

GIN 956 : RAPPORT DE PROJET DE
RECHERCHE ET DÉVELOPPEMENT

INTERFACE CERVEAU - ORDINATEUR
À PARTIR D'EEG

Étudiants :
Romain RICHARD
Florian LEPONT

Professeur responsable :
M. Jean ROUAT

Remerciements

Nous tenons tout d'abord à remercier l'université de Sherbrooke de nous avoir offert l'opportunité de réaliser ce projet lors de notre session de mobilité internationale.

Nous remercions tout particulièrement Jean Rouat de nous avoir encadré tout au long de ce projet, nous ayant permis d'étoffer nos connaissances dans le domaine des neurosciences et des interfaces cerveau-ordinateur.

Nous remercions également Alison Cellard pour le soutien technique qu'elle nous a apporté, ainsi que l'ensemble des membres du laboratoire NECOTIS pour leur accueil et leur sympathie au quotidien.

Sommaire

Introduction	7
1 Présentation du projet	9
1.1 Objectifs	9
1.2 Planning	9
2 Neurophysiologie	10
2.1 Cartographie du cerveau humain	10
2.1.1 Lobes externes	10
2.1.2 Lobes limbiques	11
2.2 Activité neuronale et potentiel d'action	12
2.3 Signaux EEG et rythmes du cerveau	12
3 Choix du matériel et du logiciel	14
3.1 Méthodes d'enregistrement d'un signal EEG	14
3.2 Caractéristiques générales du matériel	14
3.2.1 Électrodes	14
3.2.2 Amplificateur	15
3.3 Choix du matériel	16
3.4 Choix du logiciel	18
4 Traitement des signaux EEG	19
4.1 Caractéristiques des signaux étudiés	19
4.2 Différentes interfaces cerveau-ordinateur	20
4.3 Traitement et analyse des signaux EEG	22
4.3.1 Pré-traitement, méthodes d'optimisation des données	23
4.3.2 Méthodes d'extraction des caractéristiques	30
4.3.3 Méthodes de classification	31
5 Utilisation du casque EPOC et d'OpenViBE	35

5.1	OpenViBE	35
5.1.1	Généralités sur le fonctionnement d'OpenViBE	35
5.1.2	Installation d'OpenViBE	35
5.1.3	Utilisation d'OpenViBE designer	36
5.1.4	Description des blocs utilisés	37
5.1.5	Utilisation du serveur d'acquisition d'OpenViBE	39
5.2	Utilisation du casque EPOC	39
5.2.1	Généralités sur le fonctionnement du casque	39
5.2.2	Installation de Emotiv Xavier Control Panel Premium et du SDK. .	40
5.2.3	Préparation du casque	40
5.2.4	Conseils sur l'utilisation du casque	40
5.2.5	Vérification de la qualité des signaux	41
6	Expérience simple d'interaction	42
6.1	Présentation de l'expérience	42
6.2	Développement des scénarios sous OpenViBE	42
6.2.1	Scénario 1 : Acquisition des données pour apprentissage	42
6.2.2	Scénario 2 : Entrainement du filtre spatial CSP	43
6.2.3	Scénario 3 : Entrainement du classificateur	46
6.2.4	Scénario 4 : Utilisation en temps réel de l'interface	48
6.3	Résultats obtenus et analyse	49
7	Intégration du projet dans le cadre de cours	54
7.1	Choix des outils mathématiques d'apprentissage	54
7.2	Cours de neurosciences computationnelles	55
7.2.1	Résumé du travail pratique proposé	55
7.2.2	Préparation du travail pratique par le professeur	55
7.3	Cours de traitement de signal avancé	58
7.3.1	Résumé du travail pratique proposé	58
7.3.2	Développement de nouveaux scénarios	58
7.3.3	Fonctions Matlab et corrections	59
7.3.4	Préparation du travail pratique par le professeur	60
8	Bilan du projet	62
8.1	Connaissances acquises	62
8.2	Problèmes rencontrés	62
8.3	Pistes d'amélioration et poursuite de nos travaux	63
Conclusion		64
Annexes		66

Table des figures

1.1	Planning de Gantt prévisionnel	9
2.1	Les deux hémisphères du cerveau	11
2.2	Représentation des différents lobes du cerveau	11
2.3	Système limbique du cerveau humain.	12
2.4	Structure d'une cellule neuronale	13
3.1	Chaine d'acquisition d'un signal EEG	14
3.2	Standard International 10-20 System à 23 électrodes.	15
3.3	CAP100C de Biopac	16
3.4	GRASS amplifier 15LT	16
3.5	Comparaison entre les différents casques Emotiv	17
3.6	EPOC headset - Emotiv	18
4.1	Signaux des différents rythmes du cerveau	19
4.2	Représentation des phénomènes d'oscillations ERS et ERD du rythme bêta du cerveau	20
4.3	Signal EEG de type SSVEP, en réponse à une stimulation visuelle de 15Hz .	21
4.4	Réponse P300	21
4.5	Principales étapes constituant une interface cerveau-ordinateur.	22
4.6	Chaine de traitement d'un signal EEG	22
4.7	Chaine de traitement de plusieurs signaux EEG	23
4.8	Bruit du flux de donnée X et Y	25
4.9	Redondances possibles dans les données dans deux flux de données R1 et R2.	25
4.10	ICA et principe du BSS	27
4.11	Décomposition en valeurs singulières	28
4.12	STFT du signal $f(t)$, avec fenêtrage par le signal $g(t)$	31
4.13	Classification LDA	32
4.14	Principe de LDA 1	33
4.15	Principe de LDA 2	33

5.1	Interface de contrôle de la simulation sous OpenViBE.	36
5.2	Interface du répertoire des blocs OpenViBE.	36
5.3	Espace de travail d'OpenViBE.	37
5.4	Lancement du serveur d'acquisition	39
5.5	Serveur démarré	40
5.6	Qualité du signal provenant des électrodes	41
6.1	Stimulation visuelle pour l'apprentissage.	43
6.2	Scénario 1 : Acquisition.	44
6.3	Scénario 2 : Entrainement du filtre spatial	45
6.4	Sortie console de l'entraînement du CSP	46
6.5	Sortie console résultant de l'entraînement du classificateur.	48
6.6	Scénario 3 : Entrainement du classificateur	50
6.7	Scénario 4 : Utilisation en temps réel	51
6.8	Signaux bruts observés en sortie du casque Emotiv	52
6.9	Signaux filtrés	52
6.10	Réduction du nombre de sorties grâce au filtrage CSP	52
6.11	Analyse spectrale des signaux	52
6.12	Sortie visuelle temps réel après classification	53
7.1	Scénario temps réel proposé dans le cadre du cours de neurosciences computationnelles.	56
7.2	Interface de visualisation proposée aux étudiants du cours de neurosciences	57
7.3	Scénario d'acquisition avec sortie au format CSV	58
7.4	Scénario d'entraînement du classificateur à partir de fichiers .csv	59
7.5	Scénario d'observation à partir de fichiers .csv issus de Matlab	60
8.1	Traitemennt des signaux EEG sous OpenViBE.	77
8.2	Interface de contrôle de la simulation sous OpenViBE.	78
8.3	Interface du répertoire des blocs OpenViBE.	79
8.4	Espace de travail d'OpenViBE.	79
8.5	Traitemennt des signaux EEG sous OpenViBE.	83
8.6	Interface de contrôle de la simulation sous OpenViBE.	84
8.7	Interface du répertoire des blocs OpenViBE.	84
8.8	Espace de travail d'OpenViBE.	85

Introduction

Une interface Cerveau-Ordinateur ou BCI (Brain Computer Interface) permet de réaliser une communication allant du cerveau vers un système numérique (ordinateur par exemple). Pour cela, on s'appuie sur une étude de l'activité neuronale du cerveau grâce à un système électroencéphalographique (EEG) équipé d'électrodes. Celles-ci sont placées à la surface du crâne afin de capter l'activité électrique du cerveau. Un traitement et une analyse des signaux électriques saisis sont réalisés afin de les traduire en informations exploitables par les systèmes numériques.

Il existe de nombreuses applications possibles pour ce type de technologie :

- **Recherche.** L'étude du cerveau et son activité neuronale, afin d'en comprendre les différents mécanismes de fonctionnement.
- **Médicale.** Ce dispositif permet d'aider à comprendre et à guérir certaines maladies (Parkinson, etc.). Il peut également assister les médecins dans le traitement de l'hyperactivité chez certaines personnes, en améliorant leur concentration. Enfin, ce système permet le contrôle de prothèses robotisées pour les personnes amputées.
- **Vidéo-ludique.** Les liaisons cerveau-ordinateur sont de plus en plus utilisées dans l'univers des jeux-videos.

Le but de cette étude est de comprendre le fonctionnement d'une liaison Cerveau-Ordinateur et d'appliquer les concepts retenus grâce à la réalisation d'une expérience vidéo-ludique simple (interagir avec un ordinateur). Ce travail est réalisé dans le cadre de notre projet de recherche et de développement (GIN956), lors de la session d'hiver 2015.

La première étape de notre démarche est donc d'étudier la physiologie du cerveau et les principes biologiques qui y sont rattachés. On examinera par la suite les différents systèmes EEG présents sur le marché et on choisira le plus approprié. Dans un troisième temps, on propose d'explorer les différentes méthodes existantes de traitement et d'analyse des signaux EEG. Afin de mettre en application les informations acquises, on réalisera une expérience simple d'interaction. Enfin, on propose d'intégrer nos travaux dans le cadre de cours dispensés à l'université de Sherbrooke. Il s'agit de soumettre des travaux pratiques en lien avec les outils que nous utilisons dans le cadre de notre projet, afin de donner un aspect plus concret à ces cours. Voici les cours concernés.

- **Cours gradué de techniques avancées en traitement des signaux.** Ce cours enseigne les différentes méthodes de représentation d'un signal dans le domaine spectral (transformée de Fourier, transformation en ondelettes, etc.), ainsi que les outils de filtrage spectral (analyse en composantes principales (PCA), analyse en composantes indépendantes (ICA)). Cours enseigné à la session d'automne.
- **APP d'intelligence probabiliste.** Ce cours s'intéresse aux différentes méthodes de classification, notamment les méthodes paramétriques (Bayesiennes) et non pa-

ramétriques (k-moyen, SVM, etc.). Cours enseigné à la session d'automne.

- **Cours de neurosciences computationnelles.** Ce cours enseigne les principes de base de la physiologie des réseaux neuronaux et leur modélisation. Il offre également une vision d'ensemble des différents domaines d'applications. Cours enseigné à la session d'hiver.

Chapitre 1

Présentation du projet

1.1 Objectifs

Les objectifs demandés pour ce projet de spécialisation sont multiples :

- Choisir la combinaison équipement/logiciel la plus efficiente en termes de coût, d'efficacité du traitement des informations en temps réel et de la simplicité de la prise en main des outils.
- Utiliser l'équipement et les logiciels choisis pour monter une expérience simple d'interaction entre l'étudiant et l'ordinateur.
- Regarder avec l'assistance du professeur responsable de quelle façon intégrer cette expérience dans une problématique de la session en S8 (en limitant le plus possible la complexité).
- Regarder avec l'assistance du professeur responsable de quelle manière intégrer cette expérience dans le cadre de cours.

1.2 Planning

Voici le planning de Gantt prévisionnel de notre projet :

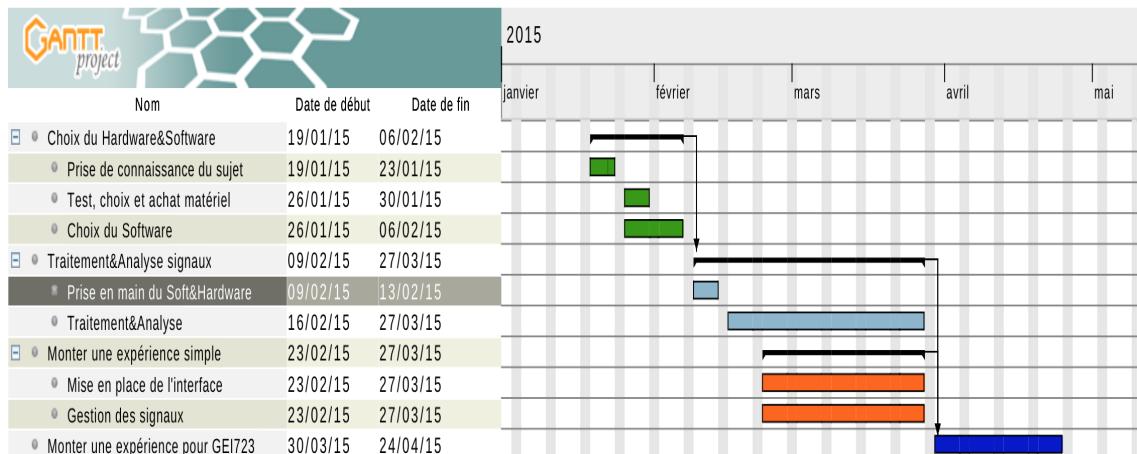


FIGURE 1.1 – Planning de Gantt prévisionnel

Chapitre 2

Neurophysiologie

Le système nerveux est un ensemble de structures complexes permettant à un animal d'interagir avec les différentes parties de son corps, ainsi qu'avec son environnement extérieur. Celui de l'Homme est composé de deux structures :

- Le Système Nerveux Central (SNC). Centre de décision, Il est composé du cerveau et de la moelle épinière.
- Le Système Nerveux Périphérique (SNP). Il est constitué de nerfs et de ganglions, qui ont pour rôle de connecter les muscles ainsi que les structures éfferentes au Système Nerveux Central (SNC). Il est également composé de centres de décision autonomes, on parle alors de système nerveux autonomes (exemple : le système nerveux sympathique gérant de nombreuses activités dites inconscientes, comme le rythme cardiaque).

Nous nous intéressons plus particulièrement au SNC, les interfaces cerveau-ordinateur étant effectuées à partir de l'étude des signaux produits par le cerveau.

2.1 Cartographie du cerveau humain

Le cerveau humain est constitué de 2 éléments : l'aire corticale (elle-même composée des lobes externes) et une partie interne, (composée des lobes limbiques) [?].

2.1.1 Lobes externes

Le cortex cérébral est constitué de deux hémisphères : un hémisphère droit et un hémisphère gauche (souvent appelé cerveau droit et cerveau gauche) contrôlant l'ensemble de nos fonctions cognitives : mouvements volontaires, pensée, mémoire, etc. D'une manière générale, l'hémisphère droit commande le côté gauche du corps et inversement. (Figure 2.1)

Chaque hémisphère est partagé en quatre lobes : le lobe frontal, le lobe pariétal, le lobe temporal et le lobe occipital (Figure 2.2).

Chaque lobe gère différentes fonctions :

- Les lobes frontaux : parole et langage, raisonnement, mémoire, prise de décision, personnalité, jugement, mouvements.
- Les lobes pariétaux : lecture, repérage dans l'espace, sensibilité.
- Les lobes occipitaux : vision.

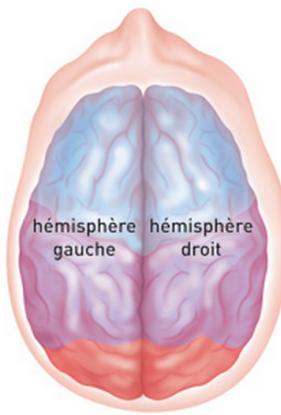


FIGURE 2.1 – Les deux hémisphères du cerveau.

Source : <http://intelligences.livehost.fr/>

- Les lobes temporaux : langage, mémoire, émotions.

Le cortex cérébral est l'élément du cerveau humain qui le différencie des autres espèces animales. Elle regroupe 75% des 100 milliards de neurones que possède l'être humain.

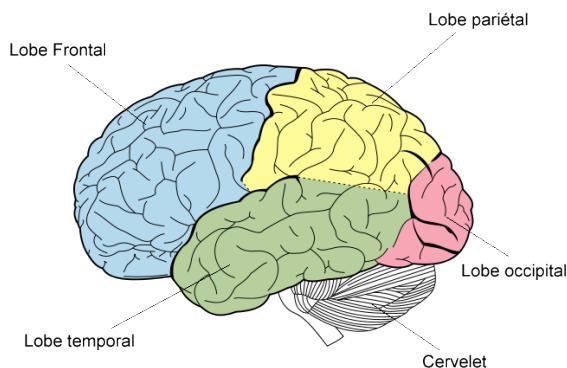


FIGURE 2.2 – Représentation des différents lobes du cerveau

Source : http://commons.wikimedia.org/wiki/File:Lobes_of_the_brain_NL.svg?uselang=fr

2.1.2 Lobes limbiques

Les lobes limbiques sont les éléments du cerveau qui ne sont visibles que lorsque l'on en réalise une coupe. (Figure 2.3) En voici la liste, ainsi que leurs principales fonctions :

- **Tronc cérébral.** Responsable des rythmes biologiques (respiration, rythme cardiaque, etc.).
- **Hippocampe.** Mémoire et navigation spatiale.
- **Amygdale.** Régulation des émotions et apprentissage.
- **Thalamus.** Relais et intégration des informations sensorielles (touché, vision, audition, goût).
- **Ganglions de la base.** Jouent un rôle dans le contrôle des mouvements.
- **Hypothalamus.** Responsable de l'équilibre intérieur, c'est-à-dire les sensations de soif, de faim, la température corporelle, la sécrétion d'hormones, etc.

- **Circonvolution cingulaire.** Ajuste certains paramètres corporels en fonction des émotions (pression sanguine, taille des pupilles, etc.)
- **Corps calleux.** Relie les deux hémisphères du cortex cérébral.
- **Cervelet.** Joue un rôle essentiel dans le contrôle moteur.

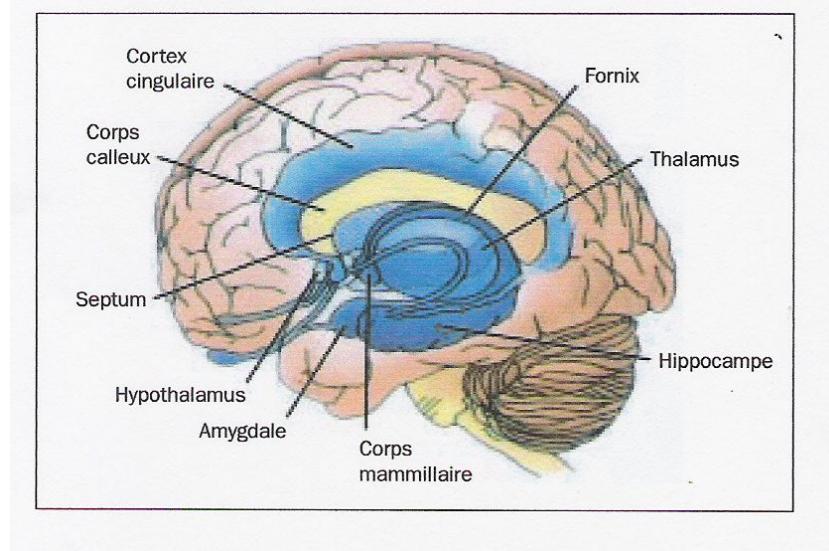


FIGURE 2.3 – Système limbique du cerveau humain.

Source : <http://lucebarrault-psych.wifeo.com/article-73000-le-systeme-emotionnel.html>

2.2 Activité neuronale et potentiel d'action

Le système nerveux est constitué de cellules nerveuses et de cellules gliales, situées entre les neurones. Chaque cellule nerveuse est composée de trois éléments : l'arbre dendritique, la synapse et le corps cellulaire (ou soma) (Figure 2.4). Celles-ci répondent à des stimuli et envoient des informations vers d'autres neurones. L'axone est un long cylindre transmettant des impulsions électriques. Chaque dendrite est connectée aux axones ou aux dendrites d'autres neurones (environ 10 000), formant ainsi un réseau neuronal. La zone de connexion entre une dendrite et un axone est appelée la synapse. Le neurone post-synaptique reçoit alors de la part du neurone pré-synaptique une impulsion, puis la relaie à d'autres cellules nerveuses. L'activité neuronale du cerveau est en partie liée aux courants électriques échangés au niveau des synapses. Sans activité synaptique, on mesure un potentiel membranaire de -70mV au niveau du neurone, on dit qu'il est au repos. Ce potentiel membranaire change en fonction de l'activité des synapses. Si celui-ci dépasse un certain seuil, le neurone va alors émettre un potentiel d'action vers un neurone post-synaptique [?].

2.3 Signaux EEG et rythmes du cerveau

Un signal électroencéphalographique (ou signal EEG) correspond au champ électrique créé lors des échanges de courant entre les différents neurones du cerveau, au niveau des synapses. Celui-ci est mesurable au niveau de la boîte crânienne grâce à des électrodes. Lors d'un EEG, on observe plusieurs signaux caractéristiques (ou ondes cérébrales). Elles correspondent à un état cognitif particulier du cerveau, i.e. l'activité physique ou intellectuelle de l'individu. Elles sont décomposées en fonction de leur domaine fréquentiel [?] :

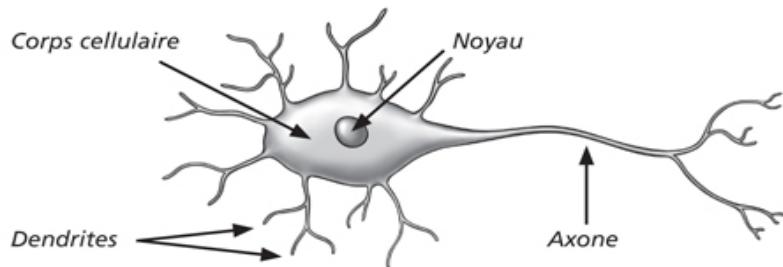


FIGURE 2.4 – Structure d'une cellule neuronale

Source : <http://www.thierrysouccar.com/bien-etre/info/comment-fonctionne-le-cerveau-386>

- **<4 Hz : Ondes Delta.** Elles sont généralement associées à un sommeil profond ou à certains cycles du sommeil.
- **4-8 Hz : Ondes Thêta.** Ces ondes sont la plupart du temps liées à l'état de somnolence d'un individu, la méditation profonde, la créativité et dans un cadre plus général à tout ce qui se rapporte à l'inconscience. C'est pour cela que le changement de rythme des ondes thêta est souvent observé lorsque l'on étudie l'activité émotionnelle d'une personne. Ces ondes proviennent la plupart du temps de l'activité du thalamus.
- **9-14 Hz : Ondes Alpha.** Elles caractérisent un état de conscience dans lequel un individu est dans une phase de "relaxation". Se sont les ondes les plus courantes dans le cerveau. Elles proviennent en général de la région occipitale du cortex.
- **15-30 Hz : Ondes Bêta.** Ces ondes sont liées à l'état d'éveil du cerveau, notamment lorsqu'une personne est dans un état de réflexion ou d'attention important. Ces signaux sont en général observés dans la région frontale et centrale du cortex.
- **>30 Hz : Ondes Gamma.** Elles sont associées à certaines actions motrices du corps humain et sont généralement observées au niveau de la zone frontale du cortex.

Dans le cadre de notre projet, nous nous intéresserons essentiellement aux ondes alpha et bêta.

Chapitre 3

Choix du matériel et du logiciel

3.1 Méthodes d'enregistrement d'un signal EEG

Il existe deux types de systèmes électroencéphalographiques :

- **Les EEG intracrâniens.** Il s'agit de systèmes dits invasifs : on vient enregistrer l'activité neuronale du cerveau en implantant des électrodes directement dans la boîte crânienne. Bien que cette technique offre des résultats probants, elle nécessite une intervention chirurgicale, ce qui n'est pas réalisable dans le cadre de notre projet.
- **Les EEG de scalpe (ou de surface).** Ce système étudie les variations du courant diffusé par le cerveau sur la surface du crâne. Il s'agit d'un système non invasif et c'est en toute logique que nous utiliserons ce type de dispositif.

La visualisation et l'enregistrement d'un signal EEG passe donc par l'utilisation d'électrodes. Il s'agit de capteurs sensibles aux variations du champ électrique générées par les échanges électriques entre les neurones. Le signal EEG est atténué par des éléments physiques comme le crâne ou la peau, il doit donc être amplifié afin d'être visualisé et traité. Du bruit est également observé lors de l'acquisition de signaux EEG. Celui-ci provient à la fois du cerveau et de l'environnement extérieur.

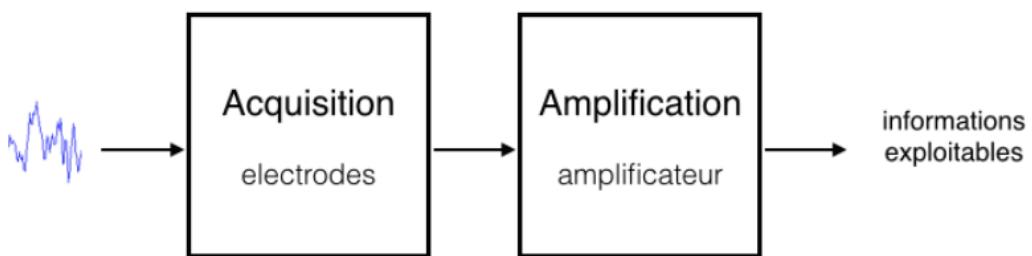


FIGURE 3.1 – Chaîne d'acquisition d'un signal EEG

3.2 Caractéristiques générales du matériel

3.2.1 Électrodes

Il existe différents types d'électrodes :

- **Les électrodes jetables.**

- **Les électrodes sèches.** Se sont les plus simples à utiliser car elles ne nécessitent pas l'utilisation de gel pour réaliser le contact avec la peau. Elles sont cependant moins précises que des électrodes à gel.
- **Les électrodes à gel.** Ce type d'électrode utilise un gel afin d'améliorer le contact entre l'électrode et la surface du crâne.
- **Les électrodes réutilisables.** Elles sont constituées de matériaux conducteurs permettant de capter les courants présents. Différents matériaux peuvent être utilisés (or, acier inoxydable, argent, etc.), chacun possédant une qualité de conduction plus ou moins élevée.
- **Les casques à électrodes.** Ce sont les dispositifs les plus utilisés en R&D. Ils offrent une stabilité accrue vis à vis du placement des électrodes et évitent ainsi l'introduction de bruits.
- **Les électrodes à aiguilles.** Il s'agit d'électrodes possédant une partie sous-cutanée. Elles offrent de bons résultats mais la pose nécessite l'assistance d'un médecin.

Après étude des avantages et des inconvénients des différents systèmes, notre choix se porte sur l'utilisation d'un casque à électrodes.

