# Innovative decoding algorithms for Chronic ECoG-based Brain Computer Interface (BCI) for motor disabled subjects in laboratory and at home

Alexandre Moly

# THÈSE

Pour obtenir le grade de

## DOCTEUR DE L'UNIVERSITE GRENOBLE ALPES

Spécialité : MBS - Modèles, méthodes et algorithmes en biologie, santé et environnement

Arrêté ministériel : 25 mai 2016

Présentée par


## Alexandre / MOLY

Thèse dirigée par **Tetiana AKSENOVA**
Encadrant CEA : par **Guillaume CHARVET**

préparée au sein du **Laboratoire CLINATEC**
dans **l'École Doctorale Ingénierie pour la Santé la Cognition et l'Environnement**


# Une interface cerveau-machine basée sur des algorithmes de décodage innovants pour le contrôle d'effecteurs complexes en vue d'un usage au quotidien par des patients en situation de handicap moteur

# Innovative decoding algorithms for Chronic ECoG-based Brain Computer Interface (BCI) for motor disabled subjects in laboratory and at home

Thèse soutenue publiquement le **10 décembre 2020**,
devant le jury composé de :

**Madame, Tetiana, AKSENOVA**
INGENIEUR HDR, CEA GRENOBLE, Directrice de thèse
**Monsieur Saïd MOUSSAOUI**
PROFESSEUR, ECOLE CENTRALE NANTES, Rapporteur
**Monsieur, Fabien, LOTTE**
DIRECTEUR DE RECHERCHE, INRIA CENTRE BORDEAUX - SUDOUEST, Rapporteur
**Monsieur, François, CABESTAING**
PROFESSEUR DES UNIVERSITES, UNIVERSITE DE LILLE, Président
**Monsieur Solaiman SHOKUR**
DOCTEUR EN SCIENCES, ECOLE POLYTECH FEDERALE LAUSANNE SUISSE, Examinateur
**Monsieur, Alim Louis, BENABID**
PROFESSEUR EMERITE, Université Grenoble Alpes, Examinateur

# Acknowledgements

Premièrement, je tiens sincèrement à remercier le Professeur François CABESTAING, le Professeur Alim Louis BENABID et Monsieur Solaiman SHOKUR pour avoir accepté de faire partie des membres du jury et de participer à l'évaluation de mes travaux de recherche. Plus particulièrement, j'aimerai faire part de ma gratitude la plus sincère au Docteur Fabien LOTTE et au Professeur Saïd MOUSSAOUI pour avoir endossé les rôles de rapporteurs de mon manuscrit de thèse.

Je tiens ensuite à remercier de tout mon cœur le Docteur Tetiana AKSENOVA pour tous ses conseils qui ont permis de mener à terme les recherches de cette thèse et l'écriture du manuscrit. Je n'aurais pas pu réaliser tout ce travail sans sa patience, sa gentillesse et sa disponibilité durant les trois dernières années. Merci de m'avoir expliqué plusieurs dizaines de fois les mêmes notions mathématiques sans perdre votre calme. Merci également de votre efficacité pour la correction du manuscrit de thèse. Cette dernière année aura été un challenge que nous avons bien géré.

Merci à Guillaume CHARVET pour toutes ses astuces et stratégies pour améliorer mes présentations et toujours voir les plus petits détails qui échappent à tout le monde. Grâce à tes conseils je maitrîse toutes les arcanes de Power Point. Merci également à l'ensemble des personnes travaillant à CLINATEC avec qui j'aurai passé trois superbes années et qui m'auront soutenu tout au long de la thèse.

Je tiens bien évidemment à remercier toute ma famille, ma mère, mon père, ma sœur, mon beau-frère, et mes grands parents. A ma mère et mon père, merci de m'avoir toujours aidé dans tous mes problèmes et pour votre réconfort moral qui a été permanent durant ces trois longues années. Je n'aurais jamais pu faire cette thèse sans vous. A ma sœur Agathe, son futur mari Arnaud, merci pour votre bonne humeur permanante. Vous retrouvez été toujours un immense moment de bonheur et j'espère avoir plus de temps pour venir vous voir (je n'oublie pas Odin que je remercie aussi). Malgré tous les problèmes de cette fin d'année, je vous souhaite tous mes vœux de bonheur et il me tarde de vous retrouver au plus vite pour célébrer votre union.

A partir d'ici commence la longue liste des personnes que je dois remercier pour m'avoir soutenu d'une façon ou d'une autre durant cette longue aventure. Merci Julie et Félix, j'ai passé beaucoup de temps avec vous, nous avons beaucoup ri durant ces trois années et je me souviendrai toujours de toutes nos soirées jeux et de nos lundis films.

Antoine et Maud, merci pour votre bonne humeur et les nombreuses discussions que nous avons eu (merci aussi pour l'aide au déménagement, il me tarde de venir voir votre belle maison).

Vincent, nous avons passé beaucoup de temps à discuter, que ce soit pour le travail ou dans tout autre domaine, nos conversations ont toujours été passionantes et ton humour m'a toujours remonté le moral (nous continuerons de parler manga, jeux vidéo, etc.).

Tamara, merci pour tous ces bons moments, ma deuxième année aurait été bien différente sans toi (il me tarde de pouvoir râler et critiquer encore avec toi). Courage pour la thèse, je sais que tu feras du très bon travail comme toujours.

Lucas, merci d'avoir été là durant cette deuxième année de thèse, nous aurons bien ri et parlé de mangas. Nous nous ressemblons beaucoup, il est très pratique d'avoir quelqu'un qui soit toujours du même avis que soi. Il me tarde de te revoir au plus vite.

Marion, que dire Marion, d'une simple rencontre à CLINATEC s'est créée une vrai amitié, merci pour tous tes conseils et ta gentillesse constante. Je ne pense pas connaître de personne plus douce

et gentille. Tous mes vœux de bonheur et il me tarde d'être en 2021 pour fêter cet heureux évènement.

Coline, nous aurons bien bavardé durant ces deux premières années dans notre carré magique avec toi et Clémentine. C'est toi qui m'a accueilli dans les locaux et qui m'a intégré à l'équipe. Un membre fondateur des « jeunes de CLINATEC », il me tarde de te revoir et que tu me racontes toutes tes histoires d'école vétérinaire.

Maxime et Guillaume, vous étiez là pendant 6 mois pour vos stages de fin d'étude, merci pour votre bonne humeur communicative qui me motivait tous les jours pour aller travailler.

A tous mes amis de l'escalade, sans vous, j'aurai sûrement perdu la raison. Partiquer le sport avec vous m'a permis d'évacuer bien du stress. Merci donc à Manon, Meije, Gaspard, Anto, Grand Guillaume et bien d'autres.

Alexia, Tamara, Marie, Camille et Benjamin, cela fait maintenant plus de huit ans que l'on se connait et que l'on fait les quatre cent coups ensemble. Je tiens à vous remercier car ma vie avant et pendant le doctorat aurait été bien différente sans vous. Il me tarde de vous revoir et de partager de bons moments avec vous.

Le groupe des « Grenoblois », (Anaïs, Agathe, Alice, Estelle, Alessia, Damien, Paul, Jeremy, Jordan, Vincent, Loïs, etc.), j'imagine qu'ils se reconnaitront tous et toutes, merci de m'avoir si bien intégré dans votre groupe depuis un an, j'ai passé des moments inoubliables avec vous et il me tarde déjà d'être à nos prochains repas/ week-end/ soirées tous ensemble.

Enfin, les meilleurs pour la fin…

Merci Gaël pour beaucoup trop de choses pour être mentionnées ici. Merci pour ta gentillesse, ta rigueur, ton attention, ta bonne humeur permanente, etc. Tu m'as toujours poussé à faire mieux et tu as toujours été présent durant la thèse pour me faire à la fois décompresser et mieux réfléchir à mon sujet. J'espère pouvoir encore partager beaucoup de bons moments avec toi. Nous allons bien rire durant les prochaines années !

Merci Clémentine, pour ces trois années passées dos à dos, ces trois années à se plaindre, ces trois années à se supporter jour après jour afin de réussir à boucler nos thèses. Il me tarde déjà d'assister à ta soutenance qui, je le sais, sera incroyable. Tu m'as aidé tellement de fois dans ma vie personnelle et professionnelle que j'ai arrêté de tenir les comptes. Il me tarde de passer encore de nombreuses années à discuter et me plaindre de tout avec toi. Que tu le veuilles ou non, malheuresement, tu devras faire avec moi en tant qu'ami durant tes années futures

Enfin, Gwen, merci pour cette dernière année. L'année 2020 aura été bien étrange sur de nombreux aspects, mais tu es restée à mes cotés durant cette période difficile. Malgré mon mauvais caractère et mon envie permanente de dormir et faire la sieste, nous avons beaucoup ri, nous nous sommes plaints sur de nombreux sujets mais cette dernière année restera inoubliable grâce à toi. Merci du fond du cœur, j'ai hâte de voir ce que l'avenir nous réserve. Plus de chats ? un Chinchilla ? Qui sait de quoi sera fait 2021, 2022, 2023, … 2100 ( ?), etc.

# Publications

## Journal

Moly A., Costecalde T., Martel F, Karakas S., Verney A, Charvet G, Chabardes S, Benabid A. L., Aksenova T. Adaptive closed-loop direct neural decoder for long-term stable bimanual exoskeleton control by a tetraplegic person using ECoG. Scientific Reports **(Under Review)**

Moly A., Aksenov A., Benabid A. L., Aksenova T. Online adaptive group-wise sparse NPLS for ECoG neural signal decoding. ArXiv

Benabid, A.L., Costecalde, T., Eliseyev, A., Charvet, G., Verney, A., Karakas, S., Foerster, M., Lambert, A., Morinière, B., Abroug, N., Schaeffer, M.-C., Moly, A., Sauter-Starace, F., Ratel, D., Moro, C., Torres-Martinez, N., Langar, L., Oddoux, M., Polosan, M., Pezzani, S., Auboiroux, V., Aksenova, T., Mestais, C., Chabardes, S., 2019. An exoskeleton controlled by an epidural wireless brain–machine interface in a tetraplegic patient: a proof-of-concept demonstration. The Lancet Neurology 0. https://doi.org/10.1016/S1474-4422(19)30321-7

Martel, F., Dupuy, T., Moly, A., Chabardès, S., Aksenova, T., 2020. Evaluation criteria for closed-loop adaptive dynamic discrete-continuous brain-computer interfaces: clinical study case with tetraplegic patient., in: 2020 International Joint Conference on Neural Networks (IJCNN). Presented at the 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–10. https://doi.org/10.1109/IJCNN48605.2020.9207243

## Oral Communications

Moly A., Costecalde T., Martel F, Lassauce A., Karakas S., Reganha G., Verney A, Milville B., Charvet G, Chabardes S, Benabid A. L., Aksenova T. High-dimensional (8D) control of complex effectors such as an exoskeleton by a tetraplegic subject using chronic ECoG recordings using stable and robust over time adaptive direct neural decoder. Annual BCI AWARDS 2020 (13/10/2020)

Moly A., Martel F., Costecalde T., Charvet G., Chabardes S., Benabid A.L., Aksenova T. Adaptive/incremental closed-loop ECoG-based BCI decoder for stepwise patient progress from brain switch to four-limb exoskeleton. Cortico 2020 (21/10/2020)

Schaeffer M.-C., Moly A. and Aksenova T. Markov mixture of experts to decode limb trajectories from ECoG for Brain Computer Interfacing, 13th Neural Coding Workshop, Torino, September 9–14, 2018

## Reward

Annual BCI AWARDS 2020: 2nd place winner. Moly A., Costecalde T., Martel F, Lassauce A., Karakas S., Reganha G., Verney A, Milville B., Charvet G, Chabardes S, Benabid A. L., Aksenova T. High-dimensional (8D) control of complex effectors such as an exoskeleton by a tetraplegic subject using chronic ECoG recordings using stable and robust over time adaptive direct neural decoder.

## Patents

Méthode de calibration iterative d'une interface neuronal directe utilisant un mélange markovien d'experts à régression multivariée.

Méthode de calibration en ligne et à facteur d'oubli d'une interface neuronale directe à régréssion multivariée pénalisée.

# Résumé

**Chapitre 1 : Introduction**

Les Interfaces Cerveau-Machine (ICM), plus connues sous le nom anglais de « Brain Computer Interface », ou BCI, sont des systèmes permettant de traduire l'activité cérébrale d'un individu grâce à un ordinateur afin d'effectuer des tâches nécessitant normalement une action des nerfs périphériques et/ou musculaires. L'essor des ICMs a permis de créer de nombreuses applications dans divers domaines comme la communication, la neuro-réhabilitation ou neuro-rééducation, le domaine du jeux vidéo, etc. Cependant, le domaine des ICMs a particulièrement été développé dans le but d'aider des patients souffrant de handicaps moteurs sévères. Les disfonctionnements moteurs peuvent être une des conséquences d'accidents comme une lésion de la moelle épinière, un accident vasculaire cérébral (AVC), ou encore la conséquence de maladies neurodégénératives ou neuromusculaires comme une sclérose latérale amyotrophique, ou une Myopathie de Duchenne, etc. Les ICMs dont le but est de palier, remplacer ou aider à la rééducation fonctionnelle de patients souffrant d'un handicap moteur sont appelés ICM motrices. Avec le développement de l'intelligence artificielle, l'augmentation de la puissance de calcul des ordinateurs et considérant les bénéfices hypothétiques pour les patients handicapés, la recherche sur les ICMs est devenue un domaine suscitant beaucoup d'espoir. Ce champ de recherche, et les applications qui en découlent, proposent de nombreux challenges à relever dans de nombreux domaines (en électronique, biologie, mathématiques, mécanique, etc.).

*Les éléments composant une ICMs.*

Une ICM permet l'interaction d'un patient avec son environnement grâce au contrôle d'un effecteur dont les actions sont régies par l'activité cérébrale du sujet. Une ICM est composée de quatre éléments principaux.

- Système d'acquisition.

Premièrement, une ICM comprend **un système d'acquisition** dont le but est d'enregistrer l'activité cérébrale du sujet. De nombreux systèmes d'enregistrement ont été développés au fil du temps avec pour objectif premier d'améliorer le ratio entre la qualité du signal et l'invasivité du système. Un large spectre de système d'acquisition existe allant du système très invasif enregistrant l'activité cérébrale proche de la source des signaux (les neurones) au système non-invasif recueillant l'activité globale du cerveau.

Les matrices de microélectrodes ou « Microelectrode array » (MEA) en anglais regroupent les systèmes les plus invasifs et les plus précis utilisés à ce jour pour l'acquisition de signaux neuronaux. Les MEAs enregistrent directement le milieu extracellulaire proche des neurones dans une zone restreinte du cerveau. A titre d'exemple, les MEAs les plus communes dans le domaine des ICMs sont les matrices

Utah qui sont composées d'une centaine d'électrodes réparties sur quelques $mm^2$. Les électrodes font entre 0.5mm et 1.5mm de long, 400µm de large et sont espacées d'environ 0.4mm. Bien que les MEAs permettent l'acquisition des signaux cérébraux avec une grande qualité et un très bon rapport signal-sur-bruit, une dégradation de la qualité des enregistrements au cours du temps a été signalée dans de nombreux articles. En ne considérant pas les quelques exemples de recherches utilisant des MEAs plus de 1000 jours après implantation, les MEAs génèrent une réaction immunitaire dégradant les signaux. A ce jour, les MEAs sont limitées à des enregistrements filaires. Cette limitation est problématique dans le cas d'enregistrements chroniques et dans le cas d'une utilisation quotidienne de par l'augmentation du risque d'infection en comparaison avec des systèmes sans fil. Les expériences d'ICMs cliniques sur une longue période de temps utilisant des MEAs sont donc peu nombreuses et souvent limitées aux contrôles d'effecteurs avec un faible nombre de degrés de liberté.

L'électroencéphalographie (EEG) est une méthode d'acquisition des signaux cérébraux à l'autre bout du spectre des possibles par rapport aux MEAs. L'EEG est un système d'acquisition non-invasif utilisant des macro-électrodes (entre 64 et 256) positionnées à la surface du crâne. Un signal EEG enregistre l'activité de millions de neurones distribués sur une grande région du cerveau (10 $cm^2$ ou plus) à la surface du crâne. De par sa simplicité d'utilisation, son cout et son risque zéro pour le patient, l'EEG est le système d'acquisition le plus commun dans le domaine des ICMs. Cependant, ce système d'acquisition comporte de nombreuses limites en comparaison aux systèmes d'acquisition plus invasifs. La résolution spatiale est limitée tandis que les bandes de fréquences exploitables sont cantonnées entre 0Hz et 100Hz. De plus, les signaux EEGs sont bruités et très sensibles aux artefacts provenant de plusieurs sources (mouvements des yeux, activités musculaires, rythme cardiaque, lignes électriques à 50Hz,etc.). Enfin, la mise en place du casque EEG est une tâche complexe, technique et couteuse en temps qui requiert une personne qualifiée afin d'obtenir des signaux exploitables. De plus, il est très difficile (voire impossible) de replacer exactement au même endroit les électrodes entre deux expériences, ce qui augmente encore la variabilité des signaux cérébraux enregistrés entre deux expériences.

L'électrocorticographie (ECoG) est un système d'acquisition se situant entre l'EEG et les MEA. L'ECoG est un système composé de macro-électrodes (avec un diamètre de l'ordre du mm) réparties sur une grille implantée au-dessus (on parlera de ECoG épidurale) ou en dessous (on parlera de ECoG subdurale) de la Dure-mère. Tout comme l'EEG, les ECoG enregistrent la somme des signaux cérébraux de milliers de neurones. Cependant les ECoGs ont démontré une bien meilleure qualité de signaux, une plus grande bande fréquence exploitable et moins d'artéfacts que les enregistrements EEG. Bien que les systèmes ECoGs n'enregistrent pas directement l'activité neuronale comme les MEAs, ils sont beaucoup moins invasifs et ont montré une stabilité d'enregistrement dans le temps plus importante.

- Le décodage des signaux cérébraux.

Après avoir été enregistrés, **les signaux cérébraux doivent être traduits** en commandes pour contrôler un effecteur. Cette étape de traduction des signaux cérébraux se déroule en quatre phases, le pré-processing, l'extraction des caractéristiques, le décodage et le post-processing. Durant la phase de pré-processing, les signaux cérébraux sont filtrés et « nettoyés » afin d'améliorer le rapport signal-sur-bruit des signaux en enlevant les éventuels artéfacts par exemple. L'extraction des caractéristiques a pour but de calculer, à partir des signaux cérébraux bruts, les informations utiles au décodage. L'étape de décodage, comme son nom l'indique, analyse les caractéristiques précédemment calculées afin de générer une commande pour l'effecteur. Les décodeurs peuvent générer des commandes discrètes (un état mental, une action binaire, etc.) ou continues (déplacer un curseur dans l'espace, bouger un bras robotique, etc.). De nombreux algorithmes ont été développés pour obtenir le décodage des signaux cérébraux le plus précis possible, mais à ce jour, aucun consensus sur la meilleure approche n'a été trouvé. Enfin, le post-processing traite la commande générée à l'étape de décodage afin d'y appliquer des contraintes supplémentaires ou d'améliorer la prédiction. L'ajout d'un seuil d'activation, de transition entre différents états ou l'application d'une moyenne glissante sur les prédictions du décodeur sont des exemples courants de post-processing.

- Le contrôle d'effecteurs.

Les commandes générées sont réalisées par **un effecteur**. De nombreux effecteurs de tous types ont été utilisés dans le domaine des ICMs. Ils peuvent être classés dans deux catégories : les effecteurs virtuels comme les curseurs d'ordinateurs, les avatars virtuels, les jeux vidéos, les systèmes de communications, etc. et les effecteurs réels regroupant tous les systèmes impliquant une interaction directe entre l'environnement et le sujet comme les bras robotiques, les orthèses, les prothèses, les fauteuils roulants, les exosquelettes, etc.

- Le retour sensoriel (feedback).

Enfin, une étape essentielle des ICMs est le retour sensoriel (feedback) suscité par les actions de l'effecteur sur le patient. Dans la plupart des ICMs controlées par des patients handicapés, le retour sensoriel généré par leurs actions est visuel. Cette étape est très importante car le retour sensoriel influe directement sur l'activité cérébrale du sujet. On parle de système en boucle fermé (closed-loop). Quelques exemples d'expériences ICMs intégrant des retours haptiques ont été menées avec des patients handicapés.

*Stratégie de contrôle d'une ICM.*

**Afin de contrôler une ICM, plusieurs stratégies** ont été mises en place. Les stratégies de contrôle peuvent se diviser en deux catégories : le contrôle exogène et endogène. Les systèmes exogènes utilisent des stimuli extérieurs (auditifs, visuels, etc.) afin de générer une modulation des signaux cérébraux du patient facilement reconnaissable et interprétable. A l'inverse, les systèmes endogènes permettent le contrôle d'effecteurs via

les variations des signaux cérébraux directement induites par le patient sans stimuli extérieurs. Ce manuscrit se concentre plus particulièrement sur les ICMs endogènes. Les stratégies endogènes se décomposent en deux approches : le « remapping » somatotopique et le décodage neural direct.

**Le « remapping » somatotopique, ou imagerie motrice (IM)** associe une commande spécifique de l'effecteur (bouger un curseur vers le haut, la gauche, activer la marche d'un robot, etc.) a une action réelle ou imaginaire arbitrairement choisie comme bouger la langue, le coude, l'épaule, les jambes, les doigts, etc. Chaque action réelle ou imaginaire génère une activité cérébrale différente qui peut être liée artificiellement à une commande spécifique de l'effecteur.

**Le décodage neural direct ou biomimétique** utilise l'activité cérébrale enregistrée lors de la réalisation d'un mouvement (ou de l'imagination de celui-ci) pour générer un ordre de l'effecteur directement lié aux signaux cérébraux. Comme son nom l'indique, le décodage neural direct traduit directement les intentions du patient en action.

A titre d'exemple, considérons une expérience où un patient a pour tâche de contrôler un bras robotique dans le but d'atteindre une cible située en hauteur. Une stratégie de type IM non-direct associe cette tâche à une action du patient (imaginée ou non) sans lien avec la tâche à accomplir (bouger son épaule, son coude, sa langue, etc.). Dans le cas d'un décodage neural direct, afin de bouger le bras robotique vers le haut, le patient devra simplement imaginer bouger son bras vers le haut.

Bien que la stratégie de contrôle neural direct soit bien plus complexe à décoder, elle présente de nombreux avantages pour le patient. La stratégie de décodage neural direct est plus naturelle et demande une charge mentale moins importante pour le patient afin de réaliser des tâches complexes. De plus, la stratégie IM est limitée dans le nombre possible d'actions réalisables dans la même expérience : après avoir associé les jambes, les poignets, les coudes, etc. à une tâche spécifique, il devient difficile d'ajouter d'autres degrés de liberté. Cependant, à ce jour, le décodage en temps réel des mouvements d'un effecteur durant des expériences cliniques utilisant la stratégie de décodage neural direct n'a été réalisé qu'avec le système d'acquisition MEA.

*ICM pour une utilisation dans la vie quotidienne.*

Le but final des ICMs est de créer des systèmes utilisables par des personnes handicapées dans la vie quotidienne afin d'améliorer leur niveau de vie et leur autonomie. Dans cette optique, les ICMs doivent répondre à de nombreuses exigences.

Pour une application quotidienne, **le système d'enregistrement** doit permettre un enregistrement chronique et stable des signaux cérébraux. Bien que quelques études aient démontré la stabilité des implants de type MEA sur plusieurs centaines de jours, l'enregistrement direct de l'activité extracellulaire a montré une grande variabilité dans la stabilité des implants (dégradation au cours du temps) et une grande variabilité dans

l'activité des neurones. De plus, les MEAs sont pour l'instant limitées à des enregistrements filaires. Les enregistrements ECoG ont montré de meilleures performances avec de nombreuses études démontrant la qualité et la stabilité des enregistrements plusieurs années après implantation. Bien que les systèmes de type EEG soientt non-invasif, la vulnérabilité aux artefacts, la grande variabilité des enregistrements et la nécessité de replacer au même endroit les électrodes à chaque utilisation ne font pas des EEGs de bons candidats pour une utilisation quotidienne.

Il est nécessaire que l'ICM créée soit **suffisamment précise et offre assez de liberté** à l'utilisateur. A ce jour, les ICMs ayant démontrées un grand nombre de degrés de liberté sont celles basées sur les systèmes d'acquisition MEAs. Un patient tétraplégique a pu contrôler, grâce à une MEA implantée dans le cortex moteur, un bras robotique avec 10 degrés de liberté (déplacement du bras dans l'espace 3D, rotation 3D du bras ainsi que 4 positions de main différentes). Pour les systèmes moins invasifs, plusieurs études ont démontré le contrôle 3D de curseurs et bras robotiques à partir d'ICM utilisant des enregistrements ECoGs, tandis que le décodage de signaux EEG se limite généralement à du contrôle 1D ou 2D.

Les actions de la vie courante nécessitent souvent l'utilisation simultanée ou séquentielle de **plusieurs membres comme** la réalisation de **mouvements bimanuels**. Le décodage multi-membres est un domaine très peu étudié dans le domaine des ICMs bien que la possibilité de réaliser des mouvements simultanés ou alternatifs de plusieurs membres puisse être un grand avantage pour les personnes handicapées. La plupart des expériences d'ICM se concentrent sur le décodage d'un membre en particulier ou une action spécifique d'un effecteur. Le peu d'expériences multi-membres réalisées se limite à des études précliniques sur singe ou des expériences cliniques de décodage des mouvements des doigts.

L'un des critères principaux du développement des ICMs est la capacité à proposer aux utilisateurs **un contrôle asynchrone** sur l'effecteur. Les systèmes synchrones donnent au sujet le contrôle d'un effecteur dans des fenêtres de temps limitées périodiquement proposées par un opérateur afin d'activer ou désactiver le contrôle de l'ICMs. A l'inverse, un ICM asynchrone est disponible en permanence. Il est capable de décoder un état mental de repos nommé « idle state » (IS) dans ce manuscrit. Lorsque l'état de repos est détecté, aucune action n'est réalisée par l'effecteur.

La plupart des expériences de l'état de l'art sont des ICMs synchrones. Ceci est d'autant plus vrai dans les expériences dont le but est le contrôle de nombreux degrés de liberté comme par exemple, l'étude citée précédemment où un patient utilisait un bras robotique avec 10 degrés de liberté. Un contrôle asynchrone est obligatoire dans le cas d'ICM pour la vie courante et d'autant plus dans le cas de contrôle d'effecteurs multi-membres.

Un ICM utilisable quotidiennement doit décoder les signaux **en temps réel** afin de limiter la latence entre l'action désirée par le sujet et la réalisation de celle-ci par l'effecteur (~10-20Hz). La majorité des études d'ICMs sont pratiquées sur des jeux de données en offline. Dans ces conditions, le temps et la puissance de calcul nécessaires pour décoder les signaux cérébraux ne sont pas des facteurs limitants. Pour répondre aux besoins d'une utilisation en temps réel, les algorithmes de décodage doivent être suffisamment simples et optimisés pour nécessiter un temps de calcul très faible.

