Interface Cerveau Machine avec adaptation automatique à l'utilisateur

Xavier Artusi

ÉCOLE CENTRALE NANTES

ÉCOLE DOCTORALE

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET MATHEMATIQUES

Année : 2012

Thèse de Doctorat de l'École Centrale de Nantes

Spécialité : INFORMATIQUE, AUTOMATIQUE, ELECTRONIQUE ET GENIE ELECTRIQUE

Présentée et soutenue publiquement par :

Xavier Artusi

le 15 février 2012

à l'École Centrale de Nantes

TITRE

Interface Cerveau Machine avec adaptation automatique à l'utilisateur

Jury

Rapporteurs :	Alain Rakotomamonjy	Professeur - Université de Rouen
	Guy Carrault	Professeur - Université de Rennes I
Examinateurs :	Christian Jutten	Professeur - Université Joseph Fourier de Grenoble
	Dario Farina	Professeur - Göttingen University, Allemagne
	Marie-Françoise Lucas	Maître de Conférences - Ecole Centrale de Nantes
	Jérôme Idier	Directeur de recherche - CNRS

Directeur de thèse : **Jérôme IDIER**
Laboratoire / composante : IRCCyN / CNRS
Co-encadrant : **Marie-Françoise LUCAS**
Laboratoire / composante : IRCCyN / École Centrale de Nantes

N° ED : 503-156

Table des matières

Introduction générale

Certains patients victimes d'un accident vasculaire cérébral grave restent dans un état de paralysie musculaire complète (LIS : Locked-In Syndrom). Dans la plupart des cas, ils gardent toutes leurs facultés mentales mais ne sont plus capables de communiquer avec leur entourage par les canaux de communication standards que sont les nerfs périphériques et les muscles. Les interfaces cerveau-machine sont l'approche qui semble actuellement la plus prometteuse pour pallier ce handicap. Ce domaine de recherche a émergé récemment et est actuellement en plein développement.

Une interface cerveau-machine (BCI, Brain Computer Interface) est un système qui permet la communication directe entre le cerveau d'un individu et un ordinateur ou un robot, sans solliciter les nerfs périphériques et les muscles. Le but est de permettre à des personnes souffrant d'un handicap neuromusculaire sévère de communiquer avec leur entourage ou de commander une prothèse par la pensée. Nous nous intéressons ici à des systèmes permettant de commander une prothèse.

L'entrée d'un tel système est constituée de signaux électroencéphalographiques (EEG) enregistrés à la surface du scalp à l'aplomb des aires sensorimotrices du cortex et reliés à l'activité volontaire du patient. La sortie est une décision se traduisant par une action de commande d'une prothèse pour effectuer un mouvement particulier. Le cœur du BCI est un algorithme de classification caractérisé par le choix des descripteurs des signaux à traiter et des règles d'affectation des signaux. Une période d'apprentissage est nécessaire pour construire les fonctions de décision du système ; au cours de cette période, le sujet ayant connaissance des réponses du système s'y adapte en affinant sa stratégie de commande (biofeedback).

Les applications pratiques de la recherche dans ce domaine sont actuellement très limitées. En effet, la commande d'une prothèse suppose de pouvoir distinguer non seulement des mouvements différents, mais également différentes modalités d'un même mouvement. En raison d'un très faible rapport signal sur bruit des signaux EEG, les fonctions de décision doivent être apprises sur des essais nombreux, très fatigants pour le patient, et les résultats de la classification sont peu précis. D'autre part, la plupart des systèmes ne s'adaptent pas aux évolutions éventuelles de l'utilisateur et nécessitent de fréquents entraînements.

L'objet de cette thèse est de développer un système BCI plus précis, capable d'améliorer ses performances en cours d'utilisation et de s'adapter à l'utilisateur sans nécessiter de multiples sessions d'apprentissage. Nous combinons deux moyens pour y parvenir. Le premier consiste à augmenter la précision du système de décision en recherchant des descripteurs pertinents vis à vis de l'objectif de classification. Le second est d'inclure un retour de l'utilisateur sur le système de décision : l'idée est d'estimer l'erreur du BCI à partir de potentiels cérébraux évoqués, reflétant l'état émotionnel du patient corrélé au succès ou à l'échec de la décision prise par le BCI, et de corriger le système de décision du BCI en conséquence. Le système complet comporte ainsi 2 modules d'analyse et décision de signaux EEG. La fonction du module principal est de produire et afficher une décision de mouvement à partir des signaux EEG image de l'intention

de mouvements. Celle du module secondaire est d'estimer la présence ou l'absence d'une erreur sur la décision précédente en analysant les signaux image d'émotion. L'erreur estimée est utilisée pour adapter les règles de décision du module principal. On a ainsi une double adaptation : celle de l'utilisateur au système de décision, et celle du système à l'utilisateur.

La réalisation d'un tel dispositif nécessite un système de décision le plus fiable possible et pouvant apprendre en ligne, un moyen permettant de détecter ses erreurs, et une stratégie de mise à jour de la population d'apprentissage.

Pour développer ces différents aspects, le document est organisé en 5 chapitres. Dans le chapitre 1 nous rappelons les types de BCI existants, les modalités d'enregistrement de l'activité cérébrale et les différents types de signaux associés à la commande des BCI. Ensuite, nous présentons l'ensemble du système étudié et le protocole expérimental utilisé pour enregistrer les signaux analysés.

L'objet du chapitre 2 est de présenter des méthodes d'optimisation d'espaces de représentation à base d'ondelettes dans l'objectif d'améliorer la précision du système de décision. Nous rappelons les méthodes proposées dans des travaux antérieurs, basées sur l'optimisation de l'ondelette mère de la transformée en ondelettes ou de la base de décomposition de la transformée en paquets d'ondelettes, et nous proposons une extension de ces méthodes au cas multivoies. Nous présentons les résultats obtenus sur des signaux simulés et des signaux EEG expérimentaux correspondant à deux modalités d'un même mouvement.

Le chapitre 3 est consacré aux outils de décision utilisés, qui doivent fonctionner en ligne pour permettre la mise à jour des fonctions de décision au cours du temps. Notre choix s'est porté sur les machines à vecteurs support (SVM), dont nous rappelons tout d'abord la théorie générale, puis nous détaillons deux algorithmes permettant de les implémenter en ligne.

Le module de détection d'erreurs a pour premier objectif de corriger les décisions du module principal. Dans le chapitre 4, nous proposons de comparer deux approches de la détection des potentiels d'erreur, en quantifiant les performances théoriques du système corrigé. Nous présentons les résultats obtenus sur des données expérimentales.

Le second objectif du module de détection d'erreurs est de permettre d'apprendre en cours d'utilisation, en fournissant des essais labellisés pour la mise à jour de l'ensemble d'apprentissage. Le chapitre 5 présente le simulateur développé pour mettre au point le système ainsi bouclé, et en particulier pour étudier les performances de différentes stratégies possibles pour la mise à jour de la population d'apprentissage.

En conclusion, nous résumons les différents apports de ce travail, et nous indiquons les questions ouvertes et les perspectives pour de futures études.

Ce travail a été réalisé en étroite collaboration avec le "center for Sensory-Motor Interaction" (SMI) de l'Université d'Aalborg et le "Bernstein Freiburg Neurotechnology" (BFNT) de l'Université de Göttingen.

Chapitre 1

Contexte et présentation du système proposé

Sommaire

Comme nous l'avons précisé en introduction, une interface cerveau-machine est un système qui permet la communication directe entre le cerveau d'un individu et un ordinateur ou un robot, sans solliciter les nerfs périphériques et les muscles, les messages étant directement contenus dans l'activité cérébrale. Il existe une grande diversité de systèmes selon les modalités de production et d'acquisition de l'activité cérébrale. L'objet de ce chapitre est de situer dans ce contexte le système auquel on s'intéresse. Nous commençons par présenter les différents types de BCI existants, les méthodes d'acquisition de l'activité cérébrale et les catégories des signaux utilisés pour contrôler les BCI. Puis nous expliquons le fonctionnement d'ensemble du système proposé ainsi que le protocole d'acquisition utilisé pour récupérer les signaux analysés au cours de la thèse.

1.1 Différentes catégories de BCI

Les différents types de BCI dépendent des capacités de l'utilisateur (valide ou non), de la technologie utilisée pour acquérir les signaux EEG et de la façon de l'activer (volontaire ou non). Ainsi les BCI sont soit :

– dépendants ou indépendants,

– invasifs ou non invasifs,
– synchrones ou asynchrones.

Nous allons détailler les différences entre ces types de BCI.

1.1.1 BCI dépendant et indépendant

Même si les BCI n'utilisent que la mesure de l'activité cérébrale, certains nécessitent que l'utilisateur ait un certain contrôle de ses muscles, par exemple la capacité à contrôler la direction du regard (Section 1.3.1). Ceux-ci sont les BCI dépendants. La plupart des BCI utilisant des Potentiels Evoqués Visuels (PEV) sont dépendants [Gao *et al.*, 2003].

Certains handicaps ne permettent pas le moindre contrôle musculaire. Dans ce cas, un BCI indépendant est obligatoire. Ces derniers utilisent les potentiels corticaux lents, les potentiels P300 évoqués ou le contrôle des rythmes β et μ [Pfurtscheller *et al.*, 2000].

Les BCI dépendants sont plus robustes et plus faciles à utiliser. Ils sont destinés à des personnes valides pour des applications plutôt ludiques telles que les jeux vidéo. Les BCI indépendants, pour le moment sujets à un taux d'erreur par bit (probabilité d'erreur de l'interface pour une action) plus important, sont destinés à des applications dans le domaine médical pour des personnes souffrant de grands handicaps moteurs.

1.1.2 BCI invasif et non invasif

La distinction entre ces deux types de BCI se fait par la manière dont l'activité cérébrale est enregistrée.

Les capteurs utilisés pour les BCI invasifs sont placés sous le crâne, alors que pour les méthodes non invasives ils sont placés à sa surface.

Les méthodes invasives sont réputées avoir des signaux plus faciles à traiter et donc de meilleures performances. Cependant des opérations lourdes à intervalles réguliers sont nécessaires.

Les signaux obtenus avec des méthodes non invasives sont beaucoup plus bruités et donc plus durs à utiliser (le signal enregistré passe à travers la boîte crânienne avant d'atteindre les capteurs), mais le placement des capteurs est beaucoup moins contraignant.

On pense souvent que les méthodes invasives sont plus performantes, cependant ceci reste à confirmer et est encore débattu au sein de la communauté. Même si les signaux des méthodes non invasives sont plus bruités, certaines études montrent que les deux méthodes peuvent atteindre des taux de transfert d'information (nombre de bits par minutes) similaires. Wolpaw et McFarland [2004] montrent qu'avec un algorithme adapté, un BCI non invasif peut atteindre des performances similaires à celles des BCI invasifs en terme de nombre de commandes à reconnaître.

Les techniques non invasives sont actuellement les plus utilisées.

1.1.3 BCI synchrone et asynchrone

Pour les méthodes synchrones, l'utilisateur doit effectuer la tâche mentale sur des périodes de temps imposées, au contraire des BCI asynchrones où l'utilisateur peut effectuer la tâche mentale à n'importe quel moment.

A terme le but serait que toutes les BCI soient asynchrones. Cependant elles sont dans ce cas beaucoup plus difficiles à mettre en œuvre car l'activité cérébrale doit être analysée en continu

pour déterminer quand l'utilisateur souhaite interagir avec le système. Et une fois cet instant déterminé, l'interface doit être capable de déterminer l'état mental de l'utilisateur.

La complexité de la mise en œuvre de systèmes asynchrones fait que la majorité des BCI sont synchrones. Toutes les classes de BCI présentées précédemment fonctionnent sur le même principe. Le système dans son ensemble, représenté figure 1.1 :

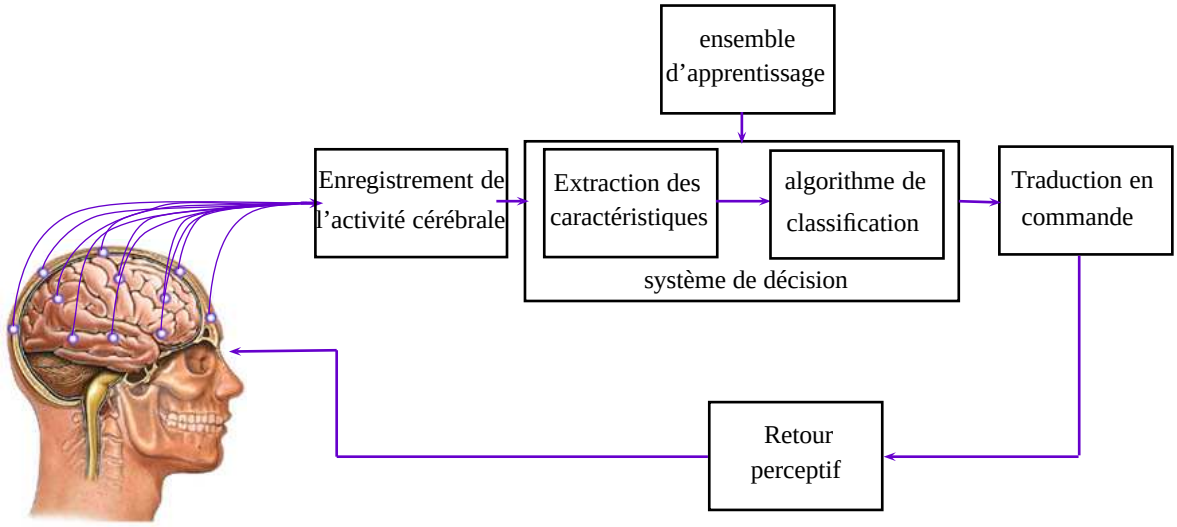


FIGURE 1.1 – Schéma général des interfaces cerveau-machine

- mesure l'activité cérébrale,
- extrait des caractéristiques du signal,
- utilise ces caractéristiques pour déterminer l'intention de l'utilisateur,
- transforme la décision du classifieur en commande,
- renvoie à l'utilisateur un retour de la décision du système.

1.2 Mesure de l'activité cérébrale

Il existe de nombreuses techniques pour mesurer l'activité cérébrale :
- Magnétoencephalographie
- Imagerie à résonance magnétique fonctionnelle
- Electrocorticographie : électrodes posées à la surface du cortex
- Electrodes implantées dans le cerveau
- Electroencéphalographie (EEG) : électrodes posées à la surface du scalp

C'est cette dernière technique qui est le plus souvent utilisée et qui nous intéressera plus particulièrement. Elle est peu coûteuse, non invasive, portable et fournit une bonne résolution temporelle.

Les signaux EEG sont enregistrés à l'aide d'électrodes (de 1 à 256) placées à la surface du scalp et maintenues généralement grâce à un casque. Afin de minimiser l'impédance entre les électrodes et le crâne, un gel ou une pâte conducteurs sont utilisés, ce qui peut être long et fastidieux à installer.

Certains systèmes n'utilisent pas de gel, cependant leurs performances sont moindres.

Le placement des électrodes se fait suivant le modèle standard "10-20 international system" représenté figure 1.2 (conçu pour 19 électrodes mais pouvant être étendu à plus).

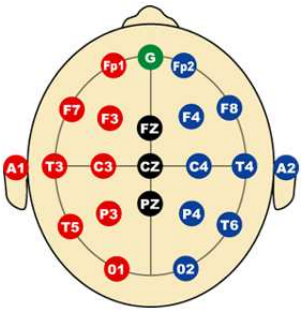


FIGURE 1.2 – Positionnement des électrodes du système 10-20.

Chaque électrode est référencée par une lettre et un chiffre ou la lettre "z". La première lettre indique le lobe du cerveau dont l'activité est enregistrée et le chiffre l'hémisphère considéré (droit : chiffres pairs, gauche : chiffre impair et ligne médiane : "z"). Sur la figure 1.3 sont représentées les principales parties du cerveau. L'activité électrique du lobe :

- frontal est associée aux électrodes commençant par "F" ou "C". Il n'existe pas de lobe central et la lettre "C" est juste introduite dans un but de différenciation.
- pariétal est associée aux électrodes commençant par "P",
- occipital est associée aux électrodes commençant par "O",
- temporal est associée aux électrodes commençant par "T".

Les lettres "A" et "G" correspondent respectivement aux électrodes de référence et à la masse.

Il est possible d'utiliser plus de 19 électrodes pour enregistrer les signaux EEG. Dans ce cas certaines électrodes seront référencées par deux lettres au lieu d'une seule. Par exemple, on peut avoir l'électrode FC_z. Cette électrode sera placée entre les électrode F_z et C_z.

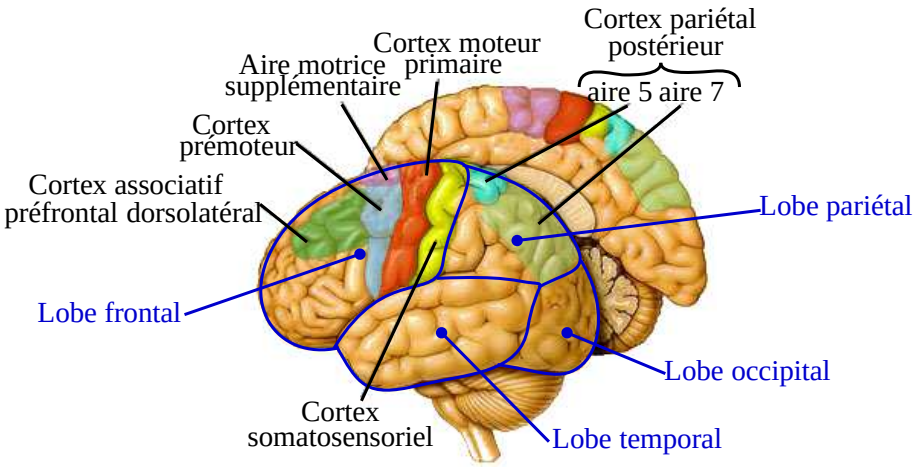


FIGURE 1.3 – Schéma des différentes zones du cerveau.

L'EEG mesure la somme des potentiels post-synaptiques générés par des milliers de neurones qui ont la même orientation radiale par rapport au scalp. Selon une terminologie médicale, les ondes cérébrales sont composées de différents rythmes :

- Delta : fréquence inférieure à 4Hz, c'est le rythme dont l'amplitude est la plus élevée. Il s'observe chez les adultes en état de sommeil profond, et est localisé principalement dans le cortex frontal.
- Thêta : fréquence entre 4 et 7Hz. Il s'observe surtout chez les jeunes enfants ou chez l'adulte lors de somnolence ou de phase d'éveil. Il est aussi associé à des états de méditation ou

de relaxation.
- Alpha : fréquence comprise entre 8 et 13Hz et d'amplitude comprise entre 30 et 50μV. Ce rythme s'observe dans le cerveau d'une personne au repos, yeux clos, et prédomine dans les cortex pariétaux, occipitaux et temporaux. Il s'interrompt dès lors qu'une stimulation est appliquée (blocage alpha aussi appelée réaction d'arrêt) ou que le sujet démarre une activité intellectuelle. Ce rythme, présent dès les premières années de l'enfant, se développe avec l'âge.
- Mu : fréquence comprise entre 8 et 13Hz (principalement 9-11Hz). C'est un cas particulier de rythme alpha. Observable en situation de veille à partir du cortex frontal et pariétal. D'amplitude inférieure à 50 μV, il émerge principalement de l'activité du cortex moteur mais également du cortex sensoriel. Il est bloqué ou largement diminué par l'initiation d'un mouvement, ou seulement sa préparation mentale, ainsi que par une stimulation tactile.
- Bêta : fréquence supérieure entre 13 et 30Hz et d'amplitude faible caractérisant le tracé encéphalographique de l'éveil. Le rythme bêta est un rythme rapide observable principalement durant l'éveil lorsque le sujet est alerte et traite de l'information, et s'atténue pendant l'endormissement. Ce rythme trouve son origine dans les régions antérieures (frontales) et pariétales du cortex, ce qui explique que certains auteurs l'associent à l'émergence de la conscience.
- Gamma : fréquence élevée supérieure à 30Hz et d'amplitude légèrement supérieure à celle du rythme bêta, caractérisant l'activité consciente, les processus cognitifs en phase d'éveil, et présent également lors de la phase de sommeil paradoxal. Cependant l'activité motrice entraîne une décohérence significative de ce rythme.

1.3 Les types de signaux utilisés pour contrôler les BCI

Les BCI actuels peuvent être divisés en 2 groupes principaux selon les types de signaux utilisés.
- Les signaux évoqués : ils sont générés de manière inconsciente par l'utilisateur en réponse à un stimulus extérieur. Ils sont appelés potentiels évoqués (EP)
- Les signaux spontanés : ils sont volontairement générés par l'utilisateur lorsqu'il réalise différentes taches mentales.

1.3.1 Les potentiels évoqués

Ils se divisent en deux catégories principales : les Steady State Evoked Potentials (SSEP) et les Event Related Potentials (ERP). L'avantage des EP est qu'ils permettent de créer des BCI qui ne nécessitent que peu ou pas d'entrainement pour l'utilisateur. Leur désavantage est que l'utilisateur doit attendre que le stimulus utile apparaisse. De plus l'utilisateur doit rester concentré constamment sur des tâches rapides et répétitives, ce qui peut être fatiguant.

Steady State Evoked Potentials Les SSEP sont émis en réponse à un stimulus visuel ou auditif périodique. Le signal à une fréquence fixe provoque une augmentation de l'activité cérébrale à cette fréquence ainsi qu'à ses harmoniques et/ou sous-harmoniques (cf Figure 1.4). Un SSVEP (Steady State Visual Evoked Potentials) peut être détecté en examinant le spectre du signal issu des électrodes O_1 et O_2 du système international 10-20 (cf Figure 1.2).

Pour une utilisation dans les BCI, on associe une action particulière à des cibles clignotant à des fréquences différentes. L'utilisateur peut alors contrôler le BCI en regardant la cible correspondant à l'action désirée.

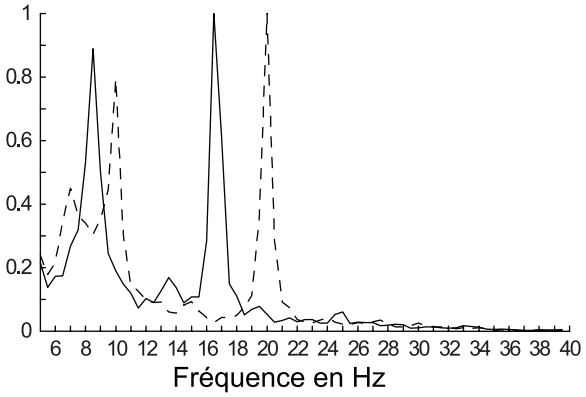


FIGURE 1.4 – Densité spectrale de puissance de signaux EEG pour une stimulation visuelle à 17Hz (trait continu) et 20Hz (trait pointillé). On voit un pic à la fréquence de stimulation.

Event Related Potentials Les ERP sont des potentiels qui apparaissent dans l'activité cérébrale en réponse à un stimulus rare et pertinent. Parmi les ERP, le plus connu et utilisé pour les BCI est le P300. C'est une ondulation positive dont le pic se situe à environ 300ms après que le stimulus soit apparu. Ce potentiel est principalement localisé dans la région pariétale.

Les BCI basés sur le P300 fonctionnent de la même façon que ceux basés sur les SSVEP : Plusieurs boutons ou objets sont affichés sur un écran. Ces boutons sont surlignés aléatoirement et l'utilisateur doit compter le nombre de fois, sur une certaine durée, que le bouton qu'il veut est allumé. Une des applications les plus connues de l'utilisation de ce potentiel est le "P300-speller" [Farwell et Donchin, 1988]. Dans une matrice de lettres (Figure 1.5), les colonnes et les lignes s'éclairent séquentiellement. L'utilisateur compte le nombre de fois que la lettre désirée s'éclaire. Une fois la séquence terminée, le BCI analyse les signaux enregistrés pour en déduire la lettre choisie.

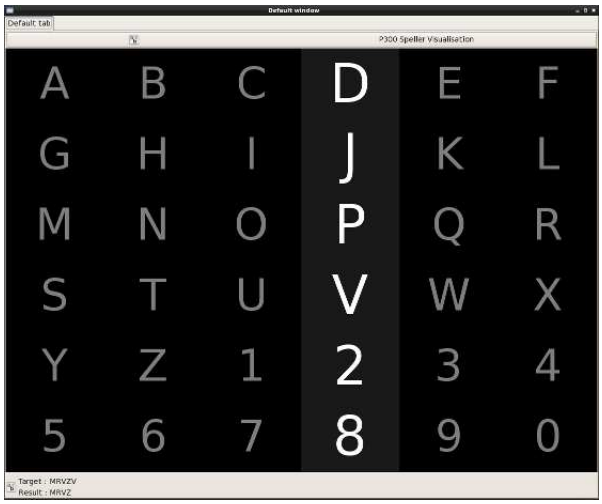


FIGURE 1.5 – Matrice de lettres du "P300 speller".

1.3.2 Les signaux spontanés

Les potentiels évoqués ne permettent qu'un fonctionnement synchrone des BCI : ce n'est pas l'utilisateur qui décide quand il active le BCI.

Si on veut que l'utilisateur décide, il faut développer des protocoles asynchrones commandés par des changements d'état mental volontaires du sujet, utilisant des signaux dits "spontanés".

Parmi les activités spontanées, on distingue plusieurs catégories correspondant à différents types d'activité cérébrale.

Taches cognitives non-motrices Un nombre important de tâches cognitives non-motrices sont utilisées pour contrôler un BCI. Ces tâches peuvent être du calcul mental, des rotations mentales... Toutes ces tâches provoquent des variations de l'EEG dans des régions du cortex et à des fréquences spécifiques, ce qui les rend assez faciles à reconnaître.

Rythmes sensorimoteurs Ce sont les rythmes du cerveau liés à des actions motrices, telles que bouger une jambe. Ces rythmes, présents principalement dans les bandes de fréquence μ (8-13Hz) et β (13-30Hz) et localisés au-dessus du cortex moteur, peuvent être contrôlés par l'utilisateur. Deux techniques sont utilisées afin que l'utilisateur puisse contrôler ces rythmes.

- **Conditionnement de l'opérateur :** Un sujet peut apprendre à modifier volontairement l'amplitude de son rythme sensorimoteur grâce à un long entrainement.
  Pour ce faire l'utilisateur choisit librement une tâche mentale avec laquelle il est à l'aise. L'inconvénient de cette méthode est que l'entrainement peut durer plusieurs semaines voir plusieurs mois. Cependant les performances obtenues sont très bonnes [Wolpaw et McFarland, 2004].
- **Imagination motrice :** L'utilisateur doit imaginer un mouvement d'un de ses membres, comme lever la pointe de son pied. Une telle tâche possède des caractéristiques temporelles, spatiales et fréquentielles très particulières ce qui permet de la détecter relativement facilement [Townsend *et al.*, 2004].
  Les BCI basés sur ce type de signaux ne nécessitent pas énormément d'entrainement, certaines interfaces n'en demandent pas du tout [Blankertz *et al.*, 2007].

Potentiels corticaux lents (PCL) Ces signaux sont des variations très lentes de l'activité corticale, pouvant prendre de quelques centaines de millisecondes à plusieurs secondes. Il est possible d'apprendre à produire ces variations positives ou négatives en utilisant le conditionnement de l'opérateur.

Les PCL sont utilisés pour des BCI à commande binaire suivant le signe du potentiel. L'entrainement est très long, même plus long que pour les rythmes sensorimoteurs. Cependant il semble que les PCL soient plus stables.

1.4 Présentation du système proposé

Dans le cadre de ma thèse, le BCI développé est de type indépendant, non-invasif et synchrone.

Indépendant : Il est destiné à des utilisateurs pouvant être dans l'incapacité d'utiliser la totalité de leurs muscles, même les mouvements oculaires.

Non-invasifs : Les électrodes sont posées sur le scalp.

Synchrone : Au stade expérimental nous développons une interface synchrone car plus robuste et plus facile à mettre en place, l'objectif à long terme étant de développer un système asynchrone.

Nous rappelons que l'objet de ce travail : développer un système précis capable d'améliorer ses performances en cours d'utilisation et de s'adapter à l'utilisateur sans nécessiter de multiples sessions d'apprentissage.

1.4.1 Travaux antérieurs

Pour adapter le système de décision à l'utilisateur, il est possible d'utiliser directement les sorties du système sans aucune autre information, ceci correspond à une adaptation non supervisée. Cette approche a été développée dans [Yuanqing et Cuntai, 2006] pour réduire les sessions d'entraînement et dans [Vidaurre *et al.*, 2011] pour donner à l'utilisateur un retour pendant l'entraînement et ainsi éviter un changement entre les conditions d'entraînement et celles d'utilisation du BCI en ligne. L'approche non supervisée semble se révéler efficace lorsque le taux d'erreur du BCI, avec l'apprentissage initial, est suffisamment faible. Cependant pour un système ayant une précision viable (entre 65% et 85% selon les sujets), cette technique est inefficace et déstabilise le système car l'étiquette des nouveaux individus ajoutés à l'ensemble d'apprentissage n'est pas assez fiable.

Une adaptation supervisée implique de pouvoir identifier l'ensemble des essais correctement étiquetés par le système de décision. Plusieurs auteurs ont travaillé sur la possibilité d'obtenir une information sur la validité des décisions du système en utilisant les signaux EEG. Falkenstein *et al.* [2000] ont montré que, lorsque le sujet réalise qu'il a fait une erreur, des changements spécifiques apparaissent dans son EEG. Schalk *et al.* [2000] ont présenté des résultats lors du contrôle du mouvement d'un curseur et ont prouvé la présence d'un potentiel d'erreur lorsque le sujet commet une erreur. Blankertz *et al.* [2002] ont proposé une méthode permettant de détecter ces potentiels d'erreur à chaque essai. Cependant dans tous ces cas, l'erreur est effectuée par le sujet lui-même et non par le système de décision.

Ce n'est que plus récemment que certains auteurs ont identifié et détecté des potentiels d'erreur (ErrP) générés en réponse à une erreur faite par le système de décision et ont montré qu'il était possible d'améliorer les performances du BCI dans le cas d'interface utilisant des signaux spontanés [Buttfield *et al.*, 2006 ; Chavarriaga et Millán, 2010 ; Ferrez et Millán, Graz 2008] ou évoqués [Dal Seno *et al.*, 2010].

1.4.2 Système proposé

La thèse s'inscrit dans ce contexte, le système étudié comporte ainsi 2 modules d'analyse et décision de signaux EEG. Le module principal produit et affiche une décision de mouvement à partir des signaux EEG images de l'intention de mouvement. Le module secondaire estime la présence ou l'absence d'une erreur sur la décision précédente en analysant les signaux évoqués. Sa sortie est utilisée pour adapter les règles de décision du module principal, et enrichir la base d'apprentissage.

Sur la figure 1.6 est représenté le fonctionnement de l'interface adaptative. Un tel schéma nécessite un système de décision pouvant apprendre en ligne, un dispositif permettant de détecter ses erreurs et une stratégie de mise à jour de la population d'apprentissage :

1. **Un système de décision pouvant apprendre en ligne.** Le cœur du BCI est un algorithme de décision dans lequel les signaux sont représentés dans l'espace des descripteurs et classés par des fonctions de décision apprises sur un ensemble d'apprentissage. Les performances de cette étape dépendent de l'espace de représentation et des fonctions discriminantes.

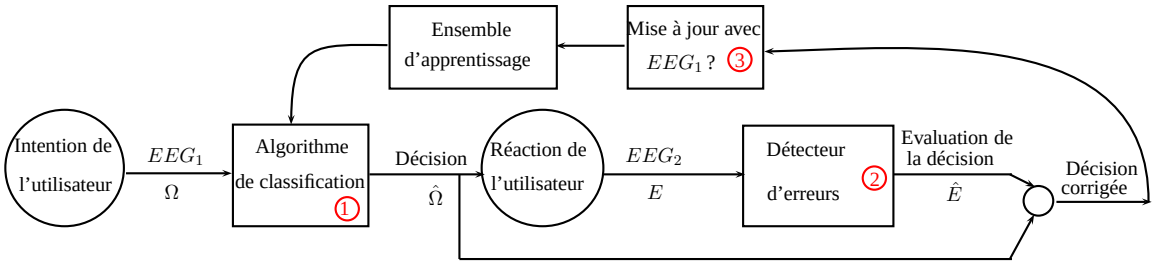


FIGURE 1.6 – Représentation schématique du BCI adaptatif proposé

. Espace de représentation. Des travaux précédents de l'équipe ADTSI [Vautrin *et al.*, 2009 ; Farina *et al.*, 2007] ont montré l'intérêt de l'utilisation des marginales de transformées en ondelettes comme descripteurs des signaux : ce type de descripteurs permet, en optimisant l'ondelette mère ou la base d'ondelettes, d'adapter l'espace de représentation aux signaux. Nous reprendrons et développerons cette caractérisation.

. Apprentissage des règles de décision en ligne. Dans cet espace de représentation, des fonctions discriminantes doivent être calculées sur une population d'apprentissage afin de reconnaître les différentes intentions de mouvement de l'utilisateur. Il existe différentes méthodes de classification correspondant à différents types de fonctions : classifieurs bayésiens, réseaux de neurones, machines à noyau... Dans notre système la méthode utilisée doit donner des résultats satisfaisants même avec de petites populations d'apprentissage (vis-à-vis de la dimension de l'espace de représentation) et doit pouvoir fonctionner en ligne en intégrant et oubliant des individus afin d'adapter la population d'apprentissage aux évolutions éventuelles du sujet et d'enrichir l'ensemble d'apprentissage avec de nouveaux exemples. C'est pourquoi nous avons retenu les SVM : l'approche est réputée robuste dans ce contexte et peut être utilisée en ligne de façon décrémentale et incrémentale. Nous avons utilisé les méthodes décrites dans [Cauwenberghs et Poggio, 2001] et [Vishwanathan *et al.*, 2003].

2. **Un dispositif permettant de détecter ses erreurs.** Afin de corriger le système de décision il est impératif de pouvoir détecter les erreurs commises par le système. Ceci est fait dans un premier temps en analysant les potentiels cérébraux évoqués après avoir donné la réponse du BCI à l'utilisateur. Nous avons étudié et mis en œuvre deux méthodes permettant d'estimer la justesse de la réponse du système de décision (1) :
   – la première méthode est une approche détection basée sur le principe du filtrage adapté,
   – la seconde méthode est une approche classification classique.
   Pour déterminer la meilleure méthode, nous avons quantifié théoriquement l'amélioration apportée par chaque méthode en terme de probabilité d'erreur en sortie du système corrigé.

3. **Une stratégie de mise à jour de la population d'apprentissage.** Une fois la réponse donnée par le SVM et l'avis de l'expert pris en compte, on doit intégrer ou retirer des exemples dans l'ensemble d'apprentissage afin de l'adapter à l'évolution de l'utilisateur et d'améliorer les décisions du BCI. Cependant toutes les données ne sont pas forcément pertinentes. Pour tester différentes stratégies, nous avons développé un simulateur.

1.4.3 Protocole d'acquisition du BCI étudié dans la thèse

Les différentes parties du BCI ont été testées de façon indépendante sur des signaux EEG enregistrés à l'Université d'Aalborg au Danemark ainsi qu'à l'Université de Göttingen en Allemagne. Le protocole correspondant à un fonctionnement complet en ligne ne pouvait être mis

en œuvre sans que chacun des modules du système ne soit développé et testé. Nous avons défini un protocole permettant d'enregistrer les signaux EEG_1 et de produire des signaux d'EEG_2 en réponse à un pseudo-feedback du BCI (la décision affichée par le BCI est aléatoire). Les résultats présentés dans la thèse utilisent les signaux enregistrés selon ce protocole.

Nous allons détailler dans un premier temps l'obtention des signaux EEG_1 (intention de mouvement) puis celle des EEG_2 avant de voir le déroulement d'une session complète. Tous les essais ont été réalisés sur des sujets sains ne souffrant d'aucun handicap sensorimoteur.

1.4.3.1 Enregistrement des EEG_1

L'enregistrement des EEG_1 correspond à un fonctionnement classique d'un BCI. Les instructions sont affichées sur un écran d'ordinateur. Un essai se décompose en quatre phases :

- **Concentration :** Dans la première phase on demande au sujet de se concentrer, il doit :
  - fixer l'écran,
  - ne pas bouger,
  - ne pas cligner des yeux.
- **Préparation :** Ensuite on affiche au sujet la tâche motrice qu'il devra effectuer. Dans notre protocole il y a deux possibilités :
  - une flexion rapide du bras droit,
  - une flexion lente du bras droit.

  Pendant que le type de mouvement s'affiche un compte à rebours se déclenche.
- **Tâche motrice :** A la fin du compte à rebours le sujet doit réaliser la tâche demandée. Durant les essais à l'Université d'Aalborg ils devaient réellement effectuer le mouvement. Lors des essais à l'Université de Göttingen ils devaient imaginer le mouvement sans le réaliser. Les différences électrophysiologiques entre une tâche motrice réelle et imaginaire se situent après le début de la tâche. Les différences dépendent des zones du cerveau. Cependant les potentiels des cortex frontal et moteurs sont similaires pour les mouvements réels et imaginaires [Sano et Bakardjian, 2009].
- **Maintien :** La durée de la phase de la tâche motrice variant suivant le type de mouvement, on demande ensuite au sujet de maintenir la contraction de son bras de façon réelle ou imaginaire. La durée de cette phase varie de manière à ce que la somme des durées des phases "Tâche motrice" et "Maintien" soit identique quel que soit le mouvement demandé au sujet.

Les EEG enregistrés 0.5s avant le début de la tâche motrice et durant 2s constituent les signaux d'EEG_1 que nous avons utilisés pour tester les différentes méthodes d'extraction de caractéristiques dans le chapitre 2.

1.4.3.2 Enregistrement des EEG_2

Une fois la tâche effectuée par le sujet, on enregistre les signaux EEG_2 qui correspondent à la réaction du sujet à la décision affichée par le module de classification de la tâche motrice. Pour obtenir ces signaux on affiche un texte (*lent/rapide*) sur l'écran d'ordinateur, qui correspond aléatoirement (pseudo-feedback) à une classification juste ou fausse (75% *juste* et 25 % *faux*).

Le pseudo-feedback permet d'isoler le problème de détection des potentiels d'erreur d'autres aspects (comme la variabilité des résultats de classification parmi les sujets). Malgré le fait que la réponse affichée est aléatoire, les utilisateurs pensent qu'elle correspond à la classification par le BCI de leur état mental.

1.4.3.3 Déroulement des enregistrements

Les enregistrements ont été effectués sur 6 sujets sains au SMI à Aalborg (réalisation du mouvement) et 4 sujets sains au BCNT à Göttingen (imagination du mouvement). On demande à chaque sujet de faire une session complète.

Une session est composée de 120 essais, un essai correspond à l'enregistrement des EEG_1, EEG_2 et d'un temps de repos (cf tableau 1.1 pour les temps de chaque phase).