Plus le nombre d'électrodes utilisées est élevé, plus la qualité de l'information spatiale sera importante. Le placement des électrodes de surface au niveau de la boîte crânienne répond au standard "International 10-20 System" (Figure 3.2). Chaque zone du cerveau est donc scrutée par une électrode. On parle alors de canal. (exemple : O1 correspond au canal analysant par la partie occipitale gauche du cortex, O2 la partie droite).

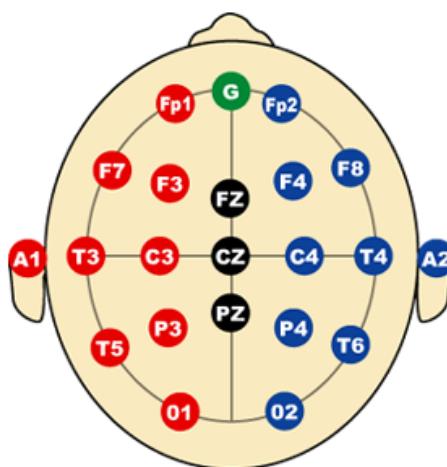


FIGURE 3.2 – Standard International 10-20 System à 23 électrodes

Source : <https://tel.archives-ouvertes.fr/tel-00822833/document>

3.2.2 Amplificateur

L'amplificateur effectue deux tâches : l'amplification des signaux EEG et leur échantillonnage. Certains amplificateurs proposent également de réaliser un filtrage temporel. Nous souhaitons cependant l'implémenter par nos propres moyens dans ce projet. Un amplificateur est caractérisé par :

- Son gain.
- Son taux d'échantillonnage.
- Sa résolution numérique.
- Son ratio signal sur bruit.

Il existe des amplificateurs directement intégrés dans le casque EEG, cependant leur qualité reste moindre que les systèmes conventionnels. Leur usage est principalement destiné aux jeux vidéos et n'est pas adapté à la réalisation d'examens médicaux.

3.3 Choix du matériel

Pour ce projet, nous disposons du matériel suivant :

- Le casque biopac CAP100C, composé de 19 électrodes à gel [?] (Figure 3.3).
- L'amplificateur 15LT de Grass Technologies. Il s'agit d'un amplificateur de 16 canaux. Il est utilisable à la fois pour les EEG, les EMG et les ECG. Celui-ci est fourni avec un logiciel propriétaire, GrassLab, permettant l'acquisition et l'exportation des données [?] (Figure 3.4).



FIGURE 3.3 – CAP100C de Biopac.

Source : <http://www.biopac.com/BioNomadix-EEG-CAP-medium>



FIGURE 3.4 – GRASS amplifier 15LT.

Source : <http://www.grasstechnologies.com/products/ampsystems/15lt.html>

Pour commencer, nous avons utilisé le CAP100C de Biopac avec l'amplificateur GRASS. Nous avons pu observer les différents signaux EEG (concentration, calcul mental simple, repos, lecture, etc). Notre objectif étant de réaliser une interface cerveau-ordinateur dans un cadre pédagogique, ce matériel ne semble pas adapté pour plusieurs raisons :

- Casque et amplificateur Grass onéreux (difficile d'en avoir plusieurs exemplaires)
- Logiciel propriétaire GrassLab obsolète, le support technique n'est plus assuré.
- Confort d'utilisation du casque discutable.

- Le produit s'adresse essentiellement au marché médical.

On souhaite acquérir un autre système EEG plus adapté à l'utilisation dans le cadre d'un cours. Une étude comparative de différents produits du marché est proposée (Voir le tableau présenté en annexe). D'après l'analyse comparative, notre choix se porte sur la gamme de produit de la marque Emotiv. En effet, le prix des produits est attractif pour les performances proposées, et est cohérent avec notre budget. Voici un tableau qui présente les principales différences entre les 3 casques de la marque.

FEATURE	EPOC+	EPOC	INSIGHT
SENSOR COUNT	14+2 references	14+2 references	5+2 references
SENSORS	AF3, AF4, F3, F4, FC5, FC6, F7, F8, T7, T8, P7, P8, O1, O2	AF3, AF4, F3, F4, FC5, FC6, F7, F8, T7, T8, P7, P8, O1, O2	AF3, AF4, T7, T8, Pz
REFERENCES	Left/Right Mastoid Process / Alt P3-P4	Left/Right Mastoid Process / Alt P3-P4	Left Mastoid Process
SAMPLE RATE	2048 internal, filtered and downsampled to 128 or 256 per sec per channel	2048 internal, filtered and downsampled to 128 per sec	2048 internal, filtered and downsampled to 128 or 256 per sec per channel
CONTACT QUALITY	Realtime CQ monitor (patented)	Realtime CQ monitor (patented)	Realtime CQ monitor (patented)
FREQUENCY RESPONSE	0.16 - 43 Hz	0.16 - 43 Hz	0.5 - 43 Hz
RESOLUTION	14 bit or 16 bit per channel	14 bit per channel	14 bit or 16 bit per channel
LSB RESOLUTION	0.51 µV @ 14 bit / 0.13 uV @ 16 bit	0.51 µV	0.51 µV @ 14 bit / 0.31 uV @ 16 bit
DYNAMIC RANGE	±4.17 mV	±4.17 mV	±4.17 mV
SENSOR TECHNOLOGY	Saline soaked felt pads	Saline soaked felt pads	Long life semi-dry polymer
MONITOR SENSOR	9 axis sensor (3x gyro, 3x accelerometer, 3x magnetometer)	2 axis gyro (nod, head shake)	9 axis sensor (3x gyro, 3x accelerometer, 3x magnetometer)
CONNECTIVITY	Proprietary 2.4GHz wireless (custom USB receiver) Bluetooth® Smart (wired USB available only using Extender accessory)	Proprietary 2.4GHz wireless (custom USB receiver)	Proprietary 2.4GHz wireless (custom USB receiver) Bluetooth® Smart (wired USB available only using Insight Extender isolation)

FIGURE 3.5 – Comparaison entre les différents casques Emotiv
Source : <http://emotiv.com>

Le casque Emotiv INSIGHT n'est pas assez performant pour nos travaux et convient plus pour des activités vidéoludiques (il ne dispose que de 5 capteurs). En revanche, les casques EPOC et EPOC+ semblent plus adaptés [?]. Il est possible d'accéder aux données EEG brutes, à condition d'acquérir une licence SDK. Après étude des produits, nous proposons à l'université de Sherbrooke d'acquérir le casque EPOC. Voici le devis :

- Casque : 399\$US ;
- SDK "licence de recherche individuelle" : 300\$US.
- Licence supplémentaire Mac OS X : 90 \$US.
- Total : 789\$US.



FIGURE 3.6 – EPOC headset - Emotiv.

Source : <http://emotiv.com>

3.4 Choix du logiciel

Il existe principalement deux logiciels libres de droit, compatibles avec le casque EPOC : EEGLAB et OpenViBE (Open Virtual Brain Environment).

EEGLAB fonctionne avec l'outil Simulink de Matlab. Il est majoritairement utilisé pour le traitement du signal. Il nécessite cependant l'acquisition d'une licence Matlab pour chaque poste de travail, ce qui peut s'avérer onéreux dans le cadre d'un cours suivi par plusieurs étudiants.

OpenVIBE propose quant à lui sa propre interface (i.e. fonctionne sans logiciel tiers). Il est majoritairement utilisé dans le développement de liaisons cerveau-ordinateur. Celui-ci est gratuit, fonctionnel avec le casque EPOC, bien documenté et performant. Il fonctionne de plus sur les plateformes Windows et Linux. Ce logiciel a été développé par l'équipe de l'Institut National de Recherche en Informatique et Automatique (INRIA) de Rennes en France.

Chapitre 4

Traitements des signaux EEG

Le schéma conceptuel en annexe présente les différentes méthodes de traitement et d'analyse des signaux EEG, ainsi que les différentes interfaces cerveau-ordinateur existantes.

4.1 Caractéristiques des signaux étudiés

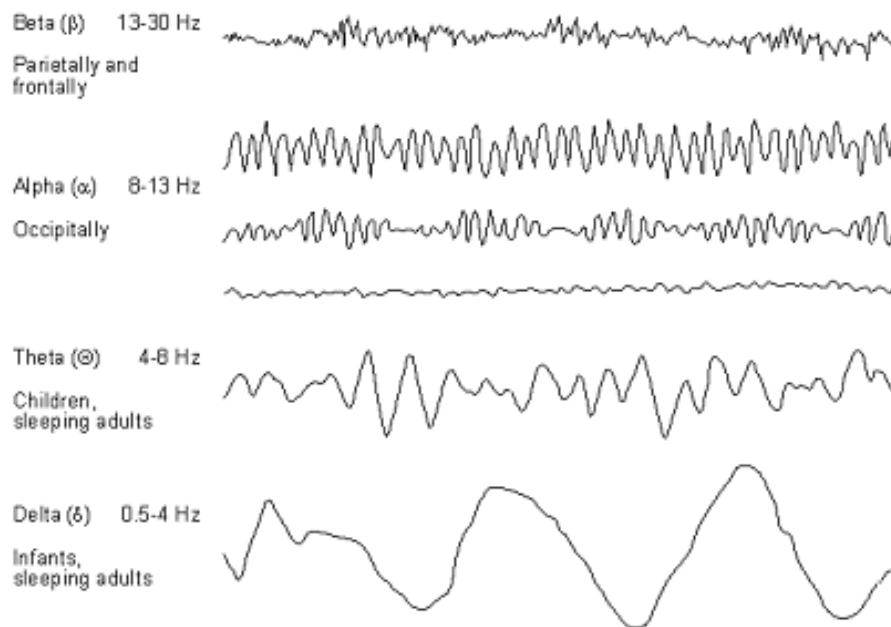


FIGURE 4.1 – Signaux des différents rythmes du cerveau.

Source : <http://www.computer.org/csdl/proceedings/bibe/2012/4357/00/06399691.pdf>

A partir des rythmes du cerveau, il est possible d'extraire différentes informations, dont les 3 plus utilisées sont :

- **Information spectrale.** Variations la puissance du signal dans certaines bandes de fréquences.
- **Information temporelle.** Variations du signal en fonction du temps.
- **Information spatiale.** Position d'une source. Pour obtenir cette information, on doit faire appel à plusieurs signaux EEG, en utilisant plusieurs électrodes.

4.2 Différentes interfaces cerveau-ordinateur

Il existe plusieurs types d'interfaces cerveau-ordinateur, les deux principales sont :

- Celles basées sur les activités oscillatoires.
- Celles basées sur les potentiels évoqués.

Liaisons cerveau-ordinateur basées sur les activités oscillatoires

Il s'agit de la méthode la plus couramment utilisée. Elle consiste en l'étude des variations de la puissance sur certaines bandes de fréquence, en fonction des changements d'états mentaux de certaines zones du cerveau. Elle exploite donc l'information spatiale et spectrale. L'activité oscillatoire d'un signal EEG peut être caractérisée par deux événements (Figure 4.2) :

- **ERS, Synchronisation des événements liés.** Diminution de l'énergie due à l'activation corrélée du système nerveux dans certaines aires corticales.
- **ERD, Désynchronisation des événements liés.** Augmentation de l'énergie due à la désactivation corrélée du système nerveux dans certaines aires corticales.

Ces phénomènes oscillatoires sont essentiellement observés dans les rythmes alpha et bêta du cerveau.

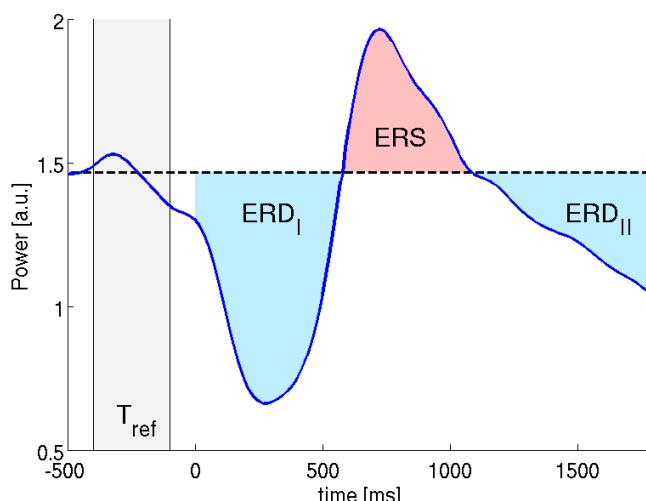


FIGURE 4.2 – Représentation des phénomènes d'oscillations ERS et ERD du rythme bêta du cerveau.

Source : <http://www.bbci.de/supplementary/conditionalERD/conditionalERD.htm>

Les interfaces cerveau-ordinateur qui s'appuient sur les activités oscillatoires du cerveau sont principalement employées dans des dispositifs s'appuyant sur des stimulations visuelles, ou lorsqu'on fait appel à la représentation et à la visualisation mentale d'objets en mouvement. Voici les deux interfaces cerveau-ordinateur basées sur les activités oscillatoires les plus utilisées :

- **SSVEP - Potentiel évoqué visuellement.** Ce dispositif repose sur une stimulation visuelle [?]. En effet, lorsque la rétine est excitée par une stimulation ayant une certaine fréquence (généralement entre 3.5 Hz et 75 Hz), le cerveau génère des signaux électriques (les SSVEP) de même fréquence que le stimulus. Cette méthode a l'avantage de nécessiter une période d'entraînement relativement courte (Figure 4.3).

- **Imagerie motrice.** Ce système repose quant à lui sur des méthodes d'entraînement à l'imagination mentale, i.e. on demande à un individu de penser à une action, comme bouger un cube dans un espace en 3 dimensions [?]. On observe alors des fluctuations dans les rythmes du cerveau que l'on pourra par la suite interpréter, et en déduire l'action pensée par la personne. L'inconvénient de cette méthode est qu'elle nécessite plusieurs sessions d'entraînement. Elle n'exige cependant pas de stimulation visuelle.

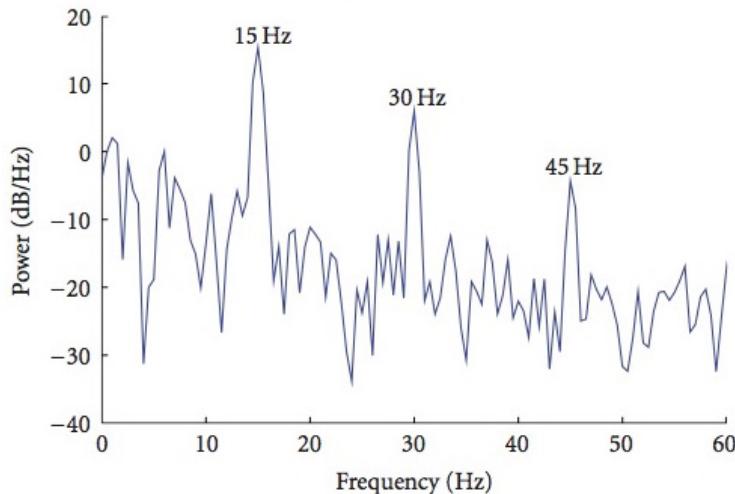


FIGURE 4.3 – Signal EEG de type SSVEP, en réponse à une stimulation visuelle de 15Hz.
Source : <http://www.hindawi.com/journals/cin/2010/702357/fig2/>

Liaisons cerveau-ordinateur basées sur les potentiels évoqués

Il s'agit d'une méthode qui s'appuie sur l'étude de l'information temporelle et spatiale des signaux EEG du cerveau [?]. Ce dispositif étudie les potentiels à événements liés (ERP). Il s'agit de signaux émis par le cerveau lorsque celui-ci est excité par un stimulus (visuel, tactile, etc). Un des ERP les plus utilisés actuellement, et notamment dans le monde des jeux-vidéos est le P300. Il s'agit d'une déflexion positive qui intervient 300ms environ après la stimulation sensorielle (Figure 4.4).

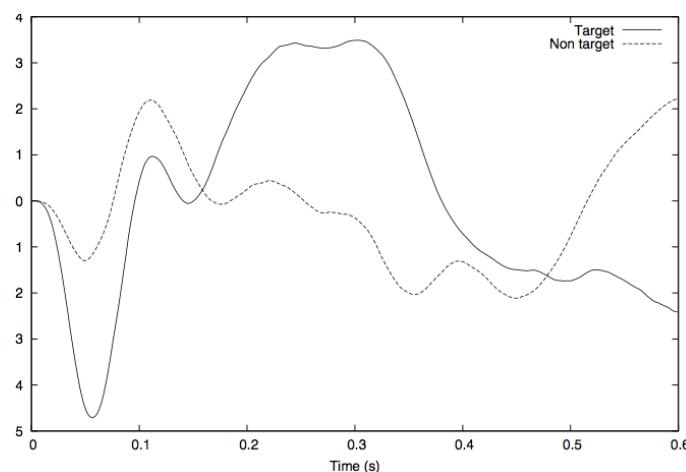


FIGURE 4.4 – Réponse P300 après un stimulus. On observe clairement la déflexion du signal après 300ms.

4.3 Traitement et analyse des signaux EEG

Le processus fonctionnel d'une interface cerveau-ordinateur peut être décrit comme un système en boucle fermée, composé de six étapes [?][?] (Figure 4.5) :

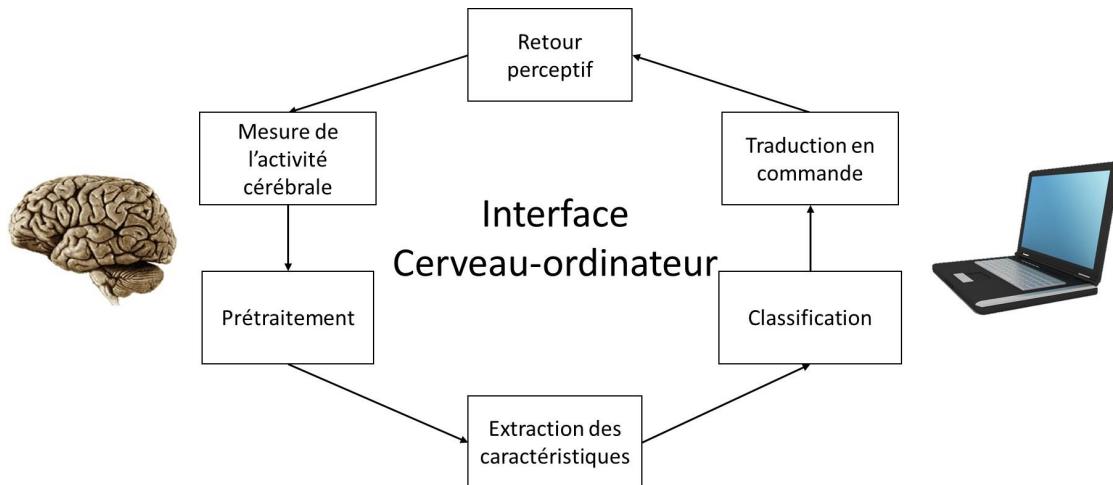


FIGURE 4.5 – Principales étapes constituant une interface cerveau-ordinateur.

1. Mesure de l'activité cérébrale via un dispositif électroencéphalographique. On parle de phase d'acquisition des signaux.
2. Pré-traitement et filtrage des signaux cérébraux permettant notamment de dissocier les différents rythmes du cerveau.
3. Extraction des caractéristiques des signaux, afin de ne conserver que des informations pertinentes.
4. Classification des signaux, afin d'identifier et de caractériser l'activité mentale et de l'interpréter.
5. Traduction en une commande envoyée à l'ordinateur. On fait ici le lien entre l'interprétation et la tâche qu'il doit effectuer.
6. Retour perceptif. Il peut par exemple s'agir d'un retour visuel lié à l'activité neuronale perçue. L'utilisateur va ainsi progressivement apprendre à mieux contrôler le système, et ainsi boucler le synoptique de l'interface cerveau-ordinateur.

On cherche ici à étudier le traitement et l'analyse des différents signaux EEG acquis grâce au casque EEG Emotiv. On s'intéressera principalement aux étapes 1, 2, 3 et 4. On a donc la chaîne de traitement suivante (Figure 4.6).

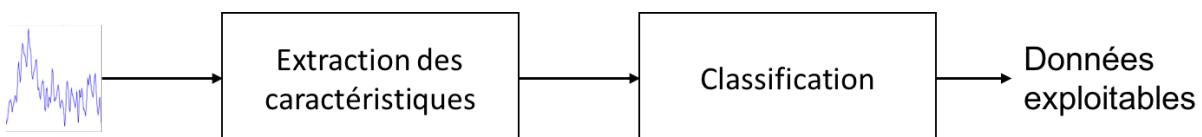


FIGURE 4.6 – Chaîne de traitement d'un signal EEG

Cette trame est utilisée pour le traitement d'un seul signal, c'est-à-dire pour un signal provenant d'une seule électrode. Or, étudier le signal de plusieurs électrodes permet d'augmenter la précision du système. Cela permet également d'accéder à une information supplémentaire : la localisation spatiale des signaux. Ce type de procédé augmente inévitablement la quantité de calculs à effectuer, d'autant plus que les informations doivent être

traitées en temps réel. Pour palier à ce problème, on ajoute une étape d'optimisation des données (prétraitement) afin de réduire le nombre d'informations à manipuler. On obtient donc cette nouvelle chaîne de traitement (Figure 4.7) :

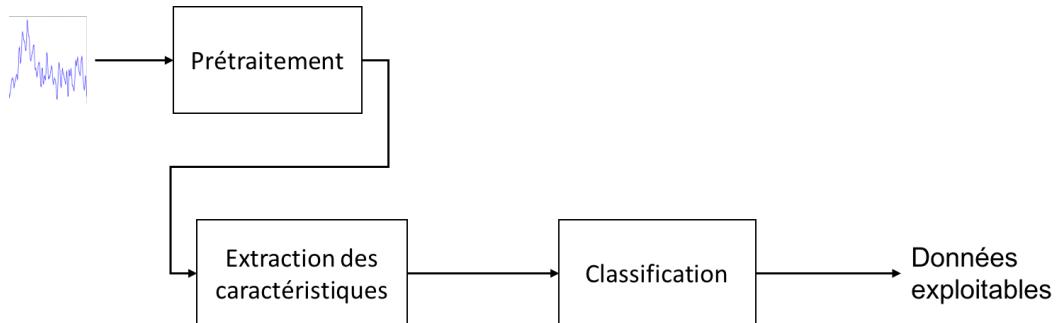


FIGURE 4.7 – Chaîne de traitement de plusieurs signaux EEG

4.3.1 Pré-traitement, méthodes d'optimisation des données

Il existe différents algorithmes d'optimisation des données, afin de réduire la quantité d'informations à traiter en temps-réel [?].

- **Sélection des caractéristiques.** Algorithme permettant de sélectionner automatiquement les données les plus pertinentes.
- **Sélection des entrées (électrodes).** Sélectionne automatiquement les électrodes dont les données sont les plus pertinentes. Cet algorithme est en général plus efficace que les méthodes de sélection des caractéristiques.
- **Filtrage spatial.** On réalise une combinaison linéaire des signaux issus des électrodes par suppression des redondances afin de réduire la quantité d'informations à traiter.

Filtrage spatial

Le filtrage spatial permet de réduire la quantité d'informations à traiter en fusionnant des chaînes entre elles [?] [?]. Il a également un sens du point de vu physiologique car il permet de retrouver les signaux sources. On définit donc le filtrage spatial comme une combinaison linéaire de l'ensemble des chaînes, chaque chaîne étant pondérée par un poids :

$$\tilde{x} = \sum w_i * x_i \quad (4.1)$$

Où \tilde{x} est le signal filtré spatialement, i le canal traité, w_i le poids attribué à i , x_i le signal EEG de i .

Il existe plusieurs méthodes de filtrage spatial :

- **Filtrage spatial fixe.** Dans un filtrage fixe, le poids attribué à chaque canal est prédéterminé par l'utilisateur. On utilise cette méthode principalement pour réduire le bruit de fond. En voici deux couramment utilisées :

Filtre Bipolaire. On calcule la différence entre deux canaux à proximité du canal que l'on souhaite étudier : $C_z = FC_3 - C_4$, où C_z , FC_3 et C_4 correspondent au placement des électrodes selon la norme Internationale 10-20.

Filtre Laplacien. $C_z = 4C_z - F_z - P_z - C_3 - C_4$. où C_z , F_z , P_z , C_3 , et C_4 correspondent au placement des électrodes selon la norme Internationale 10-20.

— **Filtrage spatial avec apprentissage.** Les poids sont déterminés automatiquement, grâce à un d'apprentissage.

PCA, Analyse en composantes principales. La PCA permet d'extraire des informations à partir de signaux bruités et de réduire le nombre de données à traiter.

ICA, Analyse en composantes indépendantes. L'ICA est utilisée en BSS (Séparation des sources masquées), c'est-à-dire lors de l'extraction de sources à partir de plusieurs signaux (ex : identifier et extraire une voix dans une soirée à partir de plusieurs microphones).

On s'intéressera ici plus particulièrement aux filtres spatiaux avec apprentissage car la sélection des poids à appliquer à chaque chaîne est faite automatiquement.

Analyse en composantes principales (PCA)

C'est une méthode d'analyse statistique des données qui permet d'extraire certaines informations à partir d'un flot de données diffus. On réduit ainsi la complexité des données. Pour cela, on les exprime dans un nouveau repère en effectuant une transformation linéaire, afin de décorrélérer les données entre elles. Un flot de donnée peut être caractérisé par deux informations : le bruit et la redondance [?].

Soit X une matrice contenant les échantillons de données de plusieurs sources ($m \times n$, m correspondant aux différentes sources X_i avec $0 < i < m$, n aux valeurs des sources à chaque instant t avec $0 < j < n$), Y la ré-écriture de X dans le nouveau repère. On cherche donc à déterminer la transformation linéaire P permettant de réduire à la fois le bruit et la redondance des données [?], tel que

$$Y = PX \quad (4.2)$$

Avec :

$$\begin{matrix} X_1 \\ X_2 \\ X_3 \\ \dots \\ X_m \end{matrix} \left(\begin{matrix} t_1 & t_2 & t_3 & \dots & t_{fin} \end{matrix} \right) = X \quad (4.3)$$

Le bruit. Il est exprimé par l'équation $SNR = \frac{\sigma_{signal}^2}{\sigma_{bruit}^2}$, avec SNR (Signal Noise Ratio) la mesure du bruit, σ_{signal}^2 la variance du signal et σ_{bruit}^2 la variance du bruit. Plus la valeur SNR est élevée, plus le signal est pur (sans bruit) [?] [?]. La variance caractérisant la dispersion d'un jeu de données autour de sa moyenne, on cherche donc à déterminer la transformation linéaire qui la maximise (Figure : 4.8).