En plus d'être utilisable en temps réel, l'algorithme de décodage doit garder **des performances stables au cours du temps**. Les signaux cérébraux sont non-stationnaires, ils changent au cours du temps et ne génèrent pas les mêmes variations d'amplitudes au cours du temps pour la même action. Cette évolution des signaux cérébraux au cours du temps engendre une baisse des performances des modèles de décodage et requiert donc de recalibrer les modèles. Cependant une étape de recalibration quotidienne des modèles de décodage (voir plus) n'est pas envisageable dans le cas d'utilisation d'ICM dans la vie courante. Ce sont des évènements complexes qui peuvent s'avérer fatiguants pour le patient et qui doivent rester exceptionnels. Des stratégies pour mettre à jour les modèles facilement doivent être considérées pour une utilisation quotidienne d'un ICM.

Enfin, comme mentionné précédemment, l**e retour visuel (feedback) lié à l'utilisation d'un effecteur modifie l'activité cérébrale** de l'utilisateur. Plusieurs études ont démontré que l'intégration dans la phase de calibration des signaux cérébraux générés par le retour visuel peut améliorer grandement les performances de décodage durant des expériences en boucle fermée (closed-loop). Plusieurs stratégies ont été développées et sont regroupées sous le nom d'adaptation du décodeur en boucle fermée : (« Closed-Loop Decoder Adaptation », CLDA, en anglais). Ces stratégies mettent à jour le modèle en prenant en compte les feedbacks du patient grâce à des procédures de calibration offlines ou en temps réel. Les algorithmes intégrant une procédure de CLDA ont montré de meilleures performances et une meilleure stabilité au cours du temps lors d'expériences d'ICM en temps réel. L'intégration de procédures similaires semble nécessaire pour garantir la stabilité et les performances de tout ICM dédiée à la vie quotidienne.

*Recherche de doctorat.*

Les recherches de thèses présentées dans ce manuscrit ont été menées dans le but de créer de nouveaux algorithmes de décodage répondant aux critères des ICMs précédemment mentionnés. Les algorithmes de cette thèse auront donc pour but d'être utilisables en temps réel, d'être asynchrones, d'être utilisables pour le contrôle multi-membres et d'être capables d'être facilement mis à jour.

Ce doctorat est mené dans le cadre du projet « BCI and Tétraplégie» de CLINATEC. CLINATEC est un centre de recherche biomédical situé sur le site du CEA-Grenoble en collaboration avec l'Université Grenoble Alpes (UGA) et le Centre Hospitalier

Universitaire de Grenoble Alpes (CHUGA). Le but de l'essai clinique est de permettre à des patients tétraplégiques de contrôler des effecteurs complexes tel qu'un exosquelette quatre membres via un décodage neural direct de signaux cérébraux chroniques enregistrés grâce à deux implants sans fil ECoG épiduraux nommés WIMAGINE. L'ensemble des études et expériences menées dans le cadre de la thèse reposeront sur des enregistrements ECoG uniques, et viseront à répondre aux besoins de l'essai clinique et à permettre au patient de contrôler les différents membres de l'exosquelette.

**Chapitre 2 : L'essai clinique « BCI et Tétraplégie »**

L'essai clinique de CLINATEC « BCI et Tétraplégie » a pour but de faire la preuve de concept qu'un patient tétraplégique implanté avec des électrodes ECoG épidurales peut contrôler un effecteur complexe tel qu'un exosquelette en utilisant un décodage neural direct. L'essai clinique toujours en cours a été référencé dans le registre ClinicalTrials.gov sous l'identifiant NCT02550522 le 11/09/2015. Cet essai clinique a été approuvé par l'Agence Nationale de Sécurité du Médicament et des produits de santé (ANSM) sous l'identifiant 2015-A00650-49 et le Comité de Protection des Personnes (CPP) sous le nom 15-CHUG-19.

Entre le début de l'essai clinique et juin 2020, trois patients ont été inclus dans le protocole clinique. Le premier patient a été implanté en mai 2016. Au fait d'un problème technique lié aux implants, le patient a été explanté et retiré de l'essai clinique. Le deuxième patient a été implanté en Juin 2017. Un troisième patient a été inclu dans l'essai clinique fin 2019. Les résultats décrits dans cette thèse ne se concentrent que sur les analyses, les études et les expériences menées avec le deuxième patient. Ce patient tétraplégique a subi une lésion complète de la moelle épinière au niveau des vertèbres C4-C5. Son score ASIA (American Spinal Injury Association Impairment) a été évalué à 4 et 5 pour le côté droit et gauche du corps tandis que la contraction des extenseurs du poignet gauche (droit) a reçu la note de 3 (0). Tous les autres muscles ont été évalués à un score ASIA de 0.

Durant l'essai clinique, le deuxième patient a réalisé plusieurs expériences avec différents effecteurs réels et virtuels (exosquelette, fauteuil roulant, avatar virtuel, jeux vidéos, etc.). Les résultats obtenus dans cette thèse se concentrent uniquement sur les performances du patient à contrôler l'exosquelette et l'avatar virtuel. L'avatar virtuel est une reproduction virtuelle de l'exosquelette permettant au patient de simuler le contrôle de l'exosquelette à domicile.

Le patient a été entrainé à contrôler graduellement des effecteurs de plus en plus complexes. L'exosquelette et l'avatar virtuel proposent 12 degrés de liberté : le contrôle continu 3D des mouvements de la main gauche et droite, le contrôle continu 1D de la rotation des poignets du bras gauche et droit, un système de préhension 1D discret de la main gauche et droite, la marche et l'état de repos. Le patient a d'abord appris à contrôler

individuellement chaque degré de liberté avant d'essayer des expériences plus complexes où les paradigmes expérimentaux ont été combinés. Dans chaque expérience, il est demandé au patient d'atteindre des cibles dépendantes du paradigme expérimental utilisé. Par exemple, dans le cas du mouvement 3D de la main gauche, il est demandé au patient d'atteindre une cible avec la main gauche. Lors d'une tâche de rotation du poignet, le patient doit faire tourner le poignet de l'exosquelette jusqu'à une certaine limite.

Une expérience d'ICM en temps réel se déroule en deux phases. Premièrement une phase optionnelle de calibration du modèle dans le cas où le modèle est initialisé à zéro ou si le modèle doit être mis à jour. Durant la phase de calibration, les signaux cérébraux sont labélisés afin de mettre à jour le modèle en temps réel. Durant la seconde phase, le modèle est fixé et des tâches similaires aux tâches proposées durant la phase de calibration sont effectuées afin d'évaluer les performances du modèle.

**Chapitre 3 : Décodeur de Signaux Cérébraux**

*Les types de décodeurs*

De nombreux algorithmes de décodage ont été développés et utilisés dans la communauté des ICMs dans le but de décoder le plus précisément possible les signaux cérébraux. Les algorithmes de décodage peuvent se diviser en deux catégories, les algorithmes dont le but est de classer l'activité cérébrale dans des états discrets (classifieurs) et ceux dont le but est de traduire l'activité cérébrale par une variable continue (modèle de régression ou filtre bayésiens). La plupart des études évaluant les performances de classifieur sont réalisées via des enregistrements de type EEG tandis que le décodage de variables continues est le domaine de prédilection des MEAs. A ce jour, il n'existe aucun consensus sur les meilleurs algorithmes à utiliser pour le décodage des signaux cérébraux car leurs performances sont très variables selon la tâche à effectuer, le type d'enregistrement etc.

En général, les modèles de régression et les filtres bayésiens échouent à fournir des estimations neutres (associées à une prédiction en vitesse nulle) qui ne génèrent aucun mouvement de l'effecteur. Cette caractéristique des modèles de prédictions continues est très problématique dans le cadre de la création d'un ICM asynchrone qui doit être capable de fournir un état de repos fiable à l'utilisateur.

Pour résoudre ce problème, deux stratégies ont émergé, à savoir : la détection de l'état de repos par post-traitement, et l'intégration directe de la détection de l'état de repos dans l'algorithme de décodage. Dans le deuxième cas, la détection de l'état de repos et souvent synonyme d'introduction de non-linéarités dans le décodeur. Pour ce faire, des algorithmes dit hybrides utilisant un algorithme de classification en parallèle d'un ou de plusieurs modèles de régressions ont été testés. Le but des algorithmes hybrides est

d'utiliser la variable latente discrète décodée par le modèle de classification afin d'introduire un comportement non-linéaire dans le décodeur de variables continues. Dans le cas de la détection de l'état de repos, le classifieur est en mesure d'inhiber le décodeur de variables continues afin de générer une commande de contrôle neutre (ex. vitesse de prédiction nulle).

*La mise à jour des modèles*

Afin de prendre en compte les variations des signaux cérébraux au cours du temps et les signaux cérébraux liés aux retours sensitifs du patient lors de l'utilisation d'un effecteur, il est nécessaire d'utiliser des algorithmes intégrant une procédure CLDA. Les procédures CLDA existantes peuvent être classées selon la fréquence de mise à jour du modèle.

Le modèle peut être mis à jour à la fin de chaque session, à chaque essai, toutes les 15 minutes etc. Généralement, dans le cas où le temps entre deux mises à jour est assez long, la procédure se déroule en offline. A l'inverse, la mise à jour peut être opérée dans des intervalles de temps beaucoup plus courts de l'ordre de la minute, de quelques secondes voir moins. Dans ce cas, la procédure est généralement réalisée en temps réel durant l'expérience. Les procédures CLDA en temps réel utilisent souvent des algorithmes adaptatifs dits incrémentaux. Un algorithme incrémental modifie le modèle de décodage en se basant sur les nouvelles données enregistrées en temps réel au cours de l'expérience et le modèle précédemment calibré.

Les algorithmes adaptatifs incrémentaux ont été utilisés dans plusieurs études cliniques et précliniques dans le but de mettre à jour en temps réel les modèles de décodage. Les algorithmes adaptatifs incrémentaux ont montré des résultats prometteurs pour contrer la variabilité naturelle des signaux cérébraux et pour intégrer directement le patient dans le processus de calibration du modèle. Le patient apprend du modèle tout autant que le modèle apprend du patient.

La plupart des classifieurs adaptatifs incrémentaux ont été testés dans des expériences utilisant des enregistrements de type EEGs. Ils sont basés sur des algorithmes bien connus dans le domaine du machine learning mais adaptés afin d'intégrer une mise à jour incrémentale. Parmi eux, nous pouvons citer l'« incremental adaptive linear discriminant analysis (LDA)», l'« incremental adaptive Kalman-LDA», le « adaptive information matrix (ADIM) » ou encore le « incremental Support Vector Machine (ISVM) »etc.

A l'inverse, les algorithmes incrémentaux de décodage de variables continues ont surtout été étudiés en se basant sur l'analyse de signaux cérébraux enregistrés via MEAs. Les algorithmes adaptatifs de ce type les plus connus font partie de la famille des filtres de Kalman. Un seul algorithme de décodage de variables continues a été créé dans le cadre de décodage de signaux ECoG. Cet algorithme incrémental se nomme « Recursive Exponentially Weighted N-way Partial Least Square (REW-NPLS) ». A notre

connaissance, aucun algorithme incrémental adaptatif hybride n'a été appliqué dans le domaine des ICMs.

**Chapitre 4 : Décodeur REW-MSLM**

Le but de cette thèse est de proposer un algorithme incrémental adaptatif permettant le contrôle asynchrone en temps réel d'un effecteur multi-membres. Cette thèse propose donc un nouvel algorithme nommé « Recursive Eponentially Weighted Markov Switching multi-Linear Model (REW-MSLM)» inspiré des algorithmes « Recursive Exponentially Weighted N-way Partial Least Square (REW-NPLS) » et « Markov Switching Linear Model (MSLM) ». L'algorithme REW-MSLM s'inspire de la structure et de l'algorithme de classification dynamique utilisé dans MSLM.

REW-MSLM est un modèle se basant sur une architecture de « Mixtures d'Experts (ME)». Le principe des ME est d'estimer les prédictions de plusieurs décodeurs de variables continues (régression) que l'on nomme « experts ». La prédiction de chaque expert est alors pondérée (inhibée ou stimulée) par un poids reflétant la probabilité que chaque expert a d'être actif. Le modèle estimant le poids de chaque expert est un classifieur discret appelé « gate ». La valeur de cette variable discrète latente (le poids de chaque expert) est directement déduite des signaux cérébraux et de leurs distributions. Tout comme le MSLM, le REW-MSLM réalise une estimation dynamique de la probabilité d'activation des experts (la valeur de la variable latente discrète). Dans l'algorithme REW-MSLM, la séquence de probabilité des états latents est générée par une chaîne de Markov d'ordre 1. Plus précisément, les probabilités sont estimées par un Modèle de Markov Caché nommé « Hidden Markov Model (HMM) » en anglais. Intégrer un classifieur dynamique de type HMM dans un modèle de type ME a pour but de réduire le nombre de fausses détections des états actifs et de l'état de repos.

REW-MSLM regroupe donc plusieurs sous-modèles : les experts et la gate. Chaque sous-modèle doit être calibré afin de traduire les signaux cérébraux en commande pour l'effecteur. La calibration de chaque expert et de la gate est réalisée individuellement en utilisant une procédure d'apprentissage supervisé. L'apprentissage supervisé de chaque sous-modèle est effectué grâce à l'algorithme incrémental adaptatif REW-NPLS. REW-NPLS est une évolution de l'algorithme « Partial Least Square (PLS) » connu pour sa stabilité dans le traitement de données à grandes dimensions. REW-NPLS traite les données sous forme tensorielle et effectue une estimation des paramètres d'un modèle multilinéaires en temps réel de façon incrémentale. En plus de l'évaluation du modèle en temps réel, une procédure nommée Validation Récursive ou « Recursive Validation (RV) » est intégrée dans l'algorithme afin de déterminer en temps réel l'hyperparamètre de l'algorithme REW-NPLS. L'hyperparamètre de REW-NPLS noté $f$ est la dimension de l'espace des variables latentes dans lequel sont projetées les données afin d'estimer le modèle de décodage.

La combinaison de l'algorithme MSLM et REW-NPLS permet la création d'un modèle de type ME applicable en temps réel avec des sous-modèles recalibrés de manière incrémentale en temps réel. REW-MSLM présente de nombreux avantages pour contrôler un effecteur asynchrone multi-membres en temps réel.

Premièrement, chaque expert peut être associé à un membre ou groupe de degrés de liberté. Chaque expert est alors calibré individuellement pour réaliser une action spécifique : aucun expert ne doit apprendre tous les degrés de liberté. C'est une création de modèle « par morceaux » où chaque expert à son domaine de prédiction. Dans le cas d'application de ce manuscrit, chaque expert est associé à un mouvement spécifique de l'effecteur comme bouger le bras gauche, tourner le poignet droit, etc. La gate a pour rôle de choisir quel expert sélectionner, et donc quel mouvement doit être réalisé par l'effecteur. De surcroît, la gate est calibrée pour détecter l'état de repos (idle state : IS) afin de proposer un contrôle asynchrone de l'effecteur au patient.

**Chapitre 5 : Décodeur PREW-NPLS et APREW-NPLS**

Le décodage de signaux cérébraux, bien que reposant sur certains signaux neurologiques généralisables à tous les individus, est extrêmement dépendant du sujet. Afin de créer une ICM pouvant être calibrée et utilisée par le plus grand nombre, de nombreuses caractéristiques (features) sont extraites des signaux cérébraux. La phase de calibration a pour rôle de déterminer, parmi toutes les caractéristiques calculées pour un sujet donné les caractéristiques les plus pertinentes pour décoder les signaux cérébraux.

Afin de s'assurer que les caractéristiques les plus pertinentes soient calculées pour chaque sujet, l'espace des caractéristiques est souvent de grande dimension. Cependant, le décodage des signaux cérébraux à partir d'un grand nombre de caractéristiques peut engendrer plusieurs problèmes.

Du point de vue de l'apprentissage des modèles, si les caractéristiques sont corrélées ou si la base de données d'entraînement est trop petite, un grand nombre de caractéristiques peut mener à un problème de surapprentissage ou fléau de la dimension (« overfitting ou curse of dimentionality »). De plus, un grand nombre de caractéristiques demande une puissance de calcul plus importante pour calibrer et appliquer le modèle, ce qui peut être problématique dans le cas d'application d'une ICM portable (avec une puissance de calcul limitée). Enfin un modèle créé à partir d'un nombre de caractéristiques important est plus compliqué à interpréter.

Afin de résoudre les difficultés citées précédemment, les algorithmes Penalized REW-NPSL (PREW-NPLS) et Automatic PREW-NPLS (APREW-NPLS) ont été développés et testés. Ces algorithmes ont pour but d'être intégrés dans REW-MSLM à la place de REW-NPLS.

PREW-NPLS est une version pénalisée de REW-NPLS. Plus précisément, PREW-NPLS intègre une pénalisation de la norme $L_0$, $L_{0.5}$ ou $L_1$ au modèle REW-NPLS. Ces trois types de régularisation ont pour but de réduire le nombre de caractéristiques impliquées dans la prédiction des signaux cérébraux en imposant un poids de 0 aux caractéristiques les moins utiles. La pénalisation de REW-NPLS se déroule durant l'algorithme de décomposition tensorielle (PARAFAC).

PREW-NPLS est, tout comme REW-NPLS, un algorithme adaptatif incrémental qui peut être calibré et appliqué en temps réel. Cependant PREW-NPLS ajoute un nouvel hyperparamètre nommé coefficient de pénalisation noté $\lambda$. Le coefficient de pénalisation est une variable qui détermine le « degré » de pénalisation (la pénalité) imposé aux coefficients du modèle. Plus $\lambda$ est important, plus le modèle est pénalisé et donc plus le nombre de paramètres du modèle fixés avec un poids de 0 augmente. Cet hyperparamètre a une grande influence sur le modèle estimé et la performance de celui-ci. Malheureusement, comme tout hyperparamètre, $\lambda$ doit être choisi avant la phase de calibration du modèle. Il est nécessaire de mener des études préliminaires afin de déterminer la valeur $\lambda$ la plus optimale avant d'utiliser PREW-NPLS en temps réel.

Cette procédure préliminaire d'optimisation de $\lambda$, généralement réalisée en offline, est contre-intuitive pour un algorithme incrémental adaptatif en temps réel. De plus, aucune étude ne permet d'affirmer que l'hyperparamètre de pénalisation optimal $\lambda$ déterminé durant des analyses offline soit le même que le coefficient de pénalisation optimal pour un application en temps réel, et que celui-ci ne change pas au cours du temps (et de l'expérience).

L'algorithme APREW-NPLS a été développé afin de déterminer en temps réel durant la période de calibration le meilleur hyperparamètre de pénalisation $\lambda$ parmi un groupe de valeurs disponibles. APREW-NPLS considère la sélection du coefficient de pénalisation comme le problème du bandit manchot («Multi-armed bandit problem »). Le problème est résolu en utilisant une stratégie d'apprentissage par renforcement (reinforcement learning). Les différents modèles PREW-NPLS (avec différents hyperparamètres de pénalisation) sont mis en compétition durant la phase de calibration incrémentale. A chaque incrément de calibration, les modèles obtenant les meilleures performances dans l'algorithme de Validation Récursive (RV) sont sélectionnés pour être recalibrés à la prochaine mise à jour tandis que les modèles les moins performants ne le sont pas. Cette stratégie permet de choisir un coefficient de pénalisation optimal en temps réel parmi plusieurs coefficients $\lambda$ disponibles.

**Chapitre 6 : Classifieur H2M2 pour la gate de REW-MSLM**

Dans le but d'améliorer les performances de prédiction de la gate de REW-MSLM, un nouvel algorithme (H2M2) inspiré du Modèle de Markov Caché Hiérarchique ou « Hierarchical Hidden Markov Model (HHMM) » est proposé dans cette thèse.

A l'inverse du HMM qui considère tous les états discrets de façon équivalente, HHMM structure les états en couches hiérarchisées. Les états d'une couche inférieure sont dépendants des états de la couche supérieure. Ce type de modèle s'avère intéressant dans le cas de données naturellement structurées.

Dans notre cas d'application, les états permettant le contrôle des membres d'un exosquelette peuvent être hiérarchisés. Considérons le cas de contrôle asynchrone du mouvement 3D de la main gauche et de la main droite ainsi que la rotation 1D du poignet gauche et droit (5 états discrets avec l'état de repos). Dans ce cas, le nouvel algorithme H2M2 est composé de trois sous-modèles de type HMM. Le premier HMM de la couche supérieure se concentre sur l'activation de l'état de repos (IS) ou l'état « membres du côté gauche » (regroupant les états de mouvement de la main et de rotation du poignet du bras gauche) ou l'activation de l'état « membres du côté droit ». Ce premier HMM classifie trois états différents. Les deux autres HMMs font partie de la couche inférieure et se concentrent sur la classification binaire entre l'état de mouvement ou de rotation d'une des mains (un HMM pour la main gauche et l'autre pour la main droite).

Cette architecture a pour but d'améliorer les performances de décodage et réduire le temps de latence de détection des états.

**Chapitre 7: Données expérimentales**

Tous les algorithmes ont été testés durant une première phase de test offline en utilisant une procédure dite pseudo-online. Le but des études pseudo-onlines est de simuler en offline les conditions d'entraînement et de calibration réalisées durant les expériences en temps réel. Bien que les résultats obtenus durant les études pseudo-onlines ne soient pas généralisables au cas de l'application en temps réel, ils permettent de donner une tendance sur les résultats pouvant être obtenus durant les expériences onlines. Tous les jeux de données utilisés pour les études pseudo-online sont tirés d'expériences réalisées en temps réel par le patient.

*Jeux de données d'évaluation de l'algorithme REW-MSLM.*

Les tests pseudo-onlines ont été réalisés sur des jeux de données provenant d'expériences en temps réel où le patient contrôlait alternativement le mouvement des deux bras dans l'espace 3D (expériences 6D). Durant ces expériences en temps réel, l'algorithmes REW-NPLS a été utilisé pour prédire le mouvement des deux bras. Afin d'évaluer au mieux les performances de REW-MSLM, l'algorithme a été testé en pseudo-online selon trois paradigmes différents. Durant la première série d'expériences, chaque

expérience est considérée individuellement, les modèles sont initialisés à zéro, avant d'être calibrés et testés durant la même session. Durant la deuxième série d'expériences, les modèles sont calibrés et testés durant chaque session mais les modèles ne sont pas réinitialisés à zéro entre deux expériences (la calibration est cumulative au cours des sessions). Enfin, durant la troisième série, le modèle calibré à la dernière session de la série précédente est testé sur de nouvelles données sans réaliser de recalibration au début de chaque session dans le but d'évaluer la stabilité du modèle.

Après avoir évalué les performances du modèle avec des études pseudo-onlines, REW-MSLM a été intégré dans la plateforme de décodage en temps réel de CLINATEC nommé « Adaptive Brain Signal Decoder (ABSD) » afin de réaliser des expériences 8D en temps réel. Durant ces expériences le patient devait contrôler 8 degrés de liberté continue regroupés en 5 états : le contrôle discret de l'état de repos, le contrôle continu des mouvements de la main gauche et droite dans l'espace 3D et le contrôle 1D de la rotation du poignet gauche et droit. Durant ces expériences, le patient contrôlait l'exosquelette ou l'avatar virtuel selon si l'expérience se déroulait à CLINATEC ou au domicile du patient. Un modèle différent a été calibré pour le contrôle de chaque effecteur. Chaque modèle a été calibré durant 6 expériences. Le modèle calibré pour le contrôle de l'exosquelette a été testé durant 15 expériences réparties entre 0 et 167 jours après la fin de la calibration du modèle. Le modèle calibré pour le contrôle de l'avatar virtuel a quant à lui été testé durant 37 expériences réparties entre 5 et 203 jours après la fin de la calibration du modèle.

*Jeux de données d'évaluation de l'algorithme PREW-NPLS et APREW- NPLS.*

Les expériences 8D réalisées avec l'avatar virtuel et l'algorithme REW-MSLM en temps réel, présentées précédemment, ont été utilisées pour mener des études pseudo-onlines des performances des algorithmes PREW-NPLS et APREW-NPLS. Plus particulièrement, les analyses pseudo-onlines se sont concentrées sur les données de contrôle 3D des mouvements de la main gauche et de la main droite.

*Jeux de données d'évaluation de l'algorithme H2M2.*

De nouvelles expériences de contrôle de l'avatar virtuel ont été réalisées avec le patient afin d'évaluer les performances pseudo-onlines de H2M2. Durant ces expériences, l'algorithme REW-MSLM a été utilisé pour décoder les signaux cérébraux en temps réel. Le patient devait contrôler 5D continus regroupées en 5 états : le contrôle discret de l'état de repos, le contrôle continu des mouvements de la main gauche et droite dans l'espace 1D et le contrôle 1D de la rotation du poignet gauche et droit. Le but des expériences était de réaliser le plus de transitions possibles entre les 5 états (l'état de repos, l'état de contrôle du mouvement de la main gauche, l'état de contrôle de rotation de la main gauche, l'état de contrôle du mouvement de la main droite, l'état de contrôle de rotation de la main droite) afin d'étudier la capacité de H2M2 à transiter de façon efficace entre les états.

**Chapitre 8: Intégration de REW-MSLM**

L'algorithme adaptatif REW-MSLM a été intégré à la plateforme de décodage en temps réel ABSD. ABSD réalise les étapes de pré-processing, extraction des caractéristiques, application et calibration de REW-MSLM et post-processing. Les signaux cérébraux sont décodés à une fréquence de 10Hz tandis que le modèle est mis à jour toutes les 10 à 15 secondes.

Durant l'étape de pré-processing, les signaux cérébraux enregistrés par les implants WIMAGINE à 586Hz sur 64 électrodes (32 par implant) sont filtrés entre 0.5Hz et 300Hz. Les signaux sont ensuite amplifiés et les valeurs aberrantes retirées.

Les caractéristiques sont générées toutes les 100ms à partir de la dernière seconde d'enregistrement. Les signaux cérébraux sont transformés dans le domaine temps-fréquence via une transformée en ondelettes continue. Le signal est décomposé en 15 bandes de fréquences allant de 10Hz à 150Hz. Enfin, la fenêtre de temps est sous-échantillonnée pour créer un tenseur de caractéristiques $\underline{\mathbf{X}}_t \in \mathbb{R}^{10x15x64}$. En parallèle, les labels discrets et continus associés aux signaux cérébraux sont calculés pour l'apprentissage supervisé des experts et de la gate de l'algorithme REW-MSLM.

Une limite de vitesse est appliquée en post-processing aux prédictions estimées par le REW-MSLM.

**Chapitre 9: Evaluation des performances**

*Evaluation pseudo-online des performances de REW-MSLM.*

REW-MSLM est composé de modèles de décodage continu (les experts) et d'un modèle de décodage discret (la gate). Les performances de chaque modèle ont été évaluées indépendamment les unes des autres et ont été comparées aux performance de REW-NPLS.

Durant les analyses pseudo-onlines, les performances discrètes ont été estimées grâce au F-score et à la précision (Accuracy). Les indicateurs évaluant la fréquence des erreurs, leurs durées, et la latence entre l'ordre donné par l'expérimentateur et la réalisation de l'action par l'effecteur sont aussi calculés.

Les performances de décodage des variables continues ont été évaluées grâce au produit scalaire entre la prédiction du modèle et la prédiction optimale qu'aurait pu prédire le modèle. Plus le produit scalaire de deux vecteurs normalisés est important, plus les vecteurs sont similaires.