TABLE 1.1 – Temps des différentes phases pour un seul essai

Concentration	Préparation	lent/rapide	Maintien	Affichage	Repos
2s	2s	2s/0.1s	1s/2.9s	1s	3s-4s

Dans les 120 essais nous avons :
- pour les EEG_1 : 60 flexions lentes et 60 flexions rapides,
- pour les EEG_2 : 30 pseudo-décisions de la classe *faux* et 90 pseudo-décisions de la classe *juste*.

Les essais contenants des mouvements des yeux sont rejetés.

Nous ne disposions pas du même matériel dans les deux Universités pour enregistrer les signaux EEG.
- Aalborg : signaux enregistrés sur 6 électrodes (F_z, FC_z, C_z, CP_z, C_1, C_2) avec une fréquence d'échantillonnage de 1024Hz. La bande passante de l'amplificateur utilisé est de 0.1-100Hz.
- Göttingen : signaux enregistrés sur 8 électrodes (F_z, FC_z, C_z, P_z, FC_1, FC_2, CP_1, CP_2) avec une fréquence d'échantillonnage de 1200Hz. La bande passante de l'amplificateur utilisé est de 0.1-400Hz.

Chapitre 2

Espaces de représentation optimaux à base d'ondelettes

Sommaire

La première étape du BCI adaptatif consiste à classer les différentes intentions de mouvement de l'utilisateur. Nous travaillons sur une interface devant permettre de discriminer différentes modalités d'un même mouvement imaginaire (lent/rapide, couple faible/fort). Pour ce faire un espace de représentation doit être choisi. C'est dans cet espace que les signaux EEG seront décrits et classés. Les performances d'un système de classification dépendent fortement du choix de l'espace de représentation. Celui-ci peut être choisi *a priori* ou bien adapté à l'objectif de classification. Les descripteurs que nous avons choisis sont issus des ondelettes. Cette représentation possède l'avantage de pouvoir être adaptée aux signaux à étudier en optimisant soit l'ondelette mère de la transformée discrète en ondelettes (DWT) soit la base de décomposition en paquets d'ondelettes (DWPT). Ces méthodes ont déjà prouvé leur efficacité sur des signaux EMG [Maitrot *et al.*, 2005b,a ; Lucas *et al.*, 2008] et EEG [Farina *et al.*, 2007 ; Vautrin *et al.*,

2009] à partir des travaux initiés par A. Maitrot [Maitrot, 2005] pour l'optimisation de l'ondelette mère et D. Vautrin [Vautrin, 2008] pour l'optimisation de la base de décomposition en paquets d'ondelettes. Dans le cadre de la thèse nous élargirons ces méthodes à des descripteurs différents et en proposant d'autres critères d'optimisation.

Nous présentons dans ce chapitre les différents espaces de représentation que nous avons utilisés dans le cadre de ce travail. Nous rappellerons tout d'abord les propriétés de la DWT et nous définirons les descripteurs issus des ondelettes utilisés pour la classification des signaux EEG. Nous détaillerons l'obtention des différents espaces de représentation en optimisant soit l'ondelette mère de la transformée discrète en ondelettes, soit la base de décomposition utilisée pour la transformée discrète en paquets d'ondelettes. Enfin nous analyserons les performances de la classification dans ces espaces de représentation sur des signaux simulés puis des signaux EEG réels. Nous commençons par présenter les signaux étudiés.

2.1 Potentiels corticaux relatifs aux mouvements

Plusieurs études ont montré la présence de potentiels liés à l'exécution ou l'imagination des tâches motrices (MRP, movement related potentials) au niveau du cortex moteur. Ces potentiels ont été décrits pour plusieurs types de mouvement :

- mouvement des yeux [Shibasaki *et al.*, 1981],
- mouvement de la main [Ikeda et Shibasaki, 1992],
- mouvement des jambes [Jentzsch et Leuthold, 2002].

Romero et al. [Romero *et al.*, 2000] ont mis en avant que les modèles neuronaux associés aux tâches motrices lorsque le mouvement est réellement effectué sont également présents lors de la simple imagination de ces même tâches.

Ce n'est que plus récemment que l'idée d'utiliser ces potentiels pour contrôler des BCI est apparue. Nascimento *et al.* [2006] ont illustré que les paramètres (lent/rapide - couple fort/faible) des tâches motrices modulaient l'activité corticale lors de l'imagination de flexions du pied (MRCP, Movement Related Cortical Potential). Nascimento et Farina [2008] ont ensuite prouvé que les MRCP permettaient la discrimination des paramètres cinétiques des flexions du pied. Les informations permettant de distinguer les différents types de mouvement se situent dans les basses fréquences (autour de 2 Hz) des signaux.

C'est sur ces signaux que nous avons décidé de travailler pour contrôler le BCI. L'étude sera basée sur la reconnaissance des différentes modalités d'un même mouvement qui sont : flexion lente/rapide du bras droit.

Analyse qualitative des signaux EEG_1 enregistrés. L'étude des signaux EEG sur un seul essai ne montre pas de différences significatives entre les deux classes. Sur la figure 2.1, nous montrons la moyenne et l'écart-type des signaux enregistrés sur l'électrode C_z pour un sujet effectuant réellement les tâches motrices (à gauche) et un sujet imaginant les tâches (à droite). Le signal représenté dure 2 s à partir de 0.5 s avant le début de la tâche. Un filtre passe bas de fréquence de coupure 10 Hz a été appliqué aux signaux afin de mettre en évidence les MRCPs.

On observe une décroissance du potentiel qui débute environ 500ms avant le début de la tâche et se termine légèrement avant le début de la tâche pour les mouvements lents et légèrement après pour les mouvements rapides. Cette décroissance est suivie d'une augmentation du potentiel. On retrouve des signaux similaires dans d'autres articles ([Nascimento *et al.*, 2006]). On constate que la variation de potentiel est plus rapide dans le cas de la tâche rapide, montrant que la tâche

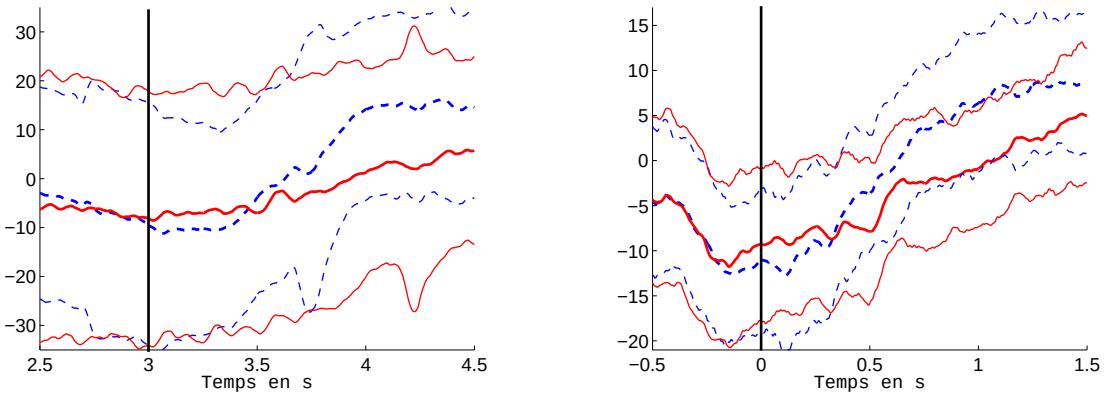


FIGURE 2.1 – Moyennes des signaux d'EEG$_1$ enregistrés sur l'électrode C_z pour les deux tâches motrices chez un sujet réalisant le mouvement (à gauche) et un sujet imaginant les mouvements (à droite). En trait pointillé : flexion du bras rapide. En trait plein : flexion du bras lente. Le trait vertical représente l'instant où le sujet doit commencer à réaliser la tâche motrice.

module bien les potentiels corticaux. L'observation de la variance montre cependant la très forte disparité des signaux au sein d'une même classe. On remarque aussi une variance des signaux plus importante dans le cas du sujet réalisant la tâche motrice. Cette différence peut venir du fait que le matériel pour enregistrer les EEG n'était pas le même pour les deux sujets.

En conclusion : l'observation des signaux non moyennés ne permet pas de déduire *a priori* des descripteurs pertinents. C'est pourquoi, en nous appuyant sur des travaux précédents nous testerons différents types de descripteurs en optimisant l'espace de représentation.

2.2 La transformée discrète en ondelettes (DWT)

Nous présentons dans cette partie les notions fondamentales sur la DWT qui sont utiles pour la définition et l'optimisation des espaces de représentation (pour plus de détails, voir les ouvrages [Abry, 1997 ; Mallat, 1999]). Etant donné un signal x et une ondelette mère ψ vérifiant les conditions d'admissibilité suivantes :

$$\int_{-\infty}^{+\infty} \psi(t)dt = 0 \tag{2.1}$$

$$\int_{0}^{+\infty} \frac{|\hat{\psi}(\xi)|^2}{\xi} dt < +\infty \tag{2.2}$$

la DWT est une transformée non redondante qui décompose le signal x sur la base d'ondelettes discrètes correspondant à toutes les versions de ψ dilatées et décalées suivant une grille dyadique. Elle fournit un ensemble de coefficients :

$$DWT_x(j,k) = \langle x(t), \psi_{j,k}(t) \rangle \tag{2.3}$$

avec $\psi_{j,k}(t) = 2^{-j/2}\psi(\frac{t-2^j k}{2^j})$, j et k correspondant aux indices d'échelle et de temps. L'ondelette $\psi_{j,k}$ analyse le signal à l'échelle 2^j autour de l'instant $2^j k$. En termes fréquentiels, la DWT réalise un banc de filtre : l'analyse aux grandes échelles rend compte du comportement basses fréquences du signal tandis que l'analyse aux petites échelles met en évidence son comportement aux hautes fréquences. La longueur de bande des filtres d'analyse n'est pas uniforme mais correspond à un découpage dyadique de l'axe fréquentiel.

2.2.1 L'analyse multirésolution

La théorie de l'analyse multirésolution (AMR), développée par Mallat [1999], Daubechies [2006], et Meyer [1990], donne un cadre formel à la construction de bases d'ondelettes présentant des propriétés d'orthogonalité. Elle permet notamment :

- la construction de base d'ondelettes $\{\psi_{j,k}\}_{(j,k)\in\mathbb{Z}^2}$ et leurs duales,
- la mise en œuvre pratique du calcul de la DWT à base de bancs de filtres sans qu'il soit nécessaire de manipuler des produits scalaires.

On peut dans ce cadre interpréter la DWT comme une décomposition du signal x en approximations et détails.

2.2.1.1 Espaces d'approximation, espaces d'ondelettes et DWT

Les espaces d'approximation : Une AMR de $\mathcal{L}^2(\mathbb{R})$ consiste en une suite croissante $\{V_j\}_{j\in\mathbb{Z}}$ de sous-espaces fermés de $\mathcal{L}^2(\mathbb{R})$ tels que :

- $$\bigcap_{j\in\mathbb{Z}} V_j = \{0\}, \quad \bigcup_{j\in\mathbb{Z}} V_j \text{ dense dans } \mathcal{L}^2(\mathbb{R}) \tag{2.4}$$
- la suite de $\{V_j\}_{j\in\mathbb{Z}}$ est croissante et les V_j sont emboîtés les uns dans les autres :
$$\forall j \in \mathbb{Z}, \quad V_{j+1} \subset V_j \tag{2.5}$$
- la projection d'une fonction $f \in \mathcal{L}^2(\mathbb{R})$ sur un de ces V_j constitue une approximation plus ou moins grossière de f et on peut passer d'un espace à l'autre par changement d'échelle dyadique :
$$\forall f \in \mathcal{L}^2(\mathbb{R}), \forall j \in \mathbb{Z}, \quad f(t) \in V_j \Leftrightarrow f(\frac{t}{2}) \in V_{j+1} \tag{2.6}$$
- chaque espace est invariant par translation :
$$\forall f \in \mathcal{L}^2(\mathbb{R}), \forall (j,k) \in \mathbb{Z}^2, \quad f(t) \in V_j \Leftrightarrow f(t-2^j k) \in V_j \tag{2.7}$$
- il existe une fonction ϕ telle que $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ soit une base de Riesz [1] de V_0 :
$$\forall x \in V_0, \quad x(t) = \sum_{k\in\mathbb{Z}} a[k]\phi(t-k) \tag{2.8}$$

Les propriétés vues précédemment permettent de définir des familles $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$, bases de Riesz de chaque V_j où :

$$\phi_{j,k}(t) = \frac{1}{\sqrt{2^j}}\phi(\frac{t-2^j k}{2^j}), \quad (j,k) \in \mathbb{Z}^2 \tag{2.9}$$

On passe donc d'un espace d'approximation à l'autre par changement d'échelle dyadique et chaque espace est invariant par translation. L'approximation d'un signal x au niveau j (i.e. sa projection sur un espace V_j) correspond à son analyse à la résolution (ou échelle) 2^j et est donnée par ses coefficients d'approximation :

$$a_j[k] = \langle x, \phi_{j,k} \rangle. \tag{2.10}$$

En définissant la base duale $\{\tilde{\phi}_{j,k}\}_{k\in\mathbb{Z}}$ de la base $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$, le signal d'approximation $A_j(t)$ de $x(t)$ au niveau j est :

$$A_j(t) = \sum_k a_j[k]\tilde{\phi}_{j,k}(t). \tag{2.11}$$

1. Une suite $\{\psi_i\}_{i\in\mathbb{Z}}$ forme une base de Riesz s'il existe A et B tel que $A\sum_i |c_i|^2 \leq \|\sum_i c_i\psi_i\|^2 \leq B\sum_i |c_i|^2$ avec c_i quelconques. Dans le cas d'une base orthonormale, nous avons $A = B = 1$ et $\|\sum_i c_i\psi_i\|^2 = \sum_i |c_i|^2$.

Les espaces de détail : L'approximation A_j étant plus grossière que A_{j-1}, l'information perdue entre deux approximations successives constitue les signaux de détail :

$$D_j(t) = A_{j-1}(t) - A_j(t) \tag{2.12}$$

Les signaux de détails sont des projections du signal x dans des espaces $\{W_j\}_{j\in\mathbb{Z}}$, appelés espaces d'ondelettes, où W_j est le complémentaire de V_j dans V_{j-1} :

$$V_j + W_j = V_{j-1} \tag{2.13}$$

On peut construire pour chaque W_j une base d'ondelettes $\{\psi_{j,k}(t) = \frac{1}{\sqrt{2^j}}\psi(\frac{t-2^j k}{2^j})\}_{k\in\mathbb{Z}}$ par dilatation et translation d'une ondelette mère ψ, elle même construite à partir de la fonction ϕ.

A partir des ondelettes $\psi_{j,k}$, on définit les coefficients de détail d'un signal x au niveau j par :

$$d_j[k] = \langle x,\psi_{j,k}\rangle = DWT_x(j,k). \tag{2.14}$$

En définissant la base duale $\{\tilde{\psi}_{j,k}\}_{k\in\mathbb{Z}}$ de la base $\{\psi_{j,k}\}_{k\in\mathbb{Z}}$, le signal de détail $D_j(t)$ de $x(t)$ au niveau j est :

$$D_j(t) = \sum_k d_j[k]\tilde{\psi}_{j,k}(t). \tag{2.15}$$

2.2.1.2 Les relations à deux échelles

D'après les propriétés vues précédemment, on peut écrire $\phi(\frac{t}{2})$ comme une combinaison linéaire des $\phi(t-k)$. Cette combinaison linéaire introduit l'existence d'une séquence génératrice de la fonction échelle (ou filtre échelle) $h = \{h[k]\}_{k\in\mathbb{Z}}$ et nous obtenons la première équation à deux échelles :

$$\phi(t/2) = \sqrt{2}\sum_k h[k]\phi(t-k). \tag{2.16}$$

De plus $\psi(\frac{t}{2}) \in W_1$ et $W_1 \subset V_0$. Ainsi $\psi(\frac{t}{2})$ peut s'écrire comme une combinaison linéaire des $\phi(t-k)$. En introduisant la séquence génératrice de la fonction ondelette (ou filtre ondelette) $g = \{g[k]\}_{k\in\mathbb{Z}}$, nous obtenons la seconde équation à deux échelles :

$$\psi(t/2) = \sqrt{2}\sum_k g[k]\phi(t-k). \tag{2.17}$$

Ces équations mettent en évidence l'existence de deux séquences génératrices h et g qui définissent la DWT. Ce sont ces deux séquences et non les fonctions ψ et ϕ qui sont utilisées en pratique pour son calcul et pour la conception d'ondelettes.

2.2.2 AMR et mise en œuvre de la DWT

Pour un signal à temps continu $x(t)$, la DWT est la collection des coefficients de la projection de x dans les différents espaces W_j :

$$DWT_x = \{d_j[k] = \langle x,\psi_{j,k}\rangle\}_{(j,k)\in\mathbb{Z}^2} \tag{2.18}$$

Les relations à deux échelles permettent d'exprimer le coefficient d'approximation et de détail à un niveau donné comme le filtrage des coefficients d'approximation du niveau précédent par les séquences génératrice $\overline{h}$ et $\overline{g}$ (où $\overline{h}[k] = h[-k]$), suivi d'une décimation :

$$a_j[k] =\downarrow 2(\overline{h} * a_{j-1,\cdot}[k]) \tag{2.19}$$

$$d_j[k] =\downarrow 2(\overline{g} * a_{j-1,\cdot}[k]) \tag{2.20}$$

et de calculer la DWT par la mise en œuvre du banc de filtre ($\overline{h}$,$\overline{g}$) (Algorithme de Mallat [Mallat, 1999]).

Pour un signal discret $x[k]$ de longueur N, on fait le plus souvent l'approximation $a_0[k] = x[k]$. Le nombre de niveaux de décomposition est fini et limité par la longueur du signal. La DWT est alors définie par :

$$DWT_x = \{\{a_J[k]\}_{k=0,\ldots,\frac{N}{2^J}-1},\{\{d_j[k]\}_{k=0,\ldots,\frac{N}{2^J}-1}\}_{j=1,\ldots,J}\} \tag{2.21}$$

avec $J \leq \log_2(N)$ le niveau de décomposition maximum. Afin de simplifier les notations on notera $a_j = \{a_j[k]\}_{k=0,\ldots,N/2^J-1}$, de même pour d_j.

2.3 Descripteurs issus de la transformée discrète en ondelettes

Dans la mesure où on ne sait pas *a priori* ce qui distingue les signaux, nous avons utilisé un approche pragmatique qui consiste à tester différents types de descripteurs pour classer les signaux EEG.

2.3.1 Coefficients de la DWT

La DWT est une représentation dont les coefficients correspondent à des séquences temporelles à différentes échelles. On peut utiliser ces coefficients comme descripteurs dans un contexte de classification si l'on suppose que les différents signaux sont suffisamment synchronisés pour que leurs vecteurs descripteurs soient comparables.

Pour un signal monovoie x de longueur N, les descripteurs seront les coefficients de détails de la DWT à chaque niveau j ainsi que l'approximation au dernier niveau de la décomposition $J \leq \log_2(N)$:

$$DWT_x = [a_J,d_1,\ldots,d_J]. \tag{2.22}$$

Pour un signal multivoies : $x = [x_1,\ldots,x_{Nv}]$, le vecteur descripteur sera composé du regroupement des descripteurs de chaque voie.

$$DWT_x = [DWT_{x_1},\ldots,DWT_{x_{N_v}}]. \tag{2.23}$$

Le risque de cette représentation est d'avoir trop de descripteurs, ce qui augmente le risque de surapprentissage. Dans ce cas, pour diminuer le nombre de descripteurs, on peut ne garder que les coefficients correspondant aux grandes échelles en se basant sur la connaissance *a priori* des signaux à classer.

Dans le cadre de la thèse nous savons que nous voulons distinguer deux types de MRCPs, ceux qui correspondent aux mouvements imaginaires lents et ceux qui correspondent aux mouvements imaginaire rapides. La fréquence des MRCPs se situant autour de $2Hz$, nous pouvons ne garder que les coefficients qui caractérisent cette bande de fréquences. (Pour un signal d'une durée de $2s$ et échantillonné à $1024Hz$, nous avons décidé de n'utiliser que l'approximation au niveau $J = 11$ et les 3 derniers niveaux de détail.)

2.3.2 Marginales de la DWT

La représentation précédente ne sera pas pertinente si les signaux ne sont pas parfaitement synchronisés entre eux. On peut alors ne pas tenir compte des informations temporelles et n'utiliser que les marginales fréquentielles de la décomposition. Ces marginales s'obtiennent en sommant l'ensemble des valeurs absolues des coefficients de la DWT correspondant à chaque bande de fréquence. Pour un signal x monovoie de longueur N les descripteurs seront :

$$\begin{cases} M_x(j) = \sum_k |d_j[k]|, & j = 1, \ldots, J \\ M_x(J+1) = \sum_k |a_J[k]|, \end{cases} \tag{2.24}$$

avec $J \leq \log_2(N)$ niveau de décomposition maximum.

Si le signal est composé de plusieurs voies : $x = [x_1, \ldots, x_{Nv}]$, le vecteur descripteur sera composé du regroupement des marginales de toutes les voies : $M_x = [M_{x_1}, \ldots, M_{x_{Nv}}]$.

Les marginales ont l'avantage de réduire la dimension de l'espace de représentation, mais on perd toute information temporelle. Cette représentation a été proposée par Aude Maitrot et appliquée à des signaux EMG [Maitrot *et al.*, 2005b] et EEG [Farina *et al.*, 2007]. Comme dans le cas précédent on sera amené à réduire la dimension de l'espace de représentation en retenant les bandes les plus pertinentes.

Nous allons voir dans la suite les différentes méthodes que nous avons utilisées pour optimiser ces descripteurs.

2.4 Optimisation de l'ondelette mère

Aude Maitrot, dans le même article [Maitrot *et al.*, 2005b], a introduit une méthode d'optimisation de l'ondelette mère pour la représentation de signaux et l'a appliquée à la classification de signaux EMG. Cette technique a été reprise pour des signaux EEG dans [Farina *et al.*, 2007]. Cette méthode propose d'une part d'utiliser une forme paramétrique de l'ondelette mère et d'autre part de définir un critère de qualité adapté à la classification.

2.4.1 Paramétrisation de l'ondelette

Comme nous l'avons vu, dans le cadre de l'analyse multirésolution (section 2.2.1), la DWT est définie à partir d'une fonction d'échelle $\phi(t)$ et d'une ondelette mère $\psi(t)$, ou de manière équivalente par les filtres d'échelle et d'ondelette h et g correspondants :

$$\begin{aligned} \phi(t/2) &= \sqrt{2} \sum_k h[k] \phi(t-k) \\ \psi(t/2) &= \sqrt{2} \sum_k g[k] \phi(t-k). \end{aligned}$$

Les équations à deux échelles montrent que l'ondelette ψ se construit à partir de la fonction échelle ϕ mais n'apportent aucun autre renseignement sur la procédure à suivre pour mener à bien cette construction. En effet le lien entre ψ et ϕ vient de la complémentarité entre les deux espace W_j et V_j. Or il existe plusieurs manières de choisir le complémentaire de V_j dans V_{j-1} ainsi que de définir les bases de ces espaces. Trois choix, du plus contraint au plus général, aboutissent aux ondelettes orthogonales, semiorthogonales et biorthogonales. Dans le cadre de notre étude nous nous limiterons aux ondelettes orthogonales.

Cas des ondelettes orthogonales Dans le cas d'ondelettes orthogonales, g peut être déduit de h à partir de la relation :

$$g[k] = (-1)^{1-k} h[1-k]. \tag{2.25}$$

Dans ce cas h définit ψ. Cependant pour créer une ondelette multirésolution, h doit satisfaire certaines contraintes. Pour un filtre à réponse impulsionnelle finie (RIF) de longueur L, il y a $L/2+1$ conditions suffisantes pour assurer l'existence et l'orthogonalité de la fonction échelle et des ondelettes [Lawton, 1991] :

$$\sum_{k=1}^{L} h[k] = \sqrt{2} \tag{2.26}$$

$$\sum_{k=1}^{L} h^2[k] = 0 \tag{2.27}$$

$$\sum_{k=1}^{L} h[k]h[k-2n] = 0 \quad n = 1,\ldots,L/2-1 \tag{2.28}$$

Ainsi il reste $L/2-1$ degrés de liberté pour créer le filtre h. La paramétrisation de Vaidyanathan et Hoang décrite dans [Vaidyanathan et Hoang, 1988] permet de construire h en fonction de $L/2-1$ nouveaux paramètres libres. Nous noterons θ le vecteur de ces paramètres. Par exemple si $L=6$, θ a 2 composantes. Les expressions de h en fonction de L sont données dans [Burrus *et al.*, 1997][Selesnick, 1997]. Pour des raisons de coût de calcul nous nous sommes limités aux ondelettes de longueur $L=4$, ainsi nous devons optimiser l'ondelette selon un seul paramètre θ. Dans ce cas h est défini par :

$$\begin{cases} h[i] = (1-\cos(\theta)+(-1)^i \sin(\theta))/(2\sqrt{2}) & \text{si } i = 0,3 \\ h[i] = (1+\cos(\theta)+(-1)^{i-1} \sin(\theta))/(2\sqrt{2}) & \text{si } i = 1,2 \end{cases} \tag{2.29}$$

Pour choisir θ optimal, il faut définir un critère d'optimisation. Nous présentons dans la partie suivante les deux critères que nous avons étudiés.

2.4.2 Critères de qualité

2.4.2.1 Critère de probabilité d'erreur

Le but de la recherche d'un espace optimal est de minimiser la probabilité d'erreur de classification. Le premier critère que nous avons utilisé pour optimiser les paramètres de l'ondelette mère est une estimation de la probabilité d'erreur de classification comme proposé dans [Farina *et al.*, 2007]. On considère un problème à n_c classes différentes. Il y a une erreur de classification chaque fois que l'on classe un signal x dans la classe ω_i, alors qu'il appartient à la classe ω_j avec $i \neq j$. Pour une ondelette paramétrée par θ, la probabilité d'erreur correspondante est :

$$P_e^{\theta}(\omega_i) = \text{Prob}(x \text{ affecté à } \omega_i | x \in \omega_j, j \neq i), 1, \ldots, n_c \tag{2.30}$$

La probabilité d'erreur globale est la moyenne pondérée par la probabilité *a priori* de chaque classe $P(\omega_i)$ des n_c probabilités $P_e^{\theta}(\omega_i)$:

$$P_e^{\theta} = \sum_{i=1}^{n_c} P_e^{\theta}(\omega_i) P(\omega_i) \tag{2.31}$$

Calcul empirique et validation croisée. Une estimation régularisée de cette probabilité d'erreur de classification est calculée à partir des données d'apprentissage en utilisant une procédure de validation croisée. Cela consiste à diviser l'ensemble des individus dont on connaît la classe en N sous-ensembles contenant environ le même nombre d'individus :

- $N-1$ des sous-ensembles constituent la population d'apprentissage, leurs individus servent à calculer une séparatrice ;
- le dernier sous-ensemble constitue la population de test, ses individus servent à évaluer la séparatrice préalablement calculée.

Pour les N configurations possibles, on calcule le pourcentage de mal classés. On effectue la moyenne des pourcentages de mal classés calculés dans les N configurations possibles pour obtenir la probabilité d'erreur de classification.

Lorsque N est égal au nombre d'individus nous sommes dans le cas du "leave-One-Out" (LOO).

On optimise θ en minimisant la probabilité d'erreur de classification. Une limitation de ce critère est le temps de calcul lors de la validation croisée. De plus ce critère se généralise mal lorsque la probabilité d'erreur est estimée sur de faibles populations d'apprentissage. C'est pourquoi nous proposons un second critère avec un faible coup de calcul et susceptible d'une meilleure généralisation avec des petites populations d'apprentissage.

2.4.2.2 Critère de Fisher

L'utilisation du critère de Fisher pour optimiser l'espace de représentation vient d'une constatation intuitive : pour pouvoir discriminer les classes facilement, les individus d'une même classe ne doivent pas être trop dispersés et la distance entre les centres des classes doit être la plus grande possible. Le critère de Fisher, à maximiser, est le rapport entre l'inertie interclasse et l'inertie intraclasse.

Soit un problème à n_c classes avec une population $\mathcal{X} = \{\mathcal{X}_1,...,\mathcal{X}_i,...,\mathcal{X}_{n_c}\}$, où $\mathcal{X}_i$ est composé de n_i signaux x appartenant à la classe ω_i. Pour un θ donné, chaque signal x de $\mathcal{X}$ est représenté par son vecteur de descripteurs D_x^θ (correspondant aux coefficients d'ondelettes DWT_x^θ ou aux marginales M_x^θ). Le critère de Fisher est alors défini par :

$$F(\theta) = \frac{\sum_{i=1}^{n_c} \frac{n_i}{n} \|\overline{D_i^\theta} - \overline{D^\theta}\|^2}{\sum_{i=1}^{n_c} \frac{n_i}{n} \overline{\overline{D_i^\theta}}} \tag{2.32}$$

avec :

- $\overline{D_i^\theta} = \frac{1}{n_i} \sum_{x \in \mathcal{X}_i} D_x^\theta$ le représentant de la classe ω_i,
- $\overline{D^\theta} = \frac{1}{n_c} \sum_{i=1}^{n_c} \overline{D_i^\theta}$ le centre de gravité de l'ensemble des classes,
- $\overline{\overline{D_i^\theta}} = \frac{1}{n_i} \sum_{x \in \omega_i} \|D_x^\theta - \overline{D_i^\theta}\|^2$ l'inertie de la classe ω_i.

2.4.3 Optimisation des paramètres dans le cas de signaux mono-voies

Les critères précédemment définis sont très irréguliers en fonction de θ. Il existe plusieurs méthodes d'optimisation globale : échantillonnage irrégulier, recuit simulé, filtrage particulaire, algorithme génétique... Le choix d'une technique d'optimisation pour minimiser le critère n'est pas l'objet de cette thèse. Nous avons utilisé un échantillonnage du vecteur de paramètres θ et minimisé le critère sur cette grille. Le temps de calcul de cette méthode reste raisonnable tant que la dimension de θ ne dépasse pas 2 (dans le cas d'ondelettes de longueur $L = 4$ la dimension de θ vaut 1). Le pas d'échantillonnage choisi est celui pour lequel une valeur plus petite n'entraîne pas d'amélioration notable.

2.4.4 Optimisation des paramètres dans le cas de signaux multi-voies

Pour des signaux multi-voies, deux méthodes peuvent être envisagées pour optimiser l'espace de représentation.

La première consiste à appliquer une ondelette mère identique sur toutes les voies. Ainsi il suffit de regrouper les descripteurs de chaque voies en un seul vecteur descripteur et de faire une recherche exhaustive selon un grille de paramètres θ, comme pour les signaux mono-voies. C'est cette approche qui a été utilisée dans les travaux précédents de l'équipe [Lucas *et al.*, 2008].

Nous faisons ici le choix de déterminer une ondelette optimale pour chaque voie. Le but étant d'obtenir un ensemble d'ondelettes globalement optimal au sens du critère à optimiser (Pourcentage de mal classés, Fisher). Cependant une analyse de toutes les combinaisons possibles des ondelettes mères n'est pas possible en pratique car trop coûteuse en temps de calcul. C'est pourquoi nous proposons un algorithme qui résout les problèmes d'optimisation mono-voie de manière séquentielle pour converger vers une solution sous-optimale.

Soit un signal $x = [x_1, \ldots, x_{N_v}]$ à N_v voies et $\Theta = [\theta_1, \ldots, \theta_{N_v}]$ le vecteur des paramètres des filtres des ondelettes associées à chaque voie. L'optimisation du vecteur de paramètres Θ se fait de la façon suivante :

- On initialise $\Theta = [\theta_1, \ldots, \theta_{N_v}]$ par des filtres de Daubechies de longueur 4. On utilise ces filtres comme initialisation car ils possèdent de bonnes propriétés. L'algorithme n'assurant pas une convergence vers un minimum global, il est important de partir d'une solution donnant de bons résultats.
- Puis on optimise θ_k (cas mono-voie), en fixant les valeurs de $[\theta_1, \ldots, \theta_{k-1}, \theta_{k+1}, \ldots, \theta_{N_v}]$, par une recherche exhaustive sur une grille échantillonnant θ_k et en calculant le critère en chaque nœud. Cette valeur temporaire de θ_k est utilisée pour mettre à jour Θ. Ensuite on optimise θ_{k+1} de la même façon. On itère jusqu'à la convergence, qui correspond à N_v (nombre de voies) étapes successives sans qu'il y ait de changement dans la valeur du critère.

L'algorithme est le suivant :

Initialisation :

- Initialisation de toutes les voies avec des filtres de Daubechies : $\Theta = [\theta_1, \ldots, \theta_{N_v}]$ avec $\theta_k = \theta_{daub}, \quad k = 1, \ldots, N_v$.
- calcul de l'ensemble des vecteurs descripteurs $\{D_x^{\Theta}\}_{x \in \Omega}$ (Ω ensemble des individus de l'apprentissage).
- Calcul du critère $F(\Theta)$.
- initialisation $nb\_change = 0$ et $k = 1$.

Optimisation

- Tant que $nb\_change < N_v$
  1 Choix d'une valeur de θ_k sur la grille de valeurs.
  2 Calcul de $\{D_{x_k}^{\theta_k}\}_{x \in \Omega}$ et mise à jour de $\{D_x^{\Theta}\}_{x \in \Omega}$.
  3 Calcul du critère $F(\Theta)$.
  - On répète les étapes [1 2 3] pour toutes les valeurs de θ_k et on choisi $\hat{\theta}_k$ qui minimise $F(\Theta)$
    Si il y a amélioration de $F(\Theta)$
      $nb\_change = 0$
      Mise à jour de Θ avec $\hat{\theta}_k$ et $\{D_x^{\Theta}\}_{x \in \Omega}$ avec $\{D_{x_k}^{\theta_k}\}_{x \in \Omega}$.
    Sinon
      $nb\_change = nb\_change + 1$
    fin
  - $k = k + 1$ (Voie suivante)

- Fin

Nous avons exposé différentes méthodes pour optimiser l'ondelette de la DWT. Cependant la DWT impose une exploration dyadique de l'axe des fréquences. Dans la partie suivante nous allons voir qu'il est possible d'optimiser également la base de la décomposition d'une façon adaptée aux signaux à classer.

2.5 Meilleure base de décomposition en paquets d'ondelettes

L'utilisation de la décomposition en paquets d'ondelettes (DWPT) permet de s'affranchir de l'exploration dyadique de l'axe fréquentiel imposée par la DWT et de sélectionner une base de décomposition adaptée à l'objectif et aux signaux. La méthode que nous allons étudier est basée sur la définition d'un critère de contraste entre les classes permettant d'obtenir la décomposition fréquentielle la plus pertinente (on définit ainsi le banc de filtres le plus discriminant). Ce critère est additif, ce qui permet l'utilisation d'un algorithme rapide de recherche de l'optimum ce qui n'aurait pas été possible en utilisant un critère de probabilité d'erreur.

L'algorithme a été proposé par [Vautrin, 2008] et je l'ai mis en œuvre, en collaboration avec D. Vautrin pour la classification de signaux EEG [Vautrin *et al.*, 2009 ; **?**].

2.5.1 Décomposition en paquets d'ondelettes (DWPT)

La DWPT est une généralisation de la DWT, qui réalise une exploration dichotomique et redondante du contenu fréquentiel d'un signal [Mallat, 1999]. Etant donnés une fonction échelle $\phi(t)$, les filtres échelle h et ondelette g, les ondelettes à la résolution $j+1$ sont définies par :

$$\begin{aligned} \psi_{j+1}^{2p}(t) &= \sum_k h[k]\psi_j^p(t-2^j k) \\ \psi_{j+1}^{2p+1}(t) &= \sum_k g[k]\psi_j^p(t-2^j k) \\ p = 0,..,2^j-1 & \end{aligned}$$

avec $\psi_0^0(t) = \phi(t)$. Ces fonctions sont organisées selon un arbre binaire où chaque nœud (paquet d'ondelettes) est une sous-base $\Psi_j^p = \{\psi_j^p(t-2^j k)\}_k$ explorant la bande de fréquence normalisée $2^{-(j+1)} \cdot [p,p+1]$ (Figure 2.2). Les coefficients de la décomposition d'un signal discret x de longueur N sont obtenus par l'extension de l'algorithme de Mallat [Mallat, 1999].

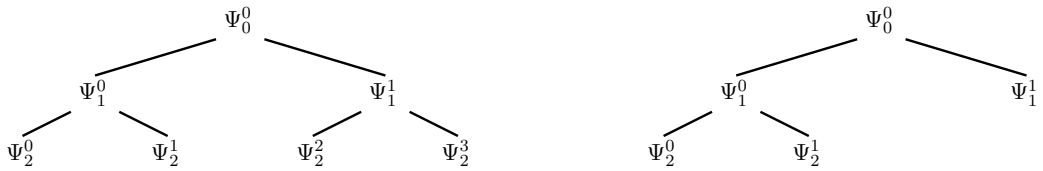


FIGURE 2.2 – A gauche, arbre de paquets d'ondelettes, avec $J=2$. La sous-base Ψ_j^p explore la bande de fréquence normalisée $2^{-(j+1)}[p,p+1]$. A droite, l'arbre des sous-bases de la DWT.

Chaque paquet de coefficients $\Psi_j^p x = \{d_j^p[k]\}_{k=0,N/2^j-1}$, où $d_j^p[k] = \langle x, \psi_j^p(t-2^j k)\rangle$, contient de l'information sur toute la durée du signal dans la bande de fréquence explorée par la sous-base Ψ_j^p. L'arbre entier de paquets d'ondelettes est très redondant (Ψ_j^p explore le même intervalle fréquentiel que l'union de ses deux fils $\Psi_{j+1}^{2p} \cup \Psi_{j+1}^{2p+1}$). Pour avoir une représentation non redondante

on doit extraire une base, c'est-à-dire un ensemble de paquets d'ondelettes $\{\Psi_j^p\}$ couvrant l'axe des fréquences sans recouvrement ; cela correspond aux feuilles d'un arbre admissible (c'est-à-dire un arbre dont chaque nœud a 0 ou 2 fils), comme illustré dans l'exemple de la figure 2.3.

La décomposition du signal x sur une base B est définie par :

$$Bx = \{\Psi_j^p x | \Psi_j^p \in B\} \tag{2.33}$$

Un exemple de base est donné Figure 2.3 avec les bandes de fréquence correspondantes.

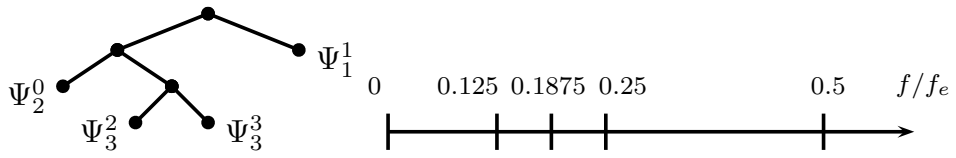


FIGURE 2.3 – Une base correspond aux feuilles d'un arbre binaire admissible, et explore l'axe fréquentiel sans recouvrement. On présente ici un exemple de base ($J = 3$) et les bandes fréquentielles correspondantes (f_e : fréquence d'échantillonnage).