La redondance des données. Plusieurs électrodes peuvent avoir enregistré la même information [?] [?]. On cherche donc à supprimer ses informations répétitives afin de diminuer la quantité d'informations à traiter. La mesure de la covariance représentant le degrés de relation linéaire entre deux jeux de données. On recherche alors la transformation linéaire permettant de diminuer sa valeur (figure : 4.9).

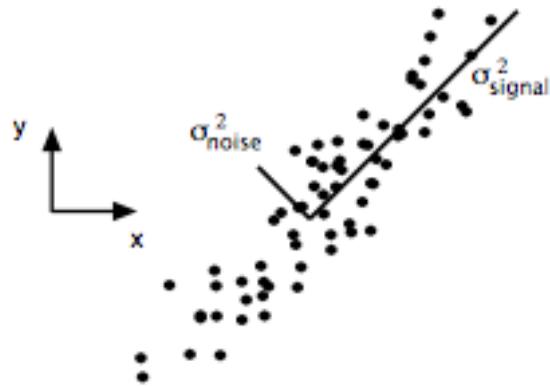


FIGURE 4.8 – Flux de données e X et Y . On représente ici la variance du bruit ainsi que du signal. [?]



FIGURE 4.9 – Redondances possibles des données dans deux flux de données R1 et R2. Les deux mesures à gauche ne sont pas corrélées. Les deux mesures à droite sont quant à elles fortement corrélées. [?]

La matrice de covariance. Elle est constituée de la covariance et de la variance d'un ensemble de jeux de données [?]. On cherche donc à maximiser la variance (donc SNR) et à annuler les valeurs de covariances (i.e. réduire la redondance). La solution optimale revient donc à diagonaliser la matrice.

$$M = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} & \dots & \sigma_{x_1 x_m} \\ \sigma_{x_2 x_1} & \sigma_{x_2}^2 & \dots & \sigma_{x_2 x_m} \\ \dots & \dots & \sigma_{x_3}^2 & \dots \\ \sigma_{x_m x_1} & \sigma_{x_m x_2} & \dots & \sigma_{x_m}^2 \end{bmatrix}$$

Diagonalisation de la matrice de covariance. Soit S_X la matrice de covariance de X , on cherche à déterminer la transformation linéaire P , de manière à ce que S_Y , la matrice de covariance de Y , soit diagonalisée [?] [?]. On a donc :

$$\begin{aligned} S_Y &= \frac{1}{n} YY^T \\ &= \frac{1}{n} (PX)(PX)^T \\ &= \frac{1}{n} PXX^TP^T \\ S_Y &= PS_XP^T \end{aligned} \tag{4.4}$$

D'après le théorème de diagonalisation orthogonale d'une matrice symétrique (une matrice de covariance étant par définition symétrique), S_X peut être exprimée par EDE^T , où D est une matrice diagonale et E est la matrice de vecteurs propres de A , rangés en colonnes. On sélectionne P de manière à ce que chacune de ses colonnes soit un vecteur propre de S_X . On pose $P = E^T$, on a donc $A = P^T D P$.

On obtient :

$$\begin{aligned}
 S_Y &= PS_x P^T \\
 &= P(E^T DE)P^T \\
 &= P(P^T DP)P^T \\
 &= (PP^T)D(PP^T) \\
 &= (PP^{-1})D(PP^{-1}) \\
 S_Y &= D
 \end{aligned} \tag{4.5}$$

Le choix de P diagonalise bien S_X . Les composantes principales de X sont les vecteurs propres de S_X .

Algorithme du PCA [?] [?]:

1. Soustraire la moyenne de chaque source à elle-même. La moyenne des données devient alors nulle, simplifiant ainsi le calcul de la variance et de la covariance.
2. Calculer la matrice de covariance.
3. Calculer les valeurs propres et vecteurs propres de la matrice de covariance.
4. Trier les vecteurs propres en fonction de la valeur des valeurs propres qui y sont rattachées (du plus grand au plus petit). Le vecteur des valeurs propres triées correspond au vecteur caractéristique.
5. Dériver les nouvelles données. On multiplie la transposée du vecteur caractéristique par les données dé-moyennées.

Analyse en composantes indépendantes (ICA)

La méthode de filtrage spatial par analyse des composantes indépendantes permet de réduire le nombre de données à traiter comme la PCA. De plus, elle permet de retrouver les signaux sources à partir des signaux enregistrés par les électrodes[?] (Figure : 4.10).

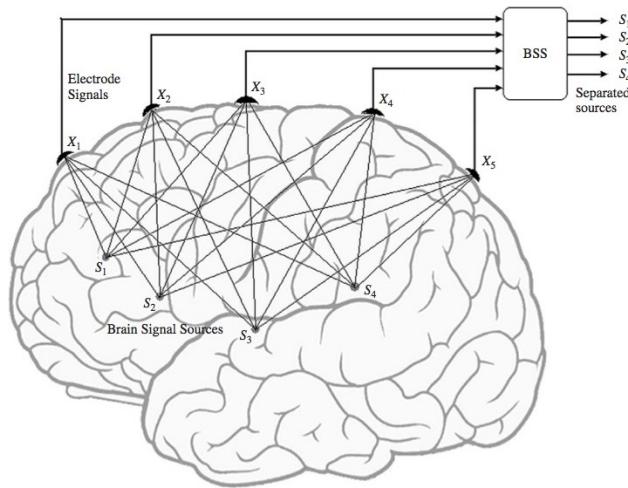
Soient S_i les signaux sources indépendants (l'amplitude de S_1 ne dépend pas de celle de S_2), X_i les signaux enregistrés par les électrodes, A la transformation linéaire mixant les sources S_i en X_i .

$$X_i = AS_i \tag{4.6}$$

Le but est donc de retrouver la matrice A et son inverse afin de reconstruire les signaux sources S_i à partir de ceux observés X_i . Soit S'_i une estimation des sources S_i , on doit donc déterminer la matrice W , elle-même une approximation de A^{-1}

$$S'_i = WX_i \tag{4.7}$$

On a alors une équation à deux inconnues, ce qui est impossible à résoudre sous la forme actuelle. Une solution est d'utiliser la décomposition en valeurs singulières (SVD).



La décomposition en valeurs singulières. Une matrice est décomposable en trois opérations linéaires simples [?] :

- Une rotation V . Il s'agit alors d'une matrice orthogonale.
 - Un étirement (ou compression) Σ . Il s'agit alors d'une matrice diagonale.
 - Une rotation U . Il s'agit alors d'une matrice orthogonale.

tel que :

$$A = U\Sigma V^T \quad (4.8)$$

Avec $A \in R^{m \times n}$, $U \in R^{m \times m}$, $\Sigma \in R^{m \times n}$, $V \in R^{n \times n}$.

Cette transformation linéaire s'apparente à un changement de repère, ce qui n'est pas sans rappeler la méthode utilisée pour la PCA (Figure : 4.11).

On peut écrire :

$$W = A^{-1} = V\Sigma^{-1}U^T \quad (4.9)$$

Étude de la covariance de X_i . On émet l'hypothèse suivante : soit S_X la covariance des signaux sources, on a $S_X = I$ (une matrice de covariance étant par définition symétrique). De plus, on rappelle que pour une matrice orthogonale, $V^{-1} = V^T$. La covariance des signaux observés par les électrodes peut alors être exprimée comme suit :

$$\begin{aligned}
S_X &= \frac{1}{n} XX^T \\
&= \frac{1}{n} (AS)(AS)^T \\
&= \frac{1}{n} (U\Sigma V^T S)(U\Sigma V^T S)^T \\
&= U\Sigma V^T \left(\frac{1}{n} SS^T\right) V\Sigma U^T \\
&= U\Sigma V^{-1} V\Sigma U^T \\
S_X &= U\Sigma^2 U^T
\end{aligned} \tag{4.10}$$

Si on pose $E = U$ avec E une matrice dont les colonnes correspondent aux vecteurs

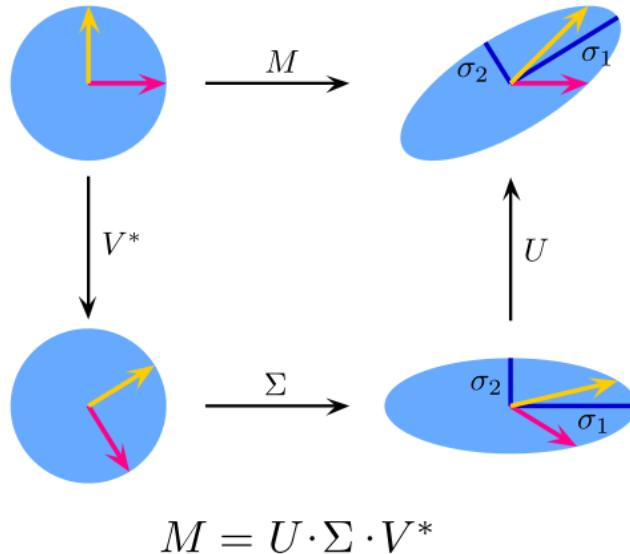


FIGURE 4.11 – La décomposition en valeurs singulières s'apparente à décomposer une matrice en trois opérations linéaires simples : deux rotations U et V et un étirement Σ . σ_1 et σ_2 correspondent aux valeurs singulières.

Source : https://fr.wikipedia.org/wiki/D%C3%A9composition_en_valeurs_singuli%C3%A8res

propres de S_X , $D = \Sigma^2$ avec D une matrice diagonale. On obtient alors :

$$S_X = EDE^T \quad (4.11)$$

Ce qui correspond à la propriété sur les matrices symétriques qui dit qu'une matrice peut être diagonalisée par ses vecteurs propres. On peut alors reformuler la valeur de W :

$$W = VD^{-\frac{1}{2}}E^T \quad (4.12)$$

Multiplier par $D^{-\frac{1}{2}}E^T$ revient jusqu'ici à réaliser une analyse en composantes principales, c'est à dire à supprimer les composantes dépendantes (redondance des informations). Soit $X_w = D^{-\frac{1}{2}}E^T X$ les données de X dont les dépendances ont été supprimées et d'après l'équation 4.4, on a :

$$S'_i = V X_w \quad (4.13)$$

Il nous reste donc à caractériser la transformation linéaire V .

Résolution de V. On cherche la valeur de la matrice de rotation V tel que les sources S'_i soient statistiquement indépendantes [?]. i.e.

$$P(s) = \prod P(S_i) \quad (4.14)$$

Avec $P(s)$ la loi de probabilité de la source S et $P(S_i)$ la loi de probabilité de l'ensemble des sources S_i . La mesure des statistiques indépendantes entre plusieurs variables peut être déterminée par l'équation de multi-information suivante :

$$I(S) = \int P(S) \log_2 \frac{P(S)}{\prod_i P(S_i)} dy \quad (4.15)$$

Si $I(S) = 0$, cela signifie que les sources sont totalement indépendantes. En effet, si $P(S) = \prod_i P(S_i)$, on a $\log\left(\frac{P(S)}{\prod_i P(S_i)}\right) = \log(1) = 0$. Il faut donc trouver la matrice V tel que $I(S) = 0$.

Algorithme de l'ICA [?]

1. Soustraire la moyenne de chaque source à elle-même. La moyenne des données devient alors nulle, simplifiant ainsi le calcul de la variance et de la covariance.
2. Calculer la matrice de covariance.
3. Calculer les valeurs propres D et vecteurs propres E de la matrice de covariance.
4. Dériver les nouvelles données. On multiplie la transposée du vecteur caractéristique par les données dé-moyennées. Le résultat correspond aux signaux observés sur les électrodes, dont on a supprimé les dépendances. Soit $E^T X$.
5. Multiplier le résultat précédent par la racine du vecteur des valeurs propres. soit $X_w = D^{-\frac{1}{2}} E^T X$.
6. Calculer la rotation (V) qui maximise la corrélation des données.
7. Calculer $W = V D^{-\frac{1}{2}} E^T$.
8. On peut alors retrouver les signaux sources S_i en calculant $S_i = W \times X_0$.

Les étapes 1 à 4 correspondent à une analyse en composantes principales.

Configuration Spatiale Commune (CSP)

La méthode CSP est utilisée à la fois en filtrage spatial et pour les algorithmes de sélection automatique de canaux (signaux de sortie des électrodes) [?] [?]. Tout comme l'algorithme du PCA, on cherche ici à réaliser une transformation linéaire sur un flot de données, afin de décorrélérer les sources entre elles. À la différence qu'ici, on a en entrée du filtre deux jeux de données, que l'on appellera des classes. Dans notre cas, il peut s'agir d'une classe "stimulus gauche" correspondant aux signaux générés après le mouvement de la main gauche, ou d'une classe "stimulus droit" correspondant aux signaux générés après le mouvement de la main droite. On cherche alors à maximiser la variance et à minimiser la covariance entre les deux classes. On a donc deux étapes principales :

1. Maximiser la variance et minimiser la covariance des sources de chacune des classes.
2. Maximiser la variance de chacune des deux classes et minimiser la covariance entre les deux classes.

Soit R_H et R_F les covariances des classes H et F , tels que :

$$R_H = \frac{X_H X_H^T}{\text{Trace}(X_H X_H^T)} \quad R_F = \frac{X_F X_F^T}{\text{Trace}(X_F X_F^T)} \quad (4.16)$$

Avec X_H et X_F les données de la classe H et X_F les données de la classe F . La covariance spatiale totale correspond à la somme des covariances des deux classes :

$$R = R_H + R_F \quad (4.17)$$

Algorithme du CSP

1. Soit X_g un ensemble de sources de données EEG correspondant aux réponses cérébrales lors du mouvement de la main gauche et X_d un ensemble de données correspondant aux réponses cérébrales lors du mouvement de la main droite. On soustrait la moyenne de chacune des sources des deux ensembles à elles-mêmes. La moyenne des données devient alors nulle, simplifiant ainsi le calcul de la variance et de la covariance.
2. Calculer la matrice de covariance des deux ensembles X_g et X_d . On appellera R_g la matrice de covariance de l'ensemble X_g et R_d la matrice de covariance de l'ensemble X_d .
3. Calculer les valeurs propres D et vecteurs propres E de la matrice de R_g et R_d .
4. Dériver les nouvelles données. On multiplie la transposée du vecteur caractéristique par les données dé-moyennées. Le résultat correspond aux signaux observés sur les électrodes, dont on a supprimé les dépendances, soit $E^T X$.
5. Multiplier le résultat précédent par la racine du vecteur des valeurs propres. soit $X_w = D^{-\frac{1}{2}} E^T X$.
6. Calculer la valeur des sources des deux ensembles, soit $S_g = W R_g W^T$ et $S_d = W R_d W^T$.
7. Calculer la rotation (V) qui maximise la corrélation des données.
8. Calculer $W = V D^{-\frac{1}{2}} E^T$.
9. Calculer les valeurs propres D et vecteurs propres généralisés B des sources S_g et S_d .
10. Trier les vecteurs propres en fonction de la valeur des valeurs propres qui y sont rattachées (du plus grand au plus petit).
11. Le calcul du poids à attribuer à chacun des canaux est déterminé de la manière suivante : $Poids = B' W$

4.3.2 Méthodes d'extraction des caractéristiques

Filtrage temporel

La phase d'extraction des données consiste dans un premier temps à réaliser un filtrage temporel afin d'extraire du signal EEG les rythmes du cerveau (caractéristiques) qui nous intéressent. Pour cela, on réalise un filtrage passe-bande. Les deux types de filtres les plus utilisés en traitement du signal sont les filtres de Butterworth et Chebychev.

- **Filtre de Butterworth.** Il s'agit d'un filtre qui possède un gain quasiment constant dans sa bande-passante (on parle de critère de méplat). L'avantage est qu'il engendre très peu de distorsions dans la bande passante. Sa réponse en phase est presque linéaire. Cependant sa réjection est moins importante dans la bande coupante, dû à la "limitation" de la bande de coupure.
- **Filtre de Chebychev.** Il s'agit d'un filtre dans lequel on introduit une oscillation (distorsion contrôlée), permettant d'obtenir une "vitesse" de coupure plus importante. L'inconvénient de ce filtre est que les oscillations introduites entraînent des distorsions de phase sur le signal de sortie.

Il est préférable d'utiliser une filtre de type Butterworth dans notre cas car on ne peut pas tolérer l'introduction de distorsions dans les rythmes du cerveau. On propose donc d'utiliser un filtre passe-bande d'ordre 5.

Représentation spectrale

Une fois les caractéristiques extraites, on souhaite les représenter dans le domaine spectral afin de structurer les données pour la classification. La méthode majoritairement utilisée dans le traitement des signaux EEG est la transformée de Fourier à court terme (TFCT). Il s'agit d'une méthode dérivée de la transformée de Fourier rapide (FFT) qui gère la non-stationnarité des signaux d'entrée en les fenêtrant. Concrètement, cela revient à calculer la FFT, puis à appliquer un fenêtrage [?].

$$STFT(x(t)) = X(\tau, u) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-j2\pi.u.t}dt$$

Avec $X(t, \tau)$ la transformée de $x(t)w(t - \tau)$, $x(t)$ le signal à transformer, $w(t)$ le signal de fenêtrage, τ la valeur de décalage temporel des fenêtres et u la fréquence. Il s'agit la plupart du temps d'un fenêtrage de pondération de Hanning ou de Gauss, qui offre à la fois une bonne résolution spectrale et une bonne dynamique au filtrage. Si on discrétise les valeurs, on obtient le calcul de la STFT suivant :

$$STFT(x[n]) = \sum_0^{N-1} x(n)w(n - m)e^{-j2.\pi.\frac{nk}{N}}$$

avec k l'échantillon calculé, N le nombre de points et n la variable.

La largeur des fenêtres étant constante, on observe une diminution de la résolution temporelle lorsque les fréquences deviennent trop faibles.

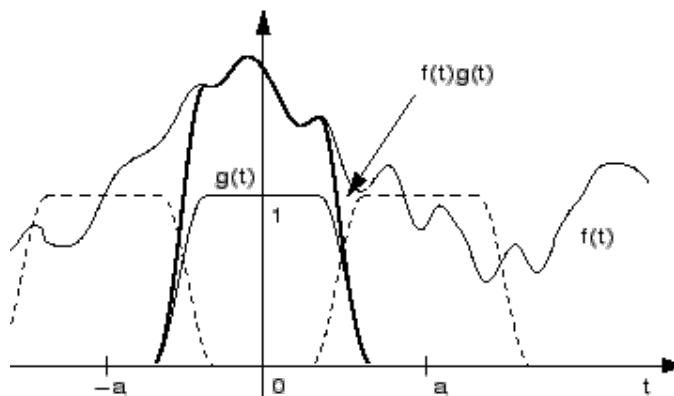


FIGURE 4.12 – STFT du signal $f(t)$, avec fenêtrage par le signal $g(t)$
<https://www.math.ucdavis.edu/strohmer/research/gabor/gaborintro/img47.gif>

4.3.3 Méthodes de classification

La phase de classification consiste à trier et à classifier des échantillons dans une catégorie (classe).

Dans le cas général, les données d'entrées sont les suivantes :

- P classes d'objets du même type.
- N observables définies pour toutes les classes.
- Pour chacun des P classes, un échantillon d'objets N_k (k étant l'indice de classe).

Pour classifier un nouvel objet dans une des classes définies, on passe par cinq phases distinctes :

- La collecte des données.
- La caractérisation des classes.
- Le choix d'un algorithme de classification.
- L'apprentissage.
- Les tests.

Nous avons étudié les 3 algorithmes de classification suivants et nous les avons comparé en fonction leurs applications possibles, leurs avantages et leurs inconvénients [?].

Linear Discriminant Analysis

L'analyse discriminante linéaire (LDA) est l'algorithme le plus populaire dans le domaine des interfaces cerveau-ordinateur (BCI).

Il est réputé comme étant un classificateur stable en raison de sa faible complexité, il s'appuie sur des calculs simples, et est adapté à la classification en temps réel (Figure 4.13).

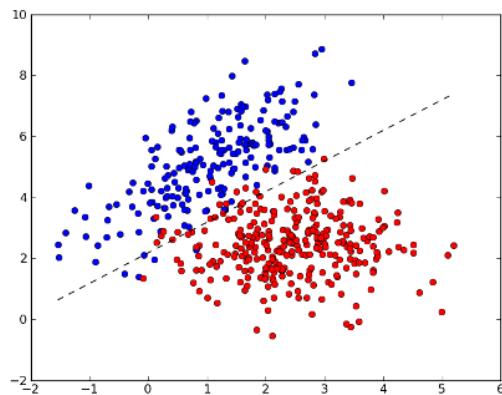


FIGURE 4.13 – Classification LDA

Principes de LDA :

Le principe de la méthode LDA est illustré par les deux figures suivantes. Il a été supposé que deux observables, X et Y, étaient accessibles à l'observateur. Les zones bleues sont éliminées par les coupures (qui sont représentées par les lignes bleues).

Les deux premiers graphiques montrent le résultat de coupes classiques : on peut voir le compromis réalisé dans les deux cas entre précision et efficacité (Figure 4.14).

LDA consiste à découper le long d'une combinaison linéaire de tous les observables, plutôt que le long de chacun des observables. Cette combinaison linéaire est définie par une direction LDA (ou axe).

L'algorithme consiste à calculer la direction de cet axe de manière à avoir une discrimination optimale entre les classes conformément à un critère donné. Un hyperplan perpendiculaire à l'axe est alors associé à cette discrimination (Figure 4.15]).

Hypothèses et formules :

L'objectif est de produire une règle d'affectation $X(\omega) \mapsto Y(\omega)$ qui permet de prédire, pour une observation ω donnée, sa valeur associée de Y à partir des valeurs prises par X.

La règle bayésienne consiste à produire une estimation de la probabilité à posteriori

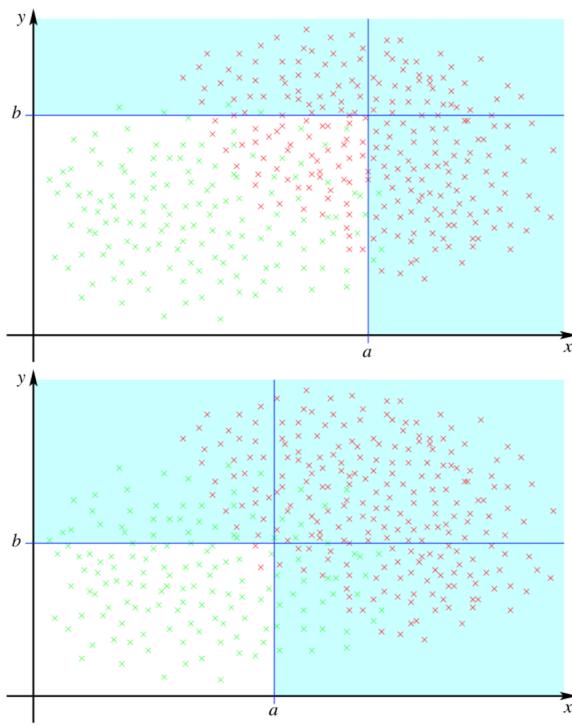


FIGURE 4.14 – Principe de LDA 1 [?]

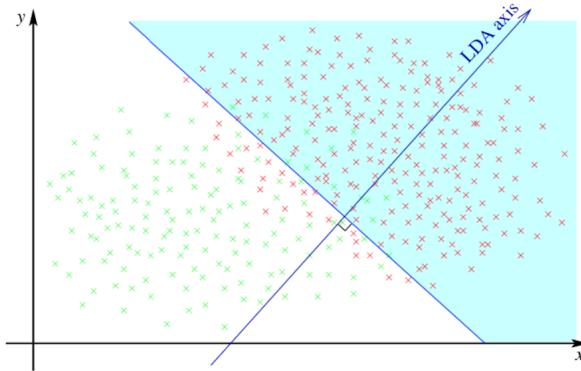


FIGURE 4.15 – Principe de LDA 2 [?]

(appuyée sur les expériences et les faits) d'affectation :

$$P(Y = y_k | X) = \frac{P(Y = y_k) \times P(X | Y = y_k)}{\sum_{i=1}^K P(Y = y_i) \times P(X | Y = y_i)} \quad (4.18)$$

$P(Y = y_k)$ est la probabilité d'appartenance à une classe. $P(X | Y = y_k)$ représente la fonction de densité des X conditionnellement à la classe y_k .

La règle d'affectation pour un individu ω à classer devient alors :

$$Y(\omega) = \arg \max_k P(Y = y_k | X(\omega)) \quad (4.19)$$

Toute la problématique de l'analyse discriminante revient alors à proposer une estima-

tion de la quantité :

$$P(X \mid Y = y_k) \quad (4.20)$$

On distingue principalement deux approches pour estimer correctement la distribution $P(X \mid Y = y_k)$:

- L'approche non-paramétrique n'effectue aucune hypothèse sur cette distribution mais propose une procédure d'estimation locale des probabilités, au voisinage de l'observation ω , à classer. Les procédures les plus connues sont la méthode d'estimation par noyau et la méthode des plus proches voisins.
- La seconde approche effectue une hypothèse sur la distribution des nuages de points conditionnels, on parle dans ce cas d'analyse discriminante paramétrique. Exemple : hypothèse de multinormalité, etc.

Support Vector Machine

SVM est une méthode de classification binaire par apprentissage supervisé, destinée à résoudre des problèmes de discrimination (décider à quelle classe appartient un échantillon) et de régression (prédirer la valeur numérique d'une variable).

Cette méthode repose sur l'existence d'un classificateur linéaire dans un espace approprié.

Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonction dites noyau qui permettent une séparation optimale des données.

Après la méthode LDA, c'est la deuxième méthode la plus populaire parmi les classificateurs linéaires.

Bayes quadratique

Cette approche est basée sur le calcul de la probabilité de chaque classe, compte tenu du vecteur de caractéristiques. Bayes quadratique assigne le signal EEG à la classe la plus probable. Ce classificateur n'est pas très populaire pour les systèmes BCI, cependant, il reste assez précis pour la classification de l'imagerie motrice (motor imagery) comparé à d'autres méthodes.

Nous avons décidé de partir sur l'algorithme LDA car nous souhaitons travailler en temps réel, et il se trouve que la méthode LDA semble être la plus performante pour ce genre de tâche.

Chapitre 5

Utilisation du casque EPOC et d'OpenViBE

5.1 OpenViBE

5.1.1 Généralités sur le fonctionnement d'OpenViBE

OpenViBE est une plateforme logicielle dédiée à la conception et à l'utilisation d'interfaces cerveau-ordinateur en temps réel. Celle-ci est éditée par l'Institut National de Recherche en Informatique et en Automatique (INRIA) de Rennes en France. Le logiciel gère à la fois l'acquisition, le traitement et l'analyse des signaux.