*Evaluation online des performances de REW-MSLM.*

Les performances discrètes des expériences en temps réel sont évaluées avec le F-score et la précision (Accuracy). Les performances continues en temps réel sont cette fois évaluées grâce au nombre de cibles touchées et le R-Ratio. Le R-Ratio est défini comme

le ratio entre la distance parcourue par la trajectoire de la main pour atteindre une cible et la distance parcourue dans le cas d'une trajectoire optimale.

*Evaluation pseudo-online des performances de PREW- NPLS et APREW-NPLS.*

Les performances des algorithmes PREW-NPLS et APREW-NPLS sont comparées aux performances de REW-NPLS en utilisant le produit scalaire. L'algorithme PREW-NPLS a été testé pour une pénalisation de la norme de type $L_0$, $L_{0.5}$ et $L_1$ et plusieurs valeurs d'hyperparamètre de pénalisation $\lambda$. De plus, un score de pénalisation est calculé pour chaque modèle. Le score de pénalisation est le ratio entre le nombre d'électrodes fixées à 0 et le nombre total d'électrodes. La pénalisation est appliquée sur les caractéristiques spatiales (les électrodes), mais une étude similaire pourrait être menée en pénalisant les caractéristiques temporelles ou fréquentielles ou les trois en même temps.

*Evaluation pseudo-online des performances de H2M2.*

Les performances discrètes de H2M2 sont comparées à deux HMMs. L'un des HMMs, nommé $HHM_{limited}$, a un nombre limité de transitions possibles entre les états. La matrice de transition a été construite de sorte que $HHM_{limited}$ ne puisse pas transiter entre les états actifs (état de mouvement main gauche, état de rotation main gauche, etc.). Pour transiter d'un état actif à un autre, $HHM_{limited}$ doit forcément activer l'état de repos entre les deux états actifs. L'autre HMM, nommé $HMM_{full}$ n'a pas été modifié. Les indicateurs de performances calculés pour évaluer les prédictions de H2M2 sont : le F-score, l'Accuracy, la Spécificité, le Bookmaker, le Coefficient Jaccard, la différence HF, la corrélation de Matthews, le Kappa et le Gmean.

**Chapitre 10: Résultats**

*Pseudo-online performances de REW-MSLM.*

La gate de REW-MSLM a montré des performances de classification statistiquement supérieures dans tous les paradigmes en comparaison à REW-NPLS avec un F-score et une Accuracy de $fsc = 86 \pm 3\%$, $acc = 93 \pm 1.8\%$. De plus, la fréquence d'erreurs prédites a drastiquement chuté (diminution moyenne de plus de 90%). Les prédictions continues de REW-MSLM n'ont pas démontrées de différences statistiquement significatives de performances en comparaison des performances de l'algorithme REW-NPLS.

*Online performances de REW-MSLM.*

Les performances discrètes ont démontré un F-score moyen de $fsc = 76 \pm 9\%$ et une Accuracy moyenne de $acc = 93 \pm 3\%$ pour le contrôle de l'avatar virtuel. Les performances discrètes de contrôle de l'exosquelette ont montré des résultats similaires avec un F-score de $75 \pm 12\%$ et une Accuracy de $92 \pm 4\%$.

Durant les expériences avec l'avatar virtuel, en moyenne $55 \pm 18\%$ des cibles présentées pour la tâche de mouvement 3D de la main gauche ont été atteintes avec un R-ratio de $5.2 \pm 3.1$ tandis que le patient a touché $53 \pm 15\%$ des cibles avec un R-ratio de $5.2 \pm 3.1$ pour la tâche de mouvement 3D de la main droite. Pour les tâches de rotations 1D du poignet gauche et droit, le patient a touché en moyenne $95 \pm 8.2\%$ (avec un ratio de $3.6 \pm 3.3$).

Durant les expériences avec l'exosquelette, le patient a touché en moyenne plus de cibles que durant les expériences avec l'avatar virtuel. $69 \pm 13\%$ des cibles ont été touchées pour des tâches de mouvement 3D de la main gauche avec un R-ratio de $6.7 \pm 5.4$. Les performances de mouvement 3D de la main droite ont été estimées à un score de $65 \pm 29\%$ des cibles atteintes en moyenne et un R-ratio de $13 \pm 4.5$. Pour les tâches de rotation 1D du poignet gauche et droit, le patient a touché en moyenne $93 \pm 12\%$ (avec un ratio de $2.9 \pm 2.4$). Il est important de souligner qu'en ne considérant que les expériences réalisées entre 0 et 37 jours après la fin de la calibration du modèle, le nombre de cibles touchées avec la main gauche et la main droite atteint en moyenne les 83% (avec un R-ratio de $6.4 \pm 2.3$).

REW-MSLM a montré une stabilité inattendue au cours du temps pour les expériences de contrôle de l'avatar virtuel et de l'exosquelette. Bien que le contrôle du bras droit de l'exosquelette ait montré des résultats avec une grande variabilité, le contrôle du bras gauche est quant à lui resté très stable même après 167 jours sans recalibration du modèle.

*Pseudo-online performances de PREW-NPLS.*

Les performances de l'algorithme PREW-NPLS sont très dépendantes de l'hyperparamètre de pénalisation $\lambda$ choisi mais, en moyenne, les performances de décodage sont similaires à celle de REW-NPLS (voir légèrement supérieures). Cependant, les modèles estimés avec l'algorithme PREW-NPLS ont jusqu'à 80% des caractéristiques pénalisées (des électrodes fixées à zéro). Les performances obtenues avec REW-NPLS utilisant les 64 électrodes sont équivalentes aux performances obtenues par PREW-NPLS en se limitant à l'utilisation de 13 électrodes. Le faible nombre d'électrodes utilisées dans le modèle PREW-NPLS représente un réel avantage par rapport au modèle estimé avec l'algorithme REW-NPLS.

*Pseudo-online performances de APREW-NPLS.*

Les performances de APREW-NPLS sont très similaires à celle de l'algorithmes PREW-NPLS. Les performances de décodage ne sont pas meilleures que les performances de l'algorithme REW-NPLS mais elles sont obtenues avec des modèles utilisant très peu d'électrodes (jusqu'à 80% des électrodes ne sont pas utilisées pour certains modèles). A la différence de l'algorithme PREW-NPLS qui nécessite une phase de calibration par modèle, l'algorithme APREW-NPLS calcule tous les modèles pénalisés en même temps

durant la même phase de calibration afin d'estimer le meilleur hyperparamètre de pénalisation.

*Pseudo-online performances de H2M2.*

En comparaison avec $HMM_{limited}$, H2M2 montre de meilleurs résultats de classification pour tous les indicateurs statiques. La différence entre $HMM_{full}$ et H2M2 est moins évidente avec en moyenne une amélioration de $1.5\% \pm 0.83$. Cependant, les indicateurs de performances dynamiques montrent de grandes différences entre les trois modèles. H2M2 réduit fortement la latence de prédiction avec une diminution du temps de latence de 40% par rapport à $HMM_{limited}$ et 26% par rapport à $HMM_{full}$. Similairement, la durée d'un bloc d'erreur est réduite de 26% et 24% par rapport à $HMM_{limited}$ et $HMM_{full}$. Cependant la fréquence d'erreurs par minute augmente d'environ 30% avec l'algorithme H2M2.

**Chapitre 11: Discussion et perspectives**

Le modèle REW-MSLM développé dans le but d'un contrôle asynchrone d'effecteur multi-membres a été testé durant des études pseudo-online où les intérêts de cet algorithme par rapport à REW-NPLS ont été mis en avant. L'intégration de REW-MSLM dans la plateforme de décodage de l'essai clinique de Clinatec a permis d'évaluer les performances de cet algorithme et de tester en condition réelle l'adaptation incrémentale de REW-MSLM. Les modèles calibrés sur plusieurs séances avec cet algorithme ont montré une grande stabilité durant les expériences d'évaluation des performances sans pour autant avoir besoin de recalibration comme cela est souvent fait dans l'état de l'art. Les performances de décodage ont montré une forte stabilité sur plus de 6 mois d'expérience sans la moindre recalibration du modèle.

Dans une deuxième partie, des algorithmes ont été développés pour améliorer les sous-modèles d'experts et de gate de REW-MSLM. PREW-NPLS et APREW-NPLS permettent d'estimer des modèles utilisant uniquement un faible nombre de caractéristiques tandis que H2M2 réduit le temps de latence de transition entre les états discrets. Toutes les conclusions obtenues grâce aux études pseudo-onlines doivent être validées à l'aide d'expériences en temps réel.

De nombreuses améliorations et modifications des algorithmes APREW-NPLS et H2M2 sont envisageables et doivent être testées. Par exemple, dans APREW-NPLS, l'algorithme permettant de déterminer le meilleur modèle se base uniquement sur des critères de performances purs. Il est possible d'envisager un algorithme qui sélectionnera le modèle avec le meilleur équilibre entre sparsité et performances de décodage.

REW-MSLM est composé de plusieurs sous-modèles (gate et experts) pouvant être changés sans impacter le reste du modèle. Des études supplémentaires seront menées pour exploiter l'architecture de REW-MSLM. Des analyses plus poussées seront faites

pour extraire et interpréter les caractéristiques principales des signaux cérébraux permettant le décodage des différentes tâches réalisées par le patient lors du contrôle des effecteurs.

Durant les expériences en temps réel de contrôle 8D de l'avatar ou de l'exosquelette, le modèle REW-MSLM a été entrainé avec un nombre limité d'expériences. Il est probable que le temps d'entrainement du modèle ne soit pas suffisant et qu'une augmentation du temps de calibration pourrait mener à de meilleures performances de décodage. Des études supplémentaires seront menées afin de déterminer le temps et la fréquence des sessions d'entrainement les plus adaptées.

Les résultats ont été obtenus en effectuant des expériences avec la collaboration d'un seul patient tétraplégique. Il est donc nécessaire de généraliser les conclusions en effectuant des expériences avec d'autres patients. Un nouveau patient a été implanté le 19 Novembre 2019. Les performances de décodage de l'algorithme REW-MSLM seront évaluées avec ce nouveau patient.

Les performances obtenues avec l'algorithme REW-MSLM pour le décodage des signaux cérébraux du cortex moteur donnent des résultats poussant à diversifier l'utilisation de cet algorithme à d'autres domaines. Le décodeur peut être utilisé pour le décodage de signaux cérébraux provenant d'autres régions cérébrales ou dans le cadre d'autres applications médicales comme la détection de crises chez les patients épileptiques.

# Abstract

Brain-computer interfaces (BCIs) create a new communication pathway between the brain and an effector without neuromuscular activation. Among the various applications, functional compensation/restoration and rehabilitation of individuals suffering from severe motor disabilities has always been a focus for BCI research. Spectacular motor BCI milestones have been reached over the years. Nevertheless, many challenging aspects remain to translate BCI systems from laboratories directly into patients' home for daily life applications. Among the reported challenges to overcome, this PhD research focused on asynchronous high-dimensional multi-limb control of complex effectors, BCI tasks closer to real life behavior and robust neural signal decoding across time.

BCIs dedicated to real life applications should provide to the user the permanent control of an effector able to perform various tasks. Asynchronous BCI system acts as a stand-alone device which do not required any external help or cue to perform neural signal decoding in the opposite to synchronous BCI. Moreover, daily life actions often require multi-limb and/or more complex actions than the one tested during the commonly reported clinical trial experiments. BCI decoder dedicated to high-dimensional and/or multi-limb effector control could greatly increase the autonomy of patients suffering from severe motor disabilities. In order to control a high-dimensional effector, complex decoders are generally required. However, complex decoders are generally computationally heavy and may be difficult to implement online due to the real-time computation constrain. Therefore simpler linear models are commonly preferred for online applications although it might lead to lower decoding performance. Finally, the decoder calibration may benefit from the integration of the patient's neural feedback into the calibration procedure in order to create "human-in-loop" models. Calibration procedures leading to "humain-in-loop" models may improve the decoder robustness to the neural signal variation across time.

The presented PhD research were part of CEA-Grenoble\Leti\CLINATEC clinical trial: "BCI and tetraplegia" referred under the identifier: NCT02550522 in the publically accessible register ClinicalTrials.gov. A tetraplegic patient was bilaterally implanted with two wireless epirdural ECoG recording system, placed in front of the sensory motor cortex.

To respond to the cited challenges, based on the first successful long-term chronic exploitation of bilateral epidural ECoG recordings, innovative incremental adaptive decoders were designed, namely, the Recursive Exponentially Weighted Markov Switching Model (REW-MSLM), the Penalized Recursive Exponentially Weighted n-way Partial Least Square (PREW-NPLS), the Automatic $\lambda$ PREW-NPLS (APREW-NPLS) and the Hierarchical structured Hidden Markov Model (H2M2).

The REW-MSLM is based on a mixture of experts (ME) architecture composed of several continuous "expert" models decoding continuous movements from the neural signals and a dynamic "gating" model activating or inhibiting the expert continuous outputs. In the REW-MSLM, the continuous linear expert models were evaluated using the Recursive Exponentially Weighted N-way Partial Least Squares (REW-NPLS) algorithm whereas the gating model is a Hidden Markov Model (HMM). The switching of several linear models (experts) may explain complex non-linear behaviors with a moderate computational load compatible with real-time neural signal decoding applications.

The PREW-NPLS and APREW-NPLS are two adaptive group-wise sparse decoders designed to reduce the feature space dimension, to improve the decoding performance and be potentially integrated in the REW-MSLM algorithms as sparse gating and/or expert models. Finally, H2M2 dynamic classifier is a dynamic hierarchical model designed to enhance the gating model responsiveness.

Before their integration into online closed-loop BCI experiments, all the proposed algorithms were evaluated offline. The REW-MSLM offline study induced the benefits of using a gating model for the detection of the idle state and the different active states related to each limb movements. Moreover, the interest of cross-session training in order to obtain decoder more robust to brain and experimental condition variability was stressed. The PREW-NPLS algorithms highlighted that significantly higher decoding performances could be obtained with group-wise sparse models whereas APREW-NPLS preliminary study induced the possibility to set in an online procedure the hyperparameters that were commonly tuned during offline studies in other state of the art research. APREW-NPLS hyperparameter tuning is based on a reinforcement learning strategies in order to confront the decoding performance of several models during the incremental calibration procedure. Finally, the H2M2 offline preliminary study highlighted the interest of using a hierarchical classifier structure in the case of complex classification tasks to improve the decoder responsiveness.

The REW-MSLM decoder highlighted promising results in multiple offline pseudo-online studies. Therefore, REW-MSLM was integrated into the online BCI platform Adaptive Brain Signal Decoder (ABSD) to perform asynchronous multi-limb online closed-loop BCI experiments. Using the REW-MSLM algorithm, a tetraplegic patient performed the online high-dimensional control of an exoskeleton and a virtual avatar. The patient achieved asynchronous 8D continuous control including alternative 3D hand reaching tasks and 1D wrist rotation for each hand distributed into 5 discrete states: idle state, left and right hands translation and left and right wrists rotation. The discrete and continuous decoding performance highlighted stable results over 6 months of clinical experiments after the last model recalibration for both effectors. As an example, for the exoskeleton experiments carried out from 0 to 37 days after the last model calibration experiments, the decoding performance highlighted a hit score of $71 \pm 12\%$ and $99 \pm 2\%$ for the 3D hand translation and 1D wrist rotation tasks whereas the

dynamic classifier showed a five-state classification F-score performance of $77 \pm 14\%$. For the experiments achieved from 0 to 167 days, the decoding performance highlighted hit scores of $67 \pm 21\%$ and $93 \pm 12\%$ for the hand translation and wrist rotation tasks whereas the dynamic classifier demonstrated a five-state classification F-score performance of $75 \pm 12\%$.

All the pseudo offline and online closed-loop BCI experiments confirmed in a long term study that direct neural decoding is not limited to individual neuron action potential driven (MEA-based) BCIs and can be achieved from population neuron recordings, particularly from epidural ECoG neural signals. These results challenge the empirical evidence that population neuron recordings are limited to the control of fewer dimensions due to lower spatio-temporal resolution and the restricted number of possible somatotopic remapping combination.

The nearest perspectives of the presented study is to integrate the developed REW-MSLM, AREW-MSLM and H2M2 algorithms into the online BCI platform in order to evaluate the benefits of these algorithms during online closed-loop experiments.

The doctoral manuscript is organized in eleven chapters describing the BCI state of the art research, the designed algorithms and the obtained offline and online results. Specifically, the Chapter 1 introduces the principle requirements of a BCI system dedicated to daily life applications. The Chapter 2 presents in detail the "BCI and Tetraplegia" clinical trial from the paradigm of control to the training timeline. Chapter 3 reports the state-of-the-art BCI transducer (preprocessing, feature extraction and decoder). The REW-MSLM, PREW-NPLS, APREW-NPLS and H2M2 algorithms are detailed in the Chapter 4, 5 and 6 whereas experiments description, integration of the decoder into the CLINATEC online BCI platform and decoder performance evaluation are presented in the Chapter 7, 8 and 9 respectively. Chapter 10 clusters the results of each decoders. Finally, Chapter 11 regroups the results discussion, the benefit of this study, the implications of the presented results in the BCI fields and the research perspective. The manuscript is completed with an abstract of the PhD research, a glossary and a section describing the required mathematical operators, notations and notions for the understanding of the PhD research.

# Glossary

| | |
|---|---|
| AAR | Adaptive AutoRegressive |
| ABSD | Adaptive Brain Signal Decoder |
| ACC | Accuracy |
| ACSP | Adaptive Common Spatial Pattern |
| ADIM | Adaptive Information Matrix |
| AKF | Adaptive Kalman filter |
| ALS | Alternative Least Square |
| ANN | Artificial Neural Network |
| ANSM | Agence Nationale de Sécurité du Médicament et des produits de santé |
| APREW-NPLS | Automatic Penalized Recursive Exponentially Weighted N-way Partial Least Square |
| ARD | Automatic Relevance Determination |
| AS | Active State |
| ASD | Alternating Slice-wise Diagonalization |
| ASIA | American Spinal Injury Association Impairment |
| BCI | Brain Computer Interface |
| BIF | Best Individual Feature |
| CAR | Common Average Reference |
| CBLSU | Correlation-Based Least Squares Update |
| CCWT | Continuous Complex Wavelet Transform |
| CLDA | Closed-Loop Decoder Adaptation |
| CNN | Convolutional Neural Networks |
| CPP | Comité de Protection des Personnes |
| CRF | Conditional Random Fields |
| CSP | Common Spatial Pattern |
| dGNc | damped Gauss-Newton with compression |
| DoF | Degree of Freedom |
| DTLD | Direct Tri-Linear Decomposition |
| ECoG | Electrocorticographic |
| EEG | Electroencephalography |

| | |
|---|---|
| EGO | Efficient Global Optimization |
| EKF | Extended Kalman filter |
| EMC | EMY Motion Controller |
| EMG | Electromyography |
| EMM | EMY Motion Manager |
| EMY | Enhancing MobilitY |
| EOG | Electrooculography |
| ERP | Event-Related Potentials |
| ER-UCB bandit | Extreme-Region Upper Confidence Bound Bandit |
| FBCSP | Filter-bank CSP |
| FES | Functional Electrical Stimulation |
| FIR | Finite Impulse Response |
| Fn | False negatives |
| fNIRS | Functional Near-Infrared Spectroscopy |
| Fp | False positive |
| Fsc | F-score |
| GA | Genetic Algorithms |
| GAM | Generalized Additive Model |
| GLM | Generalized Linear Model |
| GMM | Gaussian mixture model |
| GP-DKF | Gaussian Process Regression Discriminative Kalman Filter |
| GPR | Gaussian Process Regression |
| HF | High Frequency |
| HFSP | High-Frequency Spike Power |
| HHMM | Hierarchical Hidden Markov Model |
| HMM | Hidden Markov Model |
| HMME | Hidden Markov Mixture of Experts |
| HOPLS2 | Higher Order Partial Least Squares |
| HOSVD | High-Order Singular Value Decomposition |
| HSMM | Hidden Semi-Markov Model |
| ICA | Independent Component Analysis |

| | |
|---|---|
| IIR | Infinite Impulse Response |
| IOHMM | Input-Output HMM |
| IS | Idle State |
| IWLDA | Importance Weighted Linear Discriminant Analysis |
| Kernel- ARMA | Kernel-AutoRegressive Moving Average |
| KF | Kalman Filter |
| kNN | K Nearest Neighbor |
| KTPLS | Kernel Tensor Partial Least Squares |
| LDA | Linear Discriminant Analysis |
| LFP | Local Field Potentials |
| LMP | Local Motor Potentials |
| LR | Logistic Regression |
| LS | least Squares |
| LSTM | Long Short Term Memory |
| MAB | Multi-Armed Bandit |
| MDM | Minimum Distance to the Mean |
| ME | Mixture of Experts |
| MEA | Microelectrodes Arrays |
| MEG | Magnetoencephalography |
| MIBIF | Mutual Information-based Best Individual Feature |
| MICI | Maximal Information Compression Index |
| ML | Maximum Likelihood |
| MLP | Multi-Layer Perceptrons |
| MRNN | Multiplicative Recurrent Neural Network |
| MSLM | Markov-Switching linear model |
| MUA | Multi-Unit Activity |
| NBPW | Naïve Bayesian Parzen Window |
| NCA | Neighborhood Component Analysis |
| NPLS | N-way Partial Least Square |
| ODT-SVM | Optimal Decision Tree based SVM |
| OLE | Optimal Linear Estimation |
| OLS | Ordinary Least Squares |

| | |
|---|---|
| ORICA | Online Recursive Independent Component Analysis |
| PARAFAC | PARAllel FACtor analysis |
| PCA | Principal Component Analysis |
| PLS | Partial Least Square |
| PMF3 | Positive Matrix Factorization for 3-way arrays |
| PNPLS | Polynomial Penalized NPLS |
| PPD | Piecewise Probabilistic Decoding |
| PPF | Point Process Filter |
| PREW-NPLS | Penalized Recursive Exponentially Weighted N-way PLS |
| PSO | Particle Swarm Optimization |
| PVA | Population Vector Algorithm |
| QDA | Quadratic Discriminant Analysis |
| QPFS | Quadratic Programming Feature Selection |
| RE | Representation Entropy |
| Refit-KF | Recalibrated Feedback Intention Kalman Filter |
| REMBO | Random Embedding Bayesian Optimization |
| REW-NPLS | Recursive Exponentially Weighted N-way Partial Least Square |
| RF | Random Forest |
| RNN | Recurrent Neural Network |
| RPLS | Recursive PLS |
| RV | Recursive-Validation |
| SBFS | Sequential Backward Floating Selection |
| SeROA | Sequential Rank-One Approximation |
| SFFS | Sequential Floating Forward Selection |
| SMBO | Sequential Model-Based Optimization |
| SMAC | Sequential Model-based Algorithm Configuration |
| SMC | Sensory Motor Cortex |
| SNPLS | Sobolev NPLS |
| SSD | Spatio-Spectral Decomposition |
| SSVEP | Steady State Visual Evoked Potentials |
| STFT | Short Time Fourier Transform |
| SUA | Single-Unit Activity |

| | |
|---|---|
| SVM | Support Vector Machine |
| SVR | Support Vector Machine Regression |
| SKF | Switching Kalman Filter |
| SWATLD | Self-Weighted Alternating Tri-Linear Decomposition |
| SWLDA | Step-Wise LDA |
| Tn | True negatives |
| Tp | True positive |
| TPE | Tree-structure Parzen Estimator |
| UHF | Ultra-High Frequency |
| UKF | Unscented Kalman Filter |
| VEP | Visual Evoked Potentials |
| WHO | World Health Organization |
| WIMAGINE | Wireless Implantable Multi-channel Acquisition system for Generic Interface with NEurons |
| WISCI | Wireless Implant Software Control Interface |
| WMV | Weighted Majority Voting |
| WT | Wavelet transforms |

# MATHEMATICAL OPERATORS AND NOTATIONS

Terminologies and mathematical basis required for the understanding of the manuscript are summarized in this section.

Real value scalar are denoted by lower-cases italics letters $x \in \mathbb{R}$, vectors by bold lower-case letter $\mathbf{x} \in \mathbb{R}^{I_1}$, matrices by bold upper-case letters $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ and higher order array named tensors are noted with underlined bold upper-case letters $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ where $I_1, I_2, \ldots, I_N$ are the space dimensions.

A tensor is a multidimensional array. A real N-way, Nth order or N-mode tensor is noted $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, where $I_n \in \mathbb{N},\ n = 1, \ldots, N$ are dimensions [Cichocki et al., 2015] [Eliseyev et al., 2017] [Kolda and Bader, 2009] and $x_{i_1,\ldots,i_N} \in \mathbb{R},\ 1 \leq i_j \leq I_n,\ j = 1, \ldots, I_n$ are the elements of $\underline{\mathbf{X}}$. The total dimension of $\underline{\mathbf{X}}$ is $\prod_{n=1}^{N} I_n$. Vectors (e.g. $\mathbf{x} \in \mathbb{R}^{I_1}$) and matrices (e.g. $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$) are special cases of tensors with one and two modes respectively [Bro, 1998] [Kolda and Bader, 2009]. 3-way tensor example is represented in the Figure 1-1A and rank one tensor example is shown in Figure 1-1B.

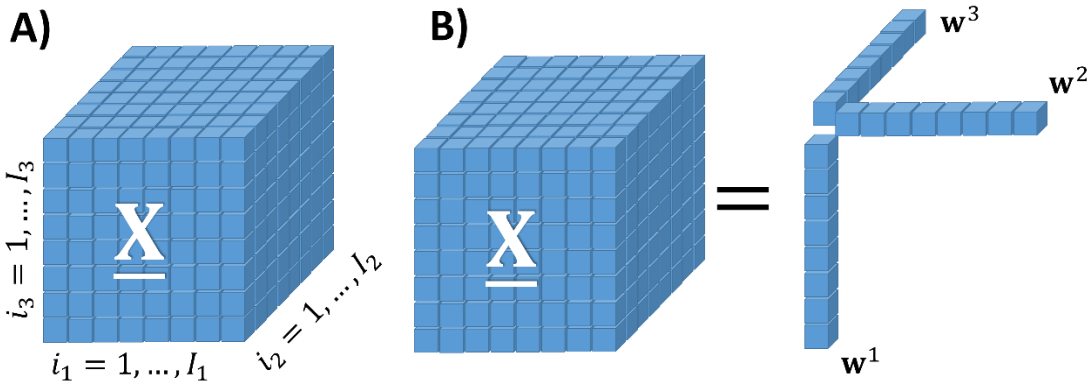



Figure 1-1: Definiton of a tensor. A) Example of third order tensor: $\underline{\boldsymbol{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. B) Rank one 3-mode tensor example.

**Definition of the norm**

The tensor Frobenius norm of $\underline{\mathbf{X}}$ is defined as

$$\|\underline{\mathbf{X}}\| = \sqrt{\sum_{i_1=1}^{I_1} \ldots \sum_{i_N=1}^{I_N} x_{i_1,\ldots,i_N}^2},$$

which is analogue of standard definitions for matrices (Frobenius norm) and vectors (Euclidian norm). In the manuscript, $\|\cdot\|$, always referred to $L_2$ norm (Frobenius, Euclidian norm depending on the variable dimensions).

