pour un taux de transfert de 250 éch. /s.

IV.2.2. Mesure de la température inhérente

L'ADS1299-AFE est le dispositif le plus cher du système proposé (~52€ / puce sur mouser.com, 2022], sa protection est donc une tâche essentielle. Elle est équipé d'un capteur de température intégré DTS (*Digital Temperature Sensor*), qui est utilisé pour mesurer la température inhérente de la puce [Ramamoorthy, P. K., & Bono, A., 2014], en générant une valeur de mesure qui indique la température actuelle de la puce afin d'assurer sa fonctionnalité. Lorsqu'un problème est survenu au niveau du dispositif (par exemple, des défauts de connexion avec les composants du découplage), le résultat sera une lecture de température hors norme, une action immédiate doit donc être prise (par exemple, une mise hors tension rapide de la puce). Pendant le test, le registre CHnSET [2:0] a été ajusté à 100, le gain est réglé à 12 (à noter que le gain 24 n'est pas possible car il conduit à la saturation du PGA), cependant l'affichage de la température de la puce à partir des données acquises nécessite la conversion décrite par l'équation (IV.4) [Instruments T., 2017]. On peut voir sur la figure IV.6 que la température moyenne enregistrée par le canal 2 pendant une période de 16 secondes est d'environ 25,03 °C.

$$T(°C) = \left(\frac{Voltage[\mu V] - 145300[\mu V]}{490[\mu V/°C]}\right) + 25°C \qquad \text{(IV.4)}$$

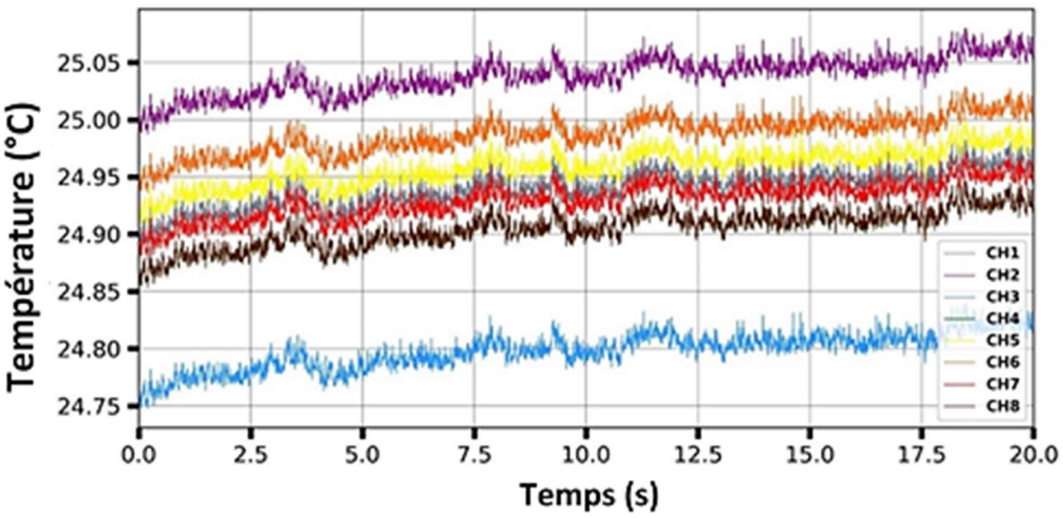


Figure IV.6. Mesure de la température du frontal ADS1299-AFE sur 8 canaux.

IV.2.3. Mesure du décalage de la composante continue (Offset-CC)

Le décalage en courant continu est un aspect inhérent aux amplificateurs PGA. Toutefois, le fait de ne pas prendre en compte cette erreur de décalage conduit finalement à une lecture erronée des données EEG [Zumbahlen H., 2011], la compensation de cette composante est donc nécessaire pour obtenir des résultats corrects. La méthode par laquelle le décalage de la composante continue a été mesurée consistait à générer un signal de test interne et à court-circuiter en interne les entrées analogiques, ensuite les valeurs de décalage ont été moyennées sur tous les canaux pour chaque valeur du gain. La figure IV.7 montre une valeur absolue de l'offset-CC égale à 643,42 µV pour un gain de 1, tandis que sa faible valeur égale à 24,8 µV pour un gain de 24.

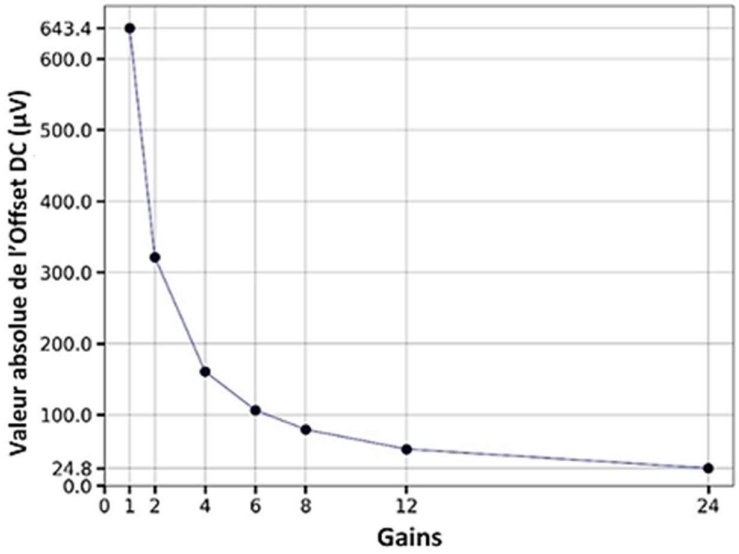


Figure IV.7. Comportement du DC-Offset en fonction du réglage du gain.

IV.2.4. Étalonnage à l'aide du signal de test interne

Tous les instruments ont tendance à dériver dans le temps et à perdre périodiquement leur précision. L'étalonnage du système d'acquisition EEG est un processus important avant sa mise en service pour garantir que sa précision reste au niveau requis. Deux modes sont fournis avec l'ADS1299-AFE ; un étalonnage avec un signal CC (fréquence 0 Hz) et l'autre avec un signal carré généré en interne [Instruments T., 2017].

Le second mode consiste à générer des ondes carrées avec une amplitude de (V_{REFP} - V_{REFN}) /2400 ou -2(V_{REFP}-V_{REFN}) /2400 et une fréquence égale à f_{CLK}/220 Hz ou égale à f_{CLK}/221 Hz, ces configurations sont établies dans le registre CONFIG2, tandis que le registre CHnSET[2:0] est réglé sur 101. Dans le scénario de test, un signal d'étalonnage carré d'une amplitude de 1887,5 µV et d'une fréquence de 1.953 Hz a été généré en interne, puis les caractéristiques du signal à la sortie ont été contrôlées. La figure IV.8 représente les signaux récupérés aux sorties des huit PGA pour un gain de 24, on remarque que leur amplitude moyenne est de 1874 µV ce qui représente une erreur de mesure de 0,71%, tandis que leur fréquence mesurée est de 1,963 Hz ce qui correspond à une erreur de 0,51%. De ce qui précède, on peut conclure que le système répond parfaitement aux recommandation décrites par le constructeur et présente de très faibles erreurs lors de l'injection du signal d'étalonnage.

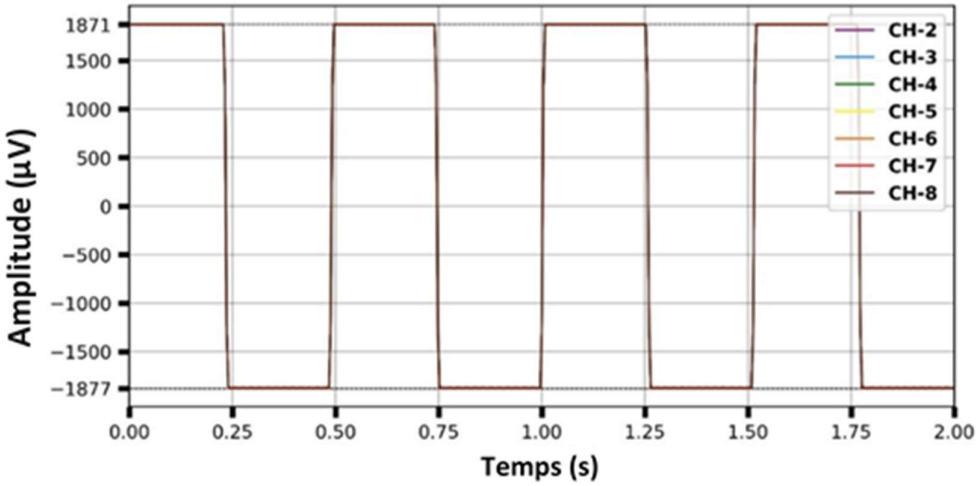


Figure IV.8. Calibrage à l'aide d'un signal de test interne.

IV.2.5. Taux de réjection en mode commun (TRMC)

Le TRMC (Dans certaines références, il est nommé CMRR, *Common Mode Rejection Ratio*) exprime la capacité d'un amplificateur d'instrumentation à rejeter un signal en mode commun ; le bruit est un signal en mode commun qui doit être rejeté par un nombre égal au TRMC du dispositif. Analytiquement, le taux de réjection en mode commun d'un amplificateur différentiel est le rapport entre le gain en mode commun A_{CM} et le gain en mode différentiel A_D [Zumbahlen H., 2011].

Etant donné que le système présenté est destiné à mesurer des signaux de très faibles amplitudes (de l'ordre de quelques µV), nous avons dû mesurer son TRMC, la méthode que nous avons adoptée est résumée dans les étapes suivantes :

1) Court-circuiter toutes les entrées inverseuses et non-inverseuses des canaux sélectionnés.

2) Injecter un signal sinusoïdal d'amplitude et de fréquence connues sur l'entrée commune.

3) Effectuer une analyse du signal enregistré à la sortie, par exemple une transformée de Fourier rapide FFT (*Fast Fourier Transform*).

4) Prendre l'amplitude de la composante de fréquence qui représente le signal d'entrée.

5) Enfin, le calcul numérique du TRMC exprimé en dB est obtenu par l'équation (IV.5) :

On a :

$$TRMC_{dB} = 20\log_{10}\left|V_{CM}\frac{A_D}{V_{OCM}}\right| \qquad \text{(IV.5)}$$

Avec :

$$TRMC_{dB} = \frac{A_D}{A_{CM}} \qquad \text{(IV.6)}$$

A_D est le gain différentiel du PGA, il est égal à 1, 4, 8, 12 ou 24.

A_{CM} : est le gain en mode commun, il est exprimé par :

$$A_{CM} = \frac{V_{OCM}}{V_{CM}} \qquad \text{(IV.7)}$$

V_{CM} est l'amplitude du signal de mode commun à l'entrée.

V_{OCM} est l'amplitude du signal de mode commun à la sortie.

Par conséquent :

$$TRMC_{dB} = 20\log_{10}\left|\frac{A_D.V_{CM}}{V_{OCM}}\right| \quad \text{(IV.8)}$$

Il faut mentionner que le scénario du test est limité seulement à la gamme de fréquences de 1 Hz à 120 Hz, qui correspond à la bande d'intérêt des signaux EEG, tandis que toutes les valeurs TRMC sont obtenues en faisant la moyenne des valeurs sur tous les canaux. En effet, les résultats obtenus sont représentés graphiquement dans la Figure IV.9, à partir de laquelle nous avons constaté que les meilleures valeurs de TRMC sont obtenues pour les gains les plus élevés, en outre pour toutes les valeurs de gain et dans la bande [1-10 Hz] nous avons un TRMC supérieur à la valeur optimale de 120 dB. Cependant, dans la bande [10-20 Hz], sa valeur est toujours supérieure à la valeur requise qui est de 110 dB, au-delà de cette plage et pour le gain de 1, il décline en dessous de la valeur typique de ~110 dB, tandis que pour les autres gains de 2 à 24, le TRMC reste supérieur à 110 dB jusqu'à la fréquence de 120 Hz.

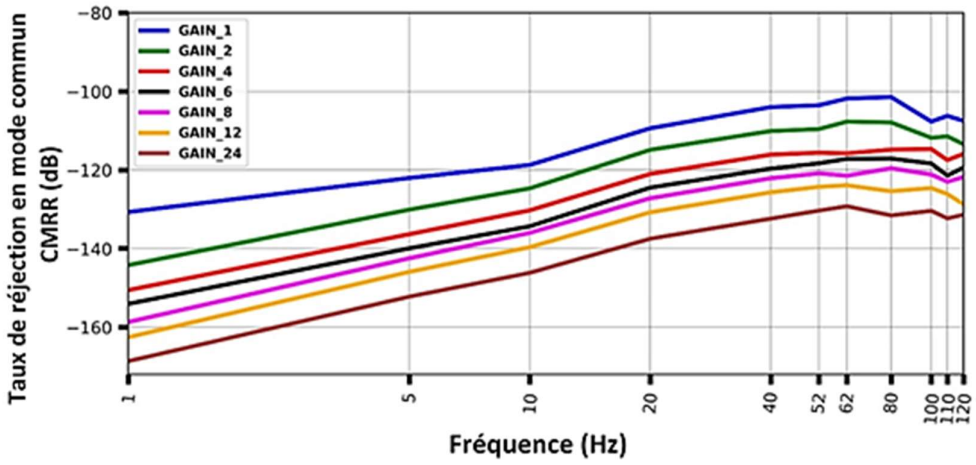


Figure IV.9. Rapport de réjection du mode commun dans la bande [1-120 Hz].

IV.2.6. Rapport de performance comparé à un modèle de référence

Nous avons considéré que l'appareil standard testé valide est celui fourni par Texas Instruments appelé EVM ADS1299EEG-FE23, qui est destiné principalement à des fins de prototypage et d'évaluation [Instruments T., 2016].

Le tableau IV.5 rapporte les performances d'évaluation du modèle par rapport à l'appareil de référence. Étant donné que les valeurs ont été mesurées au canal 1 dans les conditions suivantes : la fréquence du signal d'entrée est de 60 Hz et le gain est égal à 12.

Spécifications / Modèle	TRMC (dB)	Bruit référé à l'entrée (μV_{pp})	Offset CC (μV)
Système proposé	118,2	< 2	24,8
EVM ADS1299EEG-FE	140,69	< 1	23

Tableau IV.5. Rapport de spécifications par rapport à l'EVM ADS1299EEG-FE.

IV.3. Mesure du signal ECG

Plusieurs configurations de mesure de l'ECG sont possibles, cependant le paradigme choisi est basé sur trois électrodes et elles sont réparties comme suit :

- La première électrode est placée sur le muscle brachioradialis d'un bras et est ensuite connectée à l'entrée IN7N du canal pour capturer l'activité cardiaque.
- La deuxième est placée sur le poignet du bras opposé et est ensuite connectée à l'entrée IN7P du même canal ; cette électrode sert de référence.
- La troisième est placée sur le lobe de l'oreille gauche (A2) et connectée à l'entrée BIAS_DRV du dispositif d'acquisition ; cette référence est recommandée pour la suppression du bruit [Instruments T., 2017].

La figure IV.10 décrit le positionnement des électrodes ECG pour cette configuration.

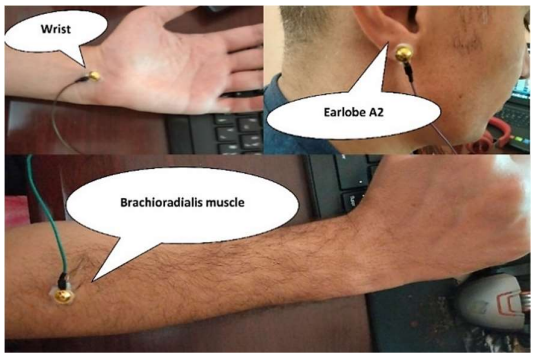


Figure IV.10. Positionnement des électrodes ECG sur le poignet, le muscle Brachioradialis, et sur le lobe de l'oreille A2.

La figure IV.11a montre des données ECG d'une durée de 10s, provenant d'un sujet adulte en bonne santé, tandis que la figure IV.11b montre le périodogramme de Welch qui permet de

surveiller les fréquences présentes dans les séries temporelles enregistrées. Des informations pertinentes peuvent être obtenues à partir de la mesure des complexes P, T et QRS des signaux ECG.

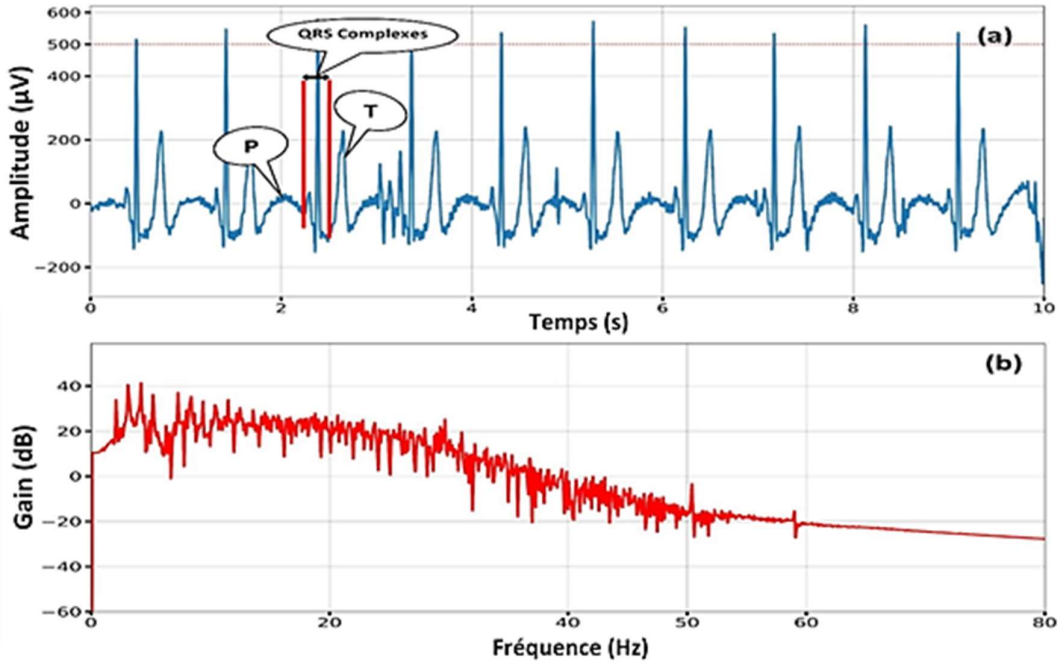


Figure IV.11. (a) Série temporelle d'ECG et (b) périodogramme de Welch correspondant en dB.

IV.4. Mesure du signal EEG

Dans le scénario réalisé, nous avons placé 6 électrodes sur les FP1, FP2, C3, O2, A1 et A2 (cependant, A1 et A2 sont les lobes auriculaires) selon le système international 10-20 comme le montre la figure IV.12b. Ensuite, ces électrodes sont connectées, respectivement, aux entrées IN1N, IN2N, IN6N, IN7N, SRB2 et BIAS_DRV du système d'acquisition. Nous avons réalisé ce travail expérimental indépendamment du contexte clinique, nous avons demandé au sujet de maintenir sa conscience et d'éviter tout mouvement physique pendant 40 secondes, puis nous avons enregistré la variation des niveaux de potentiel dans les régions sur lesquelles nous avons placé les électrodes (Figure IV.12a). Nous avons ensuite réalisé un autre paradigme dans lequel nous avons demandé au même sujet de rester éveillé mais détendu. Le spectrogramme (Figure IV.13b) montre l'apparition d'ondes alpha [8-13 Hz] captées au niveau du cortex occipital O2, lors de la fermeture des yeux (les zones encerclées), mais ces ondes disparaissent rapidement lorsque les yeux sont ouverts (en dehors de l'intervalle [8-13 Hz] de la Figure IV.13c). Les signaux enregistrés ont ensuite été filtrés par un filtre passe-bande dans la gamme [0,4 - 40 Hz]. Nous sommes satisfaits de la qualité des signaux filtrés bien qu'il existe une autre méthode qui peut donner de meilleurs résultats selon Baranowski et al [Jamal M. Z., 2012].

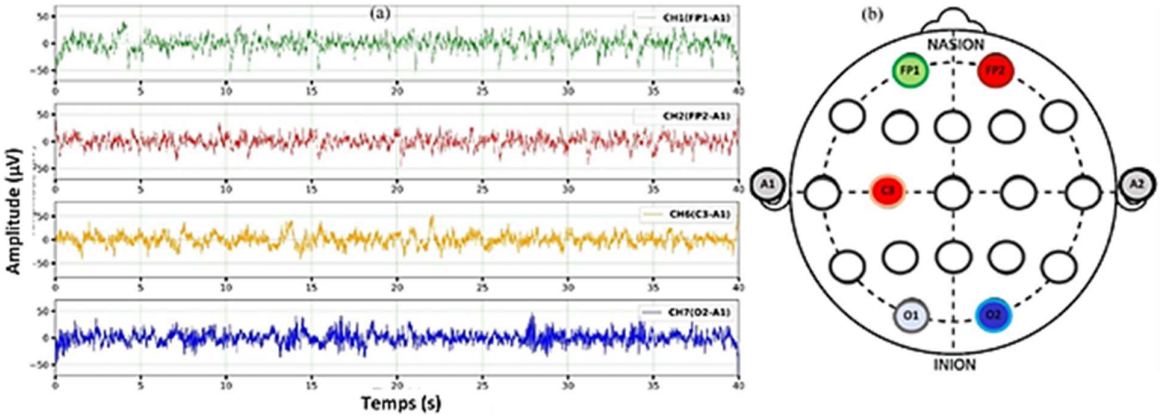


Figure IV.12. (a) EEG brut prélevé respectivement sur FP1, FP2, C3, et O2, (b) Emplacement des électrodes EEG.

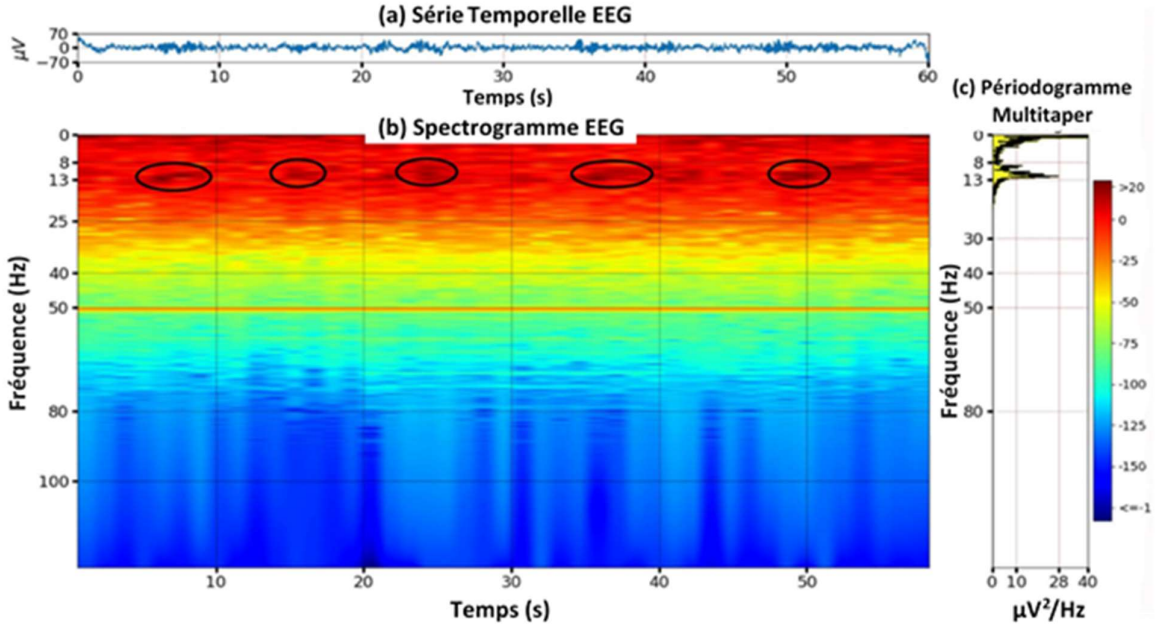


Figure IV.13. (a) Dérivation O2-A1 de l'EEG enregistré par CH7, (b) Spectrogramme O2-A1 montrant clairement les rythmes alpha (zones encerclées), (c) Densité spectrale O2-A1 avec périodogramme multitaper.

IV.5. Mesure du signal EMG

Dans cette configuration, trois électrodes ont été introduites pour capter l'activité électrique du muscle biceps brachii, qui est assez isolé, ce qui fait que nous avons peu de chance de capter des signaux indésirables provenant d'autres muscles, et donc moins de diaphonie.

La première électrode a été placée plus près de l'unité motrice du muscle (ou de son centre d'innervation), puis connectée à l'entrée IN8N du canal 8. Ensuite, la seconde électrode est placée un peu plus près de l'insertion tendineuse et toujours connectée à l'entrée IN8P du même canal.

Là encore, il faut considérer que l'axe longitudinal des deux électrodes doit être distant d'environ ~2cm et parallèle à la longueur des fibres musculaires [Stålberg E., 1980].

La troisième est la référence (utilisée pour connecter le circuit fermé à la terre), qui doit être placée sur une partie du corps très peu active (la partie la plus osseuse du corps est préférable) et ensuite connectée à l'entrée BIAS_DRV du système d'acquisition.

Le but de cette manipulation expérimentale est de surveiller l'activité du muscle brachial d'un sujet en lui demandant de le contracter deux fois pendant 3 secondes avec une pause d'environ une seconde entre les deux tests. Dans le même temps, les données sont collectées à un taux d'échantillonnage de 500 SPS pour une analyse ultérieure.

La figure IV.14 montre une série de prétraitements effectués sur le signal EMG brut, en commençant par la suppression de la ligne de base qui permet d'obtenir une valeur moyenne presque nulle (figure IV.14a).

Ensuite, une rectification du signal est effectuée afin de quantifier les phases positives et négatives (figure IV.14b), enfin, un filtrage passe-bas [0 - 10 Hz] est appliqué pour calculer l'enveloppe EMG (figure IV.14c) et son spectre correspondant (figure IV.14d).

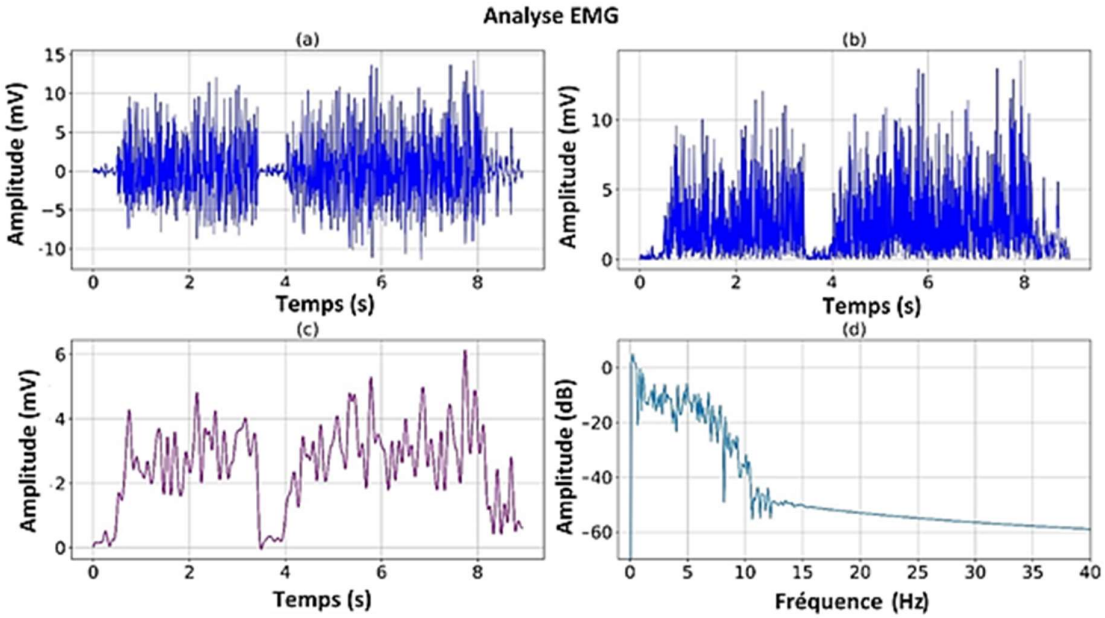


Figure IV.14. (a) EMG filtré par filtre passe-bande, (b) EMG brut rectifié, (c) Enveloppe EMG et (d) Périodogramme EMG.

Notons qu'il n'y a pas de considérations particulières concernant le choix du muscle sur lequel nous avons mené l'expérience car le but de l'étude était uniquement d'observer les

changements du signal reçu lorsque le muscle répond aux contractions et relaxations volontaires qui lui sont appliquées. Ce protocole expérimental nous a permis de savoir que nous pouvions utiliser le système proposé pour mesurer la force musculaire instantanée et ainsi suivre les états de fatigue musculaire lors d'un effort de longue durée.

IV.6. Conclusion

Les nombreux tests que nous avons effectués sur cette conception montrent que le système fonctionne correctement. La validation a été effectuée en comparant le système avec un dispositif standard, c'est l'EVM ADS1299EEG-FE23. L'introduction de l'ADS1299-AFE et de l'UCC basée sur l'ESP32 nous a permis d'obtenir un système intégré et performant avec des capacités de calcul intenses ; d'autre part, nous avons obtenu un système portable, avec un coût très bas et une consommation d'énergie minimale.

Ce système a été étalonné par un signal de test généré en interne, le signal mesuré à la sortie a montré une erreur de 0,71% en amplitude et 0,51% en fréquence. L'évaluation des performances du système a également montré sa précision et son efficacité en termes de très faible bruit référé à l'entrée ≤ 1 µV dans lequel plus de 50% des échantillons sont ≤ 0,1 µV et en termes de rapport de réjection en mode commun moyen élevé égal à ~120 dB à la fréquence 62 Hz, ces résultats ont confirmé que le système fonctionne correctement dans les conditions limites recommandées. En outre, les protocoles expérimentaux réalisés ont permis la collecte de signaux ECG et EEG dont l'analyse a été réalisée au niveau du Back-end grâce à une boîte à outils intégrale basée sur Python. Cette boîte à outils contient la plupart des scripts de prétraitement et de post-traitement nécessaires (Filtrage, recalibrage, suppression de la ligne de base, analyse en composantes indépendantes, extraction de caractéristiques, classification ...etc.). Une autre validation de l'efficacité du système a consisté à calculer la fréquence cardiaque et l'amplitude des complexes QRS de l'ECG, ainsi qu'à vérifier que la bande alpha est bien dominante dans l'EEG récupéré d'un sujet en relaxation légère, les yeux fermés.

Ce système apportera un avantage significatif aux plateformes d'applications liées aux domaines cliniques ou au domaine de l'interface cerveau-ordinateur. Compte tenu des lacunes actuelles, de nombreuses améliorations devront être apportées et peuvent être résumées dans les points suivants, qui représentent une partie des travaux futurs :

- Amélioration des méthodes de suppression du bruit.
- Test utilisant des électrodes actives.

- Conception de l'interface utilisateur graphique (GUI) supportant le streaming en temps réel, par exemple avec le langage C#.
- Évaluation de la consommation d'énergie afin d'améliorer la conception du système.
- Validation du système avec différents paradigmes tels que l'imagination motrice (IM), l'événement P300 et le schéma SSVEP.

À la lumière des résultats présentés ci-dessus, nous pouvons conclure que le travail effectué a abouti à un système temps réel polyvalent avec des fonctionnalités importantes qui peuvent être investies dans de nombreuses applications, telles que : la détection de patterns EEG, la réalisation d'interfaces cerveau-ordinateur, la rééducation cérébrale avec le neurofeedback, etc. Enfin, nous devrions proposer quelques suggestions pour rendre ce modèle efficace et précis en termes de qualité des signaux acquis, notons par exemple : l'utilisation d'électrodes actives avec des préamplificateurs intégrés (malgré leur coût plus élevé), la mise en œuvre de filtres passe-bas avec des condensateurs différentiels plutôt que des filtres avec des condensateurs mis à la terre ; ainsi cette manipulation améliorera le rapport de réjection en mode commun (TRMC) des amplificateurs d'instrumentation à l'entrée du Front-end analogique, encore, la configuration de l'amplificateur BIAS intégré est essentielle pour fournir une rétroaction négative afin d'éliminer les signaux indésirables en mode commun.

BIBLIOGRAPHIE DU CHAPITRE IV :

Anaconda Nucleus, Anaconda Individual Edition, your data science toolkit, 2020, https://www.anaconda.com/products/individual (accessed on 02 Avril 2023).

Baker, B. (2005). A Baker's Dozen: Real analog solutions for digital designers. Elsevier.

Texas-Instrument, "Noise Analysis for High-Speed Op Amps," (Revised January 2005), Application Report, available at https://www.ti.com/lit/pdf/sboa066, (accessed on 02 Avril 2023).

Texas Instruments, ADS1299 Performance Demonstration Kit, User guide (2016) https://www.ti.com/tool/ADS1299EEGFE-PDK#tech-docs.

Instruments, T. (Revised January 2017). Low-noise, 8-channel, 24-bit analog front-end for biopotential measurements ADS1299. *ADS1299 Datasheet*. (accessed on 04 Feb 2023)

Jamal, M. Z. (2012). Signal acquisition using surface EMG and circuit design considerations for robotic prosthesis. *Computational Intelligence in Electromyography Analysis-A Perspective on Current Applications and Future Challenges*, *18*, 427-448.

Mouser (2022), Texas Instruments ADS1299-x 24-Bit Analog-to-Digital Converters, https://www.mouser.fr /new/texas-instruments/ti-ads1299/ (accessed on 02 Avril 2023)

Ramamoorthy, P. K., & Bono, A. (2014, March). Measurement and characterization of die temperature sensor. In *2014 Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)* (pp. 41-44). IEEE.

Stålberg, E. (1980). Macro EMG, a new recording technique. *Journal of Neurology, Neurosurgery & Psychiatry*, *43*(6), 475-482.

Zumbahlen, H. (Ed.). (2011). *Linear circuit design handbook*. Newnes.

CHAPITRE V

DEVELOPPEMENT DE LA BOITE À OUTILS STEP2BCI

CONTENU

V.1. Introduction
V.1.1. Pourquoi Python ?
V.2. État de l'art est travaux similaires
V.3. Présentation de la boite à outils step2bci
V.4. Prétraitement du signal EEG
V.4.1. Structure des données EEG
V.4.2. Filtrage numérique
V.4.3. Sous-échantillonnage
V.4.4. Normalisation et échelonnage des données EEG
V.4.5. Correction de la ligne de base
V.4.6. Fonctions de calculs analytiques et statistiques
V.4.7. Fonctions de l'analyse spectrale et fréquentielle
V.4.7.1. Densité spectrale de puissance (DSP)
V.4.7.2. Spectrogramme via la transformée de Fourier à Court Terme (STFT)
V.4.7.3. Périodogrammes
V.4.7.4. Analyse de la puissance de bande
V.4.8. Décomposition en ondelettes et analyse multi-résolution
V.4.9. Segmentation (Epoching) des données et l'extraction des essais
V.5. Post-traitement du signal EEG
V.5.1. Extraction des caractéristiques
V.5.1.1. Extraction multi-résolution par décomposition en ondelettes
V.5.1.2. Extraction par les motifs spatiaux communs (filtrage spatiale)
V.5.1.3. Extraction de la variance logarithmique
V.5.1.4. Extraction de la puissance des sous-bandes fréquentielles
V.5.1.5. Extraction des caractéristiques par l'entropie échantillonnée (SampEn)
V.5.2. Sélection et réduction de l'espace des caractéristiques
V.5.2.1. Sélection séquentielle des caractéristiques (SFS)
V.5.2.2. Sélection par l'analyse des composantes principales (PCA)
V.5.2.3. Sélection par analyse en composantes indépendantes (ICA)
V.5.3. Classification par Apprentissage automatique
V.5.3.1. Classifieur K-plus proches voisins (KNN)
V.5.3.2. Classifieur d'analyse discriminante linéaire (LDA)
V.5.3.3. Classifieur Machine à vecteurs de support (SVM)
V.5.4. Classification par apprentissage profond
V.5.4.1. Module de gestion des dispositifs (deepDevice)
V.5.4.2. Module de personnalisation et de transformation des données EEG (neuroDataset)
V.5.4.3. Module d'apprentissage profond (deepBCI)
V.6. Conclusion
Bibliographie du Chapitre V

V.1. Introduction

L'objectif de cette partie est de fournir une boite à outils pour un système BCI, libre et open-source, écrit en un langage de programmation Python ; c'est un langage moderne, de haut niveau et qui dispose de bibliothèques standard soutenues par une communauté importante, diverse et en pleine progression [Python, 2021]. En outre, la boite à outils doit être compatible à la plupart des systèmes d'exploitation très connus par le grand public (à savoir : Windows, Linux et MacOs). "Libre et open source" signifie que le logiciel résultant doit être librement disponible pour que tout le monde puisse l'utiliser et sa source doit être libre pour que quiconque l'étudie et plutôt le modifie). En conséquence, chaque composant doit être distribué sous les termes d'une licence de logiciel libre et le code correspondant doit être mis à disposition du public.

V.1.1. Pourquoi Python ?

Python est un langage de programmation libre et open source qui est compatible à la plupart des plates-formes logicielles (Windows, Linux, MacOs, Android OS, etc.), ce qui le rend une solution idéale pour la communauté scientifique et dans la recherche universitaire, notamment en bio-informatique, biologie et mathématiques. C'est la langue d'introduction standard pour de nombreux programmes universitaires d'informatique. Sa mise en œuvre est commencé en décembre 1989 par le programmeur néerlandais Guido van Rossum [Van Rossum, G., 2021]. Depuis lors, Python a évolué à travers différentes versions ; la version 0.9, la version 1.x qui a été maintenue pendant plus de 5 ans, ensuite la version 2.x qui est sorti le 16 octobre 2000 à BeOpen [BeOpen, 2000] avec plusieurs améliorations telles que : les listes en compréhension, la détection de cycle d'un itérateur, le récupérateur de mémoire, la prise en charge d'Unicode, etc. Mais au fil du temps, il a été confirmé que le noyau Python 2 manquait de nombreuses fonctionnalités qui rend difficile l'amélioration de Python, les fondateurs ont donc dû faire des changements radicaux et donc développer une nouvelle version meilleure et plus rapide, qui est python 3.x, mais en parallèle, les mainteneurs ont continué à soutenir formellement Python 2 jusqu'en 2020 (où le projet a finalement été abandonné) afin que les développeurs puissent migrer progressivement leurs projets vers la dernière version.

Alors, la véritable révolution Python a eu lieu lors de la sortie sa version 3.0 en Décembre 2008 par PSF (*Python Software Foundation*), avec un bon nombre de fonctionnalités telles que : l'automatisation de la traduction du code Python 2 en Python 3 via l'utilitaire 2To3, le formatage de chaîne avancé, le package de multitraitement, etc.

Actuellement, Python est utilisé par des millions d'utilisateurs dans le monde [Beyene T. & Sharma A., 2021], un large éventail d'entreprises technologiques (NASA, IBM, Netflix, Google, etc.), de laboratoires de recherches et de développement, d'agences financières. En effet, il a devenu l'un des langages de programmation les plus populaires [Louden K. C. & Lambert K. A., 2011 ; Bissyandé et al., 2013] et concurrent pour plusieurs produits commerciaux comme le Matlab.