OpenViBE fonctionne sous forme de scénarios, correspondant à un ensemble de blocs fonctionnels. La suite OpenViBE est composée de deux logiciels :

- **OpenViBE designer.** Il s'agit d'un éditeur de scénario permettant de créer et modifier les schémas blocs grâce à une interface graphique. Il suffit alors de placer les différents blocs désirés, de les configurer (en cliquant dessus), et de les relier pour former une chaîne de traitement (procédure).
- **OpenViBE acquisition server.** Permet d'acquérir des données EEG brutes à partir d'un dispositif EEG. Il est par exemple possible de connecter différents types de casques comme le casque EPOC d'Emotiv.

Il est possible d'envoyer des données d'OpenViBE vers une application externe via une connexion VRPN (Virtual Reality Peripheral Network).

5.1.2 Installation d'OpenViBE

Le logiciel fonctionne sous les systèmes d'exploitations Windows (Windows XP ou ultérieur, la stabilité du logiciel sous Windows 8 n'ayant pas été vérifiée par l'éditeur) et Linux (Ubuntu, Fedora). La liste détaillée des architectures compatibles avec OpenViBE est disponible à l'adresse suivante : <http://openvibe.inria.fr/supported-architectures/>. Il est possible de télécharger l'installateur OpenViBE à l'adresse suivante : <http://openvibe.inria.fr/downloads/>. Une fois celui-ci téléchargé, il suffit de l'ouvrir et de suivre les indications affichées à l'écran.

5.1.3 Utilisation d'OpenViBE designer

Description de l'interface de contrôle de la simulation

La partie supérieure de l'interface graphique permet de contrôler la simulation, i.e. le déroulement du scénario. Ainsi, il est possible de lancer un scénario en appuyant sur le bouton "lecture", de le stopper en appuyant sur le bouton "stop", d'en accélérer le déroulement en appuyant sur le bouton "avance rapide", etc. Il est également possible de lancer plusieurs scénarios en même temps (Figure : 5.1).

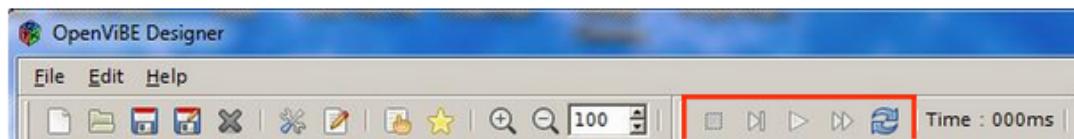


FIGURE 5.1 – Interface de contrôle de la simulation sous OpenViBE.

Description de l'interface du répertoire des blocs fonctionnels

La partie située à droite de l'interface graphique de l'éditeur permet de rechercher les blocs désirés afin de construire un scénario. Ceux-ci sont triés en différentes catégories. Un simple glisser-déposer permet de placer un bloc dans l'espace de travail (Figure : 5.2).

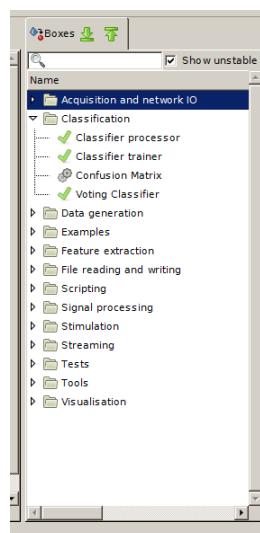


FIGURE 5.2 – Interface du répertoire des blocs OpenViBE.

Description de l'espace de travail

La partie centrale de l'interface graphique correspond à l'espace de travail d'OpenViBE. C'est ici où l'on construit les différents scénarios de l'interface cerveau-ordinateur, via des blocs fonctionnels. Chaque bloc peut être relié à un autre en maintenant la souris sur la sortie d'un bloc, vers l'entrée d'un autre bloc. On peut seulement connecter une entrée à une sortie du même type (e.g. une entrée de type "signal" vers une sortie de type "signal") (Figure : 5.3).

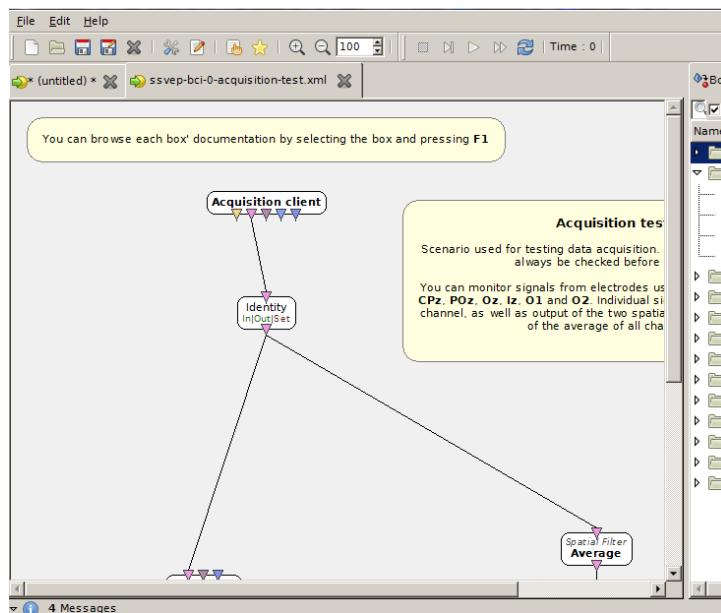


FIGURE 5.3 – Espace de travail d'OpenViBE.

5.1.4 Description des blocs utilisés

Voici la liste des différents blocs utilisés dans le cadre de notre projet :

Acquisition et Réseau

- **Acquisition Client.** Ouvre une interface de connexion pour lire les informations envoyées par le serveur d'acquisition (i.e. les données du casque en temps réel). Il faut pour cela indiquer le port utilisé par le serveur (1024 par défaut).
- **Analog VRPN Server.** Permet d'envoyer des données à une application client via le protocole VRPN.

Traitement de signal

- **Temporal Filter.** Réalise le filtrage temporel d'un signal. Il est possible de choisir la méthode de filtrage (passe-bas, passe-bande ou passe-haut), le type (Butterworth, Chebychev) et l'ordre de filtre.
- **Spatial Filter.** Permet de réaliser un filtrage spatiale statique, i.e. en paramétrant le poids attribué à chaque canal d'entrée manuellement. (filtrage bipolaire, Laplacien, etc.)
- **FastICA.** Permet de réaliser un filtrage spatial via l'algorithme ICA (4.3.1). Attention, le bloc est cependant considéré comme non stable par l'INRIA.
- **CSP Spatial Filter Trainer.** Permet de réaliser un filtrage spatial via l'algorithme CSP (4.3.1).
- **Time Based Epoching.** Permet de découper un signal en plusieurs morceaux ayant une durée et un intervalle spécifié.
- **Simple DSP.** Permet de réaliser des algorithmes mathématiques simples.
- **Signal Average.** Permet de calculer la moyenne d'un signal.
- **Spectral Analysis (FFT).** Réalise l'analyse spectrale de signaux en utilisant la FFT (Fast Fourier Transform).

- **Spectrum Average.** Calcule la moyenne de toutes les puissances de bande de fréquence pour un spectre donné.

Classification

- **Classifier Trainer.** Permet de réaliser des algorithmes de classification. Ce bloc effectue plus particulièrement l'entraînement du classificateur. Les différents algorithmes proposés sont la LDA et la SVM.
- **Classifier Processor.** Permet de réaliser la classification de vecteurs caractéristiques entrants, en utilisant un classificateur entraîné précédemment.

Lecture et écriture de fichiers

- **Generic Stream Writer.** Permet d'écrire des données dans un fichier binaire. Extension du fichier : .ov
- **Generic Stream Reader.** Permet de lire un fichier enregistré grâce à un bloc Generic File Writer.
- **CSV File Reader.** Ce bloc permet de lire des données (signaux par exemple) issues d'un fichier CSV (Comma-Separated Values). Le format CSV est ouvert et représente les données sous la forme de valeurs séparées par des virgules (format Européen), des points-virgules (format Américain) ou des tabulations. Ce bloc peut-être utilisé pour réaliser l'importation de données depuis Matlab ou Octave.
- **CSV File Writer.** Ce bloc permet d'écrire des données issues d'OpenViBE dans un fichier CSV. Il peut par exemple être utilisé pour réaliser de l'exportation de données vers Matlab ou Octave.

Simulation

- **Player Controller.** Permet de contrôler l'arrêt et la mise en pause d'un scénario à l'aide des stimulations reçues.

Visualisation

- **Signal Display.** Permet d'afficher un signal en temps réel.
- **Power Spectrum Display.** Permet d'afficher l'amplitude d'un signal dans un ensemble de bandes de fréquences.
- **2D Topographic Map.** Permet d'afficher la topographie de l'activité cérébrale en 2 dimensions.
- **3D Topographic Map.** Permet d'afficher la topographie de l'activité cérébrale en 3 dimensions.
- **Graz Visualisation.** Permet d'afficher la consigne et le résultat en temps réel.

Autres

- **Feature aggregator.** Permet de rassembler les différentes entrées du bloc vers un vecteur caractéristique.
- **Identity .** Duplique l'entrée de ce bloc vers sa sortie.

5.1.5 Utilisation du serveur d'acquisition d'OpenViBE

On s'appuie ici sur l'utilisation d'OpenViBE acquisition server afin d'acquérir les données EEG brutes à partir du casque EPOC (composé, pour rappel, de 14 canaux EEG). Afin d'accéder à ces données, il est nécessaire d'être en possession du SDK de ce casque (édition de recherche). De plus, il est nécessaire d'utiliser le système d'exploitation Windows afin d'assurer la compatibilité d'OpenVIBE avec le casque EPOC.

Au lancement du serveur d'acquisition d'OpenViBE, la fenêtre de configuration apparaît (Figure 5.4, fenêtre de gauche).

- **Driver.** Indiquer le pilote correspondant au casque utilisé via le menu déroulant. Il s'agit dans notre cas du casque EPOC d'Emotiv.
- **Connection Port.** Permet de renseigner le port de connexion du casque. Laisser la valeur à 1024.
- **Sample count per sent block.** Permet de configurer la taille des paquets envoyés du casque vers OpenViBE. Laisser à 32.

Appuyer sur le bouton "Driver properties" ouvre la fenêtre de configuration du casque (Figure 5.4, fenêtre de droite). Il faut alors renseigner les champs suivants :

- **Identifier.** Permet d'enregistrer les données de plusieurs utilisateurs. Laisser la valeur à 0.
- **Age.** Renseigner l'âge de l'utilisateur.
- **Gender.** Renseigner le sexe de l'utilisateur.

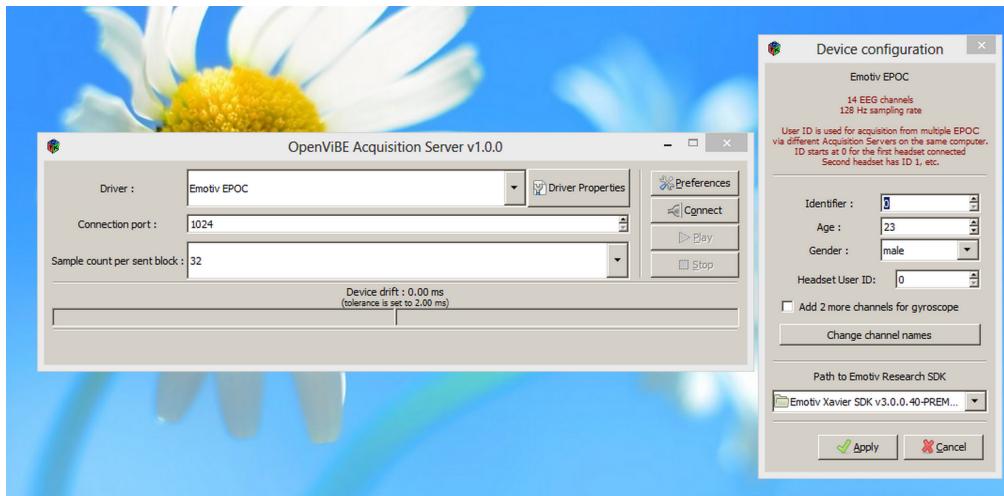


FIGURE 5.4 – Lancement du serveur d'acquisition

Ensuite, il ne reste plus qu'à appuyer sur "Connect" et sur "Play". La connexion entre le casque et OpenViBE s'effectue alors via le serveur d'acquisition. Le message "Connection succeeded" informe l'utilisateur que la connexion avec le casque a été réalisée (Figure 5.5).

5.2 Utilisation du casque EPOC

5.2.1 Généralités sur le fonctionnement du casque

Afin d'utiliser le casque EPOC d'Emotiv et d'acquérir les données EEG, il est nécessaire d'utiliser deux éléments :

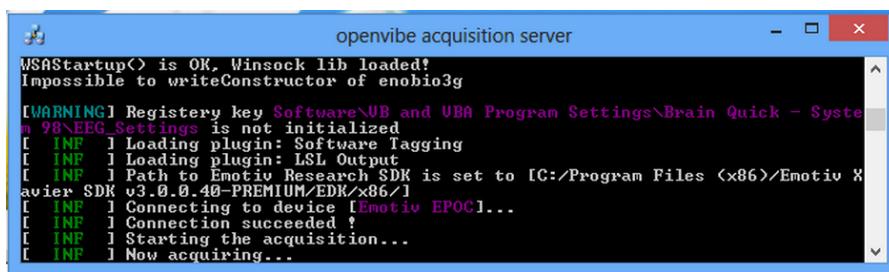


FIGURE 5.5 – Serveur démarré

- **Emotiv Xavier Control Panel Premium.** Il s'agit du panneau de configuration du casque EPOC. Il permet de le configurer et de le tester via une interface graphique.
- **SDK Emotiv.** Il s'agit du kit de développement du casque. Dans le cadre de notre projet, il nous permettra d'utiliser le casque avec OpenViBE et d'acquérir les signaux EEG.

5.2.2 Installation de Emotiv Xavier Control Panel Premium et du SDK.

Pour installer les deux utilitaires, il faut suivre la démarche suivante :

1. Se connecter sur son compte Emotiv (<https://emotiv.com/>).
2. Aller dans l'onglet "my downloads".
3. Télécharger le SDK du casque.
4. Une fois l'installateur téléchargé, l'ouvrir et suivre les instructions affichées à l'écran. Il vous sera demandé de fournir le code de la licence. Celui-ci est indiqué sur la page de téléchargement. L'installation de l'Emotiv Xavier Control Panel Premium s'effectuera lors de l'installation du SDK.

5.2.3 Préparation du casque

Voici les étapes de préparation du casque :

1. Bien humidifier chaque électrode avec la solution saline (jusqu'à saturation).
2. Insérer ces électrodes sur le casque (il faut sentir un cran).
3. Brancher la clé USB pour assurer la connexion sans fil avec le casque.
4. Allumer le casque.

5.2.4 Conseils sur l'utilisation du casque

- Ré-hydrater les électrodes toutes les heures pour garantir un bon contact dermal et donc un signal optimal.
- S'assurer du bon fonctionnement du casque en vérifiant régulièrement la qualité du signal via le logiciel "Emotiv Xavier Control Panel Premium".
- Ranger les électrodes dans leur boîte lorsque le casque n'est pas utilisé et s'assurer que le tampon soit humide (dans le cas contraire, l'humidifier avec la solution saline).
- Utiliser un tampon alcoolisé pour désinfecter les électrodes lorsque le casque n'est plus utilisé.
- Ne pas utiliser le casque lorsque sa batterie est en recharge.

5.2.5 Vérification de la qualité des signaux

Afin d'exploiter de manière optimale les données provenant du casque, il est nécessaire de vérifier la qualité du signal à partir du panneau de configuration du casque. Lors de l'ouverture logiciel, et après connexion, le menu principal apparaît. Sur l'interface, les différentes électrodes sont représentées par des points. Un point vert représente un signal de bonne qualité, un point orange de moyenne qualité et un point rouge ou noir correspond à un signal de mauvaise qualité (Figure 5.6).

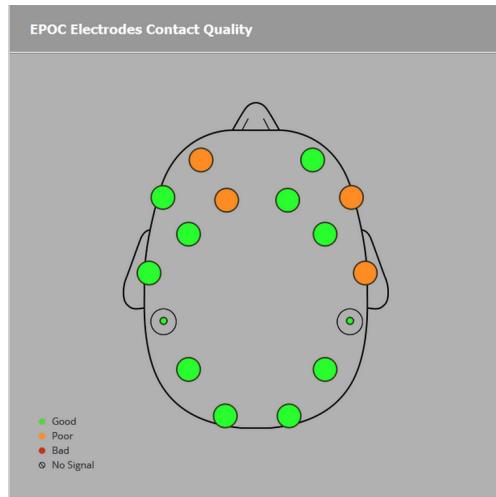


FIGURE 5.6 – Qualité du signal provenant des électrodes

Chapitre 6

Expérience simple d'interaction

6.1 Présentation de l'expérience

Cette expérience simple d'interaction permet de mettre en application les différents aspects théoriques étudiés dans la partie 4. Celle-ci s'appuie sur la méthode d'imagerie motrice qui consiste à imaginer le mouvement de différentes parties du corps, ce qui résulte en l'activation du cortex sensorimoteur modulant les rythmes alpha et bêta du cerveau. Ces variations peuvent être détectées par l'interface cerveau-ordinateur. Il est alors possible d'en déduire les intentions de l'utilisateur. Le but de l'expérience présentée est de déterminer si l'utilisateur pense à un mouvement de sa main droite ou de sa main gauche. Pour cela, on se base sur l'utilisation du casque EPOC (partie 3.3). On réalise la chaîne de traitement et d'analyse des signaux sous la forme de 5 scénarios OpenViBE (partie 3.4 et 5.1).

6.2 Développement des scénarios sous OpenViBE

6.2.1 Scénario 1 : Acquisition des données pour apprentissage

Ce scénario permet de réaliser l'acquisition des signaux EEG afin d'effectuer la configuration et l'apprentissage des filtres spatiaux et des classificateurs. Pour cela, on présente à l'utilisateur un certain nombre de stimuli visuels de façon à lui indiquer quelle est la tâche mentale à réaliser (Figure 6.2). On lui présentera par exemple 20 flèches à gauche, indiquant à celui-ci de penser à un mouvement de la main gauche et 20 flèches à droite pour imaginer un mouvement de la main droite dans un ordre aléatoire (Figure 6.1). De cette manière, OpenViBE associe un stimulus particulier (gauche ou droite) à une partie des signaux EEG, i.e. à une période temporelle correspondant à la durée du stimulus.

Étapes de déroulement du scénario (Figure 6.2) :

1. On réalise l'acquisition des données à partir du casque Epoc (bloc "Acquisition Client"). Parallèlement à cela, on génère des stimuli visuels à partir d'un script LUA (bloc "Graz Motor Imagery BCI Simulator")
2. On affiche les stimuli (Bloc " Graz Visualization") et on enregistre les signaux EEG ainsi que les labels correspondants aux stimuli visuels (bloc "Generic Stream Writer").
3. On stoppe l'exécution du script lorsque l'affichage des stimuli est terminé (bloc "Player Controller").

Il faut environ 9 minutes pour réaliser l'enregistrement des signaux EEG lors de la visualisation de 40 stimuli.

Configuration du bloc "Acquisition Client" :

- Acquisition Server HostName : "\$AcquisitionServer_HostName".
- Acquisition Server Port : Port de connexion correspondant au casque. Valeur par défaut 1024.

Configuration du bloc "Graz Motor Imagery BCI Simulator" :

Permet de configurer le script LUA générerant des stimulations visuelles. Remarquez que deux labels sont utilisés pour la configuration du bloc : "OVTK_GDF_Left" et "OVTK_GDF_Right". Ces drapeaux seront affiliés au signal EEG correspondant au stimulus visuel gauche et droit.

Configuration du bloc "Generic stream writer" :

- File Name : Indique le répertoire et le nom sous lequel le scénario doit enregistrer les données EEG (extension du fichier .ov).

Configuration du bloc "Player Controller" :

- Stimulation Name : Indique le nom du label auquel l'action de contrôle de la simulation doit être associée.
- Action to perform : On associe au drapeau définit précédemment une action de contrôle de la simulation. Il s'agit ici de la fonction "STOP" afin d'arrêter la simulation lorsque la stimulation visuelle du script LUA est terminée.

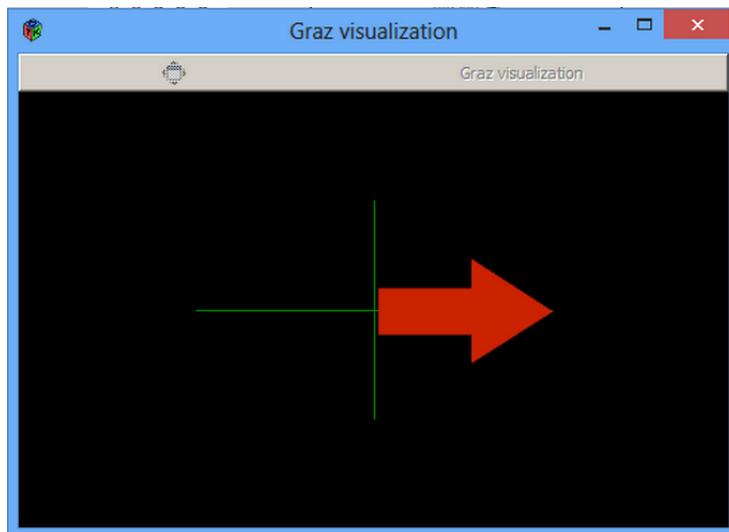


FIGURE 6.1 – Stimulation visuelle pour l'apprentissage. 40 stimuli visuels sont présentés à l'utilisateur. Lorsque celui-ci voit une flèche orientée vers la gauche, il doit penser à un mouvement de sa main gauche. Même exercice pour les flèches orientées vers la droite.

6.2.2 Scénario 2 : Entrainement du filtre spatial CSP

Ce scénario gère l'entraînement du filtre spatial de type CSP (partie 4.3.1) qui permet de réduire le nombre de canaux (sorties des électrodes) à traiter (Figure 6.3). Celui-ci fournit en sortie un fichier .cfg. Il s'agit d'un fichier contenant l'ensemble des informations de configuration du filtre CSP (poids attribué à chaque canal).

Étapes de déroulement du scénario (Figure 6.3) :

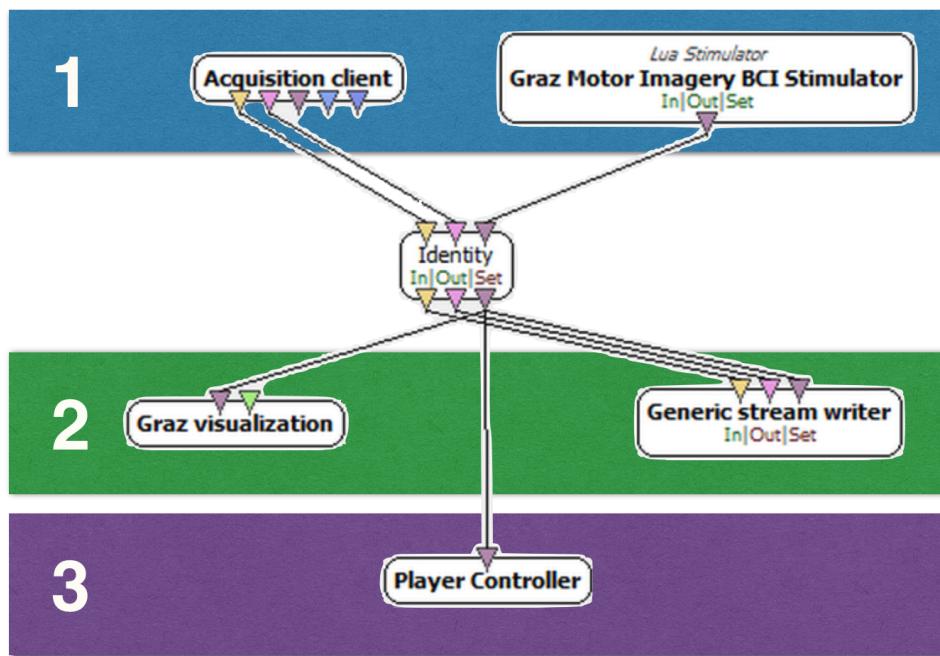


FIGURE 6.2 – Scénario 1 : Acquisition. Les signaux EEG sont acquis à partir du casque via le bloc "acquisition client". On génère des stimuli visuels grâce au script LUA contenu dans le bloc "Graz Motor Imagery BCI Simulator". Les stimuli sont affichés via le bloc "Graz Visualisation" et " Player Controller". Les données EEG sont quant à elles enregistrées avec le bloc "Generic Stream Writer", générant un fichier .ov

1. On récupère les signaux EEG bruts et les labels de stimulation enregistrés lors de l'exécution du scénario précédent.
2. On réalise un filtrage temporel des signaux EEG afin de ne conserver que les rythmes alpha et bêta du cerveau.
3. On découpe les signaux EEG en plusieurs parties, dont leur durée correspond à la durée d'un stimulus visuel. Ces sections sont triées en deux catégories : les signaux correspondant à une réponse à un stimulus gauche (flèche orientée vers la gauche) et les signaux correspondant à une réponse à un stimulus droit (flèche orientée vers la droite).
4. On envoie les deux catégories de signaux vers l'entraineur du filtre CSP.

Il faut environ 8 secondes pour que l'ordinateur calcule les paramètres du filtre (il faut utiliser la fonction de lecture rapide). A la fin de l'exécution du scénario, un message sur le terminal nous informe de la bonne réalisation de la configuration du filtre spatial (Figure 6.4).

Configuration du bloc "Generic Stream Reader" :

- File Name .ov : Indiquer le chemin vers le fichier .ov généré lors de l'exécution du scénario précédent, afin de récupérer les signaux EEG bruts et les labels de stimulation correspondants.

Configuration du bloc "Temporal Filter" :

- Type de filtre : Butterworth, Passe bande.
- Fréquences de coupure : 8Hz - 30Hz.