**Tensor unfolding transformation**

Transformation of a tensor into a matrix is named unfolding, flattening or matricization. This process flattens a tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ into a matrix along a specific dimension (or mode) $n$ and will be noted as $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \ldots I_{n-1} I_{n+1} \ldots I_N}$ [Cichocki et al., 2015] [Kolda and Bader, 2009]. An example of the three possible unfolding transformation of a third order tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is presented in the Figure 1-2. As mentioned, unfolding can be

performed following each tensor dimension leading in this three order tensor example to the three matrices $\boldsymbol{X}_{(1)} = \mathbb{R}^{I_1 \times I_2 I_3}$, $\boldsymbol{X}_{(2)} = \mathbb{R}^{I_2 \times I_1 I_3}$ and $\boldsymbol{X}_{(3)} = \mathbb{R}^{I_3 \times I_1 I_2}$.

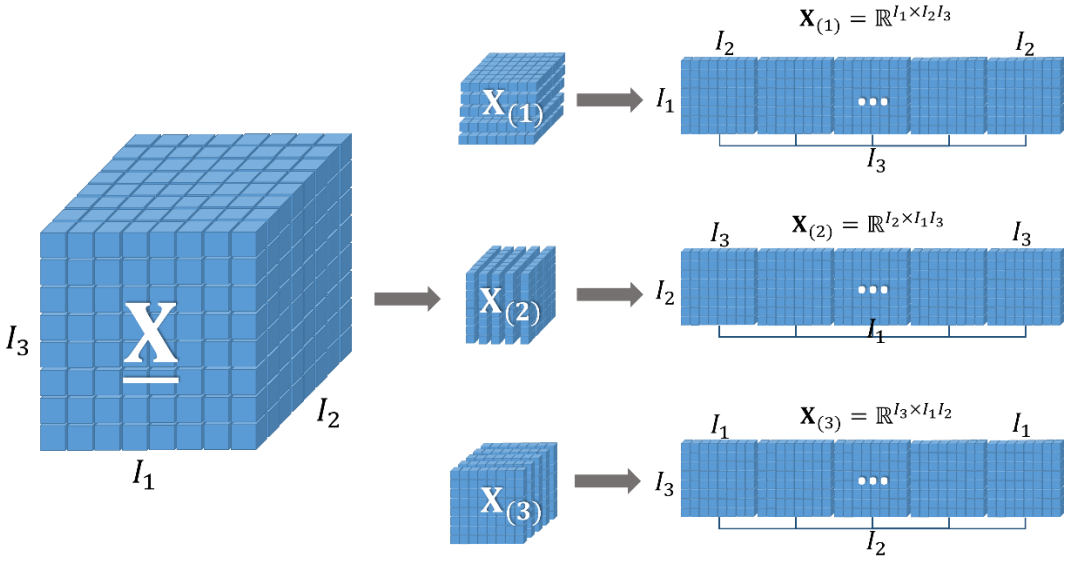



Figure 1-2: Example of third-order tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ unfolding along the 3 possible modes leading to the three possible matrices $\boldsymbol{X}_{(1)} = \mathbb{R}^{I_1 \times I_2 I_3}$, $\boldsymbol{X}_{(2)} = \mathbb{R}^{I_2 \times I_1 I_3}$ and $\boldsymbol{X}_{(3)} = \mathbb{R}^{I_3 \times I_1 I_2}$.

**Outer Product**

Let $\underline{\boldsymbol{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ be a N-order tensor and $\mathbf{w}^n \in \mathbb{R}^{I_n}$ $(n = 1 \ldots N)$ $N$ vectors, the outer product (noted " $\circ$ ") of $\mathbf{w}^n$ is defined as:

$$\underline{\boldsymbol{X}} = \mathbf{w}^1 \circ \mathbf{w}^2 \circ \ldots \circ \mathbf{w}^N$$
$$with\ x_{i_1,\ldots,i_N} = w_{i1}^1 w_{i2}^2 \ldots w_{iN}^N.$$

$\underline{X}$ is a rank one tensor as it can be expressed as the outer product of $N$ vectors.

**N-mode product**

The n-mode product (noted " $\times_n$ ") between a tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ and a matrix $\mathbf{Y} \in \mathbb{R}^{K_n \times I_n}$ is noted $\underline{\mathbf{C}} = \underline{\boldsymbol{X}} \times_n \mathbf{Y}$ with $\underline{\mathbf{C}} \in \mathbb{R}^{I_1 \times \ldots \times I_{N-1} \times K_n \times I_{N+1} \times \ldots \times I_N}$ and

$$\underline{\boldsymbol{C}}_{i_1 \ldots i_{n-1} k_n i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} y_{k_n i_n}.$$

The n-mode product $\underline{\mathbf{C}} = \underline{\boldsymbol{X}} \times_n \mathbf{Y}$ can be expressed using the unfolded tensor expression:

$$\mathbf{C}_{(n)} = \mathbf{Y}\mathbf{X}_{(n)}.$$

**Kroenecker product:**

Let $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ and $\mathbf{Y} \in \mathbb{R}^{J_1 \times J_2}$ be two matrices, the Kronecker product of these two matrices noted $\mathbf{C} = \mathbf{X} \otimes \mathbf{Y}$ with $\mathbf{C} \in \mathbb{R}^{(I_1 J_1) \times (I_2 J_2)}$ is defined by

$$\mathbf{C} = \mathbf{X} \otimes \mathbf{Y} = \begin{bmatrix} x_{11}\mathbf{Y} & x_{12}\mathbf{Y} & \cdots & x_{1I_2}\mathbf{Y} \\ x_{21}\mathbf{Y} & x_{22}\mathbf{Y} & \cdots & x_{2I_2}\mathbf{Y} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I_1 1}\mathbf{Y} & x_{I_1 2}\mathbf{Y} & \cdots & x_{I_1 I_2}\mathbf{Y} \end{bmatrix} = \left[\mathbf{x}_1 \otimes \mathbf{y}_1 \; \mathbf{x}_1 \otimes \mathbf{y}_2 \; \mathbf{x}_1 \otimes \mathbf{y}_3 \; \ldots \; \mathbf{x}_{I_2} \otimes \mathbf{y}_{J_2 - 1} \; \mathbf{x}_{I_2} \otimes \mathbf{y}_{J_2}\right].$$

**Khatri-Rao product:**

The Khatri-Rao product is the column wise Kronecker product. Let $\mathbf{X} \in \mathbb{R}^{I_1 \times K}$ and $\mathbf{Y} \in \mathbb{R}^{J_1 \times K}$ be two matrices, The Khatri-Rao product is denoted $\mathbf{C} = \mathbf{X} \odot \mathbf{Y}$ with $\mathbf{C} \in \mathbb{R}^{(I_1 J_1) \times K}$ and is defined by

$$\mathbf{C} = \mathbf{X} \odot \mathbf{Y} = \left[\mathbf{x}_1 \otimes \mathbf{y}_1 \; \mathbf{x}_2 \otimes \mathbf{y}_2 \; \ldots \; \mathbf{x}_K \otimes \mathbf{y}_K\right].$$

**Khatri-Rao product property**

These unfolded matrices and tensors products have many properties [Kolda and Bader, 2009]. For the next chapters, the following property is mentioned:

$$(\mathbf{X} \odot \mathbf{Y})^{\dagger} = \left(\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right) * \left(\mathbf{Y}^{\mathrm{T}}\mathbf{Y}\right)\right)^{\dagger} (\mathbf{X} \odot \mathbf{Y})^{\mathrm{T}}, \tag{0.1}$$

where $\mathbf{X} \in \mathbb{R}^{I_1 \times K}$ , $\mathbf{Y} \in \mathbb{R}^{J_1 \times K}$ and $\mathbf{X}^{\dagger}$ is the Moore-Penrose pseudoinverse of $\mathbf{X}$ [Kolda and Bader, 2009]

**The elementwise matrix (Hadamard) product**

Let $\mathbf{X}$ and $\mathbf{Y} \in \mathbb{R}^{I_1 \times I_2}$ two matrices of the same dimensions, the elementwise matrix product noted $\mathbf{C} = \mathbf{X} * \mathbf{Y}$ is defined by

$$\mathbf{C} = \mathbf{X} * \mathbf{Y} = \begin{bmatrix} x_{11}y_{11} & x_{12}y_{12} & \cdots & x_{1I_2}y_{1I_2} \\ x_{21}y_{21} & x_{22}y_{22} & \cdots & x_{2I_2}y_{2I_2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I_1 1}y_{I_1 1} & x_{I_1 2}y_{I_1 2} & \cdots & x_{I_1 I_2}y_{I_1 I_2} \end{bmatrix}.$$

# Table of contents

# Table of figures

# Tables

Chapitre 1

# INTRODUCTION

## Contents

Brain-Computer Interfaces (BCIs) or brain machine interfaces (BMIs) are systems allowing the control of external devices thanks to the brain neural signals without using the natural neuromuscular activation. BCIs create a new communication pathway between the brain and an effector [Vidal, 1973] [Lebedev and Nicolelis, 2006] [Mak and Wolpaw, 2009]. BCIs have been investigated for many applications such as communication, neurorehabilitation, drowsiness monitoring, computer gaming [Lebedev and Nicolelis, 2017] [Lee et al., 2019] [Ramadan et al., 2015], etc. Nevertheless, BCIs were particularly developed for patient suffering from severe motor disabilities. In this domain, BCIs are referred as motor BCI. Motor dysfunctions may be the consequences of neurological/neuromuscular disorders as spinal-cord injury [W. Wang et al., 2013], hemiplegia, brain stroke [Hochberg et al., 2012], amyotrophic lateral sclerosis (ALS), cerebral palsy [Lebedev and Nicolelis, 2006] [Mak and Wolpaw, 2009] or brain, muscular diseases as the Duchenne muscular dystrophy (DMD) [Utsumi et al., 2018]. BCIs have been used for functional rehabilitations or robotic assistance of individuals suffering from muscular activity deterioration to help patients and brain plasticity to recover [Carvalho et al., 2019] [Donati et al., 2016] [López-Larraz et al., 2018] [Mak and Wolpaw, 2009] or complete functional compensation for patients enable to perform any muscular activation [Lebedev and Nicolelis, 2017, 2006] [Mak and Wolpaw, 2009] with numerous effectors such as drones [LaFleur et al., 2013], wheelchairs [Huang et al., 2012] [Leeb et al., 2007] [Li et al., 2013], robotic limbs [Hochberg et al., 2012], or exoskeleton [Eliseyev et al., 2014] [Morinière et al., 2015]. Spinal cord injury affects 17 000 people per year in United States with an estimated prevalence of 280 000 injured persons [Eckert and Martin, 2017] whereas stroke is one of the main causes of long-term motor disability worldwide, and generally drives to functional deficits in motor control [Langhorne et al., 2011] [López-Larraz et al., 2018]. Considering the possible benefits for the disabled patient, the development of new technologies related to the implantable recording devices, the artificial intelligence and the exponential increase of computational power, BCIs translated from niche area of research to a broad and complicated field of research to analyze (Figure 1-1).

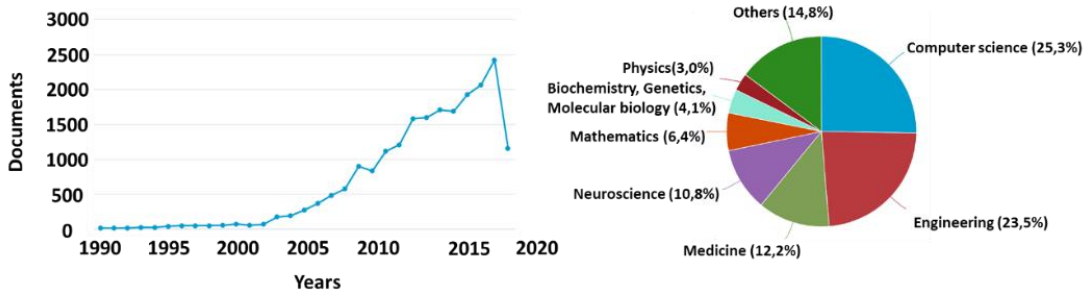



Figure 1-1 : Evolution of the BCI research field across years and domain of application. Statistics were extracted from Scopus website with the key words "BCI", "brain computer interface" and "Brain machine interface" on August 2020. The "Others" category clusters fields with less than 2% of the articles

In the next section, basic neuroscience notions are provided to understand the brain activity modulations and their possible exploitation to control various systems using BCI and motor BCI.

## 1.1. Basic Neuroscience for BCI

The nervous system, particularly the brain, is composed of nerve cells named neurons and supporting cells called neuroglia. Neurons are specialized in electrical signaling to interact with each other and transfer/process information whereas supporting cells do not produce electrical signals but assist the nerve cells [Purves et al., 2004]. Many diverse neurons exist to handle different functions, nevertheless, they all generally have the same basic components which are the dendrites, the soma and the axon. The dendrites retrieve information from the chemical components released in the synapses by other neurons. The information conveyed in the synapses up to the neuronal dendrites is integrated and processed at the origin of the axon [Purves et al., 2004]. The axon hillock, the base of the axon in the soma, generates (or not) the fundamental unit of electrical information called the action potential that carries signals at high speed across the axon to the axon termination named telodendria. In the axon termination, synaptic contacts are made with other neurons. The information is transmitted to other neurons thanks to chemical components released in the synapse. The concentration of the released chemical component is dependent on the received electrical information [Purves et al., 2004] (Figure 1-2). An action potential is a brief (1ms) all-or-nothing change of the neuron transmembrane potential from negative to positive which is triggered if an input stimulus recorded at the dendrites is above an activation threshold [Purves et al., 2004]. The action potential amplitude, shape and maximal firing rate are fixed properties dependent on the neurons type (excitatory, inhibitory etc).

The human brain (encephalon) is part of the central nervous system (CNS) with the spinal cord and is estimated to contain 100 billion neurons with complex interactions called neural networks (or neural circuits) and several times as many supporting cells [Purves et al., 2004]. Neural circuits process input information of the brain and provide the foundation of sensation, perception and behavior. Encephalon is composed of the brainstem (midbrain, the pons, the medulla oblongata), the diencephalon and the telencephalon.

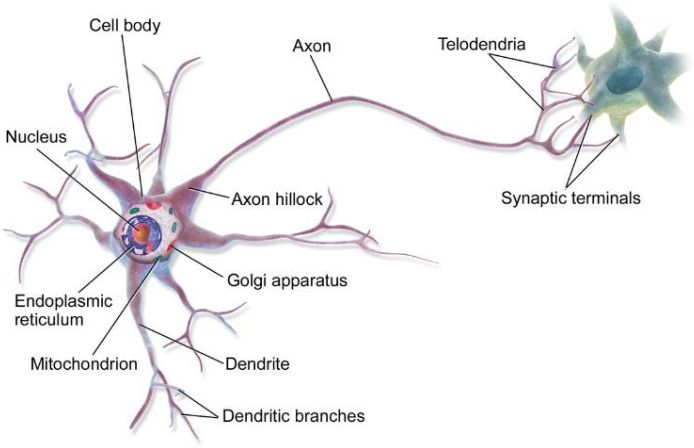



Figure 1-2: Illustration of the common neuron structure ["Neuron," 2020]. The neuron is a nerve cell which transmits electrical information through the brain. Previous neuron send chemical components released in the synapse. The receptors of dendritic branches of the postsynaptic cell received the chemical component which create a change of potential in the postsynaptic cell membrane. Depending on the chemical components transmitted as well as the postsynaptic cell receptor type, the probability of generating an action potential in the axon hillock is increased or decreased [Purves et al., 2004]. If an action potential is generated, it is sent through the axon to the synaptic terminals to transmit the information to the next neurons.

The brainstem ensures numerous major functions. It is the link between the spinal cord and the rest of the central nervous system, regroups the nuclei of the cranial nerves and is involved in cardiovascular, respiratory and consciousness control. The diencephalon clusters the basal ganglia, the thalamus, hypothalamus and plays a critical role in sensory information transmission, auditory, visual, sensory, motor and emotional information [Kandel and Tollet, 2016] [Purves et al., 2004]. The telencephalon is composed of the two cerebral hemispheres connected through white matter (as the corpus callosum). Telencephalon anatomy formed of crests (gyri) and grooves (sulci) can be separated into the occipital, temporal, parietal and frontal lobes with the central sulcus splitting the brain in two between the parietal and frontal lobes (Figure 1-3) [Kandel and Tollet, 2016] [Purves et al., 2004]. All the lobes have specific functions, occipital lobe functions are oriented on the processing of visual stimuli, the temporal lobe is engaged in the auditory functions, the parietal lobe involves proprioceptive and sensory information whereas the frontal lobe is dedicated to the motor and cognitive functions. The motor cortex (following the central sulcus) is dedicated to voluntary movements. The firing rate of the motor cortex neurons is related to low- and high-level information of attempted and realized movements.

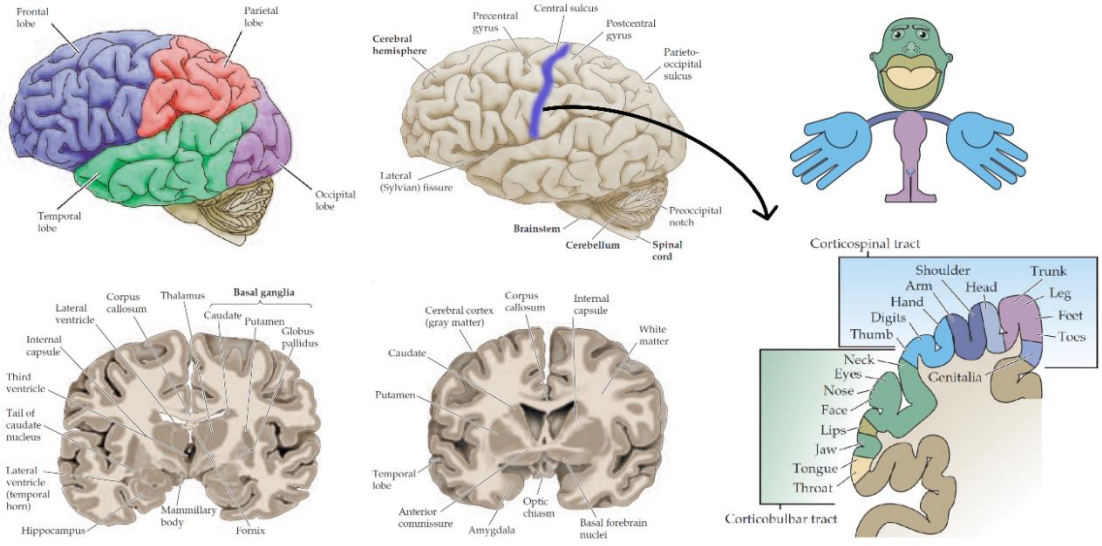



Figure 1-3: Brain anatomy and functional topographic organization of the motor cortex [Purves et al., 2004].

The neural system and, particularly the brain have a functional topographic organization (somatotopic). The spatial distribution of neurons in the cortical regions of the brain is dependent on their functions. For the motor cortex, studies highlighted the contralateral somatotopic organization of human motor control. Moreover, it was proven that neurons action potential firing rate of premotor and motor cortex are directly tuned depending on arm intended direction and action [Georgopoulos et al., 1986] [Georgopoulos and Carpenter, 2015]. Movements generate a modulation of the brain neural activity which can be recorded through various recording systems. The cortical neurons (pyramidal neurons) of the brain are aligned and interconnected into 6 layers which are firing action potentials. Their dendritic trees and axons are in parallel to each other and perpendicular to the cortical surface [Lopes da Silva, 2013]. During a movement, due to the neurons alignments and pyramidal synchronous action potentials patterns, the signals recorded by the population recording system which are equal to the summation of the cortical neurons signals lead to neural signal modulation in specific frequencies. In general, the recorded brain signals are divided into bandwidths following the notation [Lopes da Silva, 2013] [Schaeffer, 2017]:

- Delta (δ) for [0.5 Hz - 4 Hz]
- Theta (*θ*) for [4 Hz - 7 Hz]
- Alpha (α) for [8 Hz - 13 Hz]
- Mu (μ) for [8 Hz - 13 Hz] signals frequency in the central and the parietal areas
- Beta (β) for [14Hz – 30 to 35 Hz]
- Gamma (ɣ) for signal frequencies above 30 to 35 Hz.

In the case of motor control pathways degradation, the intended movement may not be transmitted to the muscles (due to spinal cord damage, or motor neuron degradation, etc.). However, the action potentials of neurons in the motor cortex are still firing/activated. Motor BCIs aim to record these neural activities and translate it into order for an effector (an actioner) that will substitute the non-realized movement by an action of the effector. To perform, BCI systems are composed of basic elements described in the next section.

## 1.2. BCI components

A Brain computer interface is built on three fundamental components that are common to every system, namely the acquisition system, the transducer and the effector. A BCI is a system allowing the interaction of a patient with the environment through the control of an effector using an acquisition system to record the brain neural signals and a signal-processing block to translate the neural activity variations into commands to the effector. Finally, the information (success, failure, etc.) about the actions carried out by the effector are sent to the patient through a feedback (visual, auditory, proprioceptive, etc.) (Figure 1-4) [Lebedev and Nicolelis, 2017] [Mak and Wolpaw, 2009]. Sensory feedback is a key element of the BCIs dedicated to daily life applications which must be considered.

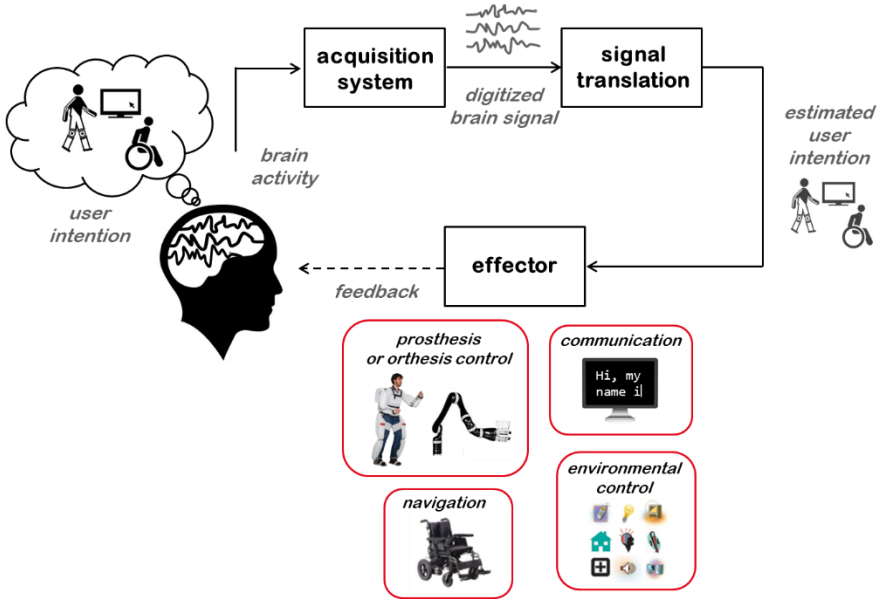



Figure 1-4: Common Brain Computer Interface (BCI) architecture. Brain neuronal activity is recorded using diverse acquisition systems and is treated to translate it to orders executed by the effector. A feedback (visual, auditory, proprioceptive, etc.) is provided to patient through effectors response.

### 1.2.1. Acquisition systems

The acquisition system aims to record the brain activity from neurophysiologic signals or using indirect measures. Acquisition systems attempt to optimize the ratio between invasiveness and resolution. Therefore, numerous devices more or less invasive have been designed with various spatio-temporal resolutions (Figure 1-5) [Kim et al., 2015] [Mak and Wolpaw, 2009] [Nicolas-Alonso and Gomez-Gil, 2012] [Schaeffer and Aksenova, 2018] [Waldert et al., 2009]. Invasive systems record information closer to the sources providing simpler signals to analyze compared to less invasive devices which record neurons activity integrated/filtered and spread into a large area due to multiple layers protecting the brain (Pia, Arachnoid, dura matter, skull and scalp). This section is reviewing the different recording systems from the more to the less invasive techniques.

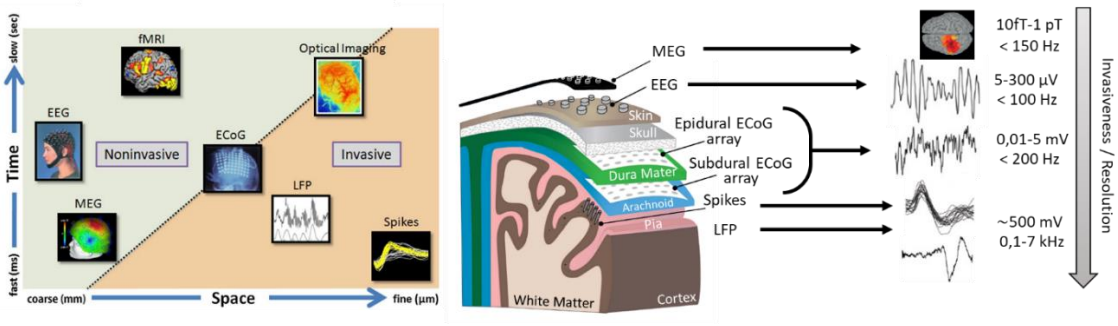



Figure 1-5: Invasiveness and resolution of BCI recording systems [Jorfi et al., 2015] [Kim et al., 2015]. Possible recording systems cover a wide range of invasiveness going from highly invasive systems to recording systems without direct interaction with the patient. A trade-off between invasiveness and resolution is to consider to select the more adapted recording system.

**Microelectrode arrays (MEA)** are intracortical recording systems for BCI applications which are implanted to directly sample the activity of the neurons from small local brain area. As an example, the Utah array is composed of hundreds of shanks distributed in few $mm^2$ with an electrode length between 0.5 and 1.5mm, 400µm pitch and spaced from each other of around 0.4mm [Gunasekera et al., 2015] [Jorfi et al., 2015] [Stieglitz et al., 2009]. MEA recording system provides high spatial and temporal signal resolution and allows recording Single-Unit Activity (SUA) and Multi-Unit Activity (MUA) which provide a direct representation of the neuronal activity near the electrodes. Depending on the signal processing of MEA neural signals, the Local Field Potentials (LFP) can also be extracted. The LFPs are the signals recorded from the summation of synchronous action potentials from thousands of neuron populations close to the electrode tip [Lebedev and Nicolelis, 2017] [Schaeffer and Aksenova, 2018]. Disabled patients controlled in real-time complex effector up to 10 Degrees of freedom (DoF) based on MEA neural signal decoding [Collinger et al., 2013] [Hochberg et al., 2012] [Wodlinger et al., 2015]. However, this control required frequent recalibration (each session or daily recalibration) due to

highly intra and inter-day signal instabilities [Perge et al., 2013]. Few Long-term studies were reported with long-term implantation of MEA up to 1000 days [Jarosiewicz et al., 2015] [Milekovic et al., 2018] [Schwemmer et al., 2018] [Simeral et al., 2011], nevertheless, degradation of the recorded signals related to the loss of electrodes or recorded neurons is recurrent and is an important limitation for clinical and daily life application. The implantation of a MEA device leads to acute damage, which generates body immune system reaction to protect brain cells and repair the damaged tissues. This reaction generates a glial encapsulation of the intracortical microelectrodes which may deteriorate the electrodes and significantly reduce the amplitude of the recorded spikes [Gunasekera et al., 2015] [Jorfi et al., 2015] [Kozai et al., 2015] [Marin and Fernandez, 2010] [Moran, 2010] [Murphy et al., 2016] [Ward et al., 2009]. This biocompatibility issue and degradation of the electrodes imply difficulties for chronic/long term SUA and MUA recordings and is one the main research topics in MEA recording system domain [Jorfi et al., 2015]. LFP is less sensitive to the spike amplitude reduction by the nature of the signal which integrates the behavior of many more neurons [Moran, 2010]. Finally, to this day, MEA recording systems applied to BCI and motor BCI are limited to wired systems due to the massive stream of data recorded at a high sampling rate. Wired-systems are not suited to daily life applications and enhance the possible risk of infection.