Des projets open source comme Numpy [Van Der Walt S. et al., 2011], Scipy [Virtanen, P., et al., 2020], Matplotlib [Hunter J. D., 2007] et Pandas [McKinney W., 2010], sont devenus la base du calcul scientifique en Python et d'autres projets comme Scikit-learn, TensorFlow [Abadi M. et al., 2016] et PyTorch [Paszke A. et al., 2019] pour les sciences des données, l'apprentissage approfondie et l'intelligence artificielle.

La PSF et la communauté des développeurs Python prennent les vulnérabilités de sécurité très au sérieux. Une équipe de réponse à la sécurité de Python a été formée pour trier toutes les vulnérabilités signalées et recommander des contre-mesures appropriées. En outre, Python 3.x est très stable ; de nouvelles versions stables sortent environ tous les 6 à 18 mois. Les développeurs publient des versions de correction de bogues pour les anciennes versions, afin d'améliorer progressivement la stabilité des versions existantes. Les versions de correction de bogues, indiquées par un troisième composant du numéro de version (par exemple 3.5.3, 3.6.2), sont gérées dans un souci de stabilité ; seules les corrections des problèmes connus sont incluses dans une version de correction de bogues, et il est garanti que les interfaces resteront les mêmes tout au long d'une série de versions de correction de bogues. Le processus Python Enhancement Proposal (PEP) [Van Rossum G., 2001] permet au langage Python d'évoluer au fur à mesure que les utilisateurs proposent des modifications de son comportement.

V.2. État de l'art est travaux similaires

Dans la littérature, ils existent des boites à outils pour les systèmes BCI :

BCI2000, [Schalk G. et al., 2004] développé par NCAN (*National Center for Adaptive Neurotechnologies*), est une plateforme polyvalente pour la recherche des interfaces cerveau-ordinateur. Elle intègre des outils logiciels pour l'acquisition, le traitement des données, la présentation des stimuli et ainsi que le retour d'information. L'avantage de la plateforme BCI2000 est qu'elle est gratuite mais uniquement pour un usage académique et non commercial. Cependant, ses limites sont qu'elle ne fonctionne que sur les systèmes Microsoft Windows et qu'elle est écrite en C/C++, ce qui rend difficile pour les non-programmeurs d'écrire leurs

propres scripts si les bibliothèques intégrées ne répondent pas à tous leurs besoins. Mais les développeurs se sont rendu compte de ce problème et ont conçu une fonction de développement rapide et flexible basée sur le projet logiciel BCPy200 [Schreiner S. et al., 2008] pour les développeurs et les cliniciens ayant une certaine expérience de Python ; ce projet est modulaire, avec trois modules indépendants : application, source de signal et traitement du signal. Désormais, les utilisateurs peuvent utiliser le module de source de signaux BCI2000, qui prend en charge un large éventail de modules d'acquisition EEG, ou exécuter les modules en tant que modules Python. Les performances en temps réel sont excellentes et intéressantes puisque tous les modules sont écrits dans un langage de bas niveau (C/C++) permettant de synchroniser les signaux EEG et autres avec une variété de biopotentiels et de dispositifs d'entrée tels que des souris et des eye-trackers. BCI2000 supporte officiellement de nombreux dispositifs professionnels tels que g.USBamp, g. MOBIlab, g.Unicorn, etc., et d'autres dispositifs mais après une contribution des utilisateurs tels que les systèmes Modular-EEG, les systèmes Emotiv EPOC, les systèmes Micromed EEG, etc.

OpenViBE créé en 2007 [Renard Y. et al., 2010] ; est une plateforme logicielle complète dédiée à la conception, au test et à l'utilisation d'interfaces cerveau-ordinateur. OpenViBE est écrit en C/C++ et compatible aux environnement Microsoft Windows et Linux. Elle contient une large gamme des applications, des programmes préconfigurés et des paradigmes qui sont prêts à être exploités selon une approche graphique de type glisser-déposer. OpenViBE est un logiciel libre inscrit sous la licence AGPL (*Affero General Public License*) pour les neurosciences qui peut être utilisé pour acquérir, filtrer, traiter, classer et visualiser les signaux cérébraux en temps réel. Cette plateforme assure une excellente performance en terme des recommandations temps réel et en terme d'utilisation simplifiée en fournissant des liaisons Python et Matlab.

BioSig [Schlögl A. and Brunner C., 2008] est l'une des boites à outils à code source ouvert les plus anciennes disponible sous la licence publique générale GPL et qui est destinée au traitement des signaux biomédicaux, comprenant par exemple l'analyse des biopotentiels tels que : l'EEG, l'ECoG, l'EMG, etc. Le projet BioSig a été conçu pour la première fois afin de favoriser les recherches dans les domaines des interfaces cerveau-ordinateur, la psychologie, la neuro-informatique, etc. en fournissant des solutions en C/C++, Python, Matlab/Octave pour l'acquisition des données, le prétraitement et la visualisation des signaux, l'extraction des caractéristiques, la classification, etc. La plateforme BioSig peut prendre en charge des dizaines de formats de données différents (EDF, GDF, BDF, …) et intègre un système BCI en temps réel (rtsBCI) implémenté dans Matlab et Simulink.

FieldTrip est une boite à outils Matlab à code source ouvert développée par l'institut DIB (Donders Institute of Brain) à l'université de Radboud (Pays-Bas) pour l'analyse des EEG et MEG. Elle est inscrite sous les termes de la licence GPL et elle contient différents algorithmes pour l'analyse temps-fréquence, la reconstruction de sources, les tests statistiques non-paramétriques, etc. FieldTrip peut supporter les formats des données des principaux systèmes EEG et MEG [Oostenveld R. et al., 2011].

BCILAB [Kothe C. and Makeig S., 2013] est une boîte à outils MATLAB pour la recherche sur les interfaces cerveau-ordinateur. Il s'agit d'une boite à outils gratuite à code source ouvert Elle a été créé par C. Kothe au centre de neurosciences computationnelles de Swartz (USA) et financé par le laboratoire de recherche de l'armée américaine. La conception de BCILAB a été inspirée par une boîte à outils obsolète appelée PhyPA écrite en 2006 par C. Kothe avec des contributions de T. Zander au Département des systèmes homme-machine à l'institut de technologie de Berlin (*Allemagne*). Cette boîte à outils n'est plus activement développée depuis 2017. Cependant, des mises à jour critiques sont toujours effectuées. BCILAB prend en charge l'exécution en ligne et hors ligne des signaux EEG, EMG, etc. et elle incorpore de divers algorithmes de traitement des signaux et d'extraction des caractéristiques et ainsi de trois classifieurs à savoir les classifieurs SVM (*Support Vector Machine*), LDA (*Linear Discriminant Analysis*) et QDA (Quadratic Discriminant Analysis).

MNE-Python [Gramfort A. et al., 2013] est un gigantesque projet à code source ouvert certifié OSI (Open Source Initiative) sous la licence BSD (*BSD-3-Clause*) ; autrement dit la redistribution et l'utilisation sous forme source et binaire, avec ou sans modification, sont autorisées à condition que certaines conditions soient respectées (lire la licence : *https://github.com/mne-tools/mne-python*). Il est fourni sous forme d'un package Python pour explorer, visualiser et analyser les données neurophysiologiques humaines : EEG, MEG, EEG, ECoG, etc. et comprend des modules pour l'E/S des données, le prétraitement, la visualisation, l'estimation des sources, l'analyse temps-fréquence, l'analyse de la connectivité, l'apprentissage automatique et les statistiques. Ce package est disponible dans le dépôt de paquets PyPI (le dépôt tiers officiel de Python). Cependant, au moment de la rédaction de cette thèse, le MNE ne peut effectuer que l'analyse hors ligne des données.

Pyff [Venthur B. et al., 2010), Mushu [Venthur B. et al., 2012] et Wyrm [Venthur B. et al., 2014] sont trois composantes indépendantes d'une même bibliothèque d'un système BCI modulaire et polyvalent conçu par Venthur B. et al à l'université technique de

Berlin (*Allemagne*). Cette plateforme est complètement gratuite, à code source ouvert et fonctionnant sur tous les systèmes d'exploitation et elle est écrite en langage moderne (Python) adapté aux calculs scientifiques. Le mot "indépendant" vient du fait que ce système n'est pas monolithique ; Autrement dit, l'utilisateur peut utiliser une ou toutes ces composantes dans une application quelconque. En d'autre terme, on peut utiliser uniquement l'application Mushu pour l'acquisition des signaux, Wyrm pour l'exécution d'expériences BCI en ligne et pour le traitement hors ligne des données EEG et/ou Pyff pour la présentation du feedback et du stimulus.

Gumpy est la plus récente boite à outils adaptée aux interfaces cerveaux ordinateur hybrides développé par T. Zied à l'université technique de Munich (*Allemagne*) [Zied T. et al., 2018], elle intègre plusieurs fonctions pour l'analyse, l'acquisition, la visualisation et la classification en temps réel des signaux EEG et EMG. Cette plateforme a été mise en œuvre et approuvée par la même équipe de développement avec plusieurs paradigmes tels que : la prédiction des mouvements par l'imagination motrice, la commande d'un robot à l'aide des potentiels évoqués visuels à l'état d'équilibre (SSVEP) et ainsi que la commande temps réel d'une prothèse à l'aide des signaux EMG de surface (sEMG).

V.3. Présentation de la boite à outils step2bci

Notre logiciel ***Step2bci*** est une boîte à outils complète conçue pour simplifier l'analyse et le traitement des données EEG. Cette boîte à outils conviviale est composée de modules et de classes variés qui offrent une gamme étendue de fonctionnalités. Grâce à *Step2bci*, les utilisateurs ont accès à un ensemble d'outils diversifiés pour pré-traiter et post-traiter les données EEG.

La boîte à outils propose de nombreuses fonctionnalités, telles que l'édition des données, le filtrage, le traitement des époques (épisodes), l'analyse par ondelettes, l'analyse spectrale de puissance, l'extraction des essais, les techniques d'extraction de caractéristiques, ainsi que la classification à l'aide de méthodes d'apprentissage automatique ou d'apprentissage profond, etc. Chaque package de la bibliothèque *Step2bci* est également équipé de modules nombreux, offrant aux utilisateurs un ensemble polyvalent d'outils pour gérer différents aspects de l'analyse des données EEG.

Avec ses fonctionnalités étendues, la boîte à outils *Step2bci* permet aux utilisateurs de prétraiter et d'analyser efficacement les données EEG, d'extraire des caractéristiques pertinentes et d'appliquer des techniques de classification avancées. Que ce soit pour explorer les

dynamiques temporelles, les composantes fréquentielles ou pour identifier des motifs et des tendances dans les signaux EEG, *Step2bci* offre une suite complète d'outils permettant de simplifier le processus d'analyse et d'obtenir des informations précieuses à partir des données EEG.

V.4. Prétraitement du signal EEG

Dans cette partie, nous nous concentrerons sur le package "*preproc*", un ensemble complet de modules conçus spécifiquement pour le prétraitement des signaux EEG dans le contexte des interfaces cerveau-ordinateur (BCI). Ce package est un élément crucial, car il fournit tous les outils et techniques nécessaires pour prétraiter efficacement les données EEG, en garantissant une qualité optimale et une adéquation aux applications BCI. En explorant les capacités et les caractéristiques du package "*preproc*", nous pouvons nous faire une idée des étapes essentielles de la préparation des signaux EEG en vue d'une analyse et d'une interprétation plus poussées dans le domaine des BCIs.

Le processus de prétraitement consiste à préparer l'activité cérébrale brute via une série des fonctions de nettoyage, de filtrage et de la mise en échelle afin de mettre en valeur l'information pertinente contenue dans l'activité EEG mesurée [Elsayed N. et al., 2017].

V.4.1. Structure des données EEG

Step2bci offre un module appelé '*IOFiles.py*' qui intègre différentes fonctions pour la lecture et l'écriture de fichiers dans divers formats tels que Matlab **(.mat**), texte **(.txt**) et valeurs séparées par une virgule (**.csv**). Ces méthodes permettent de traiter des ensembles de données provenant de différents normes et standards tels que Graz-lab, Berlin Institute et Open BCI, en plus de prendre en charge les données CSV issues de notre propre matériel. Chaque fonction de ce sous-module peut renvoyer un tableau de données, de cibles et d'essais (*trials*). Les données sont ensuite remodelées pour adopter une structure spécifique en termes d'échantillons et de nombre de canaux, tout en prenant en compte d'autres métadonnées telles que la fréquence d'échantillonnage, les étiquettes, les marqueurs, etc. Cette intégration de formats de fichiers variés facilite la manipulation et le traitement de données provenant de différentes sources, tout en préservant les informations essentielles requises pour les analyses ultérieures dans les projets BCI.

V.4.2. Filtrage numérique

- **Filtre de Butterworth**

Dans la boîte à outils *Step2bci*, nous avons implémenté un module de filtrage '*dataFilter.py*' qui offre des méthodes pour créer des filtres de Butterworth [Selesnick I. W., & Burrus C. S., 1998] et de Kaiser [Datar, A. et al, 2010]. Ces méthodes incluent des filtres passe-bas, passe-haut, passe-bande et stop-bande.

Le choix de ce type des filtres est fait pour avoir une réponse en fréquence aussi plate que possible (c'est-à-dire sans ondulations) dans la bande d'intérêt (passante), mais au prix d'une bande de transition relativement large. Dans les systèmes BCI basés sur l'activité EEG, l'utilisation de simples filtres du premier ordre peut entraîner une bande de transition (*roll-off*) trop large. C'est pourquoi il est nécessaire d'utiliser des filtres d'ordre supérieur ou d'ordre "*n*" afin d'obtenir une conception plus précise.

La classe implémenté est basée principalement sur la bibliothèque ***Scipy*** de calcul scientifique. La figure V.1 représente une réponse fréquentielle d'un filtre Butterworth passe-bas d'ordre 1 à 6 de fréquence de coupure 250 Hz, ce filtre a été appliqué à un signal échantillonné à la fréquence f_e = 1000 Hz. Pour un filtre de Butterworth, c'est le point auquel le gain diminue à $\frac{1}{\sqrt{2}}$ de celui de la bande passante (le point -3 dB). Nous constatons également que la largeur de la bande de transition dépend de l'ordre du filtre et que pour un simple filtre de premier ordre. Il est important de noter qu'il est possible de construire un filtre Butterworth passe-bande en combinant un filtre passe-bas et un filtre passe-haut. Cette méthode est couramment appelée "*en cascade*" ou "*croisée*".

Les gains des filtres de Butterworth pour les configurations passe-bas et passe-haut sont donnés par les relations suivantes :

1. Filtre passe-bas de Butterworth :

$$G_n(f) = \frac{1}{\sqrt{1+\left(\frac{f}{f_c}\right)^{2n}}} \tag{V.1}$$

2. Filtre passe-haut de Butterworth :

$$G_n(f) = \frac{\left(\frac{f}{f_c}\right)^{2n}}{\sqrt{1+\left(\frac{f}{f_c}\right)^{2n}}} \tag{V.2}$$

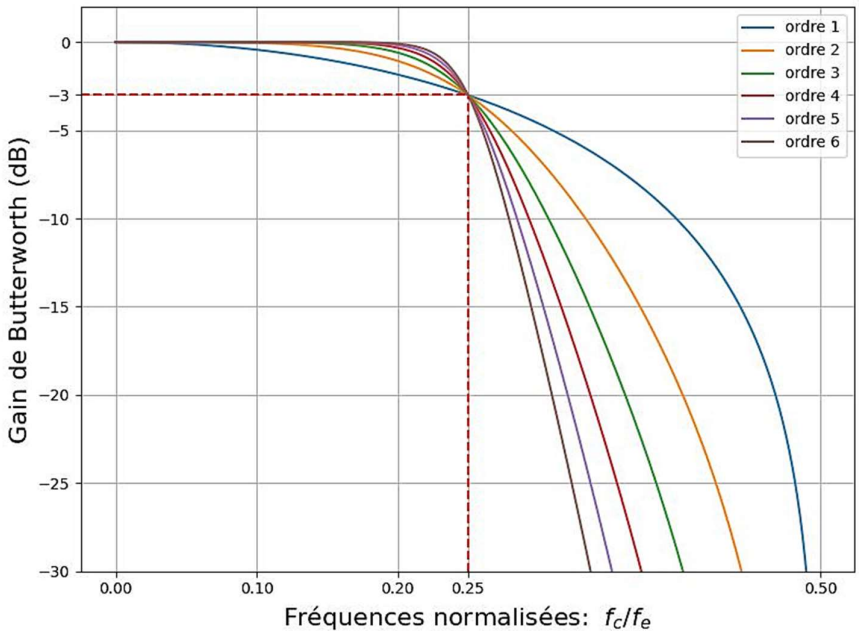


Figure V.1. Réponse fréquentielle d'un filtre Butterworth passe-bas d'ordre 1, 2, ...6.

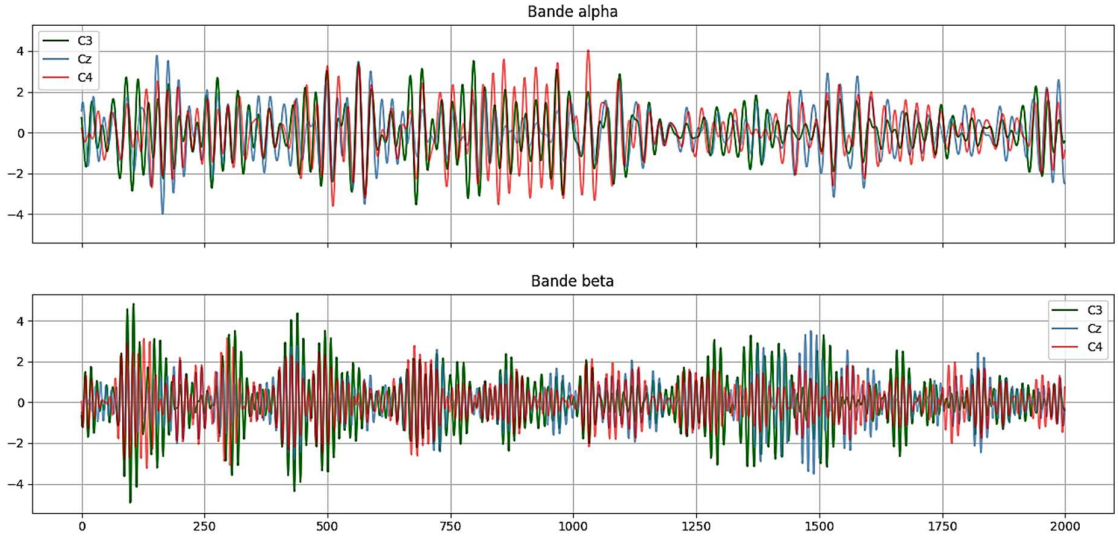


Figure V.2. Extraction de 2000 échantillons des rythmes Alpha [8- 12 Hz] et Beta [13- 28 Hz] au niveau des électrodes C3-Cz-C4 en utilisant un filtre Butterworth passe-bande de 4ème ordre.

V.4.3. Sous-échantillonnage

Afin de capturer les dynamiques rapides de l'activité cérébrale, les données EEG sont généralement enregistrées à des taux d'échantillonnage élevés. Cependant, les données à haute fréquence peuvent être difficiles à stocker et à analyser, en particulier dans les études à grande

échelle. Pour résoudre ces problèmes, le sous-échantillonnage est couramment utilisé. Cependant, lors de la réduction du taux d'échantillonnage des données EEG, il est important de prendre en compte plusieurs facteurs, notamment le contenu fréquentiel du signal, le type d'analyse et le compromis entre l'efficacité et la précision du calcul souhaité. Il est important de noter qu'une simple réduction du taux d'échantillonnage des données EEG peut entraîner une perte d'informations, à moins que l'on ne sache que les données d'intérêt se trouvent dans une bande de fréquences particulière vers laquelle on peut filtrer les données. Pour déterminer si cela est possible, il est nécessaire d'examiner les données dans le domaine fréquentiel afin de calculer la densité spectrale dans une bande de fréquence donnée.

Des études récentes ont suggéré diverses approches optimales pour le sous-échantillonnage des données EEG, notamment Donoghue T. et al. (2020), qui ont introduit une méthode qui préserve le contenu spectral du signal tout en minimisant la perte d'information au cours du processus de sous-échantillonnage et Croce P. et al (2018), qui à leur tour ont proposé une méthode qui ajuste dynamiquement le taux d'échantillonnage en fonction des caractéristiques locales du signal afin de préserver les informations pertinentes tout en réduisant les exigences de calcul.

En procédant à un sous-échantillonnage approprié des données EEG, les chercheurs peuvent accroître l'efficacité et la précision du traitement des données EEG, ce qui permet d'obtenir des informations précieuses sur les processus cognitifs et les fonctions cérébrales.

Nous avons intégré notre propre fonction '*subsampler ()*'de sous-échantillonnage dans le module '*dataEditor.py*' Cette fonction permet un sous-échantillonnage efficace des signaux EEG tout en préservant les informations essentielles. En réduisant le taux d'échantillonnage, elle permet de réduire les exigences de calcul et le temps de traitement sans perte significative de données pertinentes.

La figure V.3B présente l'application de notre fonction de sous-échantillonnage sur un signal EEG mesuré au niveau de l'emplacement cérébral C3, échantillonné à une fréquence de 1000 Hz (figure V.3A). Le résultat obtenu démontre la transformation du signal EEG vers une nouvelle fréquence d'échantillonnage de 100 Hz. Malgré cette réduction de fréquence, le signal résultant préserve les parties informatives essentielles.

V.4.4. Normalisation et échelonnage des données EEG

L'analyse de données EEG non normalisées peut soulever de nombreux problèmes. Tout d'abord, les signaux EEG présentent généralement des amplitudes différentes d'une électrode à l'autre et d'un sujet à l'autre, ce qui complique la comparaison des données et l'interprétation précise [Hu Y., 2018]. La normalisation permet de surmonter ces différences d'amplitude, en veillant à ce que l'analyse ne soit pas biaisée en faveur des électrodes présentant les amplitudes les plus élevées [Hu Y., 2018]. En outre, les données EEG non normalisées peuvent affecter l'analyse spectrale et les mesures de cohérence, car les spectres de puissance et les valeurs de cohérence peuvent être influencés par les variations d'amplitude du signal plutôt que par l'activité cérébrale sous-jacente réelle [Pouryazdian S. et al., 2018].

Plusieurs études ont souligné l'importance de la normalisation des données EEG. Par exemple, l'étude de Schomer D. L. et al., (2012) a montré comment la normalisation améliore la fiabilité et la cohérence de l'analyse de la connectivité cérébrale basée sur l'EEG en réduisant la variabilité entre les sujets. Une autre étude [Sanei S., & Chambers J. A., 2013] a démontré que la normalisation des données EEG améliore la précision du modèle de classification pour les interfaces cerveau-ordinateur, ce qui conduit à des applications plus robustes et plus efficaces.

En résumé, la normalisation des données EEG est essentielle pour éliminer les variations d'amplitude et garantir une analyse précise et significative de l'activité cérébrale [Hu Y., 2018]. Elle améliore la comparabilité entre les électrodes et les sujets, ce qui permet d'obtenir des interprétations plus fiables et d'améliorer les performances des différentes techniques d'analyse de l'EEG [Schomer D. L. et al., 2012 ; Sanei S., & Chambers J. A., 2013].

Step2bci intègre un module '*dataScaler.py*' qui comprend une série de techniques de normalisation pour l'analyse des données EEG. Ces techniques comprennent, entre autres, la normalisation robuste, la mise à l'échelle min-max, la normalisation max-abs, etc. Chaque technique fournit une approche distincte de la normalisation des données EEG, permettant la comparabilité et l'élimination des biais.

La figure V.3C montre le signal EEG normalisé, où les amplitudes ont été normalisées ce qui permet un examen clair de la dynamique du signal, le 2ème graphique (la ligne courbée en marron) illustre la dérive du signal, qui représente les changements progressifs et continus de la ligne de base du signal au fil du temps. Il est essentiel de comprendre et d'atténuer la dérive de la ligne de base pour une interprétation et une analyse précises des données EEG.

V.4.5. Correction de la ligne de base

Une ligne de base est généralement un artefact indésirable qui altère l'interprétation ou le calcul des signaux [Chen Y., 2019]. Le concept n'est pas totalement bien défini, et plusieurs noms différents coexistent, avec des significations similaires : dérive, continuum, tendance, fond, etc. Il est souvent décrit comme un niveau de référence à grande échelle, variant lentement, qui doit être supprimé pour comparer ou ajuster le signal par rapport à une référence globalement plate ou nulle. Il peut être considéré comme un décalage généralisé (non constant) ou une correction de biais. Lorsqu'il est supprimé, il est généralement plus simple de poursuivre le traitement du signal, de quantifier les paramètres et d'extraire les caractéristiques.

L'EEG est un signal avec une excellente résolution temporelle, mais on constate souvent des dérives temporelles causées par des sources internes ou externes du système d'acquisition. Généralement, il est important d'effectuer une correction dite de la ligne de base [Luck, S. J. & Kappenman, E. S., 2012].

La présence d'une ligne de base est un type d'artefact qui peut altérer l'interprétation et l'analyse des signaux EEG. La correction de la ligne de base vise à compenser cet effet en utilisant l'activité EEG enregistrée pendant une période de référence avant l'apparition d'un événement externe, afin de corriger l'activité enregistrée dans l'intervalle suivant cet événement qui est appelé intervalle post-stimulus [Bekkali, S., 2021]. Il existe différentes approches pour effectuer cette correction. La méthode traditionnelle consiste à soustraire la valeur moyenne de l'activité EEG enregistrée pendant la période de référence de chaque point temporel à la fois dans l'intervalle de référence et l'intervalle post-stimulus. En d'autres termes, on calcule la moyenne des valeurs de tension pour chaque électrode pendant une période de temps donnée, puis cette moyenne est soustraite des valeurs correspondantes de l'intervalle post-stimulus du signal enregistré.

Dans le module '*dataEditor.py*' de la boîte à outils *Step2bci*, nous avons intégré une méthode spécifique pour l'analyse des signaux EEG, visant à éliminer les variations de la ligne de base. Cette étape de prétraitement est d'une importance cruciale, car les variations de la ligne de base peuvent introduire du bruit et fausser les résultats de l'analyse. En éliminant ces variations, nous renforçons la qualité des signaux EEG en réduisant les artefacts, ce qui facilite une interprétation plus précise des activités cérébrales. Cette amélioration permettra une exploration plus approfondie des caractéristiques et des motifs des signaux EEG.

La figure V.3D illustre le même signal EEG-C3 normalisé et corrigé de sa ligne de base, avec comme caractéristique principale une réduction significative de la dérive. En corrigeant efficacement cette dérive, le comportement temporel du signal est rectifié tout en préservant son motif global. Cette représentation visuelle met en évidence le succès de l'atténuation de la dérive, garantissant une analyse des données EEG plus précise et fiable.

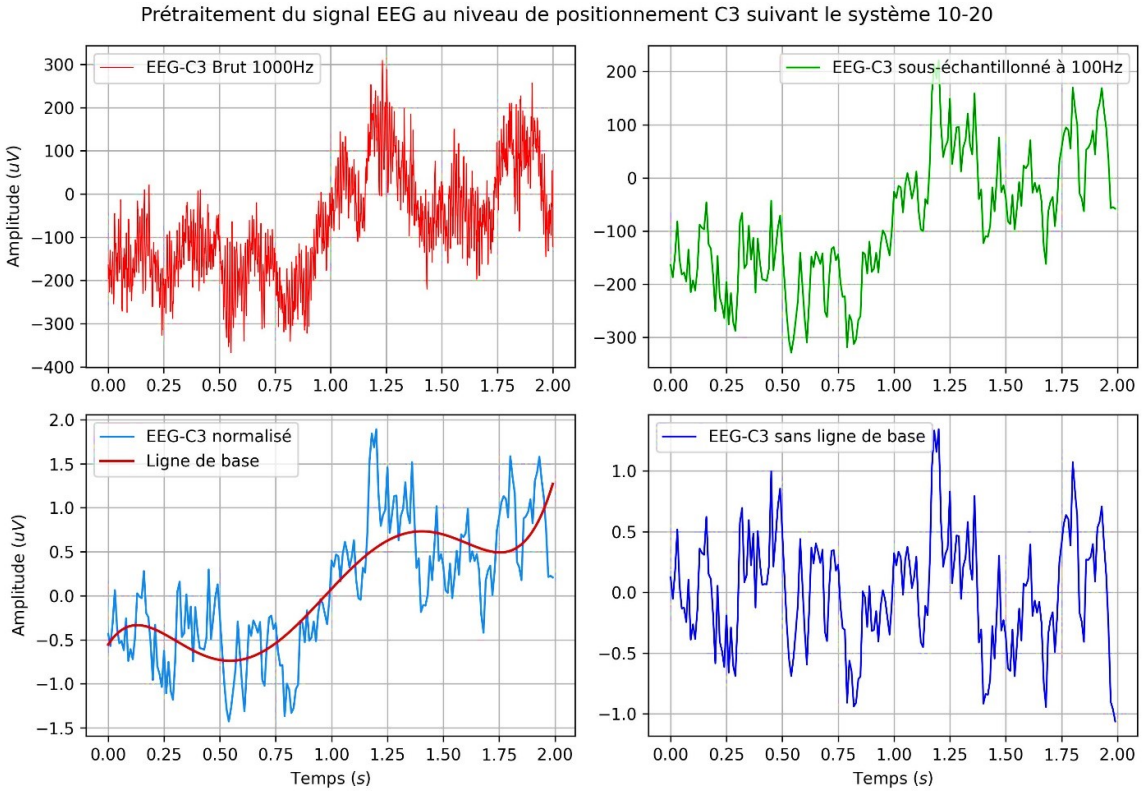


Figure V.3. Prétraitement du signal EEG mesuré au niveau du positionnement C3 suivant le système international 10-20 : (A) Signal EEG-C3 brut à 1000 Hz, (B) Signal EEG-C3 sous-échantillonné jusqu'à 100 Hz, (C) Signal EEG-C3 normalisé à l'aide de l'algorithme Robust-Scaler, la ligne de base (en marron) représente la dérive orignal du signal, (D) Signal EEG-C3 final normalisé et rectifié.

V.4.6. Fonctions de calculs analytiques et statistiques

Nous avons incorporé un autre module appelé '*dataCalculator.py*' dans notre bibliothèque *Step2bci*, qui comprend une série de fonctions permettant d'effectuer des analyses et des calculs statistiques. Ces fonctions comprennent l'estimation de la puissance, le calcul de la variance, l'analyse de la covariance, le calcul de la corrélation et le calcul de la moyenne des données. Ces outils analytiques sont particulièrement utiles lors des étapes intermédiaires du traitement des données, car ils éliminent la nécessité de réécrire le code à partir de zéro. En

exploitant les capacités du module '*dataCalculator.py*', les utilisateurs peuvent effectuer efficacement des calculs essentiels sans compromettre l'intégrité des données.

La figure V.4 présente des fonctions mises en œuvre dans ce sous-module fournissant une représentation visuelle des différents calculs analytiques et statistiques disponibles.

```python
def data_power_n(inputData=None, power=2, t_ax=0):
    data = inputData.copy()
    return data ** power


def data_logarithm(inputData=None, mode='dec'):
    # Il y a "dec", "bin", "nep"
    if mode.lower() == 'dec':
        data = inputData.copy()
        return np.log10(data)
    elif mode.lower() == 'bin':
        data = inputData.copy()
        return np.log2(data)
    elif mode.lower() == 'nep':
        data = inputData.copy()
        return np.log(data)
    else:
        return 'Erreur, mode inconnu !'


def data_variance(inputData=None, t_ax=0):
    data = inputData.copy()
    return np.var(data, axis=t_ax)


def data_covariance(inputData1=None, inputData2=None):
    data1 = inputData1.copy()
    data2 = inputData2.copy()
    return np.cov(data1, data2)


def data_correlation(inputData1=None, inputData2=None, relation="linear", dist="gauss"):
    data1 = inputData1.copy()
    data2 = inputData2.copy()
    if relation.lower() == "linear" and dist.lower() == "gauss":
        pearson = True
        spearman = False
```

Figure V.4. Implémentation des fonctions de calculs analytiques et statistiques.

V.4.7. Fonctions de l'analyse spectrale et fréquentielle

Le module le plus important du package "*preproc*" est l'analyseur de données '*dataAnalyzer.py*' qui regroupe plusieurs fonctions permettant d'effectuer des analyses spectrales avancées. Ces fonctions utilisent des techniques mathématiques spécifiques pour explorer les caractéristiques fréquentielles des signaux EEG.

L'analyse spectrale est un outil puissant pour l'analyse des signaux et consiste à étudier la distribution de puissance des signaux dans le domaine fréquentiel. Une des méthodes couramment utilisées est l'estimation de la densité spectrale de puissance (DSP) à l'aide de la

transformée de Fourier rapide FFT (*Fast Fourier Transform*) [Ameera A. et al., 2019]. La DSP d'un signal x(t) est donnée tout simplement par l'équation :

$$DSP(f) = |X(f)|^2 \quad \text{(V.3)}$$

Où :

DSP(f) représente la puissance à la fréquence *f*.

X(f) est la transformée de Fourier du signal EEG x(t) quelconque.

L'analyse spectrale permet d'identifier les composantes fréquentielles significatives et d'explorer les caractéristiques temporelles associées à ces composantes. En d'autre terme, la DSP permet d'observer la répartition de l'énergie du signal sur l'ensemble du spectre fréquentiel.

V.4.7.1. Densité spectrale de puissance (DSP)

La DSP permet d'identifier les composantes de fréquence significatives au sein d'un signal. Par exemple, dans le contexte de l'EEG, des pics de puissance distinctifs peuvent être observés dans des plages de fréquences spécifiques, telles que les ondes alpha (8-12 Hz) ou les ondes bêta (13-28 Hz). Ces caractéristiques de fréquence fournissent des informations pertinentes sur l'activité cérébrale et trouvent des applications dans divers domaines, notamment les interfaces cerveau-ordinateur.

V.4.7.2. Spectrogramme via la transformée de Fourier à Court Terme (STFT)

La représentation spectrogramme est une visualisation de l'évolution spectrale d'un signal EEG dans le temps. Elle est obtenue en appliquant la transformée de Fourier à court terme STFT (*Short Time Fourier Transform*) sur des segments de signal [Shovon, T. H., 2019]. La STFT d'un signal *x(t)* est donnée par :

$$STFT\{x(t)\}(\tau, w) = X(\tau, w) \int_{-\infty}^{+\infty} x(t) w(t-\tau) e^{-i2\pi} \quad dt \quad \text{(V.4)}$$

Où w(t) est une fenêtre temporelle et f est la fréquence. La représentation spectrogramme permet de détecter les changements de fréquence et d'intensité à travers différentes périodes temporelles.

V.4.7.3. Périodogrammes

Les périodogrammes : traditionnel, de Welch et multitaper sont des méthodes utilisées pour calculer l'estimation de la DSP d'un signal non stationnaire.

- Le périodogramme traditionnel a été introduit par A. Schuster en 1898 [Stoica P., & Moses R. L., 2005], il est basé sur une fenêtre temporelle rectangulaire et donne une estimation brute de la DSP. Pour obtenir l'estimation du périodogramme, le signal *x(n)* est divisé en un nombre fixe de segments non chevauchants de longueur égale, cette approche est relativement simple à implémenter et efficace sur le plan computationnel. L'un de ses principaux inconvénients est sa sensibilité aux variations des données, ce qui entraîne une variance élevée de la densité spectrale estimée. Il peut en résulter des estimations imprécises et bruyantes, en particulier dans les cas où le signal a un faible rapport signal/bruit.
- Peter D. Welch [Welch P., 1967] a proposé en 1967 une méthode améliorée appelée périodogramme de Welch, qui permet une estimation plus précise et moins bruitée de la densité spectrale de puissance (DSP) d'un signal EEG, cette approche consiste à diviser le signal EEG en segments qui partiellement se chevauchent, à appliquer une fonction de fenêtre *w(n)* à chaque segment *x(n)* pour réduire les fuites spectrales et les effets de la variance, puis à calculer le périodogramme pour chaque segment et finalement à faire la moyenne des périodogrammes individuels.
- Le périodogramme multitaper, qui a été introduite pour la première fois par David Thomson en 1982 [Thomson, D. J., 1982], est une méthode avancée utilisée pour estimer avec précision la densité spectrale d'un signal. Elle consiste à diviser le signal en segments se chevauchant et à appliquer un ensemble de tapers orthogonaux spéciaux appelés séquences sphéroïdales discrètes prolatées DPSS (*Discrete Prolate Spheroidal Sequences*) ou séquences de Slepian. Ces tapers sont conçus pour concentrer l'énergie dans une plage de fréquences spécifique et minimiser les fuites en dehors de cette plage.

Les algorithmes de calcul des périodogrammes traditionnel, de Welch et multitaper sont presque similaires. La principale différence réside dans la segmentation et la méthode de fenêtrage utilisés [Welch P., 1967 ; Birkelund Y. et al., 2003, Oppenheim, A. V., 2009].

L'algorithme de l'estimation spectrale pour les trois approches consiste à cinq étapes :

1- Segmenter le signal : diviser le signal en segments non chevauchants (le cas traditionnel) ou partiellement chevauchants (le cas de Welch et multitaper) de longueur L.

2- Choisir un ensemble de *M* fenêtres (le cas traditionnel et de Welch) ou *M* tapers orthogonaux (le cas multitaper) d'où la propriété d'orthogonalité des fenêtres garantit que leurs produits internes sont nuls, ce qui assure l'indépendance des estimations spectrales obtenues à partir de chaque fenêtre.

3- Créer l'ensemble de segments fenêtrés qui présentent de meilleures propriétés de localisation temps-fréquence. Cela peut s'exprimer mathématiquement comme :

$$x_{wm}(n) = x_m(n).w_m(tn) \quad \text{(V.5)}$$

Où : $x_m(n)$ est le m^ième^ segment original du signal EEG x(t).

$w_m(n)$ est la m^ième^ fenêtre.

$x_{wm}(n)$ est le signal EEG fenêtré et échantillonné.

4- Calculer les périodogrammes individuels :

$$P_m(k) = |X_m(k)|^2 = \left|\sum_{n=0}^{N-1} x_{wm}(n).e^{-2\pi j\frac{kn}{N}}\right|^2 \quad \text{(V.6)}$$

Où : $P_m(k)$ représente l'estimation du périodogramme pour le m^ième^ segment fenêtré et $X_m(k)$ représente la transformée de Fourier discret TFD du segment fenêtré.

5- Moyenner les périodogrammes individuels : l'estimation finale de la PSD est obtenue en moyennant les K périodogrammes calculés à l'étape précédente :

$$PSD(k) = \frac{1}{M}\sum_{m=0}^{M-1} P_m(k) = \frac{1}{M}\sum_{m=0}^{M-1}\left|\sum_{n=0}^{N-1} X_{wm}(n).e^{-2\pi j\frac{kn}{N}}\right|^2 \quad \text{(V.7)}$$

Pour : $0 < k \leq N$

Où : $PSD(k)$ représente l'estimation finale de la densité spectrale et la somme est effectuée sur tous les M périodogrammes.

N est la longueur du signal discret x(n).