Configuration des blocs "Simulation Based Epoching" :

- Epoch Duration : Se réfère à la durée d'un stimulus visuel produit par le script LUA, afin de découper le signal en plusieurs parties dont la durée correspond à celle du stimulus visuel associé. La durée est ici de 4 secondes.
- Epoch Offset : Se réfère à la durée entre deux stimuli visuels. La durée est ici de 0.5 secondes.
- Stimulation to epoch from : Associe les fragments des signaux découpés au label de stimulation. Dans notre cas, il s'agira du label "OVTK_GDF_Left" pour les fragments de signaux EEG correspondant aux stimuli de gauche et le label "OVTK_GDF_Right" pour les fragments de signaux EEG correspondant aux stimuli de droite.

Configuration du bloc "CSP Spatial Filter Trainer" :

- Spatial Filter Configuration : Indiquer le chemin et le nom du fichier vers lequel le scénario doit enregistrer le fichier de configuration du CSP (extension .cfg).
- Filter Dimension : Spécifie le nombre de sorties que l'on souhaite obtenir après le filtrage spatial.

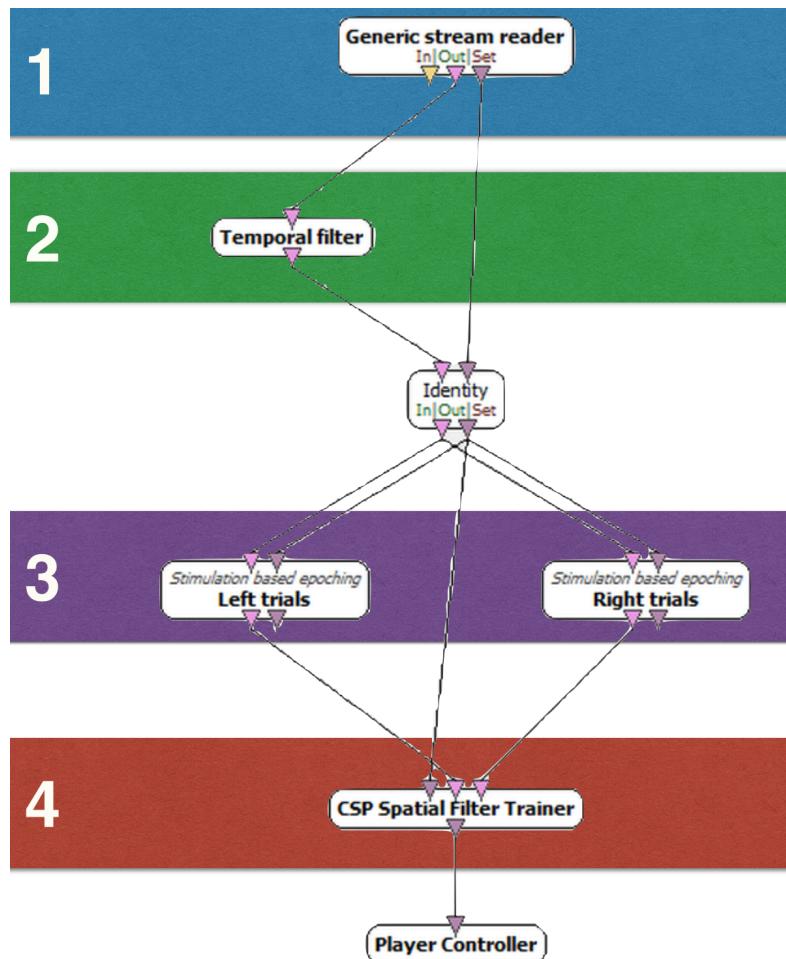
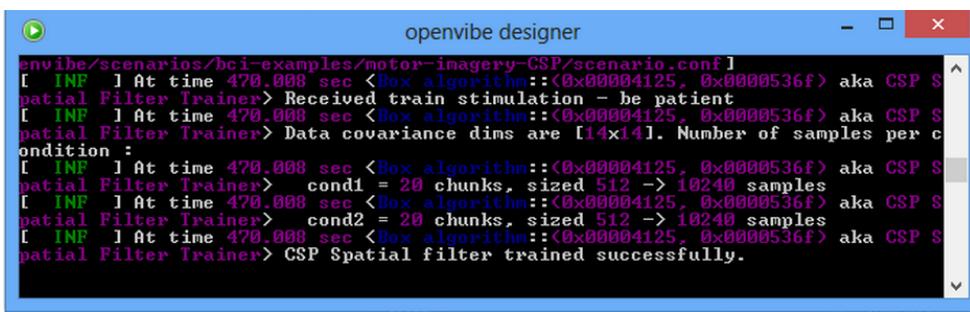


FIGURE 6.3 – Scénario 2 : Entrainement du filtre spatial. On récupère les données acquises à partir du scénario précédent (bloc "Generic Stream Reader"), puis on réalise un filtrage temporel des signaux EEG afin d'extraire les rythmes alpha et bêta. On découpe et on trie ensuite les signaux EEG en fonction des stimulations visuelles. On entraîne le CSP à partir de ces signaux découpés.



```

openvibe designer
envibe/scenarios/bci-examples/motor-imagery-CSP/scenario.conf
[ INF ] At time 470.008 sec <Box algorithm::<0x00004125, 0x0000536f> aka CSP S
patial Filter Trainer> Received train stimulation be patient
[ INF ] At time 470.008 sec <Box algorithm::<0x00004125, 0x0000536f> aka CSP S
patial Filter Trainer> Data covariance dims are [14x14]. Number of samples per c
ondition :
[ INF ] At time 470.008 sec <Box algorithm::<0x00004125, 0x0000536f> aka CSP S
patial Filter Trainer> cond1 = 20 chunks, sized 512 -> 10240 samples
[ INF ] At time 470.008 sec <Box algorithm::<0x00004125, 0x0000536f> aka CSP S
patial Filter Trainer> cond2 = 20 chunks, sized 512 -> 10240 samples
[ INF ] At time 470.008 sec <Box algorithm::<0x00004125, 0x0000536f> aka CSP S
patial Filter Trainer> CSP Spatial filter trained successfully.

```

FIGURE 6.4 – Sortie console de l’entraînement du CSP. Lorsque l’entraînement du CSP est terminé, le message "CSP Spatial filter trained successfully" apparaît.

6.2.3 Scénario 3 : Entrainement du classificateur

Ce scénario sert à entraîner le classificateur LDA (partie 4.3.3) afin que la liaison cerveau-ordinateur soit en mesure de classer les signaux EEG reçus en deux catégories : les signaux correspondants à la pensée de la main gauche, et ceux correspondants à la main droite (Figure 6.6). Le classificateur se base sur les données acquises lors de l’utilisation du premier scénario et utilise ainsi le fichier .ov qui a été généré. Il utilise également le fichier de configuration obtenu avec l’entraînement du CSP. Le scénario fournit en sortie un fichier .cfg. Il s’agit d’un fichier contenant l’ensemble des informations de configuration du classificateur LDA (Hyperplan et équation du plan).

Étapes de déroulement du scénario (Figure 6.6) :

1. On récupère les signaux EEG bruts et les labels de stimulation enregistrés lors de l’exécution du scénario 1.
2. On réalise un filtrage temporel des signaux EEG afin de ne conserver que les rythmes alpha et bêta du cerveau.
3. On réalise un filtrage spatial de type CSP, à partir de la configuration obtenue dans le scénario précédent. On obtient donc en sortie les canaux sélectionnés et pondérés par le filtre.
4. On découpe les signaux EEG en plusieurs parties, dont leur durée correspond à la durée d’un stimulus visuel. Ces sections sont triées en deux catégories : les signaux correspondant à une réponse à un stimulus gauche (flèche orientée vers la gauche) et les signaux correspondant à une réponse à un stimulus droit (flèche orientée vers la droite).
5. On recoupe de nouveau ces signaux afin d’obtenir des éléments ayant une durée de 1 seconde.
6. On réalise une analyse spectrale en décomposant le signal en série de Fourier.
7. On place les spectres des signaux dans un vecteur caractéristique.
8. On entraîne le classificateur.

À la fin de cette opération, un indice de performance est donné dans la console d’OpenViBE. Cet indice doit être supérieur à 65% si l’on veut espérer obtenir des résultats cohérents lorsque la liaison cerveau-ordinateur fonctionnera en temps réel, i.e. lors de l’executon du scénario 4 (Figure 6.5). Dans le cas contraire, il faut recommencer là partir de la phase d’acquisition (Scénario 1) pour obtenir un meilleur résultat.

Configuration du bloc "Generic Stream Reader" :

- File Name .ov : Indiquer le chemin vers le fichier .ov généré lors de l'exécution du scénario 1, afin de récupérer les signaux EEG bruts et les labels de stimulation correspondants.

Configuration du bloc "Temporal Filter" :

- Type de filtre : Butterworth, Passe bande.
- Fréquences de coupure : 8Hz - 30Hz.

Configuration du bloc "CSP Spatial Filter" :

- Sélectionner l'option "Override setting with configuration file" puis indiquer le chemin vers le fichier de configuration du CSP généré lors du scénario précédent. Les paramètres "Spatial Filter Coefficients", "Number of Output Channels" et "Number of Input Channels" se configurent alors automatiquement.

Configuration des blocs "Simulation Based Epoching" :

- Epoch Duration : Se réfère à la durée d'un stimulus visuel produit par le script LUA, afin de découper le signal en plus parties dont la durée correspond à celle du stimulus visuel associé. La durée est ici de 4 secondes.
- Epoch Offset : Se réfère à la durée entre deux stimuli visuels. La durée est ici de 0.5 secondes.
- Stimulation to epoch from : associe les fragments de signaux découpés au label de stimulation. Dans notre cas, il s'agira du label "OVTK_GDF_Left" pour les fragments de signaux EEG correspondant aux stimuli de gauche et le label "OVTK_GDF_Right" pour les fragments de signaux EEG correspondant aux stimuli de droite.

Configuration du bloc "Time Based Epoching" :

- Epoch 1 duration : Correspond à la durée des tranches de signaux. 1 seconde dans le cas présent.
- Epoch 1 intervals : Intervalle de temps entre chaque tranche de signal. 0.0625 secondes dans le cas présent.

Configuration du bloc "Spectral analysis (FFT)" :

- Spectral components : Permet de sélectionner la composante spectrale analysée. On souhaite ici utiliser le spectre d'amplitude du signal.

Configuration du bloc "Spectrum Average" :

- Ne requiert pas de configuration particulière.

Configuration du bloc "Classifier trainer" :

- Train Trigger : Permet de sélectionner le drapeau permettant de démarrer l'entraînement du classificateur. Il s'agit ici du drapeau "OVTK_Stimulationid_Train" qui est généré par les données de stimulation provenant du bloc "Generic stream reader".
- Filename to save configuration to : Permet de sélectionner le répertoire et le nom sous lequel enregistrer le fichier de configuration du classificateur (extension .cfg).
- Class 1 label : Sélectionne le label de la classe 1. Il s'agira ici du label utilisé pour la stimulation gauche (flèche orientée à gauche) "OVTK_GDF_Left".
- Class 2 label : Sélectionne le label de la classe 2. Il s'agira ici du label utilisé pour la stimulation droite (flèche orientée à droite) "OVTK_GDF_Right".
- Algorithm to use : Permet de sélectionner le type de classification à réaliser (LDA ou SVM). On décide ici d'utiliser un classificateur LDA.

```

openvibe designer
[ifier trainer] Finished with partition 1 / 5 <performance : 67.8571%>
[ INF ] At time 470.000 sec <Box algorithm::<0x66c7a9b2, 0x7da09d52> aka Class
[ifier trainer] Finished with partition 2 / 5 <performance : 68.1122%>
[ INF ] At time 470.000 sec <Box algorithm::<0x66c7a9b2, 0x7da09d52> aka Class
[ifier trainer] Finished with partition 3 / 5 <performance : 65.3061%>
[ INF ] At time 470.000 sec <Box algorithm::<0x66c7a9b2, 0x7da09d52> aka Class
[ifier trainer] Finished with partition 4 / 5 <performance : 64.5408%>
[ INF ] At time 470.000 sec <Box algorithm::<0x66c7a9b2, 0x7da09d52> aka Class
[ifier trainer] Finished with partition 5 / 5 <performance : 69.6429%>
[ INF ] At time 470.000 sec <Box algorithm::<0x66c7a9b2, 0x7da09d52> aka Class
[ifier trainer] Cross-validation test accuracy is 67.0918% <sigma = 1.88844%>
[ INF ] At time 470.000 sec <Box algorithm::<0x66c7a9b2, 0x7da09d52> aka Class
[ifier trainer] Training set accuracy is 67.5% <optimistic>

```

FIGURE 6.5 – Sortie console résultant de l'entraînement du classificateur. Une valeur correspondant en pourcentage à la précision de l'entraînement de la classification nous est fournie.

6.2.4 Scénario 4 : Utilisation en temps réel de l'interface

Ce scénario permet d'utiliser la liaison cerveau-ordinateur en temps réel. Il s'appuie sur les configurations effectuées grâce aux scénarios précédents, i.e. le filtre spatial CSP et classificateur LDA (Figure 6.7). On place dans le scénario un certain nombre de blocs permettant de visualiser les résultats obtenus ("Signal display", "Power spectrum display", "Post CSP display", "Graz Visualisation").

Étapes de déroulement du scénario (Figure 6.6) :

1. On récupère les signaux EEG à partir du casque EPOC, en temps réel (Figure : 6.8).
2. On réalise un filtrage temporel des signaux EEG afin de ne conserver que les rythmes alpha et bêta du cerveau (Figure : 6.9).
3. On réalise un filtrage spatial de type CSP, à partir de la configuration obtenue dans le scénario 2. On obtient donc en sortie les canaux sélectionnés et pondérés par le filtre (Figure : 6.10).
4. On découpe les signaux afin d'obtenir des tranches de signal ayant une durée de 1 seconde.
5. On réalise une analyse spectrale en décomposant le signal en série de Fourier (Figure : 6.11).
6. On place les spectres des signaux dans un vecteur caractéristique.
7. On réalise la classification des données à partir du classificateur entraîné grâce au scénario précédent. On obtient donc en sortie l'estimation du système quant à la pensée de l'utilisateur (Figure : 6.12).

Configuration du bloc "Acquisition Client" :

- Acquisition Server HostName : "\$AcquisitionServer_HostName".
- Acquisition Server Port : Port de connexion correspondant au casque Epoc. Valeur par défaut 1024.

Configuration du bloc "Temporal Filter" :

- Type de filtre : Butterworth, Passe bande.
- Fréquences de coupure : 8Hz - 30Hz.

Configuration du bloc "CSP Spatial Filter" :

- Sélectionner l'option "Override setting with configuration file" puis indiquer le chemin vers le fichier de configuration du CSP généré lors du scénario précédent. Les paramètres "Spatial Filter Coefficients", "Number of Output Channels" et "Number of Input Channels" se configurent alors automatiquement.

Configuration du bloc "Time Based Epoching" :

- Epoch 1 duration : Correspond à la durée des tranches de signaux. 1 seconde dans le cas présent.
- Epoch 1 intervals : Intervalle de temps entre chaque tranches de signal. 0.0625 secondes dans le cas présent.

Configuration du bloc "Spectral analysis (FFT)" :

- Spectral components : Permet de sélectionner la composante spectrale analysée. On souhaite ici utiliser le spectre d'amplitude du signal.

Configuration du bloc "Spectrum average" :

- Ne requiert pas de configuration particulière.

Configuration du bloc "Classifier processor" :

- Filename to load configuration from : Indiquer le chemin vers le fichier de configuration généré lors de l'exécution du scénario précédent.

6.3 Résultats obtenus et analyse

Au départ, notre première expérience utilisait un filtre statique de type Laplacien. Or, les résultats obtenus n'étaient pas satisfaisant (55% de performances environ). Après remplacement du filtre statique par un filtre spatial par apprentissage de type CSP, on obtient de meilleures performances. (75 à 80 % de performance avec le filtre CSP).

Deux facteurs principaux altèrent la qualité des résultats :

- **Qualité des signaux d'entrée.** Un mauvais positionnement du casque sur le crâne ou une mauvaise hydratation des électrodes peut réduire la qualité des signaux entrant dans le système, réduisant ainsi la qualité des résultats.
- **Concentration de l'utilisateur.** L'exécution du premier scénario nécessite une concentration importante de l'utilisateur. En effet, si celui-ci ne se concentre pas suffisamment sur les stimuli visuels et sur sa représentation visuelle du mouvement de ses main, la qualité de la classification sera altérée. De même si l'individu parle ou bouge durant l'acquisition des données.

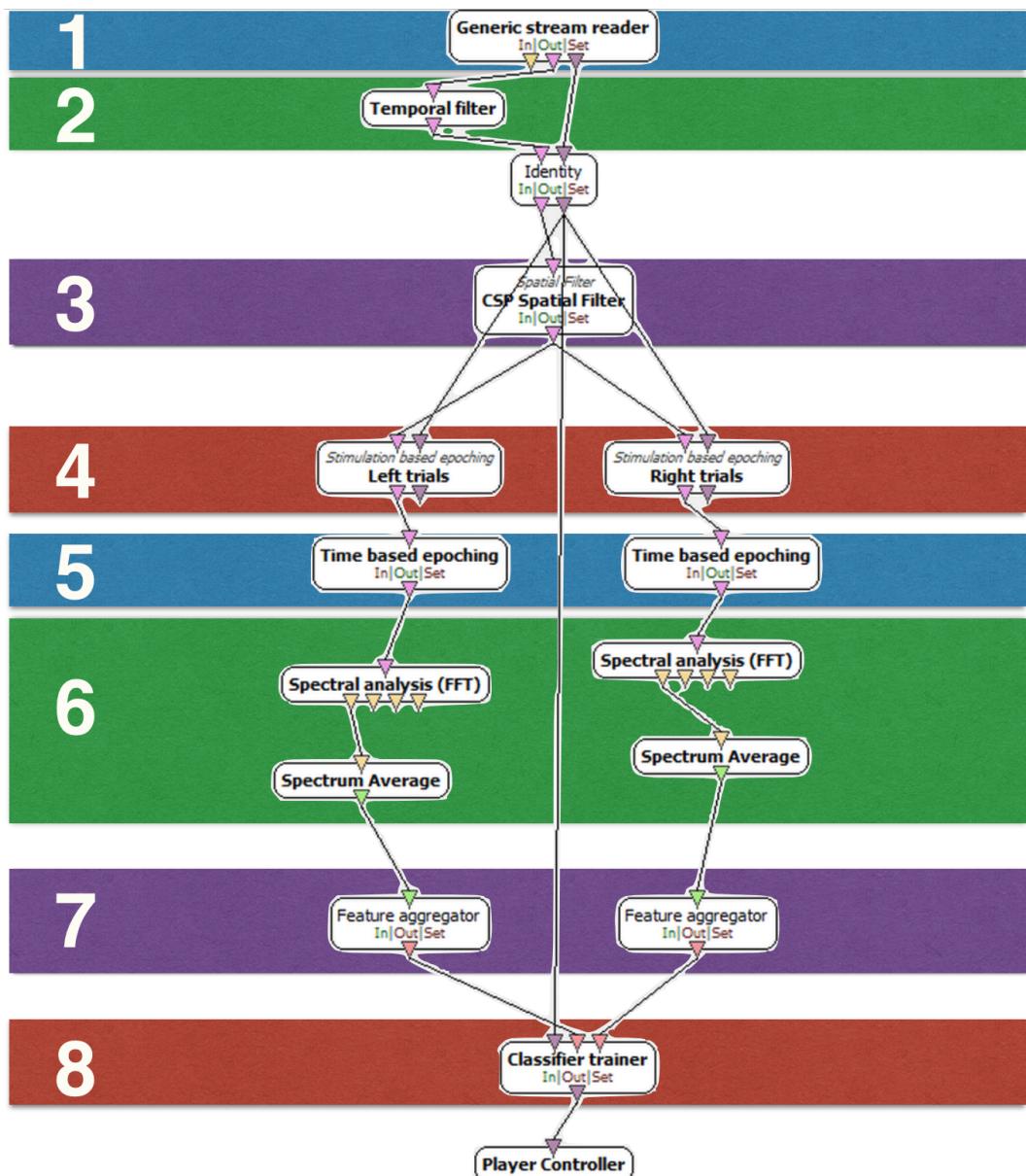


FIGURE 6.6 – Scénario 3 : Entrainement du classificateur. A partir des signaux acquis lors du scénario 1, on vient entraîner le classificateur en vue de l'utiliser pour le scénario 4 (temps réel). Pour cela, on commence par filtrer des signaux d'entrée afin d'extraire les signaux alpha et bêta du cerveau. Puis on réalise un filtrage spatial grâce au filtre CSP entraîné lors du scénario précédent. Enfin on effectue un analyse spectral des signaux filtrés pour les présenter en entrée de l'entraineur du classificateur.

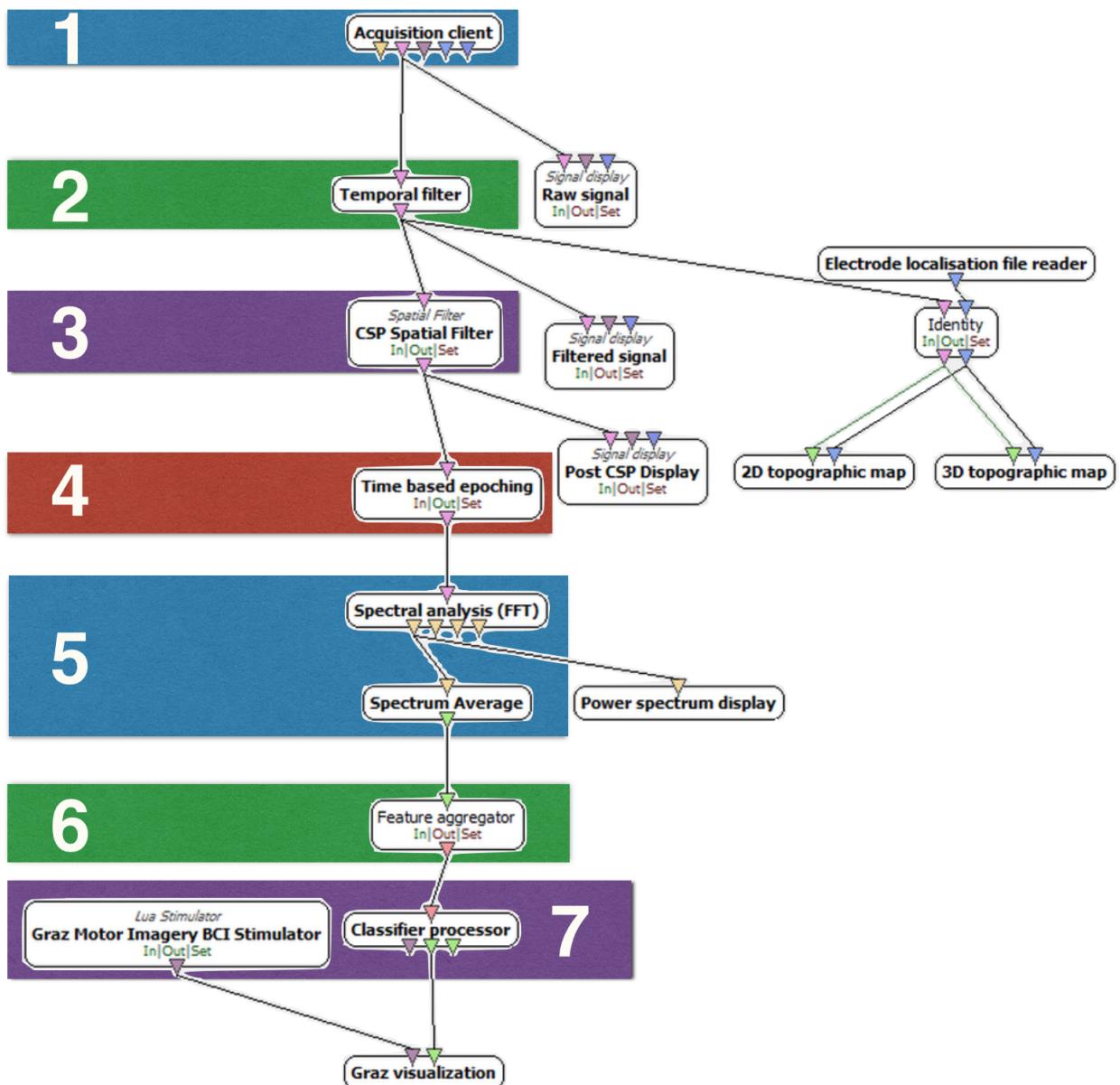


FIGURE 6.7 – Scénario 4 : Utilisation en temps réel. Celui-ci permet d'utiliser la liaison cerveau-ordinateur en temps réel. Pour cela, on enregistre en entrée les signaux provenant en temps réel du casque (bloc "acquisition client"). On effectue par la suite un filtrage temporel afin d'extraire les rythmes alpha et bêta du cerveau. On réalise à partir de ces rythmes un filtrage CSP afin de réduire le nombre de signaux en sortie. Une analyse spectrale est ensuite réalisée afin d'effectuer une classification LDA. Les données sont visualisées via les différents bloc de visualisation.

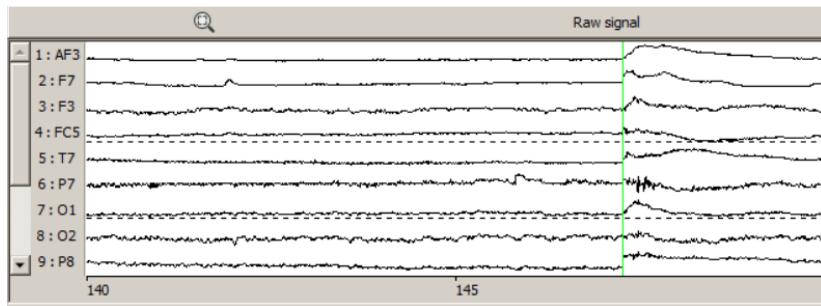


FIGURE 6.8 – Signaux bruts observés en sortie du casque Emotiv, sur les 14 canaux (électrodes).

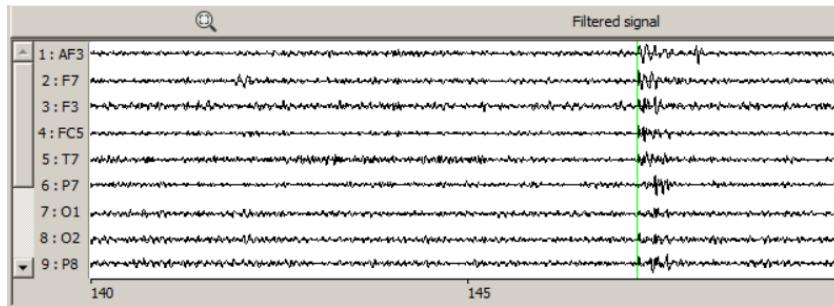


FIGURE 6.9 – Signaux filtrés temporellement afin d'extraire les rythmes alpha et bêta du cerveau.