**Electrocorticographic arrays (ECoG)** are grids of plane macro-electrodes (diameter in mm). The grid is in the order of the $cm^2$ and the electrodes are distanced by few mm. ECoG are implanted under the skull either above the Dura Matter (epidural ECoG) or below (subdural ECoG). ECoG can be considered as "semi-invasive" recordings as the patient underwent a craniotomy but the brain integrity is not affected by the operation [Lebedev and Nicolelis, 2017] [Rak et al., 2012]. Due to the dimensions of the electrodes and the distance between the electrodes and the neurons, ECoG is limited to the neural population recording of the superficial layers of the cortex. ECoG recorded signals represent the sum of the synchronous extracellular potential of the neurons [Buzsáki et al., 2012] [Waldert et al., 2009]. ECoG recording systems is a good trade-off between invasiveness and signal to noise ratio, it contains broader bandwidth, higher amplitude and fewer artifacts than non-invasive recording systems whereas clinical risk is reduced compared to MEA even though ECoG has lower spatial and temporal resolution [Leuthardt et al., 2006, 2004] [Volkova et al., 2019]. Moreover, ECoG recording systems highlighted real-time motor BCI experiments with up to 3 DoF control by tetraplegic patients [Degenhart et al., 2018] [W. Wang et al., 2013] as well as long-term stability properties [Leuthardt et al., 2004] [Volkova et al., 2019] in preclinical [Chao et al., 2010] [Costecalde et al., 2017] [Sauter-Starace et al., 2019] and clinical studies [Benabid et al., 2019] [Nurse et al., 2018]. Nevertheless, clinical BCI research based on ECoG array implantation was seldom realized for BCI purposes. In most of the studies, ECoG arrays were implanted to detect the epileptic sources of patient before surgery which limits the relevance of the study for disabled patients and the duration of the reported state of the

art experiments with at most 40 days between the implantation and explantation of the ECoG grid.

**Electroencephalography (EEG)** is a non-invasive recording device based on a helmet/headset with a large number of electrodes (64 to 256) placed on the surface of the scalp. EEG signals integrate the extracellular currents of a large neural population over a large region (10 $cm^2$ or more) [Buzsáki et al., 2012] [Rak et al., 2012] [Waldert et al., 2009]. Similarly to ECoG recordings, EEG is limited to the recording of the low-pass filtered synchronous extracellular current activity of neurons at the surface of the motor cortex. EEG is the most widely used brain activity recording system and presents many advantages compared to the previously mentioned recording systems. The non-invasiveness, ease of use and low cost of EEG recording systems tend to apply it for research with a humongous number of studies in the BCI and the motor BCI field in the past years [Lebedev and Nicolelis, 2017] [Lotte et al., 2018]. Additionally, EEG recording was used for epileptic, sleep or brain disorder detection [Rak et al., 2012]. Nevertheless, EEG-based BCIs present several limitations compared to more invasive neural recording systems. EEG recording devices have limited spatial resolution, lower than more invasive systems and restricted frequency resolution with a bandwidth from 0 to around 100 Hz with the major information between 0 and 40 Hz due to the low pass-filtering of the multiple layers protecting the brain (Pia, Arachnoid, Dura Matter, skull, skin) [Lebedev and Nicolelis, 2006] [Schalk and Leuthardt, 2011] [Waldert et al., 2009]. Recorded signals are small (from 10 μV to 100 μV), noisy and highly artifact sensitive from different sources (eye movements, muscle activity, heart pulses, power line at 50 Hz) [Lebedev and Nicolelis, 2017] [Rak et al., 2012]. Moreover, EEG recorded signals are not stable in time and showed high inter and intra subject variability [Clerc et al., 2016a]. Finally, EEG headset commonly required a skilled person for device preparation and electrodes positioning which is a time-consuming and complicated procedure.

In summary, MEA provides the best spatial resolution allowing to directly record the action potentials of the neurons whereas ECoG and EEG are limited to population neurons recordings equal to the sum of the individual neuron action potentials. Nevertheless, MEA recordings are highly invasive recording systems with a higher risk of infection, important immune response and low temporal stability due to the constant modification of the neurons activity. EEG recording system, while the spatial and time-frequency resolution is bad compared to the other presented recording systems, is non-invasive and is widely used in various BCI experiments. ECoG recording system is a trade-off between MEA and EEG. ECoG has better recording properties for BCI application than EEG and is less invasive than MEA, particularly the epidural ECoG recording systems. EEG, ECoG and MEA are the recording systems generally used in BCI applications. Each of them has strengths and weaknesses [Nicolas-Alonso and Gomez-Gil, 2012] that are summarized in Figure 1-6. However, other types of recording systems have been developed or adapted to BCI applications.

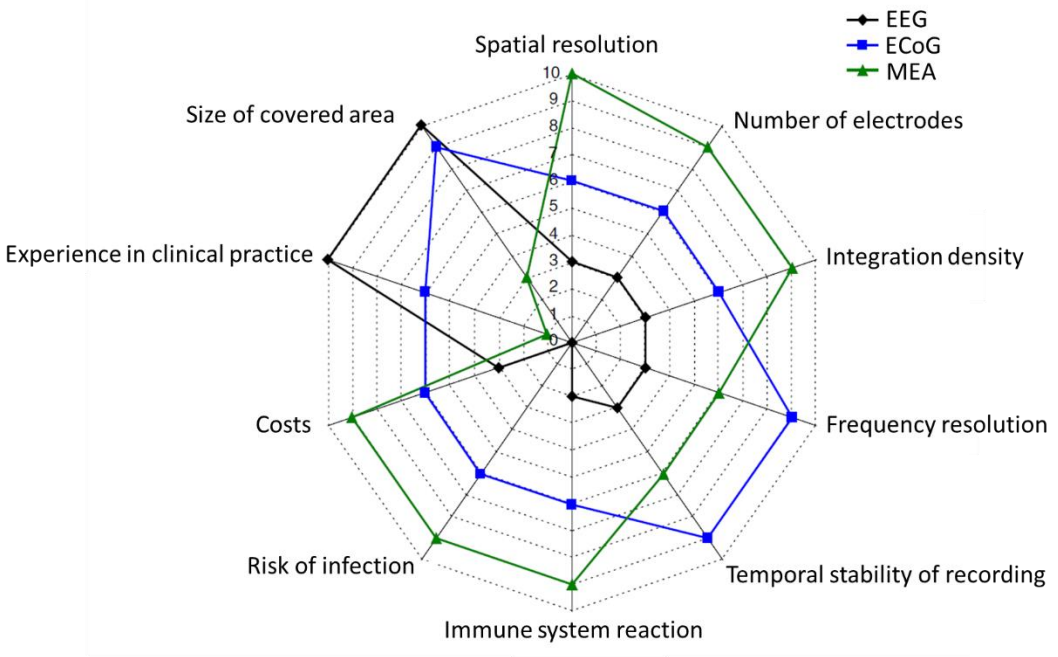



Figure 1-6: Summary of the advantages and disadvantages of the BCI recording systems. EEG, ECoG and MEA are the commonly used systems to record neural signals for clinical and preclinical BCI and motor BCI experiments. The figure is extracted from [Stieglitz et al., 2009]

**Magnetoencephalography (MEG)** is a non-invasive recording technique which measures at distance, out of contact with the patient, the brain magnetic field intensity and its variation at few centimeters above the skin. It uses arrays of superconducting quantum inference devices (SQUIDs) or spin-exchange-relaxation-free magnetometer (SERF) to record the small magnetic fields (around 10 fT to 1 pT) generated by the intracellular currents flowing through the cortical pyramidal neurons dendrites [Buzsáki et al., 2012] [Hämäläinen et al., 1993] [Lebedev and Nicolelis, 2017] [Lopes da Silva, 2013] [Waldert et al., 2009]. MEG recordings present several advantages compared to the other recording systems. Firstly, as MEG is a measurement of the magnetic field at distance from the patient, there is no contact between the patient and the recording system. MEG signals are less dependent on the extracellular space conductivity (skin, muscles, Dura Matter, etc.) and have a better signal-to-noise ratio than EEG specifically in the brain high frequency bandwidth (above 30 Hz) [Buzsáki et al., 2012] [Yelisyeyev, 2011]. MEG system has a better spatial and frequency (above 1ms) resolution than EEG [Buzsáki et al., 2012] [Hämäläinen et al., 1993]. Nevertheless, MEG recording system presents numerous practical drawbacks. MEG instrumentation is very expensive, is cumbersome requiring a lot of spaces to install the system (cooling device, magnetic recording device, etc.), required long patient preparation and the patient need to stay immobile during experiments [Hämäläinen et al., 1993] [Rak et al., 2012]. It is good to notice that MEG recording system can be used to realized pre-surgical studies and evaluate the potential

performance of ECoG recording systems or localize the more optimal localization of invasive/semi-invasive recording systems [Benabid et al., 2019] [Fukuma et al., 2016]. MEG neural signal processing was reported during BCI and motor BCI experiments [Jerbi et al., 2011] [Lebedev and Nicolelis, 2017] [Waldert et al., 2008].

**Functional Near-Infrared Spectroscopy (fNIRS)** is an indirect brain activity measurement system. Brain neural activity leads to variation in brain blood oxygenation (hemodynamic responses) with increases in the brain oxygen concentration in the blood (oxyhemoglobin HbO) or decreases of the oxygen concentration (deoxyhemoglobin HbR) [Sirpal et al., 2019] [Waldert et al., 2009]. The fNIRS recording systems are based on a helmet composed of optodes and receptors. The optodes apply light in the near infrared spectrum (600 to 1000 nm) through the skull and the cortex [Lebedev and Nicolelis, 2017] [Sirpal et al., 2019]. The unabsorbed light by the brain tissue is retrieved by the receptors. The brain near infrared light absorption is related to the brain oxygenation which allows to record the brain oxygen and thus brain activity. fNIRS recordings yield higher spatial resolution than EEG (around 1 cm) [Lebedev and Nicolelis, 2017] [Roy et al., 2018] [Sirpal et al., 2019] whereas, due to the slow dynamics of hemodynamic response, fNIRS systems have a lower temporal resolution (around 100 ms) [Lebedev and Nicolelis, 2017] [Roy et al., 2018] [Sirpal et al., 2019]. The fNIRS systems are not expensive and ease to use. fNIRS recording systems were reported in motor BCI [Khan et al., 2018] and BCI [Naseer et al., 2016a] experiments. Moreover, multiple studies mixed EEG and fNIRS recordings to bring the best of both worlds and improve BCI performance [Lebedev and Nicolelis, 2017] [Roy et al., 2018] [Sirpal et al., 2019].

**Functional Magnetic Resonance Imaging (fMRI)**. is an indirect brain activity recording system using an MRI scanner. As fNIRS, it is a measurement of blood oxygen concentration [Rak et al., 2012] [Waldert et al., 2009], specifically from blood oxygen level dependent (BOLD) activity [Lebedev and Nicolelis, 2017]. The major advantage of fMRI is its good spatial resolution for the entire brain (not only the cortex surface) allowing 3D representation of the brain activity (generally around 3 to 4 $mm^3$ voxels) [Lebedev and Nicolelis, 2017]. Nevertheless, fMRI clusters numerous drawbacks such as a low temporal resolution (around 1 to 2 seconds) and a significant delay between the brain activity and the BOLD response (around 3 to 6 seconds) [Lebedev and Nicolelis, 2017]. Moreover, fMRI instrumentation presents the same disadvantages as MEG recordings systems with very expensive and bulky platform requirements [Rak et al., 2012].

### 1.2.2. Neural signal processing

The digitized neural data are processed in a signal translation block also called transducer [Schaeffer and Aksenova, 2018]. Transducer block groups all the signal-processing steps which allow the patient to generate a command for the effector from his recorded neural signals. To do so, BCI signal processing generally consists of several steps referred to as brain signal pre-processing, neural feature extraction, decoding and

post-processing (Figure 1-7). Each of these blocks is highly dependent on numerous criteria such as the recording system, the mental strategy applied during the experiments by the patient, the goal of the BCI (motor BCI, rehabilitation BCI, communication BCI, etc.) and the effector to control (wheelchair, drone, speller, computer browsing, etc.). Nevertheless, the principle of each block can be generally defined.

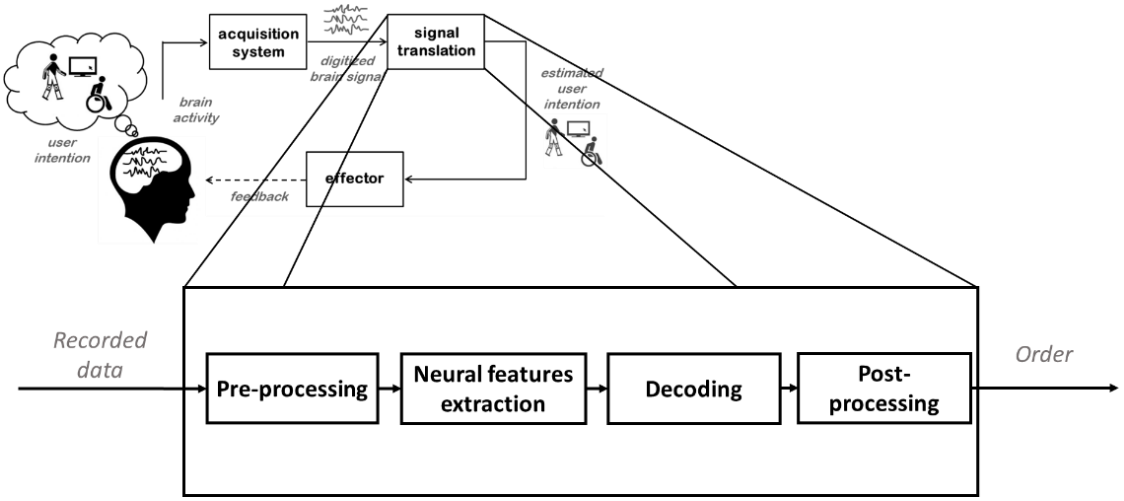



Figure 1-7: BCI signal transducer sub-steps. The transducer block transforms the neural signal recorded from the patient to coherent command for the effect. Transducer block can be split into pre-processing, neural feature extraction, decoding and post-processing steps.

**Pre-processing** block aims to enhance the signal quality and improve the signal-to-noise ratio for future steps [Anitha et al., 2019] [Bashashati et al., 2007a] [Ramadan et al., 2015]. Different operations can be performed to enhance the brain signals such as temporal and/or spatial filters or down-sampling operations as anti-aliasing filters [Hassan and Rabiul Islam, 2019] [McFarland et al., 1997] [Syan and Harnarinesingh, 2010]. Artifacts related to eye blinking, muscle activity or power line at 50 Hz can be removed/reduced through filters or decoders dedicated to the recognition of such biological patterns [Anitha et al., 2019] [Fatourechi et al., 2007]. This step is crucial in non-invasive recording systems such as EEG which is highly sensitive to artifacts and other noise sources.

**Neural feature extraction** is a crucial step which extracts from the pre-processed neural signals the features that will be used by the decoding algorithm to generate orders to the effector. Consequently, this step aims to extract the neural features that are the most discriminative to the subject desired mental tasks [Mak and Wolpaw, 2009] [Nicolas-Alonso and Gomez-Gil, 2012]. Feature extraction can be defined by a step of feature generation and an optional step of feature selection or feature space dimension reduction [Bashashati et al., 2007a]. Feature extraction is dependent on the analyzed neural signals. Spike-count strategies are specific to MEA-based BCIs to evaluate the firing rate variation of the individual neurons [Mak and Wolpaw, 2009] [Schaeffer and Aksenova, 2018]. The features extracted from the neural population recording systems are more focused on the

modulation of the neural signal amplitude, power and phase across time or time-frequency domains [Anitha et al., 2019] [Mak and Wolpaw, 2009] [Nicolas-Alonso and Gomez-Gil, 2012] [Schaeffer and Aksenova, 2018]. Time domain analysis computes interesting features to track slow dynamics responses. Time-frequency domain analysis is, for example, useful to track amplitude spectrogram modulations during movements as their related neural signals are commonly characterized by strong amplitude modulation in the low and high frequency bands [Waldert et al., 2009].

Numerous features can be extracted from several electrodes, at several frequency bands and different time segments leading generally to a high dimensional feature space [Eliseyev et al., 2017] [Nicolas-Alonso and Gomez-Gil, 2012]. High dimensional feature space may lead to numerous issues such as important computing power requirements, high computational load, irrelevant or redundant information and "curse of dimensionality" problem in the decoder training step [Bellman, 1961] [Bishop, 2006] [Nicolas-Alonso and Gomez-Gil, 2012] [Remeseiro and Bolon-Canedo, 2019]. Dimensional reduction and feature selection algorithms aim to reduce the feature space dimension to avoid the "curse of dimensionality" and improve the decoding performances. Additionally, reduction of the feature space dimension may also drastically lower the required computing time by allocating less computing resources to feature extraction step (do not compute the irrelevant features), avoid overfitting, reduce the training time of the decoder, remove correlated features (numerical stabilization), denoise the signals and lead to an easier interpretation of the results [Haufe et al., 2014]. In the case of high dimensional neural signal processing, all these aspects are relevant [Haufe et al., 2014] [Khaire and Dhanalakshmi, 2019] [Remeseiro and Bolon-Canedo, 2019] and more specifically for real-time BCI application with real-time data flow processing and decoding. Feature selection family regroups filter-based, wrapper-based and embedded techniques [Khaire and Dhanalakshmi, 2019] [Remeseiro and Bolon-Canedo, 2019]. Another approach called dimension reduction or projection aims to project the feature space into a subspace of lower dimension by a linear or non-linear combination of the initial feature space components to create few highly informative features [Haufe et al., 2014].

**Decoding** step purpose is to use a linear or non-linear model to map/translate the neural feature space to the space of possible commands/orders to send to the effector. Decoder changes independent variables (signal features) into estimates of the user movement intention dependent variables (effector control commands) [Mak and Wolpaw, 2009] [Schaeffer and Aksenova, 2018] [Wolpaw et al., 2002]. Nevertheless, the model parameters are generally patient-specific and data-driven. Therefore, various BCI decoders were tested or designed to improve the neural signal decoding. The decoding algorithms applied in the BCI field can be clustered into specific nested families.

Decoders can be regrouped depending on the type of expected user's intention. Discrete decoders also called classifiers, cluster the neural features into a limited number of defined states or classes. Classifiers can be binary decoders (classification between 2

states) or multi-class decoders (classification between $N$ states). Classification models (discrete decoders) have been created to detect discrete mental states allowing an accurate classification of patient's intentions (e.g., open or close the hand, walk or stand, etc.) [Bishop, 2006] [Lotte et al., 2018]. Continuous, regression decoders predict continuous variables to be realized by the effector (e.g. position or displacement of the hands in the space, etc.). For example, continuous decoders can predict the trajectories of a cursor or limb based on its position, velocity, acceleration or a combination of these components [Bishop, 2006] [Marathe and Taylor, 2011]. Another approach consists in mixing both continuous and discrete decoders to create so-called hybrid decoding models. These models mostly rely on switching between (multiple) continuous models. The selection of the continuous decoder is handled by the discrete output of a classifier: the more likely continuous decoder is selected [Schaeffer and Aksenova, 2016a].

The majority of BCI decoders (discrete, continuous and hybrid) are defined as static decoders. They rely on traditional statistical algorithms which assume that successive input and output variables are temporally independent. Nevertheless, in biological systems such as the neural signals (and BCI in general), this assumption is an important restriction which leads to a loss of information for the decoders. To take into consideration the natural temporal dependencies of the brain neural signals, several decoders referred to as dynamic or sequential decoders were implemented in BCI applications [Lotte et al., 2007] [Schaeffer and Aksenova, 2018].

Discrete, continuous and hybrid decoders were exploited in motor BCI applications to control various effectors [Han et al., 2020] [Lebedev and Nicolelis, 2017] [Lotte et al., 2018, 2007] [Schaeffer and Aksenova, 2018] [Volkova et al., 2019]. To this date, there is no consensus on the best decoder as the reported decoding performance were highly dependent on the patient, the recording system, the experimental paradigm, etc.

The model parameters can be automatically estimated based on artificial intelligence strategies such as machine learning and deep learning. These techniques use a finite "training" dataset representative of the relation between the neural feature space and the patient's intentions to estimate automatically the model parameters. Three major training categories have been conceptualized referred as: supervised, unsupervised and reinforcement learning depending on which information (independent and dependent variables) are provided in the "training" dataset to estimate the model parameters [Ayodele, 2010] [Bishop, 2006]. Supervised learning algorithms create a model from examples (training dataset) where each of the input/independent variables (e.g. neural features) are associated with the desired decoder output/dependent variables (e.g. labels or movements) [Ayodele, 2010] [Bishop, 2006]. Unsupervised learning strategies group the algorithms which extract a model from the input variables without any information on the corresponding desired output variables. These algorithms aim to find groups in which input data can be clustered because of their input neural features similarities [Ayodele, 2010] [Bishop, 2006]. Reinforcement learning strategies are learning the most

suitable actions to perform depending on the input variables to maximize a reward signal (representative of the output variables). The optimal output variables are unknown and must be discovered by trial and error processes [Ayodele, 2010] [Bishop, 2006] [Sutton and Barto, 2017].

Commonly, initial model parameters estimation for BCI application was performed based on supervised learning strategies. Nowadays, BCI experiments were realized in a defined environment where output variables were easily accessible. While few BCI fully unsupervised training procedures were tested during offline and online EEG-based [Hüebner et al., 2018] [Kindermans et al., 2014] and online MEA [Paraskevopoulou et al., 2014] experiments, unsupervised and reinforcement learning strategies were poorly explored for BCI applications and were preferred when output variables were not recorded.

Finally, all the presented algorithms groups are separable between the offline, online and online incremental/adaptive algorithms. The offline algorithm category gathers the algorithms limited to applications in post-treatments after the recording of the data. This limitation is generally related to the required computation time, cross-validation optimization requirements or the need to have the entire dataset (from the start to the end) to process the input data. Online algorithms denote the algorithms that can be applied during closed-loop online experiments with a fixed model trained previously offline. The training of the algorithms is heavy and cannot be realized during the experiments, nevertheless, the application of the model is sufficiently optimized to process in real-time the neural feature data-flow. Incremental/adaptive algorithms encompass the algorithms which can be applied and evaluated in real-time. Incremental adaptive decoders incrementally update their parameters, optimizing in real-time the model parameters to the user brain signals variations.

Contrary to offline and initial model parameter estimations which are generally based on supervised training strategy, adaptive algorithms were reported using both supervised and unsupervised re-updating strategies. For example, several adaptive algorithms using supervised and non-supervised adaptation during BCI experiments using EEG recordings were reviewed in [Lotte et al., 2018]. However, the majority of the online BCI algorithms were tested during offline dataset analysis. Tests during online BCI experiments should be a gold standard and deeper investigation during online BCI experiments must be achieved to evaluate the benefits of the online non-adaptive and adaptive algorithms. Online non-adaptive or online adaptive algorithms are mandatory for daily life BCI application, nevertheless, the online property of the algorithm brings new requirements and specifications to the algorithms such as computational load, complexity, [Murphy et al., 2016], etc.

**Post-processing** methods are generally used to smooth and/or reduce unlikely predictions and errors of the decoder to improve prediction performance. Post-

processing can also be used to integrate a priori knowledge on the effector possible operations, limitations or restrictions (e.g. maximum velocity limitation, physical barriers to not cross, do not allow walking and sitting state transition too fast, etc.) [Bashashati et al., 2007a] [Schaeffer and Aksenova, 2018].

### 1.2.3. Effectors

Numerous effectors have been designed and integrated into BCI systems in various domains such as the entertainment industry [LaFleur et al., 2013] [Mudgal et al., 2020] [Nicolas-Alonso and Gomez-Gil, 2012], the healthcare domain with diverse prevention and detection applications [Lee et al., 2019] [Mudgal et al., 2020] [Ramadan et al., 2015] [Roy et al., 2018], etc.

BCI committed to healthcare applications and more particularly BCI controlled by patients suffering from severe motor disabilities required to remain safe and harmless in any condition. Numerous effectors were controlled using BCI by disabled patients for various objectives and applications.

The BCIs dedicated to communication such as spellers were designed and controlled using BCI [Kim et al., 2018] [Milekovic et al., 2018] [Nagel and Spüler, 2019] [Nicolas-Alonso and Gomez-Gil, 2012] [Pels et al., 2019] [Vansteensel et al., 2016] but do not present direct risk and threat to the user even though false activations remain problematic. On the opposite, motor BCIs interact directly with the environment and the user. False activations may be problematic in out-of-lab applications.

Numerous studies using real (not virtual) effectors have been designed. In particular, several BCIs and motor BCIs created for the complete functional compensation of disabled patients were reported in [Lebedev and Nicolelis, 2017, 2006] [Mak and Wolpaw, 2009] with numerous effectors. In clinical experiments, robotic assistance of individuals unable to perform any muscular activation was performed using neuro-orthoses or prostheses for various tasks such as grasping, upper [Collinger et al., 2013] [Edelman et al., 2019] [Hochberg et al., 2012] [Morinière et al., 2015] [Wodlinger et al., 2015] or lower [He et al., 2018] [Kwak et al., 2015] [López-Larraz et al., 2016] [Zhang et al., 2018] limb movements or all the tasks together using an exoskeleton [Benabid et al., 2019] [Eliseyev et al., 2014].

A 6 DoF commercialized robotic arm JACO (from Kinova Robotics company) was used in EEG-based BCI studies for 2D and 3D movements control [Baxter et al., 2013] [Bhattacharyya et al., 2015, 2017a] [Edelman et al., 2019] [Huang et al., 2019] [Meng et al., 2016] [Postelnicu et al., 2019]. The DLR Light-Weight Robot III combined with the Five-Finger Hand was controlled in an end-point velocity space (7 DoF) by a tetraplegic patient using MEA neural signal decoder [Hochberg et al., 2012]. The same experiments were performed using the DEKA robotic arm for 6 DoF control in [Hochberg et al., 2012]. The DLR prosthetic system provides potentially a 7 DoF arm and 15 DoF hand (shoulder

abduction, shoulder flexion, humeral rotation and elbow flexion, wrist flexion, wrist rotation) and 4 DoF in the hand [Hochberg et al., 2012].

John Hopkins University designed the modular prosthetic limb, an anthropomorphic prosthesis which enables in end-point-control mode to command independently 16 degrees of freedom (3D translation and 3D orientation of the hand, as well as 1D flexion/extension of each finger, ab/adduction of the index finger, combined ab/adduction of the little and ring fingers, and 4D control of the thumb) [Collinger et al., 2013] [Johannes et al., 2011] [Wodlinger et al., 2015]. This prosthesis was used to performed 10 DoF control by a tetraplegic patient using a MEA recording system [Collinger et al., 2013] [Wodlinger et al., 2015]. Additionally, full-body exoskeleton [Benabid et al., 2019] [Eliseyev et al., 2014] have been designed and controlled via neural signals decoding. A whole-body exoskeleton with 14 actuated DoF was used for BCI experiments using epidural ECoG recording systems [Benabid et al., 2019].