M désigne le nombre total de segments ou de fenêtres en lesquels le signal est divisé.

La figure V.5 présente une comparaison visuelle des DSP obtenues à partir des trois méthodes (traditionnelle, de Welch et multitaper) pour un signal enregistré à l'emplacement C3. Cette analyse graphique met en évidence les différences de forme, d'amplitude et de distribution spectrale entre les trois approches.

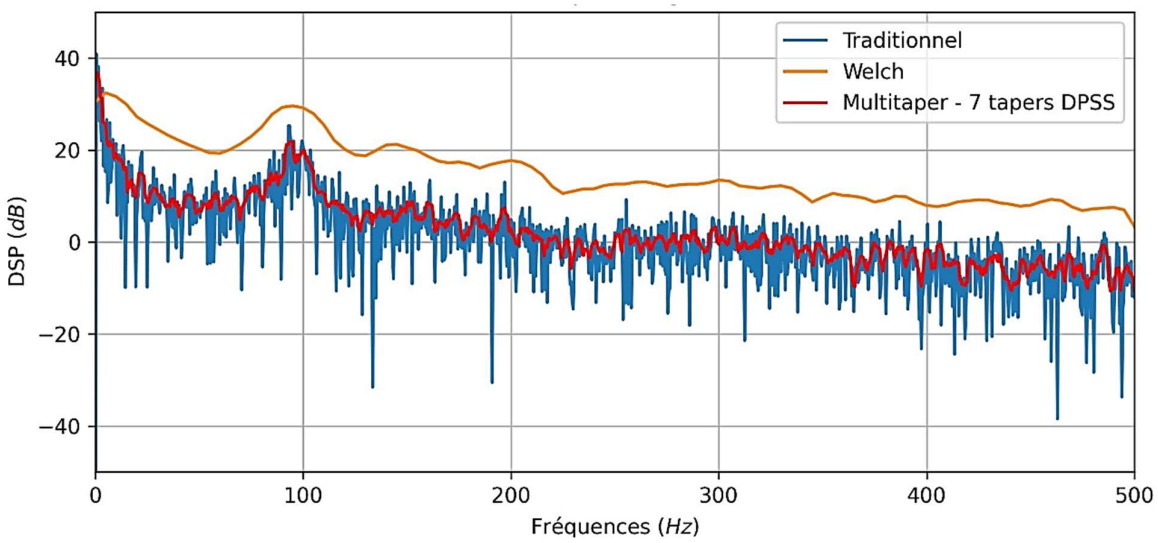


Figure V.5. Périodogrammes traditionnel, de Welch et Multitaper.

Il est clairement observé que la méthode traditionnelle présente des estimations bruyantes avec une variance élevée, tandis que la méthode de Welch présente moins de bruit et un lissage plus prononcé, mais une précision inférieure à celle de la méthode multitaper. Ces observations soulignent les compromis entre bruit, résolution et précision que chaque méthode offre dans l'estimation de la DSP.

V.4.7.4. Analyse de la puissance de bande

L'analyse de la puissance de bande [Lahane, P. et al., 2019] : elle consiste à quantifier la puissance contenue dans des plages spécifiques de fréquences. Par exemple, la puissance du rythme bêta (13-28 Hz) peut être calculée en intégrant la DSP dans la plage de fréquences correspondante. Cela permet de caractériser l'activité cérébrale dans différentes gammes de fréquences.

Les surfaces colorées dans la figure V.6 illustrent les puissances spectrales de la bande bêta relatives à la puissance totale du signal EEG-C3 obtenues à partir des DSP de Welch ou multitaper.

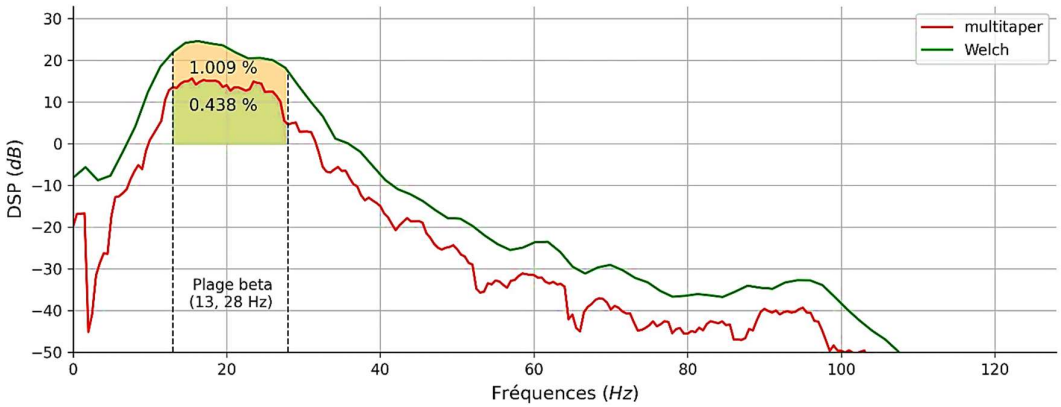


Figure V.6. Comparaison de la puissance de bande (Rythme bêta) : Welch vs. Multitaper.

Outre les fonctions susmentionnées, le module '*dataAnalyzer.py*' comprend une série d'autres fonctions utiles pour compléter l'analyse des données. Ces fonctions comprennent notamment l'analyse de la puissance moyenne, l'application de la transformation de Fourier à court terme STFT et la génération de spectrogrammes, parmi d'autres fonctionnalités, comme illustré dans la figure V.7. Chacune de ces fonctions offre des outils supplémentaires pour examiner et comprendre les caractéristiques spectrales et temporelles des signaux EEG. En incorporant ces capacités étendues dans le module '*dataAnalyzer.py*', la boîte à outils *Step2bci* améliore sa fonctionnalité globale et donne aux utilisateurs et aux praticiens des moyens avancés d'explorer et d'interpréter les données EEG.

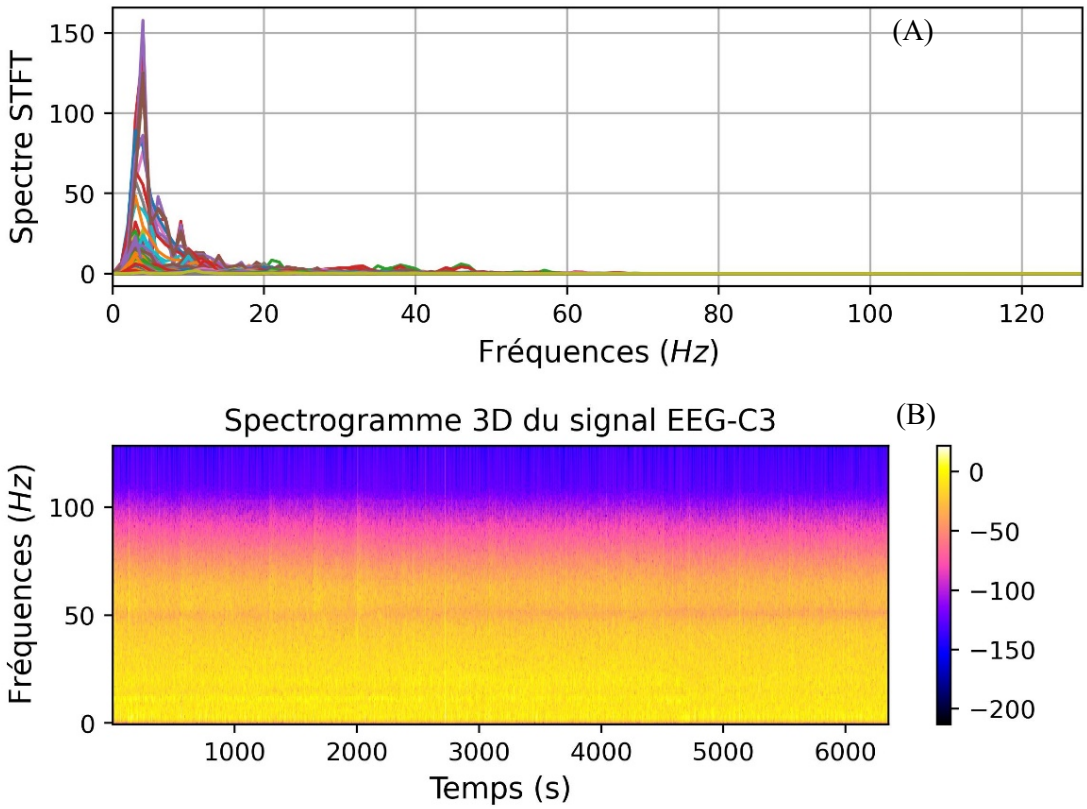


Figure V.7. (A) Spectre par la transformée de Fourier à Court Terme STFT, (B) Spectrogramme associé du signal EEG au niveau de C3.

V.4.8. Décomposition en ondelettes et analyse multi-résolution

L'analyse dans le domaine temporel permet d'obtenir des informations directes sur les caractéristiques temporelles d'un signal, telles que son amplitude, sa durée et ses propriétés temporelles. Cependant, cette approche présente des limites lorsqu'il s'agit d'extraire des informations précises sur la composition fréquentielle d'un signal complexe ou non-stationnaire telle que l'électroencéphalographie (EEG). En revanche, l'analyse dans le domaine fréquentiel se concentre sur la transformée de Fourier discret DFT (*Discrete Fourier Transform*), ce qui permet d'analyser les signaux en régime permanent et d'identifier les composantes fréquentielles individuelles. Toutefois, cette méthode a pour inconvénient de sacrifier la résolution temporelle, rendant difficile l'analyse des événements transitoires ou des signaux présentant des variations rapides de fréquence.

Afin de surmonter ces limitations, une approche couramment utilisée est l'analyse temps-fréquence basée sur la transformée STFT, qui permet d'obtenir des spectrogrammes (connu aussi comme *Transformée de Gabor*). Cette méthode offre une pondération équilibrée entre le temps et la fréquence, permettant de visualiser comment le contenu fréquentiel évolue au fil du temps.

Cependant, il est important de souligner que cette analyse présente des résolutions temporelle et fréquentielle réduites par rapport à l'analyse de séries temporelles et à la transformée de Fourier [Murugappan, M. et al, 2008]. Un compromis doit être trouvé entre la résolution temporelle et la résolution fréquentielle lors de l'utilisation de cette approche. Par exemple, l'utilisation d'une fenêtre temporelle plus courte permet d'obtenir une meilleure résolution temporelle, mais au détriment de la résolution fréquentielle, et vice versa. Ainsi, il est essentiel de prendre en compte ces facteurs pour obtenir une représentation appropriée du signal étudié.

La décomposition en ondelettes, quant à elle, offre une analyse multi-résolution en identifiant les composantes de fréquence à différentes échelles [Bajaj N., 2020]. Elle permet une excellente localisation temporelle, ce qui la rend adaptée pour analyser des signaux irréguliers ou non stationnaires et capturer des changements abrupts ou des discontinuités. Cependant, la décomposition en ondelettes peut nécessiter des ressources de calcul plus importantes, exige une sélection minutieuse des paramètres et présente des limites lors de l'analyse de contenus à haute fréquence en raison du compromis entre résolution temporelle et résolution fréquentielle.

- *Qu'est-ce que la décomposition en ondelettes ?*

La décomposition en ondelettes est largement utilisée en tant que technique mathématique puissante dans les contextes des BCIs pour analyser les signaux cérébraux et obtenir une représentation multi-résolution des activités cérébrales. Elle est basée sur le concept des ondelettes, qui sont des fonctions localisées à la fois dans le domaine temporel et fréquentiel et elles présentent deux caractéristiques fondamentales : l'échelle et la localisation. L'échelle définit le degré d'extension ou de compression d'une ondelette [Loughlin, P., & Cohen, L., 2010]. D'autre part, la localisation spécifie la position temporelle ou spatiale de l'ondelette.

Mathématiquement, la décomposition en ondelettes implique la convolution du signal avec un ensemble de fonctions de base (ondelettes). Ces fonctions d'ondelettes sont des versions mises à l'échelle et décalées d'une ondelette prototype unique (ondelette mère), notée $\psi(t)$. Le paramètre d'échelle, s, contrôle la largeur de la fonction d'ondelette dans le domaine temporel, tandis que le paramètre de translation, τ, détermine sa position le long de l'axe temporel (figure V.8).

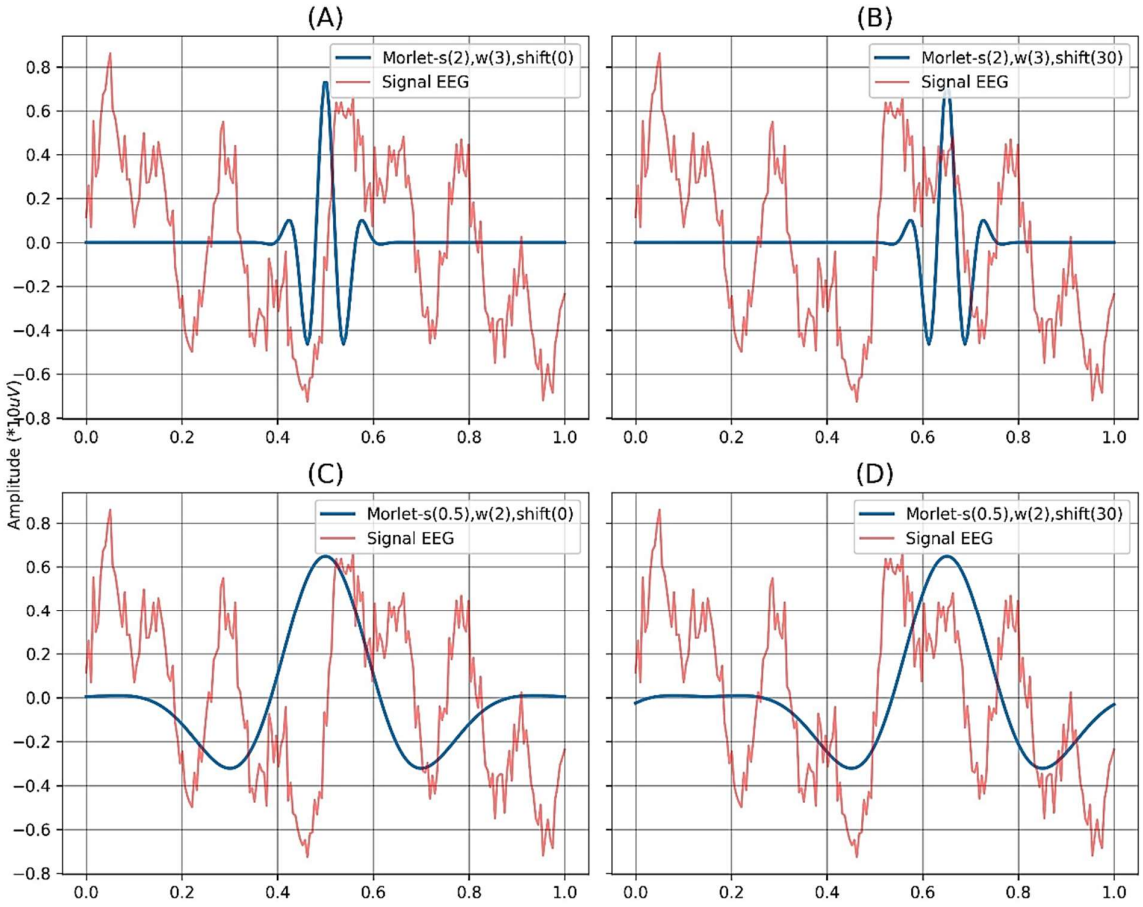


Figure V.8. Convolution du signal EEG avec une ondelette de Morlet : 1) Mise en échelle pour la détection des fréquences élevées : (A) Localisation centrée, (B) Ondelette Morlet décalée dans le temps. 2) Mise en échelle pour la détection des basses fréquences : (C) Localisation centrée, (D) Ondelette de Morlet décalée dans le temps.

Le processus de décomposition en ondelettes commence par le signal d'origine, *x(t)*, et applique une série de dilatations et de translations à la fonction d'ondelette pour générer un ensemble de fonctions de base. Les fonctions d'ondelettes dilatées et décalées peuvent s'exprimer mathématiquement comme suit :

$$\psi(s,\tau)(t) = \frac{1}{\sqrt{s}}\psi(\frac{t-\tau}{s}) \qquad \text{(V.8)}$$

Où :

s > 0 et τ sont les paramètres d'échelle et de translation, respectivement.

À chaque échelle ou niveau de décomposition, le signal *x(t)* est convolé avec ces fonctions d'ondelette mises à l'échelle et décalées. Cela donne un ensemble de coefficients qui

représentent la contribution de chaque fonction d'ondelette à cette échelle au signal. L'opération de convolution peut s'écrire mathématiquement comme suit :

$$C(s,\tau) = \int_{-\infty}^{+\infty} x(t)\psi(s,\tau)(t)dt \quad \text{(V.9)}$$

Les coefficients *C(s, τ)* sont calculés pour différentes valeurs de *s* et *τ*, capturant ainsi des informations sur le contenu fréquentiel du signal et sa localisation temporelle à chaque échelle. Le choix de la fonction d'ondelette *ψ(t)* (par exemple : ondelette de Morlet, Haar, Daubechies, symlet, etc.) détermine les propriétés spécifiques de la décomposition, telles que le niveau de lissage et de localisation.

En appliquant de manière itérative le processus de décomposition en ondelettes, une structure hiérarchique est formée, connue sous le nom de transformée en ondelettes ou d'arbre de décomposition en ondelettes. La transformée en ondelettes fournit une représentation multi-résolution du signal, les différents niveaux correspondant à différentes échelles de résolution en fréquence et en temps.

Dans le package de prétraitement '*preproc'* de la bibliothèque *Step2bci*, on a implémenté un module appelé '*waveletAnalyzer.py*' qui comprend deux fonctions pour l'analyse et la décomposition en ondelettes, celles sont : *'cwt_analysis ()'* et *'wavedec_dwt_analysis ()'* qui s'appuient sur la bibliothèque *pyWavelets* [Lee, G. et al., 2019].

La fonction *'cwt_analysis ()'* prend en entrée les données, les échelles et le type d'ondelette, et renvoi les coefficients de transformation d'ondelette ainsi que les fréquences correspondantes. Ces coefficients indiquent dans quelle mesure chaque ondelette correspond au signal à différentes échelles et positions.

Quant à la fonction *'wavedec_dwt_analysis ()*', elle prend les données, le type d'ondelette et le niveau maximal de décomposition. Elle effectue une décomposition en sous-bandes du signal à différentes échelles de résolution en fréquence et en temps. Les premiers coefficients représentent les composantes à basse résolution (coefficients d'approximation), tandis que les coefficients suivants représentent les coefficients de détails à résolutions plus élevées.

Ces fonctions fournissent des outils essentiels pour l'analyse et la décomposition d'ondes, permettant une analyse approfondie et une interprétation des caractéristiques des signaux dans le domaine du traitement des signaux EEG.

La décomposition en ondelettes présente plusieurs avantages par rapport à d'autres techniques de traitement de signal. Les fonctions de base en ondelettes ont un support compact à la fois dans les domaines temporel et fréquentiel, ce qui permet une analyse localisée des caractéristiques du signal. Cette propriété de localisation rend la décomposition en ondelettes adaptée à l'analyse des signaux présentant des composantes transitoires ou localisées. De plus, la nature multi-résolution de la décomposition en ondelettes permet l'analyse des signaux à différents niveaux de détail, offrant des informations à la fois sur les caractéristiques globales et locales du signal.

V.4.9. Segmentation (Epoching) des données et l'extraction des essais

Le package "*preproc*" de *Step2bci* comprend deux modules supplémentaires : '*dataEpocher.py*' et '*trialsProcessor.py*'. Ces modules offrent des fonctionnalités de prétraitement des signaux EEG dans le contexte spécifique de la réalisation d'une BCI.

Le module *dataEpocher.py* contient une fonction appelée *'data_epoching ()'* qui est utilisée pour découper les données en segments temporels appelés époques. Cette fonction prend en compte les marqueurs associés aux époques, ainsi que les bornes temporelles et les autres paramètres pertinents. Elle renvoie les données époquées, les temps associés à chaque époque, les classes des époques et un dictionnaire de correspondance entre les classes et leurs ordres d'occurrence dans les données.

Le module '*trialsProcessor.py*' comprend plusieurs fonctions. La fonction '*targets_count ()*' compte le nombre de classes uniques présentes dans une liste de cibles. La fonction *'trials_extract ()'* pour extraire les essais (trials) à partir des données d'entrée. Les essais sont spécifiés par une liste de marqueurs (*trials*) et les cibles correspondantes (*targets*). La fonction découpe les données en segments de taille fixe (trial size) et renvoi un dictionnaire contenant les essais pour chaque classe. La fonction *'data_class_composer ()*' permet de regrouper des données de différentes classes dans un dictionnaire avec des étiquettes spécifiées. Enfin, la fonction *'split_data_classes ()'* est utilisée pour partitionner les données en ensembles d'entraînement et de test, ce qui permet de préparer les données pour des tâches ultérieures telles que la classification.

En utilisant les modules '*dataEpocher.py*' et '*trialsProcessor.py*' ensemble, il devient possible d'effectuer diverses opérations de prétraitement et de manipulation des données EEG. Ces opérations incluent la découpe des données en époques (figure V.9), l'extraction des essais basée sur les marqueurs [Tariq, M. et al., 2018], la création de jeux de données d'entraînement et

de test, ainsi que la gestion des classes des essais. L'utilisation combinée de ces modules facilite considérablement les étapes de prétraitement et de manipulation des données EEG.

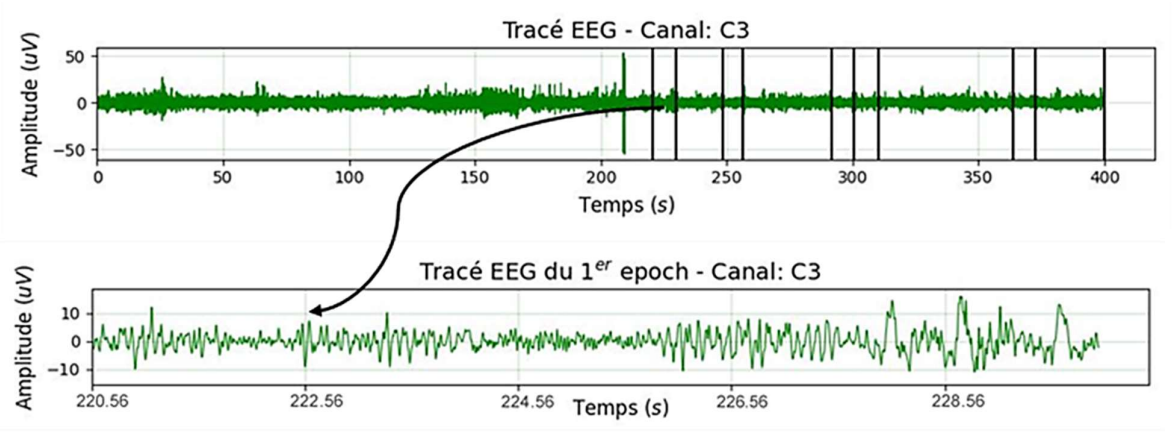


Figure V.9. (A) Série temporelle du signal EEG-C3 dans l'intervalle (0, 400 s) comportant les marqueurs associés aux essais (trials), (B) Extraction du 1er essai (essai 0).

V.5. Post-traitement du signal EEG

Le post-traitement est une étape essentielle dans le flux de traitement des signaux EEG dans le contexte d'une BCI. Il intervient après le prétraitement, qui comprend le processus de filtrage, la correction de la ligne de base, l'époching, l'extraction des essais, etc.

Le post-traitement se concentre sur des tâches spécifiques. Il englobe l'extraction des caractéristiques des signaux EEG, la sélection des caractéristiques les plus discriminantes, la réduction de la dimensionnalité des données à par l'analyse en composantes principales PCA (*Principal Component Analysis*) [Yu, X. et al., 2014] et indépendantes ICA (*Independent Component Analysis*) [Radüntz, T. et al., 2015]. De plus, le post-traitement inclut la classification des signaux EEG en utilisant des modèles d'apprentissage automatique ou d'apprentissage profond, ainsi que l'optimisation de ces modèles grâce à la validation croisée et l'ajustement des hyperparamètres.

On a développé un package appelé "*postproc*" au sein de la bibliothèque *step2bci*, dédié exclusivement aux fonctions de post-traitement de la BCI. Ce package offre un ensemble d'outils et de fonctionnalités liées au post-traitement des signaux EEG. En particulier, le package fournit des méthodes de classification avancées, en introduisant des algorithmes tels que le k-plus proches voisins KNN (*K-Nearest Neighbors*), l'analyse discriminante linéaire LDA (*Linear Discriminant Analysis*), la Machine à vecteurs de support SVM (*Support Vector Machine*), etc.

De plus, on a également créé un package distinct nommé "*deepproc*", qui met l'accent particulièrement sur les techniques de classification de l'activité cérébrale en utilisant des techniques d'apprentissage profond. Ces deux packages complémentaires offrent des fonctionnalités avancées et flexibles pour le post-traitement et l'analyse des signaux EEG dans le domaine des BCIs.

En utilisant des techniques avancées de post-traitement, on peut améliorer la sélectivité des caractéristiques extraites, augmenter la précision de la classification et optimiser les paramètres du modèle pour une meilleure performance globale du système BCI.

V.5.1. Extraction des caractéristiques

Le package *postproc* est composé de trois modules dédiés à l'extraction, la sélection et le tracé des caractéristiques, respectivement : '*featuresExtractor.py*', '*featuresSelector.py*' et '*featuresPlotter.py*'.

Le module '*featuresExtractor.py*' est chargé d'extraire les caractéristiques des données d'essai EEG. Il contient diverses fonctions :

V.5.1.1. Extraction multi-résolution par décomposition en ondelettes

La fonction *'wavelets_feat ()'* qui utilise la fonction *'wavedec_dwt_analysis ()'* déjà discutée dans la section de prétraitement. Cette fonction requiert en entrée des données, des indices d'essais, le type d'ondelette mère, le niveau maximum, le niveau à extraire, etc. elle extrait les époques de données correspondant aux essais spécifiés et applique une analyse en ondelettes en utilisant les paramètres donnés et ensuite calcule les différentes mesures statistiques telles que la médiane, l'écart-type et la moyenne à partir des coefficients d'ondelettes de canaux et de niveaux spécifiques. Ces caractéristiques donnent un aperçu des caractéristiques de l'activité cérébrale capturée par les signaux EEG. Finalement, *wavelets_feat ()* renvoi les caractéristiques extraites pour des tâches de classification.

V.5.1.2. Extraction par les motifs spatiaux communs (filtrage spatiale)

Nous avons implémenté la fonction *'csp_feat ()'* qui fait partie du module '*featuresExtractor.py*', elle exécute l'algorithme des motifs spatiaux communs CSP (*Common Spatial Pattern*) pour l'extraction de caractéristiques à partir de données EEG multicanal. L'algorithme CSP est une technique de traitement de signal largement utilisée en EEG pour identifier les motifs spatiaux discriminants associés à différentes activités cérébrales [Rayatnia, A., & Khanbabaie, R., 2019]. Le principe de l'algorithme CSP dans les BCIs de trouver des filtres spatiaux optimaux qui maximisent la différence de variance entre deux classes d'activité

cérébrale spécifiques. Les données d'entrée sont stockées dans un dictionnaire *inputData*, où chaque classe est associée à un tableau *X* de dimensions (*nombre d'essais, nombre d'échantillons, nombre de canaux*).

La fonction *'csp_feat ()'* repose sur les étapes suivantes :

- Calcul de la matrice de covariance empirique pour chaque essai dans une classe :

$$COV_{tr} = \frac{X_{tr}^T X_{tr}}{N_s} \quad \text{(V.10)}$$

Où : N_s est le nombre d'échantillons dans chaque essai (trial) de chaque classe.

X_i de dimensions (*nombre d'échantillons, nombre de canaux*) est le tableau des données d'entrées pour chaque essai de chaque classe.

- Calcul de la matrice de covariance globale pour chaque classe :

$$COV_{cls} = \frac{1}{N_{tr}} \sum_{tr=0}^{N_{tr}-1} COV_{tr} \quad \text{(V.11)}$$

- Accumuler les matrices de covariance de toutes les classes COV_j dans la matrice de covariance globale σ à l'aide de la formule :

$$\sigma = \sum_{cls=}^{N_c-1} COV_{cls} \quad \text{(V.12)}$$

- Décomposer en valeurs singulières (SVD) pour factoriser la matrice σ en un produit de trois matrices (U, l et V) telle que :

$$\sigma = U.S.V^T \quad \text{(V.13)}$$

Où : U est une matrice orthogonale de forme (*n, n*), où m est le nombre de lignes dans σ.

S est une matrice diagonale de forme (*n, m*), où *n* et *m* sont, respectivement, le nombre de lignes et de colonnes de σ. Les éléments diagonaux de S sont les valeurs singulières de σ.

V^T (Vecteurs singuliers), est la transposée d'une matrice orthogonale de forme (*m, m*), où *m* est le nombre de colonnes de σ.

- Calculer la matrice de blanchiment P à l'aide de la formule :

$$P = U.diag(S^{-1/2}) \quad \text{(V.14)}$$

- Calculer la matrice B des vecteurs singuliers gauches associés à la dernière classe du filtre CSP à l'aide de la formule :

$$B.\Sigma.Z^T = P^T.COV_{-1}.P \quad \text{(V.15)}$$

B est une matrice orthogonale obtenue par la décomposition en valeurs singulières de la matrice de covariance de la dernière classe COV_{-1}. Ces vecteurs ne sont pas nécessairement orthogonaux, mais ils sont utilisés pour calculer la matrice de filtre spatial.

- Si le paramètre *fitted = True* et qu'une matrice du filtre spatial (W) est fournie, elle est utilisée telle quelle. Sinon, la matrice de W est calculée en utilisant cette formule :

$$W = P.B \quad \text{(V.16)}$$

- Effectuer la projection des données EEG de chaque classe et de chaque essai sur l'espace CSP à l'aide de la formule :

$$CSP_{tr} = X_{tr}.W \quad \text{(V.17)}$$

- Finalement, les caractéristiques CSP sont stockées dans un dictionnaire où les clés correspondent aux étiquettes de classe et les valeurs correspondent aux caractéristiques CSP pour tous les essais de chaque classe. La fonction renvoie les données caractéristiques CSP et la matrice du filtre spatial (W).

Les filtres spatiaux optimaux, identifiés par l'algorithme CSP, maximisent la variance entre les différentes classes, améliorant ainsi la classification des activités cérébrales associées à chaque classe [Reuderink, B., & Poel, M., 2008]. Ces caractéristiques CSP peuvent ensuite être utilisées comme entrée pour des algorithmes de classification afin de décoder les intentions de l'utilisateur à partir des signaux EEG.

V.5.1.3. Extraction de la variance logarithmique

La fonction *'logvar_feat ()'* est utilisée après l'extraction des caractéristiques CSP. Ensuite, elle est appliquée séparément aux vecteurs des caractéristiques CSP de chaque classe. Cela permet de calculer la variance logarithmique pour chaque classe spécifique. La fonction logarithme est utilisée pour accentuer les variations relatives entre les valeurs de variance. Les résultats sont ensuite stockés dans un dictionnaire *trials_logvar*, avec les étiquettes des classes comme clés et les valeurs de variance logarithmique correspondantes comme valeurs. Ainsi, la fonction '*logvar_feat ()'* permet de quantifier les différences de variance entre les classes après

l'extraction des caractéristiques CSP, ce qui peut être utile pour la classification ultérieure des signaux EEG [Aljalal M. et al., 2018].

V.5.1.4. Extraction de la puissance des sous-bandes fréquentielles

La fonction '*subBP_feat ()'* a été développée dans le but d'extraire des caractéristiques de sous-bandes de fréquences à partir des données EEG. Son objectif principal est de capturer et les caractéristiques de puissance présentes dans des sous-bandes spécifiques du signal d'entrée [Bahri Z. et al., 2014]. Lorsqu'elle reçoit les données *inputData* en entrée, la fonction commence par vérifier que les données sont au format attendu. Elle vérifie les dimensions des sous bandes du signal et les ajuste si nécessaire pour assurer leur cohérence. Ensuite, pour chaque essai, la fonction sélectionne la fenêtre temporelle d'intérêt. À l'intérieur de cette dernière, les valeurs de puissance de ses sous-bandes spécifiées sont calculées. Pour représenter les caractéristiques de manière efficace, le logarithme des puissances de sous-bandes concaténées est calculé. Cette transformation logarithmique met en évidence les différences relatives entre les valeurs de puissance et offre une représentation plus significative. Enfin, la fonction renvoie le tableau des caractéristiques contenant les puissances des sous-bandes extraites.

V.5.1.5. Extraction des caractéristiques par l'entropie échantillonnée (*SampEn*)

La méthode d'extraction de caractéristiques EEG appelée "entropie échantillonnée" est basée sur la mesure de la complexité et la régularité des séries temporelles, son principe repose sur la comparaison de sous-séquences de taille définie dans la série temporelle [Wang, L. et al., 2012]. La similarité entre ces sous-séquences est évaluée en calculant les différences maximales entre les valeurs de chaque dimension de ces sous-séquences et les valeurs correspondantes dans les autres sous-séquences. Une similarité élevée entre les sous-séquences indique une plus grande régularité ou répétitivité dans la séquence temporelle, ce qui se traduit par une entropie échantillonnée plus faible.

La fonction *sampEntropy_feat ()* implémente cet algorithme de l'entropie échantillonnée pour quantifier la complexité des motifs temporels dans les données EEG en extrayant l'entropie échantillonnée des données fournies en tant que paramètre *inputData*. Elle fournit ainsi une mesure de la régularité et de la structure des signaux EEG, permettant une caractérisation précise des activités cérébrales. Certains paramètres tels que la dimension d'incrustation *emb_dim* et la tolérance *tol* influencent sur la valeur de l'entropie *SampEn*.

Dans un premier temps, la fonction '*sampEntropy_feat ()*' effectue une préparation des données d'entrée x en créant les vecteurs de modèle *X* de longueur m. Elle calcule ensuite les

distances entre les vecteurs modèles à l'aide de la distance de Tchebychev *d* afin d'obtenir les valeurs *A* et *B* tels que :

$$\sigma = std(x) \tag{V.18}$$

Si :

$$r = tol * \sigma \tag{V.19}$$

Alors :

$$A = \sum(d[X_{m+1}(i), X_{m+1}(j)] \leq r) \tag{V.20}$$

$$B = \sum(d[X_m(i), X_m(j)] \leq r) \tag{V.21}$$

Où :

σ : est l'écart type de la série temporelle *X*.

A est le nombre de paires de vecteurs modèles de longueur *m* qui ont une distance inférieure à *r*.

B est le nombre de paires de vecteurs modèles de longueur *m+1* qui ont une distance inférieure à *r*.

Sur la base de ces mesures, la fonction peut finalement renvoyer la valeur *SampEn* telle que :

$$SampEn = -log(\frac{A}{B}) \tag{V.22}$$

En appliquant la fonction *'sampEntropy_feat ()'* aux signaux EEG, on peut extraire des caractéristiques qui capturent la complexité et l'irrégularité de l'activité cérébrale. Ces caractéristiques peuvent servir à plusieurs objectifs, tels que la classification, la détection d'anomalies ou l'étude des effets de différentes conditions ou interventions sur l'activité cérébrale.

Le module '*featuresExtractor.py*' propose également une gamme d'algorithmes pour l'extraction de caractéristiques à partir des signaux EEG, comprenant l'exposant de Hurst (*Hurst Exponent*) [Aldea, R., & Tărniceriu, D., 2013], l'analyse de fluctuation détachée DFA (*Detrended Fluctuation Analysis*) [Pavlov, A., 2018], la dimension fractale de Higuchi HFD (*Higushi's Fractal Dimension*) [Bashashati, A. et a., 2003] et bien d'autres. Ces algorithmes

offrent des outils précieux pour analyser différentes facettes des données EEG. Par exemple, l'exposant de Hurst est utilisé pour quantifier la dépendance à long terme ou l'autosimilarité du signal, tandis que la DFA mesure la présence de motifs fractals liés aux propriétés statistiques du signal. La dimension fractale de Higuchi, quant à elle, caractérise la complexité du signal en examinant ses variations temporelles. Ces algorithmes peuvent être utilisés dans des recherches futures ou des protocoles personnalisés afin d'approfondir la compréhension de la dynamique EEG et de découvrir des motifs significatifs au sein des données.

V.5.2. Sélection et réduction de l'espace des caractéristiques

Le module '*featuresSelector.py*' se concentre sur la sélection des caractéristiques pertinentes à partir de l'ensemble extrait. Il met en œuvre des algorithmes tels que le sélecteur séquentiel SFS (*Sequential Features Selector*), l'analyse en composantes principales PCA et l'analyse en composantes indépendantes ICA afin d'identifier les caractéristiques les plus discriminantes pour les tâches de classification. En réduisant la dimensionnalité de l'espace des caractéristiques, ce module améliore l'efficacité des calculs et atténue le risque de sur-apprentissage.

V.5.2.1. Sélection séquentielle des caractéristiques (SFS)

L'algorithme sélecteur séquentiel des caractéristiques SFS [Gottemukkula, V., & Derakhshani, R., 2011] prend en entrée l'ensemble des caractéristiques X_{init} à *N*-dimensions tels que $X_{init} = \{x_1, x_2, \ldots, x_N\}$, il renvoie un sous-ensemble de caractéristiques *X* à *M*-dimensions tels que $X_k = \{x_j |\ j=1, 2, \ldots, M\ ;\ x_j \in X_{init}\}$ où $M < N$ et $k= \{0, 1, \ldots, M\}$. Le nombre de caractéristiques sélectionnées *M*, doit être spécifié à priori. SFS commence par un ensemble vide et ajoute itérativement la meilleure caractéristique x_+ qui maximise une fonction de critère *J* lorsqu'elle est ajoutée au sous-ensemble de caractéristiques actuel donc :

$$x_+ = \underset{x \in (X_{init} - X_k)}{\operatorname{argmax}} \{J(X_k + x)\} \tag{V.23}$$

Ce qui donne :

$$X_{k+1} = X_k + x_+ \quad , \ k = k + 1 \tag{V.24}$$

Ce processus se poursuit jusqu'à ce que le nombre de caractéristiques souhaité soit atteint et le critère final soit satisfait. L'objectif de l'algorithme est de trouver le sous-ensemble de caractéristiques qui offre la meilleure performance de classification.

Notre contribution se situe au niveau de la fonction *'SFS_select ()'* que nous avons développée. Cette fonction utilise l'algorithme SFS pour sélectionner un sous-ensemble de

caractéristiques optimales, mais avec une approche plus avancée. Contrairement à l'approche traditionnelle qui se focalise sur un seul classifieur de manière exclusive, notre fonction entraîne un ensemble de classifieurs avec différents hyperparamètres en utilisant la technique de recherche en grille avec validation croisée (*GridSearchCV*). Cela nous permet d'explorer efficacement l'espace des classifieurs et des hyperparamètres, afin d'identifier la meilleure combinaison de caractéristiques associée au meilleur classifieur.