Choix des descripteurs : Concernant les choix des descripteurs déduits de la DWPT, nous nous sommes limités à une description purement fréquentielle des signaux, nous affranchissant du risque que les signaux ne soient pas parfaitement synchronisés entre eux. Dans ce cadre, nous choisissons de représenter un signal par ses marginales ; elles sont définies en chaque nœud par :

$$M\Psi_j^p x = \sum_{k=0}^{N/2^j-1} |d_j^p[k]| \tag{2.34}$$

Pour une base ou une sous-base donnée B, le vecteur des descripteurs est donné par :

$$MBx = [\ \ldots\ M\Psi_j^p x\ \ \ldots\ ] \text{ avec } \Psi_j^p \in B \tag{2.35}$$

2.5.2 Sélection d'une meilleure base pour la classification

Comme nous l'avons vu précédemment, la DWPT est redondante et il faut extraire une base de décomposition. Pour sélectionner une base adaptée à la classification, il est nécessaire de définir un critère approprié ainsi qu'une stratégie de recherche pour l'optimiser. [Vautrin *et al.*, 2009] utilise ici la stratégie de Wickerhauser et Coifman [Coifman et Wickerhauser, 1992], et propose un nouveau critère additif, adapté à la décision multiclasse.

Stratégie de recherche. La stratégie développée par Wickerhauser et Coifman, initialement proposée dans le cadre de la compression de signaux, présente le grand intérêt d'être basée sur une recherche locale, ce qui évite le calcul du critère pour un nombre de bases prohibitif. ($> 2^{2^{J-1}}$ où J est le niveau de décomposition maximal). Elle consiste à considérer les nœuds de chaque niveau de décomposition, à partir du bas de l'arbre des paquets d'ondelettes jusqu'à la racine. Pour chaque nœud, on compare le coût de deux sous-bases possibles : celle correspondant au nœud considéré (père), et celle résultant de l'union de ses 2 fils ; on conserve la sous-base dont le coût est optimal. L'hypothèse sous-jacente est l'additivité du critère, qui garantit que l'algorithme atteint l'optimum global, et réduit considérablement le temps de calcul. Un critère $\mathcal{C}$ est additif si, étant données A_1 et A_2 deux sous-bases disjointes, $\mathcal{C}(A_1 \cup A_2) = \mathcal{C}(A_1) + \mathcal{C}(A_2)$.

Critère à optimiser. Dans un contexte de classification, on recherche donc un critère additif qui mesure le pouvoir discriminant d'une sous-base A vis-à-vis d'une population d'apprentissage. On note cette population $\mathcal{X} = \{\mathcal{X}_1,...,\mathcal{X}_i,...,\mathcal{X}_{n_c}\}$, où $\mathcal{X}_i$ est composé de n_i signaux x appartenant à la classe ω_i. Saito et Coifman [1994] ont proposé un critère additif correspondant à une mesure de contraste calculée à partir d'une distance (euclidienne ou de Kullback-Leibler) entre les représentants des classes. Ce critère, défini initialement pour comparer des représentations temps-fréquence, peut être adapté pour comparer des représentations fréquentielles (marginales). Pour une sous-base A donnée, chaque signal x de $\mathcal{X}$ est représenté par son vecteur de marginales MAx (comme défini dans l'équation 2.35). Le critère est alors défini par :

$$\mathcal{C}_1(A) = \sum_{i=1}^{n_c} \frac{n_i}{n} \|\overline{MA_i} - \overline{MA}\|^2 \tag{2.36}$$

où $\overline{MA_i} = \frac{1}{n_i} \sum_{x \in \mathcal{X}_i} MAx$ est le représentant de la classe ω_i et $\overline{MA} = \frac{1}{n_c} \sum_{i=1}^{n_c} \overline{MA_i}$. Ce critère est additif, mais ne prend pas en compte la dispersion des classes (inertie intraclasse). Afin d'utiliser cette information on peut utiliser le critère de Fisher (cf. équation 2.32). Cependant, ce critère n'est pas additif, et peut conduire à une base non optimale s'il est utilisé avec la stratégie de recherche vue précédemment. C'est pourquoi [Vautrin, 2008] propose le critère défini par :

$$\mathcal{C}_2(A) = K. \sum_{i=1}^{n_c} \frac{n_i}{n} \|\overline{MA_i} - \overline{MA}\|^2 - (1-K).(\sum_{i=1}^{n_c} \frac{n_i}{n} \overline{\overline{MA_i}}) \tag{2.37}$$

où $\overline{\overline{MA_i}} = \frac{1}{n_i} \sum_{x \in \omega_i} \|MAx - \overline{MA_i}\|^2$ est l'inertie de la classe ω_i.

Si Kest choisi entre 0 et 1, on conserve les mêmes propriétés que le critère de Fisher. Mais sous cette forme le critère est additif et donc approprié à l'utilisation de la stratégie de Wickerhauser et Coifman.

Les différentes méthodes vues dans ce chapitre ont été évaluées sur des signaux simulés et sur des signaux d'EEG.

2.6 Comparaison des méthodes sur des signaux simulés et réels

Les méthodes d'extraction des caractéristiques ont été comparées sur des signaux simulés et réels. Dans les deux cas les performances sont évaluées par le taux de mal classés sur une population de test indépendante de la population d'apprentissage. Dans le cas des signaux réels, le nombre total de signaux disponibles est relativement faible, c'est pourquoi on utilise une procédure de validation croisée.

2.6.1 Simulation des signaux

Les simulations n'ont pas pour objet de simuler de manière réaliste des EEG, mais de permettre la comparaison des méthodes de description à base d'ondelettes des signaux. L'objectif est de montrer sur des signaux dont on est certain qu'ils présentent des différences dans la localisation fréquentielle de l'information que l'optimisation de l'ondelette ou de la base de décomposition peut augmenter significativement les résultats de la classification. Dans ce cas on peut de plus évaluer les méthodes sur une population test de grande taille.

Les signaux simulés sont générés de façon à ce que chaque classe possède de l'énergie dans une bande de fréquence particulière. Pour ce faire nous avons utilisé une combinaison linéaire

d'atomes bien localisés en fréquence et de formes différentes : un ensemble d'atomes est associé à chaque classe et deux signaux d'une même classe diffèrent par une localisation aléatoire temporelle des atomes, qui est définie par une distribution Bernoulli. La simulation d'un signal x de longueur N est définie comme suit :

$$x[k] = \sum_{m=1}^{Nv^i} \sum_{k_0=0}^{N/2^{j_m}-1} q_m^i[k_0] v_m^i[k - 2^{j_m} k_0], \quad k = 0, \ldots, N-1 \tag{2.38}$$

où les variables sont :

- i correspond à la classe du signal x.
- $\{v_m^i\}_{m=1,\ldots,Nv^i}$ est l'ensemble d'atomes utilisés pour générer le signal de la classe ω_i.
- j_m correspond à la résolution fréquentielle de l'atome v_m^i
- q_m^i est un vecteur de longueur $N/2^{j_i}$ qui a pour valeurs 0 ou 1 suivant la distribution Bernoulli.

Un bruit blanc gaussien indépendant est ensuite ajouté.

Les signaux sont simulés de façon à avoir l'information localisée au niveau des nœuds représentés en rouge et entourés sur la figure 2.4 lorsque l'on regarde l'arbre de décomposition.

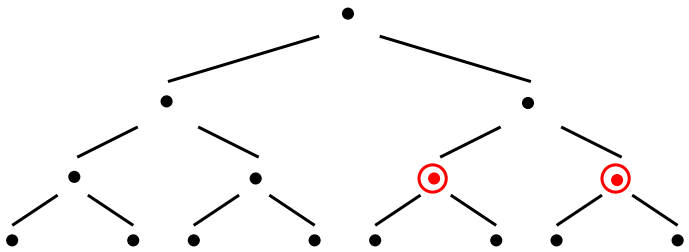


FIGURE 2.4 – Localisation de l'information dans l'arbre de décomposition temps-fréquence des signaux simulés.

2.6.2 Résultats sur des signaux simulés mono-voie

Une simulation est composée de deux classes séparées en un ensemble d'apprentissage de 30 signaux par classe et un ensemble de test de 1000 signaux par classes. Pour les deux classes les signaux sont de longueur $N = 256$ et le paramètre de la distribution Bernoulli est fixé à 0.05. Le rapport signal sur bruit est de 2dB.

Dans le Tableau 2.1 nous présentons les résultats obtenus avec les différentes méthodes.

TABLE 2.1 – Pourcentage de mal classés obtenus sur des signaux simulés.

Simulation	Marginales de la DWT	Optimisation de l'ondelette		Sélection de meilleure base
		critère du PCE	critère de Fisher	
1	45%	15%	15%	13%
2	45%	20%	20%	11%
3	49%	22%	17%	13%
4	44%	30%	18%	11%
5	44%	18%	18%	13%
Moyenne	45%±2%	21%±6%	18%±2%	12%±1%

Nous avons utilisé une ondelette de Daubechies d'ordre 2 pour la DWT sans optimisation d'ondelette et la méthode de sélection de meilleure base. On constate que les descripteurs issus des marginales de la DWT avec cette ondelette ne sont pas pertinents pour décrire les signaux (45% de mal classés). L'optimisation de l'ondelette améliore la reconnaissance des classes que ce soit en utilisant l'optimisation du pourcentage de mal classés (pce, 21% d'erreur) ou le critère de Fisher (Fisher, 18% d'erreur). Bien que les résultats de l'optimisation d'ondelette soient proches pour les deux critères, on constate grâce à la variance des résultats que le critère de Fisher donne des résultats plus stables montrant que la méthode semble plus robuste.

L'optimisation de la base de décomposition donne les meilleurs résultats (12% d'erreurs), ce qui est logique car l'information des signaux simulés est placée dans des bandes de fréquence qui sont grossièrement étudiées avec la décomposition dyadique de la DWT alors que la sélection de la meilleure base de décomposition explore plus finement ces bandes de fréquence.

2.6.3 Résultats sur des signaux EEG expérimentaux

Nous avons ensuite testé les différentes méthodes d'extraction des caractéristiques sur des signaux EEG expérimentaux obtenus selon le protocole décrit en Annexe 1.4.3. Nous disposons de 2 jeux de données :

- signaux où l'utilisateur effectue réellement la tâche motrice (6 sujets),
- signaux où l'utilisateur imagine la tâche motrice (4 sujets).

Signaux enregistrés lors d'une tâche motrice. Nous commencerons par montrer les résultats obtenus pour les signaux où le mouvement est réalisé puis nous verrons les résultats pour des signaux où le mouvement est imaginé.

Les quatre méthodes d'extraction des caractéristiques que nous avons appliquées sur ces signaux sont les suivantes :

- Marginales de la DWT en utilisant l'ondelette mère Daubechies 2.
- Marginales de la DWT en optimisant l'ondelette mère vis à vis de l'estimation du pourcentage de mal classés (PCE).
- Marginales de la DWT en optimisant l'ondelette mère vis à vis du critère de Fisher.
- Marginales de la DWPT en optimisant la base meilleure base de décomposition vis à vis du critère de Fisher.

Dans un premier temps nous avons uniquement utilisé la voie C_z pour classer les signaux. La fenêtre d'étude du signal débute 0.5 s avant que l'utilisateur effectue le mouvement et a une durée de 2 s. Concernant les marginales de la DWT nous n'avons pas pris tous les niveaux de détails pour décrire les signaux. Nous savons que la distinction entre les différentes tâches se fait à des fréquences très basses (autour de 2 Hz). Nous nous sommes donc limités aux trois derniers niveaux de détail et au coefficient d'approximation pour décrire les signaux (information dans la bande 0-4 Hz).

Les résultats obtenus sont présentés dans le Tableau 2.2. Les temps de calcul en moyenne sur tous les sujets pour effectuer 120 itérations de "leave-one-out" (LOO) sont donnés dans la dernière ligne.

On constate que par rapport à l'utilisation de l'ondelette Daubechies d'ordre 2, l'optimisation de l'ondelette mère pour la DWT améliore les résultats sur tous les sujets et diminue le pourcentage de mal classés de 5% en utilisant le PCE et de 6% en utilisant le critère de Fisher. Les résultats obtenus en utilisant les deux critères d'optimisation sont donc très proches, cependant le temps de calcul est bien supérieur lorsqu'on utilise le PCE. On remarque même que le

TABLE 2.2 – Résultats de la classification des mouvements en utilisant C_z comme voie.

Sujet	Marginales de la DWT	Optimisation de l'ondelette		Sélection de meilleure base
		critère du PCE	critère de Fisher	
1	**24%**	**24%**	**24%**	33%
2	50%	38%	30%	**18%**
3	33%	**29%**	34%	44%
4	34%	**33%**	38%	55%
5	35%	20%	**18%**	42%
6	16%	19%	**15%**	45%
Moyenne	32%±10%	27%±7%	**26%±8%**	39%±12%
Temps	10 s	409 s	9 s	44 s

temps de calcul de l'optimisation de l'ondelette avec le critère de Fisher (9 s) est inférieur au temps lorsqu'il n'y a pas d'optimisation (10 s). Ceci s'explique par le fait qu'à chaque étape de LOO on recalcule la séparatrice des SVM. Pour les méthodes sans optimisation et optimisation de l'ondelette avec le critère de Fisher, c'est ce calcul qui utilise la majeure partie du temps. Comme les classes sont plus facilement séparables avec l'optimisation de l'ondelette mère, on diminue le temps de calcul de la séparatrice des SVM et donc le temps total des 120 itérations de LOO.

Quant à la sélection de la meilleure base, exceptée pour le sujet 2, elle détériore toujours les résultats. Avec cette méthode on utilise toutes les marginales de l'arbre de la décomposition. Ainsi, dans le cas de nos signaux, on se retrouve avec un nombre important de descripteurs qui conduit au problème de la grande dimensionnalité de l'espace de représentation et n'améliore pas la classification, contrairement aux méthodes issues de la DWT où on se restreint à quatre descripteurs par voies. On en déduit que cette méthode n'est pas adaptée à l'étude de nos signaux.

Nous avons testé ensuite les quatre méthodes d'extraction de caractéristiques à des signaux multivoies. L'utilisateur effectuant une tâche motrice nous avons utilisé, parmi les canaux disponibles, ceux enregistrant l'activité cérébrale au-dessus du cortex moteur primaire. Les électrodes utilisées sont donc C_1, C_2 et C_z. Les résultats sont présentés dans le tableau 2.3.

TABLE 2.3 – Résultats de la classification des mouvements en utilisant C_z, C_1 et C_2 comme voies.

Sujet	Marginales de la DWT	Optimisation d'une ondelette par voie		Sélection de meilleure base
		critère du PCE	critère de Fisher	
1	27%	24%	**22%**	35%
2	32%	22%	**10%**	22%
3	36%	**29%**	30%	36%
4	26%	25%	**22%**	45%
5	37%	**16%**	26%	34%
6	14%	9%	**8%**	42%
Moyenne	29%±8%	21%±7%	**20%±8%**	36%±7%
Temps	10 s	1663 s	11 s	66 s

On constate que l'utilisation de plusieurs voies améliore les résultats avec toutes les méthodes montrant que l'information ne se situe pas uniquement au niveau de l'électrode C_z. L'optimisation d'une ondelette par voie améliore nettement les résultats, on passe de 29% de mal classés

avec l'ondelette Daubechies 2 à 21% et 20% en utilisant respectivement le critère du PCE et le critère de Fisher pour optimiser l'ondelette mère de la DWT. La sélection de la meilleure base de décomposition donnant toujours les moins bons résultats.

Finalement l'optimisation d'une ondelette par voie permet d'améliorer les résultats de la classification et on montre que la proposition d'utiliser le critère de Fisher plutôt que le PCE est pertinente. Les résultats pour les deux critères sont proches mais le gain en temps de calcul avec Fisher est très intéressant pour une utilisation en ligne de la méthode.

Signaux EEG enregistrés lors d'une tâche motrice imaginaire. Nous avons appliqué la méthode donnant les meilleurs résultats sur les signaux où l'utilisateur réalise la tâche motrice (optimisation de l'ondelette avec le critère de Fisher) aux signaux où l'utilisateur imagine le mouvement et nous l'avons comparée à la DWT avec l'ondelette de Daubechies 2. Sur ces signaux nous avons testé les marginales et les coefficients de la DWT comme descripteurs. Nous présentons les meilleurs résultats qui ont été obtenus avec les coefficients de la DWT des trois derniers niveaux de détail et de l'approximation. Nous avons donc 8 descripteurs par voies. Les résultats obtenus en utilisant C_z comme voie sont présentés tableau 2.4.

TABLE 2.4 – Résultats sur des signaux EEG de la tâche imaginaire en utilisant C_z comme voie et les coefficients de la DWT comme descripteurs

Sujet	Coefficients de la DWT	Optimisation de l'ondelette critère de Fisher
1	**28%**	32%
2	47%	**38%**
3	48%	**40%**
4	**22%**	**22%**
Moyenne	39%±11%	**33%±7%**

Les résultats sont moins bons que pour les signaux où les sujets effectuent réellement le mouvement. Cependant on note que l'optimisation de l'ondelette améliore les résultats. On passe de 39% de mal classés avec la DWT sans optimisation à 33% de mal classés en optimisant l'ondelette mère.

Comme pour les signaux de la partie précédente nous avons testé les méthodes d'extraction de caractéristiques en utilisant plusieurs voies. Cependant les signaux n'ayant pas été enregistrés dans le même laboratoire nous ne disposons pas des voies C_1 et C_2. Nous avons pris d'autres voies qui enregistrent l'activité cérébrale au-dessus du cortex moteur : C_z, FC_z, FC_1, FC_2, CP_1 et CP_2. Les résultats obtenus sont présentés tableau 2.5.

TABLE 2.5 – Résultats sur des signaux EEG de la tâche imaginaire en utilisant FC_z, FC_1, FC_2, C_z, CP_1 et CP_2 comme voie et les coefficients de la DWT comme descripteurs

Sujet	Coefficients de la DWT	Optimisation de l'ondelette critère de Fisher
1	**28%**	30%
2	45%	**43%**
3	37%	**23%**
4	23%	**13%**
Moyenne	33%±8%	**27%±11%**

Comme dans le cas des signaux avec mouvement, l'utilisation de plusieurs voies améliore les résultats de la classification. L'optimisation de l'ondelette mère pour la DWT permet de passer de 33% de mal classés à 27%.

Nous constatons qu'il est plus difficile de classer les signaux lorsque les utilisateurs imaginent les tâches motrices. L'imagination de telles tâches n'est pas naturelle pour quelqu'un de valide ce qui peut expliquer les différences de résultats. On constate que le sujet 4 présente de bien meilleurs résultats que les autres sujets. En effet ce sujet avait déjà une certaine expérience pour l'imagination motrice. Il est difficile de pouvoir conclure à partir d'un seul sujet, mais on voit qu'un sujet entrainé peut atteindre les mêmes performances en imaginant les tâches motrices que des sujets n'ayant jamais utilisé de BCI mais réalisant vraiment ces mêmes tâches.

2.7 Conclusion

Nous avons montré dans ce chapitre qu'il était possible de distinguer différentes modalités d'un même mouvement à partir d'enregistrements EEG. Nous voyons que la méthode consistant à optimiser une ondelette mère par voie améliore nettement les résultats de la classification par rapport à une DWT sans optimisation.

Les résultats que nous avons montrés précédemment demandent d'avoir un algorithme de classification. Dans la prochaine partie nous détaillerons la méthode des "Support Vector Machine" que nous avons utilisée comme classifieur. Nous verrons ses principes, ses avantages et comment l'utiliser pour développer une interface adaptative.

Chapitre 3

Classification

Sommaire

Une fois les descripteurs choisis, il nous faut délimiter les différentes classes à reconnaître dans l'espace de représentation. Il existe plusieurs méthodes permettant de calculer des fonctions discriminantes qui permettront d'estimer la classe d'un nouvel exemple. Dans le cadre de ma thèse nous disposons d'un ensemble d'apprentissage labélisé. Nous sommes donc dans un contexte de classification supervisée dont voici les méthodes plus connues :

- classification bayésienne,
- réseaux de neurones,
- machines à vecteurs supports.

Nous avons décidé de nous focaliser sur la méthode des machines à vecteurs supports car elle :

- ne fait aucune hypothèse forte sur la forme des classes,
- fonctionne avec de faibles populations d'apprentissage,
- possède une excellente capacité de généralisation,
- permet de s'affranchir des problèmes des espaces de représentation de trop grande dimension,
- évite le surapprentissage,
- possède une solution unique.

De plus il existe des implémentations de la méthode permettant un fonctionnement en ligne, ce qui est nécessaire pour notre application si nous voulons rendre le BCI adaptatif.

Dans la suite du chapitre, nous commencerons par exposer la théorie des SVM. Après un état de l'art sur les principaux algorithmes de résolution des SVM existants, nous détaillerons les algorithmes que nous avons testés.

3.1 Théorie des machines à vecteurs supports (SVM)

La théorie des SVM a été introduite et largement diffusée par Vapnik [1995]. Les SVM permettent de calculer une séparatrice dans le cas biclasse à partir d'un ensemble d'apprentissage labélisé.

Nous commencerons par présenter leur fonctionnement dans le cas simple du calcul d'une séparatrice linéaire avec des données linéairement séparables, puis nous verrons comment gérer le cas où les données ne sont pas séparables pour finir par le cas général du calcul d'une séparatrice non linéaire et l'utilisation des SVM lorsque l'on doit séparer plus de deux classes.

3.1.1 Notations

Pour mettre en place la technique des SVM, on dispose :
- d'une population d'apprentissage $\mathcal{X} = x_1, \ldots, x_m$ repartie en deux classe ω_{-1} et ω_1, où $x_i \in \mathbb{R}^d$ est le vecteur descripteur de l'individu i
- d'un ensemble d'étiquettes associées $\{y_i\}$ où y_i décrit la classe de x_i : $y_i = -1$ si $x_i \in \omega_{-1}$ ou $y_i = 1$ si $x_i \in \omega_1$.

L'apprentissage consiste, à partir de la liste de couples (x_i,y_i), à estimer une fonction de décision f telle que pour un individu x à classer sa classe y est donnée par la règle de décision :

$$y = \text{signe}(f(x)) \tag{3.1}$$

3.1.2 Séparatrice linéaire, cas séparable

3.1.2.1 Position du problème

Dans le cas où les données sont linéairement séparables alors il existe au moins un hyperplan séparant les deux classes d'individus. Nous rechercherons f sous la forme :

$$f(x) = w^t x + b = \langle w,x \rangle + b \tag{3.2}$$

De plus l'équation (3.1) nous donne les conditions suivantes :

$$\begin{cases} \langle w,x_i \rangle + b > 0 & \text{si } y_i = +1 \\ \langle w,x_i \rangle + b < 0 & \text{si } y_i = -1 \end{cases} \tag{3.3}$$

l'équation $\langle w, x_i \rangle + b = 0$ définissant l'hyperplan séparateur.

Cependant il existe une infinité d'hyperplans satisfaisants ces conditions. Sur la figure 3.1 quelques hyperplans satisfaisants les équations (3.2) et (3.3) ont été représentés, illustrant qu'il n'existe pas une unique solution.

Nous allons introduire la notion de marge, qui est l'espace situé de part et d'autre de la séparatrice et au sein duquel aucun exemple ne doit se trouver (cf. figure 3.2), pour sélectionner un des hyperplans. Dans le cas des SVM l'hyperplan choisi sera celui qui maximise la marge et il sera appelé hyperplan optimal, de plus il est unique.

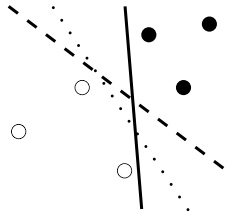


FIGURE 3.1 – Plusieurs hyperplans pouvant satisfaire la classification des données.

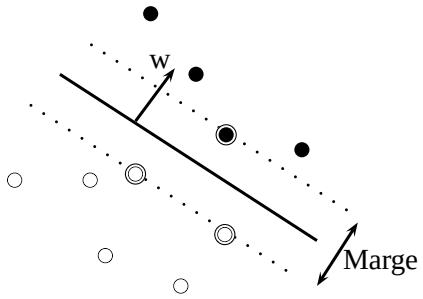


FIGURE 3.2 – Illustration de la notion de marge

L'hyperplan optimal se situe au milieu de la marge, les équations des deux hyperplans délimitant la marge sont :

$$\begin{cases} \langle w,x\rangle + b = \epsilon \\ \langle w,x\rangle + b = -\epsilon \end{cases} \tag{3.4}$$

en posant :

- $\frac{w}{\epsilon} \leftarrow w$
- $\frac{b}{\epsilon} \leftarrow b$

On obtient les équations normalisées :

$$\begin{cases} \langle w,x\rangle + b = 1 \\ \langle w,x\rangle + b = -1 \end{cases} \tag{3.5}$$

Pour maximiser la marge nous devons commencer par la calculer. La distance d'un point à la séparatrice est $(w^t x + b)/\|w\|$. Ainsi en prenant x_1 et x_2 sur les deux hyperplans qui maximisent la marge on obtient que la marge est donnée par $w^t(x_1 - x_2)/\|w\|$, soit d'après l'équation(3.5) : $2/\|w\|$. Le problème des SVM se résume donc à maximiser la marge sous la contrainte que les individus de l'apprentissage soient du bon coté de cette marge.

Maximiser $2/\|w\|$ revient à minimiser $\frac{1}{2}\|w\|^2$; nous sommes confrontés à un problème de minimisation quadratique sous contraintes :

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{3.6}$$

$$\text{s.c.} \quad y_i(\langle w,x_i\rangle + b) - 1 \geq 0 \quad i = 1,\ldots,m \tag{3.7}$$

3.1.2.2 Résolution du problème d'optimisation

Pour résoudre le problème d'optimisation précédent, il est fréquent d'utiliser les multiplicateurs de Lagrange afin de simplifier les contraintes du problème. En appliquant les conditions

de Khun-Tucker (KKT), on est amené à rechercher un point-selle (w^0,b^0,α^0) du Lagrangien

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i[y_i(\langle w,x_i\rangle + b) - 1] \quad \text{avec} \quad \alpha_i \geq 0 \quad \forall i \tag{3.8}$$

Ce point-selle vérifie les conditions de KKT :

$$\alpha_i^0[y_i(\langle w^0,x_i\rangle + b^0) - 1] = 0 \tag{3.9}$$

Les multiplicateurs de Lagrange α_i sont donc :

- positifs si la contrainte est saturée (égalité) ; ces individus se trouvent sur la marge ils sont appelés **vecteurs supports**,
- nuls ailleurs.

La résolution du problème ci-dessus est équivalente à la résolution du problème dual pour des fonctions convexes. On peut donc écrire ce problème de la façon suivante :

$$\min_{w,b}[\max_{\alpha} \mathcal{L}(w,b,\alpha)] = \max_{\alpha}[\min_{w,b} \mathcal{L}(w,b,\alpha)] \tag{3.10}$$

Le minimum doit annuler le gradient en w et b du Lagrangien on a donc :

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{3.11}$$

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{3.12}$$

Grâce aux équation (3.11) et (3.12) le problème dual se simplifie comme suit :

$$\max_{\alpha} \left[\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \right] \quad \text{s.c.} \quad \begin{cases} \alpha_i \geq 0 \quad \forall i \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases} \tag{3.13}$$

Nous sommes en présence d'un problème quadratique sous contraintes linéaires classique. De nombreux algorithmes d'optimisation ont été développés pour ce type de problèmes. Nous détaillerons ces algorithmes dans les prochaines sections.

Une fois les coefficients α_i calculés, les paramètres de l'hyperplan sont obtenus grâce aux équations (3.9) et (3.12) :

$$\hat{w} = \sum_{i=1}^{m} \hat{\alpha}_i y_i x_i \tag{3.14}$$

$$y_i(\langle \hat{w},x_i\rangle + \hat{b}) - 1 = 0 \quad \text{avec } i \text{ choisi tel que } \alpha_i \neq 0 \tag{3.15}$$

La fonction discriminante f est donnée par :

$$f(x) = \langle \hat{w},x\rangle + \hat{b} \tag{3.16}$$

3.1.3 Séparatrice linéaire, cas non séparable

Dans le cas où les individus ne sont pas séparables par un hyperplan mais que l'on veut effectuer une séparation linéaire nous devons reformuler les conditions du problème. Comme on sait qu'on ne peut pas séparer tous les individus, on autorise les mal classés en introduisant des variables de relaxation ξ_i (cf. figure 3.3). Ainsi les contraintes sont les suivantes :

$$\begin{cases} \langle w,x_i\rangle + b \geq 1 - \xi_i & \text{si } y_i = +1 \\ \langle w,x_i\rangle + b \leq -1 + \xi_i & \text{si } y_i = -1 \end{cases} \quad \text{avec } \xi_i \geq 0 \ \forall i \tag{3.17}$$

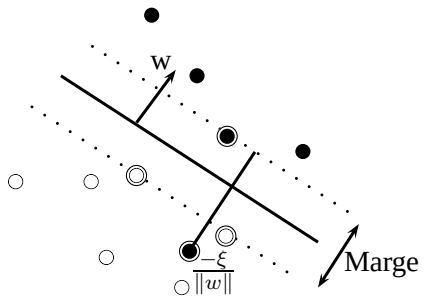


FIGURE 3.3 – Introduction des variables de relaxation ξ_i afin d'autoriser les mal classés

Si un individu x_i se trouve :
- du mauvais coté de la séparatrice alors $\xi_i > 1$,
- sur la séparatrice alors $\xi = 1$,
- entre la marge et la séparatrice on a $0 < \xi_i < 1$,
- sur la marge ou du bon coté de la marge alors $\xi_i = 0$

Les contraintes peuvent se condenser en :

$$y_i(\langle w,x_i\rangle + b) - 1 + \xi_i \geq 0 \qquad \xi_i \geq 0 \quad \forall i \in \{1,\ldots,m\} \tag{3.18}$$

Plus il y a d'individus mal classés plus le terme $\sum \xi_i$ est grand. Afin de limiter l'augmentation de ce terme, on le pénalise en l'introduisant dans la fonction à minimiser. Le coefficient C est introduit pour régler l'importance de cette pénalisation. La fonction à minimiser est :

$$\phi(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \tag{3.19}$$

Comme pour le cas séparable, la simplification du problème passe par l'utilisation du Lagrangien.

$$\mathcal{L}(w,b,\alpha,\xi,\mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\langle w,x_i\rangle + b) - 1 + \xi_i] - \sum_{i=1}^{m}\mu_i\xi_i \tag{3.20}$$

Les multiplicateurs de Lagrange :
- α_i sont introduits par la contrainte $y_i(\langle w,x_i\rangle + b) - 1 + \xi_i \geq 0$,
- μ_i sont introduits par la contrainte $\xi_i \geq 0$.

α_i et μ_i sont positifs $\forall i$

Les conditions de Kuhn-Tucker sont :

$$\alpha_i[y_i(\langle w,x_i\rangle + b) - 1 + \xi_i] = 0 \quad \forall i \tag{3.21}$$
$$\mu_i\xi_i = 0 \quad \forall i \tag{3.22}$$

On annule le gradient du Lagrangien :

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{3.23}$$
$$\frac{\partial \mathcal{L}}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{3.24}$$
$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \quad \Rightarrow \quad C - \alpha_i - \mu_i = 0 \quad \forall i \tag{3.25}$$

On obtient ainsi :

$$\alpha_i \geq 0 \quad \text{et} \quad \mu_i = C - \alpha_i \geq 0 \quad \Rightarrow \quad 0 \leq \alpha_i \leq C \tag{3.26}$$

On remarque que si un exemple x_i est :
- sur la marge, alors $y_i\langle w, x_i\rangle - 1 = 0$ et $\xi_i = 0$. On en déduit en utilisant l'équation (3.22) et l'équation (3.26) que α_i peut prendre n'importe quelle valeur entre 0 et C,
- du bon côté de la marge, alors $y_i\langle w, x_i\rangle - 1 \neq 0$ et $\xi_i = 0$. On en déduit en utilisant l'équation (3.22) que $\alpha_i = 0$,
- du mauvais côté de la marge, alors $\xi_i \neq 0$. On en déduit en utilisant l'équation (3.22) que $\mu_i = 0$ et donc en utilisant l'équation (3.26) que $\alpha_i = C$.

Le Lagrangien devient :

$$\left[\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \langle x_i,x_j\rangle\right] + (C - \alpha_i - \mu_i)\sum_{i=1}^{m} \xi_i \tag{3.27}$$

D'après l'équation (3.25) le second terme du lagrangien est nul et le problème dual se simplifie :

$$\max_{\alpha}\left[\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \langle x_i,x_j\rangle\right] \quad \text{s.c.} \quad \begin{cases} 0 \leq \alpha_i \leq C \ \forall i \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases} \tag{3.28}$$

De la même façon que dans la partie précédente, si on a $\hat{\alpha}_1, \ldots, \hat{\alpha}_m$ alors :

$$\hat{w} = \sum_{i=1}^{m} \hat{\alpha}_i y_i x_i \tag{3.29}$$

Pour déterminer $\hat{b}$ on utilise les conditions de Kuhn-Tucker. On choisit des points tels que $\xi_i = 0$, ce qui est le cas lorsque $0 < \alpha_i < C$. On utilise en général la moyenne de tous ces points. L'hyperplan que l'on obtient est appelé hyperplan optimal généralisé.

3.1.4 Séparatrice non linéaire, cas général

Nous allons voir dans cette partie comment les résultats vus précédemment peuvent se généraliser au cas où la séparatrice recherchée n'est pas une fonction linéaire des données.

Si on dispose d'un espace de dimension suffisamment grande, il est existe une séparatrice linéaire.

Par exemple, dans $\mathbb{R}^2$, la séparatrice peut être une ellipse d'équation :

$$ax_1^2 + bx_2^2 + cx_1x_2 + d = 0 \tag{3.30}$$

En se plaçant dans $\mathbb{R}^3$ avec le changement de variable :

$$z_1 = x_1^2 \tag{3.31}$$
$$z_2 = x_2^2 \tag{3.32}$$
$$z_3 = \sqrt{2}x_1x_2 \tag{3.33}$$

On obtient la séparatrice d'équation :

$$az_1 + bz_2 + \frac{c}{\sqrt{2}}z_3 + d = 0 \tag{3.34}$$

qui est linéaire dans $\mathbb{R}^3$

De façon plus générale, tout problème biclasse possédant m données d'apprentissage est linéairement séparable dans un espace de dimension $m-1$. L'idée fondamentale des SVM est de plonger les données dans un espace de représentation de dimension suffisamment grande pour qu ?une séparatrice linéaire soit pertinente. Pour changer d'espace nous utiliserons la fonction :

$$\phi : \mathbb{R}^d \to \mathcal{H} \tag{3.35}$$

où $\mathcal{H}$ est un espace de plus grande dimension (éventuellement infinie). Ainsi il suffira de choisir une fonction ϕ qui plongera les données dans un espace de dimension suffisamment grande et d'utiliser les résultats des SVM linéaires.

Cependant en pratique il n'est pas aisé de travailler avec ϕ explicitement, surtout que $\mathcal{H}$ peut être de dimension infinie. Nous pouvons remarquer que les données apparaissent toujours sous forme de produit scalaire $\langle x_i, x_j \rangle$ dans le problème des SVM. C'est ici qu'intervient l'astuce du noyau ("Kernel trick"). Si on prend $\mathcal{H}$ un espace RKHS (Reproducing Kernel Hilbert Space) alors il existe une fonction K telle que $K(x_i,x_j) = \phi(x_i).\phi(x_j)$. Le problème consiste donc à remplacer tous les produits scalaires par $K(x_i,x_j)$ dans les équations vues précédemment.

Les équations deviennent :

$$\max_{\alpha} \left[F(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i,x_j) \right] \quad \text{s.c.} \quad \begin{cases} 0 \leq \alpha_i \leq C \ \forall i \\ \sum_{i=1}^{m} \alpha_i y_i = 0 \end{cases} \tag{3.36}$$

Introduisons la forme matricielle du problème dual général. C'est sur cette forme que nous travaillerons principalement dans la prochaine section.

$$\begin{cases} \min\limits_{\alpha} & F(\alpha) = \frac{1}{2}\alpha^t Q \alpha - \mathbf{1}^t \alpha \\ s.c & y^t \alpha = 0 \\ & 0 \leq \alpha_i \leq C \end{cases} \tag{3.37}$$

avec :

- $\alpha = [\alpha_1, \ldots, \alpha_m]^t$,
- $y = [y_1, \ldots, y_m]^t$,
- $\mathbf{1} = [1, \ldots, 1]^t$,
- $Q_{ij} = y_i y_j K(x_i,x_j)$, cette matrice est nommée matrice pondérée du noyau.
- K dont $K_{ij} = K(x_i,x_j)$ est nommée matrice du noyau.

Si on note $\hat{\alpha}_1, \ldots, \hat{\alpha}_m$, les α_i solutions du problème d'optimisation, alors l'équation de la séparatrice est :

$$\sum_{i=1}^{m} \hat{\alpha}_i y_i K(x_i,x) + \hat{b} = 0 \tag{3.38}$$

$\hat{b}$ déterminé en vérifiant les conditions de Kuhn-Tucker :

$$y_i \sum_{j=1}^{m} \hat{\alpha}_j y_j K(x_j,x_i) - 1 + \hat{b} = 0 \tag{3.39}$$

où i est tel que $\hat{\alpha}_i \neq 0$.

Pour tout x à classer, la fonction de décision f est :

$$f(x) = \sum_{i=1}^{m} \hat{\alpha}_i y_i K(x,x_i) + \hat{b} \tag{3.40}$$

et la règle de décision :

$$y = \text{signe}(f(x)) \tag{3.41}$$

Voici quelques exemples de noyaux fréquemment utilisés dans les SVM :

- Noyau linéaire

$$K(x,x') = \langle x, x' \rangle \tag{3.42}$$

- Noyaux polynomiaux

$$K(x,x') = (1 + \langle x, x' \rangle)^p \tag{3.43}$$

- Noyau gaussien (RBF)

$$K(x,x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \tag{3.44}$$

Remarque : pour les SVM non linéaires nous n'avons pas formulé le problème primal. Nous nous sommes contentés de remplacer le produit scalaire des SVM linéaires par le noyau $K(.,.)$ dans la formulation duale. Certains auteurs introduisent la formulation non linéaire directement dans le problème primal et en utilisant ϕ [Evgeniou *et al.*, 2000] :

$$\begin{cases} \min\limits_{w \in \mathcal{H}, \xi, b} & \frac{1}{2}\|w\|^2 + C \sum\limits_{i=1}^{n} \xi_i \\ s.c & y_i(w^t \phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{cases} \tag{3.45}$$

La fonction de décision f est définie par $f(x) = w^t \phi(x) + b$ et $\mathcal{H}$ un espace à noyau reproduisant (RKHS, Reproducing Kernel Hilbert Space) muni du noyau $K(.,.)$. La solution du problème nous ramène à l'équation(3.40).