Numerous studies on lower or upper limb control of an exoskeleton using EEG acquisition system were reported in [AL-Quraishi et al., 2018]. However, these EEG experiments were carried out with a lower number of DoF than experiments with more invasive recording systems and were generally performed with healthy subjects.

A commonly controlled effector providing high mobility to disable patients is the wheelchair. Wheelchairs were adapted to BCI control giving back mobility to patients using EEG recordings [Huang et al., 2012] [Leeb et al., 2007] [Li et al., 2013] [Mak and Wolpaw, 2009].

BCIs were also used for functional rehabilitation of individuals suffering from muscular activity deterioration to help patients and brain plasticity to recover [Bundy David T. et al., 2017] [Cantillo-Negrete et al., 2018] [Carvalho et al., 2019] [Donati et al., 2016] [Frolov et al., 2017] [López-Larraz et al., 2018] [Mak and Wolpaw, 2009] [Mattia et al., 2013] [Mudgal et al., 2020] [Nicolas-Alonso and Gomez-Gil, 2012] [Pfurtscheller et al., 2008] [Qin et al., 2019] [van Dokkum et al., 2015] [Webb et al., 2012]. Experiments oriented for functional rehabilitation applications were reported using real effectors dedicated to grasping or wrist rotation [Bundy David T. et al., 2017] [Cantillo-Negrete et al., 2018] [Carvalho et al., 2019] [Frolov et al., 2017] [Qin et al., 2019] [van Dokkum et al., 2015], upper [Carvalho et al., 2019] [van Dokkum et al., 2015] [Webb et al., 2012] and lower [Donati et al., 2016] limb movements, functional electrical stimulation (FES) [Carvalho et al., 2019] [Mattia et al., 2013] [van Dokkum et al., 2015] etc. The systems used for functional rehabilitation generally present low DoF (1 or 2) as the possible actions provided to the patient are restricted to specific movements.

Nevertheless, the most widespread category of applications belongs to the virtual effector family due to its set up simplicity and accessibility for clinical and preclinical studies. Many experiments based on 1D, 2D or 3D cursors [Brandman et al., 2018] [Cunningham et al., 2010] [Dangi et al., 2014] [Kao et al., 2017] [Leuthardt et al., 2004] [Marathe and Taylor, 2015] [Orsborn et al., 2014] [Schalk et al., 2008] virtual avatar, arms or

environments [Huang et al., 2012] [Ifft et al., 2013] [Lebedev and Nicolelis, 2011] [Velliste et al., 2014] were reported.

In all the cases, the effectors must be designed to provide safe use to the patient and be as transparent as possible (ease of use and a high number of possible actions). Numerous safety restrictions are generally used for effectors that are in direct contact with the patients. As an example, virtual boundaries were fixed in [Hochberg et al., 2012] during the post-processing step to avoid collisions between the robotics arms, the experimental setup and the patient.

### 1.2.4. Feedback

The sensory feedbacks are the information about the task evolution, success or failure provided to the patient which allows him to react and adapt to the current situation. Feedback is crucial in BCI applications. Feedback highly influences the model convergence, parameter estimation [Cunningham et al., 2010] [Jarosiewicz et al., 2013] and performance. As an example, several studies highlighted that higher decoding frequency improved effector control [Cunningham et al., 2010] [Shanechi et al., 2017] whereas visual feedback delay significantly affected the decoding performance [Marathe and Taylor, 2015].

In the majority of the motor BCI studies dedicated to functional compassion, the feedback provided to the patient is restricted to visual feedback. Indeed, besides being easier to integrate within the BCI experiments (using a screen as visual feedback for 2D or 3D cursors control), it is generally the only feedback that can be provided to paraplegic and tetraplegic patients who lose their other sensory feedbacks. Few studies tried to add other feedbacks to the visual feedback such as kinaesthetic (sense of body movement) for the upper limb decoding, the haptic feedback (sense of touch) for grasping control or vibrotactile feedback for cursor control [Schaeffer and Aksenova, 2018].

Multiple sensory feedback types are much more common in motor BCI experiments designed for functional rehabilitation. Most of the patients did not lose their entire sensory system and used multiple feedbacks to improve their rehabilitation. Regardless of the effector, numerous reported experiments combined visual, kinesthetic and proprioceptive (sense of body positioning) and/or haptic feedbacks [Carvalho et al., 2019] [Mattia et al., 2013] [van Dokkum et al., 2015]

## 1.3. BCI control strategies: somatotopic remapping versus direct neural decoding

Numerous BCI control strategies were experimented to obtain the most accurate neural signal predictions. BCIs and control strategies can be clustered into exogenous and

endogenous BCIs [Chan et al., 2015] [Nicolas-Alonso and Gomez-Gil, 2012] according to the nature of the neural signals used to control the effector. On the one hand, exogenous BCI relies on patient's neural signal variations related to an external stimulus named evoked potentials (EP) and more specifically event-related potentials (ERP). This category regroups visual (VEP), auditory, P300 evoked potentials, error-related potentials (ErrP), etc. On the other hand, endogenous BCIs use the natural variations of the sensorimotor neural signal rhythms to decode the patient's intentions. They do not rely on external systems. Endogenous BCIs encompass two BCI control strategies: the direct neural decoding strategy and the somatotopic mapping/mental task strategy [Chan et al., 2015] [Nicolas-Alonso and Gomez-Gil, 2012].

### 1.3.1. Exogenous BCIs

Event related potentials are specific neural patterns related to external stimuli (visual, auditory). Event related potentials relying on visual external stimuli named visual evoked potentials (VEPs) are the most common ERP used for BCI applications. VEPs are strong amplitude variations signals occurring in the visual cortex which can be elicited from repetitive visual stimuli such as flashing lights, appearance or change of an image [Nicolas-Alonso and Gomez-Gil, 2012]. Depending on the frequency of the stimuli, VEP can be categorized into transient (TVEPs) and steady state (SSVEPs) evoked potentials when the frequency rate of the stimuli is below 6 Hz or at a higher frequency rate respectively [Nicolas-Alonso and Gomez-Gil, 2012].

Another evoked potential commonly used in BCI application, especially for communication BCI, is the P300 evoked potential. P300 evoked potentials is characterized by a positive variation of the neural signals elicited around 300 ms after scarce auditory, visual, or somatosensory stimuli [Nicolas-Alonso et al., 2015] [Waldert et al., 2009]. P300 evoked potentials do not require patient's training, nevertheless the bit rate information is lower than VEP and the P300 amplitude modulation may be reduced due to patient's habituation [Nicolas-Alonso et al., 2015].

Several BCI were designed based on evoked potentials decoding. Generally, they relied on EEG recording systems. Several evoked potentials were used for various BCI applications such as 2D cursor control or spellers [Dornhege et al., 2004] [Huang et al., 2012] [Nagel and Spüler, 2019]. However, P300 and SSVEP were the most common patterns decoded in the BCI field. SSVEP was used to perform online experiments for 2D cursor control [Trejo et al., 2006], wheelchair control [Li et al., 2013] [Müller et al., 2015], grasping control of a hand orthosis [Ortner et al., 2011] or lower limb exoskeleton control [Kwak et al., 2015] from EEG neural signals. EEG neural signal decoders based on P300 evoked potentials were tested during online motor BCI experiments to control a wheelchair [Annese et al., 2017] [Iturrate et al., 2009] [Li et al., 2013]. However, they were more commonly reported in online experiments [Lin et al., 2018] [Long et al., 2011] [Utsumi et al., 2018] and offline studies [Kim et al., 2018] dedicated to BCI speller applications.

On the one hand, exogenous BCI decoders are easy to train since the evoked potentials such as steady state VEP (SSVEP) or P300 VEP are naturally encoded with specific neural patterns and do not require extensive recording systems (one EEG channel can be sufficient) or patient's training. Moreover, exogenous BCIs can have fast information transfer rate. Nevertheless, they require a constant focus of the patient which can induce high mental load and tiredness [Nicolas-Alonso and Gomez-Gil, 2012]. On the other hand, endogenous BCIs are independent of any stimulus, can control more complex effectors with high DoF and are more adapted to continuous effector control and patient with affected sensory organs [Nicolas-Alonso and Gomez-Gil, 2012].

For these reasons, the BCI experiments proposed in this Ph.D. manuscript were restricted to endogenous strategy. A special focus on this strategy is carried out in the next section. Nevertheless, exogenous BCI is an active field of research and most of the decoding algorithms applied in exogenous BCI experiments (generally based on EEG neural signal recording system) can be translated to endogenous applications.

### 1.3.2. Endogenous BCIs

Endogenous BCIs provide commands to an effector directly from brain signals variations decoding without any external stimuli. Endogenous BCIs are less restricted systems than exogenous ones which rely on few state classification task based on various external stimulation. However, endogenous BCI models are more complicated to train and are more prone to errors. Endogenous BCI can be clustered into two control strategies: direct neural decoding (also named kinematic decoding) [Waldert et al., 2009] and somatotopic remapping approach also named somatotopic mapping/mental task strategy, arbitrary-mapping paradigms [Degenhart et al., 2018] [Volkova et al., 2019] [Waldert et al., 2009].

**The Somatotopic remapping approach** [Degenhart et al., 2018], is also referred as somatotopic mapping/mental task strategy [Waldert et al., 2009] or arbitrary-mapping paradigms [Volkova et al., 2019] in the literature. As mentioned in 1.1, the brain motor cortex has a somatotopic organization: specific parts of the body are associated with distinct locations and specific neural signals patterns (the left hand and left foot movement do not activate the same motor cortex area). The somatotopic mapping approach aims to associate specific BCI commands to arbitrary selected attempted (realized or imagined) movements which have distinct neural pattern activations between each other [Degenhart et al., 2018] [Volkova et al., 2019] [Waldert et al., 2009]. As an example, Figure 1-8A, shows the somatotopic mapping strategy used by five epileptic patients to control a 2D cursor with attempted real or imagined movements in [Schalk et al., 2008]. The Figure 1-8B highlights the distinct motor cortex activations related to actual and imagined movements of the tongue and the hand. The mental imagination of a motor task without its execution (without movements) is referred to as Motor Imagery (MI) in the literature.

A)

| Subject | Task (direction) | Brodmann's area | Frequency band | Avg Acc. |
|---|---|---|---|---|
| | | Actual movements | | |
| B | Tongue movement (v) | 1 | 79–85 Hz | 69% |
| | Eye movement (h) | 20 | 17–19 Hz | |
| | | 20 | 17–19 Hz | |
| | | 20 | 17–19 Hz | |
| C | Middle finger movement (v) | 40 | 70–80 Hz | 73% |
| | | | 90–100 Hz | |
| | | | 110–120 Hz | |
| | | | 130–140 Hz | |
| | Shoulder movement (h) | 4 | 70–90 Hz | |
| | | 6 | 70–90 Hz | |
| | | 5 | 100–120 Hz | |
| D | Tongue movement (v) | 6 | 77–83 Hz | 58% |
| | | | 95–101 Hz | |
| | Hand movement (h) | 4 | 77–83 Hz | |
| | | Imagined movements | | |
| A | Imagined jaw (v) | 31 | 115–121 Hz | 53% |
| | Imagined hand (h) | 47 | 47–53 Hz | |
| | | 38 | 47–53 Hz | |
| E | Imagined tongue (v) | 43 | 90–95 Hz | 60% |
| | | | 110–115 Hz | |
| | Imagined hand (h) | 3 | 85–90 Hz | |
| | | 2 | 90–95 Hz | |
| | | 6 | 115–120 Hz | |
| | | | 125–130 Hz | |

B)

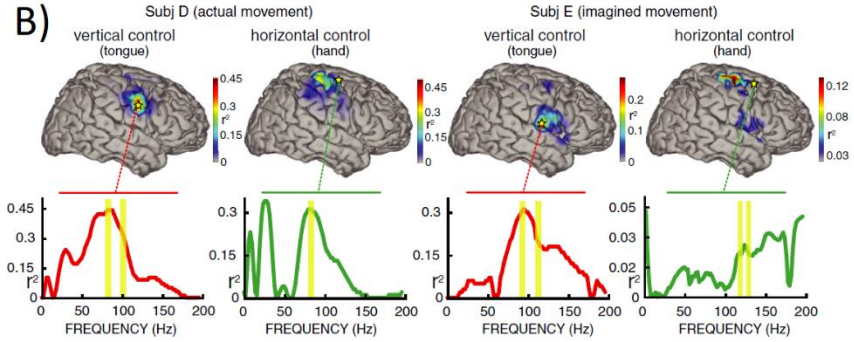



Figure 1-8: Example of motor imagery strategy carried out during clinical BCI experiments. A) mental task strategy with imagined or realized movements performed to control a 2D cursor in [Schalk et al., 2008]. B) Brain motor cortex activation related to tongue or hand real or imagined movement.

The somatotopic mapping strategies to control BCI systems are numerous and varied, tongue, jaw, hands, shoulders, elbows, fingers, legs, feet [Degenhart et al., 2018] [Schalk et al., 2008] [Volkova et al., 2019] and are not restricted to strategies based on motor cortex activations [Müller-Putz, 2020] [Waldert et al., 2009]. Scherer et al. [Scherer et al., 2015] highlighted that the use of "brain-teaser" such as mental subtraction and mental word association combined with more classical motor imagery strategies increase the classification performance compared to motor imagery strategy alone.

Mental tasks strategy was generally performed with a neural population recording system distributed in a sufficiently large area of the brain to performed distinct motor imagery (ECoG, EEG, MEG, etc.) [Müller-Putz, 2020] [Volkova et al., 2019] [Waldert et al.,

2009]. EEG studies reported accurate 2D continuous pursuit task (cursor tracking) [Edelman et al., 2019] and 3D sequential reach and grasp task (2D movements then automatized 1D grasp movements) [Meng et al., 2016] using left and right hand motor imagery for left or right movements while rest state or both hands motor imagery were associated to up and down movements. EEG based 3D virtual cursor control was reported by McFarland et al [McFarland et al., 2010] using MI ("initially employed" to refer to the article) with intensive subjects training. The four subjects underwent between 24-96min, 4 to 5 hours and 8 to 17 hours of training to control a cursor in the 1D, 2D and 3D space respectively. Virtual drones were controlled in the 3D space using hands MI in [LaFleur et al., 2013] [Royer et al., 2010].

**Somatotopic remapping strategy in ECoG-based experiments** is the most common control strategy performed. Early BCI studies using ECoG neural signal recording systems were carried out using somatotopic remapping strategy [Volkova et al., 2019] and ECoG-driven state of the art BCI continued to exploit it. Wang et al., in two online BCI experiments [Degenhart et al., 2018] [W. Wang et al., 2013], highlighted 2D and 3D cursor control by three disabled patients using motor imagery strategy and ECoG recording system. These experiments were conducted with a tetraplegic patient caused by complete C4 level spinal cord injury seven years before the study, a patient diagnosed with amyotrophic lateral sclerosis nine years prior to the study and a subject with left brachial plexus injury three years before his enrolment in the study [Degenhart et al., 2018]. These patients underwent a craniotomy to implant a high-density ECoG grid of 32 (for the two first patients) and 64 (for the third patient) electrodes embedded in a 2cm × 4cm (for the two first patients) or 4cm × 4cm grid. The proposed task was to firstly control a 2D cursor to perform a center out task with 8 targets for the first and third subjects and 4 targets for the second subject. In a next step, 3D cursor center out task was performed with 8 targets for all the subjects. Motor imagery strategies were different for each subject depending on neural signal modulation in the gamma band and spatially distinct patterns. The MI strategies performed during the experiments are represented in Figure 1-9. For each patient, the optimal motor imagery strategy was determined during prior motor screening task analysis to identify the attempted movements which generated the strongest cortical modulations [Degenhart et al., 2018].

| | 2D | | 3D | | |
|---|---|---|---|---|---|
| | M1 | M2 | M1 | M2 | M3 |
| S1 | Hand | Elbow | Hand | Elbow | Wrist |
| S2 | Thumb | Middle finger | Thumb | Little finger | Elbow |
| S3 | Ring/little finger | Elbow | Ring/little finger | Elbow | Wrist |

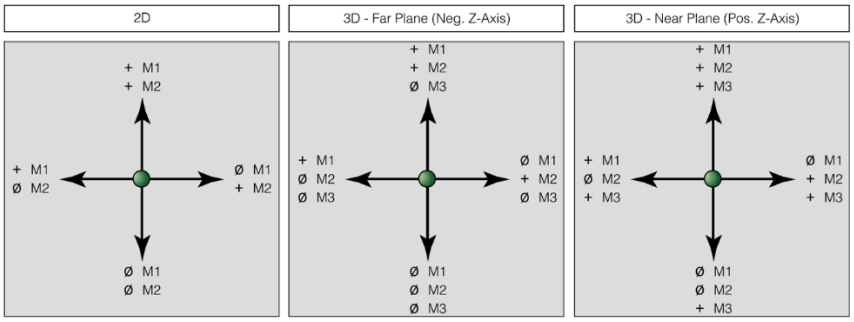



Figure 1-9: Motor imagery strategy achieved during online BCI clinical experiments using ECoG recording system. Motor imagery strategy used in [Degenhart et al., 2018] to performed 3D center out task control with 8 targets. M1 and M2 were used to control the displacement of the target in the X-Y axis whereas M3 was associated to Z-axis (depth axis). "+" represents attempted movements whereas ø" shows relaxation.

Based on somatotopic remapping strategies, patients highlighted a 85 ± 6 % and 75 ± 10 % success rates (targets hits) during online 2D center out experiments and 3D center out experiments respectively. To maintain the decoding performance, the decoder was regularly recalibrated "as needed" to improve the cursor control and reduce the sudden drop off decoding performance. Re-calibration sessions were performed before and between test trials with a total of 8, 19 and 5 re-calibration sessions for the first, second and third patients, respectively.

**Direct neural decoding** relies on the decoding of single neurons or population neural signals directly related to movement control [Waldert et al., 2009]. In the mid-1980's, Georgeopoulos et al. highlighted in preclinical experiments that the action potential firing rates of the premotor and motor cortex neurons were correlated to specific movement directions. The firing rate of a premotor and motor cortex neuron was directionally tuned by the movement direction. The activity of several neurons (named population vector) of the premotor and motor cortex appeared to provide the direction of visually guided movements. [Georgopoulos et al., 1986] [Georgopoulos and Carpenter, 2015] [Purves et al., 2004] and other characteristics such as speed, velocity [Waldert et al., 2009], etc.

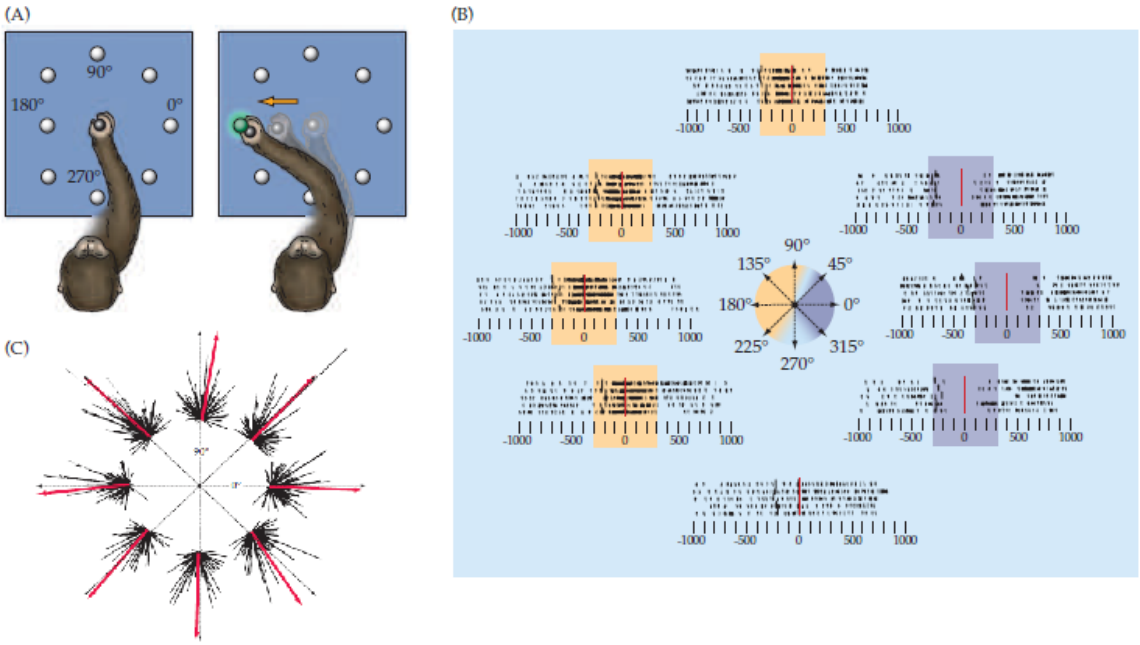



Figure 1-10: Direct neural decoding principles. The firing rate of premotor and motor cortex neurons is correlated to specific movement direction. Each neuron is tuned depending on a preferred movement direction. The figure is extracted from Neuroscience 3rd edition [Purves et al., 2004].

A milestone was reached in 2006, Hochberg et al. [Hochberg et al., 2006] demonstrated that similar firing rate directional tuning of the motor cortex neurons could be estimated through the recording of the motor cortex neurons of a tetraplegic patient imagining hand movements using an implanted 96-microelectrode array. Moreover, Hochberg et al proved that the neural signals modulations of a tetraplegic patient imagining hand movements could be exploited to control 2D cursor position and perform center-out tasks. These results highlighted that intended (and not only realized) movements were correlated to neural population firing rate recorded through invasive intracortical electrodes even for a 3 years old spinal cord injured patient.

BCI studies based on direct neural decoder were generally performed with invasive MEA recordings. As mentioned in 1.2.1, MEA systems directly record single or multi-units neuronal activities and are by definition suited to direct neural recording. High dimensional and accurate control was performed using MEA and direct neural decoding strategy. Hochberg et al. [Hochberg et al., 2012], based on Kalman filter, allowed a tetraplegic patient to perform online 3D hand translation and grasp state control of a robotic arm. Collinger et al. [Collinger et al., 2013] highlighted the online 7 dimensional control of a robotic arm using indirect optimal linear estimation (OLE) with ridge regression whereas Wodlinger et al. [Wodlinger et al., 2015] performed a 10 Dimensional control of a robotic arm by tetraplegic patient using indirect OLE. While it is less

common, kinematic decoding of lower limb bipedal walking was also highlighted in rhesus macaque using direct neural decoding strategy [Fitzsimmons et al., 2009] [Lebedev and Nicolelis, 2017].

Although most studies were performed based on SUA and MUA decoding using MEA systems, correlation between neural signals amplitude variation and center-out movements from population recording systems was reported [Waldert et al., 2009]. During center out movements, similar neural signal modulation from LFP, ECoG, EEG and MEG recordings were reviewed in [Waldert et al., 2009]. Neural signals modulation recorded from population recording system are characterized by an increase of slow cortical signals (LFP: ≤13 Hz, ECoG: ≤2 Hz, EEG/MEG≤7 Hz) during movement, amplitude reduction in the LFP: 16–42 Hz, ECoG: 6–30 Hz, EEG/MEG: 10–30 Hz frequency bands and an amplitude rise in the high frequency bands (LFP: 63–200 Hz, ECoG: 34–128 Hz, EEG/MEG: 62–87 Hz). During offline MEG and EEG neural signal analysis of nine healthy subjects, Walder et al [Waldert et al., 2008] stressed significant directional tuning of the neural signals and real movements of the hand controlling a cursor in the 2D space with a joystick (four targets) (67% accuracy with MEG recordings). Moreover, Schwarz et al. [Schwarz et al., 2020] highlighted online direct neural classification of three grasping types (palmar and lateral grasps, and wrist supinations) with 48% corrects trials performed by fifteen healthy subjects with EEG recording system.

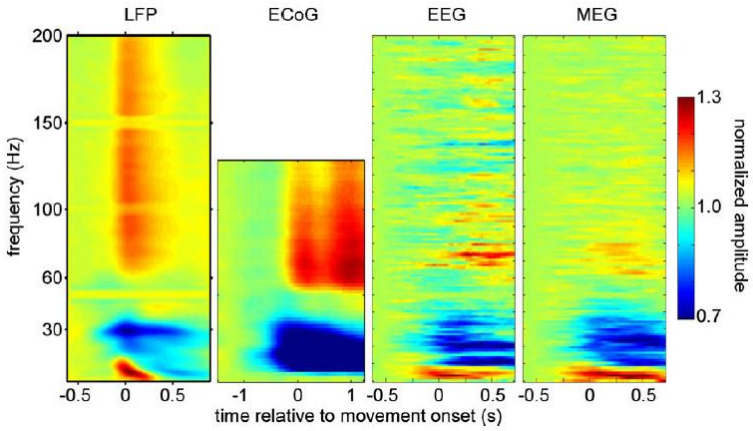



Figure 1-11: average spectrogram of center out task based on LFP, ECoG, EEG and MEG recordings during direct neural decoding experiment. The figure is extracted from [Waldert et al., 2009].

**Direct neural decoding based on ECoG recording system** was reported with performance highly dependent on the experimental paradigms, patients' status (healthy or disabled), analysis and electrodes localizations [Volkova et al., 2019]. Schalk et al. [Schalk et al., 2007] reported in 2007 an offline study where five epileptic patients

implanted with subdural ECoG performed, using a joystick, a 2D cursor tracking task of a target moving in a counterclockwise circular trajectory. The average correlation between neural signals and velocity was $0.48 \pm 0.09\%$. Ball et al. [Ball et al., 2009] highlighted the cosine tuning of four epileptic patient's between the subdural ECoG signals and their arm movements during 2D center-out tasks with four to eight targets (squared correlation coefficient of 0.67 for 34Hz-128Hz band). Anderson et al. [Anderson et al., 2012] tested the correlation between subdural ECoG signals direction, velocity and speed from seven epileptic patients. Patients performed 2D center-out and tracing tasks with a force feedback joystick. The results underlined a higher modulation of direction, velocity and speed in the motor cortical areas (depth of modulation around 0.17, 0.38, 0.23 for the direction, velocity and speed respectively) [Anderson et al., 2012]. Finally, 3D center-out task movement reconstruction was achieved by Bundy et al [Bundy et al., 2016] from five epileptic patients implanted with subdural ECoG performing 3D real hands movements. Correlation coefficients between recorded and predicted position, velocity and speed were $0.3656 \pm 0.1384$ for the position, $0.3461 \pm 0.1119$ for the velocity and $0.6208 \pm 0.1893$ for the speed.