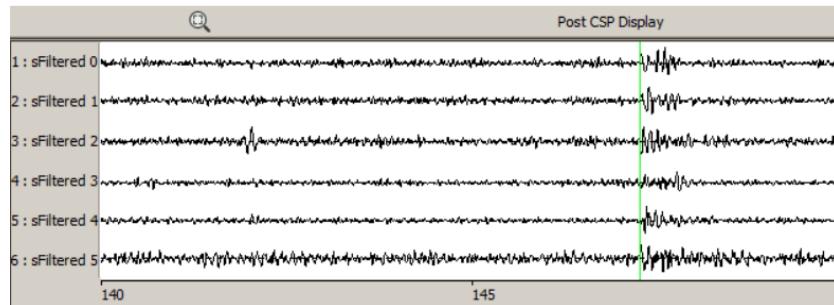


FIGURE 6.10 – Réduction du nombre de sorties grâce au filtrage CSP. On passe de 14 entrées à 6 sorties.



FIGURE 6.11 – Analyse spectrale des signaux via une transformation de Fourier.

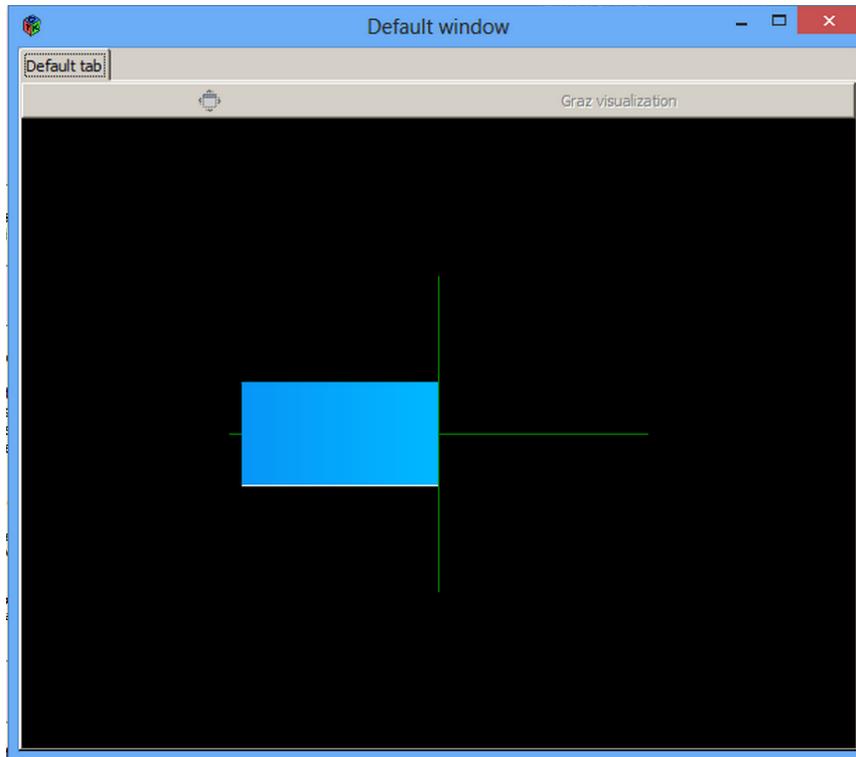


FIGURE 6.12 – Sortie visuelle temps réel après classification. La direction du curseur bleu correspond à la pensée de l'utilisateur : Si le curseur se dirige vers la gauche, cela signifie que le système estime que la personne pense à un mouvement de sa main gauche. Si au contraire le curseur se dirige vers la droite, cela signifie que le système estime que la personne pense à un mouvement de sa main droite. Plus le curseur s'éloigne du centre de l'image, plus la certitude du système est élevée.

Chapitre

7

Intégration du projet dans le cadre de cours

On souhaite intégrer notre projet dans le cadre de plusieurs cours enseignés par Jean Rouat.

- **Cours gradué de techniques avancées en traitement des signaux.** Ce cours enseigne les différentes méthodes de représentation d'un signal dans le domaine spectral (transformée de Fourier, transformation en ondelettes, etc.), ainsi que les outils de filtrage spectral (Analyse en composantes principales, analyse en composantes indépendantes). Cours enseigné à la session d'automne.
- **APP d'intelligence probabiliste.** Ce cours s'intéresse aux différentes méthodes de classification, et notamment les méthodes paramétriques (Bayesiennes) et non paramétriques (k-moyen, SVM, etc.). Cours enseigné à la session d'automne.
- **Cours de neurosciences computationnelles.** Ce cours enseigne les principes de base de la physiologie des réseaux neuronaux et leurs modélisations. Il offre également une vision d'ensemble des différents domaines d'application. Cours enseigné à la session d'hiver.

On propose donc différents travaux pratiques permettant aux étudiants de mettre en application leurs connaissances acquises au cours de la matière.

7.1 Choix des outils mathématiques d'apprentissage

Afin d'intégrer notre liaison cerveau-ordinateur dans les différents cours, on souhaite exporter les données traitées dans OpenViBE vers un environnement de développement mathématique, afin que les élèves puissent travailler sur les signaux. Trois environnements s'offrent à nous :

- **Matlab.** Il s'agit d'un langage de programmation haut niveau, associé à un environnement de développement portant le même nom. L'avantage de cet outil est qu'il est assez simple à apprêhender et que la plupart des étudiants l'ont déjà utilisé à de nombreuses reprises. OpenViBE offre la possibilité d'exécuter du code Matlab sous son environnement et de l'utiliser dans la chaîne de traitement des signaux EEG. Cela nécessite cependant d'utiliser Matlab 32 bits sur les ordinateurs des laboratoires, or ceux-ci possèdent uniquement la version 64 bits. Un autre de ses inconvénients est qu'il s'agit d'un logiciel payant.
- **Octave.** Octave est un langage de programmation mathématique de haut-niveau.

Il a l'avantage d'être totalement gratuit. Cependant, OpenViBE ne propose aucune solution d'intégration d'Octave directement dans l'interface du logiciel.

- **Python.** Il s'agit d'un langage de programmation haut-niveau. OpenViBE propose une solution d'intégration de Python dans son environnement afin de l'utiliser dans la chaîne de traitement des signaux. Cependant, cette intégration est complexe à mettre en place et nécessite de la part des élèves des connaissances autre que la simple maîtrise du langage.

Aucun des environnements proposés ne semble être intégrable directement dans OpenViBE. On propose donc comme alternative d'exporter les données traitées sous OpenViBE vers un fichier CSV. Ce format représente des données tabulaires sous forme de valeurs séparées par des virgules. Il a donc l'avantage d'être lisible directement par l'Homme et de nombreux logiciels de traitement de données, comme Matlab ou même Excel. Nous proposons donc d'utiliser Matlab comme environnement de travail pour la réalisation des travaux pratiques.

7.2 Cours de neurosciences computationnelles

7.2.1 Résumé du travail pratique proposé

Le travail pratique que nous proposons aux étudiants dans le cadre du cours de neurosciences computationnelles correspond plus à un travail d'observation et de compréhension des différentes étapes de traitement des signaux électroencéphalographiques. On leur demande d'étudier le lien entre les connaissances physiologiques acquises lors du cours et les différents aspects techniques de l'interface cerveau-ordinateur.

Pour cela, on fournit le scénario 4 (Figure 7.1) réalisé dans le cadre de l'expérience simple (partie 6). L'entraînement du classificateur et du filtre spatial CSP ayant été réalisé en amont, on demande donc aux étudiants de s'intéresser uniquement à l'exécution en temps réel de l'interface cerveau-ordinateur. Plusieurs outils de visualisation ont été intégrés au scénario, afin que les étudiants puissent observer l'activité cérébrale (Figure 7.2) :

- **2D topographic map.** Permet d'observer l'activité cérébrale de l'individu portant le casque sur une représentation en 2 dimensions du crâne.
- **3D topographic map.** Permet d'observer l'activité cérébrale de l'individu portant le casque sur une représentation en 3 dimensions du crâne.
- **Signal display.** Permet d'observer les signaux EEG à différentes étapes de leur traitement (en sortie du casque EEG, après filtrage temporel).
- **Post CSP Display.** Permet d'observer les signaux et plus particulièrement la sélection des canaux (signaux de sortie des électrodes) après le filtrage spatial de type CSP.
- **Power spectrum display.** Permet d'observer les différents spectres des signaux après transformation en série de Fourier.
- **Graz Visualisation.** Permet d'observer les résultats de l'expérience après classification des données.

7.2.2 Préparation du travail pratique par le professeur

Le TP reposant uniquement sur le scénario 4, on propose ici deux approches pour la préparation du cours de neurosciences.

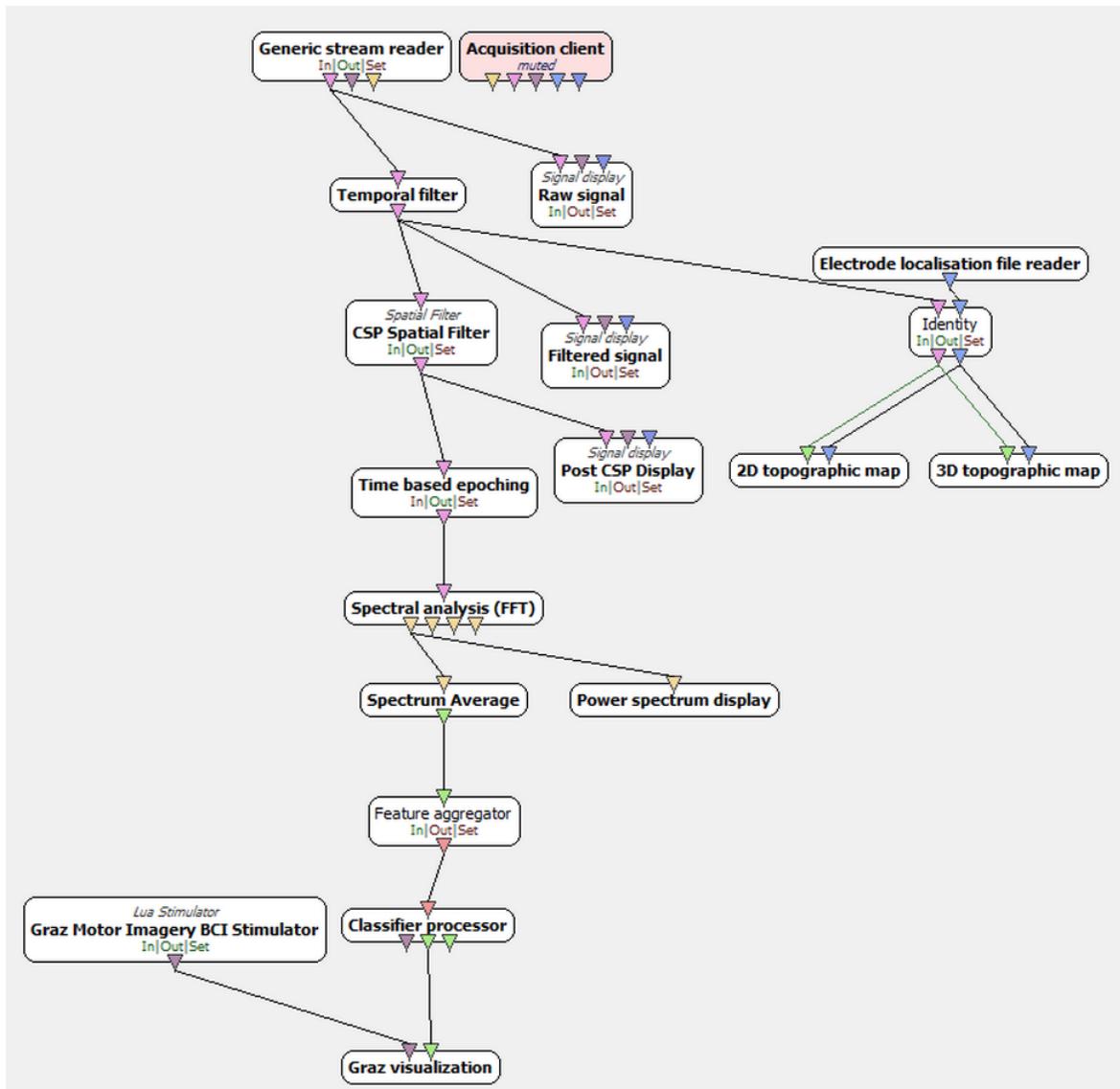


FIGURE 7.1 – Scénario temps réel proposé dans le cadre du cours de neurosciences computationnelles.

Travail en groupe

1. Le professeur réalise en amont l'acquisition des signaux EEG (scénario 1), l'entraînement du filtre CSP (scénario 2) et l'entraînement du classificateur LDA (scénario 3).
2. La classe ne disposant que d'un seul casque, le professeur exécutera devant l'ensemble des élèves le scénario 4 en l'utilisant grâce au bloc "Acquisition Client". Cela nécessite donc de configurer préalablement OpenViBE et le casque EPOC comme présenté en partie 5.
3. Les fichiers de configuration générés par les scénarios 2 et 3 sont utilisés pour exécuter le scénario 4 lors du cours. Pour cela, on renseigne le chemin vers les fichiers de configuration dans les blocs "CSP spatial Filter" et "Classifier Processor".
4. Vérifier dans le bloc "Graz Visualisation" que la case "Show instruction" est bien décochée.

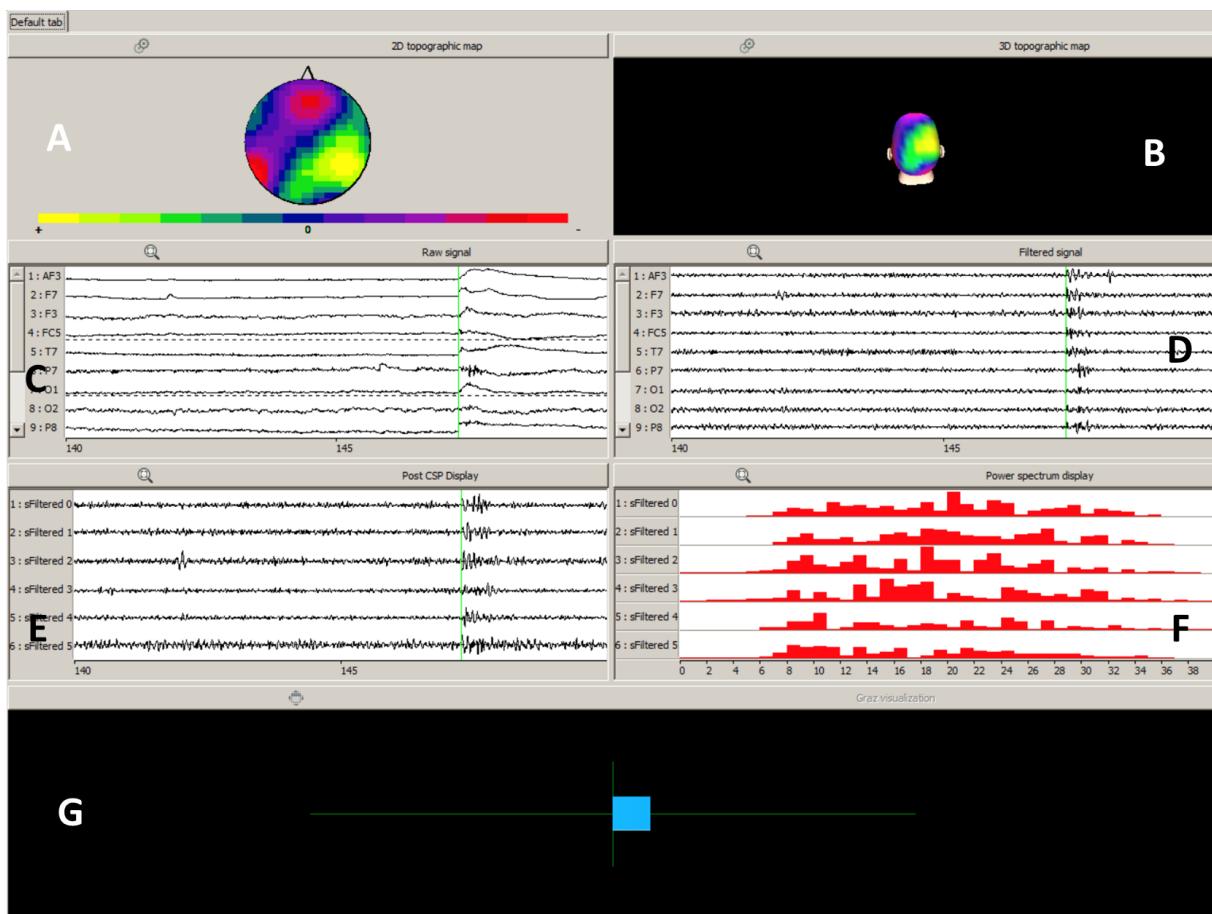


FIGURE 7.2 – Interface de visualisation proposée aux étudiants du cours de neurosciences.
 (A) Visualisation topographique de l'activité cérébrale en 2D. (B) Visualisation topographique de l'activité cérébrale en 3D. (C) Visualisation des signaux en sortie du casque, i.e. les signaux non traités. (D) Visualisation des signaux après filtrage temporel. (E) Visualisation des signaux après filtrage spatial CSP. (F) Visualisation des spectres fréquentiels après FFT. (G) Fenêtre de visualisation

Travail individuel

1. Les élèves sont dans un premier temps invités à télécharger OpenViBE.
2. On fournit alors à l'ensemble des élèves le scénario 4, ainsi que les fichiers de configuration du filtre spatial CSP et du classificateur LDA, générés par nos soins.
3. La classe ne disposant que d'un casque, on propose d'utiliser en entrée du scénario 4 un fichier contenant un enregistrement de signaux EEG (extension .ov), lisible avec le bloc "Generic Stream Reader". Chaque élève peut alors réaliser individuellement le travail pratique.
4. Vérifier dans le bloc "Graz Visualisation" que la case "Show instruction" est bien cochée, afin que l'élève puisse visualiser les consignes que l'individu avait en entrée lors de l'enregistrement.

Un sujet de travail pratique à destination des élèves a été rédigé et placé en annexe. Il contient la procédure pour réaliser la manipulation, ainsi que des questions portant sur l'interface cerveau-ordinateur.

7.3 Cours de traitement de signal avancé

7.3.1 Résumé du travail pratique proposé

Le travail pratique que nous proposons aux étudiants du cours de traitement du signal avancé consiste à réaliser une partie de la chaîne de traitement des signaux EEG de l'expérience (partie 6), sous l'environnement Matlab. L'objectif est de leur permettre de mettre en application leurs connaissances (filtrage spatial, filtrage temporel, FFT). Le reste de la chaîne de traitement étant effectué sous OpenViBE, on exporte donc les données de OpenViBE vers Matlab grâce au format CSV. Une fois les signaux traités sous Matlab, on exporte à nouveau ces données vers OpenViBE. Cette méthode de transfert entre les deux logiciels nous oblige cependant à ne pas utiliser l'interface cerveau-ordinateur en temps réel.

7.3.2 Développement de nouveaux scénarios

Pour ce cours, nous avons implémenté trois nouveaux scénarios. Un seul sera présenté aux étudiants, les deux autres servant à la préparation du travail pratique :

- **Scénario d'acquisition et de filtrage temporel.** (Figure 7.3) L'objectif de ce scénario est de pouvoir récupérer les données en sortie des électrodes du casque EEG filtrées temporellement, afin d'en extraire les rythmes alpha et bêta. Son principe de fonctionnement reste le même que le scénario 1 présenté dans la partie 6.2.1. Les signaux filtrés ainsi que les données de stimulation sont exportées dans deux fichiers au format .csv. Les données de stimulation correspondent aux stimulations visuelles envoyées à l'utilisateur lors de l'enregistrement des données EEG. Le fichier CSV contenant ces données de stimulation sera utilisé par la suite pour l'entraînement du classificateur. Le fichier CSV contenant les données EEG sera quant à lui utilisé à deux reprises, lors de l'entraînement du classificateur et lors de l'exécution de la liaison cerveau-ordinateur.

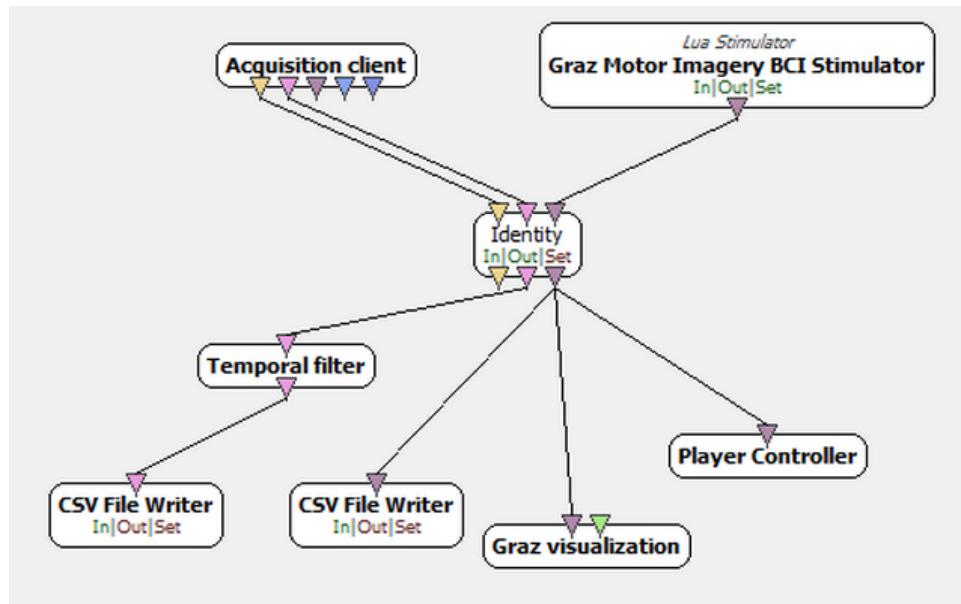


FIGURE 7.3 – Scénario d'acquisition avec sortie au format CSV

- **Scénario d'entraînement du classificateur.** Il permet d'entraîner le classificateur de type LDA (Figure 7.4). Le principe de fonctionnement reste identique au scénario présenté dans la partie 6.2.3. Le scénario nécessite à son entrée deux fichiers

CSV. Le premier contient les données reçues du casque lors de la phase d'acquisition après filtrage spatial ICA ou PCA sous Matlab. Le deuxième fichier contient les données de stimulation obtenues lors de la phase d'acquisition.