3.1.5 SVM pour des problèmes à plus de deux classes

Dans les sections précédentes nous avons vu l'utilisation des SVM pour des problèmes ne comportant que deux classes. Même si nous travaillons sur la reconnaissance de deux types de modalité de mouvement dans la thèse, les méthodes proposées doivent pouvoir s'appliquer à des problèmes à $c \geq 2$ classes afin de pouvoir les utiliser pour tous types de BCI. Il existe plusieurs méthodes permettant de prendre en compte plus de deux classes dont voici les principales :

- le "One Versus Rest",
- le "One Versus One",
- algorithmes de SVM tout-en-un (All-at-Once SVM) [Crammer et Singer, 2001].

Nous allons détailler ces trois méthodes.

One Versus Rest (OVR). Soit un problème à c classes, on sépare le problème en c problèmes biclasses et on détermine c fonctions de décision qui séparent une classe de toutes les autres. On défini f_i la fonction de décision qui sépare la classe ω_i des autres classes :

$$f_i(x) = w_i^t \phi(x) + b_i. \tag{3.46}$$

On voit sur la figure (3.4) que l'étude du simple signe des fonctions de décision ne permet pas de déterminer la classe d'un individu partout dans l'espace de représentation. Si pour un exemple x deux fonctions de décision sont positives ou si toutes les fonctions de décisions sont négatives, on se trouve dans une zone d'indétermination et on ne sait pas à quelle classe appartient l'individu x.

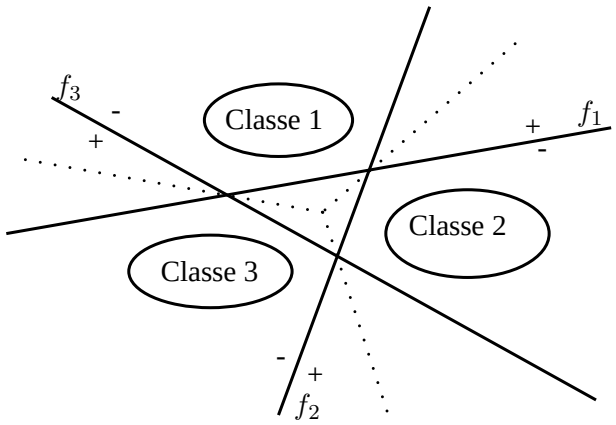


FIGURE 3.4 – Représentation des séparatrices en utilisant la méthode de OVR.

Cependant si on ne considère pas seulement le signe des fonctions de décision mais aussi leur module, on s'aperçoit que plus la valeur de $f_i(x)$ est petite, moins l'exemple x a de chance d'appartenir à la classe ω_i. Finalement pour un nouvel exemple x arrivant, la classe à laquelle il appartiendra sera celle dont il maximise la fonction de décision :

$$\hat{\omega}_i = \underset{\omega_i}{\operatorname{argmax}}\ f_i(x) \tag{3.47}$$

Cette méthode est simple à mettre en place et demande le calcul de c séparatrices. Un de ses principaux défauts est qu'elle gère des classes déséquilibrées. Par exemple si les individus sont répartis de façon équilibrée dans 10 classes différentes, on n'aura que 10% des individus dans une classe et 90% des individus pour le reste des classes lors du calcul d'une séparatrice.

One Versus One (OVO). Pour un problème à c classes, la méthode OVO consiste à calculer une séparatrice pour chaque combinaison possible de deux classes. On définit la fonction de décision entre les classes ω_i et ω_j :

$$f_{ij}(x) = w_{ij}^t \phi(x) + b_{ij} \tag{3.48}$$

Pour déterminer la classe à laquelle appartient un individu deux stratégies sont alors possibles :

– affecter x à la classe $\hat{\omega}_i$ telle que :

$$\hat{\omega}_i = \underset{\omega_i}{\operatorname{argmax}}\ f_i(x) \quad \text{avec} \quad f_i = \sum_{\substack{j=1 \\ j \neq i}}^{c} \operatorname{signe}(f_{ij}(x)) \tag{3.49}$$

Cependant avec cette règle de décision, on obtient une zone d'incertitude (sur la figure 3.5, cette zone correspond au polygone formé par les séparatrices).

– affecter x à la classe $\hat{\omega}_i$ telle que :

$$\hat{\omega}_i = \underset{\omega_i}{\operatorname{argmax}} \min_{\substack{j=1,\ldots,c \\ j \neq i}} f_{ij}(x). \tag{3.50}$$

Cette règle permet de lever toute indétermination.

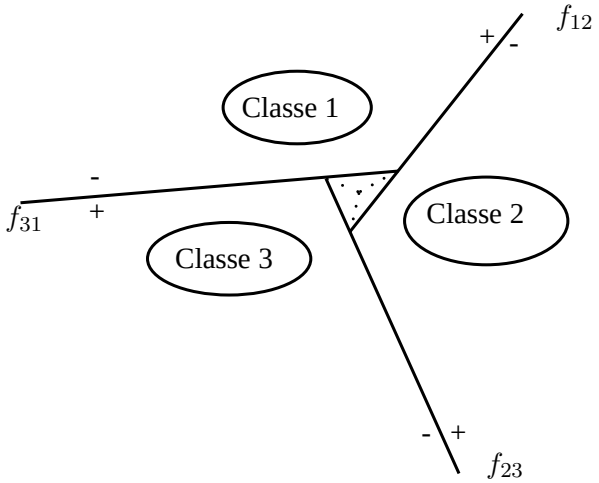


FIGURE 3.5 – Représentation des séparatrices en utilisant la méthode de OVO.

Au final, avec la méthode OVO, on calcule $c(c-1)/2$ fonctions de décision (c dans le cas OVR). Cependant chaque séparatrice demande la résolution d'un SVM biclasse plus petit et généralement plus équilibré que dans le cas OVR.

SVM multiclasses. Cette méthode consiste à déterminer toutes les séparatrices simultanément lors de la résolution du problème des SVM. Pour ce faire on modifie directement le problème d'optimisation à résoudre. Les fonctions de décisions à déterminer sont définie par :

$$f_i(x) = w_i^t \phi(x) + b_i \tag{3.51}$$

Pour qu'un individu x de la classe ω_i soit bien classé, alors $f_i(x)$ doit être maximum parmi l'ensemble des fonctions de décision $\{f_j(x), j = 1 \cdots, c\}$. On obtient l'inégalité suivante :

$$w_i^t \phi(x) + b_i > w_j^t \phi(x) + b_j \text{ pour } j = 1, \ldots, c, \quad j \neq i. \tag{3.52}$$

L'objectif des SVM reste le même que dans le cas biclasse : trouver les hyperplan qui maximisent la marge. Pour prendre en compte l'ensemble des marges ont modifie la fonction objectif et le problème devient :

$$\begin{array}{rl} \max\limits_{w_i \in \mathcal{H}, b_r, \xi} & \dfrac{1}{2}\sum\limits_{i=1}^{c} \|w_i\|^2 + C \sum\limits_{i=1}^{m} \sum\limits_{\substack{j=1 \\ j \neq y_i}}^{c} \xi_{ij} \\ s.c & w_i^t \phi(x) + b_i > w_j^t \phi(x) + b_j \text{ pour } j = 1, \ldots, c, \quad j \neq i. \\ & \xi_{ij} \geq 0 \end{array} \tag{3.53}$$

avec $y_i \in \{1, \ldots, c\}$.

Nous sommes encore en présence d'un problème de minimisation quadratique sous contraintes. Les méthodes de résolution des SVM biclasse s'appliquent toujours. Cependant les algorithmes d'optimisation doivent gérer un ensemble de "Support Vector" plus important avec cette méthode. En effet les "Support Vector" de chaque séparatrice sont traités en même ce qui augmente le temps de calcul.

Parmi les trois méthodes que nous avons expliquées aucune d'entre elle ne donne de meilleurs résultats par rapport aux autres pour tous les problèmes de classification.

Nous avons posé le problème des SVM dans le cas général à c classes où le problème n'est pas forcément séparable et la séparatrice recherchée non linéaire. Nous allons maintenant faire un état de l'art sur les différentes méthodes d'optimisation utilisées pour résoudre le problème des SVM, puis nous détaillerons plus précisément les méthodes que nous avons testées.

3.2 Les principaux algorithmes de résolution des SVM

Comme nous l'avons vu dans la partie précédente, les SVM consistent à résoudre un problème quadratique sous contraintes. Il est facile de trouver des programmes qui permettent de résoudre des problèmes quadratiques. Cependant beaucoup ne permettent pas de résoudre le problème des SVM à cause du temps d'entrainement trop important et des difficultés de stockage en mémoire de la matrice du noyau lorsque la population d'apprentissage excède quelques centaines d'exemples. Pour palier ces problèmes, plusieurs techniques ont été développées. Elles peuvent être principalement regroupées en deux groupes :

- les méthodes de point intérieur,
- les méthodes de décomposition.

Avant d'expliquer le principe de ces méthodes, commençons par relever certaines propriétés du problème qui seront utiles pour la résolution du problème d'optimisation.

3.2.1 Reformulation du problème d'optimisation

Rappelons la forme matricielle duale des SVM :

$$\begin{cases} \min\limits_{\alpha} & F(\alpha) = \dfrac{1}{2}\alpha^t Q \alpha - \mathbf{1}^t \alpha \\ s.c & y^t \alpha = 0 \\ & 0 \leq \alpha_i \leq C \end{cases} \tag{3.54}$$

avec

- $\alpha = [\alpha_1, \ldots, \alpha_m]^t$,
- $y = [y_1, \ldots, y_m]^t$,

– $\mathbf{1} = [1, \ldots, 1]^t$,
– $Q_{ij} = y_i y_j K(x_i, x_j)$.

En utilisant de nouveau le Lagrangien sur la contrainte d'égalité du problème 3.54 et en utilisant λ comme multiplicateur de Lagrange, on a :

$$\begin{cases} \min\limits_{\alpha,\lambda} & W(\alpha,\lambda) = \dfrac{1}{2}\alpha^t Q\alpha - \mathbf{1}^t\alpha + \lambda y^t\alpha \\ s.c & 0 \leq \alpha_i \leq C \end{cases} \tag{3.55}$$

La dérivé première de W par rapport à α_i et l'équation (3.40) nous donnent l'équation suivante :

$$g_i = \frac{\partial W}{\partial \alpha_i} = \sum_{j=1}^{m} Q_{ij}\alpha_j + \lambda y_i - 1 = y_i f(x_i) + (\lambda - b)y_i - 1 \tag{3.56}$$

Nous allons répartir les coefficients de Lagrange α en trois sous-ensembles distincts. En regardant le critère à optimiser W en fonction de α_i, on s'aperçoit qu'il existe trois configurations possibles en fonction de la position du minimum de $W(\alpha_i)$ par rapport aux contraintes du problème $0 \leq \alpha_i \leq C$. Sur la figure 3.6 nous avons représenté ces trois configurations (cf. remarque Section 3.1.3) :

– Le minimum non contraint de $W(\alpha_i)$ est en $\alpha_i < 0$. Dans ce cas on dira que i appartient à l'ensemble des exemples bien classés I_r (Remaining vectors). Le coefficient de Lagrange associé à ces exemples est $\alpha_i = 0$.
– Le minimum non contraint de $W(\alpha_i)$ est compris entre 0 et C. Dans ce cas on dira que i appartient à l'ensemble des exemples vecteurs supports I_s (Support Vectors). Le coefficient de Lagrange associé à ces exemples est $0 \leq \alpha_i \leq C$
– Le minimum non contraint de $W(\alpha_i)$ est en $\alpha_i > C$. Dans ce cas on dira que i appartient à l'ensemble des exemples mal classés (Error vectors) I_e. Le coefficient de Lagrange associé à ces exemples est $\alpha_i = C$.

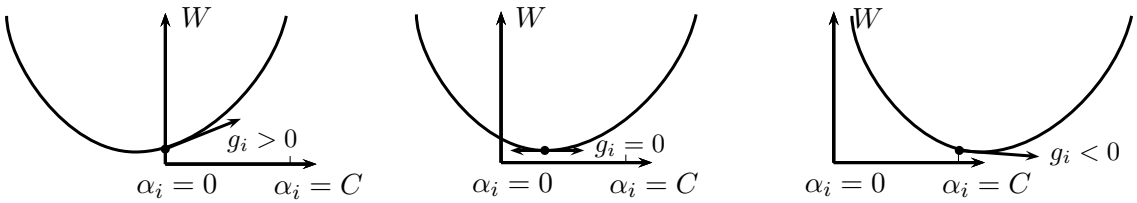


FIGURE 3.6 – Plage des valeurs de α_i et signe de g_i en fonction de la position de l'optimum de $W(\alpha_i)$

De plus en regardant la figure 3.6 et en utilisant les différents sous ensembles, la fonction objectif $W(\alpha)$ sera minimale lorsque $\forall i$ les conditions de KKT sont satisfaites (Section 3.1.3) :

$$\begin{array}{l} i \in I_s \ (0 \leq \alpha_i \leq C) \text{ et } g_i = 0 \\ i \in I_e \ (\alpha_i = C) \text{ et } g_i < 0 \\ i \in I_r \ (\alpha_i = 0) \text{ et } g_i > 0 \end{array} \tag{3.57}$$

Nous avons introduit un coefficient λ lors du passage de la formulation de l'équation (3.54) à l'équation (3.55). On sait que si x_i est vecteur support alors $y_i f(x_i) - 1 = 0$. Ainsi en identifiant à partir de $g_i = y_i f(x_i) + (\lambda - b)y_i - 1 = 0$, on en déduit :

$$\lambda = b \tag{3.58}$$

La forme du problème des SVM que nous utiliserons par la suite est :

$$\begin{cases} \min\limits_{\alpha,b} & W(\alpha,\lambda) = \frac{1}{2}\alpha^t Q\alpha - \mathbf{1}^t\alpha + by^t\alpha \\ s.c & 0 \leq \alpha_i \leq C \end{cases} \tag{3.59}$$

et les dérivées premières de la fonction à minimiser sont :

$$g_i = \frac{\partial W}{\partial \alpha_i} = y_i f(x_i) - 1 \tag{3.60}$$

$$\frac{\partial W}{\partial b} = \sum_{j=1}^{m} y_j \alpha_j = 0. \tag{3.61}$$

3.2.2 Méthodes de point intérieur.

Les méthodes de point intérieur (MPI) s'appuient sur l'algorithme proposée par Karmarkar en 1984 [Karmarkar, 1984]. Elles ont été développées à l'origine pour résoudre des problèmes d'optimisation linéaire en temps polynomial :

$$\begin{array}{ll} \min\limits_{x} & c^t x \\ \text{s.c} & Ax = b \\ & x \geq 0 \end{array} \tag{3.62}$$

On note P l'ensemble des solutions réalisables : $P = \{x : Ax = b, x \geq 0\}$. Les éléments x tels que $\{x \in P, x > 0\}$ sont appelés points intérieurs. L'idée principale des MPI est qu'il peut être difficile de minimiser $c^t x$ avec $x \in P$, mais que c'est facile pour x appartenant à une ellipsoïde. De plus la solution dans ce cas possède une forme analytique. Au lieu de résoudre le problème directement sur P on va résoudre une suite de problèmes d'optimisation sur des ellipsoïdes :

- On commence en un point $x^0 > 0$ à l'intérieur de P. On forme une ellipsoïde S_0 centrée en x^0 et contenue dans l'intérieur de P. On optimise $c^t x$ pour $x \in S_0$ ce qui donne un autre point intérieur x^1,
- on construit une autre ellipsoïde S_1 centrée en x^1 et on optimise de nouveau $c^t x$ avec x sur cette nouvelle ellipse,
- on réitère le processus jusqu'à convergence.

Contrairement à l'algorithme du simplexe, cette méthode atteint l'optimum du problème en passant par l'intérieur de P.

Les MPI ont ensuite été étendues à la programmation quadratique semi-définie positive et non linéaire par Vanderbei [1998] et Gertz et Wright [2001], ce qui a donné respectivement les algorithmes de type LOQO et OOQP.

Plusieurs auteurs ont ensuite amélioré ces algorithmes pour les appliquer au problème des SVM :

- Fine et Scheinberg [2002] montrent qu'il est possible d'accélérer les MPI si la matrice du noyau Q est de rang faible. Ils ont ensuite proposé une méthode permettant d'approcher cette matrice par une matrice de rang faible.
- Ferris et Munson [2002] ont montré qu'il était possible d'appliquer les MPI à des problèmes de SVM de grandes tailles.

De nombreuses autres méthodes ont découlé de ces études. Cependant bien qu'à la pointe de la recherche en optimisation et très efficaces, ces méthodes ne fonctionnent pas de manière en-ligne et demandent de faire une approximation de la matrice du noyau lorsque celle-ci n'est pas de

rang faible pour pouvoir résoudre des problèmes de grande taille. En effet, les méthodes de points intérieurs requièrent de partir de la base d'apprentissage complète, de plus elles n'exploitent pas la particularité des problèmes rencontrés en apprentissage qui est la parcimonie contrairement aux méthodes de décomposition.

3.2.3 Méthodes de décomposition

L'idée des méthodes de décomposition est d'optimiser le critère de façon itérative en utilisant seulement un sous-ensemble de l'apprentissage à chaque itération. Chaque itération doit optimiser le critère global du problème des SVM.

Chunking. Vapnik [1982] introduit la méthode sous le nom de "Chunking". L'algorithme utilise le fait que la valeur de la fonction objectif ne change pas si on enlève les lignes et les colonnes de la matrice correspondant aux multiplicateurs de Lagrange qui sont nuls ($\alpha_i = 0$). Ainsi, on peut diviser le problème quadratique en une série de problèmes quadratiques plus petits, dont le but est d'identifier tous les multiplicateurs de Lagrange non nuls ($\alpha_i \neq 0$) et d'éliminer tous ceux qui sont nuls. A chaque étape, le "Chunking" résout un problème quadratique constitué :

- de tous les exemples de l'itération précédente dont le α_i est différent de zéro,
- et des M exemples qui satisfont le moins les conditions de KKT.

La limite de cette méthode est la taille de la solution finale car on ne peut pas nécessairement stocker l'intégralité de la matrice noyau en mémoire.

Décomposition d'Osuna. Cette méthode est presque identique à celle du "Chunking" sauf que le sous-problème quadratique résolu à chaque itération est de taille fixe :

- on enlève M exemples du sous-ensemble sur lequel on a résolu le problème quadratique à l'itération précédente,
- on ajoute M exemples qui n'ont pas servi à l'itération précédente pour créer le nouveau problème quadratique.

En 1997 Osuna *et al.* [1997] montre que si parmi les M exemples ajoutés au nouveau sous problème, à chaque itération, au moins un de ces exemples viole les conditions de KKT alors la méthode converge vers l'optimum du critère (pas nécessairement un nombre fini d'itérations). De plus Osuna propose à chaque itération d'ajouter un seul élément qui ne satisfait pas les conditions de KKT et d'enlever un des éléments du sous-ensemble de l'itération précédente (on peut enlever des "Support Vector" du sous-ensemble). Contrairement au "Chunking" il n'y a pas de problème de mémoire quelle que soit la taille du problème à résoudre.

Sequential Minimal Optimisation (SMO). Platt [1999] propose le cas extrême de la décomposition qui consiste à ne considérer que deux points à chaque étape. Chaque sous problème quadratique a une taille suffisamment restreinte pour être résolu analytiquement, évitant ainsi une résolution numérique coûteuse. Cette méthode est l'une des plus utilisée actuellement. Une implémentation très connue est LIBSVM [Chang et Lin, 2011].

Méthodes de contraintes actives Cette méthode consiste à répartir le plus efficacement possible les exemples dans les sous-ensembles I_r, I_s et I_e, définis Section 3.2.1. On connait les valeurs de α pour les éléments appartenant aux sous-ensembles I_r et I_e. Quant aux éléments de I_s, la valeur de leur coefficient de Lagrange est obtenue en résolvant un système linéaire. Les implémentations de cette méthode diffèrent souvent par la technique employée pour résoudre

le système linéaire ou mettre à jour les différents sous-ensembles. Une implémentation souvent utilisée de cette méthode appliquée aux SVM est SimpleSVM [Vishwanathan *et al.*, 2003]. Ces méthodes sont très efficaces pour des problèmes de petite taille et de taille moyenne. De plus grâce à leur structure itérative elle sont facilement implémentables en ligne.

C'est vers ces méthodes de contraintes actives que nous nous sommes orientés pour réaliser le système de classification on-line du BCI.

3.3 Algorithmes de contraintes actives étudiés et leur fonctionnement en ligne

Nous avons testé deux algorithmes de résolution des SVM pour notre BCI. Les deux algorithmes peuvent être classés dans la catégorie des méthodes de contraintes actives. Nous allons détailler leur fonctionnement.

3.3.1 Incremental/decremental SVM

Le premier algorithme que nous avons testé est celui présenté dans [Cauwenberghs et Poggio, 2001]. Il a été spécialement développé pour un fonctionnement incrémental (ajout d'un individu) et décrémental (retrait d'un individu). C'est une méthode parfaitement adaptée à notre application pour laquelle on doit faire évoluer la population d'apprentissage au cours du temps. Nous commencerons par introduire le fonctionnement incrémental avant de développer les équations mises en jeu puis nous expliquerons comment adapter l'algorithme au cas décrémental.

L'objectif est de résoudre le problème (3.59).

Fonctionnement incrémental On suppose que l'on a la solution des SVM pour un ensemble d'apprentissage $I = \{I_s \cup I_e \cup I_r\}$. On ajoute un nouvel individu noté a, $I^{\text{new}} = \{I \cup \{a\}\}$. On doit déterminer son coefficient α_a ainsi que l'impact de cet ajout sur les coefficients α_i des autres individus déjà présents et sur le coefficient b.

Au début de l'algorithme on initialise $\alpha_a = 0$. Il va être mis à jour itérativement par incrémentations successives.

Plusieurs cas vont se présenter. Tout d'abord nous allons regarder si a ne fait pas partie directement des "remaining vectors" :

I - $g_a > 0$: l'individu a fait partie des "remaining vectors"
$I_r \leftarrow I_r \cup \{a\}$,
α_a reste égal à 0.
Tous les autres coefficient α_i et b restent inchangés.
On peut passer à l'ajout d'un nouvel individu.

II - $g_a \leq 0$: a ne fait pas partie des "remaining vectors".
Dans ce cas beaucoup de changements peuvent se produire.
Chacun de ces changements va être étudié indépendamment des autres dans l'ordre de leur apparition lors de l'augmentation de α_a.
Ainsi **on va augmenter** α_a **de** $\Delta\alpha_a$ tel que le premier des évènements suivants se produise [1] :

1. Pour chacune des conditions de changement évoquées à l'étape II, on calcule le $\Delta\alpha_a$ nécessaire de manière analytique selon les équations (3.67) (3.68) (3.69). On retient le plus petit et la condition associée.

1 - $g_a = 0$: a devient vecteur support
$I_s \leftarrow I_s \cup \{a\}$,
Mise à jour des $\mathcal{Q}$, $\mathcal{R}$, α_i, b et g_i selon les équations (3.66) (3.70) (3.67) (3.68) (3.69).
On peut passer à l'ajout d'un nouvel individu.

2 - $\alpha_a = C$: a devient "error vector"
$I_e \leftarrow I_e \cup \{a\}$,
Mise à jour des $\mathcal{Q}$, $\mathcal{R}$, α_i, b et g_i selon les équations (3.66) (3.70) (3.67) (3.68) (3.69).
On peut passer à l'ajout d'un nouvel individu.

3 - Un des éléments migre entre les ensembles I_s, I_e, I_r. Plusieurs migrations sont possibles :
a - $k \in I_e$ et $g_k = 0$: k devient vecteur support,
b - $k \in I_r$ et $g_k = 0$: k devient vecteur support,
c - $k \in I_s$ et $\alpha_k = C$: k devient "error vector",
d - $k \in I_s$ et $\alpha_k = 0$: k devient "remaining vector".
Une fois la migration déterminée on remet à jour les différents ensembles I_e, I_s et I_r.
Mise à jour des $\mathcal{Q}$, $\mathcal{R}$, α_i, b et g_i selon les équations (3.66) (3.70) (3.67) (3.68) (3.69).
On recommence l'étape II jusqu'à ce que l'évènement 1 ou 2 survienne[2].

Pour déterminer la configuration dans laquelle nous allons nous trouver, l'étude de g_i et α_i est la seule chose dont nous ayons besoin. Dans la suite nous verrons comment on peut exprimer simplement les variations Δg_i et $\Delta\alpha_i$ en fonction de $\Delta\alpha_a$ lorsque les éléments des ensembles I_s, I_e et I_r ne migrent pas.

Equations utilisées lors de la mise à jour. On appelle étape n l'étape avant l'incrémentation $\Delta\alpha_a$ et étape n+1 l'étape suivante. Tant qu'aucun individu ne migre entre les sous-ensembles I_s, I_e et I_r on peut écrire g_i grâce à l'équation (3.60) :

$$\begin{array}{lllllll} \text{Etape } n & g_i^n & = & Q_{ia}\alpha_a^n + & \sum_j Q_{ij}\alpha_j^n & + \; y_i b^n & \text{-1} \\ \text{Etape } n+1 & g_i^{n+1} & = & Q_{ia}\alpha_a^{n+1} + & \sum_j Q_{ij}\alpha_j^{n+1} & + \; y_i b^{n+1} & \text{-1} \end{array}$$

Alors la variation de g_i s'écrit comme suit :

$$\Delta g_i = Q_{ia}\Delta\alpha_a + \sum_{j\in I_s} Q_{ij}\Delta\alpha_j + y_i\Delta b \tag{3.63}$$

$\Delta\alpha_a$ est l'incrément du coefficient du nouvel individu.[3]
Dans l'équation précédente on ne tient compte que des éléments appartenant à I_s dans la somme, car si l'élément $j \notin I_s$ et que les individus ne changent pas de sous ensemble $\Delta\alpha_j = 0$. De même en écrivant l'équation (3.61) aux étapes n et $n+1$, on obtient :

$$0 = y_a\Delta\alpha_a + \sum_{j\in I_s} y_j\Delta\alpha_j \tag{3.64}$$

2. Il se peut que la migration de k ait pour conséquence la migration d'autres éléments ; ceci est traité dans l'algorithme par un incrément $\Delta\alpha_a$ nul qui nous ramène automatiquement à l'étape 3.

3. a appartient temporairement aux vecteur support, son incrémentation modifie l'équation de la séparatrice et donc Δb n'est pas nul.

Comme $g_i = 0$ et donc $\Delta g_i = 0$ pour tout individu $i \in I_s$, on peut écrire à partir de (3.63) et (3.64) et en notant $I_s = \{s_1, \ldots, s_{l_S}\}$:

$$\mathcal{Q}. \begin{bmatrix} \Delta b \\ \Delta\alpha_{s_1} \\ \vdots \\ \Delta\alpha_{s_{l_S}} \end{bmatrix} = - \begin{bmatrix} y_c \\ Q_{s_1 a} \\ \vdots \\ Q_{s_{l_S} a} \end{bmatrix} \Delta\alpha_a \tag{3.65}$$

avec

$$\mathcal{Q} = \begin{bmatrix} 0 & y_{s_1} & \cdots & y_{s_{l_S}} \\ y_{s_1} & Q_{s_1 s_1} & \cdots & Q_{s_1 s_{l_S}} \\ \vdots & \vdots & \ddots & \vdots \\ y_{s_{l_S}} & Q_{s_{l_S} s_1} & \cdots & Q_{s_{l_S} s_{l_S}} \end{bmatrix} \tag{3.66}$$

On peut donc exprimer Δb et $\Delta\alpha_j$ en fonction de $\Delta\alpha_a$:

$$\Delta b = \beta \Delta\alpha_a \tag{3.67}$$

$$\Delta\alpha_j = \beta_j \Delta\alpha_a \quad \forall j \in D_n \tag{3.68}$$

où $\beta_j = 0$ si j n'appartient pas à I_s et

$$\begin{bmatrix} \beta \\ \beta_{s_1} \\ \vdots \\ \beta_{s_{l_S}} \end{bmatrix} = -\mathcal{R} \begin{bmatrix} y_c \\ Q_{s_1 a} \\ \vdots \\ Q_{s_{l_S} a} \end{bmatrix}, \quad \mathcal{R} = \mathcal{Q}^{-1}$$

avec $\mathcal{Q}$ définie positive. En remplaçant $\Delta\alpha_j$ et Δb dans (3.63) on obtient :

$$\Delta g_i = \gamma_i \Delta\alpha_c \quad \forall i \in D_{n+1} \tag{3.69}$$

avec

$$\gamma_i = Q_{ia} + \sum_{j \in I_s} Q_{ij}\beta_j + y_i\beta, \quad \forall i \notin I_s$$

et $\gamma_i = 0$ pour i appartenant à I_s.

$\mathcal{R}$ est mis à jour de façon incrémentale à chaque fois que l'ensemble I_s est modifié, ce qui permet de gagner du temps de calcul en n'ayant pas à recalculer l'inverse de $\mathcal{Q}$ à chaque itération.

Si un élément a est ajouté à I_s alors :

$$\mathcal{R} \longleftarrow \begin{bmatrix} & & & 0 \\ & \mathcal{R} & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} + \frac{1}{\gamma_a} \begin{bmatrix} \beta \\ \beta_{s_1} \\ \vdots \\ \beta_{s_{l_S}} \\ 1 \end{bmatrix} . \begin{bmatrix} \beta, & \beta_{s_1}, & \cdots, & \beta_{s_{l_S}}, & 1 \end{bmatrix} \tag{3.70}$$

Lorsque l'on retire un élément l de I_s alors on met à jour $\mathcal{R}$ avec l'équation suivante :

$$\mathcal{R} \longleftarrow \mathcal{R}_{\overline{l+1},\overline{l+1}} - \mathcal{R}^{-1}_{l+1,l+1} \mathcal{R}_{\overline{l+1},l+1} \mathcal{R}_{l+1,\overline{l+1}} \tag{3.71}$$

où :

- $A_{a,b}$ signifie les éléments d'une matrice A localisés à la ligne a colonne b,
- $A_{\overline{a},\overline{b}}$ indique que la ligne a et la colonne b de la matrice A ont été enlevées,
- $A_{\overline{a},b}$ indique la colonne b de la matrice A dont on a enlevé la ligne a,
- $A_{a,\overline{b}}$ indique la ligne a de la matrice A dont on a enlevé la colonne b.

Fonctionnement décrémental Comme dans le cas de l'incrémentation nous allons faire varier la valeur de α_l de l'individu l que l'on souhaite retirer. Cependant contrairement à l'incrémentation on ne part pas forcément de 0 mais on diminue α_l de sa valeur positive ou nulle initiale jusqu'à 0. Une fois sa valeur à 0 on peut retirer complètement l'individu de l'ensemble d'apprentissage.

Au début de l'algorithme $\alpha_l = \alpha_{l_0}$. Il va être mis à jour itérativement par incrémentation successives.

Plusieurs cas vont se présenter. Tout d'abord nous allons regarder si l ne fait pas partie des "remaining vectors" :

I - L'individu l fait partie des "remaining vectors"
On retire l de l'ensemble d'apprentissage.
On peut passer au retrait d'un nouvel individu.

II - Sinon l'individu l fait partie soit des vecteurs support soit des "error vectors".

test préliminaire : Si l est vecteur support on met à jour $\mathcal{Q}$ et $\mathcal{R}$ suivant les équations (3.66) (3.70) en ne considérant plus l comme vecteur support.
Ensuite plusieurs changements peuvent se produire.
Chacun de ces changements va être étudié indépendamment des autres dans l'ordre de leur apparition lors de la diminution de α_l.
Ainsi **on va diminuer** α_l **de** $\Delta\alpha_l$ tel que le premier des évènements suivants se produise [4] :

1 - $\alpha_l = 0 : l$ devient "remaining vector"
On retire l de l'ensemble d'apprentissage.
Mise à jour des $\mathcal{Q}$, $\mathcal{R}$, α_i, b et g_i selon les équations (3.66) (3.70) (3.67) (3.68) (3.69).
On peut passer à l'ajout d'un nouvel individu.

2 - Un des éléments migre entre les ensembles I_s, I_e, I_r. Plusieurs migrations sont possibles :
a - $k \in I_e$ et $g_k = 0 : k$ devient vecteur support,
b - $k \in I_r$ et $g_k = 0 : k$ devient vecteur support,
c - $k \in I_s$ et $\alpha_k = C : k$ devient "error vector",
d - $k \in I_s$ et $\alpha_k = 0 : k$ devient "remaining vector".
Une fois la migration déterminée on remet à jour les différents ensembles I_e, I_s et I_r.
Mise à jour des $\mathcal{Q}$, $\mathcal{R}$, α_i, b et g_i selon les équations (3.66) (3.70) (3.67) (3.68) (3.69).
On recommence l'étape II jusqu'à ce que l'évènement 1 survienne [5].

Conclusion Cet algorithme spécialement développé afin de pouvoir ajouter ou retirer n'importe quel point de l'ensemble d'apprentissage semblait être celui qui correspondait le mieux à nos attentes. Cependant il présente des problèmes de stabilité numérique dans le cas des SVM linéaires. En effet il a été développé en supposant la matrice $\mathcal{Q}$ définie positive et donc inversible. Or dans le cas des SVM linéaires, $\mathcal{Q}$ est seulement semi-définie positive. Ainsi le calcul de la matrice $\mathcal{R}$ n'est pas toujours possible ou exact, ce qui entraine une instabilité. Une astuce, que l'on retrouve souvent dans les algorithmes de SVM, est de résoudre un problème légèrement différent en travaillant non pas sur $\mathcal{Q}$ mais sur $\mathcal{Q} + \epsilon I$. L'ajout d'un ϵ le long de la diagonale permet d'avoir une matrice définie positive dans le cas d'un noyau linéaire et il permet aussi dans le cas d'une matrice définie positive d'augmenter son conditionnement et donc de stabiliser l'algorithme. Cependant il n'existe pas de réglage automatique de ϵ :

4. comme pour l'incrémentation le $\Delta\alpha_l$ nécessaire est calculé de manière analytique selon les équations (3.66) (3.70) (3.67) (3.68) (3.69). On retient le plus petit en valeur absolue et la condition associée.

5. Il se peut que la migration de l ait pour conséquence la migration d'autres éléments ; ceci est traité dans l'algorithme par un incrément $\Delta\alpha_l$ nul qui nous ramène automatiquement à l'étape 3.

- un ϵ trop petit risque de mener à une matrice mal conditionnée et donc à une instabilité de l'algorithme,
- un ϵ trop grand risque de mener à un problème différent de celui des SVM et d'altérer les performances de classification du système.

De plus, pour cet algorithme, il suffit qu'à une itération la matrice $\mathcal{Q}$ soit mal conditionnée pour qu'il ne puisse plus converger vers la solution optimale. C'est pour quoi nous avons étudié l'algorithme suivant.

3.3.2 SimpleSVM

Le second algorithme que nous avons testé est SimpleSVM. Cet algorithme est présenté dans [Vishwanathan *et al.*, 2003].

Comme pour l'algorithme incremental/decremental SVM, nous commencerons par présenter son fonctionnement général avant de détailler la partie mathématique.

Notations. Introduisons les notations suivantes :

- $\mathbf{0}_{\{k\}}$ vecteur de taille k composé uniquement de 0,
- $\mathbf{1}_{\{k\}}$ vecteur de taille k composé uniquement de 1,
- $A_{\{k,l\}}$ la sous matrice composée des coefficients A_{ij} avec $i \in I_k$ et $j \in I_l$,
- $v_{\{k\}}$ le vecteur composé des coefficients v_i tels que $i \in I_k$.
- w^{n+1} la variable w à l'itération $n+1$. Pour simplifier les notations, nous n'utiliserons pas d'exposant pour les variables à l'itération courante.

Principe. L'algorithme part du principe que si on connaît la répartition des différents sous-ensembles I_r, I_e et I_s, les contraintes d'inégalité du problème ne sont plus nécessaires. Si $i \in I_r \Rightarrow \alpha_i = 0$ ou si $i \in I_e \Rightarrow \alpha_i = C$. Seules les valeurs α_i tels que $i \in I_s$ restent à être déterminées. Ces valeurs sont déterminée grâce à la résolution d'un système linéaire.

Cependant en pratique la répartition exacte des sous ensembles n'est pas connue. Chaque itération a pour objectif de diminuer la valeur du critère à minimiser et de faire migrer un des éléments entre les différents sous-ensembles tant que la bonne répartition n'est pas obtenue.

L'étude des conditions de KKT (équation (3.57)) permet de déterminer les migrations entre les sous-ensembles ou si nous sommes à l'optimum du problème.

Algorithme.

Initialisation - On commence par initialiser I_s, I_r, I_e, $\alpha_{\{e\}} = \mathbf{1}_{\{e\}}C$, $\alpha_r = \mathbf{0}_{\{r\}}$ et α_s quelconque. (Mettre tous les exemples dans I_r est une initialisation possible).

I - On résout le système 3.75 [6] :

- si une solution unique $\alpha^*_{\{s\}}$ existe, on pose $d = \alpha^*_{\{s\}} - \alpha_{\{s\}}$
- sinon on doit trouver une direction descente infinie d.

II - à partir de $\alpha_{\{s\}}$ on effectue un pas t le long de la direction d jusqu'à avoir la première des conditions suivantes ($\alpha^{k+1}_{\{s\}} = \alpha_{\{s\}} + td$) :

1 - $\exists i \in I_s$ tel que $\alpha_i^{k+1} = 0$, on transfère le point i de I_s vers I_r
$\Rightarrow I_s \leftarrow I_s \setminus \{i\}$, $I_r \leftarrow I_r \cup \{i\}$, $n \leftarrow n+1$ et on retourne à l'étape **I**.

2 - $\exists i \in I_s$ tel que $\alpha_i^{k+1} = C$, on transfère le point i de I_s vers I_e
$\Rightarrow I_s \leftarrow I_s \setminus \{i\}$, $I_e \leftarrow I_e \cup \{i\}$, $n \leftarrow n+1$ et on retourne à l'étape **I**.

6. Si I_s est vide on passe directement à l'étape **II-3-a**

3 - on arrive à $\alpha_{\{s\}}^{k+1} = \alpha_{\{s\}}^{*}$ [7] :
on calcule $g_{\{e\}}$ et $g_{\{r\}}$ selon l'équation(3.60)
a - si $\min(g_{\{e\}}) > 0$ ou $\min(g_{\{r\}}) < 0$
on transfère le point qui satisfait le moins les conditions de KKT ($\max([|g_{\{e\}}|^t, |g_{\{r\}}|^t])$) de I_r ou I_e vers I_s, $n \leftarrow n+1$ retour à l'étape **I**,
b - sinon on est à l'optimum du problème et l'algorithme se termine.