La fonction commence par initialiser les paramètres nécessaires, tels que la taille de la validation, le type de sélection (*Forward* ou *Backward*), le nombre de caractéristiques à sélectionner, etc. Ensuite, elle crée une liste d'estimateurs. Chaque classifieur (estimateur) est associé à un sélecteur de caractéristiques SFS correspondant. Après, un pipeline est créé pour chaque paire sélecteur-estimateur, permettant d'exécuter la sélection suivie de l'entraînement du classifieur. En utilisant la technique *GridSearchCV* avec la paire sélecteur-estimateur, les hyperparamètres seront utilisés dans la *GridSearchCV* pour trouver les meilleures performances du modèle pour chaque combinaison. Les modèles sont évalués en utilisant une métrique de performance (*scoring*) spécifiée, telle que l'exactitude (*précision*), etc.

Une fois que tous les modèles ont été évalués, la fonction retourne les résultats de la *GridSearchCV* ainsi que les pipelines associés à chaque paire sélecteur-classifieur. Ces résultats comprennent les meilleurs scores obtenus, les meilleurs hyperparamètres et les indices des caractéristiques sélectionnées par l'algorithme SFS.

V.5.2.2. Sélection par l'analyse en composantes principales (PCA)

L'algorithme PCA (*Principal Component Analysis*) [Zito, T. et al., 2009] vise à maximiser la variance des données le long des composantes principales, la première composante principale capturant la variance la plus élevée et les composantes suivantes capturant une variance décroissante. En sélectionnant un sous-ensemble de composantes principales, PCA vise à réduire la dimensionnalité des données EEG tout en préservant autant d'informations pertinentes que possible [Yu, X. et al., 2014]. Étant donné un ensemble de données X avec N observations et M caractéristiques, où $X = \{x_1, x_2, \ldots, x_N\}$ et $x_i = \{x_{i1}, x_{i2}, \ldots, x_{iM}\}$, l'algorithme peut être décrit succinctement comme suit :

- Calculer le vecteur moyen μ de X avec :

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \text{(V.25)}$$

- Centrer les données en soustrayant le vecteur moyen de chaque observation tel que :

$$X' = \{(x_1 - \mu), (x_2 - \mu), \dots, (x_N - \mu)\}$$
$$= \{x'_1, x'_2, \dots, x'_N\} \quad \text{(V.26)}$$

- Calculer la matrice de covariance COV des données centrées :

$$COV = \frac{1}{N}(X'X'^T) \quad \text{(V.27)}$$

- Effectuer la décomposition en valeurs propres de la matrice de covariance COV :

$$COV = V.U.V^T \quad \text{(V.28)}$$

  V est la matrice des vecteurs propres, où chaque colonne représente une composante principale.

  U est la matrice diagonale des valeurs propres.

- Trier les valeurs propres par ordre décroissant et choisir les k premiers vecteurs propres correspondant aux plus grandes valeurs propres pour former une matrice de projection P telle que

$$P = \{v_1, v_2, \dots, v_k\} \quad \text{(V.29)}$$

- Projeter les données centrées sur le sous-espace à k-dimensions couvert par les vecteurs propres sélectionnés :

$$X'' = X'.P \quad \text{(V.30)}$$

Où : X'' est l'ensemble de données transformé, où chaque ligne représente une observation dans l'espace réduit des caractéristiques.

La fonction fournie, *'PCA_select ()'*, utilise l'Analyse en Composantes Principales PCA pour sélectionner et réduire les caractéristiques. La fonction prend en entrée un ensemble de données *inputData* et le vecteur cible correspondant *targetData*. Elle effectue l'analyse PCA en créant une instance de la classe PCA avec le nombre de composantes spécifié *n_comp* et applique ensuite la méthode *'fit_transform ()'* aux données d'entrée. Ensuite la fonction renvoie les données sélectionnées et transformées *selected_data* ainsi que les données divisées *split_data* si la visualisation est activée ; sinon, *split_data* est défini sur *None*.

En résumé, la fonction *'PCA_select ()'* effectue une sélection PCA sur les données d'entrée, réduisant leur dimensionnalité tout en conservant les informations pertinentes. Elle

offre également des options de visualisation pour comprendre les caractéristiques sélectionnées ou les composantes principales. La fonction renvoie les données transformées et, éventuellement, les données divisées pour les besoins de la classification.

V.5.2.3. Sélection par analyse en composantes indépendantes (ICA)

L'algorithme ICA (*Independent Component Analysis*) est une technique qui vise à séparer des sources indépendantes d'un mélange de signaux observés, en supposant une distribution non gaussienne des signaux [Hyvärinen A., & Oja E., 2000]. Contrairement à d'autres méthodes telles que l'Analyse en Composantes Principales PCA, ICA utilise des statistiques d'ordre supérieur pour récupérer des signaux statistiquement indépendants. Dans le domaine des interfaces cerveau-ordinateur, les relevés EEG sont des enregistrements électriques des activités cérébrales. En règle générale, ces relevés EEG sont effectués à plusieurs endroits du cuir chevelu. Cependant, à chaque emplacement de lecture EEG, tous les signaux d'activité cérébrale sont mélangés (figure V.10), ce qui empêche de comprendre clairement le nombre de signaux sous-jacents qu'ils contiennent.

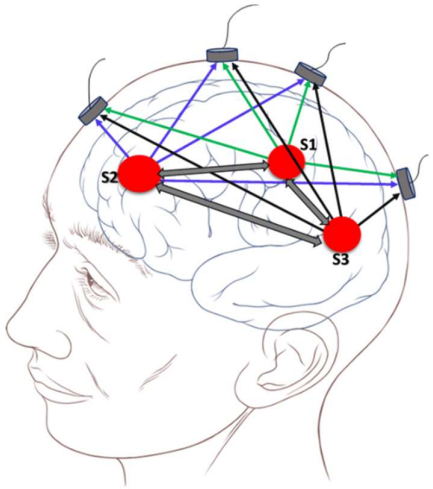


Figure V.10. Schématisation d'un mélange de l'activité cérébrale.

Avec un grand nombre d'électrodes EEG, l'ICA permettra de séparer les relevés d'activité cérébrale et d'obtenir une meilleure évaluation de l'activité neuronale globale. La technique ICA repose sur certaines hypothèses fondamentales :

- Non-Gaussianité des sources : les signaux sources doivent avoir des distributions non gaussiennes.

- Indépendance des sources : elle suppose que les signaux sources sont stochastiquement indépendants les uns des autres.

- Linéarité du mélange : les signaux observés sont le résultat de mélanges linéaires des signaux sources. Il existe une matrice de mélange qui relie les signaux sources aux signaux observés.

Un cadre mathématique plus formel pour l'ICA peut être établi en considérant ce qui suit : Étant donné un ensemble de données *X* composé de n signaux mélangés observés, l'objectif de l'ACI est d'estimer les signaux sources originaux *S*, l'algorithme ICA peut être représenté de la manière suivante :

1. Prétraitement :
   - Centrer le vecteur d'observation X(t) en soustrayant sa moyenne μ, donc on aura :

$$X_c(t) = X(t) - \mu \tag{V.31}$$

   - Appliquer une transformation de blanchiment du vecteur d'observation de sorte que ses composantes ne soient pas corrélées et donc présentent une variance unitaire.
   - Calculer la matrice de covariance empirique :

$$C = X_c . X_c^T \tag{V.32}$$

   - Effectuer la décomposition en valeurs propres de la matrice C telle que :

$$C = V . U . V^T \tag{V.33}$$

     Où : E contient les vecteurs propres et D est la matrice diagonale des valeurs propres.
   - Calculer la matrice de blanchiment :

$$P = U^{\left(-\frac{1}{2}\right)} . V^T \tag{V.34}$$

   - Blanchir les signaux observés :

$$\tilde{X} = P . X_c \tag{V.35}$$

2. Définir la matrice de démélange :
   - Choisir une valeur initiale aléatoire pour la matrice de démélange *W* de taille (*n x n*), chaque ligne de *W* correspond aux poids appliqués au mélange EEG pour estimer les signaux sources telle que :

$$S = W . \tilde{X} \tag{V.36}$$

3. Optimisation :

Maximiser l'indépendance statistique des signaux sources estimés S en minimisant la dépendance mutuelle ou en maximisant une mesure de non-gaussianité :

- La mesure de non-gaussianité J couramment utilisée est basée sur l'entropie négative telle que :

$$J(S) = H(S_gauss) - H(S) \quad \text{(V.37)}$$

Où : H représente l'entropie

- Trouver la matrice de démélange *W* qui maximise la somme des mesures de non-gaussianité des signaux sources tel que :

$$max(J(S)) = max\left(sum(J(S))\right) \quad \text{(V.38)}$$

Note : Un algorithme d'optimisation plus avancé appelé '*FastICA*' réalise la maximisation via la méthode de la descente du gradient.

4. Mise à jour itérative :

La matrice de démélange *W* est mise à jour de manière itérative pour améliorer l'estimation des signaux sources S. Ensuite il faut répéter jusqu'à avoir une convergence :

a) Calculer la matrice d'activation telle que :

$$A = W.\tilde{X} \quad \text{(V.39)}$$

b) Calculer la fonction de non-linéarité g(S) appliquée à chaque élément de A.
c) Mettre à jour la matrice de démélange *W* selon la règle de mise à jour de l'algorithme utilisé.
d) Normaliser chaque ligne de *W* pour maintenir l'invariance à l'échelle.

5. Arrêt de la mise à jour :

Le processus de mise à jour itérative se poursuit jusqu'à ce qu'un critère de convergence soit atteint, tel qu'un nombre maximum d'itérations ou un faible changement dans la matrice de démélange.

On a implémenté la fonction *'ica_decompose ()'* pour extraire et sélectionner les composantes indépendantes d'un mélange EEG en utilisant différentes techniques d'ICA disponibles dans les modules *'FastICA'* [Zarzoso, V. e al., 2006], *'Picard-python'* [Ablin, P. et

al., 2018a ; 2018b] et *'ica.infomax'* [Langlois, D. et al., 2010]. Elle renvoie les signaux sources estimés et les matrices de démélange correspondantes. Le paramètre *ortho* permet de contrôler l'orthogonalité des composantes. La fonction est flexible et peut être utilisée pour la sélection et l'extraction de composantes indépendantes dans les données EEG.

D'autres fonctions sont fournies dans le module '*featuresPlotter.py*' permettent de tracer différentes visualisations pour analyser les caractéristiques des signaux EEG. Voici un résumé de chaque fonction :

La fonction *'plot_logvar ()'* permet de tracer un graphique des variances logarithmiques moyennes des données EEG pour chaque classe. Les classes sont représentées par des barres de couleur différente. La fonction *'sampEn_dists_plot ()'* permet d'afficher un histogramme des distances de similarité pour l'entropie d'échantillonnage (*Sample Entropy*) des données EEG. Chaque histogramme est étiqueté avec le paramètre *m* correspondant. Il y a également la fonction *'features_scatter_plot ()'* qui trace un graphique de dispersion des caractéristiques des signaux EEG pour deux classes différentes. Les caractéristiques sont représentées par des points colorés selon leur classe, et un plan de décision peut être tracé si spécifiée dans les paramètres. En outre, il y a la fonction *'pca_selector_plot ()'* qui sert à visualiser des graphiques de dispersion pour les trois premières composantes principales des données EEG. Chaque graphique peut être en 2D ou 3D en fonction de la dimensionnalité des données.

V.5.3. Classification par Apprentissage automatique

Le package "*postproc*" comprend trois autres modules, à savoir '*classification.py*', '*classifierTools.py*' et '*classificationPlotter.py*', qui sont des composants intégraux traitant des techniques d'apprentissage automatique. Ces modules contribuent à la fonctionnalité globale du package en fournissant des outils essentiels pour la mise en œuvre, l'évaluation et la visualisation des algorithmes de classification. Discutons de ces modules et de leur rôle dans le contexte des interfaces cerveau-ordinateur.

Le module '*classification.py*' contient l'implémentation de divers algorithmes de classification, y compris le *K-Nearest Neighbors* KNN [Kayikcioglu, T., & Aydemir, O., 2010], la *Régression Logistique* [Isa N. M. et al, 2019], le *Naive Bayes* [Bassani, T., & Nievola, J. C., 2009], l'*Analyse Discriminante Linéaire* [Joadder, M. A. et al., 2019], l'*Arbre de Décision* [Ishfaque, A. et al., 2013], la *Forêt Aléatoire* [Masud, U. et al., 2017] et la *Machine à Vecteurs de Support* SVM [Hortal, E. et al., 2013]. Ce module est le composant central responsable de l'entraînement et de l'utilisation de ces algorithmes sur les signaux EEG ou d'autres ensembles

de données nécessitant une classification. Le module s'appuie sur des bibliothèques d'apprentissage automatique populaires telles que Scikit-learn dans le but accéder aux algorithmes bien optimisés et à leurs fonctionnalités correspondantes. En plus de fournir des implémentations d'algorithmes, le module incorpore également de la flexibilité en permettant aux utilisateurs de personnaliser les algorithmes à l'aide de paramètres optionnels. Cette flexibilité permet aux utilisateurs d'affiner les algorithmes en fonction de leurs besoins spécifiques, par exemple en ajustant les hyperparamètres ou en appliquant de plusieurs variantes du même algorithme. En encapsulant les algorithmes de classification dans un module dédié *'postproc'*, nous favorisons la réutilisation et la maintenance du code. Cette conception modulaire facilite l'intégration avec d'autres composants de notre boite à outils *Step2bci*. En outre, elle permet des extensions futures, telles que l'ajout de nouveaux algorithmes de classification ou la modification d'algorithmes existants, sans affecter la structure globale de package.

Dans cette partie, nous présentons une vue d'ensemble des algorithmes de classification incorporés dans le module '*classification.py*' et qui sont couramment utilisés dans l'analyse des signaux EEG. Nous discutons des principes sous-jacents des algorithmes et examinons leurs caractéristiques de performance.

V.5.3.1. Classifieur K-plus proches voisins (KNN)

La fonction *'knn_classifier ()'* met en œuvre l'algorithme de classification KNN (K-Nearest Neighbors). Elle prend en entrée les données d'apprentissage *X_train* et *y_train* et les données de test *X_true*. La fonction utilise la classe *KNeighborsClassifier* de la bibliothèque scikit-learn pour entraîner un modèle KNN sur les données d'apprentissage, puis prédit les étiquettes pour les données de test. L'algorithme KNN attribue une étiquette de classe à un échantillon en tenant compte des classes de ses voisins les plus proches dans l'espace des caractéristiques [Cunningham, P., & Delany, S. J., 2021]. La fonction renvoie les prédictions (y_pred) et le classificateur KNN entraîné. L'algorithme KNN est une technique non paramétrique qui classe les données en fonction de leur proximité avec des exemples étiquetés. Elle est développée pour la première fois par Evelyn Fix et Joseph Hodges en 1951 [Fix, E., & Hodges, J., 1951], puis plus tard étendue par Thomas Cover [Cover, T., & Hart, P., 1967].

De la figure V.11A, on peut observer que l'utilisation d'un seul voisin produit un plan de décision qui suit de près les données d'entraînement. À mesure que le nombre de voisins augmente, le plan de décision devient plus lisse. Cette régularité indique un modèle plus simple. En d'autres termes, un faible nombre de voisins conduit à une grosse complexité du modèle

(comme l'illustre la figure V.11A), tandis qu'un plus grand nombre de voisins correspond à une faible complexité du modèle (comme montré par la figure V.11C). Dans un cas extrême où tous les points de données de l'ensemble d'entraînement sont considérés comme voisins, chaque point de test aurait exactement les mêmes voisins (c'est-à-dire tous les points d'entraînement). Par conséquent, toutes les prédictions seraient identiques, basées sur la classe la plus fréquente observée dans l'ensemble d'entraînement.

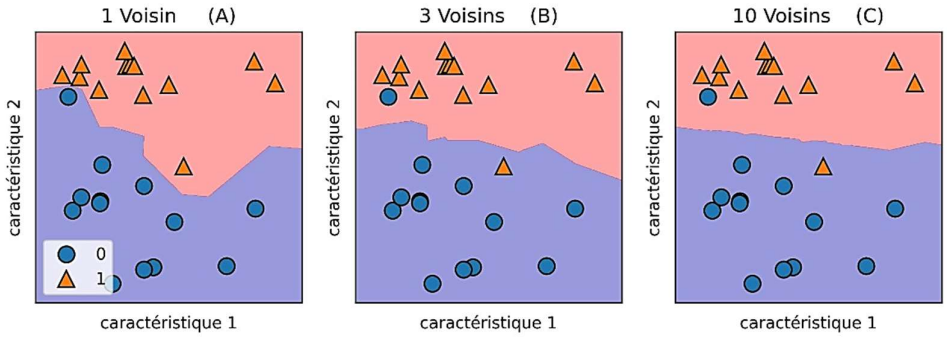


Figure V.11. Formation du plan de décision de KNN pour différentes valeurs de k : 1, 2 et 4 voisins.

V.5.3.2. Classifieur d'analyse discriminante linéaire (LDA)

La fonction *'lda_classifier ()'* implémente l'algorithme LDA en utilisant la classe *LinearDiscriminantAnalysis* du package scikit-learn. La technique originale a été mise au point en 1936 par Ronald A. Fisher [Cai, Y., & Wang, D., 2020]. L'analyse discriminante linéaire LDA est un algorithme d'apprentissage automatique supervisé couramment utilisé pour la réduction de la dimensionnalité et les tâches de classification. Il vise à trouver une combinaison linéaire de caractéristiques qui maximise la séparation entre les différentes classes tout en minimisant la dispersion à l'intérieur de la classe elle-même. Comme illustré par la figure V.12, dans l'algorithme LDA, les données d'entrée sont projetées dans un espace de dimension inférieure à l'aide d'une transformation linéaire [Ghojogh, B., & Crowley, M., 2019]. Cette transformation maximise le rapport entre la dispersion entre les classes et la dispersion à l'intérieur des classes, maximisant ainsi la séparabilité des classes. Il suppose que les données suivent une distribution gaussienne et que les classes ont des matrices de covariance égales.

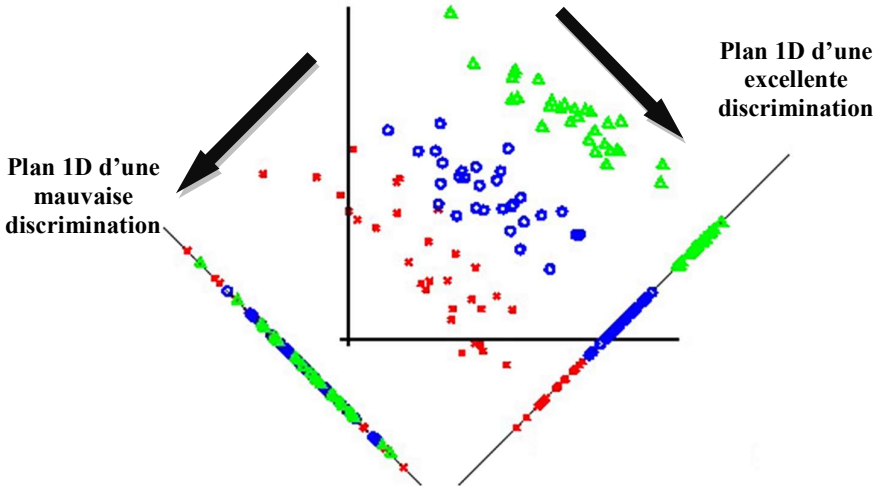


Figure V.12. Données de caractéristiques à 3 classes.

Étant donné un ensemble de données avec des caractéristiques d'entrée X et des étiquettes de classe correspondantes y, où : $X = \{x_1, x_2, \ldots, x_n\}$ et $y = \{y_1, y_2, \ldots, y_n\}$. Alors, LDA vise à trouver un vecteur de projection W qui maximise la séparation entre les classes dans un espace de dimension inférieure.

1. Calculer les moyennes spécifiques aux classes :

- Calculer le vecteur moyen m_c, pour chaque classe c, qui représente la moyenne des vecteurs de caractéristiques des échantillons appartenant à la classe c, donc :

$$m_c = \frac{1}{N_c} \sum_{i=1}^{N_c} x_i \quad \text{(V.40)}$$

Où : x_i est le vecteur de caractéristiques d'un échantillon de la classe c, et N_c est le nombre d'échantillons de la classe c. Le vecteur moyen global m est calculé en prenant la moyenne de tous les vecteurs de caractéristiques tel que :

$$m = \frac{1}{N} \sum_{i=1}^{N} m_c \quad \text{(V.41)}$$

Où : N est le nombre total d'échantillons.

2. Calculer les matrices de dispersion :

La matrice de dispersion intra-classe S_w mesure la dispersion des échantillons au sein de chaque classe :

$$S_w = \sum_{c=1}^{nc} \sum_{i=1}^{N} (x_i - m_c).(x_i - m_c)^T \quad \text{(V.42)}$$

Où : x_i est le vecteur de caractéristiques et m_c le vecteur moyen de la classe c.

La matrice de dispersion entre classes S_b mesure la dispersion entre les différentes classes :

$$S_b = \sum_{c=1}^{nc} N_c(m_c - m).(m_c - m)^T \quad \text{(V.43)}$$

Où : N_c est le nombre d'échantillons dans la classe c.

3. Calculer les valeurs propres et les vecteurs propres :

- Calculer la matrice S_p telle que :

$$S_p = S_w^{-1}.S_b \quad \text{(V.44)}$$

Où : S_w^{-1} est l'inverse de S_w.

- Calculer les vecteurs propres w et leurs valeurs propres correspondantes λ de la matrice S_p.

Les valeurs propres représentent l'importance de chaque vecteur propre dans la capture de la séparabilité des classes.

4. Sélectionnez les vecteurs discriminants :

- Trier les valeurs propres par ordre décroissant et sélectionner les vecteurs propres correspondants.
- Choisissez les k vecteurs propres correspondant aux k plus grandes valeurs propres pour former une matrice de transformation W.
- La matrice de transformation W est utilisée pour projeter les données X originales à haute dimension sur un sous-espace de dimension inférieure.

5. Classer les nouveaux échantillons :

- Projeter les nouveaux échantillons sur le sous-espace à l'aide de la matrice de transformation *W*.
- Attribuer l'étiquette de classe en fonction du plus proche voisin ou à l'aide d'un classificateur tel qu'une règle basée sur un seuil.

Dans notre implémentation, la fonction *'lda_classifier ()'* prend en entrée les données d'entraînement (*X_train*) et les étiquettes correspondantes (*y_train*), ainsi que les données de test (*X_true*). Le paramètre de rétrécissement (*shrinkage*) permet de contrôler le niveau de régularisation appliqué à la matrice de covariance. La régularisation aide à éviter que la matrice de covariance ne soit singulière ou instable. Si une valeur spécifique est fournie pour le rétrécissement, celle-ci est utilisée directement. Sinon, la valeur du rétrécissement reste inchangée. La classe *LinearDiscriminantAnalysis* est instanciée avec le paramètre de rétrécissement spécifié, ainsi qu'avec d'autres arguments (**args*) et arguments clés (***kwargs*) éventuellement passés à la fonction. Ensuite, la méthode **fit** de l'objet lda_ est appelée pour entraîner le modèle LDA en utilisant les données d'entraînement et les étiquettes cibles. Une fois le modèle entraîné, la méthode *predict* est utilisée pour effectuer des prédictions sur les données non vues.

V.5.3.3. Classifieur Machine à vecteurs de support (SVM)

La fonction *'svm_classifier ()'* est une implémentation de l'algorithme Machine à Vecteurs de Support (SVM) pour la classification. Elle prend en entrée les données d'apprentissage (*X_train*) et les étiquettes correspondantes (*y_train*), ainsi que les données de test (*X_true*). La fonction utilise la classe SVC de la bibliothèque scikit-learn pour créer un classificateur SVM. Cette fonction utilise l'algorithme SVM pour entraîner un classifieur sur les données et les étiquettes fournies. Notre fonction personnalisée offre la possibilité de spécifier un ensemble d'hyperparamètres pour le classifieur SVM. Cela nous permet d'ajuster finement le modèle en fonction de nos besoins spécifiques. De plus, la fonction intègre la possibilité d'utiliser *GridSearchCV*, une technique permettant d'effectuer une recherche exhaustive sur une grille de paramètres spécifiée, afin de déterminer automatiquement la meilleure combinaison d'hyperparamètres. En utilisant *GridSearchCV*, la fonction explore de manière systématique l'espace des hyperparamètres et identifie l'ensemble optimal de paramètres qui maximisent les performances du modèle.

Le SVM est un algorithme puissant utilisé en apprentissage automatique pour la classification et la régression, notamment dans le domaine de l'analyse des signaux EEG pour les applications BCI. Il vise à trouver un hyperplan optimal qui sépare les données EEG en fonction des différentes classes de signaux cérébraux, permettant ainsi de détecter et de classifier les motifs cérébraux [Riyadi, M. A. et al., 2019].

Dans le contexte des applications BCI, le SVM est particulièrement utile pour analyser les signaux EEG et extraire des caractéristiques discriminantes. En trouvant l'hyperplan optimal, l'algorithme SVM cherche à maximiser la séparation entre les différentes classes, ce qui facilite la classification précise des modèles d'activité cérébrale tels que les intentions motrices ou les réponses émotionnelles.

L'apprentissage de l'algorithme SVM implique la résolution d'un problème d'optimisation convexe, où l'objectif est de minimiser l'erreur de classification tout en maximisant la marge entre les vecteurs de support (la figure V.13). Différentes techniques d'optimisation, telles que la programmation quadratique et les méthodes à noyau, sont utilisées pour résoudre ce problème et obtenir les paramètres optimaux du SVM.

- **SVM linéaire**

L'approche SVM linéaire vise à trouver un hyperplan linéaire optimal qui sépare les données de différentes classes dans l'espace des caractéristiques d'origine [Cervantes, J. et al., 2020].

Formulation du problème d'optimisation :

- Trouvez un hyperplan représenté par un vecteur de poids **w** et un terme de biais *b*.
- Pour chaque instance d'entraînement (x_i, y_i), la contrainte de marge est définie comme suit :

$$y_i(\boldsymbol{w}^T x_i + b) \geq 1, \forall i \qquad \text{(V.45)}$$

Cette contrainte garantit que chaque instance est correctement classée avec une marge d'au moins *1*.

Avec : x_i est l'instance d'entraînement, y_i est l'étiquette de classe (-1 ou 1), **w** est le vecteur de poids et *b* est le terme de biais.

Détermination de la marge :

- La marge géométrique est définie comme la distance entre l'hyperplan de décision et les instances les plus proches de chaque classe, elle est exprimée par :

$$d = \frac{2}{\|\boldsymbol{w}\|} \tag{V.46}$$

Où : ||w|| est la norme euclidienne du vecteur de poids.

Formulation de l'objectif d'optimisation :

L'objectif d'optimisation est de minimiser la norme du vecteur de poids, tout en respectant la contrainte de marge (équation V.45), tel que :

$$\min_{w} \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{V.47}$$

Résolution du problème d'optimisation :

- Le problème d'optimisation peut être résolu à l'aide de techniques telles que la programmation quadratique ou l'optimisation convexe.
- L'objectif est de trouver les valeurs optimales du vecteur de poids **w** et du terme de biais *b* qui minimisent la fonction objective.

Prise de décisions :

- Après avoir obtenu les valeurs optimales de **w** et *b*, les prédictions pour de nouvelles instances peuvent être faites en utilisant la règle de décision suivante :

$$y_{pred} = sign(\boldsymbol{w^T}.x + b) \tag{V.48}$$

Où : *x* représente une nouvelle instance et y_{pred} est l'étiquette de classe prédite (-1 ou 1) en fonction du signe de l'expression.

- **SVM non-linéaire (à noyau)**

Cette approche permet de traiter des données non linéairement séparables en utilisant des fonctions de noyau [Ghosh, S. et al., 2019].

Sélection du noyau : cela dépend du problème et des caractéristiques des données. Les noyaux couramment utilisés sont les suivants :

Noyau linéaire :

$$F(x_1, x_2) = x_1^T . x_2 \quad \text{(V.49)}$$

Noyau polynomial :

$$F(x_1, x_2) = \left(\gamma((x_1^T . x_2) + r)\right)^d \quad \text{(V.50)}$$

Noyau gaussien RBF (*Radial Basis Function*) :

$$F(x_1, x_2) = e^{(-\gamma \|x_1 - x_2\|^2)} \quad \text{(V.51)}$$

Noyau sigmoïde :

$$F(x_1, x_2) = tanh(\gamma(x_1^T . x_2) + r) \quad \text{(V.52)}$$

Où : x_1, x_2 représentent deux instances d'entraînement.

γ, r et d sont des hyperparamètres spécifiques à chaque noyau.

Construction de la matrice du noyau et transformation des données :

- Calculer la matrice du noyau K en appliquant la fonction de noyau à toutes les paires d'instances d'entraînement pour mapper les données dans un espace des caractéristiques de dimension supérieure ; Cela permet de rendre les données linéairement séparables dans cet espace. La matrice du noyau est une matrice symétrique $N \times N$ où N est le nombre d'instances d'entraînement.

Formulation du problème d'optimisation :

- L'objectif est de trouver les valeurs optimales des multiplicateurs de Lagrange α qui maximisent la marge tout en satisfaisant les contraintes suivantes :

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad \text{(V.53)}$$

où : $0 \leq \alpha_i \leq C, \forall i$

C représente le paramètre de régularisation du SVM à noyau. Ce paramètre contrôle le compromis entre la maximisation de la marge et la minimisation des erreurs d'entraînement. Il joue un rôle essentiel dans la régularisation du modèle et la gestion du compromis (*trade-off*) entre la complexité du modèle et la capacité à bien généraliser sur de nouvelles données.

Le problème d'optimisation peut être formulé comme suit :

$$max_{\alpha}\left(\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j F(x_i, x_j)\right) \quad \text{(V.54)}$$

La résolution de ce problème d'optimisation complexe peut être réalisée à l'aide de méthodes numériques efficaces, telles que la programmation quadratique ou des algorithmes d'optimisation spécifiques aux contraintes non linéaires. Une fois les valeurs optimales des multiplicateurs de Lagrange α obtenues, les vecteurs de support peuvent être identifiés comme les instances d'entraînement correspondantes avec des multiplicateurs non nuls (α_i>0).

Calcul des vecteurs de support :

- Identifiez les vecteurs de support en sélectionnant les instances d'entraînement correspondantes avec des multiplicateurs non nuls (α_i>0).

Prédiction :

Pour prédire la classe d'une nouvelle instance x, on utilise la fonction de décision :

$$f(x) = \sum_{i=1}^{N}\alpha_i y_i\, F(x_i, x) + b \quad \text{(V.55)}$$

Où : x_i sont les vecteurs de support, y_i sont les étiquettes correspondantes, α_i sont les multiplicateurs de Lagrange associés et b est le biais (*intercepte*).

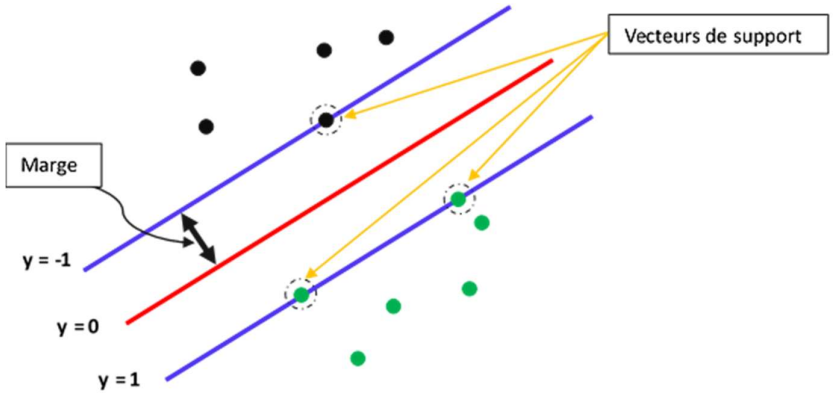


Figure V.13. Vecteurs de Support et la Marge SVM.

Le module '*classification.py*' propose une variété de fonctions pour différents algorithmes de classification. Par exemple, la fonction *'dt_classifier ()'* utilise des arbres de décision pour créer un modèle arborescent basé sur les caractéristiques des données EEG. Elle divise judicieusement les données en utilisant la meilleure caractéristique et le meilleur seuil afin

d'optimiser efficacement le gain d'informations. Les nouvelles instances sont prédites de manière précise en parcourant méthodiquement l'arbre. La fonction *'rf_classifier ()'*, quant à elle, exploite des classificateurs de forêt aléatoire, formant ainsi un ensemble d'arbres de décision. Elle crée plusieurs arbres de décision en utilisant des échantillons de données bootstrapés et des sous-ensembles de caractéristiques aléatoires. En agrégeant les prédictions des arbres individuels, elle parvient à prédire la classe de manière fiable. La fonction *'logreg_classifier ()'* met en œuvre la régression logistique, modélisant la relation entre les caractéristiques et les logarithmes de la variable cible. En utilisant l'estimation du maximum de vraisemblance, elle parvient à déterminer les paramètres optimaux et à prédire de la probabilité de chaque classe pour les nouvelles instances. La fonction *'gnb_classifier ()'* applique l'algorithme Gaussian Naive Bayes, en supposant que les caractéristiques suivent une distribution gaussienne. Elle estime la moyenne et l'écart-type de chaque classe pour calculer la probabilité d'appartenance d'une nouvelle instance à chaque classe. Enfin, la fonction *'mnb_classifier ()'* exploite avec l'algorithme Multinomial Naive Bayes, en se basant sur l'indépendance des caractéristiques. Elle calcule la probabilité de chaque classe en fonction des caractéristiques, ce qui lui permet de prédire la classe la plus probable pour les nouvelles instances. Chacune de ces fonctions offre une approche spécifique de classification, prenant soigneusement en compte les caractéristiques des données EEG pour obtenir des prédictions précises. Une discussion détaillée de leur fonctionnement interne serait évidemment exhaustive, mais des informations supplémentaires pourront être explorées selon les besoins.

Les détails complets sur le fonctionnement interne de ces fonctions sont fournis en annexe 2 de la thèse, offrant ainsi une analyse exhaustive de chaque algorithme de classification.

V.5.4. Classification par apprentissage profond

Cette partie explore l'application des techniques d'apprentissage profond dans le contexte des BCIs.

La boîte à outils S*tep2bci* intègre le package "*deepproc*", qui est spécifiquement conçu pour faciliter la classification par apprentissage profond des applications d'interface cerveau-ordinateur (BCI) basées sur l'EEG. On a conçu trois modules clés : '*deepDevice.py*', '*neuroDataset.py*' et '*deepBCI.py*', qui forment ensemble le package "*deepproc*" travaillant en coopération pour créer, entraîner et évaluer un modèle de réseau de neurones profond basé sur la bibliothèque PyTorch.

- Pourquoi PyTorch ?

PyTorch est une puissante bibliothèque d'apprentissage profond qui présente de nombreux avantages par rapport à d'autres bibliothèques populaires telles que TensorFlow, Caffe et Theano [PyTorch, 2022]. L'un des principaux atouts de PyTorch réside dans son utilisation de graphes de calcul dynamiques. Contrairement au graphique statique de TensorFlow, PyTorch permet une construction et une modification de modèle plus flexibles. Cela facilite le débogage et le prototypage rapide, car les développeurs peuvent utiliser les fonctionnalités de Python pour inspecter et manipuler les tenseurs pendant l'exécution du modèle [Karpathy, 2018]. La nature dynamique du graphe de PyTorch le rend également adapté à la gestion des entrées de longueurs variables, des réseaux neuronaux récurrents et d'autres architectures dynamiques [Paszke et al., 2019].

La bibliothèque PyTorch propose une large gamme de modules et de fonctions pour construire des réseaux de neurones, mettre en œuvre différentes couches, des fonctions de coût (ou de perte) et des algorithmes d'optimisation. En termes de performance, PyTorch prend en charge la technologie sous-jacente CUDA (*Compute Unified Device Architecture*) qui permet à PyTorch d'utiliser les GPU-Nvidia (*Graphical Processing Unit-Nvidia*) pour le calcul distribué. Cette capacité le rend adapté à l'entraînement de modèles sur des ensembles de données à grande échelle (*Big-Data*) [Paszke et al., 2019]. De plus, PyTorch permet une transition transparente entre les calculs CPU et GPU, permettant aux développeurs d'exploiter toute la puissance des GPU-Nvidia modernes [Min, S. W. et al., 2021]. En utilisant les fonctionnalités CUDA de PyTorch, on peut déplacer des tenseurs et des modèles vers le GPU, effectuer des calculs sur le GPU et récupérer les résultats sur le CPU [PyTorch, 2022].

Alors que TensorFlow bénéficie d'une adoption généralisée dans l'industrie et d'une large base d'utilisateurs, PyTorch a gagné en popularité auprès des chercheurs et des universitaires en raison de sa flexibilité et de sa facilité d'utilisation. La communauté PyTorch est très active, fournissant une documentation complète, des didacticiels, etc. Bien que Caffe soit connu pour sa vitesse d'inférence, notamment dans les applications de vision par ordinateur, PyTorch se distingue par sa flexibilité et son support pour la recherche et le développement de nouveaux modèles [Jia et al., 2014]. Theano, quant à lui, a connu un déclin de son utilisation et de son développement, PyTorch offrant une alternative plus moderne et activement développée [Bergstra et al., 2010]. Chaque bibliothèque a ses propres forces et faiblesses. Cependant, la

flexibilité, la syntaxe intuitive et la communauté active de PyTorch en font un choix populaire pour les chercheurs et les développeurs de la communauté de l'Apprentissage profond.

- Les réseaux de neurones artificiels et Apprentissage profond

Les réseaux de neurones artificiels ANN (*Artificial Neural Network*) sont des modèles d'apprentissage automatique inspirés de la structure et de la fonction des réseaux de neurones biologiques [Zhang, X., & Yao, L., 2021]. Les ANN sont constitués de neurones artificiels interconnectés et organisés en couches. Les neurones calculent une somme pondérée des entrées, telle que :

$$Z = W.X + b \tag{V.56}$$

Pour introduire la non-linéarité et permettre au réseau de neurones de modéliser des relations complexes, une fonction d'activation est appliquée à la somme pondérée z. Les fonctions d'activation les plus courantes sont :

La fonction d'unité linéaire rectifiée ReLU (*Rectified Linear Unit*) : est préférée dans les couches cachées parce qu'elle permet de résoudre le problème du gradient qui s'évanouit (*vanishing gradient*) et permet au réseau d'apprendre plus efficacement. Toutefois, il convient de faire attention au problème de la "ReLU mourante", où les neurones peuvent devenir inactifs pendant l'apprentissage et ne jamais se rétablir. Diverses modifications, telles que le Leaky ReLU et le ReLU paramétrique, ont été proposées pour résoudre ce problème. La fonction ReLU est définie comme :

$$f(x) = \max(0, x) \tag{V.57}$$

Où : x représente l'entrée de la fonction d'activation.