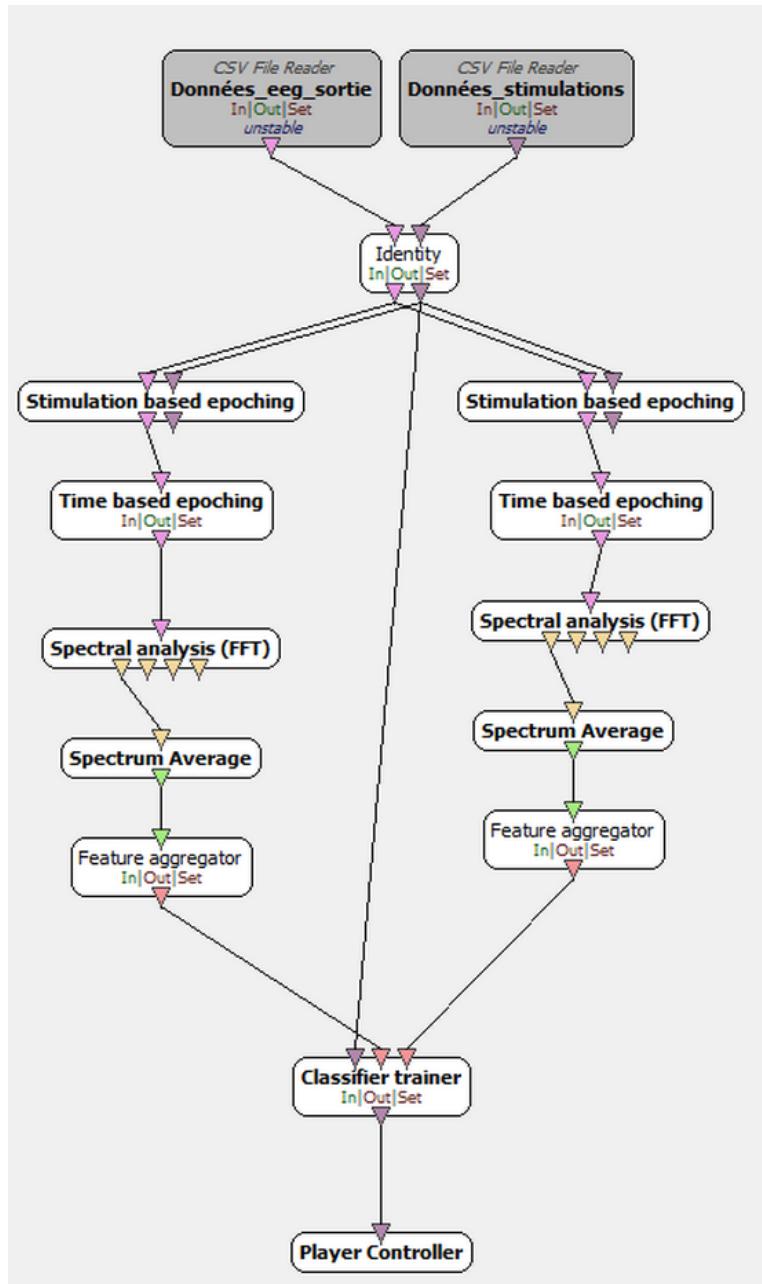


FIGURE 7.4 – Scénario d'entraînement du classificateur à partir de fichiers .csv

- **Scénario d'exécution de l'interface cerveau-ordinateur.** Il permet d'exécuter la liaison cerveau-ordinateur (Figure 7.5). Pour cela, on utilise le fichier CSV issu du premier scénario (scénario d'acquisition) filtré spatialement avec Matlab. Le principe et la configuration du reste des blocs sont les mêmes que ceux détaillés dans la partie 6.2.4

7.3.3 Fonctions Matlab et corrections

Plusieurs fonctions Matlab ont été écrites par nos soins et sont fournies au professeur et aux élèves pour ce travail pratique :

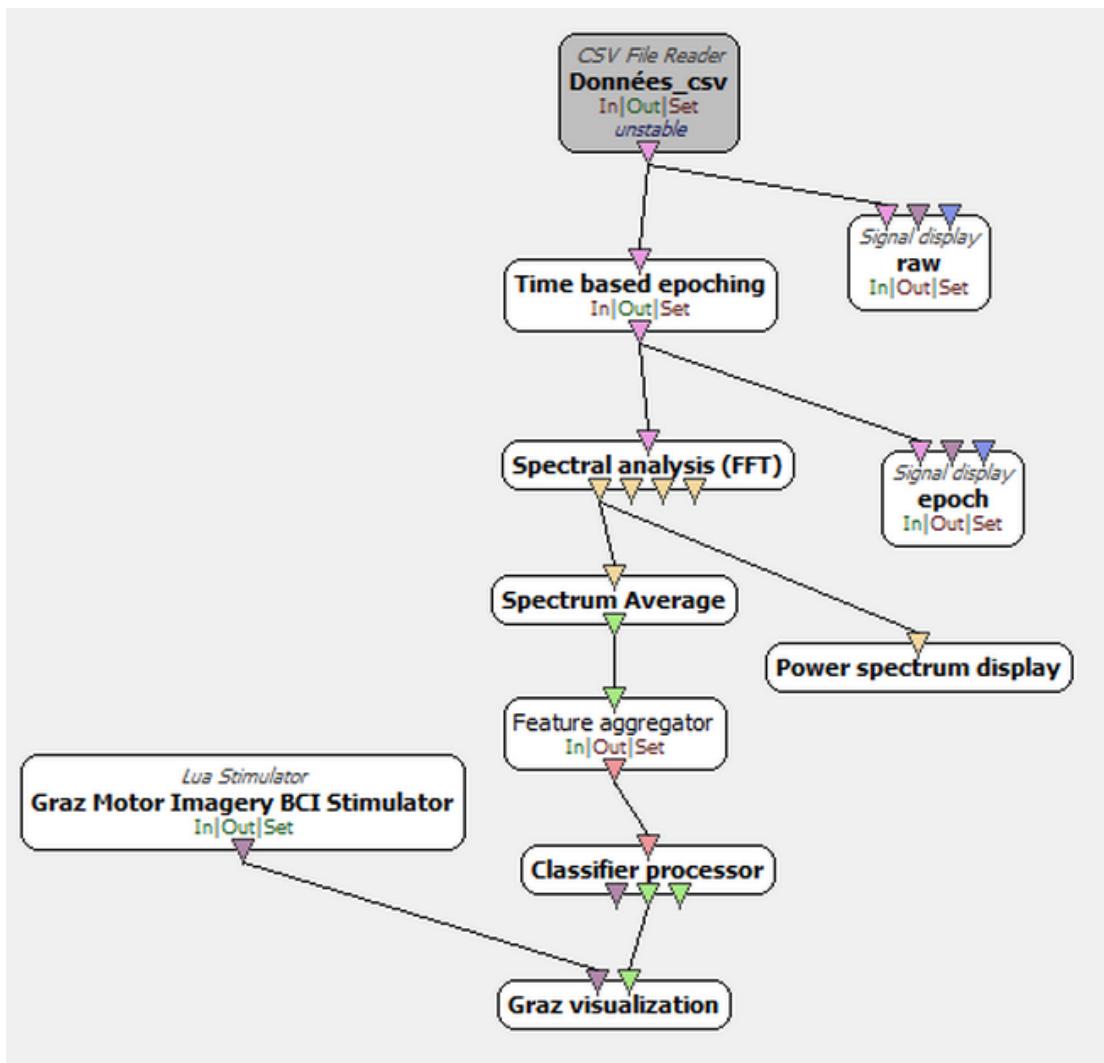


FIGURE 7.5 – Scénario d’observation à partir de fichiers .csv issus de Matlab

- Une fonction de lecture pour les fichiers CSV. Permet de lire les données EEG contenues dans un fichier de type .csv. Disponible en annexe.
- Une fonction d’écriture pour les fichiers CSV. Permet d’écrire dans un fichier .csv. Disponible en annexe.
- Un sujet pour réaliser un filtrage ICA. Contient la boucle de traitement à effectuer. La fonction permettant de faire un filtrage ICA reste à coder par les étudiants.
- La correction pour réaliser un filtrage ICA. Disponible en annexe.
- Un sujet pour réaliser un filtrage PCA. Contient la boucle de traitement à effectuer. La fonction permettant de faire un filtrage PCA reste à coder par les étudiants.
- La correction pour réaliser un filtrage PCA. Disponible en annexe.

7.3.4 Préparation du travail pratique par le professeur

Le TP reposant uniquement sur le scénario d’exécution de l’interface cerveau-ordinateur, on propose deux approches pour la préparation du cours de neurosciences.

Approche 1

Dans cette approche, le professeur réalise en amont du TP l'acquisition des données pour l'entraînement du classificateur, l'acquisition des données pour l'exécution de l'interface cerveau-ordinateur et l'entraînement du classificateur.

1. La première étape consiste à récupérer des données à partir du casque EEG et les données de stimulation. Pour cela, on utilisera le scénario d'acquisition. L'exécution du scénario génère alors deux fichiers CSV : 1 fichier "données_entrée.csv" contenant les données EEG, 1 fichier "données_stimulations.csv" l'autre les données de stimulation. Vérifiez dans le bloc "Graz Visualisation" que la case "Show instructions" est bien cochée.
2. Le professeur est ensuite invité à exécuter le code Matlab "cours_signal_pca.m" ou "cours_signal_ica.m" afin de réaliser un filtrage spatial des données EEG. Veuillez pour cela à placer le fichier "données_entrée.csv" généré dans l'étape précédente dans le même répertoire de travail que le code Matlab. Un fichier CSV "données_EEG_sortie.csv" contenant les signaux EEG filtrés est alors généré.
3. On doit à présent entraîner le classificateur LDA. Pour cela, on utilise le scénario "entraînement du classificateur". On demande de fournir en entrée du scénario le fichier "données_EEG_sortie.csv" généré précédemment ainsi que le fichier "données_stimulations.csv" généré lors de l'exécution du scénario "acquisition". L'exécution du scénario génère un fichier de configuration en sortie "Classifier.cfg".
4. Enfin, on demande d'exécuter de nouveau le scénario d'acquisition afin d'obtenir des données EEG différentes de celles utilisées pour l'entraînement du classificateur afin d'exécuter l'interface cerveau-ordinateur. On veillera cette fois-ci à décocher la case "Show instructions" dans le bloc "Graz Visualisation" afin de ne pas être influencé par la présence des stimulations lors de l'acquisition des données. L'utilisateur est alors invité à penser à sa main gauche ou sa main droite chaque fois que le repère blanc apparaît dans la fenêtre de visualisation (Celui-ci n'apparaît qu'au bout de 30 secondes).
5. Le professeur fournira à ses élèves le fichier de configuration du classificateur ("Classifier.cfg"), ainsi que le dernier fichier "données_entrée.csv" généré. Cela permettra aux étudiants d'utiliser le scénario "execution de l'interface cerveau-ordinateur".

Approche 2

Dans cette approche, le professeur n'a pas besoin de réaliser un travail préparatoire en amont du TP (entraînement classificateur, acquisition de données EEG). Celui-ci fournit simplement à ses étudiants le fichier de configuration du classificateur et le fichier "données_entrées.csv". Cela leur permettra d'utiliser le scénario "execution de l'interface cerveau-ordinateur".

Chapitre 8

Bilan du projet

8.1 Connaissances acquises

Ce projet captivant, de part par son originalité et son aspect pédagogique, nous a permis de mettre en application nos acquis en traitement du signal, en informatique et en neurosciences. D'un point de vue théorique, la pluridisciplinarité du sujet nous a offert l'opportunité d'acquérir de nouvelles connaissances théoriques (traitement de signaux EEG, physiologie du cerveau, etc.). Celui-ci nous a également permis d'apprendre à utiliser de nouveaux outils, comme un casque électroencéphalographique (EPOC d'Emotiv) ou encore le logiciel OpenViBE, permettant de réaliser des interfaces cerveau-ordinateur. Enfin, c'est véritablement au niveau de la méthodologie de recherche, de la gestion de projet et de l'autonomie que nous pensons avoir le plus progressé. Notons que celui-ci nous a notamment permis d'apprendre à utiliser l'outil de traitement de texte L^AT_EX, afin de rédiger un rapport répondant aux normes IEEE.

8.2 Problèmes rencontrés

Notre travail devait initialement s'appuyer sur les travaux réalisés par un étudiant de la session d'automne 2014. Cependant, ses résultats ne nous semblaient pas pertinents dans le cadre de notre projet. De ce fait, nous avons décidé de reprendre le projet depuis le départ (notamment la partie portant sur le choix du matériel et du logiciel, ainsi que les traitements des signaux).

D'un point de vue théorique, les connaissances que nous avions en traitement de signaux EEG et en intelligence artificielle probabiliste étaient quasiment nulles (notamment sur l'aspect filtrage spatial et méthodes de classification). Cependant, nous pensons avoir suffisamment renforcé nos connaissances dans ces domaines.

Nous avons également rencontré certaines difficultés d'ordre pratique, notamment lors de l'intégration de notre projet dans le cours de traitement de signal avancé. Nous souhaitions en effet utiliser des outils mathématiques tel que Matlab ou Python. Cependant, leur intégration avec OpenViBE s'est avérée difficile à mettre en place (intégration de Matlab possible uniquement avec la version 32 bits, utilisation de Python trop complexe dans le cadre d'un cours). Nous sommes parvenus à trouver une solution qui consiste en l'exportation des données d'OpenViBE vers le format CSV, puis en les traitant dans Matlab. Cette manière de gérer les données EEG nous oblige cependant à ne plus réaliser la liaison cerveau-ordinateur en temps réel. Enfin, Nous avons également été confronté à quelques contraintes matérielles. La commande (et la livraison) du casque EPOC en Mars nous

a obligé à implémenter l'interface cerveau-ordinateur tardivement (le travail sur l'aspect théorique du sujet a donc été privilégié dans un premier temps).

8.3 Pistes d'amélioration et poursuite de nos travaux

Par manque de temps, nous n'avons malheureusement pas pu intégrer notre expérience dans le cadre du cours d'intelligence artificielle, donné à la session d'automne. Il aurait également été intéressant d'utiliser certains outils de neurosciences computationnelles dans le cadre de notre projet, notamment pour les éléments de traitement du signal comme les filtres spatiaux ou les méthodes de classification (utilisation de réseaux neuronaux avec apprentissage).

Conclusion

Nous avons eu la chance de réaliser notre projet de recherche et développement (GIN956) au sein du laboratoire de recherche en neurosciences NECOTIS, sous la tutelle de Jean Rouat. Cette opportunité nous a permis de bénéficier d'une première expérience dans le monde des neurosciences, et plus particulièrement dans la conception d'une interface cerveau-ordinateur. À cette occasion, nous avons pu acquérir de nouvelles connaissances théoriques (traitement de signaux EEG, physiologie du cerveau, etc.), pratiques (utilisation d'un casque EEG, utilisation d'un logiciel de création de liaisons cerveau-ordinateur) et méthodologiques (gestion de projet, autonomie, apprentissage à l'utilisation de L^AT_EX, etc.)

Dans un premier temps, nous avons étudié les différents aspects physiologiques du cerveau qui nous seraient utiles dans la réalisation de l'interface cerveau-ordinateur (organisation corticale, différents rythmes du cerveau, etc.).

Nous avons ensuite réalisé une étude comparative des différents systèmes EEG proposés sur le marché (casques EEG, électrodes, etc). Celle-ci nous a permis d'arrêter notre choix sur le casque EPOC de la société Emotiv. Nous avons également étudié les différents logiciels libres de droit disponibles, permettant de réaliser une liaison cerveau-ordinateur. OpenViBE se trouve être l'outil informatique le plus adapté pour la réalisation de la BCI dans le cadre de notre projet.

Dans un troisième temps, nous avons étudié différentes techniques de traitement de signaux EEG, comme les méthodes de filtrage temporel, les filtrages spatiaux ou encore les différentes techniques de classification caractéristiques.

Nous présentons par la suite le fonctionnement du logiciel OpenViBE et du casque EPOC, ainsi que l'ensemble des outils s'y rattachant.

A partir de l'ensemble des études réalisées précédemment, on réalise une expérience interactive. Celle-ci s'appuie sur une chaîne de traitement et d'analyse des signaux EEG composée d'un filtrage temporel passe-bande de type Butterworth, un filtrage spatial de type CSP et une classification des caractéristiques des signaux de types LDA.

Enfin, on intègre les différents éléments de notre projet dans le cours de neurosciences computationnelles et de traitement du signal avancé, en s'appuyant sur des travaux pratiques (réalisation d'un filtrage spatial de type PCA/ICA) réalisés sous l'environnement Matlab.

Annexes

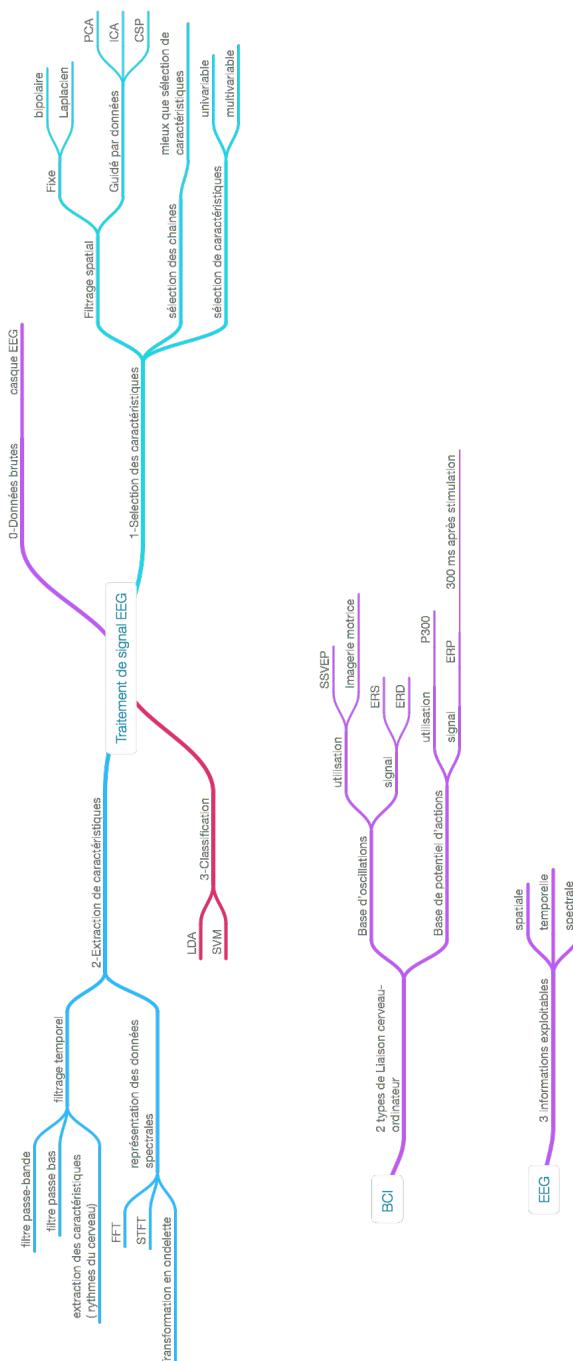
Tableau comparatif des différents casques EEG proposés sur le marché

Généralités	Nom	MindWave	EPOC	capEEG100c	Enobio	g.SAHARA
	Marques	NeuroSky	Emotiv	Biopac	Neuroelectrics	g.tech
Electrodes	Nombre électrodes	1	14 (plus CMS/DRL references, P3/P4 locations) AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4	16 or 32	8 or 20	
	Type d'électrodes	gel ou sèches	gel ou sèches	gel	gel	sèches
Électronique	Communication	Câble, sans-fil	Câble, sans-fil	câble	Câble, sans fil	Câble, sans fil
	Signal input Range	1mV pk-pk	8,4mV		0.6mV	
Prix		79\$	base : 399 \$ + SDK : 300 \$ + License éducation (25 postes) : 1799 \$	Seulement les électrodes à acheter	\$13k	1k\$
Commentaires		Précision des signaux en sortie médiocre	SDK à acheter pour utiliser les données brutes. Possibilité d'utiliser librairie OpenVibe	Typically, EL503 electrodes are recommended for evoked response measurements. Use two shielded electrodes (LEAD110S) for the signal inputs and one unshielded electrode (LEAD110) for ground. If hair is present, disposable electrodes don't work very well for scalp attachment, and you should use electrode gel (GEL100) and tape the electrode lightly in place or use a conductive adhesive paste (like Ten20® or Collodion HV®)	http://en.wikipedia.org/wiki/Enobio	g.Sahara + achat du casque adapté, g.GAMMA + Amplificateur USBamp 1k\$
				http://ixil.izt.uam.mx/lidb2/archivosequipos/bseeg100c_oe.pdf		
				Price 90\$: https://www.utechproducts.com/index.php?vmchk/571-Field-medical-or-laboratory-medical-equipment-kits-or-related-products/959567-23337324-ECG-DISPOSABLE-ELECTRODES-PK/100.html		
Amplificateur intégré						
Amplificateur	Gain	2000		Voir EEC100c		
	Sampling rate	512Hz	128 SPS		500SPS	
	Résolution	12 bits	14 bits		24bits	
	Bandwidth	3Hz to 100Hz	0.2 - 45 Hz digital notch filters at 50Hz and 60Hz		0 to 125 Hz	0.1-40 Hz
	filter	50 and 50Hz environment filtering	Built in digital 5th order Sinc filter		EMI filter 40Hz	
	Noise				< 1 uV RMS	

Tableau comparatif des différents amplificateurs proposés sur le marché

NOM	EEC100c	Model 15LT Bipolar		
Marque	Biopac	Grass Model		
Gain	5000,10000,50000	50 to 200 000		
Sampling rate				
Résolution				
Bandwidth	35Hz, 100Hz Low pass filter 0.1Hz, 1.0 Hz High pass filter	Low pass filter 0.01, 0.1, 0.3, 1, 3, 10, 30 and 100 Hz High Pass filter 0.03, 0.1, 0.3, 1 and 6 kHz		
filter	Notch filter: 50dB rejection @ 50/60Hz	Notch-type filter (50/60 Hz) jumper selectable		
Noise voltage	0.1µV rms - (0.1–35Hz)	3 µV peak to peak, referred to input, inputs shorted, 100 Hz bandwidth 10 µV peak to peak, referred to input, inputs shorted, 6 kHz bandwidth		
Output range	+/-10V			
Input Range	G5000 -> +/-2V G10000 -> +/-1V G20000 -> +/-0.5V G50000 -> +/-0.2V			
Price ?	Déjà acheté			

Carte heuristique



Sujet filtrage ICA

```
%##### SUJET DE TP DE COURS DE TRAITEMENT DE SIGNAL AVANCE #####
%##### ICA #####
%### Florian Lepont, Romain Richard### Avril 2015#####

close all
clear all

% LECTURE DES DONNEES A PARTIR DU FICHIER CSV
donnees_eeg = lecture_csv('données_entrée.csv')

% ANALYSE EN COMPOSANTES INDEPENDANTES
sources = ica(donnees_eeg) % on récupère les données du fichier

% AFFICHAGE DES DONNNEES
subplot(2,2,1)
plot(donnees_eeg')
xlabel('temps (s)')
ylabel('amplitude (mV)')

subplot(2,2,2)
plot(sources')
xlabel('temps (s)')
ylabel('amplitude (mV)')

subplot(2,2,3)
hist(donnees_eeg')
xlabel('amplitude (mV)')
ylabel('nombres de données')

subplot(2,2,4)
xlabel('temps (s)')
ylabel('amplitude du signal (mV)')

hist(sources')
xlabel('amplitude (mV)')
ylabel('nombres de données')

% CREATION DU FICHIER DE SORTIE
ecriture_csv( 'données_entrée.csv', 'données_EEG_sortie.csv', sources)
```

Correction filtrage ICA

```
%#####CORRECTION FONCTION COURS TRAITEMENT SIGNAL AVANCE#####
%#####          ICA      #####
%### Florian Lepont, Romain Richard##### Avril 2015#####

% Bibliographie : "A Tutorial on Independant Component Analysis" de Jonathon Shlens

function sources = ica(donnees)
% Analyse en composantes indépendantes.
% ENTREE : données à traiter
% SORTIE : Matrice de transformation linéaire, signaux sources calculés

[M,N] = size(donnees);

% Translation de la moyenne des données vers 0.
mn = mean(donnees,2);
donnees = donnees - repmat(mn,1,N);

% Calcul de la matrice de covariance
covariance = 1 / (N-1) * (donnees * donnees');

% Calcul des valeurs propres et des vecteurs propres de la matrice de covariance.
[E,D] = eig(covariance);

%Calcul des données décoréllées et normalisation.
X = sqrtm(pinv(D))*E'*donnees;

%Calcul de la décomposition en valeurs singulières. (calcul de la transformation linéaire - rotation - V).
[V,s,u] = svd((repmat(sum(X.*X,1),M,1).*X)*X');

%Calcul de la transformation linéaire W
W = V * sqrtm(pinv(D)) * E';

%Calcul des signaux sources
sources = W * donnees;
```

Sujet filtrage PCA

```
%##### SUJET DE TP DE COURS DE TRAITEMENT DE SIGNAL AVANCE #####
%##### PCA #####
%### Florian Lepont, Romain Richard### Avril 2015#####

close all
clear all

% LECTURE DES DONNEES A PARTIR DU FICHIER CSV
donnees_eeg = lecture_csv('données_entrée.csv')

% ANALYSE EN COMPOSANTES PRINCIPALES
% On récupère les données du fichier
[signaux_pca,composantes_principales,valeurs_propres] = pca(donnees_eeg)

% AFFICHAGE DES DONNNEES
subplot(2,2,1)
plot(donnees_eeg')
xlabel('temps (s)')
ylabel('amplitude (mV)')

subplot(2,2,2)
plot(signaux_pca')
xlabel('temps (s)')
ylabel('amplitude (mV)')

subplot(2,2,3)
hist(donnees_eeg')
xlabel('amplitude (mV)')
ylabel('nombres de données')

subplot(2,2,4)
xlabel('temps (s)')
ylabel('amplitude du signal (mV)')

hist(signaux_pca')
xlabel('amplitude (mV)')
ylabel('nombres de données')

% CREATION DU FICHIER DE SORTIE
ecriture_csv( 'données_entrée.csv', 'données_EEG_sortie.csv', signaux_pca)
```

Correction filtrage PCA

```
%#####
%##### CORRECTION FONCTION COURS TRAITEMENT SIGNAL AVANCE #####
%##### PCA #####
%### Florian Lepont, Romain Richard##### Avril 2015#####

% Bibliographie : "A Tutorial on Principal Component Analysis" de Jonathon Shlens

function [signaux,composantes_principales,valeurs_propres] = pca(donnees)
% Analyse en composantes principales.
% ENTREE : données à traiter
% SORTIE : Signaux décorrélés, vecteur des composantes principales,
% vecteurs_propres

[M,N] = size(donnees);

% Translation de la moyenne des données vers 0.
mn = mean(donnees,2);
donnees = donnees - repmat(mn,1,N);

% Calcul de la matrice de covariance
covariance = 1 / (N-1) * (donnees * donnees');

% Calcul des valeurs propres et des vecteurs propres de la matrice de covariance. Les vecteurs propres en sortie correspondent aux composantes principales.
[composantes_principales,valeurs_propres] = eig(covariance);

% Extraction des valeurs propres de la matrice vers un vecteur
valeurs_propres = diag(valeurs_propres);

% Trie des valeurs propres par ordre décroissant
[junk, indices] = sort(-1*valeurs_propres);
valeurs_propres = valeurs_propres(indices);
composantes_principales = composantes_principales(:,indices);

% On réalise la transformation linéaire des données d'entrée
signaux = composantes_principales'* donnees;
```

Lecture CSV

```
%##### FONCTION LECTURE D UNE FICHIER CSV #####
%##### CSV ##### Florian Lepont, Romain Richard##### Avril 2015#####
%
function [donnees_eeg] = lecture_csv( fichier_entree )
% Fonction lecture vers un fichier csv. La fonction permet de lire
% les données EEG contenues dans le fichier csv 'fichier_entree'.
%
% EXTRACTION DES DONNEES EEG
% Extraction des données EEG à partir du fichier CSV
% Suppression des infos temps et fréquence d'échantillonnage afin de ne conserver
% que les données des signaux
donnees_eeg(:,1) = []
donnees_eeg(:,end) = []
% Calcul de la transposée
donnees_eeg = donnees_eeg'
%
end
```

Écriture CSV

```
%##### FONCTION ECRITURE D UNE FICHIER CSV #####
%##### CSV #####
%##### Florian Lepont, Romain Richard##### Avril 2015#####

function [] = ecriture_csv( fichier_entree, fichier_sortie, donnees)
% Fonction ecriture vers un fichier csv. La fonction permet d'ecrire
% vers le fichier csv 'fichier_sortie'. Pour cela, il récupère les
% informations temps, fréq_echantill, colonnes et lignes à partir du
% fichier csv 'fichier_entree', généré par OpenViBE.

% EXTRACTION DES DONNEES EEG ET CARACTERISTIQUES ACQUISITION
% Extraction des données EEG
donnees_eeg = csvread(fichier_entree,1,0);
% Extraction de la taille du tableau de données
[lignes,colonnes] = size(donnees_eeg)
% La première colonne du fichier csv correspond aux échantillons temporels
temps = donnees_eeg(:,1)
% La valeurs de la dernière colonne du fichier csv correpond à la fréquence
% d'échantillonnage
freq_echant = donnees_eeg(1,end)

% EXTRACTION DU NOM DES VALEURS A PARTIR DU FICHIER D'ENTREE
% Ouverture du fichier csv
fid = fopen(fichier_entree,'r');
% On récupère les données du fichier
C = textscan(fid, repmat('%s',1,colonnes), 'delimiter','','','CollectOutput',true);
C = C{1};
% extraction de la première ligne du fichier, comprenant le nom des variables
nom_valeurs = C(1,:);
fclose(fid);

% CREATION DU FICHIER DE SORTIE
% Ouverture du nouveau fichier
fid1 = fopen(fichier_sortie,'w');

% Concaténations des données temps et fréquence d'échantillonnage avec
% les nouvelles valeurs décoréllées
donnees_csv = horzcat(temps, donnees')
donnees_csv(:,end+1) = zeros;
donnees_csv(1,end) = freq_echant

headerFmt = repmat('%s,',1,colonnes-1);
numFmt = repmat('%f,',1,colonnes-1);
% Écriture de l'en-tête du csv (nom des variables)
fprintf(fid1,[headerFmt,'%s\n'],nom_valeurs{1,:});
% Écriture des données
fprintf(fid1,[numFmt,'%f\n'],donnees_csv');

fclose(fid1)
end
```

GEI 723 : Travail pratique

Visualisation de signaux EEG et de l'activité cérébrale

On vous propose à travers ce travail pratique de vous familiariser avec les aspects théoriques et techniques d'une interface cerveau-ordinateur (BCI). Celle que nous vous demandons d'étudier a été mise au point sous le logiciel libre de droit OpenViBE.