Calcul de la direction de descente d**.** Pour une configuration I_s, I_e et I_r donnée on remarque que les coefficients $\alpha_{\{r\}}$ valent zéro et qu'ils n'interviennent pas dans la minimisation du critère $W(\alpha,b)$ de l'équation 3.59. En posant $\tilde{\alpha} = \begin{bmatrix}\alpha_{\{s\}}\\ \alpha_{\{e\}}\end{bmatrix}$, $\tilde{y} = \begin{bmatrix}y_{\{s\}}\\ y_{\{e\}}\end{bmatrix}$, $\tilde{\mathbf{1}} = \begin{bmatrix}\mathbf{1}_{\{s\}}\\ \mathbf{1}_{\{e\}}\end{bmatrix}$ et $\tilde{Q} = \begin{bmatrix}Q_{\{s,s\}} & Q_{\{s,e\}}\\ Q_{\{e,s\}} & Q_{\{e,e\}}\end{bmatrix}$ avec $Q_{\{s,e\}} = Q_{\{e,s\}}^t$ nous avons :

$$\begin{aligned} W(\alpha,b) &= \tfrac{1}{2}\alpha^t Q\alpha - \mathbf{1}^t\alpha + by^t\alpha = \tfrac{1}{2}\tilde{\alpha}^t\tilde{Q}\tilde{\alpha} - \tilde{\mathbf{1}}^t\tilde{\alpha} + b\tilde{y}^t\tilde{\alpha} \\ &= \tfrac{1}{2}\alpha_{\{s\}}^t Q_{\{s,s\}}\alpha_{\{s\}} + \alpha_{\{e\}}^t Q_{\{e,s\}}\alpha_{\{s\}} - \mathbf{1}_{\{s\}}^t\alpha_{\{s\}} + b(y_{\{s\}}^t\alpha_{\{s\}} + y_{\{e\}}^t\alpha_{\{e\}}) \\ &\quad + \tfrac{1}{2}\alpha_{\{e\}}^t Q_{\{e,e\}}\alpha_{\{e\}} - \mathbf{1}_{\{e\}}^t\alpha_{\{e\}} \end{aligned} \tag{3.72}$$

Le terme $\frac{1}{2}\alpha_{\{e\}}^t Q_{\{e,e\}}\alpha_{\{e\}} - \mathbf{1}_{\{e\}}^t\alpha_{\{e\}}$ ne dépend ni de $\alpha_{\{s\}}$ ni de b et le problème des SVM devient :

$$\min_{\alpha_{\{s\}},b} W(\alpha_{\{s\}},b) = \frac{1}{2}\alpha_{\{s\}}^t Q_{\{s,s\}}\alpha_{\{s\}} + \alpha_{\{e\}}^t Q_{\{e,s\}}\alpha_{\{s\}} - \mathbf{1}_{\{s\}}^t\alpha_{\{s\}} + b(y_{\{s\}}^t\alpha_{\{s\}} + y_{\{e\}}^t\alpha_{\{e\}}) \tag{3.73}$$

Pour minimiser la fonction $W(\alpha_{\{s\}},b)$, on annule les dérivées partielles par rapport à $\alpha_{\{s\}}$ et b :

$$\begin{aligned} \frac{\partial W(\alpha_{\{s\}},b)}{\partial \alpha_{\{s\}}} &= Q_{\{s,s\}}\alpha_{\{s\}} + CQ_{\{s,e\}}\mathbf{1}_{\{e\}} + by_{\{s\}} - \mathbf{1}_{\{s\}} = 0 \\ \frac{\partial W(\alpha_{\{s\}},b)}{\partial b} &= y_{\{s\}}^t\alpha_{\{s\}} + y_{\{e\}}^t\mathbf{1}_{\{e\}}C = 0 \end{aligned} \tag{3.74}$$

On obtient le système à résoudre suivant :

$$\begin{bmatrix}0 & y_{\{s\}}^t\\ y_{\{s\}} & Q_{\{s,s\}}\end{bmatrix}\begin{bmatrix}b\\ \alpha_{\{s\}}\end{bmatrix} = \begin{bmatrix}-C\mathbf{1}_{\{e\}}^t y_{\{e\}}\\ \mathbf{1}_{\{s\}} - CQ_{\{s,e\}}\mathbf{1}_{\{e\}}\end{bmatrix} \tag{3.75}$$

Cependant, comme les différents sous-ensemble ne sont pas forcément optimaux lors d'une itération de l'algorithme, plusieurs cas peuvent se présenter. Sur la figure 3.7 nous avons représenté les lignes isocritère du problème dans le cas 2D pour différents cas :

- Figure 3.7a le système possède une solution unique $\alpha_{\{s\}}^*$ qui satisfait les contraintes du problème. Dans l'algorithme on aura $d = \alpha_{\{s\}}^* - \alpha_{\{s\}}$ et $t = 1$,
- Figure 3.7b le système possède une solution unique qui ne satisfait pas les contraintes du système. Dans l'algorithme on aura $d = \alpha_{\{s\}}^* - \alpha_{\{s\}}$ et $t = \min\left(-\frac{\alpha_{\{s\}}}{d}, \frac{C-\alpha_{\{s\}}}{d}\right)$
- Figure 3.7c le système ne possède pas de solution unique, on doit trouver une direction de descente. Pour ce faire, on peut utiliser la méthode détaillée dans [Scheinberg, 2006] qui explique comment obtenir une direction de descente dite infinie suivant le niveau de singularité de la matrice $Q_{\{s,s\}}$. Lorsque la direction de descente est trouvée alors $t = \min\left(-\frac{\alpha_{\{s\}}}{d}, \frac{C-\alpha_{\{s\}}}{d}\right)$.

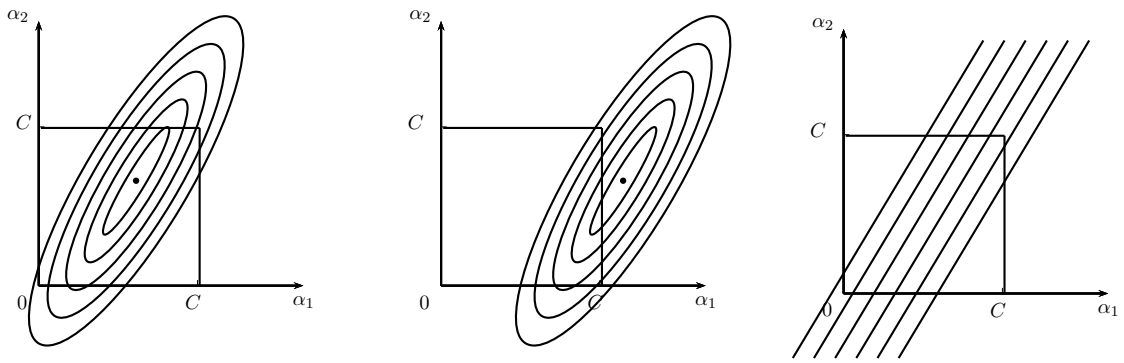


FIGURE 3.7 – Lignes isocritère du problème pour différents types de solutions.

En posant :
- $c_1 = -C\mathbf{1}^t_{\{e\}} y_{\{e\}}$,
- $c_2 = \mathbf{1}_{\{s\}} - CQ_{\{s,e\}}\mathbf{1}_{\{e\}}$,

la solution du système lorsqu'elle existe est donné par :

$$\begin{cases} b^* & = \dfrac{y^t_{\{s\}} Q^{-1}_{\{s,s\}} c_2 - c_1}{y^t_{\{s\}} Q^{-1}_{\{s,s\}} y_{\{s\}}} \\ \alpha^*_{\{s\}} & = Q^{-1}_{\{s,s\}} c_2 - b Q^{-1}_{\{s,s\}} y_{\{s\}} \end{cases} \tag{3.76}$$

A chaque itération l'ensemble I_s n'est modifié que d'un élément. Nous n'avons pas besoin de recalculer entièrement l'inverse de $Q_{\{s,s\}}$. Les équations (3.70) et (3.71) sont toujours applicables.

Algorithme en ligne. Pour ajouter un individu a, on initialise l'algorithme avec la solution calculée précédemment en affectant le nouvel individu à l'ensemble I_r ($\alpha_a = 0$). De même pour retirer un individu l, on initialise l'algorithme avec la solution précédemment calculée dont on a enlevé l de l'ensemble auquel il appartenait.

Conclusion. Contrairement à l'algorithme incrémental/décrémental, SimpleSVM permet de calculer la solution des SVM linéaires. Nous avons vu qu'il était possible lorsque la matrice $Q_{\{s,s\}}$ est non définie positive de trouver une direction de descente permettant à la méthode de converger. Cependant on trouve de nombreuses implémentations qui n'utilisent pas de direction infinie de descente et qui, comme pour les incrémental/décrémental SVM, travaille sur la matrice $Q_{\{s,s\}} + \epsilon I$ afin d'avoir une matrice toujours définie positive. Le choix de ϵ étant toujours soumis aux mêmes problèmes que ceux décrits dans la Section 3.3.1.

7. Cette condition s'obtient seulement lorsque le système 3.75 possède une solution unique et que $t = 1$.

Chapitre 4

Détection des erreurs et performances du système corrigé

Sommaire

Nous avons vu au chapitre précédent que les performances d'un BCI basé sur la reconnaissance de différentes modalités d'un même mouvement sont insuffisantes, ceci d'autant plus que la taille de la population d'apprentissage est petite. Pour obtenir des populations d'apprentissage suffisamment grandes pour ce type de BCI, il est nécessaire d'avoir des sessions d'entraînement longues et donc fastidieuses.

D'autre part, la manière de penser d'un utilisateur évolue au cours du temps. Il essaye d'affiner sa stratégie de commande en fonction des réponses du système : c'est le biofeedback. Pour développer une interface capable de fonctionner avec de faibles sessions d'entraînement et de s'adapter aux évolutions du sujet, le BCI doit pouvoir se corriger et continuer à apprendre au fil de son utilisation. Il est alors indispensable de pouvoir détecter automatiquement si la réponse donnée par le BCI est différente de l'intention de l'utilisateur. Une approche pour y parvenir est d'inclure un retour de l'utilisateur sur le système de décision, comme cela est illustré sur le schéma de la figure 4.1 :

Nous allons étudier dans ce chapitre la détection d'erreur en sortie du système de décision principal (SVM_1) et l'influence de sa prise en compte sur les performances du système. L'intérêt

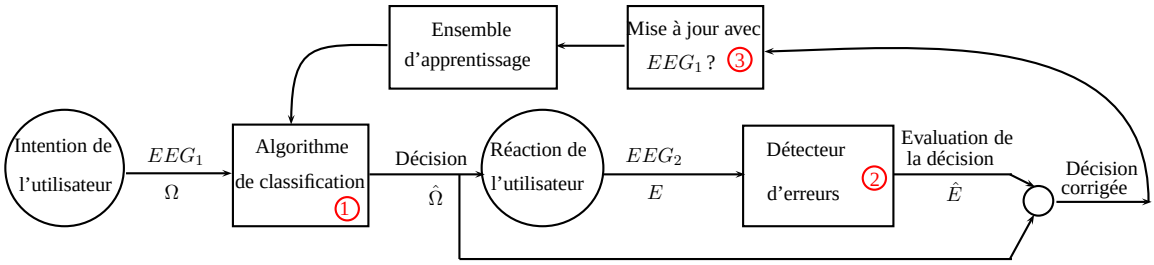


FIGURE 4.1 – Rappel du schéma du BCI adaptatif proposé

du détecteur est triple : corriger si nécessaire la décision avant d'actionner la commande, améliorer les performances du classifieur en enrichissant sa base d'apprentissage avec des exemples pertinents en cours d'utilisation, permettre l'adaptation aux évolutions de l'utilisateur. Dans la mesure où le BCI doit être utilisé par des sujets n'ayant plus aucun contrôle moteur, la détection d'erreur ne peut se faire qu'à partir d'un signal EEG. Dans le système que nous proposons, ce signal (noté EEG_2) émis involontairement par le sujet, est enregistré après affichage de la décision du SVM_1, et est supposé contenir une information pertinente sur la justesse de cette décision. Le schéma est basé sur des travaux [Chavarriaga et Millán, 2010 ; Ferrez et Millán, Graz 2008 ; Dal Seno *et al.*, 2010] qui ont montré qu'un signal EEG spécifique (Potentiel d'erreur, ErrP) apparaît après l'affichage d'une réponse fausse en sortie du BCI.

Dans ce chapitre nous présenterons un bref état de l'art sur ce sujet, puis nous analyserons qualitativement les signaux de potentiel d'erreur enregistrés grâce au protocole présenté Section 1.4.3. Ensuite nous détaillerons deux méthodes que nous avons testées pour détecter les ErrP, une approche détection et une approche classification. Puis nous étudierons l'amélioration théorique de la sortie du BCI corrigé (BCI + détecteur d'erreurs). Enfin nous présenterons les résultats obtenus en considérant les données expérimentales.

4.1 État de l'art sur les potentiels d'erreur

Les travaux de Falkenstein *et al.* [1991] et Gehring *et al.* [1993] ont mis en évidence l'apparition de potentiels d'erreur dans les signaux EEG lors d'erreurs commises par le sujet lui-même. Le protocole utilisé pour mettre en évidence et enregistrer ces signaux est le suivant : on demande au sujet d'appuyer sur deux boutons différents en fonction des informations affichées sur un écran d'ordinateur. Les informations arrivant très rapidement, l'utilisateur fait forcément des erreurs et il s'en rend compte. En moyennant sur plusieurs essais les signaux EEG liés à une erreur de l'utilisateur, les auteurs ont mis en évidence les caractéristiques des potentiels d'erreur. Ces signaux sont localisés spatialement sur la ligne médiane (ligne équidistante des oreilles) des cortex frontal, pariétal et moteur (ce qui correspond aux zones recouvertes par les électrodes F_z FC_z C_z CP_z). Les caractéristiques temporelles des signaux EEG sont l'apparition d'un pic négatif autour de t=100 ms suivi d'un pic positif à t=300 ms (la référence des temps étant le moment où l'utilisateur appuie sur la mauvaise touche).

En 2002, Blankertz *et al.* [2002] proposent une méthode pour détecter sur un seul essai la présence de ces potentiels d'erreur. Le protocole expérimental est le même que celui décrit précédemment. Pour détecter les potentiels d'erreur, ils utilisent un classifieur bayésien sur les échantillons des signaux filtrés et décimés. De plus ils quantifient l'amélioration du BCI lors de l'intégration de la détection des potentiels d'erreur en terme de taux de transfert (BpT).

Ce n'est que plus récemment que l'équipe de Ferrez et Millán [2005], a mis en évidence des

potentiels d'erreur liés aux erreurs du BCI et non de l'utilisateur. Actuellement deux équipes ont travaillé sur le sujet, celle de Ferrez-Millàn (EPFL) et celle de Dal Seno (Politecnico di Milano). Ils ont montré que ces potentiels d'erreur pouvaient être détectés sur un seul essai. Ce deuxième type de potentiel d'erreur (ErrP) est aussi caractérisé par une distribution spatiale sur la ligne médiane des cortex frontal, pariétal et moteur. Les signaux EEG sont caractérisés temporellement par l'apparition d'un pic de potentiel positif localisé à t=300 ms après l'erreur du BCI, suivi d'un pic négatif localisé à t=550 ms. Après 650 ms les signaux EEG relatifs à une erreur ou non redeviennent identiques. Selon les travaux, différents protocoles ont été utilisés avec différents types de BCI et des conditions plus ou moins proches de l'application réelle :

- pour enregistrer les signaux, [Ferrez et Millán, 2005 ; Buttfield *et al.*, 2006] demandent à l'utilisateur de piloter un robot à l'aide de 2 touches (droite et gauche). Le robot est programmé pour aller dans la mauvaise direction dans 20% des cas.
- dans les articles [Ferrez et Millán, 2008, Graz 2008 ; Chavarriaga et Millán, 2010], les BCI sont basés sur des mouvements imaginaires et les potentiels d'erreur évoqués par l'affichage d'une pseudo-classification.
- dans [Dal Seno *et al.*, 2010] la détection d'ErrP se fait lors de l'utilisation d'un "P300 speller".

Tous ces travaux montrent l'amélioration apportée au BCI grâce à la détection de potentiels d'erreur. L'apport de ma thèse au niveau de l'utilisation des potentiels d'erreur se trouve dans les points suivants :

- c'est la première fois que les potentiels d'erreur sont utilisés dans un système destiné à faire des distinctions complexes entre les tâches : il s'agit ici non pas de distinguer un mouvement du bras droit ou gauche, mais différentes modalité d'un même mouvement (flexion d'un même membre lente/rapide).
- nous caractérisons théoriquement les performances du BCI corrigé par le détecteur d'erreur.
- nous comparons deux approches pour la reconnaissance des potentiels d'erreur : une approche détection et une approche classification.
- nous avons implémenté le système complet en ligne.

4.2 Analyse qualitative des signaux

Les signaux analysés dans ce chapitre ont été enregistrés selon le protocole décrit en section 1.4.3, après l'affichage d'un pseudo-feedback correspondant à une décision aléatoire *juste/faux* donnée par le module de décision SVM_1. On dispose de 2 classes de signaux EEG_2 obtenues l'une après l'affichage de *juste*, l'autre après l'affichage de *faux*.

- **Observation des différents essais sur le canal C_z**. Les Figures 4.2 et 4.4 montrent, chez les sujets 3 et 5, des signaux EEG_2 enregistrés sur l'électrode C_z au cours de 6 essais différents, dont 3 dans le cas où le BCI affiche la bonne réponse c'est à dire sans ErrP (à gauche) et 3 dans le cas où le BCI affiche la mauvaise réponse (avec ErrP, à droite) [1]. On constate que dans le cas d'une réponse fausse sur certains essais (sous-figures b et f du sujet 3, d et f du sujet 5) un pic positif est présent sur l'intervalle 250-350 ms. Sur les autres essais ce pic est peu ou pas visible. Dans le cas d'une réponse juste, il est rare d'observer un pic positif dans cet intervalle. En résumé la simple observation visuelle du signal ne permet pas de détecter aisément la présence ou l'absence d'un potentiel d'erreur supposé présent ou non dans le signal.

1. Nous avons appliqué un filtre coupe bande autour de 50 Hz pour enlever les parasites induits par les appareils électriques ainsi qu'un filtre passe-bas 0-95 Hz pour éliminer les harmoniques du 50 Hz, sachant que la bande passante de l'amplificateur est limitée à 100 Hz.

– **Observation de la moyenne des essais par classe sur différents canaux**. Pour faire ressortir les caractéristiques des signaux possédant des ErrP nous avons fait la moyenne des signaux EEG des essais de chaque classe. Les Figures 4.3 et 4.5 montrent, pour les sujet 3 et 5, les moyennes des signaux enregistrés sur les électrodes F_z FC_z C_z et CP_z entre 150 ms et 650 ms après avoir donné la réponse à l'utilisateur. On constate l'apparition d'un pic de potentiel positif entre 250 ms et 350 ms suivi d'un potentiel négatif entre 500 ms et 600 ms et une amplitude moyenne des variations de potentiel plus importantes lorsque la réponse donnée à l'utilisateur est fausse (à droite) que dans le cas où la réponse est juste (à gauche).
  Les moyennes mettent en évidence les formes caractéristiques de la classe avec ErrP, mais avec une forte dispersion. Les moyennes des signaux de la classe sans ErrP tendent à avoir une amplitude quasi nulle. De plus on observe une forte corrélation entre les moyennes des différentes voies.

En conclusion de ces observations on peut dire que :
– elles sont en accord avec la littérature [Ferrez et Millán, 2008 ; Dal Seno *et al.*, 2010] et montrent que le protocole utilisé permet de produire des potentiels d'erreur.
– la classe du signal (avec/sans ErrP) n'est pas facilement identifiable par la simple observation du signal.
– la moyenne des signaux est caractéristique de chaque classe mais la dispersion au sein d'une classe est très forte.
– les différentes voies d'enregistrement sont fortement corrélées.

Aucune méthode pour déterminer la classe des signaux à partir d'un seul essai ne semble *a priori* s'imposer. C'est pourquoi nous allons explorer deux approches différentes.

4.3 Détection des potentiels évoqués

Deux approches ont été utilisées pour évaluer l'exactitude de la décision du BCI :
– une approche détection, que l'on notera $\text{détecteur}_\text{d}$,
– une approche classification, que l'on notera $\text{détecteur}_\text{c}$, inspirée de la littérature.

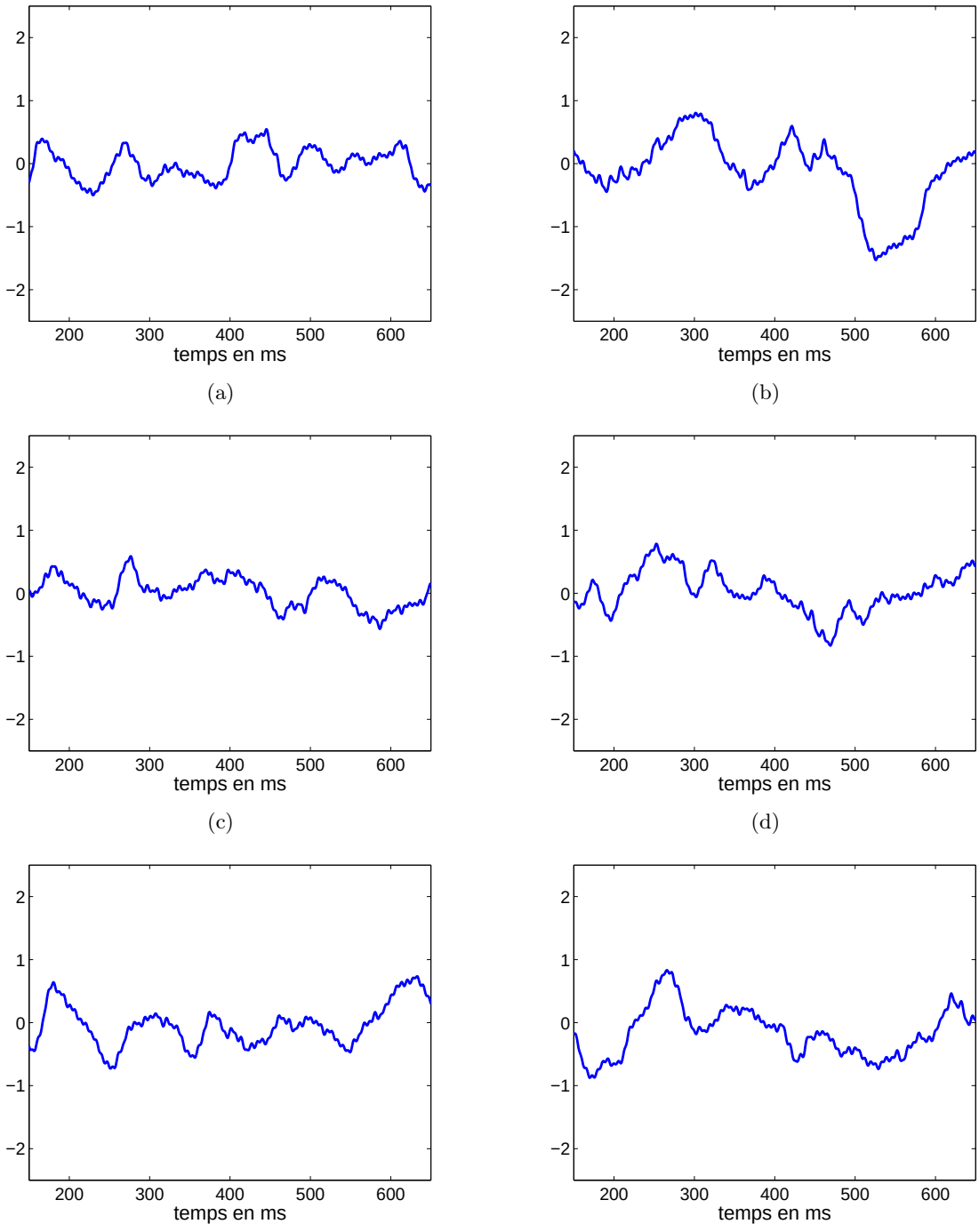
Pour ces deux approches, nous utilisons le signal contenu dans une fenêtre de 150 ms à 650 ms après l'affichage de la réponse du SVM_1(les potentiels d'erreur apparaissent dans cette fenêtre temporelle) pour décider si un ErrP est présent. Ces deux approches sont supervisées, s'est-à-dire que l'on dispose d'une population d'apprentissage.

On notera $\mathcal{X}$ l'ensemble de tous les signaux d'apprentissage et $\mathcal{X}_e$ l'ensemble des signaux de la classe e, e pouvant prendre les valeurs :
– *faux* pour les signaux possédant un potentiel d'erreur (décision fausse du SVM_1),
– *juste* pour les signaux ne possédant pas de potentiel d'erreur.

4.3.1 Approche détection ($\text{détecteur}_\text{d}$)

4.3.1.1 Indice de détection proposé

L'indice de détection proposé ici est basé sur les observations précédentes : la moyenne des EEG de la classe *faux* met en évidence un potentiel moyen caractéristique, absent de la moyenne des EEG de la classe *juste*. La détection d'un potentiel d'erreur dans un EEG va donc être basée sur la corrélation de ce signal avec le potentiel moyen de la classe *faux*. Pour un signal donné x



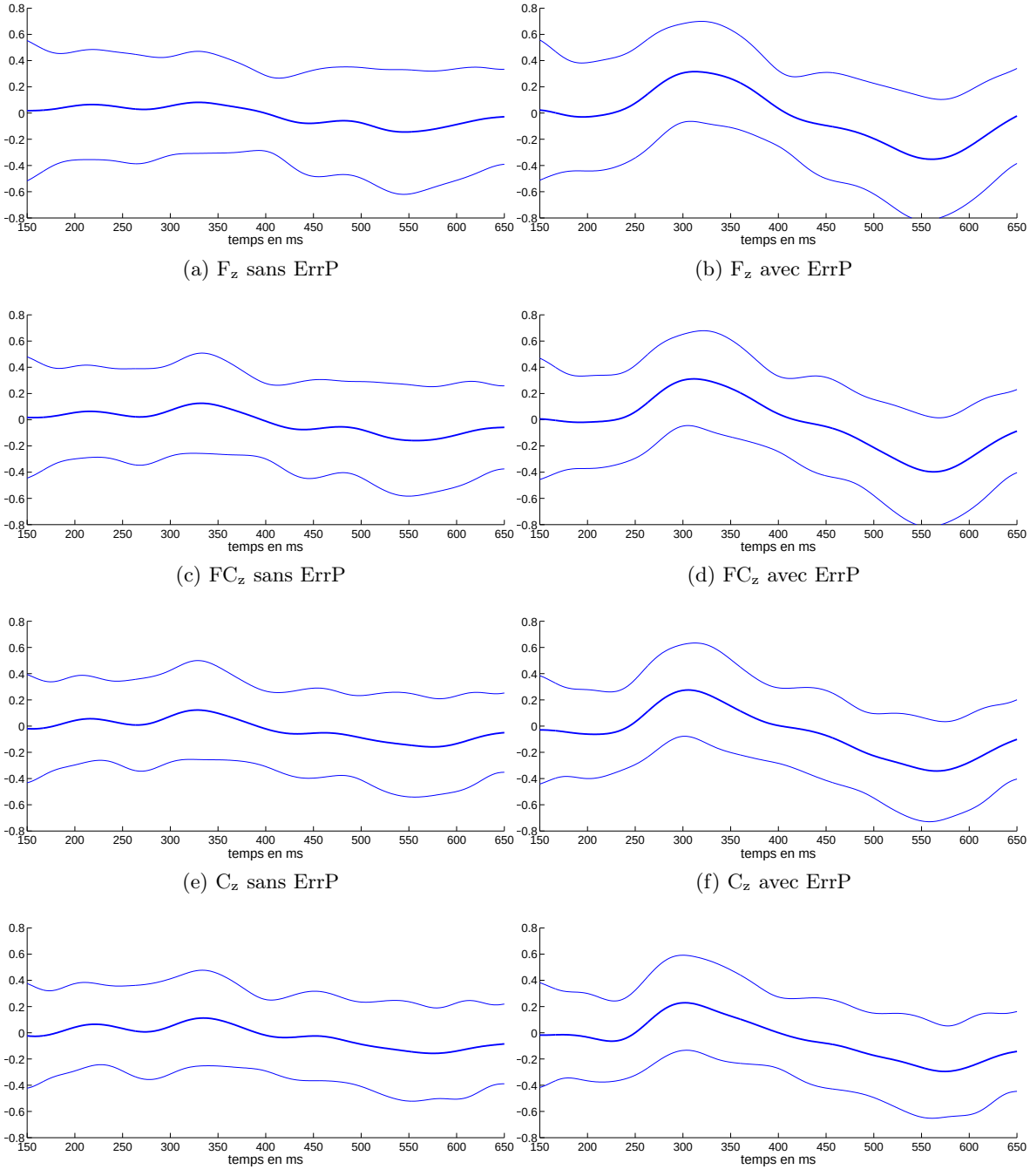
(a) (b)

(c) (d)

(e) (f)

Essais sans ErrP Essais avec ErrP

FIGURE 4.2 – Exemples de signaux EEG du sujet 3 enregistrés sur l'électrode C_z entre 150 ms et 650 ms après affichage de la réponse du BCI. A gauche : les signaux obtenus lorsque la réponse affichée est juste (sans ErrP). A droite : les signaux obtenus lorsque la réponse affichée est fausse (avec ErrP).



(a) F_z sans ErrP

(b) F_z avec ErrP

(c) FC_z sans ErrP

(d) FC_z avec ErrP

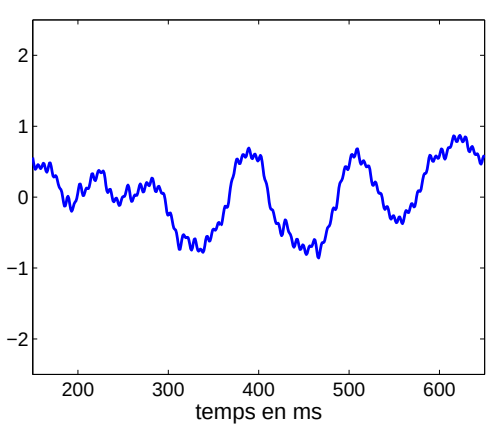
(e) C_z sans ErrP

(f) C_z avec ErrP

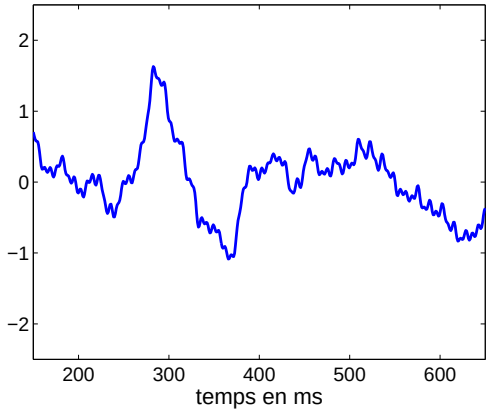
(g) CP_z sans ErrP

(h) CP_z avec ErrP

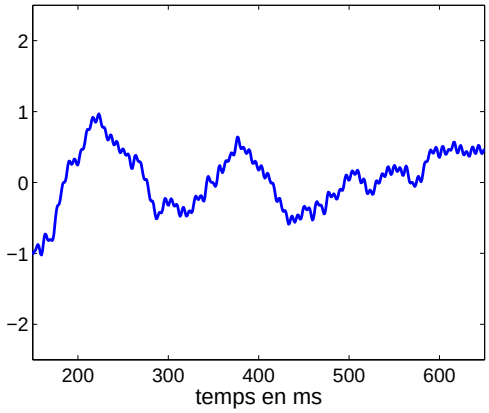
FIGURE 4.3 – En trait épais : moyennes des signaux EEG du sujet 3 enregistrés sur les électrodes F_z FC_z C_z et CP_z entre 150 ms et 650 ms après affichage de la réponse du BCI. Autour des moyennes sont représentés en traits fins plus ou moins un écart-type. A gauche : les signaux obtenus lorsque la réponse affichée est juste (sans ErrP). A droite : les signaux obtenus lorsque la réponse affichée est fausse (avec ErrP).



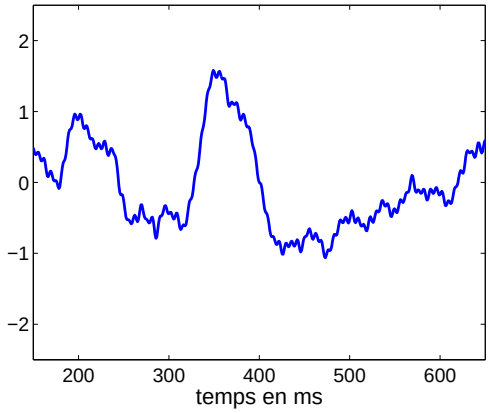
(a)



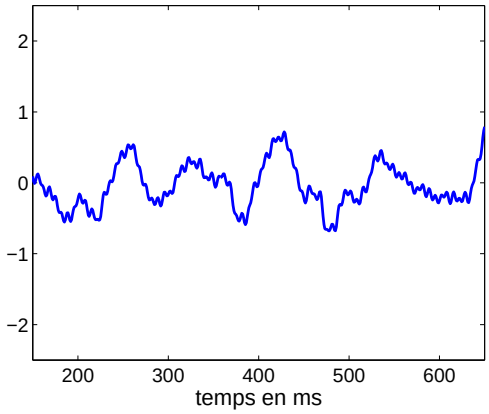
(b)



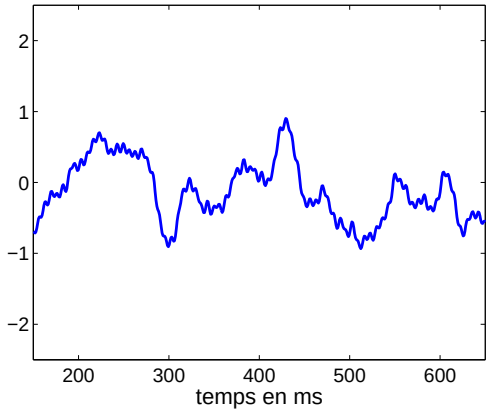
(c)



(d)

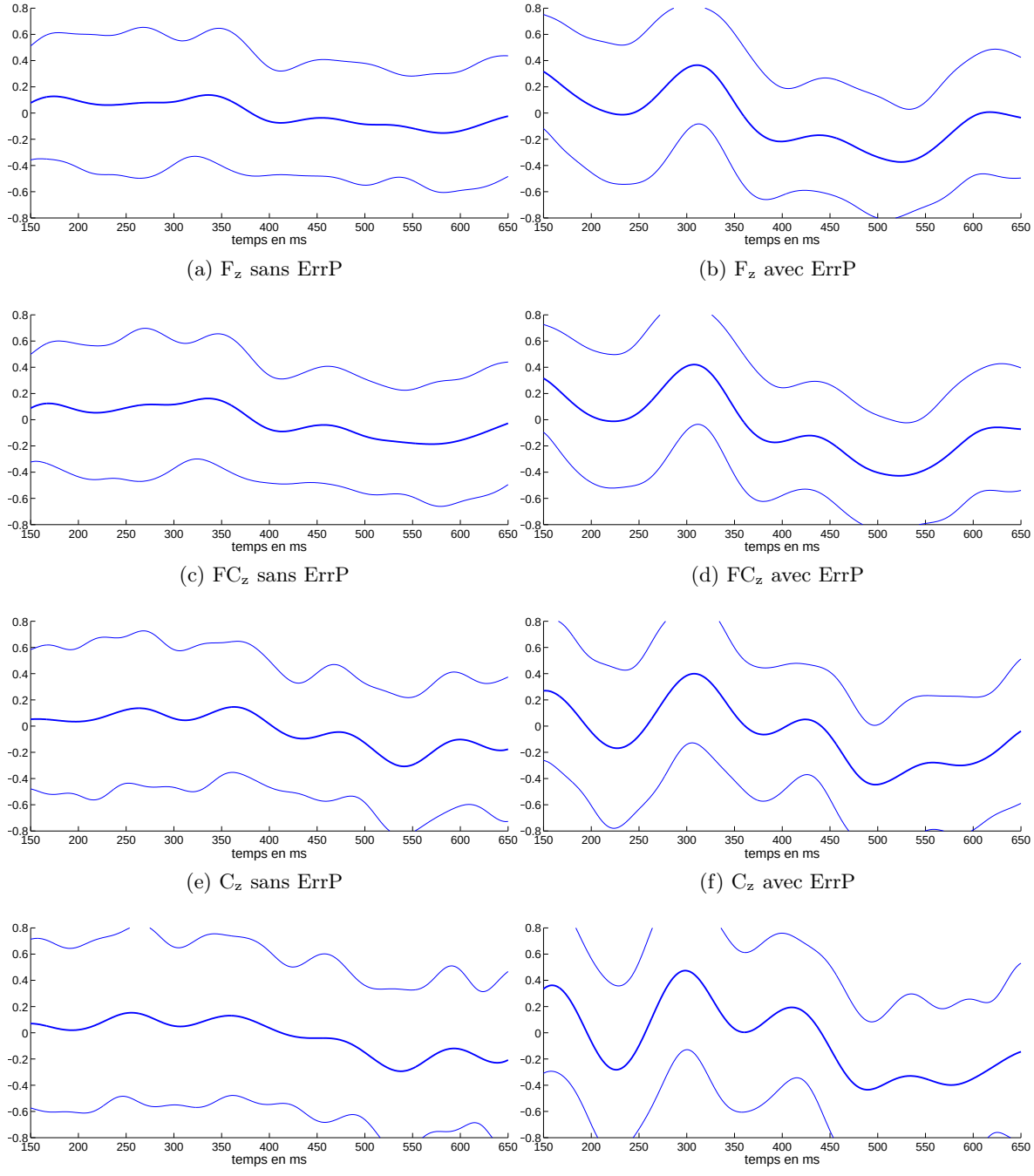


(e) Essais sans ErrP



(f)
Essais avec ErrP

FIGURE 4.4 – Exemples de signaux EEG du sujet 5 enregistrés sur l'électrode C_z entre 150 ms et 650 ms après affichage de la réponse du BCI. A gauche : les signaux obtenus lorsque la réponse affichée est juste (sans ErrP). A droite : les signaux obtenus lorsque la réponse affichée est fausse (avec ErrP).



(a) F_z sans ErrP

(b) F_z avec ErrP

(c) FC_z sans ErrP

(d) FC_z avec ErrP

(e) C_z sans ErrP

(f) C_z avec ErrP

(g) CP_z sans ErrP

(h) CP_z avec ErrP

FIGURE 4.5 – En trait épais : moyennes des signaux EEG du sujet 5 enregistrés sur les électrodes F_z FC_z C_z et CP_z entre 150 ms et 650 ms après affichage de la réponse du BCI. Autour des moyennes sont représentés en traits fins plus ou moins un écart-type. A gauche : les signaux obtenus lorsque la réponse affichée est juste (sans ErrP). A droite : les signaux obtenus lorsque la réponse affichée est fausse (avec ErrP).