La fonction sigmoïde : est souvent utilisée dans la couche de sortie des réseaux de neurones, en particulier pour les problèmes de classification binaire. Elle fait correspondre l'entrée à une valeur comprise entre 0 et 1, en fournissant une transformation lisse et non linéaire. La formule mathématique de la fonction sigmoïde est la suivante :

$$f(x) = \frac{1}{1+e^{-x}} \tag{V.58}$$

La fonction Softmax : est couramment utilisée dans la couche de sortie des réseaux de neurones pour les problèmes de classification multi-classes. Elle convertit un vecteur de valeurs

réelles arbitraires en une distribution de probabilité sur plusieurs classes, la fonction *softmax* est définie comme suit :

$$f(x) = \frac{e^x}{\sum_{i=1}^{M} e^{x_i}} \quad \text{(V.59)}$$

Où : x représente l'entrée de la fonction d'activation pour une classe quelconque, et M le nombre total de classes.

D'autre part, le perceptron multicouche MLP (*Multi-Layer Perceptron*) est un modèle de réseau de neurones à propagation en avant (*feedforward*) qui se compose de plusieurs couches de neurones, dont une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Chaque neurone du réseau est relié aux neurones des couches adjacentes par des connexions pondérées [Al Bataineh, A. et al., 2022].

Pendant la phase d'apprentissage, le MLP apprend les poids et les biais optimaux associés aux connexions entre les neurones de chaque couche [Singh, J., & Banerjee, R., 2019]. La propagation vers l'avant dans un MLP commence par introduire les données d'entrée dans la couche d'entrée du réseau ANN. Ensuite, pour chaque couche suivante, la somme pondérée des entrées de chaque neurone est calculée en utilisant les poids associés aux connexions (la figure V.14).

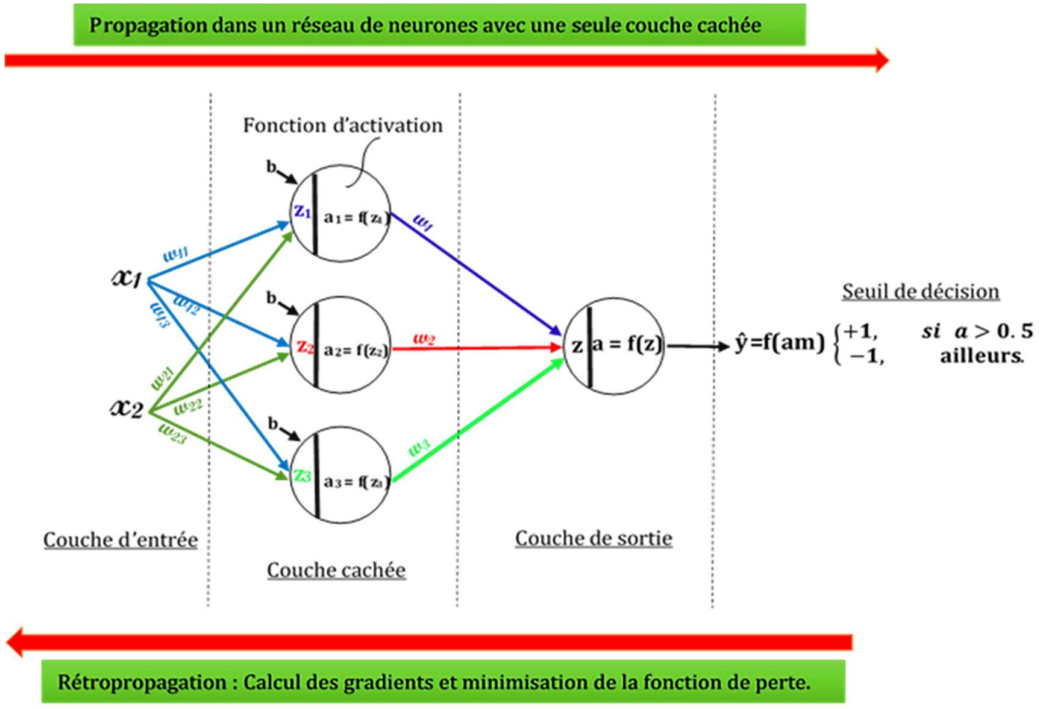


Figure V.14. Cœur d'un réseau MLP pour l'apprentissage profond

Mathématiquement, cela peut être représenté comme suit :

$$z_k = b_k + \sum_{i=1}^{N} w_{ik} \, . \, x_i \tag{V.60}$$

Où : N est le nombre des entrées, z_k représente la somme pondérée pour le $k^{ème}$ neurone dans la couche cachée. Le terme b_k représente le biais associé au $k^{ème}$ neurone, et w_{ik} représente le poids associé à la connexion entre l'entrée x_i et le $k^{ème}$ neurone.

Chaque élément z_k devient une entrée pour la fonction d'activation *f(.)*, prenons par exemple la fonction sigmoïde σ(.) donc la sortie a_k de la fonction d'activation est exprimée par :

$$a_k = \sigma(z_k) = \frac{1}{1+e^{-z_k}} \tag{V.61}$$

L'idée derrière la fonction d'activation est que l'activation d'une unité est similaire à l'activation d'un neurone lorsque ce dernier reçoit une entrée suffisamment forte. Lorsque la somme pondérée des entrées atteint un seuil donné, l'unité est activée, ce qui permet la propagation de l'information à travers le réseau. En d'autres termes, la fonction d'activation introduit la non-linéarité dans le réseau de neurones, ce qui lui permet d'apprendre et de représenter des relations complexes dans les données.

La propagation vers l'avant *'feedforward'* permet de faire progresser les entrées à travers le réseau en utilisant les poids et les biais appris, tout en introduisant des non-linéarités grâce aux fonctions d'activation. Cela active ensuite les neurones pour générer des prédictions ou des sorties.

Mais, Comment ajuster les paramètres du modèle ?

La rétropropagation joue un rôle crucial dans l'optimisation des poids et des paramètres d'un réseau de neurones en minimisant la fonction de perte (ou de coût). La première étape consiste à définir une fonction de perte appropriée qui quantifie l'écart entre les sorties prédites et les cibles réelles. Désignons la fonction de perte par L, qui peut varier en fonction du problème à résoudre [Sun, R., 2019]. Par exemple, l'erreur quadratique moyenne MSE (*Mean Squared Error*) pour la régression ou la perte de l'entropie croisée (*Cross-Entropy*) pour la classification.

Soit y la cible réelle d'un échantillon et ŷ la probabilité prédite pour la classe correspondante. La fonction de perte de Cross-Entropy peut s'exprimer de la manière suivante :

$$L = -\frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{C} y_{ij} . log\left(\hat{y}_{ij}\right) \tag{V.62}$$

Où : M représente le nombre total d'échantillons d'entraînement, C est le nombre de classes, y_{ij} représente la cible réelle (0 ou 1) pour le $i^{ème}$ échantillon dans la $j^{ème}$ classe, et $\hat{y}_{ij}$ représente la probabilité prédite pour le $i^{ème}$ échantillon dans la $j^{ème}$ classe.

La fonction de perte de cross-entropy L pénalise les grandes erreurs dans les probabilités prédites. Elle encourage les probabilités prédites à être proches de 1 pour la classe réelle et proches de 0 pour les autres classes.

L'objectif de la rétropropagation est de minimiser la fonction de perte en ajustant les poids et les paramètres du réseau MLP. Cet objectif est généralement atteint à l'aide d'algorithmes d'optimisation tels que la descente du gradient GD (*Gradient Descente*) [Zhang, J., 2019]. Cette dernière est un algorithme d'optimisation itératif qui met à jour les poids et les paramètres dans la direction opposée du gradient de la fonction de perte. L'équation de mise à jour pour chaque poids w_{ij} dans le réseau peut être écrite comme suit :

$$w_{ik} \leftarrow (w_{ik} - \rho \frac{\partial L}{\partial w_{ik}}) \tag{V.63}$$

Où : ρ est le taux d'apprentissage, qui contrôle la taille du pas des mises à jour.

Lors de la rétropropagation, les gradients de la fonction de perte par rapport aux poids (paramètres) sont calculés à l'aide de la règle de la chaîne (*chain rule*). Le gradient de la fonction de perte par rapport à la somme pondérée des entrées pour le neurone de sortie z peut être calculé comme :

$$\frac{\partial L}{\partial w_{ik}} = \frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{C} \left(\frac{y_{ij}}{\hat{y}_{ij}} \frac{\partial \hat{y}_{ij}}{\partial w_{ik}} \right) \tag{V.64}$$

Où :

L : est la fonction de perte, M est le nombre d'échantillons de données.

C : est le nombre de classes (ou le nombre de neurones de sortie).

y_{ij} représente la cible réelle $i^{ème}$ échantillon de données pour le $j^{ème}$ neurone de sortie.

$\hat{y}_{ij}$ représente la sortie prédite du $i^{ème}$ échantillon de données pour le $j^{ème}$ neurone de sortie.

Après avoir calculé les gradients, les poids et les paramètres du réseau MLP sont mis à jour en utilisant l'équation de mise à jour (équation V.63). Les poids sont ajustés dans la direction opposée de leurs gradients afin de minimiser la fonction de perte. En itérant les étapes

de rétropropagation et de mise à jour des poids, le réseau apprend progressivement à optimiser ses poids et ses paramètres, améliorant ainsi les performances et permettant des prédictions plus précises. Il est à noter que l'algorithme de rétropropagation repose sur la règle de la chaîne des dérivées pour calculer efficacement les gradients à travers le réseau, facilitant ainsi l'optimisation des paramètres du réseau et permettant aux modèles d'apprentissage profond d'apprendre des motifs et des représentations complexes dans les données.

PyTorch effectue les passes avant (propagation) et arrière (rétropropagation) dans les réseaux de neurones en utilisant un graphe de calcul et la différenciation automatique. Lors de la propagation, les données d'entrée sont diffusées à travers chaque couche du réseau, générant des résultats intermédiaires stockés dans des tenseurs. Ces tenseurs sont des structures de données similaires à des tableaux multidimensionnels. Le graphe de calcul enregistre les opérations effectuées sur les tenseurs, ce qui permet à PyTorch de calculer automatiquement les gradients lors de la rétropropagation. Les gradients sont calculés à partir de la couche de sortie en utilisant la règle de la chaîne et sont ensuite propagés en sens inverse à travers le graphe. Cette approche fait intervenir un optimisateur pour la mise à jour des paramètres du modèle. En utilisant la différenciation automatique et les opérations sur les tenseurs, PyTorch simplifie le processus d'entraînement des modèles d'apprentissage profond et permet des calculs mathématiques efficaces sur des données multidimensionnelles.

V.5.4.1. Module de gestion des dispositifs (*deepDevice*)

Le module '*deepDevice.py*' fournit une classe appelée *'DeviceSwitcher'* qui permet de basculer facilement entre les périphériques utilisés par les tenseurs PyTorch. Lors de l'initialisation, la classe prend en argument facultatif *'force_device'*, qui spécifie le périphérique souhaité comme étant soit *'Cuda (GPU)*' ou '*CPU*'. Si le paramètre *'force_device'* n'est pas fourni, le module vérifie automatiquement la disponibilité d'un périphérique CUDA et l'utilise s'il est présent ; sinon, il utilise par défaut le CPU.

La classe *'DeviceSwitcher'* possède une méthode appelée *'to_device'*, qui accepte un élément et un argument facultatif *'non_blocking'*. Si l'élément est un tuple ou une liste, la méthode applique de manière récursive le changement de périphérique à chaque élément du conteneur. Si l'élément est un tenseur PyTorch, la méthode vérifie si le paramètre *'force_device'* est spécifié et déplace le tenseur vers ce périphérique en utilisant la méthode 'to', sinon le tenseur est renvoyé tel quel.

V.5.4.2. Module de personnalisation et de transformation des données EEG (*neuroDataset*)

Le module '*neuroDataset.py*' est composé de plusieurs classes liées à la gestion des ensembles de données EEG.

La classe *'LoadSplitDataset'* est responsable du chargement et de la division de l'ensemble de données. Elle prend différents paramètres tels que le nom du fichier, les caractéristiques, les cibles et les options de lecture des données. Si un nom de fichier est fourni, elle lit le fichier CSV à l'aide de Pandas et extrait les caractéristiques et les cibles des données. Alternativement, si les caractéristiques et les cibles sont directement fournies, elles sont utilisées. La classe inclut également une méthode appelée '*split_data ()*' pour diviser l'ensemble de données en ensembles d'entraînement et de test en utilisant la fonction '*train_test_split ()*' de scikit-learn.

La classe *'CustomDataset'* est une implémentation d'ensemble de données PyTorch. Elle prend '*inputData'*, qui peut être soit un tuple de caractéristiques et de cibles, soit un dictionnaire (qui n'est pas entièrement implémenté dans le code fourni). La classe initialise le nombre de classes et de caractéristiques en fonction des données d'entrée. Les caractéristiques et les cibles sont converties en tenseurs PyTorch, et la classe fournit les méthodes nécessaires 'len' et 'getitem' pour l'indexation de l'ensemble de données.

La classe *'DatasetTransforms'* définit des transformateurs de données à l'aide du module *torchvision.transforms*. Elle inclut une méthode '*call*' qui applique les transformations définies aux données d'entrée, telles que leur conversion en tenseur et leur normalisation.

V.5.4.3. Module d'apprentissage profond (*deepBCI*)

Le module '*deepBCI.py*' dans le package "*deepproc*" comprend des classes et des méthodes pour l'entraînement, l'évaluation et la prédiction avec des modèles d'apprentissage profond dans les applications de BCI basées sur l'EEG.

La classe *'NetPytorch'* est un modèle de réseau de neurones PyTorch. Elle définit un réseau entièrement connecté avec des tailles de couches cachées variables. La taille d'entrée, la taille de sortie et les tailles des couches cachées sont spécifiées lors de l'initialisation. La méthode *'forward'* effectue la propagation-avant à travers le réseau.

La classe *'TorchTrainer'* est responsable de l'entraînement du modèle de réseau neuronal. Elle prend en entrée le modèle, les données d'entraînement et divers paramètres d'entraînement tels que la taille de lot, le taux d'apprentissage et l'optimiseur. Pendant l'entraînement, elle itère

sur les données, effectue des passages avant et arrière, met à jour les paramètres du modèle et suit les pertes d'entraînement et d'évaluation. Elle prend également en charge l'arrêt précoce basé sur la perte de validation.

La classe *'TorchEvaluator'* est utilisée pour évaluer le modèle entraîné. Elle prend en entrée le modèle, les données d'évaluation et un critère facultatif (fonction de perte). Elle calcule la perte d'évaluation, la précision et affiche la précision sur l'ensemble d'évaluation.

La classe *'TorchPredictor'* est utilisée pour effectuer des prédictions avec le modèle entraîné. Elle prend en entrée le modèle, la taille de l'entrée et le dispositif. Elle fournit des méthodes pour prédire des échantillons individuels ou des lots d'échantillons. Elle effectue le prétraitement nécessaire, fait passer les échantillons à travers le modèle et renvoie les étiquettes prédites.

V.6. Conclusion

Le chapitre présentait en détail la boite-à-outils complète ***Step2BCI*** comprenant plusieurs packages et modules essentiels pour l'analyse et la classification EEG dans les applications BCI. Le package *'preproc'* offre des fonctionnalités complètes pour le prétraitement des données EEG. Il comprend des modules tels que *'dataFilter.py'* qui permet de filtrer les données en utilisant des filtres de Butterworth d'ordre n et Kaiser pour les filtres passe-bas, passe-haut, passe-bande et coupe-bande. Le module *'dataEditor.py'* propose des fonctionnalités pour la suppression de la ligne de base afin de réduire les dérives basales, ainsi que des fonctions de sous-échantillonnage. Le module '*DataScaler.py*' fournit plusieurs fonctions de normalisation et de mise à l'échelle des données, telles que *'robust_scaler ()'*, *'max_abs_scaler ()'*, *'min_max_scaler ()'*, *'std_scaler ()'* et *'data_Normalizer ()'*. Le module *'dataIO.py'* permet de charger et de transformer des ensembles de données provenant de différentes sources, y compris des formats de fichiers tels que *.mat, *.txt et *.csv, facilitant ainsi la gestion des données de diverses origines, comme le Graz-lab (*Laboratoire de Graz*), le Berlin Institute, *Open-BCI*, etc. Ensuite, le module *'waveletAnalyzer.py'* propose des fonctionnalités pour l'analyse et la décomposition par ondelettes. Il comprend deux fonctions : *'cwt_analysis ()'* et *'wavedec_dwt_analysis ()'*, qui permettent d'effectuer des analyses et des décompositions par ondelettes sur les données EEG.

Le package *'postproc'* contient plusieurs modules dédiés au post-traitement des données EEG. Le module *'featuresExtractor.py'* offre une variété de fonctions pour extraire des caractéristiques à partir des données EEG, telles que *'wavelet_feat ()'* pour l'extraction de

caractéristiques par ondelettes, *'csp_feat ()'* pour l'extraction de caractéristiques CSP, *'logvar_feat ()'* pour l'extraction de la variance logarithmique, *'subBP_feat ()'* pour l'extraction de caractéristiques de puissance spectrale, et *'samEntropy_feat ()'* pour l'extraction d'entropie spectrale d'amplitude modifiée. Le module *'featuresSelector.py'* se concentre sur la sélection des caractéristiques pertinentes à partir de l'ensemble extrait, avec des fonctions telles que *'SFS_select ()'* pour la sélection séquentielle vers l'avant, *'PCA_select ()'* pour la sélection basée sur l'analyse en composantes principales, et *'ICA_decomposer ()'* pour la décomposition en composantes indépendantes ICA.

De plus, le package "*postproc*" comprend les modules '*classification.py*', '*classifierTools.py*' et '*ClassificationPlotter.py*' pour la classification et l'évaluation des modèles. Le module *'classification.py'* fournit des implémentations d'algorithmes de classification tels que la régression logistique, les machines à vecteurs de support (SVM), les arbres de décision, les forêts aléatoires, l'analyse discriminante linéaire (LDA), etc. Le module '*classifierTools.py*' propose des outils supplémentaires pour évaluer les performances des modèles de classification, effectuer une validation croisée et gérer les déséquilibres de classes ou d'autres tâches de prétraitement spécifiques à la classification. Le module '*ClassificationPlotter.py*' se concentre sur la visualisation et la présentation des résultats du processus de classification.

Enfin, le package "*deeppproc*" est conçu pour faciliter la classification basée sur l'apprentissage profond (Deep Learning) dans les applications BCI. Il contient plusieurs modules tels que '*neuroDevice.py*', '*neuroDataset.py*' et '*deepBCI.py*'. Le module '*neuroDevice.py*' permet de gérer facilement le commutateur de périphérique utilisé par les tenseurs PyTorch, permettant ainsi de basculer facilement entre l'utilisation du CPU et du GPU. Le module *'neuroDataset.py'* fournit des classes et des méthodes pour le chargement, la division et la transformation des ensembles de données EEG, facilitant leur intégration dans les modèles d'apprentissage profond. Enfin, le module '*deepBCI.py*' offre des classes et des méthodes pour l'entraînement, l'évaluation et la prédiction avec des modèles d'apprentissage profond dans les applications BCI. Il comprend plusieurs classes telles que : *'NetPytorch'* pour la modélisation d'un réseau de neurones à couches cachées, *'TorchTrainer'* pour l'entraînement du modèle, '*TorchEvaluator'* pour l'évaluation du modèle, et *'TorchPredictor'* pour la prédiction avec le modèle entraîné.

En résumé, la boite-à-outils *Step2bci* offre une plateforme complète pour la recherche et le développement d'applications BCI. Il permet aux étudiants, chercheurs et aux praticiens d'effectuer des analyses avancées sur les données EEG, d'extraire des caractéristiques pertinentes, de sélectionner les meilleures caractéristiques, de classifier les données avec une variété d'algorithmes, et de mettre en œuvre des modèles d'apprentissage profond pour la classification des signaux EEG. Ce travail de recherche et de développement fournit une base solide pour l'avancement des applications BCI et contribue à l'amélioration des performances et de la fiabilité des systèmes BCI basés sur l'EEG.

BIBLIOGRAPHIE DU CHAPITRE V

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Ablin, P., Cardoso, J. F., & Gramfort, A. (2018). Faster independent component analysis by preconditioning with Hessian approximations. IEEE Transactions on Signal Processing, 66(15), 4040-4049.

Ablin, P., Cardoso, J. F., & Gramfort, A. (2018, April). Faster ICA under orthogonal constraint. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4464-4468). IEEE.

Al Bataineh, A., Kaur, D., & Jalali, S. M. J. (2022). Multi-layer perceptron training optimization using nature inspired computing. *IEEE Access*, *10*, 36963-36977.

Aldea, R., & Tărniceriu, D. (2013, October). Estimating the Hurst exponent in motor imagery-based brain computer interface. In 2013 7th Conference on Speech Technology and Human-Computer Dialogue (SpeD) (pp. 1-6). IEEE.

Aljalal, M., Djemal, R., AlSharabi, K., & Ibrahim, S. (2018, April). Feature extraction of EEG based motor imagery using CSP based on logarithmic band power, entropy and energy. In *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1-6). IEEE.

Ameera, A., Saidatul, A., & Ibrahim, Z. (2019, June). Analysis of EEG spectrum bands using power spectral density for pleasure and displeasure state. In IOP conference series: Materials science and engineering (Vol. 557, No. 1, p. 012030). IOP Publishing.

Bahri, Z., Abdulaal, S., & Buallay, M. (2014, January). Sub-band-power-based efficient brain computer interface for wheelchair control. In 2014 World Symposium on Computer Applications & Research (WSCAR) (pp. 1-7). IEEE.

Bajaj, N. (2020). Wavelets for EEG Analysis. Wavelet Theory.

Bashashati, A., Ward, R. K., Birch, G. E., Hashemi, M. R., & Khalilzadeh, M. A. (2003, September). Fractal dimension-based EEG biofeedback system. In *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)* (Vol. 3, pp. 2220-2223). IEEE.

Bassani, T., & Nievola, J. C. (2009). Brain-computer interface using wavelet transformation and naive bayes classifier. In *Brain Inspired Cognitive Systems 2008* (pp. 147-165). New York, NY : Springer New York.

Bekkali, S., Youssef, G. J., Donaldson, P. H., Hyde, C., Do, M., He, J. L., ... & Enticott, P. G. (2021). Is there a relationship between EEG and sTMS neurophysiological markers of the putative human mirror neuron system?. Journal of Neuroscience Research, 99(12), 3238-3249.

BeOpen PythonLabs Team. (2000, 16th October). Python 2.0. Retrieved from official website Python : https://www.python.org/download/releases/2.0/

Beyene, T., & Sharma, A. (2021). Python programming for the next generation of scientists. arXiv preprint arXiv:2104.03253.

Birkelund, Y., Hanssen, A., & Powers, E. J. (2003). Multitaper estimators of polyspectra. *Signal Processing*, *83*(3), 545-559.

Bissyandé, T. F., Thung, F., Lo, D., Jiang, L., & Réveillere, L. (2013, July). Popularity, interoperability, and impact of programming languages in 100,000 open source projects. In *2013 IEEE 37th annual computer software and applications conference* (pp. 303-312). IEEE.

Cai, Y., & Wang, D. (2020). Deep generative LDA. *arXiv preprint arXiv:2010.16138*.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, *408*, 189-215.

Chen, Y., Zhang, C., & Wu, X. (2019, July). To Assess the Influence of Artifacts on Motor Imagery Based BCI. In 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP) (pp. 925-929). IEEE.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.

Croce, P., Conti, E., Merletti, R., & Valenza, G. (2018). Adaptive down-sampling of multichannel EEG signals based on information theory. IEEE transactions on neural systems and rehabilitation engineering, 26(5), 980-989.

Cunningham, P., & Delany, S. J. (2021). k-Nearest neighbour classifiers-A Tutorial. *ACM computing surveys (CSUR)*, *54*(6), 1-25.

Datar, A., Jain, A., & Sharma, P. C. (2010). Design of Kaiser window based optimized prototype filter for cosine modulated filter banks. *Signal Processing*, *90*(5), 1742-1749.

Donoghue, T., Haller, M., Peterson, E. J., & Varma, S. (2020). A principled approach to EEG downsampled from high to low sampling rates. Journal of neural engineering, 17(4), 046015.

Elsayed, N., Zaghloul, Z. S., & Bayoumi, M. (2017). Brain computer interface: EEG signal preprocessing issues and solutions. *Int. J. Comput. Appl*, *169*(3), 12-16.

Fix, E., & Hodges, J. (1951). Discriminatory analysis: Nonparametric discrimination - consistency properties. USAF School of Aviation Medicine, Randolph Field, Texas.

Ghojogh, B., & Crowley, M. (2019). Linear and quadratic discriminant analysis: Tutorial. *arXiv preprint arXiv:1906.02590*.

Ghosh, S., Dasgupta, A., & Swetapadma, A. (2019, February). A study on support vector machine based linear and non-linear pattern classification. In *2019 International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 24-28). IEEE.

Gottemukkula, V., & Derakhshani, R. (2011, April). Classification-guided feature selection for NIRS-based BCI. In *2011 5th International IEEE/EMBS Conference on Neural Engineering* (pp. 72-75). IEEE.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., ... & Hämäläinen, M. (2013). MNE software for processing MEG and EEG data. NeuroImage, 86, 446-460.

Hortal, E., Úbeda, A., Iáñez, E., Planelles, D., & Azorin, J. M. (2013, November). Online classification of two mental tasks using a SVM-based BCI system. In *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)* (pp. 1307-1310). IEEE.

Hu, Y., Gao, J., & Zhang, Z. (2018). Normalization of EEG-based brain connectivity metrics: A review. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26(10), 1933-1942.

Hunter, J. D. (2007). Matplotlib : A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95.

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, *13*(4-5), 411-430.

Isa, N. M., Amir, A., Ilyas, M. Z., & Razalli, M. S. (2019). Motor imagery classification in Brain computer interface (BCI) based on EEG signal by using machine learning technique. Bulletin of Electrical Engineering and Informatics, 8(1), 269-275.

Ishfaque, A., Awan, A. J., Rashid, N., & Iqbal, J. (2013, December). Evaluation of ANN, LDA and decision trees for EEG based brain computer interface. In 2013 IEEE 9th International Conference on Emerging Technologies (ICET) (pp. 1-6). IEEE.

Joadder, M. A., Siuly, S., Kabir, E., Wang, H., & Zhang, Y. (2019). A new design of mental state classification for subject independent BCI systems. *IRBM*, *40*(5), 297-305.

Kayikcioglu, T., & Aydemir, O. (2010). A polynomial fitting and k-NN based approach for improving classification of motor imagery BCI data. *Pattern Recognition Letters*, *31*(11), 1207-1215.

Kothe, C. A., & Makeig, S. (2013). BCILAB: A platform for brain-computer interface development. Journal of Neural Engineering, 10(5), 056014.

Lahane, P., Jagtap, J., Inamdar, A., Karne, N., & Dev, R. (2019, February). A review of recent trends in EEG based Brain-Computer Interface. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)* (pp. 1-6). IEEE.

Langlois, D., Chartier, S., & Gosselin, D. (2010). An introduction to independent component analysis: InfoMax and FastICA algorithms. *Tutorials in Quantitative Methods for Psychology*, *6*(1), 31-38.

Lee, G., Gommers, R., Waselewski, F., Wohlfahrt, K., & O'Leary, A. (2019). PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*, *4*(36), 1237.

Louden, K. C., & Lambert, K. A. (2011). Programming languages: Principles and practice. Cengage Learning.

Loughlin, P., & Cohen, L. (2010). Wavelets: A comparison with the spectrogram and other methods for time-frequency analysis. *The Journal of the Acoustical Society of America*, *127*(3), 1936-1936.

Luck, S. J., & Kappenman, E. S. (Eds.). (2012). The Oxford Handbook of Event-Related Potential Components (Oxford Library of Psychology) [Illustrated Edition]. Oxford University Press.

Masud, U., Baig, M. I., Akram, F., & Kim, T. S. (2017, November). A P300 brain computer interface based intelligent home control system using a random forest classifier. In 2017 IEEE symposium series on computational intelligence (SSCI) (pp. 1-5). IEEE.

McKinney, W. (2010). Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51-56).

Min, S. W., Wu, K., Huang, S., Hidayetoğlu, M., Xiong, J., Ebrahimi, E., ... & Hwu, W. M. (2021). Pytorch-direct: Enabling gpu centric data access for very large graph neural network training with irregular accesses. arXiv preprint arXiv:2101.07956.

Murugappan, M., Rizon, M., Nagarajan, R., Yaacob, S., Hazry, D., & Zunaidi, I. (2008). Time-frequency analysis of EEG signals for human emotion detection. In *4th Kuala Lumpur International Conference on Biomedical Engineering 2008: BIOMED 2008 25–28 June 2008 Kuala Lumpur, Malaysia* (pp. 262-265). Springer Berlin Heidelberg.

Naik, G. R. (Ed.). (2012). *Independent component analysis for audio and biosignal applications*. BoD–Books on Demand.

Oostenveld, R., Fries, P., Maris, E., & Schoffelen, J. M. (2011). FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. Computational Intelligence and Neuroscience, 2011, 1-9.

Oppenheim, A. V., Schafer, R. W., & Buck, J. R. (2009). Discrete-Time Signal Processing (3rd edition). Pearson.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32, 8024-8035.

Pavlov, A. N., Runnova, A. E., Maksimenko, V. A., Pavlova, O. N., Grishina, D. S., & Hramov, A. E. (2018). Detrended fluctuation analysis of EEG patterns associated with real and imaginary arm movements. *Physica A: Statistical Mechanics and its Applications*, *509*, 777-782.

Pouryazdian, S., Rezaei Tabar, M., Erfanian, A., & Shamsollahi, M. B. (2018). A comprehensive study on artifact reduction in EEG signals for BCIs. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26(5), 973-979.

Python Software Foundation. (n.d.). Python programming language, 2021, Retrieved from https://www.python.org.

PyTorch. PyTorch documentation.2022, Retrieved from https://pytorch.org /docs/stable/index.html

Radüntz, T., Scouten, J., Hochmuth, O., & Meffert, B. (2015). EEG artifact elimination by extraction of ICA-component features using image processing algorithms. *Journal of neuroscience methods*, *243*, 84-93.

Rayatnia, A., & Khanbabaie, R. (2019). Common spatial patterns feature extraction and support vector machine classification for motor imagery with the secondbrain. *International Journal of Engineering*, *32*(9), 1284-1289.

Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., & Lécuyer, A. (2010). OpenViBE: An open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. Presence: Teleoperators and Virtual Environments, 19(1), 35-53.

Reuderink, B., & Poel, M. (2008). Robustness of the common spatial patterns algorithm in the BCI-pipeline. *University of Twente, Tech. Rep*.

Riyadi, M. A., Prakoso, T., Whaillan, F. O., Wahono, M. D., & Hidayatno, A. (2019, October). Classification of EEG-based brain waves for motor imagery using support vector machine. In *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)* (pp. 422-425). IEEE.

Sanei, S., & Chambers, J. A. (2013). *EEG signal processing*. John Wiley & Sons.

Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., & Wolpaw, J. R. (2004). BCI2000: A general-purpose brain-computer interface (BCI) system. IEEE Transactions on Biomedical Engineering, 51(6), 1034-1043.

Schlögl, A., & Brunner, C. (2008). BioSig: A free and open source software library for BCI research. Computer, 41(10), 44-50.

Schomer, D. L., & Da Silva, F. L. (2012). *Niedermeyer's electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins.

Schreiner, S., Azevedo Costa, R., Freese, M., Kreilinger, A., Müller-Putz, G., & Scherer, R. (2008). BCPy2000: A new modular software package for real-time data acquisition and signal processing in BCI experiments. Clinical Neurophysiology, 119(11), e33-e34.

Selesnick, I. W., & Burrus, C. S. (1998). Generalized digital Butterworth filter design. *IEEE Transactions on signal processing*, *46*(6), 1688-1694.

Shovon, T. H., Al Nazi, Z., Dash, S., & Hossain, M. F. (2019, September). Classification of motor imagery EEG signals with multi-input convolutional neural network by augmenting STFT. In *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)* (pp. 398-403). IEEE.

Singh, J., & Banerjee, R. (2019, March). A study on single and multi-layer perceptron neural network. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 35-40). IEEE.

Stoica, P., & Moses, R. L. (2005). *Spectral analysis of signals* (Vol. 452, pp. 25-26). Upper Saddle River, NJ: Pearson Prentice Hall.

Sun, R. (2019). Optimization for deep learning: theory and algorithms. *arXiv preprint arXiv:1912.08957*.

Tariq, M., Trivailo, P. M., & Simic, M. (2018). Motor imagery based EEG features visualization for BCI applications. *Procedia computer science*, *126*, 1936-1944.

Thomson, D. J. (1982). Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, *70*(9), 1055-1096.

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. Computing in Science & Engineering, 13(2), 22-30.

Van Rossum, G., Warsaw, B., & Coghlan, N. (2001). PEP 1 -- PEP Purpose and Guidelines. Python.org. Retrieved from https://www.python.org/dev/peps/pep-0001/

Van Rossum, G. (n.d.). Biography. Retrieved from official website Python, 2021 ,https://www.python.org /doc/essays/cp4e/

Pyff: Venthur, B., Hostettler, R., & Schweitzer, G. (2010). Pyff: A Pythonic Framework for Feedback Applications and Stimulus Presentation in Neuroscience. Frontiers in Neuroinformatics, 4, 26.

Mushu: Venthur, B., Dähne, S., Hohne, J., & Blankertz, B. (2012). Mushu: Multimodal Shuffling of Time Series Data. Journal of Machine Learning Research, 13, 461-464.

Wyrm: Venthur, B., Dähne, S., & Blankertz, B. (2014). Wyrm: A Brain-Computer Interface Toolbox in Python. Neuroinformatics, 12(2), 209-225.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & Van Der Walt, S. J. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, 17(3), 261-272.

Wang, L., Xu, G., Yang, S., Wang, J., Guo, M., & Yan, W. (2012, June). Motor imagery BCI research based on sample entropy and SVM. In *2012 Sixth International Conference on Electromagnetic Field Problems and Applications* (pp. 1-4). IEEE.

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*(2), 70-73.

Yu, X., Chum, P., & Sim, K. B. (2014). Analysis the effect of PCA for feature reduction in non-stationary EEG based motor imagery of BCI system. *Optik*, *125*(3), 1498-1502.

Zarzoso, V., Comon, P., & Kallel, M. (2006, September). How fast is FastICA?. In 2006 14th European Signal Processing Conference (pp. 1-5). IEEE.

Zhang, J. (2019). Gradient descent based optimization algorithms for deep learning models training. *arXiv preprint arXiv:1903.03614*.

Zhang, X., & Yao, L. (2021). Deep Learning for EEG-based Brain-Computer Interfaces: Representations, Algorithms, and Applications.

Zied, T., Waniek, N., Fedjaev, J., Ghaboosi, N., Rychly, L., Widderich, C., ... & Cheng, G. (2018). Gumpy: A Python Toolbox Suitable for Hybrid Brain-Computer Interfaces. Journal of Neural Engineering, 15(6), 065003.

Zito, T., Wilbert, N., Wiskott, L., & Berkes, P. (2009). Modular toolkit for Data Processing (MDP): A Python data processing framework. *Frontiers in neuroinformatics*, 8.

CHAPITRE VI

PROTOCOLE EXPERIMENTAL HORS-LIGNE D'UNE BCI VIA LA BOITE A OUTILS STEP2BCI

CONTENU

VI.1. Introduction
VI.2. Protocoles réels de la BCI hors ligne utilisant step2bci
VI.3. Protocole d'imagination motrice avec une rétroaction asynchrone
  VI.3.1. Présentation des données (Dataset 1)
  VI.3.2. Préparation des données
  VI.3.3. Création des ensembles d'entraînement et de test
  VI.3.4. Calcul des caractéristiques spectrales par le périodogramme de Welch
  VI.3.5. Application de la technique CSP (Common Spatial Pattern)
  VI.3.6. Classification
    VI.3.6.1. Sélection et réduction de la dimensionnalité des caractéristiques
    VI.3.6.2. Sélection d'estimateurs
    VI.3.6.3. Création des modèles d'apprentissage automatique types à l'aide de la boite à outils Step2bci
    VI.3.6.4. Création du modèle d'apprentissage profond à l'aide de la boîte à outils Step2bci
  VI.3.7. Interprétation et discussion des résultats
VI.4. Protocole d'imagination motrice avec et sans rétroaction (feedback)
  VI.4.1. Présentation des données
  VI.4.2. Préparation des données
  VI.4.3. Création des ensembles d'apprentissage et de test
  VI.4.4. Extraction des caractéristiques par décomposition en ondelettes
  VI.4.5. Réduction de dimensionalité
    VI.4.5.1. Approche PCA
    VI.4.5.2. Approche SFS
  VI.4.6. Classification par apprentissage profond à l'aide de la boîte à outils Step2bci
  VI.4.7. Interprétation et discussion des résultats
VI.5. Conclusion
Bibliographie du Chapitre VI

VI.1. Introduction

Ce chapitre explore l'application de notre boîte à outils, *Step2bci,* dans l'analyse et la classification des données EEG. L'objectif principal de ce chapitre est de démontrer l'efficacité de *Step2bci* dans les protocoles BCI hors ligne. En intégrant les packages développés dans le chapitre précédent, nous visons à présenter l'exécution fluide des différentes étapes de l'analyse des données EEG, comprenant la préparation des données, le prétraitement, le post-traitement et la classification par apprentissage profond. Une attention particulière est accordée à la présentation de l'implémentation du code. Nous fournirons des explications détaillées sur des aspects essentiels tels que l'importation de modules, l'instanciation de classes, l'appel de fonctions, la manipulation des ensembles de données et la construction de pipelines complets pour l'extraction de caractéristiques et leur classification. Ces protocoles pratiques permettront aux lecteurs d'utiliser efficacement *step2bci* dans leurs propres projets d'analyse EEG et de réalisation des BCIs.

Tout au long de ce chapitre, nous présenterons des scénarios concrets, en examinant différents algorithmes et en analysant les résultats correspondants. En s'impliquant dans ce processus, les utilisateurs acquerront une compréhension approfondie de l'analyse EEG et obtiendront des connaissances pour orienter leurs propres recherches ou applications. Grâce à une combinaison de techniques avancées et d'une mise en œuvre pratique, nous visons à découvrir des schémas significatifs au sein des signaux cérébraux et à contribuer au développement des interfaces cerveau-ordinateur.

VI.2. Protocoles réels de la BCI hors ligne utilisant step2bci

Ce chapitre explore l'application de notre boîte à outils, step2bci, dans l'analyse et la classification des données EEG. L'objectif principal est de démontrer son efficacité dans les protocoles BCI hors ligne en présentant des cas d'utilisation réels et en analysant leurs résultats. Dans ce chapitre, nous utiliserons des ensembles de données tiers fournis par des institutions réputées dans le domaine des interfaces cerveau-ordinateur.

Les ensembles de données utilisés à cette fin sont disponible au site web https://www.bbci.de/competition/iv :

- Ensemble de données Competition IV-Dataset 1 [Blankertz, B. et al., 2007] créé et fourni par le groupe BCI de Berlin : Institut de technologie de Berlin (*Laboratoire d'apprentissage automatique*) [Berlinbci, 2022] et Fraunhofer FIRST (*Groupe d'analyse*

intelligente des données) [Fraunhofer, 2022], pour la classification de l'EEG continu sans structure d'essai.

- Ensemble de données Competition IV-Dataset 2b [Leeb, R. et al., 2008] créé et fourni par l'institut de découverte des connaissances (Laboratoire des interfaces cerveau-ordinateur), Université de technologie de Graz, Autriche [Graz, 2022], pour la classification des signaux EEG, imagination motrice (main gauche, main droite, pieds, langue).