Other ECoG-based motor BCI experiments using direct neural decoding strategy were designed to decode finger movements [Volkova et al., 2019]. Based on five epileptic patients implanted with subdural ECoG who were instructed to move specific individual fingers in response to visual cues, finger movements were extracted from neural signals using various algorithms [Elgharabawy and Wahed, 2016] [Flamary and Rakotomamonjy, 2012] [Kubánek et al., 2009] [Schaeffer, 2017] [Xie et al., 2018]. Flint et al [Flint et al., 2016] reported continuous grasp and finger joint movements decoding from 5 epileptic patients with epidural and/or subdural ECoG recording systems. They reported similar results than previous experiments, highlighting that low-frequency modulation (7-20 Hz) encodes movement onset (as grasp aperture) whereas high frequency variations (above 70 Hz) are correlated with finer movements (fingers, grip force, etc.). A study based on two epileptic patients implanted with subdural ECoG over the interhemispheric M1 area was performed to evaluate the direct neural decoding performance of ECoG recording system for lower limb control [McCrimmon et al., 2018]. The results of this study suggested that high frequency $\gamma$ band (40-200 Hz) provides information on the lower limb high-level motor control (walking duration and speed) and do not encode muscle activations or muscle trajectories. Nevertheless, lower limb kinematic decoding is a poorly developed BCI field and requires more studies to have a clear representation of the information which can be extracted from ECoG recordings.

Although motor imagery approaches lead to interesting results, this strategy may be limited in many aspects. Firstly, 10 to 30% of users are unable to control MI-BCIs [Jeunet et al., 2016]. Moreover, the control of complex effectors with high dimensional control may be difficult to handle for patients as motor imagery complexity highly increase with task complexity (highlighted by the density of articles using MI to control 2D cursor

compared to 3D ones). More complex effector control required more MI strategies, which are by definition limited (two hands, elbows, shoulders, etc.). Finally, for rehabilitation applications, patients must perform natural movements in the hope of improving the affected limb movements. However, the MI-based BCIs which are not using the natural somatotopic mapping of the brain might be useless compare to direct neural BCI for the specific case of rehabilitation applications.

## 1.4. Motor BCI requirements for daily life applications

The Motor Brain computer interfaces estimate a command from the neural activity and send it to an effector which performs the movement imagined by the patient [Lebedev and Nicolelis, 2017]. Motor BCIs are particularly useful for disabled patients who lost entirely or partially the natural neuromuscular activation path. Motor BCIs can be an interesting approach for both robotic assistance and neurorehabilitation therapy of individuals suffering from severe motor disabilities [Donati et al., 2016] [Lebedev and Nicolelis, 2017] [López-Larraz et al., 2018]. Spectacular BCI milestones have been reached over the years in the motor BCI field [Collinger et al., 2013] [Degenhart et al., 2018] [Hochberg et al., 2012] [W. Wang et al., 2013] [Wodlinger et al., 2015]. These milestones have sustained the aim of translating BCI-driven systems from laboratories directly into patients' home for daily life applications. In order to develop Motor BCI for future daily life applications, many challenging aspects and restrictions need to be addressed.

### 1.4.1. Acquisition system requirements

The primary challenge of motor BCIs for clinical and daily-life applications is safe, chronic and stable neural recordings over time. Biocompatibility as well as stability over time are mandatory for recording devices designed for long-term BCI clinical use. Brain signal recordings need to remain accurate in conditions less favorable than laboratories.

MEA recordings, post-surgery, are a safe recording system, nevertheless, they have biocompatibility issues with signal degradation over time (decreasing signal-to-noise ratio), loss of electrodes [Gunasekera et al., 2015] [Jorfi et al., 2015] [Marin and Fernandez, 2010] [Moran, 2010] [Murphy et al., 2016] [Rousche and Normann, 1999] [Volkova et al., 2019] [Ward et al., 2009] and have high across-day variation in the neural signals [Perge et al., 2013] [Sussillo et al., 2016]. However, several studies highlighted significant results with MEA implanted since hundreds days [Milekovic et al., 2018] [Simeral et al., 2011] [Wodlinger et al., 2015]. Wodlinger et al [Wodlinger et al., 2015] performed 7 dimensional and 10 dimensional robotic arm control by a tetraplegic patient from 32 to 280 days post-surgery. Simeral et al [Simeral et al., 2011] demonstrated 2D point and click control using SUA recordings by a tetraplegic patient 1000 days after the surgery despite only 57 electrodes over 96 electrodes were obtained (41 electrodes utilized). Milekovic et al [Milekovic et al., 2018] based on LFP recordings enable an ALS patient to control a BCI for

communication 550 days after implantation. However, a chronic preclinical study evaluating the MEA recording robustness over time highlighted that 56% of the recording systems failed within a year of implantation [Barrese et al., 2013].

MEA recording is restricted to wired connection to a computer due to the high sampling rate of the system. Such wired platform is limiting for daily life application and enhanced the infection risks.

Electrocorticography (ECoG) provides a fair compromise between invasiveness, signal resolution and quality [Leuthardt et al., 2006, 2004] [Schalk and Leuthardt, 2011]. ECoG recordings have fewer biocompatibility troubles than MEA. However, as previously mentioned, ECoG arrays are generally implanted to detect the epileptic sources of patients before surgery limiting the ECoG clinical trial from several days to 1 or 2 weeks (less than 28 days) of research (with an implantation from 3 to 35 days) [Bundy et al., 2016] [Degenhart et al., 2018] [Leuthardt et al., 2004] [Nakanishi et al., 2013, 2017] [Schalk et al., 2007, 2008] [Schalk and Leuthardt, 2011] [Volkova et al., 2019] [W. Wang et al., 2013] [Yanagisawa et al., 2012]. Bundy et al [Bundy et al., 2016] reconstructed 3D hand movements performed by five patients with intractable epilepsy who underwent subdural ECoG arrays implantation for 5 to 14 days to localize their epileptic foci and map cortices for pre-surgical planning. Schalk et al's study [Schalk et al., 2008] on 2D cursor control is based on five epileptic patients who had subdural ECoG arrays implanted for 7–14 days in preparation for surgery. Nevertheless, some pre-clinical [Costecalde et al., 2017] [Sauter-Starace et al., 2019] and clinical [Benabid et al., 2019] [Nurse et al., 2018] [Pels et al., 2019] [Vansteensel et al., 2016] studies showed good signal-to-noise ratio stability in ECoG signals over months or years, encouraging the use of BCIs in long-term applications. Nurse et al [Nurse et al., 2018] highlighted that subdural ECoG arrays (two ECoG grids for a total of 16 electrodes) can robustly record high frequency neural signal activities on 15 epileptic patients who underwent ECoG monitoring for 184 to 766 days. Amyotrophic Lateral Sclerosis (ALS) patients performed stable control of a subdural ECoG based-BCI system for communication over 36 months in [Pels et al., 2019] [Vansteensel et al., 2016]. Benabid et al [Benabid et al., 2019] reported epidural ECoG stability over 24 months after implantation for the BCI control of an exoskeleton by a tetraplegic patient. ECoG recording is a reliable solution for chronic BCI system. However, as for MEA-based BCI the infection risks associated to the use of tethered cables is significant [Volkova et al., 2019] but the breakthrough has been made toward the development and the test of wireless fully-implantable technologies [Benabid et al., 2019] [Pels et al., 2019] [Vansteensel et al., 2016] based on ECoG recording systems.

EEG recording systems and non-invasive wearable BCI, in general, are safe and do not present any health risk for the patient. However, they present natural restrictions and limitations for chronic stable recordings. As already mentioned, high signals variability, vulnerability to numerous artifacts (electromyographic, electro-ocular activities), small accuracy on the electrode positioning between sessions hinder non-invasive wearable

BCI for chronic BCI use. Obviously, MEG and fMRI are not considered as candidates for BCI daily life application.

All the usual recording systems used in clinical BCI studies presented benefits and drawbacks. MEA allows individual neuron recordings and highlighted impressive results in online clinical BCI experiments. However, the daily recalibration, invasiveness and wired limited recording systems are massive disadvantages which restrain the use of MEA for daily life applications. At the other end of the spectrum, EEG recording system is easy to use and is highly exploited in the BCI field for clinical research purposes. Nevertheless, poor accuracy compared to other more invasive systems, high signals variability and vulnerability to numerous artifacts are important limitations to consider. Finally, ECoG neural signal decoding highlighted better performance than EEG-based neural signal decoding in exchange for higher invasiveness, whereas lower performance was reported compared to BCI relying on more invasive recording systems such as MEA. Several chronic experiments were reported during clinical ECoG experiments and wireless recording systems were designed. ECoG electrodes can be implanted in a subdural or epidural manner. Epidural and subdural ECoG comparison was reported in [Flint et al., 2016] [Shimoda et al., 2012]. While decoding performance of epidural ECoG are lower than subdural one, epidural ECoG still presents good decoding performance and is one of the safest invasive recording methods. Additionally, epidural ECoG decoding of unimodal and bimanual upper limb movements was reported during offline preclinical experiments [Choi et al., 2018]. Therefore, epidural ECoG recording systems seem to be a good trade-off between invasiveness, safety, neural signal stability, wireless recordings and decoding performance.

## 1.4.2. control system requirements

### 1.4.2.1. Degree of freedom and accuracy

BCI system for daily life application requires providing sufficient freedom to the patients in order to enhance their independence and simplify daily life tasks. The control provided to the patients must be sufficient to reflect the user's intentions and proposes sufficient controllable degrees of freedom (DoF) or dimension to not be restricted to specific actions.

Despite that EEG-based 2D and 3D control experiments were reported in [LaFleur et al., 2013] [Lotte et al., 2018] [McFarland et al., 2010] [Royer et al., 2010] [Schalk and Leuthardt, 2011] [Schwarz et al., 2020] [Vilela and Hochberg, 2020] [Wolpaw and McFarland, 2004], they generally required specific motor imagery strategies [Lotte et al., 2018] [Schalk and Leuthardt, 2011] and high subjects training. As an example between 8 and 17 training hours were required to control a 3D cursor in [McFarland et al., 2010]. So far, the complexity of control carried out using EEG remains largely inferior to the needs of medical motor BCIs and less efficient than other (more invasive) brain neuronal activity acquisition systems [Volkova et al., 2019].

MEA-driven BCIs demonstrated better performances compared to less invasive clinical BCI systems. Using two 96-channels MEAs implanted in the left motor cortex Wodlinger and colleagues demonstrated that a tetraplegic patient was able to control 10 DoFs of a robotic arm (including 3D translation movements, 3D rotations and four hands shaping) [Wodlinger et al., 2015].

Numerous pre-clinical and clinical studies have demonstrated the interest in ECoG-based BCIs to control effectors [Benabid et al., 2019] [Bundy et al., 2016] [Chao et al., 2010] [Choi et al., 2018] [Degenhart et al., 2018] [Eliseyev et al., 2017] [Nakanishi et al., 2013] [Schaeffer and Aksenova, 2016a] [Schalk et al., 2008] [Shimoda et al., 2012] [W. Wang et al., 2013] [Yanagisawa et al., 2012]. Wang et al reported 3D robotic arm and cursor control by tetraplegic (SCI) and upper limb paralysis (ALS) subjects [Degenhart et al., 2018] [W. Wang et al., 2013] whereas Bundy et al showed offline 3D hand movement prediction [Bundy et al., 2016] based on ECoG recording strategy of epileptic subjects.

Improving the decoding accuracy and enhancing the possible interactions between the patient and various environments are the major goals of part of the BCI research community. Nevertheless, the number of controllable DoF and the decoding accuracy are not the only BCI requirements for daily life applications as some long term BCI experiments stressed the improvement in the quality of life of disabled patients using communication BCI system with only 1 DoF available [Jarosiewicz et al., 2015] [Milekovic et al., 2018] [Pels et al., 2019] [Vansteensel et al., 2016]. In [Pels et al., 2019], an amyotrophic lateral sclerosis patient used a 1DoF communication BCI system controlled through chronic (over 36 months) subdural ECoG decoding. The article stressed that the patient reported high satisfaction with the BCI system with the exception of the wired recording system which was qualified as “unsatisfied”.

#### 1.4.2.2. Multi-limb control

Daily life actions commonly required bimanual or multi-limb movements. Multi-limb decoding is a poorly explored area of the BCI field compare to single-limb movement decoding. Alternative decoding of multiple-limb could improve greatly motor-impaired patients with simultaneous or alternative multi-limb movements.

The majority of BCI studies were focused on the control of a single limb or a single effector (generally one hand or lower limb effector) [Lebedev and Nicolelis, 2011]. Bimanual movements engage multiple brain areas which are different from unimanual movements creating new neural signals modulations to decode [Donchin et al., 1998] [Lebedev and Nicolelis, 2011] [Oliveira et al., 2001] [Steinberg et al., 2002]. Only a few bimanual experiments were reported. Monkeys bimanual movements (2D for each arm) were decoded from MEA neural signals using an Unscented Kalman filter decoder to control a virtual avatar during online experiments [Ifft et al., 2013]. Offline preclinical ECoG-based movement detection studies using hierarchical partial least squares algorithm were reported in [Choi et al., 2018]. Multi-limb BCI systems were mainly

restricted to ECoG-based offline individual finger movement reconstruction studies. Hybrid models were often employed for these multi-limb/ multi-finger experiments using a classifier to detect the activated finger and continuous decoders (or multiple classifiers) to predict their respective movements [Elgharabawy and Wahed, 2016] [Flamary and Rakotomamonjy, 2012] [Hotson et al., 2016a] [Schaeffer, 2017].

The limited number of studies reporting bimanual or multi-limb experiments may be related to several technical limitations. Firstly, this shortage may be explained by a lack of experiments with bilateral implantation of invasive recording devices. During intended movements, the motor cortex activity modulation is stronger on the contralateral side. Therefore, in the case of multi-limb BCI applications with direct neural decoding, bilateral implantation is a mandatory criterion. Poor resolution of non-invasive recording systems likely impedes the related study in EEG-driven BCIs. Secondly, multi-limb effectors must be available which may be complicated to design. Finally, bimanual or multi-limb decoding requires more complex algorithms and a longer calibration procedure. Deeper investigations on such experiments should be carried out.

#### 1.4.2.3. Asynchronous BCI with idle state support

In the scope of daily life application, BCI must be a stand-alone system which can be freely used at any time by the patient without external help or cue. This feature implies discriminating the intentional movements and the idle period from the patient's neural signals. Moreover, false activation of the BCI system must be rare events.

BCI can be clustered into the cue-based triggered (synchronous) and the self-paced (asynchronous) systems (Figure 1-12). Synchronous BCIs analyze the neural signals in predefined time windows. After visual or auditory stimuli (cue) generated by the BCI or a researcher, the patient performs the mental task during a time interval which produces a command (any neural signal produced outside the time windows are ignored) [Müller-Putz et al., 2006] [Nicolas-Alonso and Gomez-Gil, 2012]. On the other hand, Asynchronous control systems continuously analyze the ongoing brain activity without any temporal restriction [Bashashati et al., 2007a] [Müller-Putz et al., 2006] [Nicolas-Alonso and Gomez-Gil, 2012] [Williams et al., 2013].

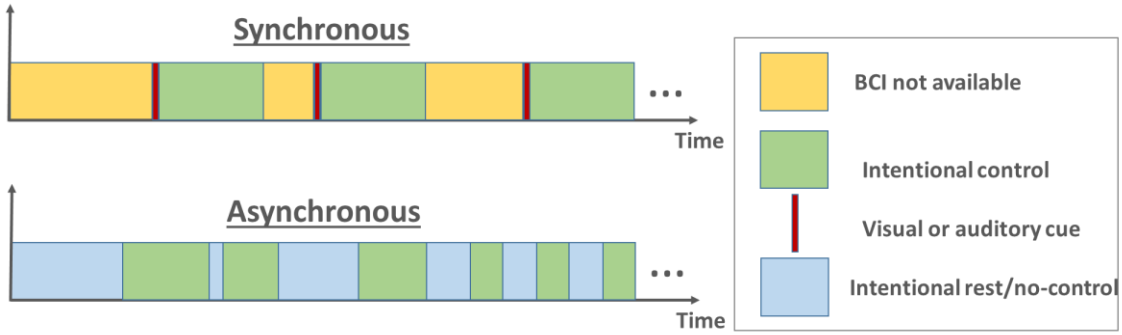



Figure 1-12: Synchonous and Asynchronous BCI system principle.

Synchronous BCIs are easier to design, train and evaluate. They only require patient's concentration in specific time windows in which the patient must focus and not create any artifact (eye blinking, muscle activation, etc.). Moreover, a decoder is only calibrated to decode a specific known mental state associated with a specific visual or auditory cue. Nevertheless, such systems are limited and do not rely on a natural control paradigm [Müller-Putz et al., 2006] [Nicolas-Alonso and Gomez-Gil, 2012]. On the opposite, Asynchronous BCI systems act as stand-alone devices which switch between intentional control and no-control phases determined by the patient's neural signals (and not an external cue). Asynchronous BCIs are able to perform a reliable rest state decoding during intentional no-control command. Consequently, asynchronous BCI provides a more natural control paradigm which does not require any external cue/stimuli. However, asynchronous BCIs are much more complicated to design, train and evaluate [Müller-Putz et al., 2006] [Nicolas-Alonso and Gomez-Gil, 2012]. They generally highlighted lower performance than synchronous BCI (low true-positive rate and high false-positive rate) [Han et al., 2020].

The majority of the state-of-art BCI decoders are synchronous [Collinger et al., 2013] [Wodlinger et al., 2015]. They do not decode an idle state. During common center-out experiments, it is assumed that subject is intentionally controlling the device at all times, nevertheless, in practice, between each trial, the cursor is replaced to the center of the screen. It is likely that during not intended control of the user, such BCI may lead to unwanted activations [Williams et al., 2013] and is therefore not adapted to "real-life" BCI applications. During daily life application, it is mandatory that false activations remain exceptionally rare events using real effector (exoskeleton, wheelchair, etc.) due to its direct contact with the patient, and the mechanical/technical limitations of the effector. Asynchronous control for BCI application is mandatory for more realistic experiments than center our tasks.

The majority of asynchronous BCI studies have been performed based on non-invasive recording systems (generally EEG) [Chae et al., 2012] [Han et al., 2020] [Kalunga et al., 2016] [Li et al., 2013] [Mason and Birch, 2000] [Müller-Putz et al., 2006] [Nagel and Spüler, 2019] [Ortner et al., 2011] [Saa and Çetin, 2013] [Yousefi et al., 2019] using various decoders and

control strategies. Several ECoG-based brain switch decoders were tested to perform idle state (IS) classification during offline pre-clinical [Chao et al., 2010] [Choi et al., 2018] [Schaeffer and Aksenova, 2016b] and clinical [Bundy et al., 2016] [Elgharabawy and Wahed, 2016] [Flamary and Rakotomamonjy, 2012] [P. T. Wang et al., 2013] studies. Finally, MEA-based single/multi-units asynchronous BCI for pre-clinical online [Suway et al., 2013] experiments and offline decoding [Achtman et al., 2007] [Ludwig et al., 2011] [Velliste et al., 2014] studies were reported.

Diverse classifiers such as Hidden Markov Model (HMM), logistic regression, linear Bayesian classifier, support-vector machine (SVM), linear discriminant analysis (LDA) have been coupled with continuous decoders such as partial least squares (PLS), Kalman Filters (KF), population vector algorithm, Laplace Gaussian filter etc. to decode offline and online asynchronous EEG-, ECoG-, MEA-driven BCIs [Bundy et al., 2016] [Chao et al., 2010] [Choi et al., 2018] [Elgharabawy and Wahed, 2016] [Flamary and Rakotomamonjy, 2012] [Leeb et al., 2007] [Mason and Birch, 2000] [Müller-Putz et al., 2006] [Nagel and Spüler, 2019] [Saa et al., 2016] [Saa and Çetin, 2013] [Schaeffer and Aksenova, 2016b] [Suway et al., 2013] [Velliste et al., 2014] [P. T. Wang et al., 2013] [Wu et al., 2004] [Yousefi et al., 2019].

Asynchronous BCI is an important field of research and numerous articles already reported relevant results for idle state detection. However, with the exception of few online MEA-based preclinical experiments e.g. [Suway et al., 2013] [Wu et al., 2004], most of the asynchronous BCI experiments relying on invasive neural signal recordings were achieved during offline neural signal analysis [Bundy et al., 2016] [Choi et al., 2018] [Elgharabawy and Wahed, 2016] [Flamary and Rakotomamonjy, 2012] [Schaeffer and Aksenova, 2016a] [Velliste et al., 2014] [P. T. Wang et al., 2013]. In order to design daily life BCI applications, further investigation on the decoding performance of asynchronous algorithms during online experiments based on invasive recording systems must be achieved.

## 1.4.3. Decoder requirements

BCI systems dedicated to daily-life applications have several constraints to be respected in order to help patients suffering from severe motor disabilities. Particularly, several criteria specifically related to the neural signal decoder must be met.

### 1.4.3.1. Real-time neural signal decoding

Obviously, to use a BCI system in daily life application, the BCI system must be sufficiently efficient to perform real-time/online/closed-loop processing of the incoming neural signals. This requirement brings new theoretical and technical demands.

From a technical point of view, processing of high dimensional data flow in real-time with minimal latency [Marathe and Taylor, 2015], and fast control rate (~10-20Hz) [Shanechi et al., 2017] are mandatory requirements of motor BCI to control robotic devices. However, the processing of high dimensional feature space and/or tensor data

structure may lead to high computational burden and time-consuming neural signal decoding which are incompatible with real-time processing. Therefore, decoders for online applications are generally restricted to linear optimized and efficient algorithms [Murphy et al., 2016].

State-of-the art decoder for online clinical BCI application generally relies on indirect optimal linear estimation (OLE) algorithm or decoder from the Kalman filter family. OLE was reported in MEA [Collinger et al., 2013] [Wodlinger et al., 2015] [Young et al., 2019] and ECoG [Degenhart et al., 2018] based experiments providing to disabled patients up to 10 DoF control based on MEA recording systems [Wodlinger et al., 2015] and 3D control with ECoG neural signals decoding [Degenhart et al., 2018]. Kalman filter algorithm allowed a tetraplegic patient to control a robotic arm to perform 3D reach and grasp movements through MEA neural signal decoding in [Hochberg et al., 2012].

Additionally, other well-known decoders were used in online BCI applications with a lower number of DoF to decode. Long-term 1D MEA-based control (up to 138 days), reported by Milekovic et al [Milekovic et al., 2018], was performed using a regularized LDA decoder. Nick F. Ramsey's team reported long-term communication BCI decoder based on smoothing and threshold optimization from ECoG neural signals [Pels et al., 2019] [Vansteensel et al., 2016]. Hotson et al highlighted the online control of the five individual fingers of a prosthetic hand by an epileptic subject performing real finger movements using a subdural ECoG recording system and a hierarchical LDA decoder [Hotson et al., 2016a].

The control of complex effectors through neural signal decoding generally requires algorithms with a high computational burden. This requirement is in contradiction with the needs of real-time data flow processing with minimal latency. Therefore, a trade-off between complexity and accuracy must be found for real-time neural signal decoding.

#### 1.4.3.2. Robust and stable decoding over time

To perform efficient, useful, and convenient use of BCI in real-life applications, decoding models must remain efficient over time and not require daily recalibration to provide accurate transcription of the patient's wills. Robust and stable BCI decoding over a long period without any model recalibration is one of the major challenges of the current BCI field.

Brain neural signals are a highly variable non-stationary environment where firing potential patterns of the neurons continuously change over hours, days and months. The non-stationarity of decoding patterns is related to inter (subject to subject) and/or intra (session to session or trials to trials) variability [Clerc et al., 2016b]. Intra subject variability is provoked by the brain plasticity and patient's factor (inattention, habituation, mental workload, etc.) [Clerc et al., 2016b]. In addition to the brain neural signals natural variability, it is mandatory that decoders remain stable in noisy environments less restricted and constrained than the laboratory settings (e.g. at home, in the street, etc.).

MEA-based BCI are sensitive to any brain neural signal variations as it records direct neuron activities and consequently require frequent (order of the day) recalibration and skilled engineer supports [Milekovic et al., 2018] [Simeral et al., 2011] [Sussillo et al., 2016] [Wodlinger et al., 2015]. Milekovic et al provided to two patients suffering from locked-in syndrome and ALS a long-term robust and durable communication BCI based on LFP recordings using brain switch decoders (1 DoF) for a period of 76 and 138 days, respectively, without recalibration and without significant loss of performance [Milekovic et al., 2018]. Schwemmer et al reported offline stable accurate four-movement classification (index flexion and extension and wrist flexion and extension) by a tetraplegic patient using 96-channel MEA in the primary motor cortex for 375 days after the end of the training period [Schwemmer et al., 2018]. Communication BCI based on ECoG recordings was used for 36 months by an ALS patient using decoder without recalibration during hundreds of days and without significant loss of performance [Pels et al., 2019] [Vansteensel et al., 2016].

However, the presented BCI systems were designed to control a low number of dimensions. BCIs dedicated to more complicated tasks require generally constant recalibration. For example, the OLE decoder used to perform 10 DoF effector control from MEA neural signals was recalibrated every day [Wodlinger et al., 2015] whereas the OLE model dedicated to the 3 DoF effector control from ECoG neural signals was frequently recalibrated after several days of experiments [Degenhart et al., 2018] [W. Wang et al., 2013]. Nowadays, long-term robust decoding of complex effector remains a major challenge of daily life application.

#### 1.4.3.3. Online closed-loop adaptive model calibration

By definition, daily-life BCI applications are closed-loop experiments. The subject's neural signals are decoded to command an effector which interacts with the environment. This interaction provides sensory feedbacks to the subject influencing the generated neural signals and future predictions, etc. In order to perform accurate and stable decoding across time, it may be preferable to take into account the neural signals patterns induced by the sensory feedback. This "human-in-loop" (closed-loop) strategy is opposed to the open-loop experiments usually performed in BCI experiments.

Open-loop sessions are experiments where the patient is passive and try to perform a mental task without feedback on his neural signals and the success or failure of the task to be carried out. However, a drop in the decoding performance was repeatedly reported applying decoders adjusted offline using open-loop (without feedback) experiments' training dataset during closed-loop experiments [Lebedev and Nicolelis, 2017, 2006] [Murphy et al., 2016] [Orsborn et al., 2014].

Experiments taking into account patient's sensory feedback (closed-loop) during the model identification highlighted drastic different parameter choices compared to protocols with passive subjects (open loop) during the calibration phase [Cunningham et

al., 2010] [Jarosiewicz et al., 2013]. The changes may be explained by the modification of the neural activity patterns between open-loop motor imagery tasks and closed-loop experiments which include new neural signals related to motor imagery and effector control feedbacks [Clerc et al., 2016b] [Schlögl et al., 2010]. Many researches underlined that closed-loop decoder identification can lead to performance rises over time [Lebedev and Nicolelis, 2017] [Murphy et al., 2016] [Orsborn et al., 2014] [Jarosiewicz et al., 2013].

Strategies to integrate the neural signals related to patient's feedback into the model calibration are based on closed-loop decoder adaptation (CLDA) procedure. CLDA updates the model parameters using closed-loop experiment dataset. Depending on the decoder and the training strategy, CLDA can be carried out in an offline or online manner at different time scales with a model adaptation occurring every sample, second, minute, trial, session, day, etc.

The classic strategy to integrate the neural signals related to the patient's feedback into the decoder calibration is divided into three steps. Firstly, the model parameters are estimated based on open-loop experiment dataset collected with the patient. Then, after the model calibration based on the open-loop data, the model is used during closed-loop experiments. Finally, the model is recalibrated based on the new closed-loop experiments [Brandman et al., 2018]. The last step can be repeated during several closed-loop experiments to improve the decoding performance of the model. This procedure is long and may lead to suboptimal calibration whereas calibration during ongoing use (in closed-loop) optimizes the quality of control during extended use of the patient.