(signal mono-voies après filtrage spatial), l'index de détection $i(x)$ est défini comme le maximum de l'intercorrélation $\phi_x(\tau)$ entre le signal et la moyenne des signaux $\overline{x_{faux}}$ de $\mathcal{X}_{faux}$:

$$i(x) = \max\{\phi_x(\tau), \tau \in [-50ms, 50ms]\} \tag{4.1}$$

Si les signaux étaient parfaitement synchronisés on prendrait la valeur de l'intercorrélation en $\tau = 0$. Cependant le temps de réaction de l'utilisateur pouvant varier de quelques millisecondes entre les essais, le maximum est recherché dans une fenêtre de 100 ms autour de 0 ms.

Une forte corrélation traduira la présence d'un potentiel d'erreur (classe *faux*).

En supposant que les densité de probabilité des indices conditionnellement aux classes *juste* et *faux* suivent une loi connue *a priori*, on peut se placer dans le cadre de la détection bayésienne.

4.3.1.2 La détection bayésienne

On note :
- I la variable aléatoire représentant l'indice de détection dont la réalisation correspondante est $i(x)$.
- $P_E(juste)$ la probabilité *a priori* de la classe *juste* (probabilité de bien classer du SVM_1).
- $P_E(faux)$ la probabilité *a priori* de la classe *faux* (probabilité que le SVM_1 se trompe).

On suppose ces probabilités connues.

On définit les densités de probabilité de I conditionnellement à la justesse de la réponse du SVM_1 :
- $P_{I|E}(i(x),juste)$: la densité de probabilité de I conditionnellement à la classe *juste*
- $P_{I|E}(i(x),faux)$: la densité de probabilité de I conditionnellement à la classe *faux*

Enfin on associe un coût $\lambda_{\hat{e}|e}$ associé à la décision $\hat{e}$ du détecteur sachant la réaction e de l'utilisateur ; les bonnes détections du détecteur $\lambda(\hat{e} = e)$ sont à coût nul.

La règle de décision bayésienne consiste à minimiser le risque total (ou coût moyen) en prenant pour tout $i(x)$ la décision qui minimise le risque conditionnel à I, soit :
- décider faux si :

$$\frac{P_{I|E}(i(x),faux)}{P_{I|E}(i(x),juste)} > \frac{\lambda_{faux|juste}.P_E(juste)}{\lambda_{juste|faux}.P_E(faux)}$$

- décider vrai sinon.

Lorsque les coûts $\lambda_{\hat{e}|e}|_{\hat{e}\neq e}$ sont égaux : on prend la décision qui maximise la probabilité *a posteriori* :
- décider $faux$ si $P_{E|I}(faux,i(x)) > P_{E|I}(juste,i(x))$, c'est-à-dire si $P_E(faux).P_{I|E}(i(x),faux) > P_E(juste).P_{I|E}(i(x),juste)$,
- juste sinon.

Si de plus les probabilités *a priori* sont égales on retrouve le maximum de vraisemblance.

La figure 4.6 illustre le problème :
- $s_0 = i(x_0)$ seuil de décision, obtenu en résolvant :

$$P_E(faux).P_{I|E}(i(x_0),faux) = P_E(juste).P_{I|E}(i(x_0),juste) \tag{4.2}$$

- $\hat{e} = \begin{cases} juste & \text{si } i(x) < s_0 \\ faux & \text{sinon} \end{cases}$
- Les probabilités d'erreur à l'intérieur d'une classe :
  - $P_{I|E}(i(x \in \mathcal{X}_{faux}) < s_0, faux)$ densité de probabilité d'erreur à l'intérieur de la classe *faux*

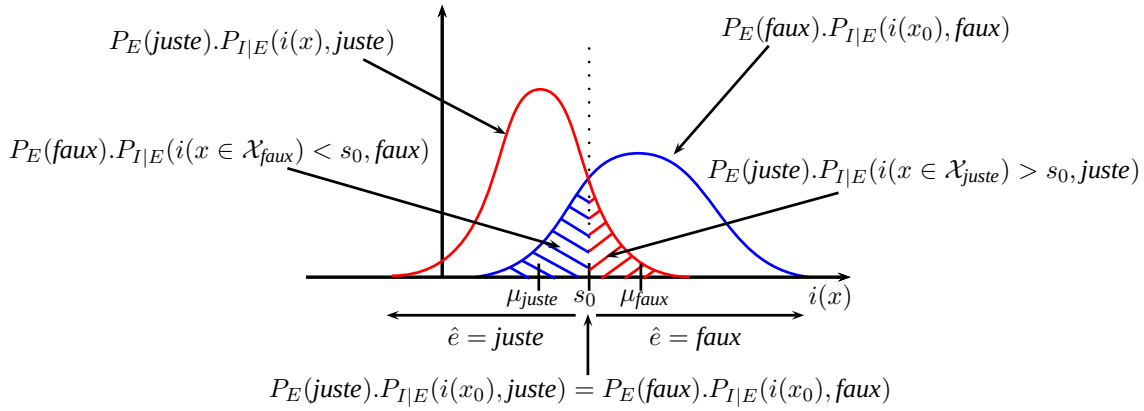


FIGURE 4.6 – Illustration de la détection dans le cas gaussien. s_0 correspond au seuil de détection. Les deux surfaces hachurées représentent la probabilité d'erreur du détecteur.

- $P_{I|E}(i(x \in \mathcal{X}_{juste}) > s_0, juste)$ densité de probabilité d'erreur à l'intérieur de la classe *juste*

En pratique on ne connaît pas les densités $P_{I|E}(i(x),e)$ et on doit les estimer. On les suppose généralement gaussiennes et on estime leurs paramètres :

$$\hat{P}_{I|E}(i(x),e) = \mathcal{N}(\hat{\mu}_e,\hat{\sigma}_e) \tag{4.3}$$

avec :
- $\hat{\mu}_e = \frac{1}{n_e}\sum_{x\in\mathcal{X}_e} i(x)$
- $\hat{\sigma}_e = \frac{1}{n_e}\sum_{x\in\mathcal{X}_e} (i(x) - \mu_e)^2$

On suppose $\hat{\mu}_{juste} < \hat{\mu}_{faux}$, si ce n'est pas le cas cela signifie que l'index choisi n'est pas adapté à la détection des potentiels d'erreur.

Lorsque les $\hat{\sigma}_e$, sont différents l'équation (4.2) possède deux solutions, cependant la solution qui nous intéresse est telle que $\hat{\mu}_{juste} < s_0 < \hat{\mu}_{faux}$.

Dans ce cas les probabilités d'erreur à l'intérieur d'une classe peuvent être estimées analytiquement :
- $\hat{P}_{I|E}(i(x \in \mathcal{X}_{faux}) < s_0, faux)$:

$$\frac{1}{2}\left(1 - \mathrm{erf}\left(\frac{|s_0 - \hat{\mu}_{faux}|}{\sqrt{2}\hat{\sigma}_{faux}}\right)\right) \tag{4.4}$$

- $\hat{P}_{I|E}(i(x \in \mathcal{X}_{juste}) > s_0, juste)$:

$$\frac{1}{2}\left(1 - \mathrm{erf}\left(\frac{|s_0 - \hat{\mu}_{juste}|}{\sqrt{2}\hat{\sigma}_{juste}}\right)\right) \tag{4.5}$$

Ici nous avons supposé les coûts $\lambda_{\hat{e}|e}|_{\hat{e}\neq e}$ égaux. Il est en réalité difficile de quantifier la valeur du coût d'une erreur à appliquer à chaque classe :
- estimer la réponse *juste* alors qu'elle est fausse commandera la prothèse à l'opposé de l'intention du patient et introduira des erreurs dans la population d'apprentissage du SVM_1 lors de sa mise à jour en ligne.
- estimer la réponse *faux* alors qu'elle est juste fera répéter le sujet inutilement.

Aucune de ces deux erreurs n'est souhaitable et il est difficile de dire laquelle a plus d'importance que l'autre. C'est pourquoi nous avons décidé d'utiliser une stratégie à deux seuils afin de limiter le taux d'erreur au sein de chaque classe.

4.3.1.3 Méthode de détection à deux seuils

L'utilisation de deux seuils permet de fixer les taux d'erreur dans une classe en les estimant sur une population d'apprentissage. En utilisant les équations (4.4) et (4.5), on peut définir les seuils haut (s_{faux}) et bas (s_{juste}) de la façon suivante :

- s_{faux} tel que $\frac{1}{2}\left(1 - \text{erf}\left(\frac{|s_{faux} - \hat{\mu}_{juste}|}{\sqrt{2}\hat{\sigma}_{juste}}\right)\right) < C_{juste}$
- s_{juste} tel que $\frac{1}{2}\left(1 - \text{erf}\left(\frac{|s_{juste} - \hat{\mu}_{faux}|}{\sqrt{2}\hat{\sigma}_{faux}}\right)\right) < C_{faux}$

avec C_{juste} et C_{faux} le pourcentage d'erreur à ne pas dépasser au sein des deux classes.

L'index $i(x)$ est ensuite comparé à ces deux seuils s_{faux}, s_{juste} (cf. figure 4.7) :

- Si $i(x) \geq s_{faux}$, la décision du BCI est estimée comme *faux*.
- Si $i(x) \leq s_{juste}$, la décision du BCI est estimée comme *juste*.

Entre les deux seuils, le détecteur ne donne pas de réponse.

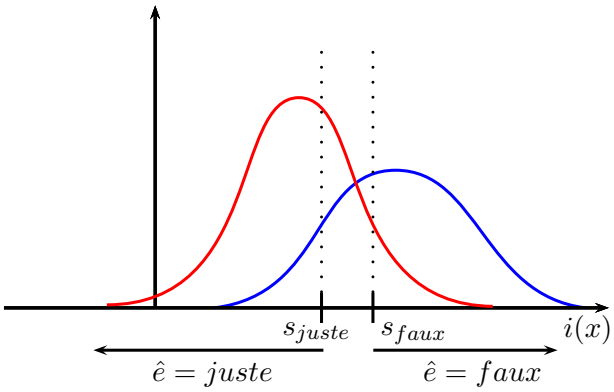


FIGURE 4.7 – Méthode à deux seuils

En réalisant sur les données expérimentales l'histogramme normalisé des index de chaque classe, nous avons obtenu les approximations des densités de probabilité conditionnelles. Nous les avons représentées sur la Figure 4.8 pour différentes voies. On constate que ces densités ne sont pas gaussiennes et donc l'approximation pour choisir les seuils n'est pas valable.

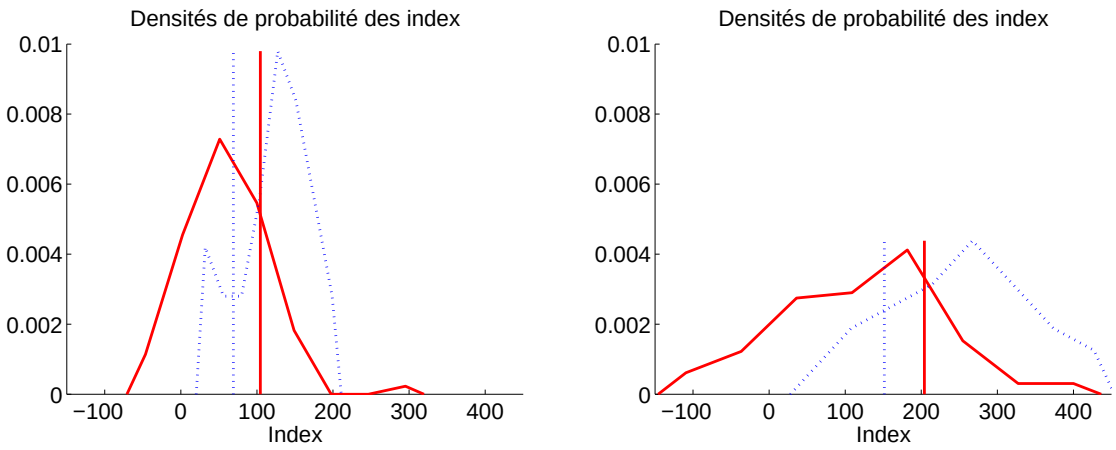
Nous proposons de choisir les seuils à partir d'une estimation empirique de la probabilité d'erreur. Pour un seuil donné, celle-ci est estimée par le taux de mal classés au sein de chaque classe.

- $\hat{P}_{I|E}(i(x \in \mathcal{X}_{juste}) > s_0, juste) = \frac{\text{Card}(x \in \mathcal{X}_{juste} | i(x) > s_{faux})}{\text{Card}(x \in \mathcal{X}_{juste})}$
- $\hat{P}_{I|E}(i(x \in \mathcal{X}_{faux}) < s_0, faux) = \frac{\text{Card}(x \in \mathcal{X}_{faux} | i(x) < s_{juste})}{\text{Card}(x \in \mathcal{X}_{faux})}$

Plusieurs valeurs de seuils sont testées et les seuils s_{juste} et s_{faux} sont fixés de manière à obtenir des taux d'erreur respectivement inférieurs à C_{juste} et C_{faux}

Sur la Figure 4.8 les seuils sont placés de manière à avoir une probabilité d'erreur à l'intérieur de chaque classe inférieure à $C_{juste} = C_{faux} = 20\%$.

En regardant les approximations des densités de probabilité des index des classes, on voit que les résultats de la détection dépendent fortement du choix des voies et que l'utilisation des deux seuils permet de réduire les taux de fausses alarmes.



(a) FC_z (b) CP_z

FIGURE 4.8 – Densité de probabilité des index des deux classes et placement des seuils pour deux canaux différents (FC_z et CP_z)

Pour le sujet représenté Figure 4.8, on note que la zone de mélange est plus faible pour l'électrode FC_z que pour l'électrode CP_z. La voie FC_z donnera donc de meilleurs résultats que la voie CP_z, d'une part en terme de pourcentage de bien classés et d'autre part en terme de taux de réponses.

4.3.2 Approche classification (détecteur$_c$)

La deuxième méthode proposée est l'approche classification. Dans cette approche nous ne supposons plus que nous avons une classe avec de l'information et une classe avec du bruit, contrairement à l'approche détection. Les deux classes, *juste* et *faux*, sont supposées contenir des informations différentes.

L'approche classification est composée de deux étapes principales :
- extraction des caractéristiques des signaux,
- choix du classifieur.

4.3.2.1 Extraction de caractéristiques

Les signaux EEG sont composés d'un grand nombre de données :
- plusieurs électrodes,
- nombre d'échantillons important sur chaque voies.

Pour obtenir de bonnes performances, il est nécessaire de se ramener à un nombre de données plus faible pour décrire les signaux.

Nous allons considérer différents types de descripteurs répartis en trois principales familles : les descripteurs temporels, fréquentiels et temps-fréquence et retiendrons ceux qui conduisent aux meilleurs performances.

Descripteurs temporels. Les descripteurs extraits du domaine temporel sont les plus utilisés pour caractériser les signaux de type potentiel évoqué, comme les P300 et les ErrP. Ils correspondent le plus souvent à l'ensemble des échantillons du signal [Rakotomamonjy *et al.*, 2005 ;

Dal Seno *et al.*, 2010] ou à des paramètres basés sur la variance du signal ou de sa dérivée, tels que les paramètres de Hjorth souvent utilisés pour décrire l'activité cérébrale [Obermeier *et al.*, 2001]. Nous utiliserons dans cette étude les échantillons du signal EEG_2, ce qui revient à décrire la forme du potentiel évoqué. Les ErrP étant caractérisés par des variations lentes, les signaux enregistrés à une fréquence de 1024 Hz sont préalablement filtrés par un filtre passe-bas de 0-10 Hz (Chebychev d'ordre 6). Ils peuvent alors être décimés [2] d'un facteur 32, de manière à réduire le plus possible la dimension de l'espace de représentation et diminuer ainsi le risque de sur-apprentissage.

Ces descripteurs sont les plus utilisés pour la classification des P300 et des ErrP.

Descripteurs fréquentiels. Les signaux EEG sont composés d'ensembles d'oscillations spécifiques, d'états de repos ou d'activité cérébrale (voir les rythmes décrits section 2.1). Nous avons considéré les descripteurs suivants :

- Energies par bande : cette technique consiste à prendre les marginales du module de la transformée de Fourier au carré dans certaines bandes de fréquence. Nous avons retenu un découpage uniforme de l'axe fréquentiel en 10 bandes.
- Marginales de la DWT : Elles caractérisent l'activité du signal dans des bandes de fréquence correspondant à un découpage dyadique de l'axe fréquentiel et donc adapté au contenu basse fréquence des signaux.

Descripteurs temps-échelle. La transformée discrète en ondelettes correspond à des représentations temporelles du signal à différentes échelles, donc dans différentes bandes de fréquence. Nous avons choisi comme descripteurs les coefficients des différents niveaux de détails et de l'approximation de la DWT.

Analyse en composantes principales des signaux pour différents types de descripteurs. Pour visualiser les classes dans l'espace des descripteurs, nous avons fait une analyse en composantes principales (PCA) des signaux décrits par plusieurs types de descripteurs. Les résultats du PCA sont présentés Figure 4.9.

La comparaison de l'inertie des axes des différents PCA est difficile, car pour chaque type de descripteur, la dimension de l'espace de représentation change. Ainsi, un PCA avec un axe principal possédant 70% de l'information dans un espace à 2 dimensions n'est pas forcément plus représentatif qu'un axe principal possédant moins d'information mais dans un espace de dimension plus grande. De plus, quels que soient les descripteurs utilisés, la séparabilité des classes dans l'espace des deux composantes principales du PCA est similaire. Il n'est donc pas possible de déterminer à partir du PCA les descripteurs les plus efficaces pour la classification. Avec une étude *a posteriori*, nous nous sommes aperçus que les échantillons des signaux filtrés décimés donnaient les meilleurs résultats de classification.

4.3.3 Choix du classifieur

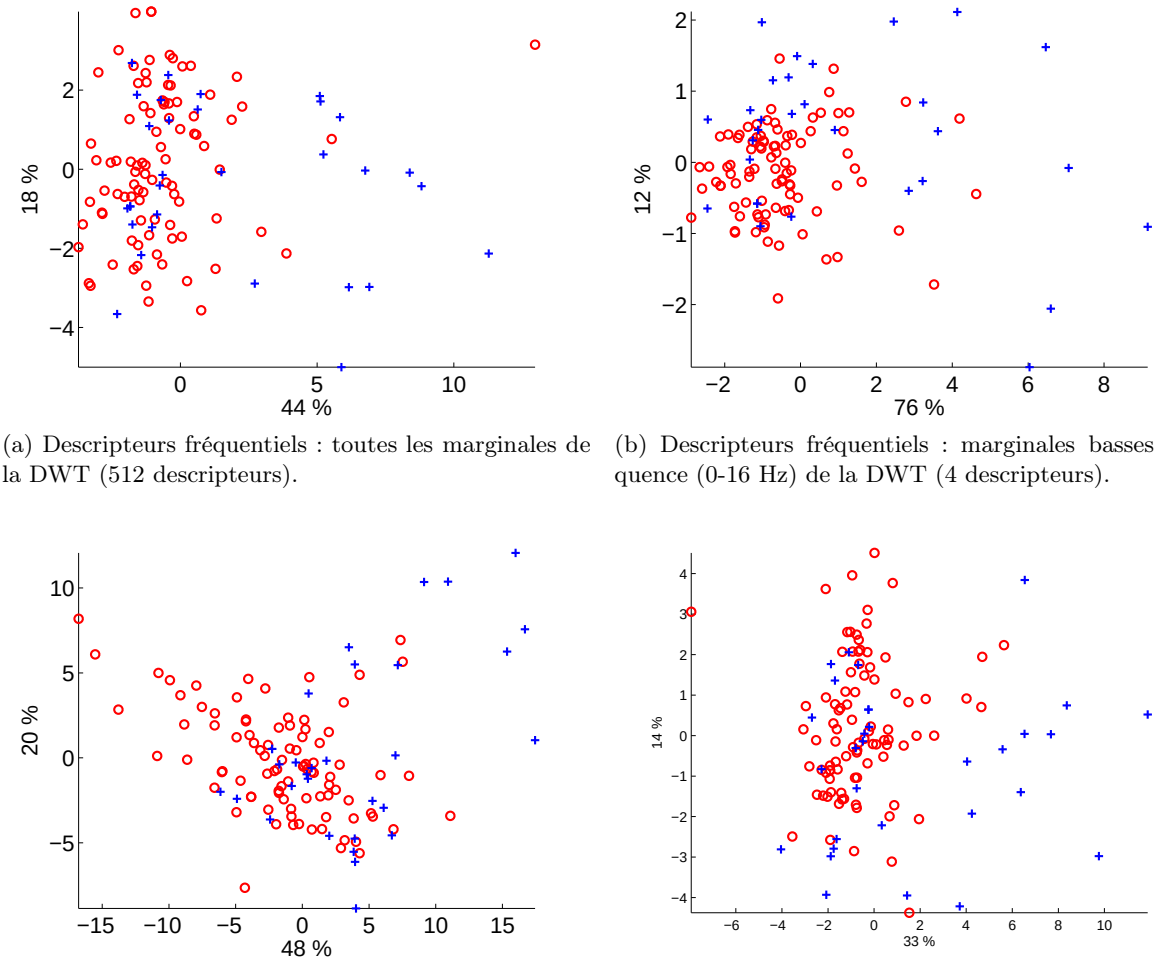
Dans notre approche, nous avons décidé d'utiliser un classifieur SVM linéaire. La méthode des SVM est connue pour être robuste même dans des conditions difficiles (dimension de l'espace

2. Pour un signal discret s, le signal s_d décimé d'un facteur N_d correspondant est défini par :

$$s_d[k] = s[N_d k] \quad \forall k \in \mathbb{N}$$

de représentation grande par rapport à la taille de l'ensemble d'apprentissage). Or selon les descripteurs utilisés, nous pouvons avoir jusqu'à 96 dimensions pour l'espace de représentation et seulement 120 essais. Dans ces conditions, il est difficile de déterminer les paramètres des classifieurs de types bayésiens, qui demandent l'estimation de la matrice de covariance et son inversion.

Finalement, pour cette méthode, les descripteurs retenus seront les échantillons filtrés et décimés des signaux des différentes voies et nous utiliserons un classifieur SVM linéaire.



(a) Descripteurs fréquentiels : toutes les marginales de la DWT (512 descripteurs).

(b) Descripteurs fréquentiels : marginales basses fréquence (0-16 Hz) de la DWT (4 descripteurs).

(c) Descripteurs temporels : échantillons des signaux filtrés et décimés (32 descripteurs).

(d) Descripteurs temps-fréquence : coefficients des niveaux de détail basse fréquence et de l'approximation de la DWT (0-16 Hz) (16 descripteurs).

FIGURE 4.9 – Représentation dans le premier plan de l'ACP des deux classes (*juste* o et *faux* +) des signaux représentés par différents descripteurs. Les pourcentages indiquent la part d'inertie retenue par chaque axe.

4.4 Performances théoriques du BCI corrigé

L'objet de cette section est de caractériser théoriquement les performances du BCI corrigé par le détecteur d'erreur. Il est en effet nécessaire de pouvoir évaluer, lors de la phase de design d'un

BCI, l'amélioration que pourrait apporter l'intégration d'un détecteur d'erreur, préalablement aux tests expérimentaux. Cette évaluation nous permettra en particulier de choisir entre les approches détection et classification du détecteur.

Les critères de performance choisis sont la probabilité d'erreur du BCI lorsqu'il prend une décision se traduisant par une commande, la probabilité que le sujet doive répéter sa tâche imaginaire et enfin le taux de transfert qui combine les deux critères précédents.

Le problème est traité dans le cas général multiclasse, où le nombre de tâches à reconnaître est supérieur à deux.

4.4.1 Notation

La figure 4.10 présente les notations que nous utiliserons par la suite.

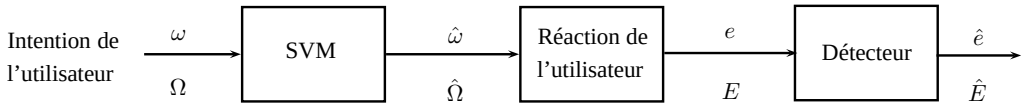


FIGURE 4.10 – Schéma du problème

Nous modélisons le problème à l'aide de quatre variables aléatoires :
- Ω et $\hat{\Omega}$, à valeur dans $\{\omega_1, \ldots, \omega_n\}$ représentant respectivement l'intention de mouvement du sujet et la décision du SVM_1 (avec n le nombre de classes), ω et $\hat{\omega}$ étant les réalisations correspondantes ;
- E et $\hat{E}$ à valeur dans $\{juste, faux\}$ représentant l'exactitude de la décision du SVM_1 et son estimation par le détecteur, e et $\hat{e}$ étant les réalisations correspondantes ($e = juste$ si $\omega = \hat{\omega}$, $faux$ sinon).

On note :
- $P_{\Omega}(\omega) = \text{Prob}(\Omega = \omega)$: probabilité *a priori* de la classe ω,
- $P_{\hat{\Omega}|\Omega}(\hat{\omega},\omega) = \text{Prob}(\hat{\Omega} = \hat{\omega}|\Omega = \omega)$: probabilité de la décision du SVM_1 conditionnellement à la classe de mouvement,
- $P_{\hat{E}|\hat{\Omega}\Omega}(\hat{e},\hat{\omega},\omega) = P_{\hat{E}|E}(\hat{e},e) = \text{Prob}(\hat{E} = \hat{e}|E = e)$: probabilité de la décision du détecteur conditionnellement à l'exactitude de la réponse.
- $P_{\Omega|\hat{\Omega}\hat{E}}(\omega,\hat{\omega},\hat{e}) = \text{Prob}(\Omega = \omega|\hat{\Omega} = \hat{\omega},\hat{E} = \hat{e})$: probabilité *a posteriori* de la classe ω.

4.4.2 Probabilité d'erreur

La probabilité d'erreur du système non corrigé, notée P_{Err_0}, correspond à la probabilité d'erreur du SVM_1 :

$$P_{Err_0} = \sum_{\substack{i,j \\ i \neq j}} P_{\hat{\Omega}|\Omega}(\hat{\omega}_j,\omega_i).P_{\Omega}(\omega_i) \tag{4.6}$$

Pour calculer la probabilité d'erreur du système corrigé, il est nécessaire :
- de calculer la probabilité *a posteriori* d'une classe sachant la réponse du SVM_1 et du détecteur d'erreur,
- de définir au préalable une stratégie de décision à partir de ces réponses.

4.4.2.1 Probabilité *a posteriori* d'une classe

La probabilité *a posteriori* de la classe ω sachant la réponse du classifieur ($\hat{\omega}$) et du détecteur ($\hat{e}$) est donnée par le théorème de Bayes [Duda *et al.*, 2001] :

$$P_{\Omega|\hat{\Omega}\hat{E}}(\omega,\hat{\omega},\hat{e}) = \frac{P_{\hat{E}|\hat{\Omega}\Omega}(\hat{e},\hat{\omega},\omega).P_{\hat{\Omega}|\Omega}(\hat{\omega},\omega).P_{\Omega}(\omega)}{P_{\hat{\Omega}\hat{E}}(\hat{\omega},\hat{e})} \tag{4.7}$$

avec $P_{\hat{\Omega}\hat{E}}(\hat{\omega},\hat{e}) = \sum_{i=1}^{n} P_{\hat{\Omega}|\Omega}(\hat{\omega},\omega_i).P_{\hat{E}|\hat{\Omega}\Omega}(\hat{e},\hat{\omega},\omega_i).P_{\Omega}(\omega_i)$ et n le nombre de classes.

4.4.2.2 Stratégie de décision

Le détecteur fournit une information sur la justesse de la réponse du SVM_1, mais pas sur la classe. Si la réponse donnée par le SVM_1 est détectée comme *faux*, dans le cas général où le nombre de classe est supérieur à 2, il n'est pas possible d'en déduire la classe réelle même avec un détecteur parfait. Dans le cas biclasse il serait possible de corriger la classe en prenant la classe opposée si le détecteur estime la réponse du SVM_1 comme *faux*. Cependant l'observation des résultats présentés dans la section suivante permet d'observer (tableau 4.5) que pour certains sujets cette stratégie risque de conduire à un taux d'erreur du système corrigé plus grand qu'avec un système sans détecteur d'erreurs, dans la mesure où la probabilité *a posteriori* de la classe opposée est en moyenne de l'ordre de 50 à 60%. Ainsi une stratégie naturelle pour intégrer le détecteur dans le système complet est la suivante :

- si la réponse du SVM_1 est estimée comme *juste*, on valide la décision.
- si la décision est estimée comme *faux*, on demande au sujet de répéter la tâche mentale ;

Dans le cas où le détecteur est basé sur l'algorithme de détection à deux seuils, il ne fournit pas toujours une réponse. Nous proposons :

- s'il répond : appliquer la stratégie précédente,
- sinon, on choisit la classe en accord avec la décision du SVM_1.

4.4.3 Probabilité d'erreur totale du BCI corrigé

En accord avec cette stratégie, la probabilité d'erreur du BCI, lorsque le détecteur donne une réponse, correspond à la probabilité que la classe $\hat{\omega}$ donnée par le SVM_1 soit différente de la réelle intention ω de l'utilisateur alors que le détecteur estime sa réponse juste. On la note P_{Err_1} :

$$\begin{aligned} P_{Err_1} &= \sum_{\substack{i,j \\ i\neq j}} P_{\Omega\hat{\Omega}|\hat{E}}(\omega_i,\hat{\omega}_j,juste) \\ &= \sum_{\substack{i,j \\ i\neq j}} P_{\Omega|\hat{\Omega}\hat{E}}(\omega_i,\hat{\omega}_j,juste)P_{\hat{\Omega}|\hat{E}}(\hat{\omega}_j,juste) \\ &= \sum_{\substack{i,j \\ i\neq j}} \frac{P_{\Omega|\hat{\Omega}\hat{E}}(\omega_i,\hat{\omega}_j,juste)P_{\hat{\Omega}\hat{E}}(\hat{\omega}_j,juste)}{P_{\hat{E}}(juste)} \end{aligned} \tag{4.8}$$

Ce qui peut s'écrire en utilisant l'équation (4.7) :

$$P_{Err_1} = \sum_{\substack{i,j \\ i\neq j}} \frac{P_{\hat{E}|\hat{\Omega}\Omega}(juste,\hat{\omega}_j,\omega_i).P_{\hat{\Omega}|\Omega}(\hat{\omega}_j,\omega_i).P_{\Omega}(\omega_i)}{P_{\hat{E}}(juste)} \tag{4.9}$$

avec

$$P_{\hat{E}}(juste) = \sum_{e} P_{\hat{E}|E}(juste,e).P_E(e)$$

$$P_E(e) = \sum_{i,j} P_{\hat{\Omega}|\Omega}(\hat{\omega}_j,\omega_i)P_\Omega(\omega_i) \text{ avec } \begin{cases} i = j \text{ si } e \textit{ juste} \\ i \neq j \text{ si } e \textit{ faux} \end{cases} .$$

On peut remarquer que comme i est différent de j, $P_{\hat{E}|\hat{\Omega}\Omega}(juste,\hat{\omega}_j,\omega_i)$ est égal à la probabilité $P_{\hat{E}|E}(juste,faux)$ que le détecteur annonce la décision du SVM_1 comme juste alors qu'elle est fausse et est donc indépendant de i et j. On peut donc écrire :

$$\begin{aligned} P_{Err_1} &= \sum_{\substack{i,j \\ i\neq j}} \frac{P_{\hat{E}|E}(juste,faux)}{P_{\hat{E}}(juste)}.P_{\hat{\Omega}|\Omega}(\hat{\omega}_j,\omega_i).P_\Omega(\omega_i) \\ &= \frac{P_{\hat{E}|E}(juste,faux)}{P_{\hat{E}}(juste)}.P_E(faux) \end{aligned} \tag{4.10}$$

$P_E(faux) = P_{Err_0}$ étant la probabilité d'erreur du SVM_1 seul :

$$P_{Err_1} = \frac{P_{\hat{E}|E}(juste,faux)}{P_{\hat{E}}(juste)}.P_{Err_0} \tag{4.11}$$

En intégrant $P_{reponse}$ la probabilité de réponse du détecteur, le taux d'erreur global devient :

$$\begin{aligned} P_{Err} &= \frac{(1-P_{reponse}).P_{Err_0} + P_{reponse}.P_{\hat{E}}(juste).P_{Err_1}}{(1-P_{reponse}) + P_{reponse}.P_{\hat{E}}(juste)} \\ &= \frac{(1-P_{reponse}) + P_{reponse}.P_{\hat{E}|E}(juste,faux)}{(1-P_{reponse}) + P_{reponse}.P_{\hat{E}}(juste)}.P_{Err_0} \end{aligned} \tag{4.12}$$

Dans le cas de l'approche classification pour la détection d'erreurs, $P_{reponse} = 1$ et $P_{Err} = P_{Err_1}$.

4.4.4 Probabilité de répétition et taux de transfert

La probabilité d'erreur totale ne suffit pas à caractériser les performances du système. En effet, on a vu que la stratégie de décision pouvait nécessiter de répéter la tâche. Il n'est évidemment pas souhaitable que le sujet répète indéfiniment sa tâche pour que la probabilité d'erreur soit nulle. Les performances du système corrigé sont donc aussi caractérisées par la probabilité que le sujet doive répéter la tâche imaginaire. Ce qui correspond à la probabilité que le détecteur estime la décision du SVM_1 comme *faux* lorsqu'il donne une réponse :

$$P_{repete} = P_{reponse}.P_{\hat{E}}(faux) \tag{4.13}$$

et la probabilité que l'utilisateur ait à répéter la même tâche m fois est $(P_{repete})^m$.

Un critère de qualité couramment utilisé dans le domaine des BCI, le taux de transfert en bits (BpT, Bit per Trial) [Blankertz *et al.*, 2002], va nous permettre de combiner les deux. Le taux de transfert est le nombre de bits (mouvements bien classés) qui sont transmis par unité de temps (durée d'un essai). Plus ce taux est élevé, plus le BCI est performant.

En reprenant les probabilités définies ci-dessus, sa formulation classique (dans le cas où le BCI prend toujours une décision) pour un problème à n classes est la suivante :

$$\text{BpT} = \log_2(n) + (1-P_{Err})\log_2(1-P_{Err}) + P_{Err}\log_2(\frac{P_{Err}}{n-1}) \tag{4.14}$$

Cette expression est celle qui sera utilisée pour caractériser le BCI non corrigé (avec $P_{Err} = P_{Err_0}$).

Dans le cas du BCI corrigé, qui ne donne pas une réponse à tous les essais, nous utiliserons une expression modifiée pour prendre en compte le taux de répétition :

$$\mathrm{BpT} = \left[\log_2(n) + (1-P_{Err})\log_2(1-P_{Err}) + P_{Err}\log_2(\frac{P_{Err}}{n-1})\right](1-P_{repete}) \tag{4.15}$$

Ces formules seront utilisées dans la section 4.5.2 avec des probabilités conditionnelles estimées sur des signaux expérimentaux.

Le taux de transfert ne peut à lui seul caractériser l'amélioration des performances du système, bien qu'il prenne en compte la probabilité d'erreur et le taux de répétition. En effet, le taux d'amélioration du BpT dépend des performances du système non corrigé : plus elles sont faibles, meilleure est l'amélioration du BpT avec un même détecteur d'erreur.

4.5 Résultats

Nous présentons ici les résultats obtenus sur 6 sujets, sur des signaux enregistrés selon le protocole décrit en Annexe 1.4.3.

4.5.1 Reconnaissance des ErrP

4.5.1.1 Approche détection (détecteur$_\mathbf{d}$)

Les résultats présentés tableau 4.1 et 4.2 ont été obtenus avec l'approche détection en utilisant dans un premier temps toutes les voies F_z, FC_z, CP_z, C_1, C_2, C_z puis en nous restreignant aux voies C_z et FC_z. Dans les deux cas, et comme nous l'avons vu en Section 4.3.1.1, nous réalisons un filtrage spatial, c'est-à-dire une combinaison linéaire des différentes voies, soit en les pondérant avec des poids égaux (moyenne) soit en optimisant un critère de type rapport signal sur bruit (CSP) ou caractérisant la capacité à discriminer les classes (Fisher).

Les seuils pour la détection s_{faux} et s_{juste} ont été fixés de manière à obtenir moins de 20% de vrais négatifs ($C_{juste} < 20\%$) parmi l'ensemble des signaux de la classe *juste* ($\mathcal{X}_j$) et moins de 20% de faux négatifs ($C_{faux} < 20\%$) parmi l'ensemble des signaux de la classe *faux* ($\mathcal{X}_f$).

Utilisation des 6 voies Le tableau 4.1 présente le pourcentage de bonnes détection des classes *juste* et *faux* ainsi que le pourcentage de réponses obtenus avec l'approche détection pour les trois filtres spatiaux décrits dans la Section A.2.

On constate que l'utilisation de filtres spatiaux n'augmente pas de façon significative les performances de l'approche détection. En moyenne, si le CSP et Fisher augmentent légèrement le pourcentage de réponses (Moyenne : 57.6%, CSP : 58.9%, Fisher : 61.4%), le pourcentage de bien classés de la classes *juste* est meilleur avec la moyenne des voies (Moyenne : 66%, CSP : 62%, Fisher : 64%). Le pourcentage de bien classés de la classe *faux* est sensiblement le même pour les trois méthodes (Moyenne : 72%, CSP : 73%, Fisher : 72%). Si on accorde plus d'importance au pourcentage de bien classés qu'au taux de réponse, la moyenne des voies doit être retenue comme méthode. Nous avons ici utilisé toutes les voies, dans le prochain paragraphe, nous donnerons les résultats en utilisant un nombre réduit de voies (FC_z et C_z).

TABLE 4.1 – Resultats de l'approche détection des ErrP en utilisant les différents filtres spatiaux sur les 6 voies (FC_z, F_z, C_z, CP_z, C_1, C_2). Les filtres spatiaux sont optimisés sur un ensemble d'apprentissage. On indique les taux de réponse et les pourcentages de bonnes réponses du détecteur.

Sujets		1	2	3	4	5	6	Moyenne
Filtre moyenne	% réponses	57.5%	38.3%	75.0%	75.8%	50.8%	48.3%	57.6%
	classe *juste*	69%	51%	78%	75%	58%	66%	**66%**
	classe *faux*	75%	55%	80%	81%	76%	64%	72%
	Moyenne	71%	52%	79%	77%	63%	66%	**68%**
Filtre CSP	% réponses	45.8%	27.5%	72.5%	75.8%	55.8%	75.8%	58.9%
	classe *juste*	59%	15%	77%	77%	67%	75%	62%
	classe *faux*	72%	62%	81%	75%	72%	77%	**73%**
	Moyenne	63%	27%	78%	77%	68%	76%	65%
Filtre Fisher	% réponses	38.3%	46.7%	77.5%	82.5%	52.5%	70.8%	**61.4%**
	classe *juste*	26%	67%	76%	77%	64%	75%	64%
	classe *faux*	64%	55%	80%	83%	73%	77%	72%
	Moyenne	36%	64%	77%	79%	66%	76%	66%

Utilisation des voies FC_z et C_z. Nous savons que les informations sur les ErrP se situent sur la ligne médiane du crâne et nous avons remarqué (Section 4.2) que les signaux des voies C_z et CP_z sont très proches ainsi que ceux enregistrés sur les électrodes F_z et FC_z. Nous avons donc étudié ensuite la détection en utilisant simplement les voies FC_z et C_z. Dans le tableau 4.2 nous montrons les résultats en optimisant les différents filtres spatiaux sur ces 2 voies.