VI.3. Protocole d'imagination motrice avec une rétroaction asynchrone

VI.3.1. Présentation des données (Dataset 1)

Cet ensemble de données a été enregistré à partir de sujets en bonne santé. Pendant toute la session, des imaginations motrices ont été réalisées sans rétroaction (feedback). Pour chaque sujet, deux classes cibles ont été sélectionnées parmi les trois classes suivantes : main gauche, main droite et pied (côté choisi par le sujet ; éventuellement les deux pieds) [Blankertz, B. et al., 2007].

Données de calibration : Lors des deux premières séries, des flèches indiquant la direction gauche, droite ou bas étaient présentées en tant que repères visuels sur un écran d'ordinateur. Les repères étaient affichés pendant une période de 4 secondes, pendant laquelle le sujet devait effectuer la tâche motrice indiquée. Ces périodes étaient entrecoupées de 2 secondes d'écran vide et de 2 secondes avec une croix de fixation affichée au centre de l'écran. La croix de fixation était superposée aux repères, c'est-à-dire qu'elle était affichée pendant 6 secondes. Ces ensembles de données sont fournis avec des informations de marquage complètes.

Données d'évaluation : Ensuite, quatre séries ont suivi, qui sont utilisées pour l'évaluation. Dans cet ensemble de données, les tâches d'imagination motrice étaient indiquées par des stimuli acoustiques doux (mots : gauche, droite et pied) pendant des périodes de durée variable allant de 1,5 à 8 secondes. La fin de la période d'imagination motrice était indiquée par le mot "*stop*". Les périodes intermittentes avaient également une durée variable de 1,5 à 8 secondes. À noter que dans les données d'évaluation, il n'y a pas nécessairement autant d'essais pour chaque condition.

Format des données : Les signaux EEG continus de 59 canaux sont fournis, ainsi que, pour les données de calibration, des marqueurs indiquant les moments de présentation des repères et les classes cibles correspondantes. Les données sont fournies au format Matlab (**.mat**) et contiennent les variables suivantes :

- *cnt* : les signaux EEG continus, de format (temps, canaux) et de type INT16. Pour les convertir en valeurs en microvolts (µV), on doit utiliser cnt = 0.1 * double(cnt).
- *mrk* : structure d'informations sur les repères cibles.
- *pos* : vecteur de positions des repères dans les signaux EEG, exprimées en échantillons unitaires.
- *y* : vecteur de classes cibles (-1 pour la première classe ou 1 pour la deuxième classe).
- *nfo* : structure fournissant des métadonnées (informations supplémentaires).

Il s'agit d'un enregistrement de grande envergure : 59 électrodes ont été utilisées, réparties sur l'ensemble du cuir chevelu. Le sujet a reçu un signal et a ensuite imaginé soit le mouvement de la main droite, soit le mouvement de ses pieds (ou la main gauche). Comme le montre l'homoncule à la figure I.10 (Chapitre I), le mouvement du pied est contrôlé au centre du cortex moteur (ce qui rend difficile la distinction entre le pied gauche et le pied droit), tandis que le mouvement de la main est contrôlé plus latéralement.

VI.3.2. Préparation des données

Dans cette section, nous aborderons les étapes de la préparation des données et l'extraction des caractéristiques à l'aide de la boîte à outils *Step2bci*. Nous importerons des modules et des classes spécifiques de la boîte à outils pour faciliter ces opérations. L'exemple de code suivant suit notre convention d'importation d'un module à partir d'un package :

```
import numpy as np
import preproc.IOfiles as iof
import preproc.trialsProcessor as trproc
import preproc.dataPlotter as plotter
import preproc.dataAnalyzer as analysis
import preproc.dataFilter as filt
```

Nous chargeons les données à l'aide de la fonction *'read_matlab_file ()'* du module **iof**. Nous spécifions le chemin d'accès au fichier MATLAB contenant les données et l'utilisateur peut définir le paramètre de verbosité sur *'True'* ou *'False'*. Les données chargées sont affectées aux variables (*X, trials, targets, attributes*). Ces variables représentent respectivement les signaux EEG, les informations sur les essais, les étiquettes des cibles et les informations sur les attributs.

```
X, trials, targets, attributes = iof.read_matlab_file (full_path='../datasets/berlindata/comp4_calib_ds1d
/BCICIV_calib_ds1d.mat', verbosity=False, dtype='EEG')
```

Nous extrayons les informations pertinentes des attributs. Le nombre d'échantillons et de canaux dans les données est stocké dans les variables *nsamples* et *nchannels*. La fréquence

d'échantillonnage est assignée à *sfreq*. Les classes et les noms des canaux sont stockés dans les variables *classes* et *channels*, respectivement. Les codes cibles uniques sont stockés dans *target_codes*. Le nombre de classes est déterminé par *nclasses*, et les noms des classes sont assignés aux variables *cl1* et *cl2*.

```
nsamples, nchannels = X.shape
sfreq = attributes['sfreq']
classes = attributes['Classes']
channels = attributes['Channels']
target_codes = np.unique(targets)
nclasses = len(classes)
cl1, cl2 = classes[0], classes[1]
```

Maintenant, nous extrayons les données des essais en utilisant la fonction *trials_extract* du module *trproc*. Nous passons les données d'entrée *X*, les informations sur les essais *trials*, les étiquettes cibles *targets* et spécifions une fenêtre temporelle d'intérêt [0,5 – 2,5] secondes après l'apparition du marqueur. Le paramètre de version est défini sur (*version = berlin)* pour assurer la compatibilité avec l'ensemble de données du groupe Berlin BCI. Les données des essais extraites sont stockées dans la variable *trials_data*.

```
trials_data = trproc.trials_extract(inputData=X, trials=trials, targets=targets, window=[0.5, 2.5], version='berlin',
                sfreq=sfreq)
```

Après avoir extrait les données des essais, nous affichons la forme des données pour chaque classe. La première ligne de code montre le format des données pour la classe "*gauche*" (- 1), indiquant le nombre d'essais, de points de données et de canaux. De même, la deuxième ligne de code présente le format des données pour la classe "*droite*" (1). Ces informations nous donnent un aperçu de la structure des données pour chaque classe et nous permettent de mieux comprendre leur organisation.

```
print('Format des données pour la classe (left) -1 (n_trials, data, channel)', trials_data[target_codes[0]].shape)
print('Format des données pour la classe (right) 1 (n_trials, data, channel)', trials_data[target_codes[1]].shape)
```

```
Output :
Format des données pour la classe (left) -1 (n_trials, data, channel) (100, 200, 59)
Format des données pour la classe (right) 1 (n_trials, data, channel) (100, 200, 59)
```

VI.3.3. Création des ensembles d'entraînement et de test

Dans cette étape, nous procédons à la séparation des données d'essai en ensembles d'entraînement et de test. Le dictionnaire '*trials_data*' contient deux clés (-1, 1), représentant les classes respectives.

Donc, nous concaténons les valeurs correspondantes à ces clés pour créer l'ensemble de données global '*data*', par la suite, nous générons les étiquettes cibles correspondantes dans la variable *targets* comme indiqué par ce code :

```
data = np.concatenate((trials_data[target_codes[0]], trials_data[target_codes[1]]), axis=0)
targets = np.concatenate((np.full((trials_data[target_codes[0]].shape[0],), target_codes[0]), \
        np.full((trials_data[target_codes[1]].shape[0],), target_codes[1])), axis=0)
```

Ensuite, en utilisant la fonction *'train_test_clf ()'* du module '*classifierTools.py*', les données et les étiquettes sont divisées en ensembles d'apprentissage et de test. En spécifiant le paramètre *test_size* de 0,25 (soit 25% des données), la variable *'splitted_data'* contient les ensembles divisés prêts à être utilisés dans les étapes ultérieures de l'analyse ou de la classification.

```
splitted_data = tools.train_test_clf(features=data, targets=targets, test_size=0.25, random_state=1234)
```

La fonction *'train_true_composer ()'* est utilisée pour regrouper les données en vue de l'extraction des caractéristiques à l'aide de la fonction *'csp_feat ()'*. Elle effectue une transformation des données pour les mettre dans un format approprié en les organisant dans des dictionnaires distincts en fonction des codes cibles fournis.

```
train, true = trproc.train_true_composer(splitted_data, classes=classes, target_codes=target_codes)
```

VI.3.4. Calcul des caractéristiques spectrales par le périodogramme de Welch

Lorsqu'une personne observe ou imagine un mouvement spécifique, comme l'imagination de la main gauche ou droite, on observe une diminution de l'activité μ dans la région correspondante du cortex moteur [Tariq, M. et al., 2020]. L'activité μ, qui est une composante spécifique des rythmes cérébraux, est souvent étudiée en relation avec le mouvement et l'imagination motrice. Elle se manifeste principalement dans la bande de fréquence des ondes alpha (8- 12 Hz) et est généralement observée dans les régions du cortex moteur telles que C3, Cz et C4 [Sazgar M. et al., 2019]. L'activité μ est étroitement liée à la planification et à l'exécution du mouvement, ainsi qu'à la perception des actions réalisées. Elle est donc d'un grand intérêt dans le domaine des BCIs pour la détection et la classification des intentions motrices.

Étant donné que la caractéristique recherchée, à savoir la diminution de l'activité μ dans la bande de fréquence (7, 12 Hz), est une caractéristique de fréquence, il est nécessaire d'utiliser des méthodes d'analyse telles que le périodogramme traditionnel, la méthode de Welch ou le multitaper pour localiser la présence des potentiels de synchronisation/désynchronisation (ERS/ERD) [Pfurtscheller, G., 2001] liés à l'évènement (Exécution/Intention motrice).

```
freqs_f, welch_f = analysis.welch_periodogram_analysis(inputData=train[classes[0]], sliding_win=200, sfreq=sfreq,
t_ax=1)
freqs_r, welch_r = analysis.welch_periodogram_analysis(inputData=train[classes[1]], sliding_win=200, sfreq=sfreq,
t_ax=1)
```

La fonction *'welch_periodogram_analysis ()'* est appliquée aux données des essais de chaque classe spécifique (-1 : *left* et 1 *right*). Les paramètres tels que la taille de la fenêtre glissante *sliding_win*, la fréquence d'échantillonnage *sfreq* et l'axe temporel *t_ax* sont passés à la fonction pour effectuer le calcul. Le résultat de cette opération est renvoyé sous la forme de deux tableaux de type *numpy.ndarray* [Python, 2022] : *freqs_f* et *welch_f* pour la classe (left, -1), et *freqs_r* et *welch_r* pour la classe (right", 1). Ces tableaux contiennent les fréquences et les valeurs du périodogramme de Welch correspondant à chaque fréquence pour chaque classe.

Le rythme μ est principalement observé dans la région centrale du cerveau, qui comprend le cortex moteur. Les électrodes C3, Cz et C4 sont généralement placées dans cette région. En traçant le périodogramme de Welch de la bande μ sur ces sites d'électrodes (figure VI.1), nous pouvons visualiser l'estimation de la densité spectrale de puissance. Cela permet d'obtenir des informations sur les schémas d'activation corticale et l'implication du cortex moteur dans les tâches liées au mouvement.

```
trials_psd = trproc.data_class_composer(welch_f, welch_r, classes=classes)
plotter.trials_psd_plot(trials_psd, freqs_f, [channels.index(ch) for ch in ['C3', 'Cz', 'C4']], chan_lab=['C3', 'Cz', 'C4'],\
            maxy=800, xlabel='Fréquence ($Hz$)', ylabel='PSD de Welch', colors=['green', 'red'])
```

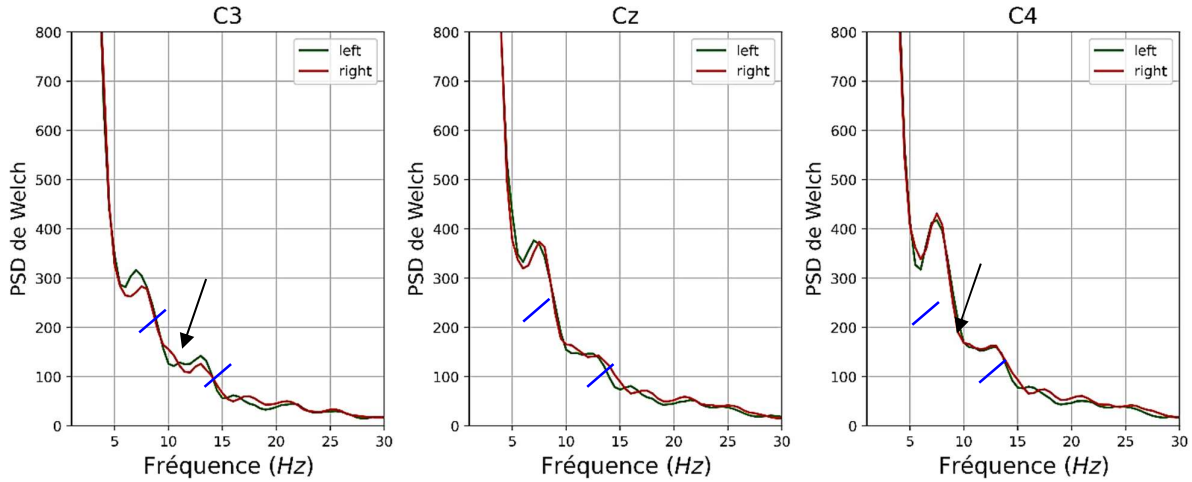


Figure VI.1. Estimation de la densité spectrale de puissance Welch des canaux C3, Cz et C4.

L'activité µ présente un pic sur chaque canal pour les deux classes (-1, 1). Dans l'hémisphère droit, l'activité µ est moins intense pour le mouvement de la main gauche par rapport au mouvement de la main droite en raison de la désynchronisation liée à l'événement (ERD) (voir la section I.5.1.3 du 1er chapitre). Au niveau de l'électrode gauche, l'activité µ est réduite pour le mouvement de la main droite, tandis qu'au niveau de l'électrode centrale, l'activité µ est à peu près égale pour les deux classes. Ces observations sont conformes à la théorie selon laquelle la main gauche est contrôlée par l'hémisphère droit, la main droite par l'hémisphère gauche et les pieds sont contrôlés au niveau central (voir la section I.3.2.2 du 1er chapitre).

Dans les graphiques obtenus à partir du périodogramme de Welch, la région délimitée par les deux lignes correspond à une activité dans la bande de fréquences de 8 à 15 Hz. Afin de mettre en évidence les caractéristiques spécifiques de cette activité, il est recommandé d'appliquer un filtre passe-bande pour isoler les fréquences en dehors de cette plage. Cela permet de mieux visualiser et analyser les motifs pertinents de l'activité dans la bande µ.

Le code ci-dessous est utilisé pour effectuer un filtrage des données d'apprentissage et de test afin d'isoler les fréquences comprises entre 8 et 15 Hz, correspondant à l'activité de la bande µ. Ensuite, il crée des dictionnaires contenant les données des deux classes (left, right) :

```
filtered_train_f = filt.band_pass(train[classes[0]], lowfreq=8, highfreq=15, order=6, sfreq=100, t_ax=1)
filtered_train_r = filt.band_pass(train[classes[1]], lowfreq=8, highfreq=15, order=6, sfreq=100, t_ax=1)
train_filt = trproc.data_class_composer(filtered_train_f, filtered_train_r, classes=classes)

filtered_true_f = filt.band_pass(true[classes[0]], lowfreq=8, highfreq=15, order=6, sfreq=100, t_ax=1)
filtered_true_r = filt.band_pass(true[classes[1]], lowfreq=8, highfreq=15, order=6, sfreq=100, t_ax=1)
true_filt = trproc.data_class_composer(filtered_true_f, filtered_true_r, classes=classes)
```

En réalisant une analyse du périodogramme de Welch sur les données filtrées, il sera possible d'estimer la densité spectrale de puissance (DSP) exclusivement dans la plage de fréquences de 8 à 15 Hz pour chaque classe (figure VI.2).

```
freqs_f, welch_f = analysis.welch_periodogram_analysis(inputData=filtered_train_f, sliding_win=200, sfreq=sfreq, t_ax=1)
freqs_r, welch_r = analysis.welch_periodogram_analysis(inputData=filtered_train_r, sliding_win=200, sfreq=sfreq, t_ax=1)
trials_psd = trproc.data_class_composer(welch_f, welch_r, classes=classes)
plotter.trials_psd_plot(trials_PSD, freqs_f, [channels.index(ch) for ch in ['C3', 'Cz', 'C4']],\
                chan_lab=['C3', 'Cz', 'C4'], maxy=400, xlabel='Fréquence ($Hz$)', \
                ylabel='PSD de Welch', colors=['green', 'red'])
```

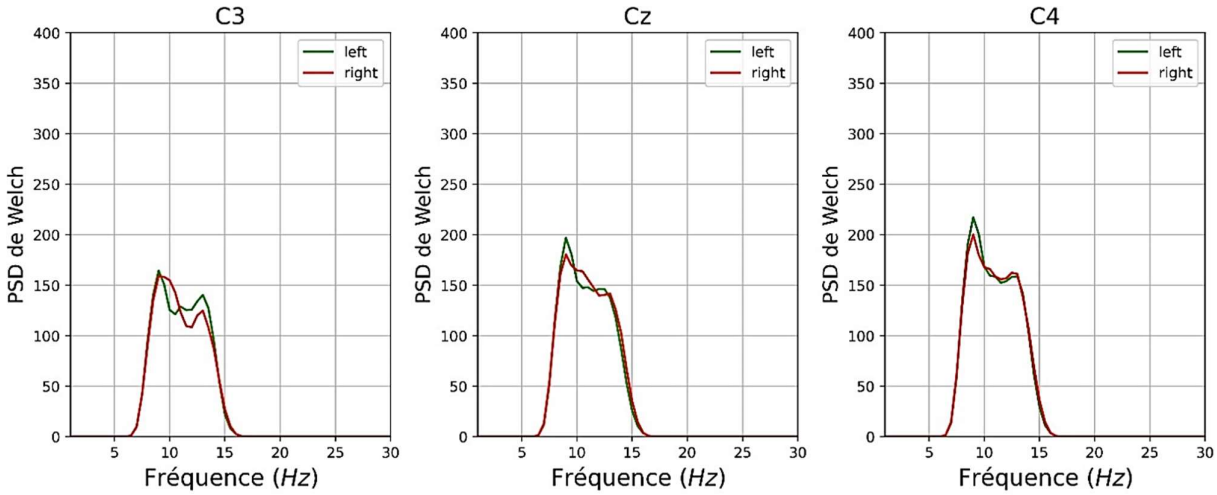


Figure VI.2. Estimation de la densité spectrale de puissance Welch dans la plage (8, 15 Hz) des canaux C3, Cz et C4.

En faisant appel à la fonction *'logvar ()'* du module '*featuresExtractor.py*', nous sommes en mesure d'extraire la variance logarithmique, une mesure statistique qui saisit les fluctuations d'amplitude du signal dans le domaine logarithmique.

```
logvar_f = featext.logvar_feat(filtered_train_f, t_ax=1)
logvar_r = featext.logvar_feat(filtered_train_r, t_ax=1)
```

Pour visualiser les résultats de l'extraction de la variance logarithmique calculée pour chaque essai EEG, la fonction *'plot_logvar ()'* aide à identifier des motifs distinctifs dans les données d'essais *trials_data*, ce qui peut être utile pour la classification ou l'analyse plus approfondie (figure VI.3).

```
trials_logvar = trproc.data_class_composer(logvar_f, logvar_r, classes=classes)
params = dict(title='caractéristique Log-var de chaque canal/composante', ylabel='Log-var',
xlabel='Canaux/composantes (#)')
featplt.plot_logvar(trials_logvar, trial_axis=0, colors=['green', 'red'], **params)
```

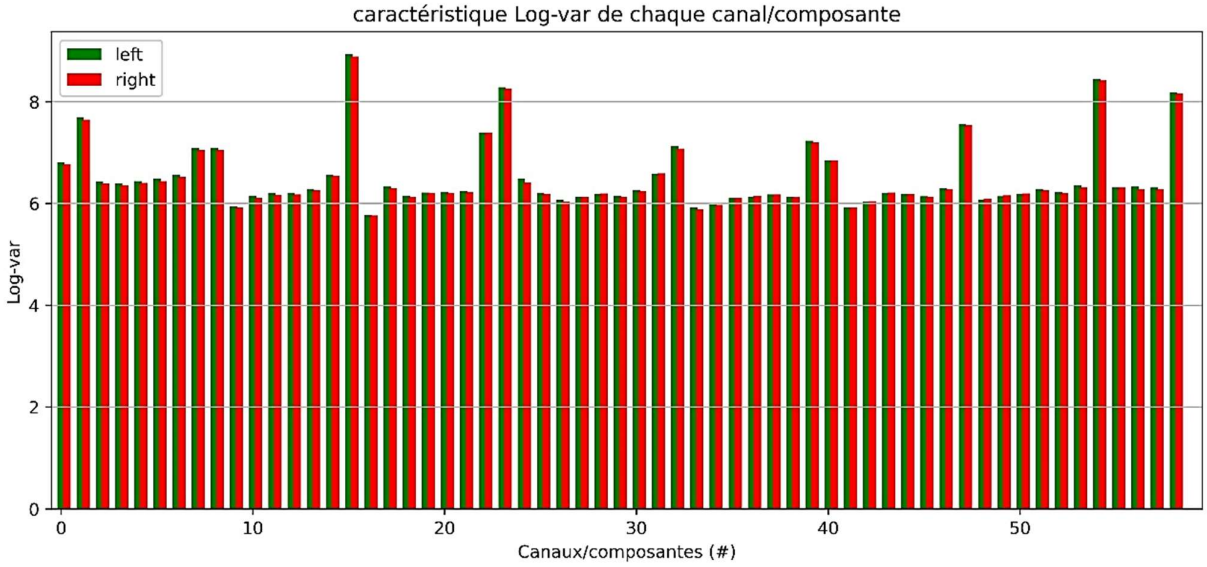


Figure VI.3. Histogramme illustrant la discrimination réduite en raison de la faible variance logarithmique des données brutes.

VI.3.5. Application de la technique CSP (Common Spatial Pattern)

Dans la section précédente, nous avons effectué le filtrage passe-bande avec un filtre Butterworth de 6ème ordre, donc nous avons parvenu à obtenir des caractéristiques plus claires dans la plage de fréquences de 8 à 15 Hz. Cependant, la variance (figure VI.3) entre les caractéristiques des deux classes reste relativement faible, ce qui rend difficile leur discrimination. Pour surmonter cette limitation, nous avons recours à la technique CSP implémenté dans notre boite à outils ***step2bci***, ce qui va permettre de maximiser la séparation entre les caractéristiques des différentes classes.

Lors de l'application du processus CSP à l'ensemble d'apprentissage et de test filtré à l'aide de la fonction *'csp_feat ()'*, il est important de noter que les données de test ne doivent pas être ajustées (*fitted=False*). Nous utilisons le filtre spatial W calculé lors de la première exécution de la fonction *'csp_feat ()'* pour les données d'apprentissage et l'appliquons également aux données de test. En conséquence, nous anticipons l'obtention de caractéristiques plus clairement séparées, favorisant ainsi leur classification.

```
trials_csp_train, W = featext.csp_feat(train_filt, fitted=False, t_ax=1)
trials_csp_true, _ = featext.csp_feat(true_filt, fitted=True, W=W, t_ax=1)
```

Pour voir les résultats de la technique CSP, nous traçons le Log-var pour les données *trials_csp* comme nous l'avons fait précédemment avec les données *trials_data*.

```
train_logvar_f = featext.logvar_feat(trials_csp_train[classes[0]], t_ax=1)
train_logvar_r = featext.logvar_feat(trials_csp_train[classes[1]], t_ax=1)
train_logvar = trproc.data_class_composer(train_logvar_f, train_logvar_r, classes=classes)

params = dict(title='caractéristique Log-var de chaque canal/composante', ylabel='Log-var',
xlabel='Canaux/composantes (#)')
featplt.plot_logvar(trials_logvar, trial_axis=0, colors=['green', red], **params)
```

Au lieu de 59 canaux, nous avons maintenant 118 mélanges de canaux, appelés composantes CSP. Elles sont le résultat de 118 filtres spatiaux appliqués aux données. Les premiers filtres maximisent la variance de la première classe, tout en minimisant la variance de la seconde. Les derniers filtres maximisent la variance de la deuxième classe, tout en minimisant la variance de la première (figure VI.4).

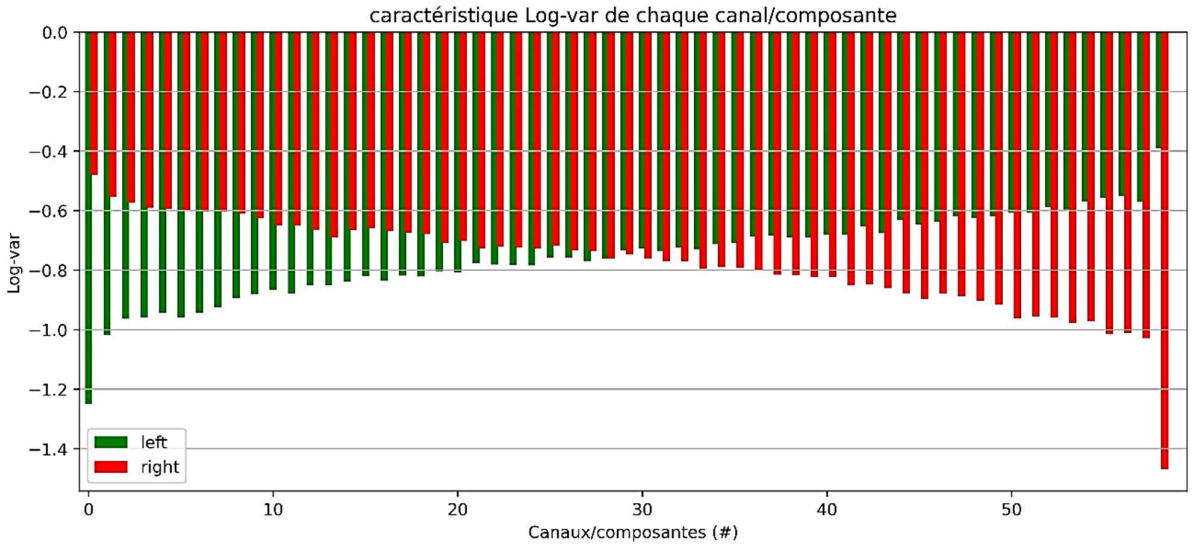


Figure VI.4. Histogramme illustrant la discrimination des caractéristiques par la maximisation de la variance au moyen de la technique du CSP.

Ceci devra également être visible dans un périodogramme de Welch. Le code ci-dessous trace le périodogramme pour les premières et dernières composantes, ainsi que pour la composante centrale :

```
freqs_f, welch_f = analysis.welch_periodogram_analysis(inputData=trials_csp_train[classes[0]],
                                    sliding_win=200, sfreq=sfreq, t_ax=1)
freqs_r, welch_r = analysis.welch_periodogram_analysis(inputData=trials_csp_train[classes[1]],
                                    sliding_win=200, sfreq=sfreq, t_ax=1)
trials_psd = trproc.data_class_composer(welch_f, welch_r, classes=classes)
plotter.trials_psd_plot(trials_psd, freqs_f, chan_ind=[0, 29, -1], chan_lab=['1ère Comp.', 'Comp. centrale', 'Dernière Comp.'],
                maxy=0.75, xlabel='Fréquence ("$Hz$)', ylabel='PSD de Welch', colors=['green', 'red'])
```

Nous constatons aux figures VI.4 et VI.5 une discrimination significative entre les données des deux classes. Ce qui suggère des perspectives prometteuses pour une classification précise. Cette discrimination accrue résulte de la capacité des filtres spatiaux à maximiser la séparation des classes.

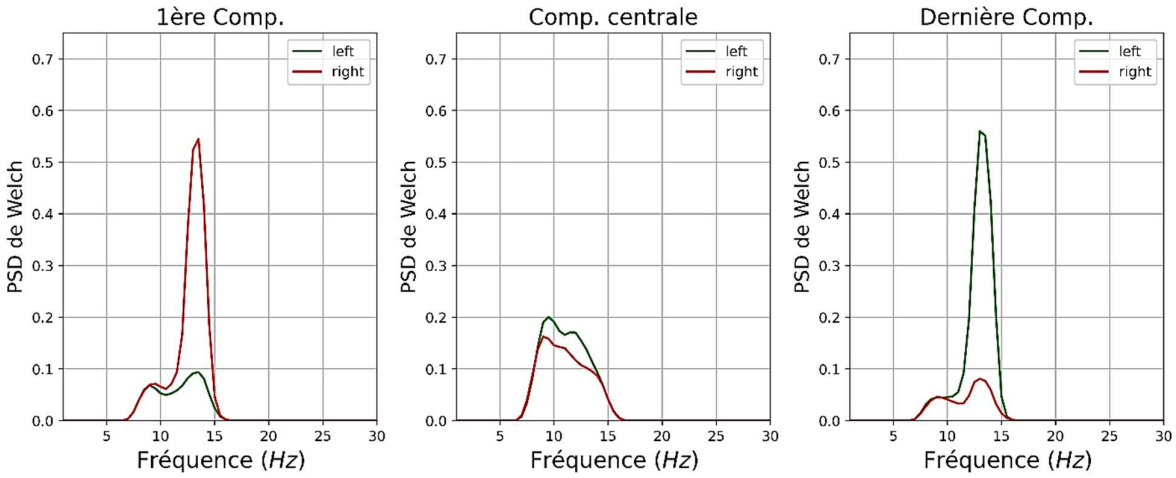


Figure VI.5. Périodogramme de Welch pour les trois composantes discriminantes : 1, 29 et 59.

VI.3.6. Classification

Pour effectuer la classification et la prédiction de la classe d'un échantillon de données non étiqueté, nous devrons créer un modèle d'apprentissage automatique et l'entraîner sur des données étiquetées [Badillo S. et al., 2020]. Cette étape vise à assigner une classe ou une catégorie à chaque échantillon en se basant sur les caractéristiques extraites auparavant. Pour cela, nous préparons les données d'apprentissage et de test en utilisant la fonction *'logvar_csp_composer ()'* qui crée les ensembles de données et les cibles à partir des caractéristiques extraites par la méthode du CSP en intégrant la fonction de calcul de la variance logarithmique :

```
splitted_data = featext.logvar_csp_composer(trials_csp_train, trials_csp_true, classes, target_codes, plot=True)
```

VI.3.6.1. Sélection et réduction de la dimensionnalité des caractéristiques

La fonction *'logvar_csp_composer ()'* focalise spécifiquement sur la sélection des premières et dernières composantes des caractéristiques CSP Il est clair, selon la figure VI.5, que ces deux composantes présentent la variance maximale entre les différentes classes.

Par conséquent, il n'est pas nécessaire de recourir à des fonctions de sélection ou de réduction de la dimensionnalité des tableaux des caractéristiques, telles que *'SFS_select ()'* ou

'PCA_select ()'. En effet, ces deux composantes sélectionnées capturent efficacement les informations les plus discriminantes pour la classification.

VI.3.6.2. Sélection d'estimateurs

Nous utilisons la fonction '*estimator_select ()'* du module '*classifierTools.py*', qui effectue une sélection de modèles en utilisant la méthode de recherche en grille avec validation croisée (*GridSearchCV*). Cette fonction teste plusieurs modèles d'estimateurs, tels que le KNN, la régression logistique, le Naive Bayes multinomial ou gaussien, le LDA, les arbres de décision, la forêt aléatoire, le SVM, etc. Elle évalue chaque modèle en termes de précision et d'autres métriques de performance, et fournit les scores de performance ainsi que les prédictions sur les données de test.

Dans notre cas, nous avons déjà créé des données caractéristiques composées, ce qui signifie que nous n'avons pas besoin de fournir les tableaux *features* et *targets* séparément. Au lieu de cela, nous passons directement la variable *splitted_data* au paramètre *splitted_data*. Cela nous permet d'utiliser les données caractéristiques composées directement dans le processus de sélection des modèles.

```
scores, y_pred = tools.estimator_select(features=None, targets=None, cv=6, ts=None, rs=None,
                                        splitted_data=splitted_data, with_predict=True)
```

Technique d'extraction des caractéristiques : Common Spatial Pattern (CSP), CV=6, test_size=25%					
Estimateur	Meilleurs Paramètres			Précision CV Moyenne	Précision sur l'ensemble de Test
KNN	N_neighbors : 8			**0,9333**	**0,96**
Logistic Regression	C : 0,1	Solver : liblinear		**0,9266**	**0,94**
Gaussian Naïve Bayes	Var_smoothing : 1,0			**0,9333**	**0,94**
Linear Discriminant Analysis	Shrinkage : None	Solver : SVD		**0,95**	**0,94**
Random Forest	Criterion: entropy	Max_depth : None	N_estimators: 200	**0,9266**	**0,94**
Support Vector Machine	C : 20	Gamma : Auto	Kernel : rbf	**0,94**	**0,96**
Decision Tree	Max_Depth: 10	Splitter : Random		**0,9133**	**0,92**

Tableau VI.1. Récapitulatif des performances de la classification par des estimateurs de *Step2bci*.

Nous avons évalué plusieurs estimateurs de classification pour notre étude. Les résultats obtenus révèlent des performances intéressantes pour chaque estimateur. Le K plus proches voisins (KNN) a affiché une précision moyenne de 93,33% lors de la validation croisée et une précision de 96% sur les données de test, ce qui en fait un choix solide pour la classification. La régression logistique (LogReg) a également montré de bonnes performances, avec une précision moyenne de 92,66% lors de la validation croisée et une précision de 94% sur les données de test, il est considéré comme un modèle linéaire simple mais efficace. Quant à l'estimateur Naïve Bayes Gaussien (GNB), il a obtenu des résultats comparables avec une précision moyenne de 93,33% lors de la validation croisée et une précision de 94% sur les données de test, il est particulièrement adapté aux données présentant une distribution gaussienne. L'Analyse discriminante linéaire (LDA) s'est également révélée efficace avec une précision moyenne de 95% lors de la validation croisée et une précision de 94% sur les données de test. En revanche, le Forêt d'arbres décisionnels a obtenu une précision moyenne de 92,66% lors de la validation croisée et une précision de 94% sur les données de test. Le classifieur Machine à vecteurs de support (SVM) a présenté d'excellentes performances avec une précision moyenne de 94% lors de la validation croisée et une précision de 96% sur les données de test, il est souvent efficace pour les tâches de classification complexes. Enfin, l'estimateur de l'Arbre de décision a affiché la précision la plus basse, avec une précision moyenne de 91,33% lors de la validation croisée et une précision de 92% sur les données de test. Ces résultats suggèrent que le choix de l'estimateur dépendra des caractéristiques des données et des exigences spécifiques de la tâche de classification.

En considérant ces performances, nous pouvons conclure que les estimateurs KNN, régression logistique, Naïve Bayes Gaussien, LDA, Forêt d'arbres décisionnels et SVM ont tous montré de bonnes capacités de classification sur nos données, tandis que l'arbre de décision a obtenu des résultats légèrement inférieurs.

VI.3.6.3. Création des modèles d'apprentissage automatique types à l'aide de la boite à outils Step2bci

Dans cette section nous créons des modèles de classification en utilisant différents algorithmes d'apprentissage automatique. Nous utilisons trois types de classifieurs : KNN, LDA et SVM. Pour chaque classifieur, nous définissons les paramètres appropriés et entraînons le modèle sur les données d'entraînement (*X_train*) avec les étiquettes correspondantes (*y_train*). Ensuite, nous utilisons le modèle entraîné pour prédire les étiquettes des données

réelles (*X_true*) et calculons la précision de classification en comparant les prédictions aux véritables étiquettes (*y_true*).

Enfin, nous générons des rapports de classification qui fournissent des informations détaillées sur les performances du modèle, y compris la matrice de confusion. Les rapports nous aident à évaluer la qualité de la classification réalisée par chaque modèle.

- Classifieur KNN

```
params = {'n_neighbors': 8}
y_pred_knn, knn = clf.knn_classifier(X_train=train, y_train=train_targets, X_true=true, **params)
accuracy_knn = tools.accuracy_clf(y_true=true_targets, y_pred=y_pred_knn)
reports_knn = tools.report_clf(estimator=knn, y_true=true_targets, y_pred=y_pred_knn, conf_matrix_plot=True)

print("Précision de la classification : %.2f" % accuracy_knn)
```

```
Output :
Précision de la classification : 0.96
```

- Classifieur LDA

```
params = {'shrinkage': None, 'solver': 'svd'}
y_pred_lda, lda = clf.lda_classifier(X_train=train, y_train=train_targets, X_true=true, **params)

accuracy_lda = tools.accuracy_clf(y_true=true_targets, y_pred=y_pred_lda)
reports_lda = tools.report_clf(estimator=lda, y_true=true_targets, y_pred=y_pred_lda, conf_matrix_plot=True)

print("Précision de la classification : %.2f" % accuracy_lda)
```

```
Output :
Précision de la classification : 0.94
```

- Classifieur SVM

```
params = {'C': 20, 'gamma': 'auto', 'kernel': 'rbf'}
y_pred_svm, svm = clf.svm_classifier(X_train=train, y_train=train_targets, X_true=true, fastmode=True,
                        gridsearch=False, max_iter=50, **params)

accuracy_svm = tools.accuracy_clf(y_true=true_targets, y_pred=y_pred_svm)
reports_svm = tools.report_clf(estimator=svm, y_true=true_targets, y_pred=y_pred_svm, conf_matrix_plot=True)

print("Précision de la classification : %.2f" % accuracy_svm)
```

```
Output :
Précision de la classification : 0.96
```

Les sorties affichées montrent la précision de la classification obtenue pour chaque classifieur. Dans ce cas, le classifieur KNN a une précision de 0,96, le classifieur LDA a une précision de 0,94 et le classifieur SVM a également une précision de 0,96.

Les matrices de confusion (figure VI.6) fournissent des informations importantes sur les performances des classifieurs. Pour le classifieur KNN, la matrice de confusion montre qu'il a correctement prédit 22 instances en tant que négatives et 26 instances en tant que positives, avec seulement 2 erreurs de faux négatifs. Le classifieur LDA a également obtenu des résultats similaires, avec 22 prédictions correctes pour les instances négatives et 26 prédictions correctes pour les instances positives, mais également 2 erreurs de faux négatifs. En revanche, le classifieur SVM a obtenu 21 prédictions correctes pour les instances négatives, 27 prédictions correctes pour les instances positives, mais a commis une erreur de faux positif et une erreur de faux négatif.