Interface cerveau-ordinateur

L'expérience que l'on vous propose d'étudier s'appuie sur la méthode d'imagerie motrice, qui consiste à imaginer le mouvement de différentes parties du corps. Cela résulte en l'activation du cortex sensorimoteur qui module certains rythmes du cerveau. Ces variations peuvent être détectées par des électrodes EEG. L'interface cerveau-ordinateur en déduit les intentions de l'utilisateur à partir de ces signaux. Dans notre cas, celle-ci doit être capable de déterminer si l'utilisateur pense à un mouvement de sa main droite ou de sa main gauche.

La chaîne de traitement des signaux EEG de l'interface cerveau-ordinateur se compose en 5 étapes (Figure 8.1) :

1. **L'acquisition des données.** En temps normal, on récupère en temps réel les données directement issues d'un casque EEG (par exemple le casque EPOC). Comme vous ne possédez pas ce genre de casque, on vous fournit un fichier .ov contenant des signaux EEG pré-enregistrés d'une personne pensant aléatoirement à sa main gauche ou droite.
2. **Filtrage temporel.** Ce filtrage permet d'extraire les rythmes du cerveau qui nous intéressent.
3. **Filtrage spatial CSP.** Permet de réduire la quantité d'informations à traiter en réalisant une décorrélation et en pondérant certaines des sorties du casque.
4. **Analyse spectrale.** Réalise une analyse spectrale via une transformée de Fourier.
5. **Classification.** Effectue une classification des caractéristiques à son entrée afin de déterminer à quelle main pensait l'utilisateur à un instant donné.

Le filtre spatial et le classificateur nécessitent d'être préalablement entraînés grâce à d'autres scénarios, générant alors un fichier de configuration pour ces deux blocs ("Classifier.cfg" et "csp-spatial-filter.cfg"). On vous fournit ici directement ces fichiers de configuration.

Voici la description des différents blocs utilisés dans ce scénario :

Traitement de signal

- **Temporal Filter.** Réalise le filtrage temporel d'un signal. Il est possible de choisir la méthode de filtrage (passe-bas, passe-bande ou passe-haut), le type (Butterworth, Chebychev) et l'ordre de filtre.
- **Spatial Filter.** Permet de réaliser du filtrage spatiale statique (filtrage bipolaire, Laplacien, etc.)
- **CSP Spatial Filter Trainer.** Permet de réaliser un filtrage spatial via l'algorithme CSP (voir Partie 4).
- **Time Based Epoching.** Permet de découper un signal en plusieurs morceaux ayant une durée et un intervalle spécifié.
- **Spectral Analysis (FFT).** Réalise l'analyse spectrale de signaux en utilisant la FFT (Fast Fourier Transform).
- **Spectrum Average.** Calcule la moyenne de toutes les puissances de bandes de fréquences pour un spectre donné.

Classification

- **Classifier Processor.** Permet de réaliser la classification de vecteurs caractéristiques entrants en utilisant un classificateur entraîné précédemment.

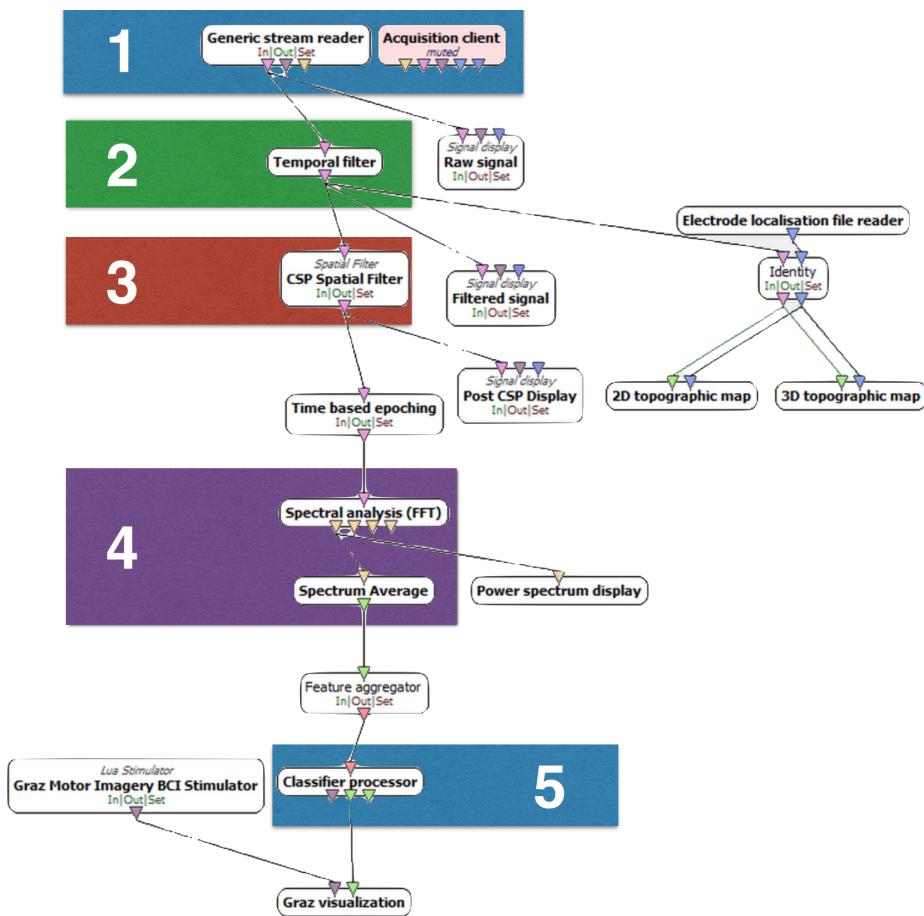


FIGURE 8.1 – Traitement des signaux EEG sous OpenViBE.

Lecture et écriture de fichiers

- **Generic Stream Reader.** Permet de lire un fichier enregistré grâce à un bloc Generic File Writer.

Visualisation

- **Signal Display.** Permet d'afficher un signal en temps réel.
- **Power Spectrum Display.** Permet d'afficher l'amplitude d'un signal dans un ensemble de bandes de fréquences.
- **2D Topographic Map.** Permet d'afficher la topographie de l'activité cérébrale en 2 dimensions.
- **3D Topographic Map.** Permet d'afficher la topographie de l'activité cérébrale en 3 dimensions.
- **Graz Visualisation.** Permet d'afficher la consigne et le résultat en temps réel de la première expérience que nous avons réalisé.

Autres

- **Feature aggregator.** Permet de rassembler les différentes entrées du bloc vers un vecteur caractéristique.
- **Identity .** Duplique l'entrée de ce bloc vers sa sortie.

OpenViBE

Présentation du logiciel

OpenViBE est une plateforme logicielle dédiée à la conception et à l'utilisation des interfaces cerveau-ordinateur en temps réel. Celle-ci est conçue par l'Institut National de Recherche en Informatique et en Automatique (INRIA) de Rennes (France). Le logiciel gère à la fois l'acquisition, le traitement et l'analyse des signaux.

Le logiciel fonctionne sous la forme de scénarios, correspondant à un ensemble de blocs fonctionnels. La suite OpenViBE est composée de deux logiciels :

- **OpenViBE designer.** Il s'agit d'un éditeur de scénario permettant de créer et modifier des schémas blocs grâce à une interface graphique. Il suffit alors de placer les différents blocs désirés, de les configurer (en cliquant dessus) et de les relier pour former une chaîne de traitement (procédure).
- **OpenViBE acquisition server.** Permet d'acquérir des données EEG brutes à partir de dispositifs EEG. Il est par exemple possible de connecter différents types de casques, dont le casque EPOC.

Téléchargement et installation d'OpenViBE

Le logiciel fonctionne sous les systèmes d'exploitations Windows (Windows XP ou ultérieur, la stabilité du logiciel sous Windows 8 n'ayant pas été vérifiée par l'éditeur) et Linux (Ubuntu, Fedora). La liste détaillée des architectures compatibles avec OpenViBE est disponible à l'adresse suivante : <http://openvibe.inria.fr/supported-architectures/>. Il est possible de télécharger l'installateur OpenViBE à l'adresse suivante : <http://openvibe.inria.fr/downloads/>. Une fois celui-ci téléchargé, il suffit de l'ouvrir et de suivre les indications affichées à l'écran.

Présentation de l'interface utilisateur d'OpenViBE Designer

Description de l'interface de contrôle de la simulation

La partie supérieure permet de contrôler la simulation, i.e. le déroulement du scénario. Ainsi, il est possible de lancer un scénario en appuyant sur le bouton "lecture", de le stopper en appuyant sur le bouton "stop" et d'en accélérer le déroulement en appuyant sur le bouton "avance rapide". Il est également possible de lancer plusieurs scénarios en même temps (Figure : 8.2).



FIGURE 8.2 – Interface de contrôle de la simulation sous OpenViBE.

Description de l'interface du répertoire des blocs fonctionnels

La partie située à droite de l'interface graphique du designer permet de rechercher des blocs fonctionnels afin de construire un scénario. Ceux-ci sont triés en différentes catégories. Un simple glisser-déposer permet de placer un bloc dans l'espace de travail (Figure : 8.3).

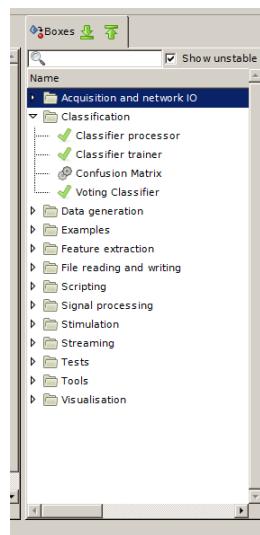


FIGURE 8.3 – Interface du répertoire des blocs OpenViBE.

Description de l'espace de travail

La partie centrale de l'interface graphique correspond à l'espace de travail d'OpenViBE. C'est ici où l'on construit les différents scénarios de l'interface cerveau-ordinateur, via des blocs fonctionnels. Chaque bloc peut être relié à un autre. On peut seulement connecter une entrée à une sortie du même type (e.g. une entrée de type "signal" vers une sortie de type "signal") (Figure : 8.4).

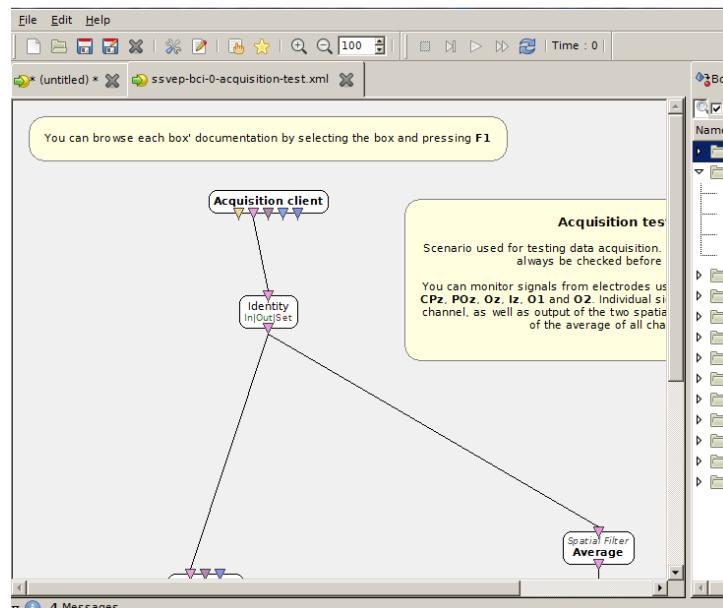


FIGURE 8.4 – Espace de travail d'OpenViBE.

Manipulation

1. Commencez par télécharger et installer OpenViBE.
2. Une fois OpenViBE installé, ouvrez le fichier "online-neuroscience.xml". Il s'agit du scénario comprenant la chaîne de traitement des signaux EEG.
3. Double-cliquez sur le bloc "Generic Stream Reader" et indiquez le chemin vers le fichier "motor-imagery-csp.ov" qui vous a été fourni.
4. Double-cliquez sur le bloc "csp-spatial-filter.cfg" et indiquez le chemin vers le fichier csp-spatial-filter.cfg" qui vous a été fourni.
5. Double-cliquez sur le bloc "Classifier Processor" et indiquez le chemin vers le fichier "classifier.cfg" qui vous a été fourni.
6. Double-cliquez sur le bloc "Graz motor imagery bci simulator" et indiquez le chemin vers le fichier "motor-imagery-bci-graz-stimulator.lua" qui vous a été fourni.
7. Démarrez l'interface cerveau-ordinateur en cliquant sur le bouton "lecture".

Questions

1. Une fois le scénario lancé, aucune activité cérébrale n'apparaît dans la fenêtre de visualisation. Le filtre temporel passe-bande a été configuré pour ne laisser passer aucun signal (fréquences de coupures 0Hz, 0Hz). Sachant que l'on souhaite dans le cadre de notre application extraire uniquement certains rythmes du cerveau, comment devez-vous configurer le filtre temporel ? Modifiez la configuration du filtre en double-cliquant dessus.
2. Quel est le nom des rythmes que l'on souhaite conserver ?
3. Observez les différentes étapes de traitement des signaux EEG.
4. L'analyse spectrale des signaux est effectuée avec une FFT. Est-ce la solution la plus optimale ? Pourquoi ?
5. Proposez une autre solution d'analyse spectrale.

GEI 752 : Travail pratique

Traitement de signaux EEG et applications dans une interface cerveau-ordinateur

On vous propose à travers ce travail pratique de vous familiariser avec les aspects théoriques et techniques d'une interface cerveau-ordinateur et d'utiliser vos connaissances en traitement du signal afin de participer à la conception de la chaîne de traitement des données électroencéphalographiques (EEG).

Interface cerveau-ordinateur

Une interface Cerveau-Ordinateur ou BCI (Brain Computer Interface) est une interface permettant de réaliser une communication allant du cerveau vers un système numérique (ordinateur par exemple). Pour cela, on s'appuie sur une étude de l'activité neuronale du cerveau grâce à un système électroencéphalographique (EEG) utilisant des électrodes. Celles-ci sont placées à la surface du crâne afin de capter son activité électrique. Un traitement et une analyse des signaux électriques saisis sont réalisés afin de les traduire en informations exploitables par les systèmes numériques. L'expérience que l'on vous propose d'étudier ici s'appuie sur la méthode d'imagerie motrice, qui consiste à imaginer le mouvement de différentes parties du corps. Cela résulte en l'activation du cortex sensorimoteur qui module certains rythmes du cerveau. Ces variations peuvent être détectées par des électrodes EEG. L'interface cerveau-ordinateur en déduit les intentions de l'utilisateur à partir de ces signaux. Dans notre cas, celle-ci doit être capable de déterminer si l'utilisateur pense à un mouvement de sa main droite ou de sa main gauche.

La chaîne de traitement des signaux EEG de l'interface cerveau-ordinateur se compose de 5 étapes (Figure 8.5) :

1. **L'acquisition des données.** En temps normal, on récupère les données directement issues d'un casque EEG (par exemple le casque EPOC de la société Emotiv). Comme vous ne possédez pas ce genre de casque, on vous fournit un fichier .csv contenant des signaux EEG pré-enregistrés d'une personne pensant aléatoirement à sa main gauche ou droite.
2. **Filtrage temporel.** Ce filtrage permet d'extraire les rythmes du cerveau qui nous intéressent.
3. **Filtrage spatial.** On souhaite utiliser un filtrage spatial afin d'optimiser nos données. Il vous sera demandé ultérieurement d'implémenter ce filtre sous Matlab.
4. **Filtrage spatial CSP.** Permet de réduire la quantité d'informations à traiter en réalisant une décorrélation et en pondérant certaines des sorties du casque.
5. **Analyse spectrale.** Réalise une analyse spectrale via une transformée de Fourier.
6. **Classification.** Effectue une classification des caractéristiques présentes à son entrée afin de déterminer à quelle main pensait l'utilisateur à un instant donné.

Le classificateur nécessite d'être préalablement entraîné grâce à d'autres scénarios, qui génèrent alors son fichier de configuration. On vous fournit directement le fichier de configuration ("classifier.cfg"). Veuillez également noter que vous ne possédez pas le scénario d'acquisition permettant de récupérer les données EEG. On vous fournit donc également les données filtrées temporellement dans le fichier ("données_entrée.csv").

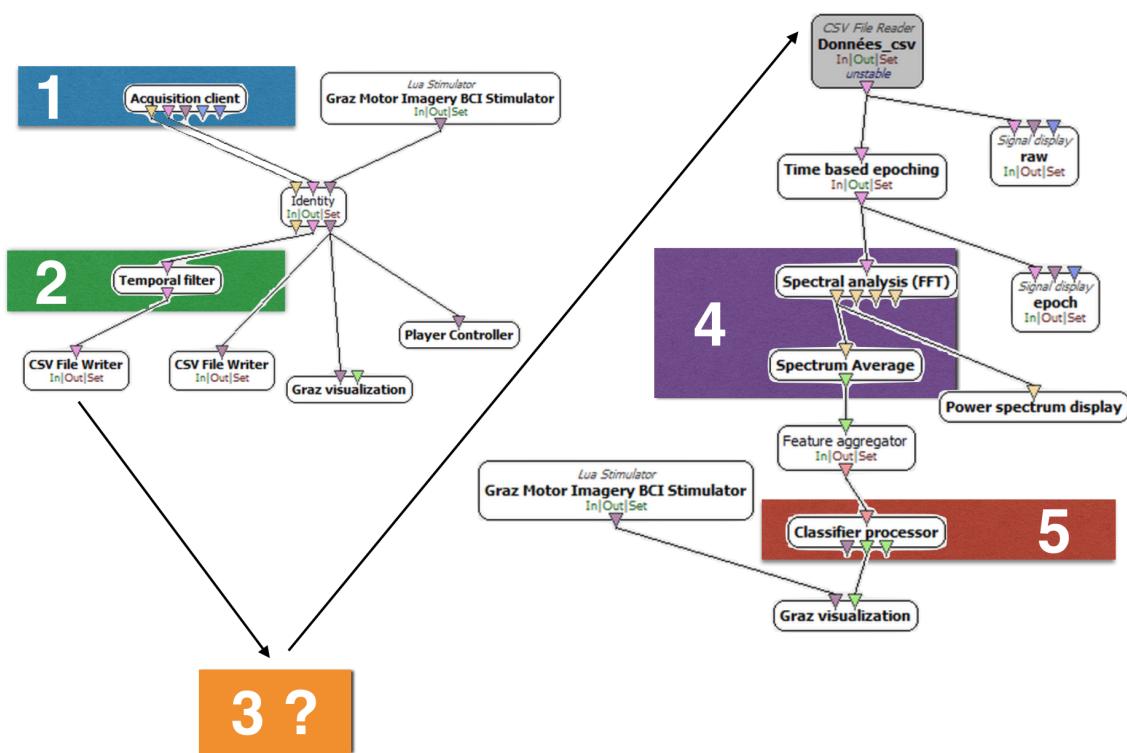


FIGURE 8.5 – Traitement des signaux EEG sous OpenViBE.

OpenViBE

Présentation du logiciel

OpenViBE est une plateforme logicielle dédiée à la conception et à l'utilisation d'interfaces cerveau-ordinateur en temps réel. Celle-ci est éditée par l'Institut National de Recherche en Informatique et en Automatique (INRIA) de Rennes (France). Le logiciel gère à la fois l'acquisition, le traitement et l'analyse des signaux.

Le logiciel fonctionne sous la forme de scénarios, correspondant à un ensemble de blocs fonctionnels. La suite OpenViBE est composée de deux logiciels :

- **OpenViBE designer.** Il s'agit d'un éditeur de scénario permettant de créer et modifier les schémas blocs grâce à une interface graphique. Il suffit alors de placer les différents blocs, de les configurer (en cliquant dessus), et de les relier pour former une chaîne de traitement.
- **OpenViBE acquisition server.** Permet d'acquérir des données EEG brutes à partir de dispositifs EEG. Il est par exemple possible de connecter différents types de casques, dont le casque EPOC.

Téléchargement et installation d'OpenViBE

Le logiciel fonctionne sous les systèmes d'exploitation Windows (Windows XP ou ultérieur, la stabilité du logiciel sous Windows 8 n'ayant pas été vérifiée par l'éditeur) et Linux (Ubuntu, Fedora). La liste détaillée des architectures compatibles avec OpenViBE est disponible à l'adresse suivante : <http://openvibe.inria.fr/supported-architectures/>. Il est possible de télécharger l'installateur OpenViBE à l'adresse suivante : <http://openvibe.inria.fr/downloads/>. Une fois celui-ci téléchargé, il suffit de l'ouvrir et de suivre les indications affichées à l'écran.

Présentation de l'interface utilisateur d'OpenViBE Designer

Description de l'interface de contrôle de la simulation

La partie supérieure de l'interface graphique permet de contrôler la simulation, i.e. le déroulement du scénario. Ainsi, il est possible de lancer un scénario en appuyant sur le bouton "lecture", de le stopper en appuyant sur le bouton "stop" et d'en accélérer le déroulement en appuyant sur le bouton "avance rapide". Il est également possible de lancer plusieurs scénarios en même temps (Figure : 8.6).

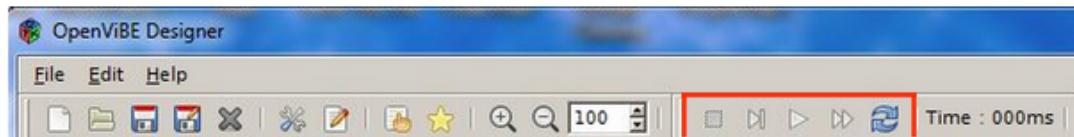


FIGURE 8.6 – Interface de contrôle de la simulation sous OpenViBE.

Description de l'interface du répertoire des blocs fonctionnels

La partie située à droite de l'interface graphique du designer permet de rechercher les blocs afin de construire un scénario. Ceux-ci sont triés en différentes catégories. un simple glisser-déposer permet de placer un bloc dans l'espace de travail (Figure : 8.7).

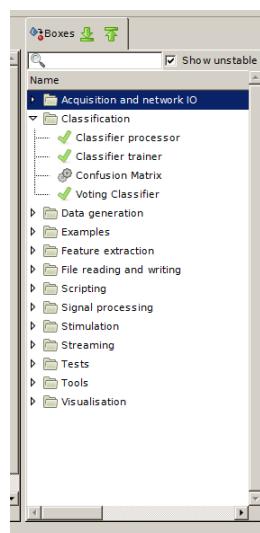


FIGURE 8.7 – Interface du répertoire des blocs OpenViBE.

Description de l'espace de travail

La partie centrale de l'interface graphique correspond à l'espace de travail d'OpenViBE. C'est ici que l'on construit les différents scénarios de l'interface cerveau-ordinateur, via des blocs fonctionnels. Chaque bloc peut être relié à un autre. On peut seulement connecter une entrée à une sortie du même type (e.g. une entrée de type "signal" vers une sortie de type "signal") (Figure : 8.8).

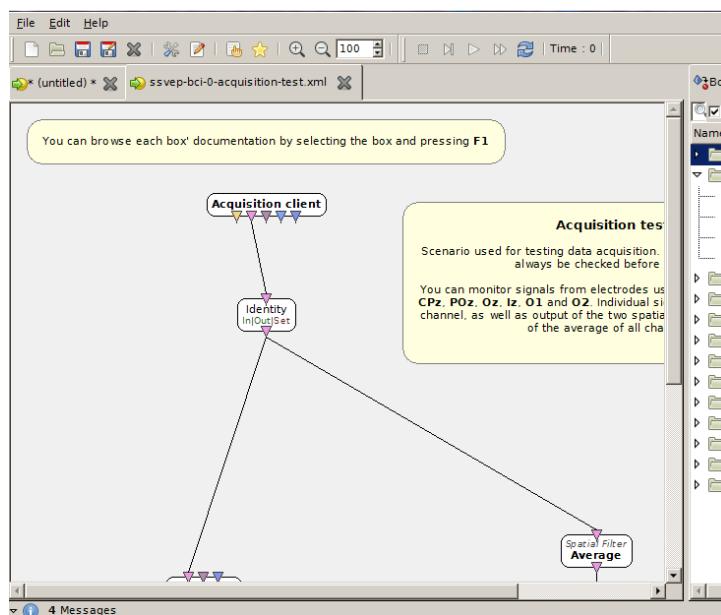


FIGURE 8.8 – Espace de travail d'OpenViBE.

Manipulation

1. Commencez dans un premier temps par télécharger et installer OpenViBE. Celui-ci vous servira dans la question 3.
2. On vous demande de coder un filtre spatial PCA sous Matlab et de traiter les données EEG contenues dans le fichier "données_entrée.csv". Le code Matlab n'étant pas directement exécutable dans OpenViBE, vous devez utiliser la fonction Matlab "lecture_csv.m" afin d'extraire les données EEG du fichier "données_entrée.csv", ainsi que la fonction "écriture_csv.m" afin d'écrire les données après filtrage spatial vers le fichier CSV "données_EEG_sortie.csv".
3. Une fois le code implémenté et vos données filtrées, on exporte les signaux vers OpenViBE. Pour cela, démarrez OpenViBE. Ouvrez ensuite le fichier "online.xml".
4. Double-cliquez sur le bloc "CSV File Reader" et indiquez le chemin vers le fichier que vous avez généré à partir de votre code Matlab. Par défaut, ce fichier s'appelle "données_EEG_sortie.csv".
5. Double-cliquez sur le bloc "Classifier Processor" et indiquez le chemin vers le fichier "classifier.cfg" qui vous a été fourni.
6. Double-cliquez sur le bloc "Graz motor imagery bci simulator" et indiquez le chemin vers le fichier "motor-imagery-bci-graz-stimulator.lua" qui vous a été fourni.
7. Démarrez l'interface cerveau-ordinateur en cliquant sur le bouton "lecture". Qu'observez-vous dans la fenêtre de visualisation ?
8. De la même manière que dans la question 2, on vous demande à présent d'implémenter un filtre spatial ICA sous Matlab. Exportez de nouveau vos données vers OpenViBE. Qu'observez-vous ?