TABLE 4.2 – Résultats de l'approche détection des ErrP en optimisant les différents filtres spatiaux sur les voies FC_z et C_z. Les filtres spatiaux sont optimisés sur un ensemble d'apprentissage. On indique les taux de réponse et les pourcentages de bonnes réponses du détecteur.

Sujets		1	2	3	4	5	6	Moy.
Filtre moyenne	% réponses	63.3%	37.5%	85.0%	81.7%	61.7%	49.2%	**63.1%**
	classe *juste*	74%	50%	78%	77%	67%	65%	**69%**
	classe *faux*	74%	62%	80%	80%	77%	62%	**73%**
	Moyenne	74%	53%	79%	78%	70%	64%	**70%**
Filtre CSP	% réponses	60.0%	35.8%	79.2%	56.7%	65.8%	51.7%	58.2%
	classe *juste*	74%	53%	77%	67%	73%	65%	68%
	classe *faux*	74%	63%	80%	83%	76%	62%	73%
	Moyenne	74%	56%	78%	71%	74%	64%	**70%**
Filtre Fisher	% réponses	57.5%	41.7%	78.3%	69.2%	55.0%	45.8%	57.9%
	classe *juste*	67%	58%	85%	73%	65%	61%	68%
	classe *faux*	78%	50%	78%	79%	72%	64%	70%
	Moyenne	70%	56%	83%	75%	67%	62%	69%

Quelle que soit la méthode, on constate que les résultats sont meilleurs avec les voies FC_z et C_z qu'en utilisant toutes les voies. De plus, l'utilisation de la moyenne des voies donne de meilleurs résultats en tous points :

- Pourcentage de réponses (Moyenne : 63.1% , CSP : 58.2% , Fisher : 57.9%)
- Pourcentage de bien classé de la classe *juste* (Moyenne : 69% , CSP : 68% , Fisher : 68%)
- Pourcentage de bien classé de la classe *faux* (Moyenne : 73% , CSP : 73% , Fisher : 70%)

Nous utiliserons les résultats obtenus avec la moyenne des signaux mesurés sur les voies FC_z et C_z pour les estimations de probabilité d'erreur de l'approche détection.

4.5.1.2 Approche classification (détecteur$_c$)

Cette partie montre les résultats obtenus en utilisant l'approche classification. Nous présentons les résultats obtenus en utilisant toutes les voies, puis en utilisant seulement les voies FC_z et C_z.

Les descripteurs de chaque voie utilisés sont les échantillons des signaux filtrés par un filtre passe bas 0-10 Hz et décimés 32 fois. On obtient 16 descripteurs par voies. Le vecteur descripteur du signal multi-voies regroupe les descripteurs des différentes voies.

TABLE 4.3 – Resultats de l'approche classification des ErrP en utilisant toutes les voies F_z, FC_z, CP_z, C_1, C_2, C_z. On indique les pourcentages de bonnes réponses du détecteur.

Sujets		1	2	3	4	5	6	Moy.
% bien classés	classe *juste*	82%	81%	79%	77%	73%	87%	80%
	classe *faux*	63%	37%	43%	50%	50%	70%	52%
	Moyenne	77%	70%	70%	70%	67%	83%	73%

Utilisation des 6 voies Nous remarquons que l'approche classification a tendance à mieux classer les signaux de la classe *juste* (80% de bien classés) que les signaux de la classe *faux* (52% de bien classés). L'utilisation de toutes les voies nous donne une classification dans un espace de dimension 96. Nous ne possédons que 120 essais pour déterminer la séparatrice, ce qui est peu dans un espace de cette taille.

Pour réduire la dimension de l'espace de représentation nous avons utilisé l'approche classification avec seulement les voies FC_z et C_z.

Utilisation des voies FC_z et C_z Nous présentons tableau 4.4 les résultats que nous avons obtenus pour les 6 individus avec l'approche classification en utilisant les voies FC_z et C_z.

TABLE 4.4 – Resultats de l'approche classification des ErrP en utilisant les voies FC_z et C_z. On indique les pourcentages de bonnes réponses du détecteur.

Sujets		1	2	3	4	5	6	Moy.
% bien classés	classe *juste*	92%	91%	93%	87%	83%	80%	88%
	classe *faux*	73%	43%	60%	60%	37%	63%	56%
	Moyenne	87%	79%	85%	80%	72%	76%	80%

L'utilisation de seulement deux voies augmente sensiblement les performances de la méthode. Le taux de bien classés de la classe *juste* passe de 80% à 88% et celui de la classe *faux* passe de 52% à 56%. L'approche classification classe toujours mieux les signaux de la classe *juste* que ceux de la classe *faux*, tandis que l'approche détection permet d'avoir un taux de bien classés mieux réparti dans les deux classes. Cependant, avec les seuils fixés de façon à avoir moins de 20% de fausses alarmes, l'approche détection n'évalue que 63% des signaux en moyenne sur tous les sujets (les signaux dont l'index est situé entre les deux seuils ne sont pas traités par le détecteur).

Les résultats en moyenne du tableau 4.4 seront utilisés comme estimation des probabilités conditionnelles $P_{\hat{E}|\hat{\Omega}\Omega}(\hat{e},\hat{\omega},\omega)$ dans la partie suivante.

A cette étape, il est difficile de décider quelle est la meilleure méthode simplement au vu des résultats précédents. Seules les performances en sortie du système corrigé (probabilité d'erreur, taux de transfert) permettront de faire un choix et de savoir si la correction par ce détecteur imparfait est pertinente.

4.5.2 Résultats de l'amélioration du BCI corrigé

4.5.2.1 Probabilités *a posteriori*

Dans cette partie nous avons utilisé les résultats de la classification des mouvements en utilisant les marginales de la DWT non optimisée de la voie C_z (première colonne du tableau 2.2) comme estimation des probabilités conditionnelles du SVM_1. Les probabilités conditionnelles du $détecteur_d$ (resp. $détecteur_c$) sont obtenues en utilisant les résultats dans le tableau 4.2 avec l'utilisation de la moyenne des voies (resp. tableau 4.4).

Les probabilités *a posteriori* que nous obtenons sont données tableau 4.5.

TABLE 4.5 – $P_{\Omega|\hat{\Omega}\hat{E}}(\omega,\hat{\omega}=\omega,e)$: Probabilité *a posteriori* de la classe ω donnée par le SVM_1, en fonction de la réponse du détecteur, pour les deux approches ($détecteur_d$ et $détecteur_c$) de la détection d'erreurs.

	$détecteur_d$		$détecteur_c$	
Sujets	$\hat{e}=faux$	$\hat{e}=juste$	$\hat{e}=faux$	$\hat{e}=juste$
1	53%	90%	26%	92%
2	45%	57%	17%	61%
3	36%	89%	19%	83%
4	36%	88%	29%	81%
5	44%	84%	46%	71%
6	75%	90%	63%	92%
Moy.	48%	83%	33%	80%

Lorsque le détecteur estime que la classe donnée par le SVM_1 est juste, la probabilité d'avoir cette est de 83% avec le $détecteur_c$ (80% avec le $détecteur_d$). Il est alors justifié de lui faire confiance. En revanche lorsque le détecteur estime que la classe choisie par le système de décision est fausse, la probabilité de la classe opposée n'est que de 48% (resp. 33%). Ceci justifie la stratégie proposée de demander au sujet de répéter la tâche.

4.5.2.2 Résultats en sortie du BCI corrigé

Dans le Tableau 4.6, nous avons reporté les résultats obtenus avec les signaux expérimentaux.

En moyenne sur tous les sujets, en utilisant l'approche classification pour la détection d'erreurs, la probabilité d'erreur globale est de 20% (24% avec l'approche détection) et le taux de répétition est de 26% (27% pour l'approche détection). Ces résultats sont à comparer avec les 32% d'erreur dans le cas du SVM_1 seul. La moyenne du taux de transfert initial est de 0.13 et il est amélioré de 46% avec l'approche détection (0.19 BpT) et de 92% avec l'approche classification (0.25 BpT). Ces résultats montrent que même avec une faible détection des ErrP, son

TABLE 4.6 – Probabilité d'erreur globale, probabilité de répétition et taux de transfert en bits du système sans correction, corrigé avec le détecteur$_{\mathrm{d}}$ et avec le détecteur$_{\mathrm{c}}$

	Sans correction		détecteur$_{\mathrm{d}}$			détecteur$_{\mathrm{c}}$		
Suj.	P_{Err}	BpT	P_{Err}	P_{repete}	BpT	P_{Err}	P_{repete}	BpT
1	24%	0.21	17.3%	23.8%	0.27	8.4%	23.6%	0.44
2	50%	0.00	48.5%	21.0%	0.00	38.5%	26.0%	0.03
3	33%	0.09	17.7%	32.3%	0.22	17.5%	25.5%	0.25
4	34%	0.08	18.0%	34.6%	0.21	19.1%	28.9%	0.21
5	35%	0.07	26.1%	29.9%	0.12	29.0%	24.0%	0.10
6	16%	0.37	13.8%	19.3%	0.34	8.1%	26.9%	0.43
Moy.	32%	0.13	23.5%	26.7%	0.19	20.1%	25.6%	0.25

incorporation dans le BCI augmente significativement ses performances. Ces résultats montrent aussi que l'approche classification est plus efficace que l'approche détection.

Dans le tableau suivant, nous présentons les résultats du système corrigé obtenus en moyenne sur tous les sujets avec pour estimation de la probabilité d'erreur du SVM$_1$ les résultats de la classification des signaux d'EEG$_1$ en utilisant comme descripteurs :

- les marginales de la DWT de la voie C$_{\mathrm{z}}$ sans optimisation d'ondelette (méthode 1), méthode utilisée dans le cas des résultats du tableau 4.6.
- les marginales de la DWT de la voie C$_{\mathrm{z}}$ en optimisant l'ondelette mère avec le critère de Fisher (méthode 2),
- les marginales de la DWT des voies C$_{\mathrm{z}}$, C$_1$ et C$_2$ sans optimisation d'ondelette (méthode 3),
- les marginales de la DWT des voies C$_{\mathrm{z}}$, C$_1$ et C$_2$ en optimisant une ondelette mère par voie avec le critère de Fisher (méthode 4).

TABLE 4.7 – Probabilité d'erreur globale, probabilité de répétition et taux de transfert en bits du système sans correction, corrigé avec le détecteur$_{\mathrm{d}}$ et avec le détecteur$_{\mathrm{c}}$ en moyenne sur tous les individus pour différentes méthodes d'extraction des caractéristiques des signaux EEG$_1$.

	Sans correction		détecteur$_{\mathrm{d}}$			détecteur$_{\mathrm{c}}$		
Méthode	P_{Err}	BpT	P_{Err}	P_{repete}	BpT	P_{Err}	P_{repete}	BpT
1	32%	0.13	23.5%	26.7%	0.19	20.1%	25.6%	0.25
2	26%	0.19	18.4%	26.4%	0.24	15.3%	24.3%	0.31
3	29%	0.16	20.4%	26.6%	0.21	17.3%	24.5%	0.28
4	**20%**	**0.32**	**12.8%**	**24.8%**	**0.35**	**11.0%**	**21.1%**	**0.42**

Les meilleurs performances théoriques du système corrigé sont évidemment obtenues dans le cas où le classifieur SVM$_1$ est le plus précis, c'est-à-dire quand on utilise un signal multivoie en optimisant une ondelette par voie. Cependant on peut noter que dans ce cas, l'amélioration du BpT est seulement de 31% alors qu'elle est de 92% dans le cas où le classifieur est le moins performant.

4.6 Conclusion

L'étude précédente permet de quantifier l'amélioration des performances d'un BCI apportée par un détecteur donné.

Nous avons montré qu'il était possible de détecter automatiquement les erreurs du BCI afin d'améliorer son taux d'erreur et son taux de transfert.

A titre de comparaison et avec le BCI (SVM_1) faisant 32% d'erreur :
- un détecteur parfait conduirait à un taux d'erreur nul, une probabilité de répétition de 32% et un taux de transfert de 0.68.
- un détecteur qui répondrait toujours juste c'est-à-dire 75% de bonnes décisions (plus que le $\text{détecteur}_\text{d}$) ne changerait en rien les probabilités du BCI sans correction.
- un détecteur purement aléatoire conduirait à une probabilité d'erreur de 32%, un taux de répétition de 50% et un taux de transfert de 0.07, dégradant ainsi les performances du BCI non corrigé.

D'une façon plus générale, la figure suivante représente les performances du BCI corrigé en terme de taux de transfert en fonction des taux de bonnes réponses du détecteur.

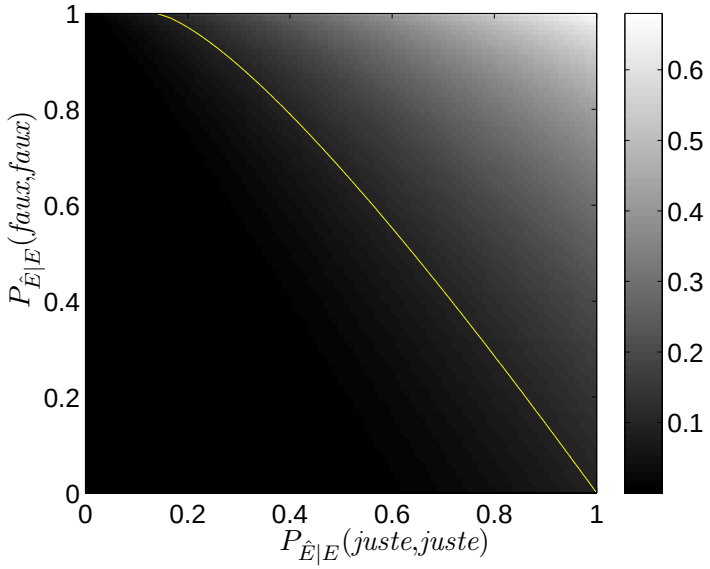


FIGURE 4.11 – Graphe représentant le taux de transfert du système corrigé en fonction des différentes probabilités du détecteur. La ligne claire représente les valeurs pour lesquelles le taux de transfert du système corrigé est égal au taux de transfert du BCI sans correction. Au dessus de cette ligne, le détecteur améliore les performances du BCI, en dessous le détecteur dégrade les performances du BCI.

Ces résultats peuvent être utilisés pour le design de tout BCI où l'enregistrement d'ErrP est possible.

Dans la prochaine partie nous verrons comment utiliser les résultats du détecteur pour ajouter des exemples à l'ensemble d'apprentissage et améliorer le pourcentage de mal classé du SVM_1.

Chapitre 5

Simulation du BCI adaptatif

Sommaire

Nous avons vu qu'il était possible d'améliorer les performances du BCI en corrigeant la sortie du système de décision grâce à la réponse du détecteur d'erreur. Il est également nécessaire d'intégrer en ligne de nouveaux éléments à la population d'apprentissage pour éviter un apprentissage trop long, et permettre au système de s'adapter aux changements de l'utilisateur. La mise au point d'un tel système d'apprentissage en ligne requiert le choix d'une stratégie de mise à jour, et le réglage de différents paramètres, qui ne peuvent se faire dans des conditions réelles. C'est pourquoi nous avons développé un simulateur.

Dans ce chapitre nous commencerons par présenter le fonctionnement du simulateur en détaillant comment sont simulés les différents éléments du BCI adaptatif. Puis nous verrons les résultats de simulation pour différentes évolutions des classes et différentes stratégies de mise à jour de la population d'apprentissage.

5.1 Simulateur

L'étude en simulation permet d'estimer à chaque instant les performances du système évolutif (qui ne peut pas être fait sur des données expérimentales à partir d'un essai) et les comparer à des références connues.

Le simulateur que nous avons développé représente le fonctionnement du BCI complet (Figure 1.1). Le système se décompose en quatre parties :

- simulation des données à classer,
- le système de classification,

– le détecteur d'erreur,
– la stratégie de mise à jour de l'ensemble d'apprentissage.

Nous allons détailler l'implémentation de chaque partie dans le simulateur.

5.1.1 Simulation des données à classer

Le but du simulateur n'est pas de simuler des EEG réalistes, mais des classes caractérisées par des densités de probabilités connues afin de pouvoir disposer de références pour évaluer les performances du système. Nous avons simulé directement les valeurs des vecteurs descripteurs des signaux.

Nous avons considéré c classes $\{\omega_1, \ldots, \omega_c\}$. Soit x un vecteur de dimension m, élément de la classe ω_i (un élément simulant un essai), ses composantes sont simulées grâce à une loi normale multidimensionnelle :

$$P_{X|\Omega}(x,\omega_i) = \mathcal{N}(\mu_i(t), \Sigma_i(t)) = \frac{1}{(2\pi)^{\frac{m}{2}}|\Sigma_i(t)|^{\frac{1}{2}}} \exp\left(\frac{1}{2}(x-\mu_i(t))^t \Sigma_i(t)^{-1}(x-\mu_i(t))\right) \quad (5.1)$$

avec $|\Sigma_i(t)|$ le déterminant de la matrice $\Sigma_i(t)$.

Les moyennes $(\mu_i(t))$ et variances $(\Sigma_i(t))$ des distributions peuvent varier au cours du temps, permettant de simuler le changement d'état mental d'un utilisateur. Pour ce faire nous pouvons prédéfinir la trajectoire que les classes doivent suivre pendant la simulation.

Nous voulons pouvoir afficher la position des classes, des individus générés et des séparatrices calculées, aussi les simulations sont faites dans un espace de représentation à deux dimensions $(m = 2)$.

5.1.2 Le système de classification (SVM_1)

Les deux algorithmes de SVM étudiés Sections 3.3.1 et 3.3.2 ont été implémentés. Dans le cas où nous simulons plus de deux classes, une procédure de OVR ou de OVO est utilisée (Section 3.1.5) pour discriminer toutes les classes.

Pour évaluer les performances du SVM_1 au cours de la simulation nous les avons comparées à celles d'un classifieur bayesien. Les classes simulées suivent des lois gaussiennes dont on connaît parfaitement les paramètres. Dans ce cas le classifieur bayesien est optimal. Contrairement aux SVM qui utilisent seulement les éléments simulés (ensemble d'apprentissage) pour calculer les fonctions de décision, le classifieur bayesien utilise les paramètres des lois utilisées pour simuler les individus.

Classifieur bayesien. Le classifieur bayesien consiste à affecter l'individu à la classe dont il maximise la probabilité *a posteriori*. Pour un individu x la probabilité *a posteriori* d'appartenir à la classe ω_i est donnée par :

$$P_{\Omega|X}(\omega_i,x) = P_\Omega(\omega_i).P_{X|\Omega}(x,\omega_i) \quad (5.2)$$

avec $P_\Omega(\omega_i)$ la probabilité *a priori* de la classe ω_i. On affecte donc x à la classe $\hat{\omega}_i$ telle que :

$$\hat{\omega}_i = \underset{\omega_i}{\operatorname{argmax}}\ P_{\Omega|X}(\omega_i|x) \quad (5.3)$$

5.1.3 Le détecteur d'erreur

Après avoir estimé la classe $\hat{\omega}$ d'un individu x dont la classe réelle est ω, on simule la réponse du détecteur d'erreur par la réalisation $\hat{e}$ d'une variable aléatoire binaire $\hat{E}$ de densité conditionnelle $P_{\hat{E}|E}$. On génère $\hat{e}$ en tirant une valeur a de la variable aléatoire A de densité de probabilité uniforme entre 0 et 1 puis :

- si $\hat{\omega} = \omega$
  - $a < P_{\hat{E}|E}(juste,juste)$ alors $\hat{e} = juste$
  - sinon $\hat{e} = faux$
- si $\hat{\omega}_i \neq \omega_i$
  - $a < P_{\hat{E}|E}(faux,faux)$ alors $\hat{e} = faux$
  - sinon $\hat{e} = juste$

5.1.4 Ensemble d'apprentissage et mise à jour

A chaque instant le classifieur apprend sa règle de décision sur une population contenue dans une fenêtre glissante. Pour que le système s'adapte aux changements de l'état mental de l'utilisateur, sans dégrader ses performances, l'ensemble d'apprentissage doit être mis à jour avec de nouveaux exemples pertinents dont l'étiquette est suffisamment fiable. Le détecteur ne fournissant qu'une information sur la justesse la décision du SVM_1, dans le cas général multiclasses, il n'est pas possible de corriger la classe lorsqu'elle est détectée fausse. Cependant nous avons vu dans la Section 4.7 que, lorsque le détecteur estime la réponse *juste*, il y a une forte probabilité pour qu'elle le soit réellement. C'est pourquoi dans les stratégies de mise à jour étudiées, seuls les éléments estimés *juste* sont intégrés au nouvel ensemble d'apprentissage. Parmi les exemples estimés *juste*, deux stratégies permettant de les intégrer à la population d'apprentissage ont été implémentées :

- prendre tous les exemples estimés *juste* par le détecteur,
- parmi les exemples estimés *juste* par le détecteur, n'utiliser que ceux qui sont proches de la fonction de décision précédente. Cette technique correspond à l'échantillonnage par incertitude utilisé dans l'apprentissage actif [Schohn et Cohn, 2000]. La proximité par rapport à la séparatrice est mesurée par la valeur de la fonction de décision f dans le cas des SVM. Nous avons choisi de n'utiliser que les éléments qui se situent à l'intérieur des marges (pour un individu x estimé appartenir à la classe ω_i, il est ajouté à l'ensemble d'apprentissage seulement si $|f_i(x)| \leqslant 1$).

Une fois la taille de la population d'apprentissage maximale atteinte, lorsqu'un élément est ajouté, l'élément le plus ancien de l'ensemble d'apprentissage est retiré. C'est celui qui est supposé représenter le moins l'état mental actuel de l'utilisateur.

Comme initialisation, l'ensemble d'apprentissage est composé de n_{init} exemples étiquetés. Ensuite, les nouveaux éléments sont ou non intégrés avec l'étiquette estimée par le SVM_1, selon l'opinion $\hat{e}$ du détecteur et la stratégie d'intégration choisie. La taille de l'ensemble d'apprentissage augmente jusqu'à avoir n_w individus, correspondant à la taille de la fenêtre glissante. Une fois que l'ensemble d'apprentissage a atteint la taille de la fenêtre glissante, à chaque ajout d'un individu dans la population d'apprentissage on en retire un. L'utilisation d'une fenêtre glissante permet de suivre l'évolution des classes.

5.1.5 L'interface

La figure 5.1 présente l'interface du simulateur développé.

L'interface se divise en trois parties :

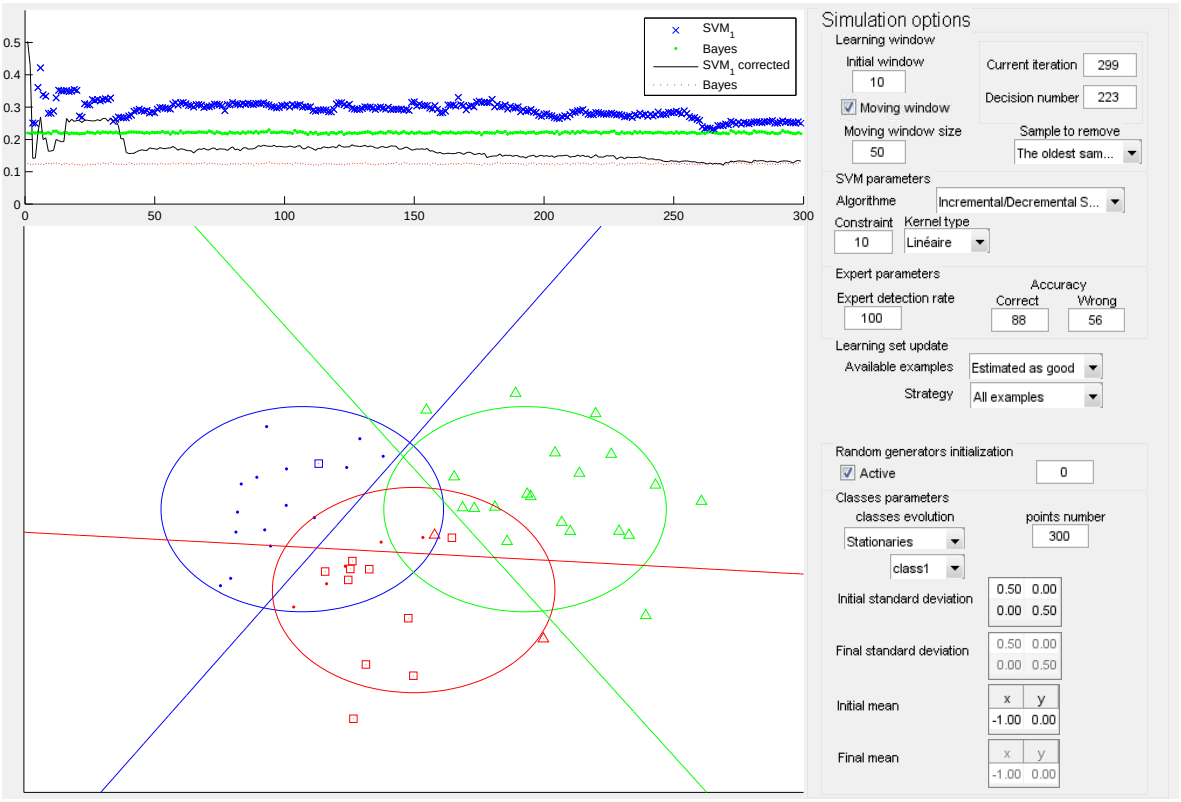


FIGURE 5.1 – Interface graphique du simulateur

- le menu à droite permet de gérer les différents paramètres de simulation,
- le graphique en bas à gauche représente la position des classes (ellipse de confiance à 80%), des séparatrices du SVM_1 et des exemples utilisés dans l'ensemble d'apprentissage à l'itération courante,
- en haut à gauche, les évolutions de différents pourcentages de mal classés au fil des itérations sont représentées. A chaque instant, les probabilités d'erreur de classification sont estimées empiriquement à partir d'un ensemble de test de 10000 exemples générés selon les densités courantes $\mathcal{N}(\mu_i(t),\Sigma_i(i))$. Sur la Figure 5.2 sont détaillés les différents tracés :

  Croix SVM_1 sans détecteur d'erreur : la décision du SVM_1 n'est pas corrigée et l'apprentissage est mis à jour en intégrant chaque nouvel exemple avec l'étiquette de la classe décodée par le SVM_1,

  Points classifieur bayésien idéal connaissant la densité théorique à tout instant, sans détecteur d'erreur pour corriger sa décision,

  Trait continu SVM_1 et le détecteur d'erreur : la décision du SVM_1 est corrigée et l'ensemble d'apprentissage mis à jour selon le détecteur,

  Trait pointillé classifieur bayésien idéal corrigé par le détecteur d'erreur.

5.2 Résultats de simulations

Nous avons fait plusieurs simulations pour voir l'influence des différents paramètres sur les pourcentages de mal classés du système. Les paramètres étudiés sont :

- l'influence de la taille de la fenêtre glissante,
- l'effet de la stratégie de mise à jour de la population d'apprentissage.

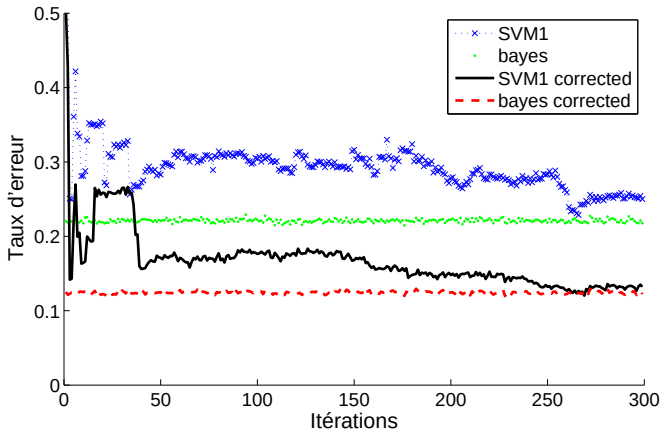


FIGURE 5.2 – Détail des différents tracés des pourcentages de mal classés du simulateur.

On se place dans le cas où trois classes sont à distinguer. Les paramètres sont testés sur trois profils d'évolution des classes. Les différentes évolutions de la position des classes sont représentées Figure 5.3 :

- les classes sont fixes, Figure 5.3a,
- les centres des classes se rapprochent, Figure 5.3b. Cette simulation correspond à un utilisateur qui a de plus en plus de difficulté à se concentrer et à changer d'état mental pour les différentes tâche à accomplir (fatigue, perte de concentration...),
- les centres des classes s'éloignent et se translatent, Figure 5.3c. L'éloignement des classes représente un utilisateur qui améliore sa capacité à contrôler le BCI alors que la dérive des classes peut s'expliquer par son changement de stratégie ou un problème sur le matériel (gel perdant de la conductivité, matériel qui chauffe...).

Pour toutes les simulations nous avons utilisé le même détecteur d'erreur. Ses performances proviennent des résultats obtenus avec le détecteur$_c$ au chapitre précédent. Les probabilités conditionnelles du détecteur correspondent aux valeurs moyennes obtenues sur les données expérimentales :

- $P_{\hat{E}|E}(juste,juste) = 88\%$,
- $P_{\hat{E}|E}(faux,faux) = 56\%$.

et le taux de réponse est de 100%.

Les simulations sont effectuées sur 300 itérations (une itération correspond à la génération d'un nouvel individu). La population initiale, dont l'étiquette des individus est connue, est n_{init}=10 (répartis sur toutes les classes). Pour chaque test, on effectue une série de 20 simulations et on trace la moyenne des pourcentages de mal classés.

5.2.1 Influence de la taille de la fenêtre glissante

Trois tailles différentes de fenêtres glissantes ont été étudiées, n_{w1}=10 (taille de la population initiale), n_{w2}=50 et n_{w3}=300 (aucun individu n'est retiré de la population d'apprentissage). Les tests ont été effectués pour les trois profils d'évolution des classes, mais les résultats sont présentés seulement pour des classes qui s'éloignent et se translatent. Excepté pour la fenêtre de taille n_{w1}, les résultats obtenus sur les deux autres profils d'évolution des classes sont similaires.

La Figure 5.4 présente les résultats obtenus pour les différentes tailles de fenêtres avec des classes qui s'éloignent et se translatent. La stratégie de mise à jour de la fenêtre glissante consiste :

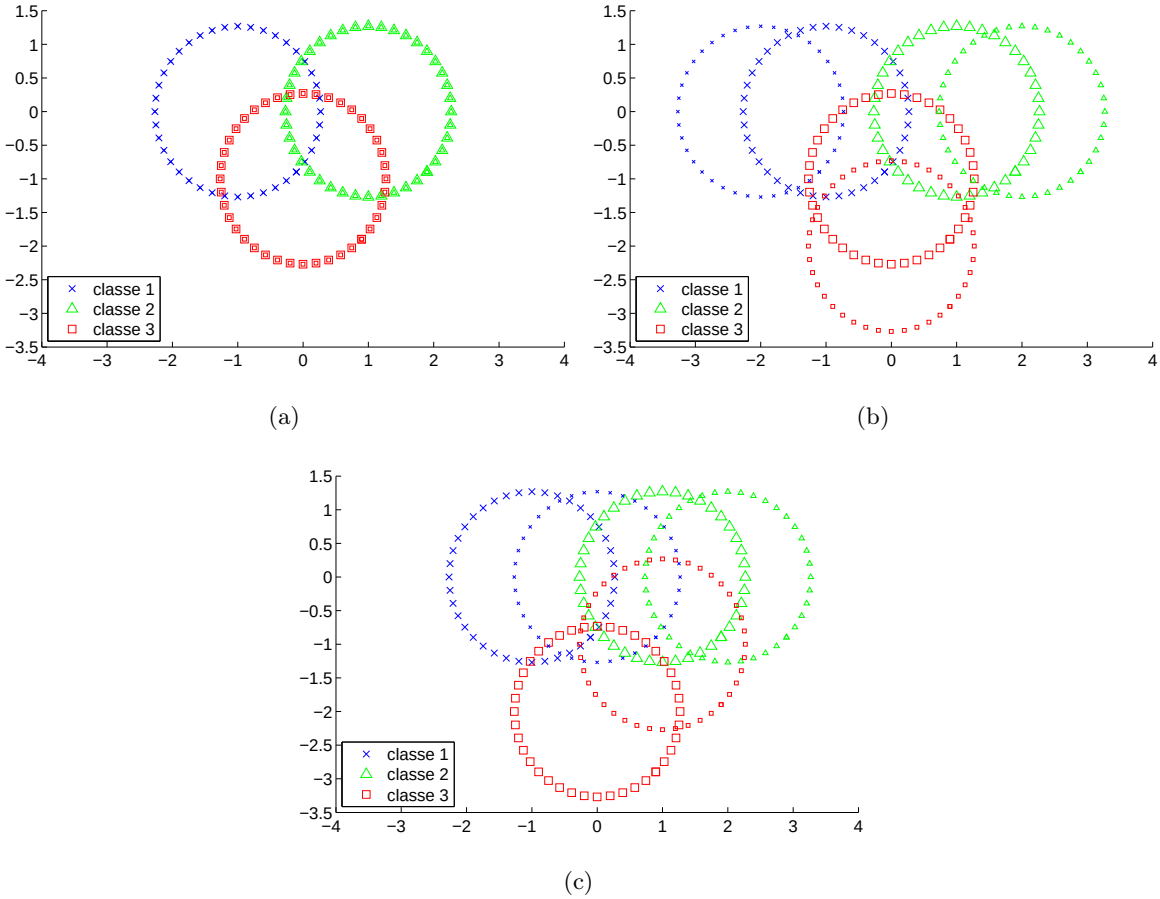


FIGURE 5.3 – Représentation de l'évolution de la répartition des classes pour trois scénarios différents. Les classes sont représentées par leur ellipse de confiance à 80% dans l'espace de représentation 2D. Les petits marqueurs correspondent à la position initiale et les marqueurs plus gros à la position finale des classes.

– à ajouter tous les individus estimés comme *juste*.
– pour chaque ajout, retirer l'individu le plus ancien.

Sur ces essais on voit l'importance du choix de la taille de la fenêtre glissante. Une fenêtre trop petite (Figure 5.4a) entraine une forte instabilité dans les résultats et en moyenne un taux d'erreur plus grand que pour les autres fenêtres. En effet chaque individu ajouté ou retiré de la population d'apprentissage a un impact important dans le calcul des séparatrices. De plus, une fenêtre trop petite possède l'inconvénient de mal caractériser les classes à discriminer car elles sont représentées par trop peu d'individus. Plus la population d'apprentissage est grande, moins un individu a d'influence sur le système de décision, mieux les classes sont représentées et moins l'intégration d'un élément mal étiqueté déstabilise le système.

Cependant, comme on le voit sur la Figure 5.4c, dans le cas où les classes bougent, une fenêtre trop grande ne permet pas de suivre leurs évolutions en particulier lorsqu'elles sont en translation. Pour cette évolution des classes, les points utilisés pour apprendre les séparatrices au début de la simulation, ne sont plus représentatifs de la position des classes à partir d'un moment (augmentation du pourcentage de mal classés autour de l'itération 150). C'est pourquoi il est important de pouvoir oublier des individus pour s'adapter aux changements des classes.

Pour nos simulations, le meilleur compromis entre la caractérisation des classes et le suivi

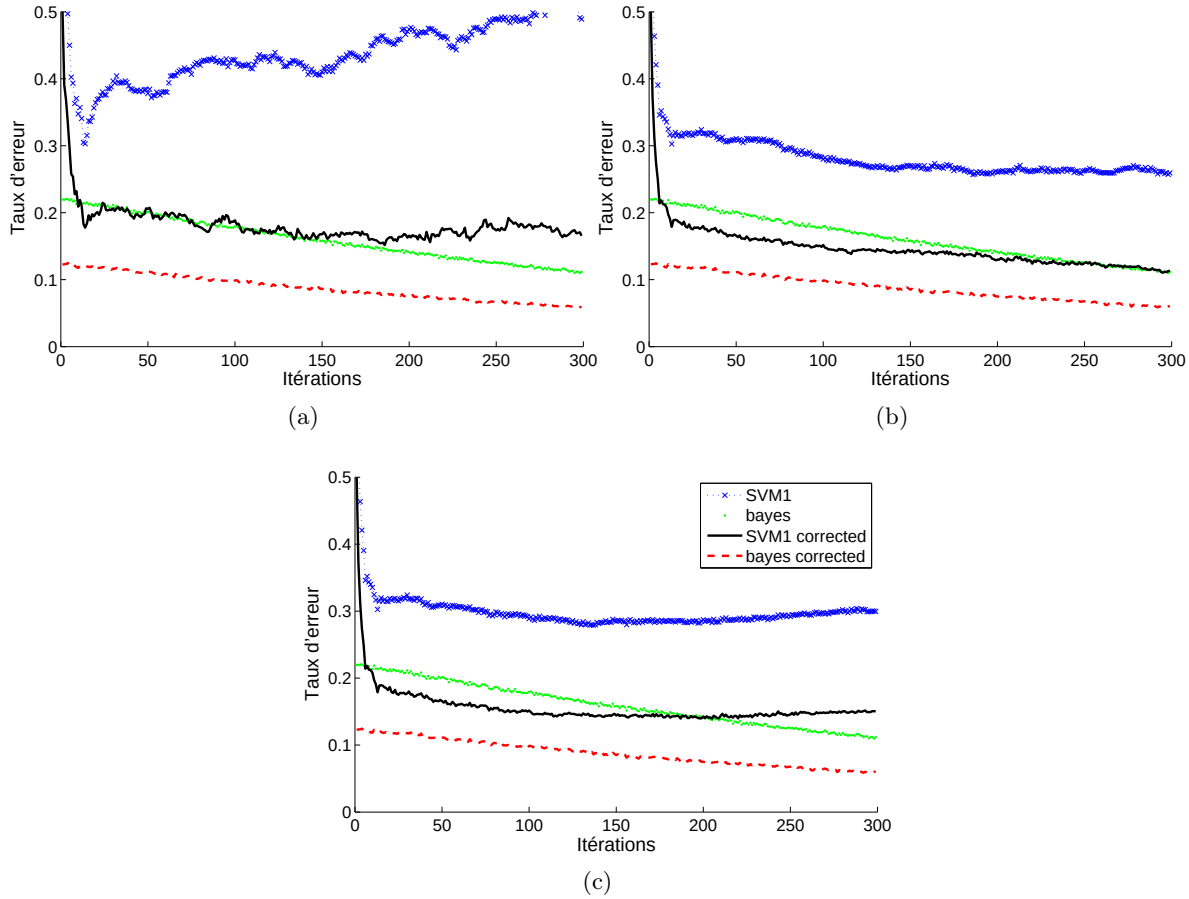


FIGURE 5.4 – Moyenne de 20 simulations de l'ensemble du système adaptatif avec des classes qui s'éloignent et translatent pour différentes tailles de fenêtres glissantes : les taux d'erreur du SVM_1 seul et du SVM_1 + détecteur sont comparés aux performances correspondantes du classifieur bayesien parfait. Les tailles des fenêtres glissantes sont : (a) n_{w1}=10, (b) n_{w2}=50, (c) n_{w3}=300.

de leur évolution est obtenu avec une fenêtre glissante de 50 individus, même si en moyenne on ne converge pas vers le pourcentage de mal classés obtenu avec le classifieur bayesien. Le choix de la taille de la fenêtre dépend fortement de la vitesse de déplacement des classes.