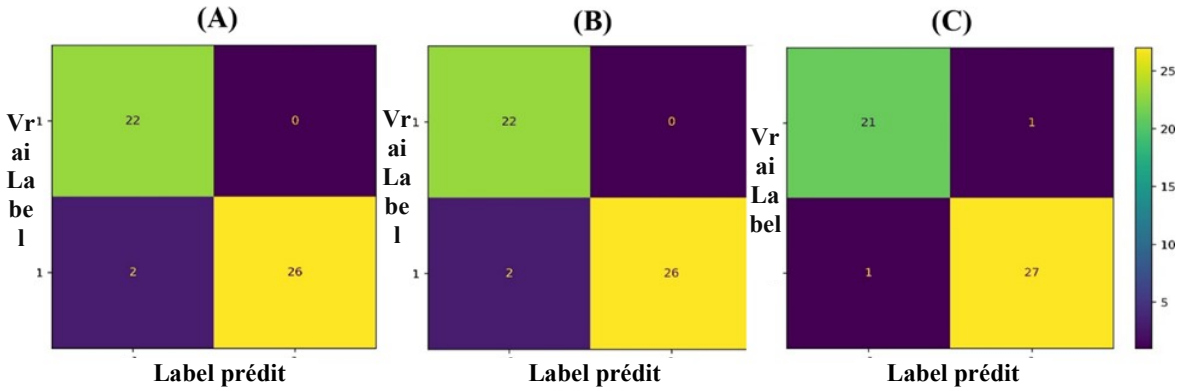


Figure VI.6. Résultats de la classification par la matrice de confusion (A) KNN, (B) LDA, (C)SVM.

Ces matrices de confusion nous permettent de comprendre la précision et les erreurs de classification de chaque modèle, ce qui nous aide à évaluer leur efficacité dans la prédiction précise des étiquettes réelles.

VI.3.6.4. Création du modèle d'apprentissage profond à l'aide de la boîte à outils Step2bci

Pour permettre l'utilisation des fonctionnalités de classification par apprentissage profond fournies par le package *'deepproc'*, nous suivons notre convention d'importation en important les packages et classes nécessaires. Tout d'abord, nous importons *'torch.optim'* qui fournit les optimiseurs nécessaires pour l'entraînement du modèle. Ensuite, nous importons les classes *'NetPytorch', 'TorchTrainer', 'TorchEvaluator' et 'TorchPredictor'* du module '*deepBCI.py*' du package « *deepproc* ». Ces classes jouent des rôles essentiels dans l'architecture du modèle, l'entraînement, l'évaluation et les prédictions avec les modèles d'apprentissage approfondi. Enfin, nous importons la classe *CustomDataset* du module '*neuroDataset.py*' qui nous permet de créer des jeux de données personnalisés adaptés à notre application. De plus, nous importons la classe

'DeviceSwitcher' du module '*neuroDevice.py*' qui facilite la gestion des dispositifs (CPU, GPU) utilisés pour l'entraînement et l'inférence du modèle.

```
import torch.optim
from deepproc.deepBCI import NetPytorch, TorchTrainer, TorchEvaluator, TorchPredictor
from deepproc.neuroDataset import CustomDataset
from deepproc.neuroDevice import DeviceSwitcher
```

Tout d'abord, les données d'entraînement et de test sont préparées et les étiquettes cibles sont recodées pour correspondre aux valeurs 0 et 1 au lieu de -1 et 1.

```
train_targets = np.where(train_targets == -1, 0, train_targets)
true_targets = np.where(true_targets == -1, 0, true_targets)
```

Les données sont divisées en ensembles d'entraînement et de test à l'aide de la classe *'CustomDataset'* du module *'neuroDataset.py'*.

```
dataset_train = CustomDataset((train, train_targets), None, None)
dataset_true = CustomDataset((true, true_targets), None, None)
```

Ensuite, les hyperparamètres du modèle sont définis, tels que le nombre de caractéristiques, le nombre de classes, la taille des couches cachées, la taille des lots *batch size*, le taux d'apprentissage, la probabilité d'abandon *'dropout'*, le nombre d'époques, etc.

```
n_features = train.shape[1]
nclasses = len(np.unique(true_targets))
hidden_sizes = [8]
batch_size1 = len(train_targets) // 8
batch_size2 = len(true_targets) // 8
learning_rate = 0.19
dropout_prob = .2
n_epochs = 1000
momentum = .94
weight_decay = 0.1
```

Nous avons spécifié '*cpu*' comme argument lors de la création de l'objet '*DeviceSwitcher.py*', ce qui signifie que nous utilisons le processeur central (CPU) pour nos opérations de calcul.

```
device = DeviceSwitcher('cpu').device
```

Le modèle est ensuite défini à l'aide de la classe *'NetPytorch'* du module '*deepBCI.py*', qui spécifie l'architecture du réseau neuronal.

```
model = NetPytorch(n_features, nclasses, *hidden_sizes, dropout_prob=dropout_prob)
```

La ligne de code ci-dessous initialise un objet *'TorchTrainer'* appelé *trainer*, qui est chargé d'entraîner un modèle d'apprentissage profond. Elle prend plusieurs paramètres, notamment l'architecture du modèle, les données d'entraînement, la taille des lots *batch size*, le

nombre d'époques maximum *max_epochs*, le taux d'apprentissage *learning rate*, la quantité de momentum (*momentum*), la décroissance du poids *weight_decay*, le dispositif de calcul (CPU ou GPU) et le type d'optimisateur. La classe '*TorchTrainer'* encapsule le processus d'entraînement, mettant à jour de manière itérative les paramètres du modèle pour minimiser la fonction de perte (de coût).

```
trainer = TorchTrainer(model=model, train_data=dataset_train, batch_size=batch_size1, max_epochs=n_epochs,
                       learning_rate=learning_rate, momentum=momentum, weight_decay=weight_decay, device=device,
                       optimizer=torch.optim.Adam)

trained_model, train_accuracy, epoch_losses, eval_losses = trainer.train(eval_train=True, patience=50,
save_model=False)
```

Ensuite, nous allons créer un objet '*TorchEvaluator'* appelé *evaluator*, qui est utilisé pour évaluer les performances du modèle entraîné. Il prend plusieurs paramètres, dont le modèle entraîné, le critère de perte (dans ce cas, la fonction de perte *CrossEntropyLoss* de PyTorch), la taille des lots *batch size*, et le dispositif de calcul (CPU, GPU). La classe *'TorchEvaluator'* calcule la précision et la perte du modèle sur les données de validation. En appelant la méthode *evaluate* sur l'objet *evaluator* et en fournissant les données de validation, on obtient la précision et la perte du modèle sur ces données.

```
evaluator = TorchEvaluator(model=trained_model, criterion=torch.nn.CrossEntropyLoss, batch_size=batch_size2,
                           device=device)
test_accuracy, test_loss = evaluator.evaluate(val_data=dataset_true, eval_train=False)
```

Ces métriques permettent d'évaluer les performances du modèle sur un ensemble de données indépendant, fournissant ainsi une mesure de sa capacité à généraliser à de nouvelles données. Le modèle est utilisé pour faire des prédictions sur l'ensemble de test à l'aide de la classe *'TorchPredictor'*. Donc, nous allons créer un objet *'TorchPredictor'* appelé *predictor*, cet objet est utilisé pour effectuer des prédictions à l'aide du modèle entraîné. Il prend plusieurs paramètres, notamment le modèle entraîné (*model*), la taille de l'entrée (nombre de caractéristiques), et le dispositif de calcul.

```
predictor = TorchPredictor(model=model, input_size=n_features, device=device)
```

Finalement, nous utilisons la méthode *'predict_batch ()'* (prédiction par lot) de l'objet *predictor* pour effectuer des prédictions sur un lot d'échantillons spécifique. Dans cet exemple, les prédictions sont faites uniquement sur 30 échantillons des données de test et d'évaluation. Les tenseurs de prédiction renvoyés sont stockés dans la variable *predictions*.

```
predictions = predictor.predict_batch(dataset_true.data[20:50])
```

```
Output :
$ Arrêt précoce après 59 epochs.
$ Précision sur l'ensemble de validation : 0.98000
$ Epoch [59/1000] ... loss : 0.36398205492231583 ... Précision sur l'ensemble
d'entraînement : 0.9333 ... validation loss : 0.3143145226769977

$ tensor([0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1])
$ tensor([0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1])
```

La précision de l'entraînement et du test (ou de validation), ainsi que les pertes d'entraînement et d'évaluation, sont affichées (Figure VI.7). Un graphique est également tracé pour visualiser les performances du modèle au cours de l'étape d'apprentissage.

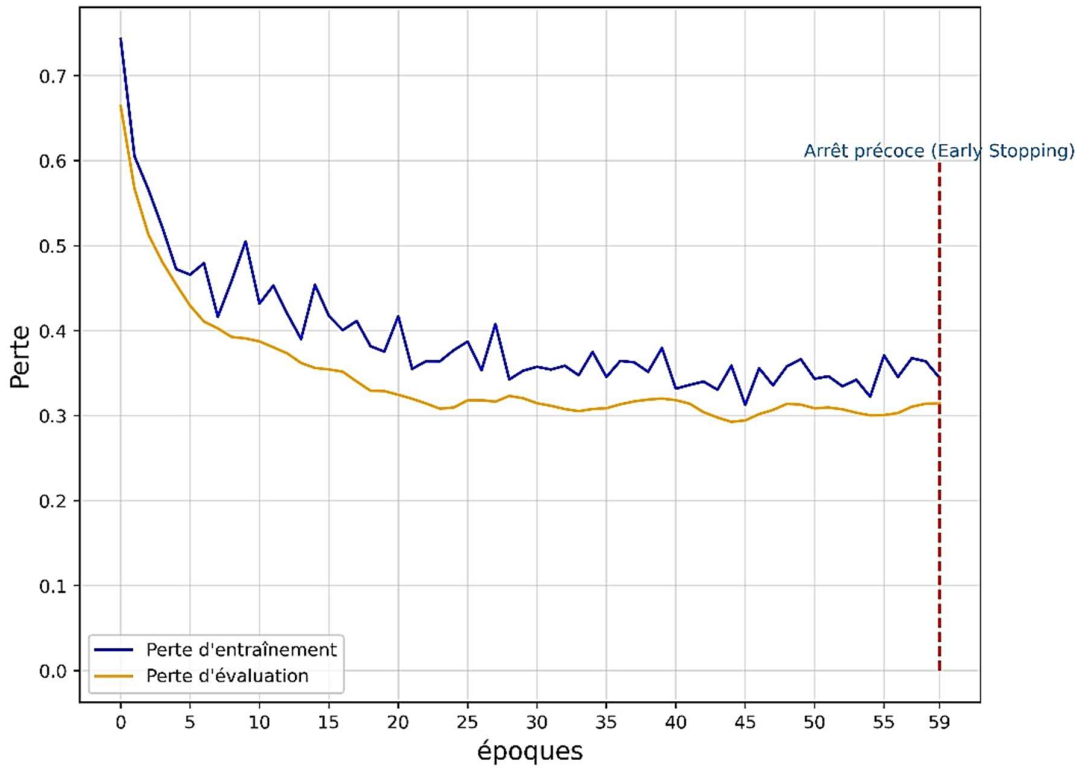


Figure VI.7. Évaluation des performances du modèle Pytorch au cours de la phase d'apprentissage.

VI.3.7. Interprétation et discussion des résultats

Ce protocole expérimental permet donc de charger les données, de former un modèle d'apprentissage profond, de l'évaluer et de faire des prédictions sur de nouveaux exemples avec une précision qui atteigne 98%.

L'apprentissage automatique utilise des algorithmes classiques tels que : KNN, Régression logistique, Naïve Bayes Gaussien, LDA, forêts aléatoires et SVM. Les résultats d'apprentissage automatique obtenus dans cet exemple sont prometteurs, avec des précisions de

test allant de 0,92 à 0,96. D'autre part, l'apprentissage profond, qui repose sur des réseaux MLP, permet d'obtenir des résultats plus puissants en exploitant des architectures complexes. Dans l'exemple donné, un modèle d'apprentissage profond a été entraîné pendant 59 epochs et l'entraînement a été arrêté prématurément (*Early Stopping*).

En comparant les résultats, on peut observer que l'apprentissage profond a atteint une précision légèrement supérieure à celle de l'apprentissage automatique dans cet exemple spécifique. Cependant, il convient de noter que la comparaison entre les deux approches nécessite une évaluation plus approfondie, car les performances peuvent varier en fonction du jeu de données et du problème traité. L'apprentissage automatique peut être plus adapté pour des tâches simples ou lorsque les données d'entraînement sont limitées, tandis que l'apprentissage profond excelle généralement dans des tâches plus complexes et avec des ensembles de données plus vastes.

VI.4. Protocole d'imagination motrice avec et sans rétroaction (feedback)

VI.4.1. Présentation des données

Ce jeu de données comprend des données EEG provenant d'une étude publiée dans [Leeb R. et al., 2007] portant sur un sujet ayant une vision normale. Pendant l'expérience le sujet était assis dans un fauteuil en regardant un écran plat placé à environ 1 mètre de distance, à hauteur des yeux. Le jeu de données propose 5 sessions pour le sujet, les deux premières sessions étant des données d'entraînement sans rétroaction, et les trois dernières sessions enregistrées avec rétroaction.

Chaque session se compose de plusieurs séries. Au début de chaque session, un enregistrement d'environ 5 minutes était réalisé pour évaluer l'influence de l'EOG (*ÉlectroOculoGramme*). Ensuite, trois enregistrements bipolaires (C3, Cz et C4) ont été enregistrés à une fréquence d'échantillonnage de 250 Hz (figure VI.8).

Le protocole synchrone (cue-based) basé sur les indices se composait de deux classes, à savoir l'imagination motrice (MI) de la main gauche (classe 1) et de la main droite (classe 2). Le sujet a participé à deux sessions sans rétroaction (*without feedback*) enregistrées les jours différents dans les deux semaines. Chaque session se composait de 120 essais par session pour chaque classe ce qui donne au total 240 essais par session.

Chaque essai commençait par une croix de fixation et un bref signal sonore d'avertissement (1 kHz, 70 ms). Quelques secondes plus tard, un indice visuel (une flèche pointant vers la gauche ou la droite, selon la classe demandée) était présenté pendant

1,25 seconde. Ensuite, les sujets devaient imaginer le mouvement de la main correspondante pendant une période de 4 secondes. Chaque essai était suivi d'une courte pause d'au moins 1,5 seconde. Un temps aléatoire allant jusqu'à 1 seconde était ajouté à la pause pour éviter l'adaptation.

Ensuite, le sujet a participé à trois sessions avec rétroaction (*with feedback*). Au début de chaque essai (seconde 0), la rétroaction (un sourire gris) était centrée sur l'écran. À la seconde 2, un bref bip d'avertissement (1 kHz, 70 ms) était émis. L'indice était présenté de la seconde 3 à la seconde 7,5. En fonction de l'indice, les sujets devaient déplacer le sourire vers la gauche ou la droite en imaginant des mouvements de la main gauche ou droite, respectivement.



Figure VI.8. Montage des électrodes des trois canaux EEG bipolaires (C3, C4 et Cz).

VI.4.2. Préparation des données

Nous importerons des modules et des classes spécifiques de la boîte à outils pour faciliter ces opérations.

```
import numpy as np
import matplotlib.pyplot as plt
import preproc.IOfiles as iof
import preproc.dataFilter as filt
import postproc.featuresExtractor as featext
import postproc.featuresSelector as featselect
import preproc.dataScaler as scal
import preproc.trialsProcessor as tr_process
```

Nous procédons au chargement des données d'entraînement *X_T* (*Données d'entraînement*) et d'évaluation *X_E* (Données d'évaluation) respectivement à partir des fichiers ***B01T.mat*** et ***B01E.mat*** à l'aide de la fonction '*iof.read_graz_file ()*'. Les étiquettes cibles des ensembles d'entraînement et de test (évaluation) sont stockées dans les variables *targets_T* et *targets_E*.

```
X_T, trials_T, targets_T, attributes_T = iof.read_graz_file('../datasets/grazdata/B01T.mat')
X_E, trials_E, targets_E, attributes_E = iof.read_graz_file('../datasets/grazdata/B01E.mat')
```

Ensuite, un filtre passe-bande de Butterworth d'ordre 6 est utilisé pour atténuer les fréquences indésirables qui se situent en dehors de la plage de fréquences contenant le rythme μ, lequel est compris entre 8 et 15 Hz.

```
signal_T = filt.band_pass(inputData=X_T, lowfreq=8, highfreq=15, order=6, sfreq=sfreq, t_ax=0)
signal_E = filt.band_pass(inputData=X_E, lowfreq=8, highfreq=15, order=6, sfreq=sfreq, t_ax=0)
```

La normalisation des données est une pratique courante en traitement des signaux EEG qui vise à mettre toutes les valeurs sur une échelle comparable. En effet, nous appliquons une normalisation aux variables *signal_T* et *signal_E*. La fonction *'std_scaler ()'* du module *'dataScaler.py'* effectue une normalisation standard en soustrayant la moyenne et en divisant par l'écart type du signal. L'argument (*with_std=True*) indique que l'écart type doit être utilisé pour la normalisation, tandis que (*with_mean=0*) indique que la moyenne doit être soustraite.

```
normalized_signal_T = scal.std_scaler(signal_T, with_std=True, with_mean=0
normalized_signal_E = scal.std_scaler(signal_E, with_std=True, with_mean=0)
```

VI.4.3. Création des ensembles d'apprentissage et de test

Les ensembles d'entraînement et de test sont déjà fournis séparément dans les fichiers B01T.mat et B01E.mat mentionnés. Les variables *normalized_signal_T* et *normalized_signal_E* représentent les données d'entraînement et d'évaluation normalisées, tandis que les étiquettes cibles des deux ensembles sont stockées respectivement dans *targets_T* et *targets_E*. Ainsi, les données sont déjà divisées et prêtes à être utilisées pour l'entraînement et l'évaluation, éliminant ainsi la nécessité d'utiliser la fonction *'train_test_clf ()'* du module *'classifierTools.py'*.

VI.4.4. Extraction des caractéristiques par décomposition en ondelettes

La décomposition en ondelettes est utilisée pour extraire les caractéristiques des signaux EEG en capturant simultanément les informations temporelles et fréquentielles. Elle permet d'analyser les signaux à différentes échelles et résolutions, en identifiant les composantes à haute et basse fréquence. Cette méthode met en évidence les événements transitoires et les modèles oscillatoires pertinents pour l'analyse EEG, ainsi que les bandes de fréquence associées à différentes activités cérébrales.

On effectue une analyse en ondelettes discrètes DWT (*Digital Wavelet Transform*) sur les données d'entraînement pour extraire des coefficients de décomposition. Les données d'entraînement sont divisées en deux classes (class1 : *Reft* et class2 : *Right*) en fonction des codes cibles. Les données d'entrée doivent être un tableau tridimensionnel avec le format suivant

: *(Nombre de classes, Coefficients de basse/haute fréquence, Nombre de canaux)*. La fonction *'wavelets_decomp_plot ()'* du module *'dataPlotter.py'* génère des sous-graphiques (figure VI.9 et figure VI.10) pour chaque classe, affichant les coefficients moyens à un niveau donné, ainsi que les coefficients de différence pour les niveaux restants. Les signaux correspondants des canaux C3 et C4 sont analysés en utilisant la fonction *'wavedec_dwt_analysis ()'* avec une ondelette mère de type *'Debauchies 4'* et un niveau de décomposition égal à 6.

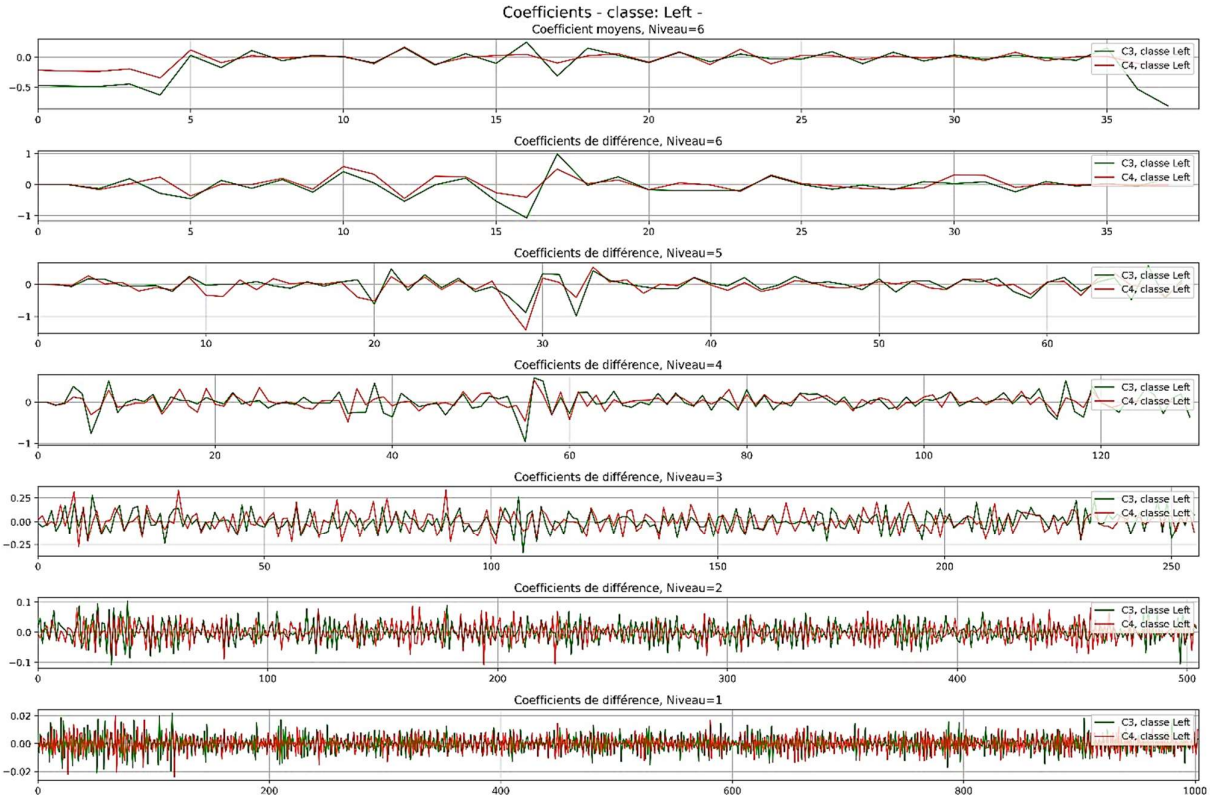


Figure VI.9. Les coefficients de décomposition de 6 niveaux pour la classe 1 (Left) des canaux C3 et C4.

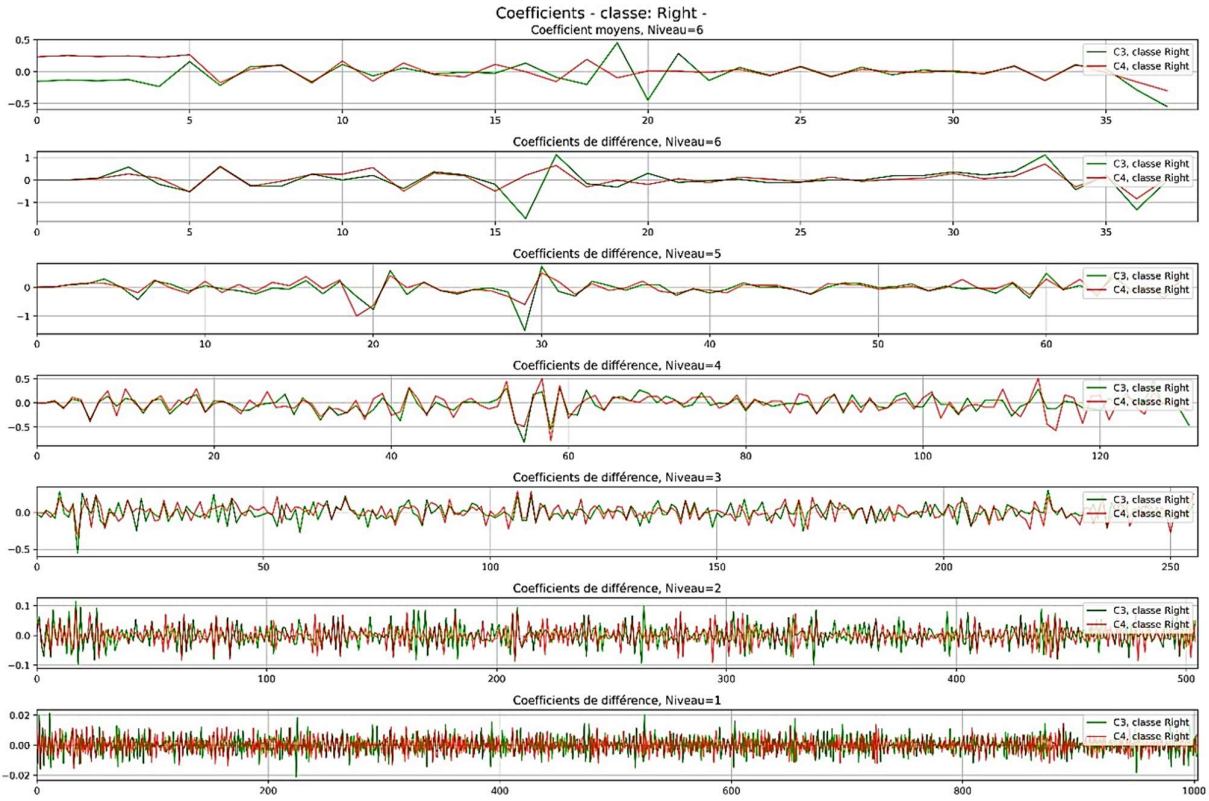


Figure VI.10. Les coefficients de décomposition de 6 niveaux pour la classe 1 (Left) des canaux C3 et C4.

La fonction *'featext.wavelets_feat ()'* est utilisé pour extraire des caractéristiques à partir de signaux *normalized_signal_T* et *normalized_signal_E*. Les caractéristiques sont calculées en utilisant différentes fenêtres temporelles telles que :

```
window = [-125, 125]
w1_a = [-125, 400]
w2_a = [400, 925]
w1_b = [-150, 50]
w2_b = [70, 670]
```

L'utilisation de différentes fenêtres temporelles permet d'explorer les caractéristiques du signal à différentes échelles temporelles, ce qui permet d'extraire des informations à des résolutions temporelles variées. Cette approche permet d'obtenir une représentation plus exhaustive et détaillée des motifs et des variations présentes dans le signal EEG, offrant ainsi une vue plus complète et précise de ses caractéristiques.

```
features1_T = featext.wavelets_feat(normalized_signal_T, trials=trials_T, window=w1_a, wavelet='db6', timeStep=4,
                        level_to_extract=4, sfreq=sfreq, max_level=5, t_ax=0)
features2_T = featext.wavelets_feat(normalized_signal_T, trials=trials_T, window=w2_a, wavelet='db4', timeStep=4,
                        level_to_extract=4, sfreq=sfreq, max_level=5, t_ax=0)
features3_T = featext.wavelets_feat(normalized_signal_T, trials=trials_T, window=w2_b, wavelet='db2', timeStep=4,
                        level_to_extract=3, sfreq=sfreq, max_level=5, t_ax=0)
features4_T = featext.wavelets_feat(normalized_signal_T, trials=trials_T, window=w1_a, wavelet='db2', timeStep=4,
                        level_to_extract=3, sfreq=sfreq, max_level=5, t_ax=0)
features5_T = featext.wavelets_feat(normalized_signal_T, trials=trials_T, window=window, wavelet='db6', timeStep=4,
                        level_to_extract=4, sfreq=sfreq, max_level=5, t_ax=0)

features1_E = featext.wavelets_feat(normalized_signal_E, trials=trials_E, window=w1_a, wavelet='db6', timeStep=4,
                        level_to_extract=4, sfreq=sfreq, max_level=5, t_ax=0)
features2_E = featext.wavelets_feat(normalized_signal_E, trials=trials_E, window=w2_a, wavelet='db4', timeStep=4,
                        level_to_extract=4, sfreq=sfreq, max_level=5, t_ax=0)
features3_E = featext.wavelets_feat(normalized_signal_E, trials=trials_E, window=w2_b, wavelet='db2', timeStep=4,
                        level_to_extract=3, sfreq=sfreq, max_level=5, t_ax=0)
features4_E = featext.wavelets_feat(normalized_signal_E, trials=trials_E, window=w1_a, wavelet='db2', timeStep=4,
                        level_to_extract=3, sfreq=sfreq, max_level=5, t_ax=0)
features5_E = featext.wavelets_feat(normalized_signal_E, trials=trials_E, window=window, wavelet='db6', timeStep=4,
                        level_to_extract=4, sfreq=sfreq, max_level=5, t_ax=0)
```

Ensuite nous réalisons la concaténation des caractéristiques extraites à partir des différentes fenêtres temporelles pour les ensembles d'entraînement et de test. La fonction *'np.concatenate ()'* est utilisée pour fusionner les caractéristiques extraites de chaque fenêtre temporelle en un seul tableau *np.ndarray*.

```
features_T = np.concatenate((features1_T, features2_T, features3_T, features4_T, features5_T), axis=1)
features_E = np.concatenate((features1_E, features2_E, features3_E, features4_E, features5_E), axis=1)
```

Les variables *features_T* et *features_E* contiennent les caractéristiques extraites des différentes fenêtres temporelles pour les ensembles d'entraînement et de test, respectivement, regroupées dans une matrice où chaque ligne représente une fenêtre temporelle et chaque colonne représente une caractéristique spécifique.

Chaque fenêtre temporelle extrait 9 caractéristiques, et étant donné qu'il y a 5 fenêtres utilisées (*features1_T*, *features2_T*, *features3_T*, *features4_T*, *features5_T*) pour l'ensemble d'entraînement (*features_T*) et également pour l'ensemble d'évaluation (*features_E*), le nombre total des caractéristiques dans les matrices *features_T* et *features_E* sera de 5 * 9 = 45. Par conséquent, les matrices *features_T* et *features_E* auront toutes deux une dimension de (*nombre d'essais*, *45*).

```
print(f'Format des données d\'origine : {features.shape}')
```

```
Output :
$ Format des données d'origine : (552, 45)
```

Bien que 45 caractéristiques ne soient pas nécessairement excessives pour notre cas d'étude (*Nombre d'essais égal à 552*), il peut être utile de considérer la réduction de dimensionnalité pour éviter des problèmes de complexité du modèle, de sur-apprentissage ou d'interprétabilité.

VI.4.5. Réduction de dimensionnalité

Le processus de réduction de la dimensionnalité peut être réalisé en utilisant deux approches différentes : l'analyse en composantes principales PCA via la fonction *PCA_select ()'* ou la sélection séquentielle de caractéristiques SFS via la fonction *'SFS_select ()'*. PCA transforme les caractéristiques en composantes non corrélées, réduisant ainsi la dimensionnalité des données tout en préservant une grande partie de leur variance. En revanche, la SFS évalue différentes combinaisons de caractéristiques pour sélectionner un sous-ensemble optimal qui maximise les performances du modèle.

VI.4.5.1. Approche PCA

On utilise la fonction *'estimate_pca_dimensions ()'* du module *'featSelect.py'* pour estimer le nombre optimal de composantes PCA à conserver. Cela est fait en passant les données *features_T* et *features_E*, ainsi qu'un seuil de variance expliquée de 0,95. La fonction retourne le nombre de composantes PCA à conserver.

```
n_components = featselect.estimate_pca_dimensions(train=features_T, true=features_E, threshold=0.95, plot=True)
```

La figure VI.11 représente le taux de variance expliquée cumulée, qui indique la part totale de variance des données d'origine expliquée par un nombre croissant de composantes principales.

Nous avons déterminé que le nombre optimal de composantes PCA à conserver est de 27, en partant des 45 caractéristiques d'origine.

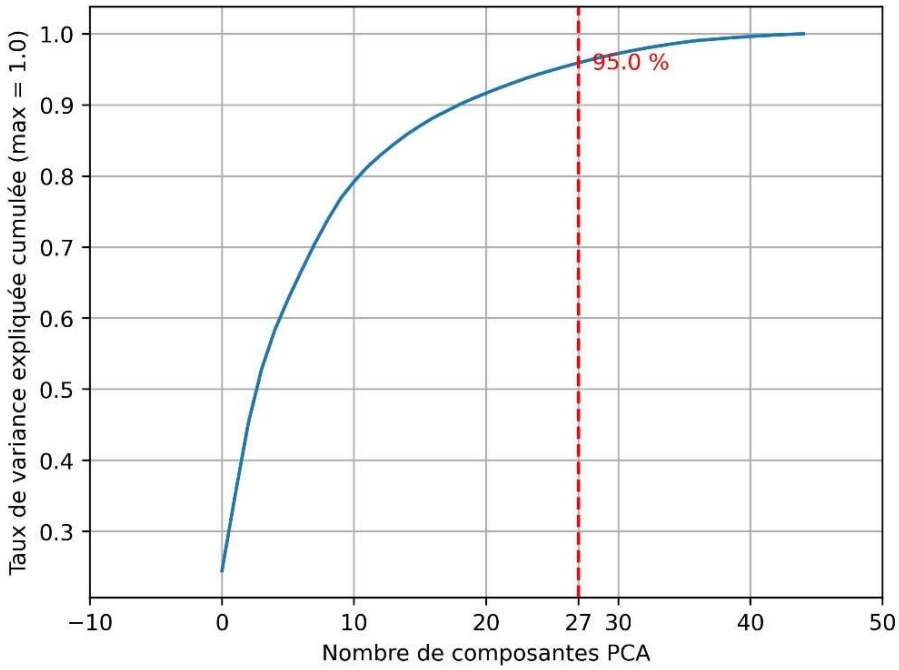


Figure VI.11. Taux de la variance expliquée cumulée en fonction du nombre de composantes, et le seuil de variance est un critère de choix.

La fonction *'PCA_select ()'* est utilisée pour effectuer la réduction de dimensionnalité en utilisant l'algorithme PCA. On passe comme paramètre le nombre de composantes PCA obtenu précédemment. Les variables *pca_data_T et pca_data_E* contiennent les données réduites en dimension pour les ensembles d'entraînement et de test, respectivement.

```
pca_data_T, _ = featselect.PCA_select(features_T, targets_T, n_comp=n_components, pca_plot=True)
pca_data_E, _ = featselect.PCA_select(features_E, targets_E, n_comp=n_components, pca_plot=True)
print(f'Format des données d'entraînement réduites en dimensions : {pca_data_T.shape}')
```

```
Output :
$ Format des données d'entraînement réduites en dimensions : (552, 27)
```

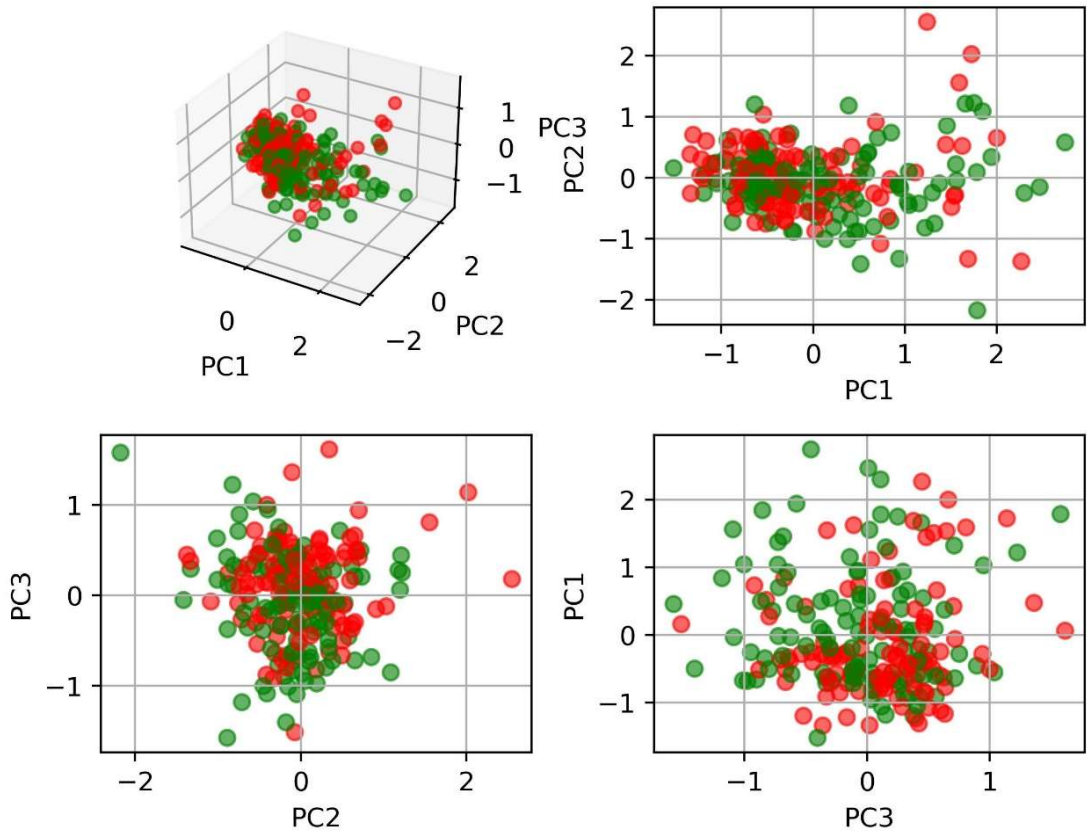


Figure VI.12. Illustration 2D et 3D des trois premières composantes principales de l'ensemble d'entrainement *pca_data_T* calculées par PCA.

VI.4.5.2. Approche SFS

```
gscv, pipeline = featselect.SFS_select(features, targets, validation_size=0.2, forward=True, floating=True, clfs=['knn',
                                                'svm', 'lda'], k_feat_=27, verbosity=True, all_cpu=True, cv=2, fitting_all_data=False, refit=False)
```

L'appel à la fonction *'SFS_select ()'* du module *'featuresSelector.py'* effectue une sélection de caractéristiques en utilisant la méthode de recherche séquentielle (SFS) avec validation croisée (*CV*). Les paramètres spécifiés incluent l'utilisation d'une taille de validation de 20%, l'activation de la recherche en avant et flottante, et la sélection des classifieurs KNN, SVM et LDA. La fonction recherche les meilleures caractéristiques en utilisant un nombre fixe par notre choix de 27.

Lors de l'exécution, la verbosité est activée, ce qui affiche les résultats intermédiaires.

```
$ [2023-06-22 11:46:45] Features: 26/27 -- score: 0.7782145782145782[Parallel(n_jobs=-
1)]: Using backend ThreadingBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done 19 out of 19 | elapsed:    0.1s finished
[Parallel(n_jobs=-1)]: Using backend ThreadingBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done 26 out of 26 | elapsed:    0.2s finished
[2023-04-22 11 :46 :45] Features : 27/27 -- score : 0.7782555282555282
```

L'exemple donné montre l'utilisation de plusieurs tâches parallèles avec 4 concurrents (*concurrent workers*) pour accélérer le processus via le paramètre (*all_cpu=True*). La sortie affiche également le temps écoulé pour chaque tâche exécutée.

De plus, (*refit=False*) indique qu'une fois que les meilleures caractéristiques ont été identifiées à l'aide de la sélection séquentielle, le modèle correspondant ne sera pas entraîné à nouveau sur toutes les données disponibles. L'appel à la fonction retourne les objets *gscv* et *pipeline*, qui contiennent respectivement les modèles ajustés et les pipelines utilisés pour la sélection des caractéristiques. Cette approche permet de réduire l'espace des caractéristiques et d'identifier les caractéristiques les plus informatives pour les classifieurs spécifiés, ce qui peut potentiellement améliorer les performances du modèle et faciliter l'interprétation des résultats.