Lorsque les classes se translatent très vite, une petite fenêtre est plus adaptée pour suivre leurs évolutions.

5.2.2 Comparaison des stratégies de mise à jour

Deux stratégies de mise à jour de la population d'apprentissage ont été étudiées :
– prendre tous les exemples estimés *juste* par le détecteur,
– échantillonnage par incertitudes parmi les exemples estimés *juste* par le détecteur.

Ces deux stratégies ont été testées sur les trois profils d'évolution des classes. Les résultats sont présentés sur la Figure 5.5.

Les résultats obtenus pour des classes fixes ou qui se rapprochent sont très similaires avec les deux stratégies de mise à jour. On remarque que la stratégie consistant à utiliser tous les individus disponibles permet d'obtenir des résultats légèrement meilleurs en moyenne.

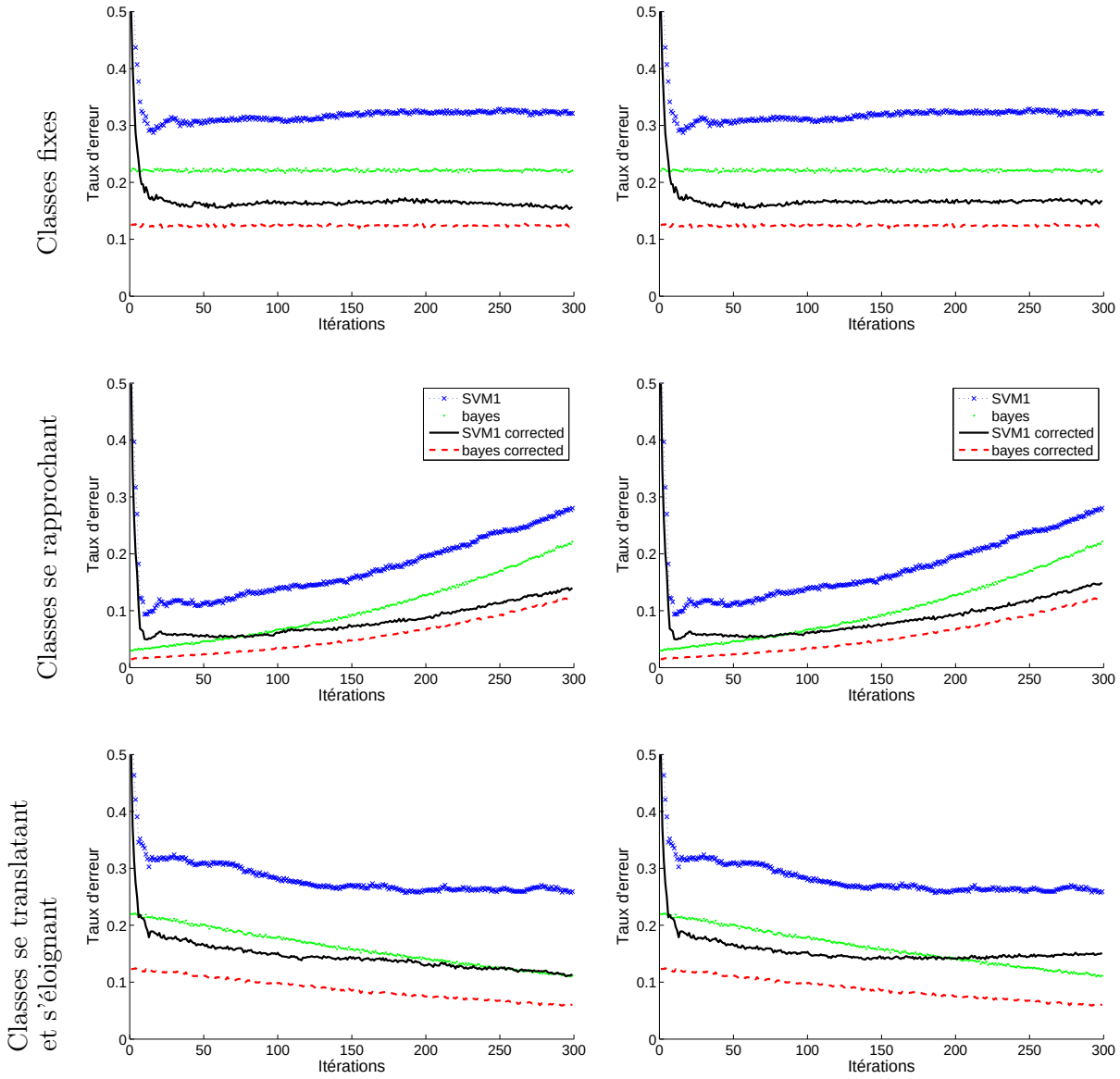


FIGURE 5.5 – Moyenne de 20 simulations de l'ensemble du système adaptatif avec trois profils d'évolution des classes : les taux d'erreur du SVM_1 seul et du SVM_1 + détecteur sont comparés aux performances correspondantes du classifieur bayesien parfait. Les stratégies utilisées pour mettre à jour l'ensemble d'apprentissages sont : (à gauche) prendre tous les exemples estimés comme *juste* par le détecteur, (à droite) appliquer la technique de l'échantillonnage par incertitudes (parmi les exemples qui sont estimés comme *juste* prendre seulement les exemples proches des séparatrices).

Pour les classes qui s'éloigne et se translatent, la stratégie d'échantillonnage par incertitude ne permet pas le suivi des classes. Comme pour le cas précédent où la fenêtre glissante était trop grande, on constate qu'il y a divergence entre le pourcentage de mal classés du SVM et celle du classifieur bayesien. La stratégie d'échantillonnage par incertitude, qui consiste à n'utiliser que les individus situés à l'intérieur la marge des SVM, a pour conséquence de diminuer cette dernière à chaque individu ajouté à l'ensemble d'apprentissage. En effet pour une fonction discriminante $f_i(x) = \langle w_i,x\rangle + b_i$, la taille de la marge est donnée par $2/\|w_i\|$ et est délimitée par les vecteurs

supports. Lors de l'ajout d'un individu à l'intérieur de la marge, il y a deux possibilités :
- l'individu ajouté devient vecteurs support : la marge diminue (car l'individu est à l'intérieur de la marge),
- l'individu ajouté est placé du mauvais côté de la marge, son coefficient $\alpha = C > 0$. Or $\|w_i\|^2 = \sum_j \alpha_j^2 \|x_j\|^2$ et l'ajout de l'individu augment cette norme, d'où la diminution de la taille de marge $(2/\|w_i\|)$.

Lorsque la taille de la marge devient trop petite, plus aucun individu n'est ajouté à l'ensemble d'apprentissage et le système ne s'adapte plus.

Une stratégie plus adaptée, est d'intégrer les individus x tels que $f_i(x) < d\|w_i\|$. Cette opération permet d'intégrer les exemples situés à une distance d de la séparatrice qui reste constante.

Cependant, même en utilisant cette stratégie, le suivi des classes n'est pas assuré : si les classes se sont trop translatées par rapport à la position des séparatrices, aucun nouvel individu n'est assez proche des séparatrices pour être ajouté à l'ensemble d'apprentissage et le système n'est plus mis à jour.

Finalement la meilleure stratégie, pour notre système, est l'intégration de tous les éléments estimés *juste* par le détecteur.

Sur la Figure 5.6 nous avons tracé l'évolution des pourcentages de mal classés pour deux simulations.

On constate que la stratégie intégrant tous les individus permet de converger vers la solution optimale alors que l'échantillonnage par incertitudes diverge. Sur ces simulations, en plus d'améliorer nettement le pourcentage de mal classés, le détecteur d'erreur permet d'avoir des résultats plus stables car moins d'individus mal étiquetés sont intégrés à l'ensemble d'apprentissage. Dans la deuxième simulation, on note un décrochage autour de l'itération 150 avant de converger de nouveau vers le pourcentage de mal classés optimal. Ces décrochages apparaissent à des endroits différents suivant les simulations, ce qui explique pourquoi en moyenne, le pourcentage de mal classés du SVM corrigé par le détecteur ne converge pas vers la solution du classifieur de bayes.

5.3 Conclusion

Dans ce chapitre nous avons étudié, en simulation, l'influence de différents paramètres sur la capacité du système de décision à s'améliorer et à suivre des classes qui se déplacent. On remarque que l'utilisation du détecteur, en plus de diminuer le pourcentage de mal classés, permet d'obtenir des résultats plus stables et plus proche du classifieur bayesien optimal. Les simulations des évolutions des classes n'ont pas pour but de représenter les évolutions réelles de l'état mental d'un utilisateur, elles permettent d'étudier l'influence des différents paramètres du système pour le suivi des classes. A travers les différentes simulations nous avons montré qu'il est important d'utiliser une taille de fenêtre glissante et une stratégie d'intégration des individus adaptées.

J'ai implémenté tout le système pour une utilisation dans des conditions réelles à l'Université de Göttingen. Les différents éléments du BCI sont les suivants :
- le système d'électrodes utilisé pour enregistrer les EEG est "actiCAP" de la société Brain Products,
- l'amplificateur est un "g.USBamp" de la société g.tec,
- la machine sur laquelle est implémentée la partie d'acquisition et de traitement du signal du BCI est un ordinateur portable doté d'un processeur Core i5 cadencé à 2.30GHz et de 4Go de RAM,

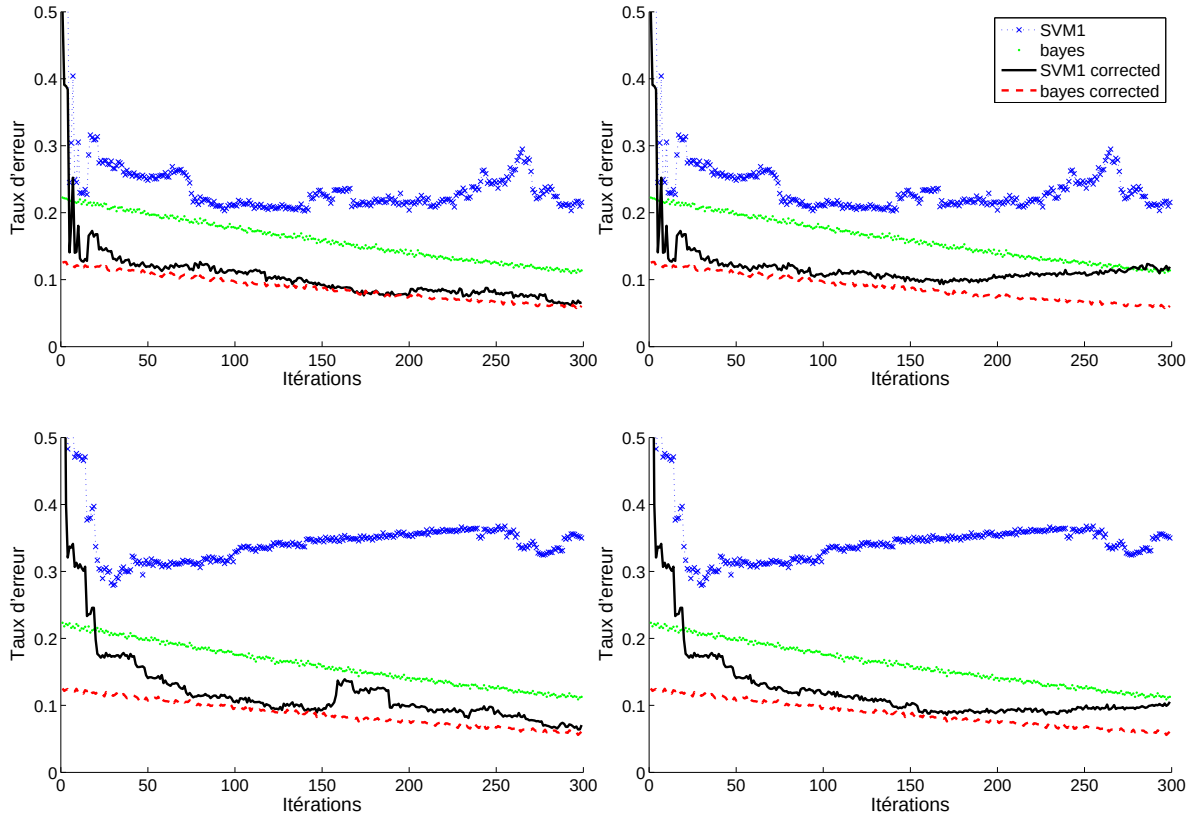


FIGURE 5.6 – Deux simulations de l'ensemble du système adaptatif avec des classes qui s'éloignent et se translatent : Les stratégies utilisées pour mettre à jour l'ensemble d'apprentissages sont : (à gauche) prendre tous les exemples estimés comme *juste* par le détecteur, (à droite) appliquer la technique de l'échantillonnage par incertitudes (parmi les exemples qui sont estimés comme *juste* prendre seulement les exemples proches des séparatrices). On observe sur ces 2 exemples que la première stratégie permet de mieux suivre l'évolution des classes, même après des décrochements temporaires.

- l'acquisition des signaux est réalisée grâce à la boîte à outils "Data Acquisition" de Matlab,
- l'interface visuelle pour l'utilisateur utilise le système de "GUI" de Matlab,
- toute la partie traitement du signal est réalisée en Matlab.

L'interface fonctionne en temps réel et des tests sur des sujets sont en cours de réalisation.

Conclusion et perspectives

Conclusion

L'utilisation des potentiels d'erreur pour améliorer les performances d'une interface cerveau-machine est récente. L'apport de cette thèse par rapport aux travaux antérieurs porte sur les points suivants :

- c'est la première fois que les potentiels d'erreur sont utilisés dans un système destiné à faire des distinctions complexes entre les tâches : il s'agit ici non pas de distinguer un mouvement du bras droit ou gauche, mais différentes modalités d'un même mouvement : lent/rapide, couple faible/fort,
- une méthode d'optimisation de l'espace de représentation des signaux multivoies a été proposée,
- deux approches pour la reconnaissance des potentiels d'erreur ont été comparées : une approche détection et une approche classification,
- l'étude théorique des performances du système corrigé par le détecteur peut être utilisée pour la phase de design de tout BCI où des potentiels d'erreur peuvent être enregistrés,
- un simulateur du système corrigé et bouclé par la réponse du détecteur a permis d'observer le comportement du système global et de comparer différentes stratégies de mise à jour de l'ensemble d'apprentissage,
- le système complet a été implémenté à Göttingen et il fonctionne en ligne.

La précision du système de décision principal est une composante importante du fonctionnement d'un tel système et le choix de l'espace de représentation en constitue l'élément principal. L'optimisation de cet espace de représentation permet d'augmenter significativement les résultats de la classification. Les méthodes à base d'ondelettes (optimisation de l'ondelette mère de la transformée discrète en ondelettes (DWT) ou de la base de décomposition de la transformée discrète en paquets d'ondelettes (DWPT)) ayant fait leur preuve dans des études antérieures pour classer les signaux de MRCPs, nous nous sommes inspirés de ces méthodes. Sur les signaux simulés nous avons montré que les deux méthodes permettaient d'améliorer les performances de la classification. Cependant sur des signaux réels, l'optimisation de la base de la DWPT détériore les résultats contrairement à l'optimisation de l'ondelette mère de la DWT. Nous avons donc décidé d'étendre l'optimisation de l'ondelette mère de la DWT à la recherche d'une ondelette mère optimale par voie. La méthode était utilisée seulement pour des signaux monovoies ou avec une ondelette mère optimale identique pour toutes les voies dans le cas multivoies. Cette extension n'a pu se faire qu'en proposant un critère de qualité (Fisher) différent du critère classique de probabilité d'erreur, et un algorithme sous-optimal, conduisant à un temps de calcul compatible avec un fonctionnement en ligne.

Pour corriger le BCI, il est nécessaire de pouvoir détecter ses erreurs. Nous avons comparé deux méthodes permettant de détecter les potentiels d'erreur générés chez le sujet lorsque le BCI affiche une mauvaise réponse. La première méthode est inspirée du filtrage adapté et est basée

sur l'utilisation de deux seuils pour permettre de limiter le nombre de fausses détections. La seconde approche est une méthode de classification classique. La comparaison des pourcentages de mal classés des détecteurs seuls ne permettant pas de déterminer la meilleure méthode, les améliorations théoriques apportées au système corrigé par les deux approches ont été calculées. Cette amélioration est exprimée en termes de probabilité d'erreur, taux de répétition et taux de transfert. Ceci nous a permis de conclure que les meilleurs résultats sont obtenus avec l'approche classification.

En plus de l'amélioration du taux de transfert du BCI corrigé, le détecteur permet d'augmenter la confiance de l'étiquette attribuée à un nouvel exemple lorsqu'il est estimé comme *juste*. Ainsi, on peut enrichir l'ensemble d'apprentissage en limitant les erreurs sur les étiquettes des éléments ajoutés (condition nécessaire pour ne pas détériorer les performances du système). Un classifieur qui se met à jour rapidement est indispensable pour une application en temps réel. Nous avons étudié et implémenté deux algorithmes de SVM fonctionnant par incrémentation et décrémentation, évitant de recalculer entièrement les fonctions de décision lors de l'ajout ou du retrait d'un nouvel individu dans l'ensemble d'apprentissage.

La dernière partie du travail de thèse a permis de tester la capacité du système complet à s'adapter aux évolutions de l'utilisateur. Dans les conditions réelles, on ne dispose que d'un nombre très limité d'essais pour estimer à chaque instant les performances du BCI. De plus, la mise au point du système nécessite de tester différentes stratégies et paramètres de mise à jour de l'apprentissage ; ce qui conduit à multiplier des expériences éprouvantes pour l'utilisateur. C'est pourquoi nous avons développé un simulateur permettant d'étudier le taux de mal classés du BCI corrigé pour différentes évolutions des classes à discriminer et stratégies de mises à jour de l'ensemble d'apprentissage. Nous avons montré que l'utilisation d'une fenêtre glissante, avec une taille et une stratégie d'ajout et de retrait des individus adaptées, permet d'améliorer les performances du système comparé à l'intégration de de la totalité des individus estimés comme *justes* sans en oublier.

Finalement, le système complet a été implémenté pour une utilisation en conditions réelles à l'Université de Göttingen.

Perspectives

Même si le système complet est implémenté et fonctionne dans des conditions réelles, nous n'avons pas de moyen pour savoir s'il s'améliore ou se détériore en cours d'utilisation. Une étape importante avant de pouvoir valider le BCI lors d'une utilisation en ligne est de trouver un critère permettant de quantifier l'évolution de ses performances au cours du temps. Une solution possible serait d'estimer, à chaque itération, le pourcentage de mal classés sur la fenêtre glissante utilisée pour l'apprentissage des fonctions de décision. Cette estimation peut se faire en utilisant une procédure de leave-one-out grâce au fonctionnement par incrémentation/décrémentation des SVM.

L'algorithme que nous avons proposé pour la recherche de la meilleure ondelette mère de la DWT est sous-optimal. Le choix de cet algorithme est un domaine de recherche en soi pour lequel il existe de nombreux travaux. Une amélioration possible serait d'implémenter et tester une de ces techniques (algorithme génétique, méthode de Monte Carlo...).

La recherche de la meilleure base de décomposition en paquet d'ondelette est inefficace sur les signaux EEG. Une explication est que cette méthode ne permet pas de limiter le nombre de descripteurs et conduit à un espace de représentation trop grand. L'implémentation de méthodes de réduction de dimension telle que le PCA ou le choix d'un nombre limité des meilleurs

descripteurs sont des améliorations possibles pour une utilisation de la méthode sur des données réelles.

Actuellement la recherche de la meilleure ondelette est faite uniquement sur la population d'apprentissage initiale. Il pourrait être intéressant d'essayer d'optimiser l'espace de représentation à chaque ajout ou retrait d'un individu tout en conservant la même procédure de mise à jour des fonctions de décision.

Annexe A

Filtrage spatial

A.1 Notations

On note :
- $\mathcal{X}$ ensemble des signaux d'apprentissage.
- $n = \text{card}\{\mathcal{X}\}$ nombre de signaux d'apprentissage.
- n_ω nombre de classes.
- $\mathcal{X}_i$ ensemble des signaux de la classe i et $\mathcal{X} = \bigcup_{i=1}^{n_\omega} \text{card}\{\mathcal{X}_i\}$
- $n_i = \text{card}\{\mathcal{X}_i\}$ nombre de signaux de la classe i.
- $\overline{x_i} = \frac{1}{n_i} \sum_j x_{i,j}$ moyenne des signaux de la classe i.

A.2 filtrage spatial

Le filtrage spatial est une méthode consistant à transformer un signal multi-voies en signal mono-voies par combinaison linéaire.

Soit x un signal multi-voies à N_v voies et N_p échantillons, et x_f le signal mono-voie après filtrage. On définit l'application F_f telle que :

$$\begin{array}{ccc} x & \longmapsto & x_f = F_f(x) = fx \\ N_v \times N_p & \longrightarrow & 1 \times N_p \end{array} \tag{A.1}$$

f correspond au vecteur de pondération des canaux de chaque signal. Différentes techniques permettent de déterminer f de façon à obtenir le signal mono-voies x_f le plus discriminant possible au sens d'un critère préalablement défini.

Nous avons étudié trois méthodes différentes pour obtenir le vecteur de pondération f dans le cadre de cette thèse :

Moyenne des voies : Ce filtrage consiste à faire la moyenne des échantillons des différentes voies :

$$f = \frac{1}{N_v}[1, \ldots, 1] \tag{A.2}$$

Il permet de prendre en compte de manière équivalente les informations des différentes voies. On suppose ici que toutes les voies sont aussi discriminantes.

Common Spatial Pattern (CSP) : On cherche f_{CSP} qui maximise le critère suivant ([Boudet *et al.*, 2007]) :

$$J_1 = \frac{fC_2f'}{fC_1f'} \tag{A.3}$$

avec C_i la moyenne des matrices de covariance $C_{i,j} = x_{i,j}x'_{i,j}$ des essais de la classe i.
Maximiser ce critère revient à maximiser la variance des signaux de la classe 2 tout en minimisant celle de la classe 1.
Ce filtrage semble bien adapté à détection des potentiels d'erreur puisqu'il est utilisé principalement pour séparer une classe qui contient des informations (dans notre cas la classe *fausse*) et une classe supposée ne contenir que du bruit (la classe *juste*).

Optimisation du critère de Fisher : Un autre filtre spatial, proposé par Hoffmann *et al.* [2006], consiste à maximiser le critère suivant :

$$J_2 = \frac{fS_bf'}{fS_wf} \tag{A.4}$$

avec
- $S_b = \sum_{i=1}^{n_\omega} \frac{n_i}{n}(\overline{x}_i - \overline{x})(\overline{x}_i - \overline{x})'$
- $S_w = \sum_{i=1}^{n_\omega} \frac{n_i}{n} \sum_{j=1}^{n_i}(x_{i,j} - \overline{x}_i)(x_{i,j} - \overline{x}_i)'$

Ce critère peut-être interprété comme la mesure de la séparation entre deux classes. Maximiser J_2 consiste à maximiser l'écart des moyennes inter-classes des signaux tout en minimisant l'inertie intra-classe des signaux.

Les critères J_1 et J_2 sont des quotients de Rayleigh généralisés ($R(A,B;x) = \frac{x'Ax}{x'Bx}$ avec x vecteur colonne).

Il existe une méthode analytique pour trouver un optimum : on diagonalise $B^{-1}A$. Soit λ_M la plus grande valeur propre de $B^{-1}A$; elle est associée au vecteur propre x_M. Le vecteur f_{max} recherché correspond au vecteur x'_M.

Liste des publications

Revues

X. Artusi, I-K Niazi, M. Lucas, et D. Farina. Performance of a simulated adaptive BCI based on experimental classification of movement-related and error potentials. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems, special issue on Brain-Machine Interfaces*, à paraître, 2012.

D. Vautrin, X. Artusi, M. Lucas, and D. Farina. A novel criterion of wavelet packet best basis selection for signal classification with application to brain-computer interfaces. *IEEE transactions on bio-medical engineering*, pages 2734-2738, 2009.

Conférences internationales avec comité de lecture et publication des actes

X. Artusi, I-K Niazi, M. Lucas, et D. Farina. Accuracy of a BCI based on movement-related and error potentials. *33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011.

X. Artusi, I-K Niazi, M. Lucas, et D. Farina. Theoretical framework and simulation of an adaptive BCI based on movement-related and error potentials. *Graz, 5th International Brain-Computer Interface Conference*, 2011.

Conférences nationales avec comité de lecture et publication des actes

D. Vautrin, X. Artusi, M.F. Lucas, et D. Farina. Un critère de sélection de meilleure base d'ondelettes pour la classification, application aux interfaces cerveau-machines. *Colloque GRETSI*, 2009.

X. Artusi, I-K Niazi, M.F. Lucas, et D. Farina. Simulation of an adaptive BCI based on movement-related and error potentials. *Colloque GRETSI*, 2011.

Conférences sans publication des actes

X. Artusi. Interface cerveau-machine avec adaptation à l'utilisateur. *JDOC, Journée des Doctorants de l'école centrale*, 2010.

X. Artusi, M.-F. Lucas, et D. Farina. Identification of incorrect decisions of brain-computer interfaces from cortical potentials. *ISEK, International Society of Electrophysiology and Kinesiology*, 2010.

Bibliographie

P. Abry : *Ondelettes et turbulence – multiresolutions, algorithmes de décomposition, invariance d'échelle et signaux de pression.* Diderot, Editeur des Sciences et des Arts, Paris, 1997.

B. Blankertz, G. Dornhege, M. Krauledat, K. Müller et G. Curio : The non-invasive Berlin brain-computer interface : Fast acquisition of effective performance in untrained subjects. *NeuroImage*, 37(2):539–550, 2007.

B. Blankertz, C. Schäfer, G. Dornhege et G. Curio : Single trial detection of EEG error potentials : A tool for increasing BCI transmission rates. *Artificial Neural Networks-ICANN 2002*, 2415:1137–1143, 2002.

S. Boudet, L. Peyrodie, P. Gallois et C. Vasseur : Mise en évidence des potentiels évoqués par CSP pour le dispositif d'interface P300 Speller. *In XXIe colloque GRETSI.* GRETSI, Groupe d'Études du Traitement du Signal et des Images, 2007.

C. Burrus, R. Gopinath, H. Guo, J. Odegard et I. Selesnick : *Introduction to wavelets and wavelet transforms : a primer.* Prentice Hall Upper Saddle River, NJ, 1997.

A. Buttfield, P. Ferrez et J. Millán : Towards a robust BCI : error potentials and online learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):164–168, 2006.

G. Cauwenberghs et T. Poggio : Incremental and decremental support vector machine learning. *In Advances in neural information processing systems 13*, p. 409–415. The MIT Press, 2001.

C. Chang et C. Lin : LIBSVM : A library for support vector machines. *ACM Transaction on Intelligence Systems and Technology*, 2:1–27, May 2011.

R. Chavarriaga et J. Millán : Learning from EEG error-related potentials in noninvasive brain-computer Interfaces. *IEEE Transactions on Rehabilitation Engineering*, 18(4):381–388, 2010.

R. Coifman et M. Wickerhauser : Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2 Part 2):713–718, 1992.

K. Crammer et Y. Singer : On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

B. Dal Seno, M. Matteucci et L. Mainardi : Online Detection of P300 and Error Potentials in a BCI Speller. *Computational Intelligence and Neuroscience*, 2010:1–5, 2010.

I. Daubechies : *Ten lectures on wavelets.* Society for industrial and applied mathematics, 2006.

R. Duda, P. Hart et D. Stork : *Pattern classification*, vol. 2. Wiley, 2001.

T. Evgeniou, M. Pontil et T. Poggio : Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–53, 2000.

M. Falkenstein, J. Hohnsbein, J. Hoormann et L. Blanke : Error processing in choice reaction tasks. *Electroencephalography and Clinical Neurophysiology*, 78(6):447–455, 1991.

M. Falkenstein, J. Hoormann, S. Christ et J. Hohnsbein : ERP components on reaction errors and their functional significance : a tutorial. *Biological Psychology*, 51(2-3):87–107, 2000. ISSN 0301-0511.

D. Farina, O. Nascimento, M. Lucas et C. Doncarli : Optimization of wavelets for classification of movement-related cortical potentials generated by variation of force-related parameters. *Journal of neuroscience methods*, 162(1-2):357–363, 2007.

L. A. Farwell et E. Donchin : Talking off the top of your head : Toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70:510–523, 1988.

P. Ferrez et J. Millán : You are wrong !-automatic detection of interaction errors from brain waves. *In Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005.

P. Ferrez et J. Millán : Error-related EEG potentials generated during simulated brain-computer interaction. *IEEE transactions on biomedical engineering*, 55(3):923–929, 2008.

P. Ferrez et J. Millán : Simultaneous real-time detection of motor imaginary and error related potentials for improved BCI accuracy. *In Proceedings 4th International Brain-Computer Interface workshop and Training Course*, p. 197–202, Graz 2008.

M. Ferris et T. Munson : Interior-point methods for massive support vector machines. *SIAM Journal on Optimization*, 13:783–804, August 2002.

S. Fine et K. Scheinberg : Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2002.

X. Gao, D. Xu, M. Cheng et S. Gao : A BCI-based environmental controller for the motion-disabled. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):137–140, 2003.

W. Gehring, B. Goss, M. Coles, D. Meyer et E. Donchin : A neural system for error detection and compensation. *Psychological Science*, 4:385–390, 1993.

E. M. Gertz et S. Wright : OOQP User Guide. Rap. tech., Argonne National Laboratory, 2001.

U. Hoffmann, J. Vesin et T. Ebrahimi : Spatial filters for the classification of event-related potentials. *In European Symposium on Artificial Neural Networks Bruges (Belgium)*, 2006.

A. Ikeda et H. Shibasaki : Invasive recording of movement-related cortical potentials in humans. *Journal of Clinical Neurophysiology*, 4:509–520, 1992.

I. Jentzsch et H. Leuthold : Advance movement preparation of eye, foot, and hand : a comparative study using movement-related brain potentials. *Cognitive Brain Research*, 4 (2):201–217, 2002.

N. Karmarkar : A new polynomial-time algorithm for linear programming. *In Proceedings of the sixteenth annual ACM symposium on Theory of computing*, p. 302–311, 1984.

W. Lawton : Necessary and sufficient conditions for constructing orthonormal wavelet bases. *journal of Mathematical Physics*, 32:57–61, 1991.

M. Lucas, A. Gaufriau, S. Pascual, C. Doncarli et D. Farina : Multi-channel surface EMG classification using support vector machines and signal-based wavelet optimization. *Biomedical Signal Processing and Control*, 3(2):169–174, 2008.

A. Maitrot : *Optimisation d'ondelettes pour la classification supervisée de signaux. Application aux électromyogrammes.* Thèse de doctorat, Ecole Centrale de Nantes, 2005.

A. Maitrot, M. Lucas, C. Doncarli et D. Farina : Signal-dependent wavelets for electromyogram classification. *Medical and Biological Engineering and Computing*, 43:487–492, 2005a.

A. Maitrot, M. Lucas et C. Doncarli : Design of wavelets adapted to signals and application. *In IEEE International Conference on Acoustics, Speech, and Signal Processing.*, vol. 4, 2005b.

S. Mallat : *A wavelet tour of signal processing.* Academic Pr, 1999.

Y. Meyer : *Ondelettes et opérateurs.* Hermann, 1990.

O. Nascimento et D. Farina : Movement-related cortical potentials allow discrimination of rate of torque development in imaginary isometric plantar flexion. *IEEE Transactions on Biomedical Engineering*, 55(11):2675–2678, November 2008.

O. Nascimento, K. Nielsen et M. Voigt : Movement-related parameters modulate cortical activity during imaginary isometric plantar-flexions. *Experimental Brain Research*, 171:78–90, 2006. ISSN 0014–4819.

B. Obermeier, C. Guger, C. Neuper et G. Pfurtsheller : Hidden markov models for online classification of single trial EEG. *Pattern recognition letters*, 22:1299–1309, 2001.

E. Osuna, R. Freund et F. Girosi : An improved training algorithm for support vector machines. *In Neural Networks for Signal Processing VII.*, p. 276–285. IEEE, 1997.

G. Pfurtscheller, C. Neuper, C. Guger, W. Harkam, H. Ramoser, A. Schlogl, B. Obermaier, M. Pregenzer *et al.* : Current trends in Graz brain-computer interface (BCI) research. *IEEE Transactions on Rehabilitation Engineering*, 8(2):216–219, 2000.

J. Platt : Sequential minimal optimization : A fast algorithm for training support vector machines. *Advances in Kernel Methods-Support Vector Learning*, 208:1–21, 1999.

A. Rakotomamonjy, V. Guigue, G. Mallet et V. Alvarado : Ensemble of SVMs for improving brain computer interface P300 speller performances. *In Artificial Neural Networks : Biological Inspirations–ICANN 2005*, p. 45–50. Springer, 2005.

D. Romero, M. Lacourse, K. Lawrence, S. Schandler et M. Cohen : Event-related potentials as a function of movement parameter variations during motor imagery and isometric action. *Behavioural Brain Research*, 117(1-2):83–96, 2000.

N. Saito et R. Coifman : Local discriminant bases. *In proc. SPIE*, vol. 2303, p. 2–14, 1994.

A. Sano et H. Bakardjian : Movement-related cortical evoked potentials using four-limb imagery. *International Journal of Neuroscience*, 119(5):639–663, 2009.

G. Schalk, J. Wolpaw, D. McFarland et G. Pfurtscheller : EEG-based communication : presence of an error potential. *Clinical Neurophysiology*, 111(12):2138–2144, 2000.

K. Scheinberg : An efficient implementation of an active set method for SVMs. *Journal of Machine Learning Research*, 7:2237–2257, 2006.

G. Schohn et D. Cohn : Less is more : Active learning with support vector machines. *In In Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

I. Selesnick : Maple and the parameterization of orthogonal wavelet bases. Rap. tech., Rice University, 1997.

H. Shibasaki, G. Barrett, E. Halliday et A. Halliday : Cortical potentials associated with voluntary foot movement in man. *Electroencephalography and Clinical Neurophysiology*, 52(6):507–516, 1981.

G. Townsend, B. Graimann et G. Pfurtscheller : Continuous EEG classification during motor imagery-simulation of an asynchronous BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 12(2):258–265, june 2004.

P. Vaidyanathan et P. Hoang : Lattice structures for optimal design and robust implementation of two-channel perfect-reconstruction QMF banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(1):81–94, 1988.

R. J. Vanderbei : LOQO : An interior point code for quadratic programming. Rap. tech., Optimization Methods and Software, 1998.

V. Vapnik : *Estimation of dependences based on empirical data : Springer series in statistics*. Springer-Verlag New York, Inc., 1982.

V. N. Vapnik : *Statistical learning theory*. John Wiley & Sons, New York, 1995.

D. Vautrin : Classification de signaux EEG pour les interfaces cerveau-machine : Recherche de meilleures bases d'ondelettes. Mémoire de D.E.A., Ecole Centrale de Nantes, 2008.

D. Vautrin, X. Artusi, M. Lucas et D. Farina : A novel criterion of wavelet packet best basis selection for signal classification with application to brain-computer interfaces. *IEEE transactions on bio-medical engineering*, 56:2734–2738, 2009.

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

S. Vishwanathan, A. Smola et M. Murty : SimpleSVM. *In International Conference on Machine Learning*, vol. 20, p. 760–767, 2003.

J. Wolpaw et D. McFarland : Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences*, 101 (51):17849–17854, 2004.

L. Yuanqing et G. Cuntai : An extended EM algorithm for joint feature extraction and classification in brain-computer interfaces. *Neural Computation*, 18(11):2730–2761, 2006.

Résumé

Nous nous intéressons ici à une interface cerveau-machine (BCI, Brain Computer Interface) permettant de commander une prothèse par la pensée. Le rôle du BCI est de décoder à partir de signaux électroencéphalographiques (EEG) le mouvement désiré par le sujet. Le cœur du BCI est un algorithme de classification caractérisé par le choix des descripteurs des signaux et des règles de décision. L'objet de cette thèse est de développer un système BCI précis, capable d'améliorer ses performances en cours d'utilisation et de s'adapter à l'utilisateur sans nécessiter de multiples sessions d'apprentissage. Nous combinons deux moyens pour y parvenir. Le premier consiste à augmenter la précision du système de décision en recherchant des descripteurs pertinents vis à vis de l'objectif de classification. Le second est d'inclure un retour de l'utilisateur sur le système de décision : l'idée est d'estimer l'erreur du BCI à partir de potentiels cérébraux évoqués, reflétant l'état émotionnel du patient corrélé au succès ou à l'échec de la décision prise par le BCI, et de corriger le système de décision du BCI en conséquence.

Les principales contributions de la thèse sont les suivantes : nous avons proposé une méthode d'optimisation de descripteurs à bases d'ondelettes pour des signaux EEG multivoies ; nous avons quantifié théoriquement l'amélioration des performances apportée par le détecteur ; un simulateur du système corrigé et bouclé a été développé pour observer le comportement du système global et comparer différentes stratégies de mise à jour de l'ensemble d'apprentissage ; le système complet a été implémenté et fonctionne en ligne dans des conditions réelles.

Mots clés

Interface cerveau machine, potentiel d'erreur, optimisation d'ondelettes, machine à vecteurs support.

Title

Brain Computer Interface with automatic adaptation to the user.

Abstract

We study a brain computer interface (BCI) to control a prosthesis with thought. The aim of the BCI is to decode the movement desired by the subject from electroencephalographic (EEG) signals. The core of the BCI is a classification algorithm characterized by the choice of signals descriptors and decision rules. The purpose of this thesis is to develop an accurate BCI system, able to improve its performance during its use and to adapt to the user evolutions without requiring multiple learning sessions. We combine two ways to achieve this. The first one is to increase the precision of the decision system by looking for relevant descriptors for the classification. The second one is to include a feedback to the user on the system decision : the idea is to estimate the error of the BCI from evoked brain potentials, reflecting the emotional state of the patient correlated to the success or failure of the decision taken by the BCI, and to correct the decision system of the BCI accordingly.

The main contributions are : we have proposed a method to optimize the feature space based on wavelets for multi-channel EEG signals ; we quantified theoretically the performances of the complete system improved by the detector ; a simulator of the corrected and looped system has been developed to observe the behavior of the overall system and to compare different strategies to update the learning set ; the complete system has been implemented and works online in real conditions.

Keywords

Brain computer interface, error potential, wavelet optimization, support vector machine.

Discipline

Informatique, automatique, électronique et génie électrique.