VI.4.6. Classification par apprentissage profond à l'aide de la boîte à outils Step2bci

Dans cette phase du processus de classification, nous nous penchons sur les l'apprentissage profond à l'aide notre bibliothèque personnalisée. Pour ce faire, nous commençons par importer les bibliothèques nécessaires :

```
import torch.optim
from deepproc.deepBCI import NetPytorch, TorchTrainer, TorchEvaluator, TorchPredictor
from deepproc.neuroDataset import CustomDataset
from deepproc.neuroDevice import DeviceSwitcher
```

Afin d'assurer la compatibilité avec les opérations de calcul effectuées lors des processus d'entraînement et d'inférence, les étiquettes cibles doivent être encodées. Dans notre cas, les étiquettes cibles sont représentées par [1, 2], PyTorch peut s'attendre à ce qu'elles soient encodées en indices à partir de zéro, tels que [0, 1]. Cela permet à PyTorch de traiter les étiquettes comme des variables catégorielles et d'effectuer des opérations telles que l'encodage one-hot ou les calculs softmax de manière plus efficace.

```
targets_T = targets_T - 1
targets_E = targets_E - 1
```

Ensuite, nous préparons nos données pour l'entraînement et l'évaluation. À cet effet, nous créons des instances de la classe *'CustomDataset'*, qui nous permettent d'encapsuler nos données d'entraînement (*pca_data_T*, *pca_data_E*) et de test avec leurs étiquettes correspondantes (*targets_T*, *targets_E*).

```
dataset_train = CustomDataset((pca_data_T, targets_T), transform=None, target_transform=None)
dataset_true = CustomDataset((pca_data_E, targets_E), transform=None, target_transform=None)
```

Voici les hyperparamètres utilisés lors de l'entraînement du modèle d'apprentissage profond :

```
n_features = pca_T.shape[1]
nclasses = len(np.unique(targets_T))
hidden_sizes = [32]
batch_size1 = len(targets_T) // 6
batch_size2 = len(targets_E) // 6
learning_rate = 0.15
dropout_prob = .5
n_epochs = 1000
momentum = .94
weight_decay = 0.01
```

Ces hyperparamètres sont définis tels que :

- *hidden_sizes :* représentent les tailles des couches cachées.
- *batch_size1* et *batch_size2 :* sont les tailles des lots d'entraînement et de test et souvent sont déterminées en fonction du nombre global d'échantillons.
- *learning_rate :* est le taux d'apprentissage qui contrôle les ajustements des poids.
- *dropout_prob :* a probabilité de désactivation aléatoire des neurones, il permet de réduire le sur-apprentissage (overfitting).

Le choix du dispositif pour l'entraînement et l'inférence est configuré en utilisant la classe *'DeviceSwitcher'*. Cela nous permet de spécifier sur quel périphérique les calculs seront exécutés.

```
device = DeviceSwitcher('cpu').device
```

Par la suite, nous définissons l'architecture du modèle d'apprentissage profond. Le modèle est créé à l'aide de la classe NetPytorch, qui prend en entrée le nombre de caractéristiques, le nombre de classes et la taille des couches cachées. La notation *hidden_sizes permet de passer un nombre variable d'arguments sous la forme d'un tuple au constructeur de *'NetPytorch'* du module *'deepBCI.py'*. En outre, la probabilité d'abandon est spécifiée à l'aide du paramètre *dropout_prob*. En définissant l'architecture du modèle, nous établissons la structure et les paramètres qui seront utilisés pendant les processus de formation et d'évaluation.

```
model = NetPytorch(n_features, nclasses, *hidden_sizes, dropout_prob=dropout_prob)
```

Dans la partie suivante du code, nous créons un objet *trainer* en instanciant la classe *'TorchTrainer'* du module *'deepBCI.py'*. Cet objet est configuré avec les hyperparamètres préalablement définis, ainsi que le modèle lui-même, les données d'entraînement, le périphérique

sur lequel les calculs seront effectués (dans ce cas, *'device'*), et l'optimiseur utilisé (dans ce cas, *'torch.optim.Adam'*).

```
trainer = TorchTrainer(model=model, train_data=dataset_train, batch_size=batch_size1, max_epochs=n_epochs,
                       learning_rate=learning_rate, momentum=momentum, weight_decay=weight_decay, device=device,
                       optimizer=torch.optim.Adam)
```

Cela nous permettra d'utiliser cet objet pour entraîner le modèle en utilisant les données fournies et les paramètres spécifiés.

```
trained_model, train_accuracy, epoch_losses, eval_losses = trainer.train(eval_train=True, patience=15,
                                    save_model=False)
```

Nous créons l'objet *evaluator* en instanciant la classe *TorchEvaluator* du module *'deepBCI.py'*. Cet objet est configuré avec le modèle déjà entraîné *trained_model*, la fonction de perte utilisée pour l'évaluation est *torch.nn.CrossEntropyLoss*, la taille du lot *batch_size2* et le périphérique sur lequel les calculs seront effectués *'device'*. La figure VI.13 représente le tracé l'évaluation des performances en terme de la fonction de perte pendant la phase d'apprentissage. Le graphique montre la perte d'entraînement et la perte de validation au fil des époques. La ligne verticale pointillée en rouge représente l'époque à laquelle l'arrêt précoce (*Early Stopping*) est appliqué.

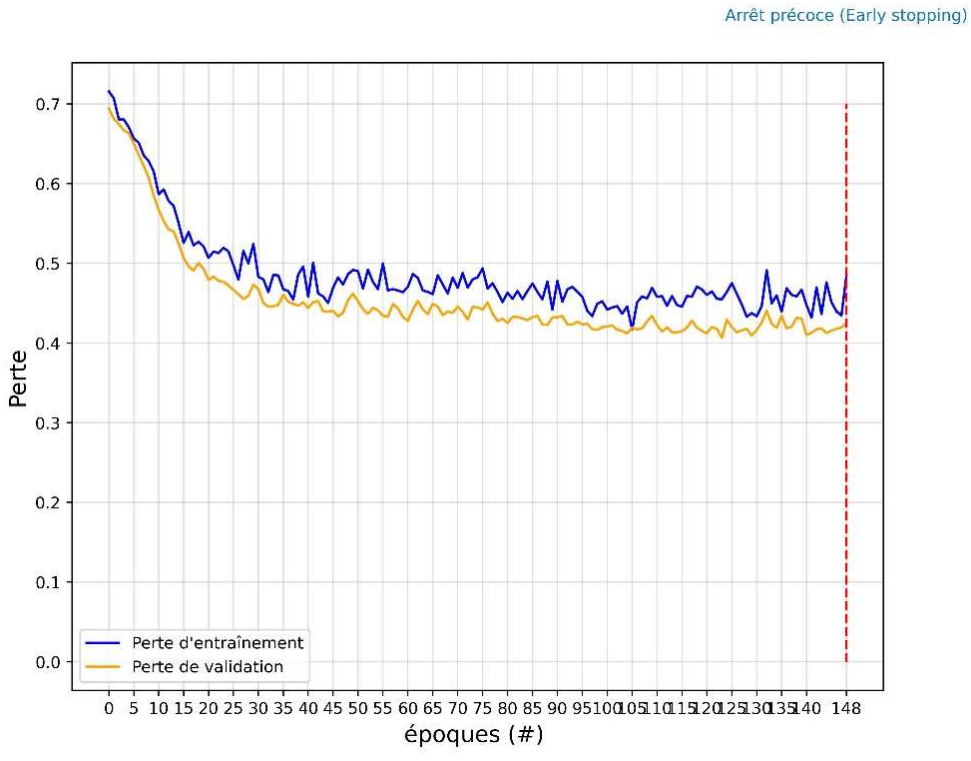


Figure VI.13. Évaluation des performances du modèle d'apprentissage profond en terme de la fonction de coût (perte) durant le processus d'apprentissage.

Ensuite, l'objet *evaluator* est utilisé pour évaluer les performances du modèle sur les données de validation *dataset_true*. L'évaluation est effectuée en calculant à la fois la précision et la perte sur l'ensemble d'évaluation (test). L'argument (*eval_train=False*) indique que l'évaluation n'est pas effectuée sur les données d'entraînement, mais uniquement sur les données de test.

```
evaluator = TorchEvaluator(model=trained_model, criterion=torch.nn.CrossEntropyLoss, batch_size=batch_size2,
                                                    device=device)

test_accuracy, test_loss = evaluator.evaluate(val_data=dataset_true, eval_train=False)
```

Ces lignes de code permettent donc d'évaluer les performances du modèle entraîné sur les données de validation et de récupérer la précision et la perte obtenues.

Finalement, nous définissons un objet *predictor* en instanciant la classe *'TorchPredictor'* du module *'deepBCI.py'*. Cet objet est configuré avec le modèle préalablement entraîné *trained_model*, la taille de l'entrée *input_size* correspondant au nombre de caractéristiques, et le périphérique sur lequel les calculs seront effectués *device*.

```
Output :
$ Epoch [149/1000] ... epoch loss : 0.4842588355143865 ... Précision sur
l'ensemble d'entraînement : 0.81173 ... validation loss : 0.42488717039426166
Arrêt précoce après 148 epochs Précision sur l'ensemble de validation :
0.76754
```

Une fois que l'objet *predictor* est créé, il peut être utilisé pour effectuer des prédictions sur l'ensemble de test. En utilisant le modèle préalablement entraîné *trained_model*, l'objet *predictor* est utilisé pour prédire les étiquettes de classe correspondantes aux données de test. Dans ce cas, les prédictions sont effectuées sur un lot de données comprenant les indices 20 à 49 (utilisés à titre d'exemple) de l'ensemble de test extrait de *dataset_true.data*. Les résultats des prédictions sont ensuite stockés dans la variable *predictions*, d'où les résultats de la prédiction sont donnés par :

```
Output :
$ tensor ([1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1,
0, 1, 0, 1, 1, 1, 1, 1])

$ tensor ([1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1,
1, 1, 0, 1, 1, 1, 1, 1])
```

VI.4.7. Interprétation et discussion des résultats

L'entraînement du modèle se fait sur 1000 époques. Après 149 époques, la précision sur l'ensemble d'entraînement est de 0,81173. Cela signifie que le modèle parvient à prédire

correctement environ 81% des exemples dans l'ensemble d'entraînement. La valeur de la perte durant l'entraînement est de 0,484.

Pour l'ensemble de validation, la précision obtenue est de 0,76754, c'est-à-dire le modèle peut prédire correctement environ 77% des exemples dans l'ensemble de validation. La perte de validation est de 0,425, ce qui indique l'erreur moyenne du modèle sur cet ensemble.

Nous remarquons que les précisions sur les ensembles d'entraînement et de test sont très proches depuis qu'il est préférable que la précision de ces deux ensembles ne diffère pas trop car un écart important entre la précision de ces deux ensembles peut être un indicateur de problèmes potentiels tels que le sur-apprentissage (overfitting) et la difficulté à généraliser. Il est donc souhaitable que ces deux mesures soient proches pour avoir confiance dans les performances réelles du modèle.

L'arrêt précoce est déclenché après 148 époques, ce qui conduira ç ce que l'apprentissage est arrêté prématurément car il n'y a pas eu d'amélioration significative des performances du modèle sur l'ensemble de validation au cours des 15 dernières époques. Les prédictions sont effectuées sur un sous-ensemble de l'ensemble de test (de taille égale à 30 échantillons). Les résultats des prédictions sont stockés dans la variable '*predictions'* sous forme de tenseurs, où chaque valeur du tenseur représente une étiquette de classe prédite (1 ou 0 dans ce cas).

En résumé, les résultats indiquent que le modèle a été entraîné avec une précision raisonnable sur l'ensemble d'entraînement et a obtenu des performances acceptables sur l'ensemble de validation.

VI.5. Conclusion

Ce chapitre a exploré l'application de la boîte à outils ***step2bci*** dans l'analyse et la classification des données EEG. L'objectif principal était de démontrer l'efficacité de ***step2bci*** dans les protocoles BCI hors ligne. En intégrant les packages développés précédemment, ce chapitre a présenté l'exécution fluide des différentes étapes de l'analyse des données EEG, allant de la préparation des données au prétraitement, au post-traitement et à la classification par apprentissage automatique et profond.

Une attention particulière a été accordée à l'implémentation du code, avec des explications détaillées sur des aspects essentiels tels que l'importation de modules, l'instanciation de classes, l'appel de fonctions, la manipulation des ensembles de données et la construction de pipelines complets pour l'extraction de caractéristiques et leur classification. Ces protocoles

pratiques nous permettrons d'utiliser efficacement ***step2bci*** dans nos propres projets d'analyse EEG et de réalisation des BCI.

Le chapitre a présenté des scénarios concrets en examinant différents algorithmes et en analysant les résultats correspondants. En utilisant des ensembles de données tiers provenant d'institutions réputées dans le domaine des interfaces cerveau-ordinateur, tels que l'ensemble de données Competition IV-Dataset 1 et l'ensemble de données Competition IV-Dataset 2b.

Les données ont été préparées et les caractéristiques ont été extraites. Les ensembles d'apprentissage et de test ont été créés, et les caractéristiques spectrales et temps-fréquence ont été calculées à l'aide du périodogramme de Welch et de la décomposition en ondelettes. Ensuite, la technique CSP a été appliquée pour maximiser la séparation entre les caractéristiques des différentes classes. Enfin, si nécessaire, une réduction de la dimensionnalité de l'espace des caractéristiques a été effectuée en utilisant soit l'analyse en composantes principales (PCA), soit la sélection séquentielle des caractéristiques (SFS).

Après avoir extrait les caractéristiques et effectué la réduction de dimensionnalité, nous avons procédé à la classification des données. Pour évaluer les performances des modèles d'apprentissage automatique ou d'apprentissage profond, nous avons utilisé plusieurs types de classifieurs tels que SVM, KNN, LDA, et d'autres encore. De plus, pour l'apprentissage profond, nous avons utilisé PyTorch.

Les résultats obtenus ont montré une meilleure discrimination entre les caractéristiques des différentes classes, ce qui facilite leur classification ultérieure. En utilisant step2bci, nous avons pu obtenir des informations précieuses sur les schémas d'activation corticale et l'implication du cortex moteur dans les tâches liées au mouvement.

BIBLIOGRAPHIE DU CHAPITRE VI

Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., ... & Zhang, J. D. (2020). An introduction to machine learning. *Clinical pharmacology & therapeutics*, *107*(4), 871-885.

Berlin BCI group: Berlin Institute of Technology (Machine Learning Laboratory). (n.d.), 2022, Retrieved from https://www.tu.berlin/en/ml

Blankertz, B., Dornhege, G., Krauledat, M., Müller, K. R., & Curio, G. (2007). The non-invasive Berlin Brain-Computer Interface: Fast acquisition of effective performance in untrained subjects. NeuroImage, 37(2), 539-550.

Fraunhofer FIRST. Artificial Intelligence, 2022, Retrieved from https://www.iais.fraunhofer.de /en/research/artificial-intelligence.html.

Institute of Neural Engineering Graz Brain-Computer Interface Lab. (2022). Retrieved from https://www.tugraz.at/institute/ine/home

Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., & Pfurtscheller, G. (2007). Brain–computer communication: motivation, aim, and impact of exploring a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *15*(4), 473-482.

Leeb, R., Brunner, C., Müller-Putz, G. R., Schlögl, A., & Pfurtscheller, G. (n.d.). BCI Competition 2008 - Graz data set B. Institute for Knowledge Discovery, Graz University of Technology, Austria, and Institute for Human-Computer Interfaces, Graz University of Technology, Austria.

Pfurtscheller, G. (2001). Functional brain imaging based on ERD/ERS. Vision Research, 41(10-11), 1257-1260. https://doi.org/10.1016/S0042-6989(00)00235-2.

Python: NumPy developers. (n.d.), numpy.ndarray, 2022, Retrieved from https://numpy.org/doc/stable/reference/generated /numpy.ndarray.html

Sazgar, M., Young, M. G., Sazgar, M., & Young, M. G. (2019). Overview of EEG, electrode placement, and montages. *Absolute Epilepsy and EEG Rotation Review: Essentials for Trainees*, 117-125.

Tariq, M., Trivailo, P. M., & Simic, M. (2020). Mu-Beta event-related (de) synchronization and EEG classification of left-right foot dorsiflexion kinaesthetic motor imagery for BCI. *Plos one*, *15*(3), e0230184.

CONCLUSION GENERALE

Cette recherche a été guidée par la question fondamentale de comprendre et d'explorer les Interfaces Cerveau-Ordinateur (BCI) sous divers aspects, allant de leur base neurologique à leur mise en œuvre pratique. Elle a exploré en profondeur le domaine des Interfaces Cerveau-Ordinateur (BCI), en commençant par une base solide dans l'anatomie et le fonctionnement du système nerveux. Nous avons examiné l'importance cruciale du cerveau en tant qu'organe central du système nerveux. Le cortex cérébral, en particulier, a été mis en lumière pour son rôle central dans la recherche sur les BCI, étant impliqué dans des fonctions motrices, sensorielles et cognitives.

Nous avons également plongé dans les BCI, révélant leur complexité et leur potentiel dans des domaines tels que la rééducation, la robotique, la réalité virtuelle et bien d'autres.

La conception d'une BCI a été expliquée en détail, soulignant l'importance de l'acquisition et du traitement des signaux EEG, ainsi que des différentes catégories de BCI en fonction de leurs caractéristiques et de leurs applications.

Un système d'acquisition EEG basé sur l'ADS1299-AFE a été développé, offrant une plateforme modulaire et extensible pour la recherche en BCI. Ce système a été validé rigoureusement et a démontré son efficacité et sa possibilité de collecter des données EEG de haute qualité.

De plus, l'introduction de la boîte à outils Step2BCI a offert un ensemble complet de ressources logicielles pour l'analyse et la classification des données EEG, renforçant ainsi l'arsenal de la recherche BCI.

Ces efforts ont abouti à la confirmation que les BCI représentent une avancée majeure dans l'interaction homme-machine.

Finalement, Step2BCI a été appliqué à des données EEG réelles, démontrant son efficacité dans des protocoles BCI hors ligne. Nous avons examiné la préparation des données, l'extraction de caractéristiques, la classification par apprentissage automatique et profond, montrant comment cette boîte à outils peut être utilisée pour analyser les données EEG dans diverses applications.

Dans le contexte de la recherche sur les Interfaces Cerveau-Ordinateur (BCI), cette étude s'inscrit dans la continuité des travaux antérieurs qui ont déjà mis en évidence le potentiel révolutionnaire de ces BCI. Toutefois, notre contribution se distingue par le développement de solutions concrètes, notamment le système d'acquisition EEG et la boîte à outils *Step2BCI*, ainsi que par la démonstration de leur efficacité pratique. Ces ressources viennent ainsi enrichir

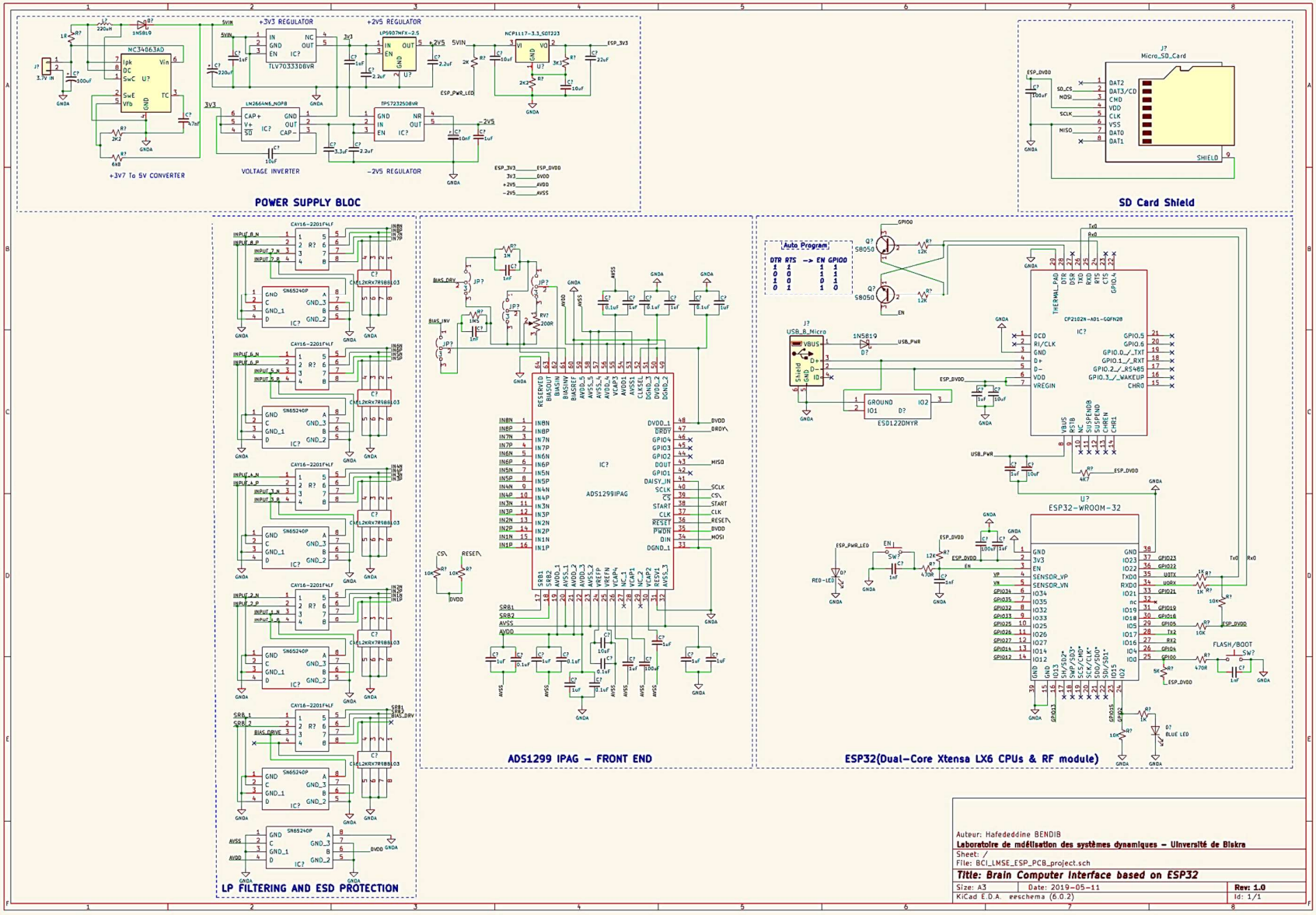
l'arsenal d'outils à disposition des chercheurs dans ce domaine en constante évolution, renforçant la base de connaissances existante et élargissant les ressources disponibles pour les futures recherches en BCI.

Certaines limites persistent malgré les accomplissements significatifs de cette recherche. Les méthodes de réduction du bruit, l'utilisation d'électrodes actives, le développement d'une interface utilisateur conviviale, l'évaluation approfondie de la consommation énergétique et l'exploration de divers paradigmes BCI nécessitent des améliorations continues. De plus, il est impératif de reconnaître que des défis subsistent. Bien que le système d'acquisition EEG ait été validé avec succès dans des conditions idéales, il nécessitera des tests plus approfondis dans des environnements variés pour évaluer sa robustesse et sa fiabilité. De même, la boîte à outils Step2BCI, bien que complète, pourrait nécessiter des améliorations futures pour répondre de manière optimale aux besoins spécifiques de la recherche BCI.

L'avenir de la recherche sur les BCI est prometteur. Des recherches futures pourraient se concentrer sur l'amélioration des méthodes de prétraitement des données EEG, l'exploration de modèles d'apprentissage profond plus avancés, la création d'interfaces utilisateur graphiques conviviales, l'évaluation approfondie de la consommation d'énergie, la validation clinique des BCI et la réflexion continue sur les implications éthiques et sociales de ces technologies.

En conclusion, cette recherche a contribué de manière significative à la compréhension et au développement des Interfaces Cerveau-Ordinateur. Elle a fourni des bases théoriques, des systèmes d'acquisition, des boîtes à outils et des applications pratiques qui renforceront la recherche future dans ce domaine en constante évolution. Les BCI représentent une avancée majeure pour l'interaction homme-machine et la réhabilitation, et cette recherche ouvre la voie à des innovations continues dans les années à venir. En continuant à explorer des axes de recherche tels que la réduction du bruit, l'efficacité énergétique, et l'élargissement des paradigmes BCI testés, cette étude jette les bases pour des développements futurs qui pourraient transformer radicalement notre interaction avec les machines grâce à la puissance du cerveau humain.

ANNEXE 1 Schémas électronique du système d'acquisition EEG conçu à l'aide du logiciel KiCad



ANNEXES 2

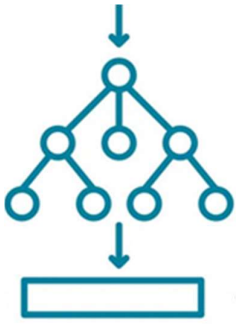
I) Algorithme des arbres de décision

Un arbre de décision est un modèle très simple. Etant donnée plusieurs caractéristiques, la décision se commence par un de ces caractéristiques ; si ce n'ai pas suffisant, on utilise une autre, ainsi de suite. Il est largement connu et utilisé dans de nombreuses entreprises pour faciliter le processus de prise de décision et l'analyse des risques. Il a été largement utilisé dans les années 1960-1980 pour la construction de systèmes experts. Les règles sont introduites manuellement, pour cette raison ce modèle a perdu sa popularité après les années 80. L'apparition des méthodes mathématiques pour construire les arbres de décision fait revenir ce modèle à la bataille des algorithmes de l'apprentissage automatique.

Il existe plusieurs algorithmes automatiques pour construire les arbres de décision, par exemple :

- **ID3** (Iterative Dichotomiser 3) : dévelopé en 1986 par Ross Quinlan. Il peut être appliqué seulement sur les caractéristiques nominales. Il est utilisé pour le classement.
- **CART** (Classification and Regression Trees)

L'algorithme général de création d'un arbre de décision :



a) Déterminer la meilleure caractéristique dans l'ensemble de données d'entrainement.
b) Diviser les données d'entrainement en sous-ensembles contenant les valeurs possibles de la meilleure caractéristique.
c) Générez de manière récursive de nouveaux arbres de décision en utilisant les sous-ensembles de données créés.
d) Lorsqu'on ne peut plus classifier les données, on s'arrête.

Sélection de la meilleure caractéristique

Cet algorithme utilise la fonction entropie et le gain d'information pour décider quelle est la meilleure caractéristique. Etant donnée un ensemble de classes C, l'entropie d'ensemble de donnée S est exprimée par :

$$H(S) = \sum_{c_i \in C} -P(c_i) log_2 P(c_i)$$

Où (en divisant le nombre des échantillons d'une certaine classe sur le nombre de tous les échantillons dans les données d'entrainement) :

$$P(c_i) = \frac{|C_i|}{|S|}$$

Etant donnée un vecteur de caractéristiques $\vec{\boldsymbol{f}}$, en utilisant les valuers d'une caractéristique $\boldsymbol{f_j}$, on peut diviser l'ensemble de donnée S en plusieurs sous-ensembles groupés dans un ensemble $\boldsymbol{S_j}$. Le gain d'information est mesuré en se basant sur la différence entre l'entropie originale de S et celle après sa division en se basant sur une caractéristique $\boldsymbol{f_j}$.

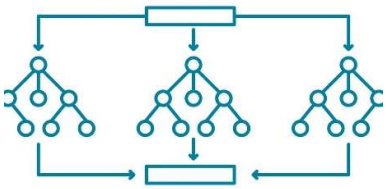
$$IG(S, f_j) = H(S) - \sum_{S_{jk} \in \boldsymbol{S_j}} P(S_{jk}) H(S_{jk})$$

Où :

$$P(c_i) = \frac{|S_{jk}|}{|S|}$$

La caractéristique ayant plus de gain d'information est celle sélectionnée comme meilleure. Aussi, la valeur avec entropie nulle est considérée comme feuille de l'arbre.

II) lgorithme des forêts aléatoires



On pousse la perturbation un peu plus loin : à chaque nœud de l'arbre, on sélectionne la variable parmi un échantillon des variables disponibles.
On procède ainsi :

1. On sélectionne (généralement avec remise) un échantillon des données de taille ***c***.
2. On construit un arbre ***b*** de la façon suivante :
   - On part, à la racine, avec toutes les observations de l'échantillon sélectionné en 1.
   - On sélectionne ***d*** variables parmi l'ensemble des variables disponibles.
   - Pour chacune des ***d*** variables sélectionnées en (2.), on calcule l'indice de Gini (ou un autre critère) pour chacune des divisions (ou combinaisons de modalités) possibles.
   - On choisit la variable et la division qui fait le plus descendre l'indice de Gini et on crée deux nouveaux nœuds sur cette base.
   - Pour chacun des deux nœuds créés, on répète les étapes ii à iv jusqu'à l'atteinte d'un certain critère d'arrêt.
3. On calcule une prévision pour chaque observation : $\hat{y}_i^{(b)}$
4. On répète les étapes 1 à 3 B fois

$$(b = (1,2\ldots, B)$$

5. On agrège les valeurs des B arbres (par exemple, on peut prendre :

$$\hat{y}_i = \frac{\sum_{b=1}^{B} \hat{y}_i^{(b)}}{B}$$

Les forêts aléatoires sont plus robustes que le bagging car le fait de choisir un échantillon des variables à chaque nœud entraîne une décorrélation des arbres. Ils sont donc en général plus performants tout en requérant moins de temps de calcul que le bagging.

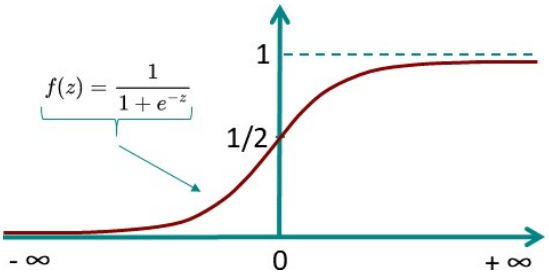
La construction d'une forêt aléatoire implique cependant la sélection d'un hyperparamètre de plus : ***d***, le nombre d'arbres que l'on sélectionne à chaque nœud.

Il est à noter que Breiman (2001)[A.1] a démontré la convergence presque sûre des forêts aléatoires: plus le nombre d'arbres augmente, plus on se dirige vers le meilleur modèle.

Référence (A.1) : *Breiman, Leo. 2001. "Random Forests." Machine Learning 45 (1): 5–32.*

III) Algorithme de la régression logistique

La régression logistique est une technique d'apprentissage supervisé utilisée pour la classification binaire (c'est-à-dire, les problèmes où la sortie est l'une des deux classes possibles). Elle estime la probabilité qu'une instance appartient à une classe particulière.



Principe

1. **Formule de base** :

La régression logistique utilise une fonction logistique (ou sigmoïde) pour transformer la sortie d'une combinaison linéaire des caractéristiques en une probabilité.

La fonction sigmoïde est définie par :

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Où z est une combinaison linéaire des caractéristiques :

$$z = \omega_0 + \omega_0\omega_0 + \omega_1 x_1 + \cdots + \omega_n\omega_n$$

La sortie p de la fonction sigmoïde est interprétée comme la probabilité que l'instance appartienne à la classe positive :

$$p = \sigma(z)$$

2. **Décision de classe**

- Si $p \geq 0.5$, l'instance est classée comme appartenant à la classe positive.
- Sinon, elle est classée comme appartenant à la classe négative.

3. **Fonction de coût (Log-vraisemblance négative)**

La fonction de coût utilisée pour ajuster les paramètres est la log-vraisemblance, souvent exprimée sous forme de perte logistique :

$$L(\omega) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

Où y_i est la véritable étiquette de l'instance i et p_i est la probabilité prédite.

La régression logistique Utilisée pour les problèmes de classification binaire où la variable cible est catégorielle (deux classes).

4. Décision et Classification :

Pour la classification, on choisit un seuil de probabilité (par exemple, 0,5). Si la probabilité prédite est supérieure au seuil, on classe l'observation dans la classe positive (1), sinon dans la classe négative (0).

IV) Algorithme Gaussian Naive Bayes

L'algorithme Gaussian Naive Bayes est un algorithme de classification probabiliste basé sur le théorème de Bayes, avec l'hypothèse d'indépendance naïve entre les caractéristiques. Il est particulièrement adapté aux problèmes de classification où les caractéristiques continues suivent une distribution normale (gaussienne).

5. Théorème de Bayes

Le théorème de Bayes permet de calculer la probabilité d'une classe ***C*** donnée un ensemble de caractéristiques ***x*** :

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

Où :

- **P(C|x)** est la probabilité que l'instance ***x*** appartienne à la classe ***C***. (probabilité a posteriori).
- **P(x|C)** est la probabilité d'observer ***x*** étant donné la classe ***C*** (vraisemblance).
- **P(C)** est la probabilité a priori de la classe ***C***.
- **P(x)** est la probabilité totale d'observer ***x***.

2. Hypothèse Naïve d'Indépendance

L'hypothèse naïve suppose que les caractéristiques sont conditionnellement indépendantes les unes des autres étant donné la classe ***C***. Ainsi, la vraisemblance ***P(x|C)*** se décompose en :

$$P(x|C) = P(x_1, x_2, \dots, x_n \mid C) = P(x_1|C).P(x_2|C) \cdots P(x_n|C)$$

3. Distribution Gaussienne

Pour le Gaussian Naive Bayes, on suppose que chaque caractéristique x_i suit une distribution normale (gaussienne) :

$$P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} e^{\left(-\frac{(x_i-\mu_C)^2}{2\sigma_C^2}\right)}$$

Où :

- μ_C est la moyenne des caractéristiques pour la classe ***C***.

- σ_C^2 est la variance des caractéristiques pour la classe ***C***.

4. Calcul des Paramètres

Pour chaque classe ***C***, on calcule :

- La probabilité a priori ***P(C)*** :

$$P(C) = \frac{Nombre\ instances\ dans\ la\ classe\ C}{Nombre\ total\ d'instances}$$

- La moyenne μ_C et la variance σ_C^2 pour chaque caractéristique $\boldsymbol{x_i}$:

$$\mu_{C,i} = \frac{1}{N_C} \sum_{j \in C} x_{j,i}$$

$$\sigma_{C,i}^2 = \frac{1}{N_C} \sum_{j \in C} \left(x_{j,i} - \mu_{C,i}\right)^2$$

Où N_C est le nombre d'instances dans la classe ***C***.

5. Classification

Pour classifier une nouvelle instance xxx, on calcule la probabilité a posteriori pour chaque classe CCC :

$$\mathrm{P(C|x)} \propto P(C) \prod_{i=1}^{n} P(x_i|C)$$

On choisit la classe ***C*** qui maximise cette probabilité.

V) Algorithme Multinomial Naive Bayes (MNB)

Le Multinomial Naive Bayes est un algorithme de classification adapté aux données discrètes telles que les comptes d'occurrences dans des ensembles de données textuelles ou autres. Lorsque nous appliquons cet algorithme aux données EEG pour l'activité motrice, nous considérons chaque segment de signal comme une observation avec des caractéristiques discrètes.

Principe de Base

Naive Bayes est basé sur l'application du théorème de Bayes avec une forte hypothèse d'indépendance (naïve) entre les caractéristiques. Dans le cas de Multinomial Naive Bayes, les caractéristiques sont discrètes et représentent généralement des comptes d'occurrences.

Théorème de Bayes

Le théorème de Bayes s'énonce comme suit :

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{P(x)}$$

Où :

- $P(C_k|x)$ est la probabilité que l'instance x appartienne à la classe C_k (probabilité a posteriori).
- $P(x|C_k)$ est la probabilité d'observer x étant donné la classe C_k (vraisemblance).
- $P(C_k)$ est la probabilité a priori de la classe C_k.
- $P(x)$ est la probabilité totale d'observer x.

Hypothèses de Multinomial Naive Bayes

- **Indépendance des caractéristiques :** Les caractéristiques (mots, termes) sont supposées indépendantes les unes des autres dans chaque classe.
- **Distribution multinomiale :** Les caractéristiques suivent une distribution multinomiale.

Étapes de l'algorithme

1. Prétraitement des Données

- Les données EEG sont segmentées en fenêtres de temps et des caractéristiques discrètes sont extraites de chaque fenêtre. Par exemple, les caractéristiques peuvent inclure des fréquences d'occurrences de certains motifs d'onde, amplitudes ou autres mesures pertinentes.
- Ces caractéristiques sont ensuite converties en une matrice de caractéristiques, où chaque ligne représente un segment de signal EEG et chaque colonne représente une caractéristique spécifique.

2. Calcul des Probabilités A Priori P(C_k)

$$P(C_k) = \frac{Nombre\ de\ segments\ dans\ la\ classe\ C_k}{Nombre\ total\ de\ segments}$$

Par exemple, si nous avons deux classes d'activités motrices : "Mouvement de la main" et "Mouvement du pied", nous calculons la proportion de segments appartenant à chaque classe.

3. Calcul des Probabilités Conditionnelles *P(x/C_k)* :

Pour chaque caractéristique x_i dans chaque classe C_k, nous calculons la probabilité conditionnelle :

$$P(x_i|C_k) = \frac{n_{ik} + \alpha}{N_k + \alpha N}$$

Où :

- n_{ik} est le nombre de fois que la caractéristique x_i apparaît dans les segments de la classe C_k.
- N est le nombre total de caractéristiques dans les segments de la classe C_k.
- N est le nombre total de caractéristiques dans toutes les classes.
- α est un paramètre de lissage pour éviter les zéros.

4. Calcul de la Probabilité de la Classe pour une Nouvelle Instance :

Pour un nouveau segment de signal EEG, représenté par le vecteur ***x***, la probabilité de chaque classe est calculée en multipliant les probabilités conditionnelles des caractéristiques et la probabilité a priori de la classe :

$$\mathrm{P}(\mathrm{C}_k|\mathrm{x}) \propto P(C_k) \prod_{i=1}^{n} P(x_i|C_k)^{x_i}$$

5. Prédiction :

La classe avec la probabilité totale la plus élevée est choisie comme prédiction pour le nouveau segment de signal EEG :

$$\hat{y} = \underset{C_k}{\operatorname{argmax}}\, P(C_k|x)$$

L'algorithme Multinomial Naive Bayes est particulièrement efficace pour les données discrètes. Lorsqu'il est appliqué aux données EEG pour la classification d'activités motrices, il permet de faire des prédictions précises en se basant sur la fréquence et la répartition des caractéristiques extraites des segments de signal.

ANNEXE 3

Forker et mettre à jour ou synchroniser un référentiel déjà forké sur GitHub

Cette approche offre plus de contrôle et est préférée par les développeurs habitués à la ligne de commande.

- **Étape 1** : Clonez votre référentiel forké (s'il n'est pas déjà cloné)

```
git clone https://github.com/BCIs-maniac81/Step2BCI.git

cd Step2BCI
```

- **Étape 2** : Ajoutez le référentiel d'origine en tant que référentiel distant appelé « Upstream »

```
git remote add upstream https://github.com/BCIs-maniac81/Step2BCI.git
```

- **Étape 3 :** Récupérer les modifications Upstream

```
git fetch upstream
```

- **Étape 4 :** Fusionner les modifications Upstream dans votre fork

- Basculez vers la branche principale (généralement « main » ou « master »).

```
git checkout master
```

- Fusionnez les modifications Upstream dans votre référentiel local.

```
git merge upstream/master
```

- **Étape 5 :** envoyez les mises à jour sur votre fork GitHub

```
git push origin main
```