Online incremental adaptive decoders are a group of the CLDA algorithms which update their parameters in an incremental manner with new incoming data, optimizing the model parameters in real-time and providing adaptation of the decoder to the patient and vice versa. Several studies stressed the benefits of algorithms integrating online adaptive calibration such as easier and faster use than offline computed models, more convenient for disabled patients who may struggle to remain alert and engaged during long sessions of calibration, etc. [Brandman et al., 2018] [Cunningham et al., 2010] [Jarosiewicz et al., 2013] [Orsborn et al., 2014] [Sussillo et al., 2016].

Adaptive real-time decoder identification is an important request of real life BCI application for easy use, faster integration of the feedback related neural signals during the model calibration and decoding performance improvements. In order to integrate decoding model identification into the closed-loop BCI session, several adaptive decoder identification procedures have been designed.

Several conventional classifiers (linear and non-linear) LDA, QDA, SVM, fuzzy inference system were adapted to closed loop decoder requirements and tested during closed loop BCI experiments [Bamdadian et al., 2013] [Hazrati and Erfanian, 2010] [Nicolas-Alonso et al., 2015] [Rong et al., 2018a] [Schlögl et al., 2010] [Spüler et al., 2012b] [Vidaurre et al., 2007, 2011] [Wen and Huang, 2017]. Early studies on adaptive continuous algorithms were based on

the adaptation of mu and beta rhythms amplitude linear combination during EEG recorded experiments [Wolpaw and McFarland, 2004]. Nowadays, conventional adaptive continuous decoders are often belonging to KF family [Shanechi et al., 2017] such as Adaptive KF, ReFIT KF or gaussian process discriminative KF (GP-DKF) [Brandman et al., 2018] [Dangi et al., 2014, 2011] [Gilja et al., 2012] [Jarosiewicz et al., 2015] [Li et al., 2011] [Orsborn et al., 2012], and are majorly applied in MEA based experiments. Other less conventional (MEA based BCI) strategies using Wiener filter with feedback error learning [Suminski et al., 2013], recurrent neural network [Sussillo et al., 2016], or adaptive Point Process Filters (PPF) [Shanechi et al., 2017, 2016] were reported.

Adaptive models with "Human-in-loop" update strategy seems to be a promising approach for accurate and robust BCI applications without daily decoder recalibration. While several adaptive linear and nonlinear regression and classification decoders have been designed for MEA [Brandman et al., 2018] [Dangi et al., 2014] [Gilja et al., 2012] [Li et al., 2011] [Shanechi et al., 2017] [Sussillo et al., 2016] and electroencephalography (EEG) [Hazrati and Erfanian, 2010] [Lotte et al., 2018] [Nicolas-Alonso et al., 2015] [Rong et al., 2018a] [Vidaurre et al., 2011] recording systems, only few adaptive decoders were developed for ECoG recording systems [Eliseyev et al., 2017]. Additionally, most reported adaptive algorithms designed were restricted to linear decoders, which may be limiting for complex effector control with a high number of DoFs. Therefore, deeper investigation on the development of closed-loop decoder adaptation must be achieved. Moreover, more online clinical experiments based on adaptive decoder must be carried out for BCI application, as this is the only solution to evaluate the impact of patients and decoder mutual adaptation.

## 1.5. BCI project at CLINATEC

The experiments and results presented in this manuscript were part of CEA-Grenoble\Leti\CLINATEC clinical trial: "BCI and tetraplegia". CLINATEC is a laboratory of the CEA-Grenoble in partnership with the University Grenoble Alpes (UGA) and University Hospital of Grenoble (CHUGA). This ongoing clinical trial was approved by the French competent authorities and is referred under the identifier NCT02550522 in ClinicalTrials.gov ["ClinicalTrial NCT02550522," n.d.]. The clinical protocol which started in end 2015 plans to include five patients in five years.

### 1.5.1. Concept and context

"BCI and Tetraplegia" clinical trial aims to bring the proof of concept that a tetraplegic patient can control a complex effector in real-time using ECOG recording system and direct neural decoding strategy. To succeed, innovative chronic epidural electrocorticographic recording arrays and a complex motorized multi-limb exoskeleton effector were designed. After preclinical experiments on monkeys [Eliseyev et al., 2014],

the clinical trial started and to this date, three tetraplegic patients were included in the protocol and underwent bilateral implantation of chronic wireless epidural electrocorticographic arrays. Two of them are still training using our homemade BCI platform in order to control multiple effectors (3D Avatar, exoskeleton and wheelchair, etc.).

## 1.5.2. Principles / Approaches

The principle of the presented BCI platform is shown in Figure 1-13. The patient neural activity is monitored through two wireless, safe and chronic "semi-invasive" epidural ECoG implants located above the dura matter of the motor cortex of both encephalon which digitize and pre-filter the neural signals. These digitized neural signals are recorded and processed using a signal processing software platform (signal translation block) which sends commands to a complex effector (four-limbs exoskeleton) to perform the attempted action of the patient and provides visual feedback to him (closed-loop BCI).

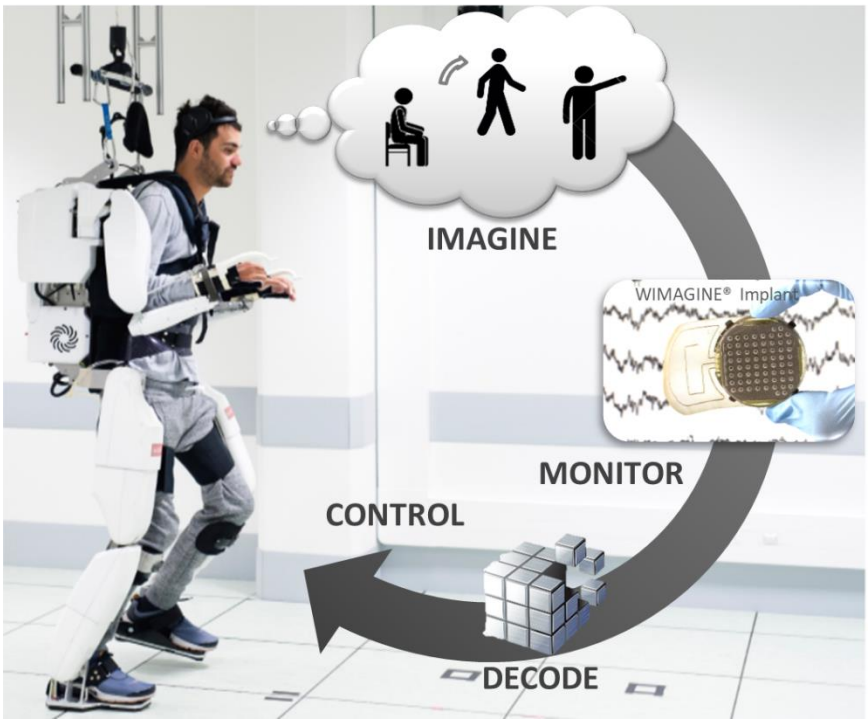



Figure 1-13: Clinatec BCI platform with the four main pillars of the project : imagine, monitor, decode and control [Eliseyev et al., 2014].

## 1.5.3. Project specific requirements

Taking into account the objective of the BCI project, numerous specific claims emerged. Without considering the obvious requirement related to semi-invasive ECoG recording

systems and exoskeleton safety, several needs related to the BCI transducer arose (Figure 1-14).

Firstly, the transducer must apply to real-time uses which restricted the possible pre-processing, feature extraction, decoding and post-processing steps to low computational burden with sufficiently simple and straightforward algorithms in order to apply the transducer block (decision rate) in about a hundred millisecond time scale.

Secondly, as mentioned in 1.4.2.3, the asynchronous attribute is a major characteristic for BCI applications which control real/physical effectors. The BCI project of CLINATEC aims to control a complex multi-limb exoskeleton. Such an effector must be active and responsive to any generated command and additionally, remain static if the patient decides not to attempt any movement. Synchronous BCI systems can only be used in specific prefixed time periods which is incompatible with this application.

To control an exoskeleton with several limbs, it is mandatory to allow the patient to command independently each limb with strong discrimination between the activation of each limb. Indeed, if the patient is focusing on precise manual tasks, the lower limb must stay inactive without any unwanted activation.

The decoder used to translate the ECoG neural signals of the patient into commands should be able to control a complex effector with a high number of dimensions. The control of the four limbs of the exoskeleton represents numerous DoF to decode such as 3D space for the left hand, 3D space for the right hand, both wrists rotations, grasping, etc. The decoder must be sufficiently efficient to control each DoF using a direct neural decoding strategy.

Finally, BCI transducer must be as stable as possible to perform robust neural signal decoding without recurrent calibration sessions. Decoder with stable performance over days, weeks, months, etc. are easier to use and less frustrating for the user.

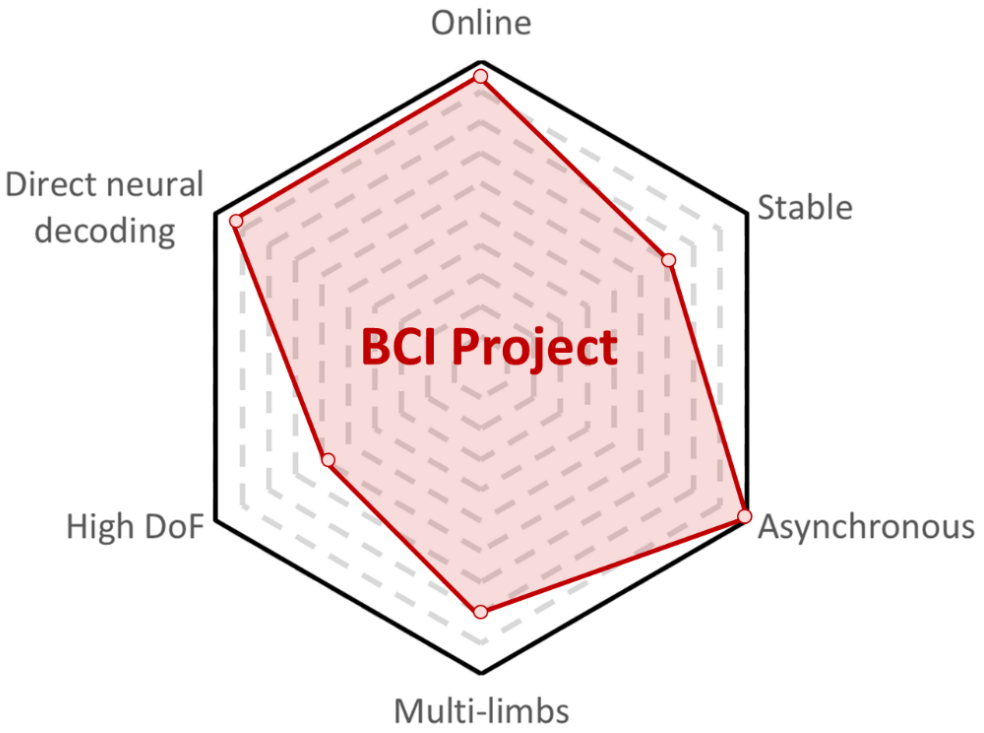



Figure 1-14: Main requirements of the CLINATEC BCI project "BCI and Tetraplegia".

## 1.6. PhD goals

The following doctoral work was completed within the framework of CLINATEC clinical trial "BCI and tetraplegia" and the motor Brain-Computer-Interface (BCI) project. The presented results were obtained based on the online experiments recorded with one of the patients of the clinical trial from mid-2017 to mid-2020. This doctoral work is mainly focused on the development of new innovative decoders which suited the requirements and objectives of the "BCI and Tetraplegia" clinical trial. The proposed decoders were designed in order to create an online adaptive asynchronous algorithm for high multi-limb effectors control and meet the requirements of the BCI clinical trials stressed in the Figure 1-14.

### 1.6.1. Real-time application

To imagine BCI daily life application, the BCI platform must be sufficiently efficient (low computational time) to provide to the patient a transparent control of the effector. Consequently, the online application of the decoder may be limited to simple and/or efficient algorithms [Murphy et al., 2016]. Nevertheless, these decoders are generally linear models, which might be limited for complex multi-limb control objectives proposed in the "BCI and Tetraplegia" clinical trial. Online decoding restriction is one of the major requirements for the clinical trial and every algorithms proposed in this

manuscript were especially focused on real-time application purpose and related restrictions (e.g. decision rate below 300ms).

### 1.6.2. Asynchronous multi-limbs decoder

As mentioned in 1.4.2.3, the majority of the state of the art high dimensional control experiments were performed based on synchronous limited center-out tasks and only few online continuous decoders integrated asynchronous control. However, during daily life applications, numerous situations require to sit back and not perform any movement (queue up, take the subway, rest, etc.). Moreover, in the case of real effector control (prosthetics, FES, wheelchair, exoskeleton, etc.) which has direct interaction with the user, false activations may lead to particularly disturbing and stressful situations. False activations must remain exceptionally rare events. As an example, in the case of gait cycle initiated with a false positive activation, the following false-positive activations during the gait cycle will not be taken into account (as the patient is already walking). In this situation, rare long false activations are less disturbing than high-frequency false activations.

Additionally, daily life actions commonly require bimanual (or generally multi-limb) movements. Multi-limb decoding is a poorly explored area of the BCI field compare to single limb movement decoding. Providing tetraplegic patients with simultaneous or alternative bimanual and walk control will greatly enhance patient's mobility, independence and improve their quality of life. Therefore, if a patient is attempting a high precision task with the left arm of the exoskeleton, the multi-limb decoder must be able to compute zero-velocity outputs for the other limbs. In the scope of the CLINATEC clinical trial, the new online decoders presented in the manuscript must integrate asynchronous control and highly efficient limb discrimination to enable a tetraplegic patient to perform a stable idle state (IS) and alternative multi-limb tasks at his earliest convenience.

### 1.6.3. Online incremental adaptive decoder

Finally, one of the BCI project requirements is the stability and robustness of the model across time and experiments even though it is well known that patient neural signals are non-stationary signals. The proposed solution is to design an adaptive decoder which re-estimates the model parameters across experiments. The reported adaptive decoder brought several interesting properties in the BCI field.

As mentioned in section 1.4.3.3, the adaptation of the decoder during closed-loop experiments lead to different model parameter convergence, better performance compared to decoders trained offline during open-loop experiments [Lebedev and Nicolelis, 2017, 2006] [Murphy et al., 2016] [Orsborn et al., 2014], easier/faster training procedure [Brandman et al., 2018] and adaptation to neural signal variations across time.

The present Ph.D. study is specifically focused on the previously presented decoder characteristics which also respond to the clinical trial requirements. In order to perform online closed-loop experiments, all the decoders designed in the Ph.D. researches are oriented towards online adaptive closed-loop asynchronous multi-limb BCI applications.

## 1.7. Manuscript organization

The doctoral manuscript presents the new incremental adaptive asynchronous multi-limb decoders implemented in online closed-loop experiments with a tetraplegic patient or tested offline in pseudo-online decoding performance comparative studies. Chapter 2 presents in detail the "BCI and Tetraplegia" clinical trial from the paradigm of control to the training timeline. Chapter 3 reports state-of-the-art BCI transducers (preprocessing, feature extraction and decoder). The new decoders designed are detailed in Chapter 4, 5 and 6 whereas experiments description, integration of the decoder into the CLINATEC online BCI platform and decoder performance evaluation are presented in Chapter 7, 8 and 9 respectively. Chapter 10 clusters the results of each decoder. Finally, Chapter 11 regroups the discussion, the added value of this study, the implications of the presented results in the BCI field and the research perspective.

Chapter 2

# Scientific context: BCI and Tetraplegia clinical trial

**Contents**

As mentioned is the first chapter, "BCI and Tetraplegia" clinical trial was created with the purpose to provide the proof of concept that a tetraplegic patient implanted with epidural ECoG can control a complex multi-limb effector through direct neural decoding strategy. The ongoing clinical trial was cataloged the 11/09/2015 in the publically accessible register named ClinicalTrials.gov, under the identifier: NCT02550522 ["ClinicalTrial NCT02550522"] ["ICTRP clinical trial NCT02550522"]. The clinical trial was approved by the French authorities: "Agence Nationale de Sécurité du Médicament et des produits de santé (ANSM)" with the registration Number: 2015-A00650-49 and an ethical committee (Comité de Protection des Personnes - CPP) with the Registration number: 15-CHUG-19. The informed consent for the clinical trial was obtained from the patient as well as the consent to publish the information/image(s) in online open access publications.

This chapter presents all the information related to the clinical trial and the research environment of the Ph.D. study. Particularly, this chapter is focused on the patients' condition, the BCI recording system, the experimental environment, the global software chain in which the decoding algorithms proposed in this manuscript were integrated and the controlled effectors (Figure 2-1).

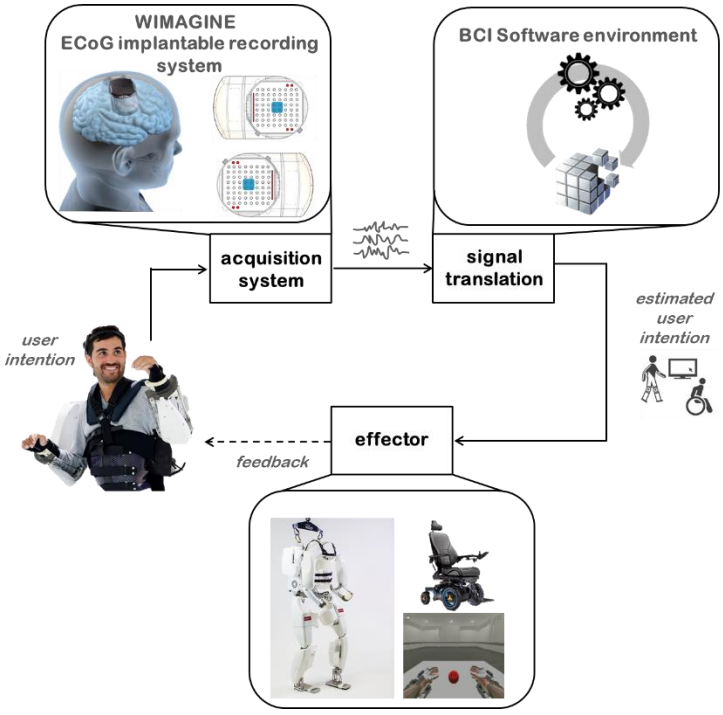



Figure 2-1: CLINATEC "BCI and Tetralegia" specific BCI platform.

## 2.1. Inclusion criteria of the clinical trial

According to the inclusion criteria of the clinical trial defined by the Principal investigator and formulated in the World Health Organization (WHO) International

Clinical Trials Registry Platform (ICTRP) ["ClinicalTrial NCT02550522," n.d.] ["ICTRP clinical trial NCT02550522," n.d.], the participants must be a French fluent male or female patient between 18 and 45 years old with stable neurological deficits. Moreover, the patients should claim a need for additional mobility, be registered in the French social security system, have stable ambulatory or hospitalized monitoring, and signed informed consent [Benabid et al., 2019] ["ClinicalTrial NCT02550522," n.d.] ["ICTRP clinical trial NCT02550522," n.d.]. Previous brain surgery, anticoagulant treatments, neuropsychological sequelae, depression, substance dependence or misuse, and contraindications to MEG, EEG, or MRI were considered as exclusion criteria

## 2.2. Participants of the clinical trial

Between the start of the clinical trial and mid-2020, three patients were included in the clinical protocol.

The first tetraplegic patient was successfully implanted in May 2016 with two WIMAGINE implants. After the surgery, the implants of the first patient stopped communicating with the base-station. Further investigation highlighted technical issues in the firmware of the microcontroller unit. The recording systems were explanted and the patient was excluded from the clinical trial. The issues were corrected and after ANSM and ethics committee revision, the clinical trial was authorized to restart in February 2017. Further details are given in the supplementary data of [Benabid et al., 2019].

The second subject, recruited in mid-2017, was a 29-year-old right-handed male with traumatic sensorimotor tetraplegia caused by a complete C4–C5 spinal cord injury 2 years prior to the study. The patient can perform neck, shoulder and small upper limb movements by contraction of the biceps at the elbow and extensors of the wrists. American Spinal Injury Association Impairment (ASIA) scores [Roberts et al., 2017] evaluation was achieved under the supervision of Professor Benabid and Professor Chabardès. The contraction of the biceps close to the elbow was scored at 4 and 5 for the right and left body side, whereas extensors contractions were rated at 0 and 3 for the right and left wrists respectively. With the exception of the cited muscles, all other muscles below were scored 0 on the ASIA scale. Moreover, the sensory-motor deficit was complete (Figure 2-2) [Benabid et al., 2019].

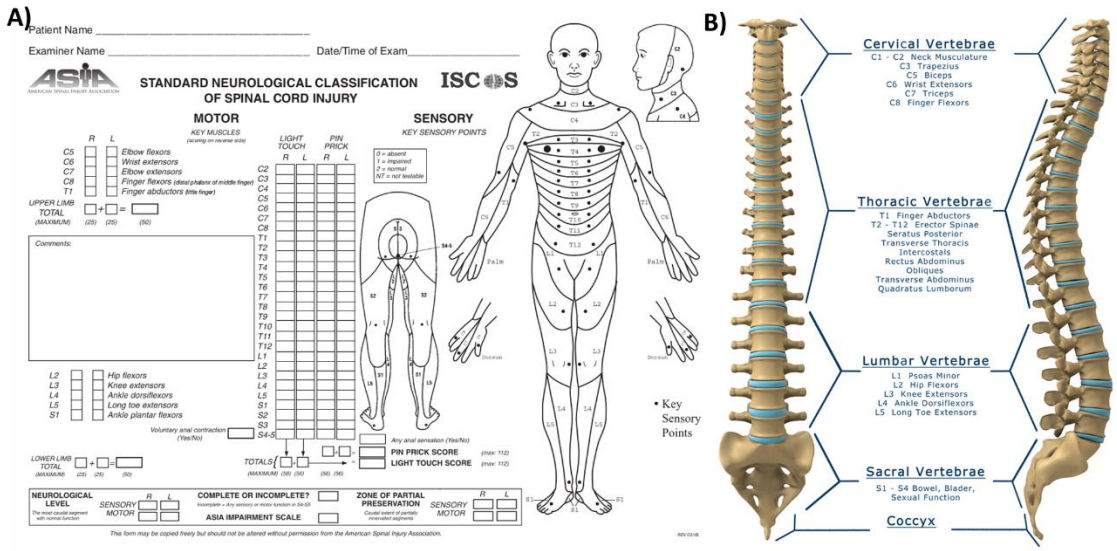



Figure 2-2: Typical sheet for American Spinal Injury Association Impairment (ASIA) score evaluation and typical muscular group functions with the corresponding vertebrae. (A) ASIA determines the functional impairment resulting from a spinal cord injury through a myotomal-based motor examination, dermatomal based sensory examination [Roberts et al., 2017]. Motor examination grades five specific muscle groups in the upper body side and five specific muscle groups in the lower body side using a score going from 0 to 5 [Roberts et al., 2017]. (B) Lesion in a vertebrae lead to muscular deficiency. The figure displays for each vertebrae the muscular groups related to.

A third subject was included to the study in late 2019. Nevertheless, at the end of the Ph.D. experimental period, the third patient was in an early training phase and was consequently not yet able to perform the complex task related to the Ph.D. study. Therefore, the proposed Ph.D. study is only focused on the experiments performed and the results obtained with the second patient.

## 2.3. Implantation

The patients underwent bilateral implantation of two long-term safe and chronic wireless implants for epidural ECoG signal recordings under general anesthesia using Image Guided Functional NeuroSurgery on May 2016, June 21th 2017 and November 19th 2019, respectively. The epidural ECoG wireless implants named WIMAGINE were implanted into the skull in contact with the dura mater within a 25 mm radius craniotomy placed in front of the sensory motor cortex (SMC) area. The subjects' SMC were localized clearly using functional imaging (fMRI and MEG) as they imagined upper and lower limb virtual movements or performed real motor tasks when possible. 100 and 80 trials were performed with MEG and fMRI respectively to optimize the implants positioning before the surgery. Details of the protocol are provided in the previous study [Benabid et al., 2019].

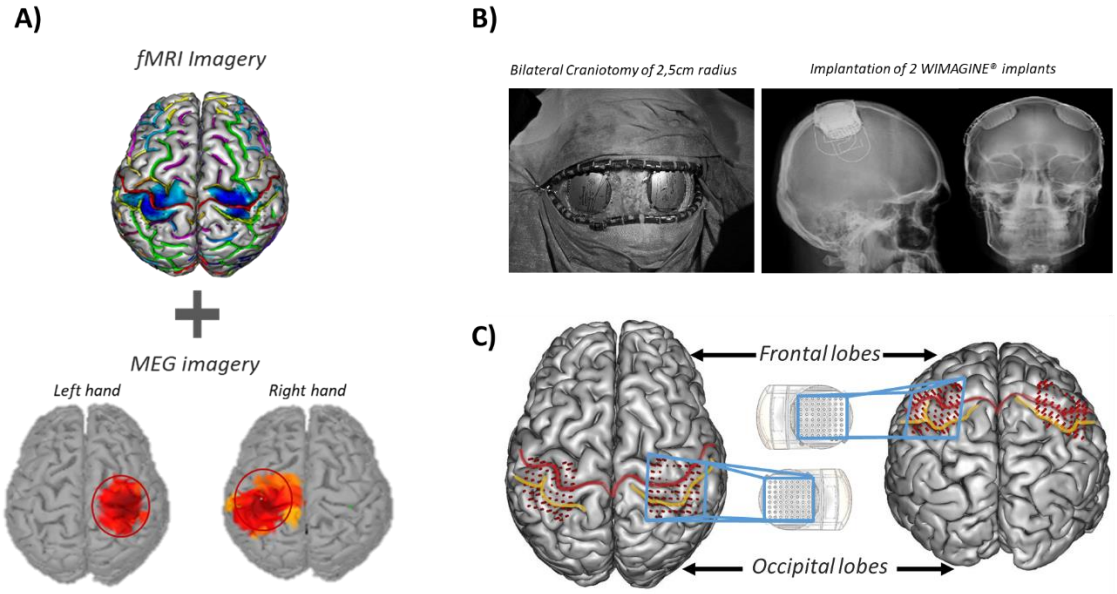



Figure 2-3: Offline study achieved to localize the sensory motor cortex of the patient before the surgery. A). Localization of the optimal electrode position before surgery through MEG and fMRI motor Imagery study. B). Bilateral implantation using Image Guided Functional NeuroSurgery of two WIMAGINE implants. C) Localization of the electrodes array after surgery compared to the sensory sulcus (SS) and the motor sulcus (MS) represented in yellow and red curves respectively.

The second patient implanted in June 21, 2017 started the training procedure in early July and since has been training for more than 36 months to control several effectors with various complexity.

## 2.4. Experimental platform

### 2.4.1. Recording system

The primary challenge of motor BCIs for clinical and daily-life applications is the development of safe, chronic and stable neural recording systems over time. In order to fulfill these requirements for chronic brain neural signal processing, CLINATEC designed an innovative wireless epidural ECoG recording system named Wireless Implantable Multi-channel Acquisition system for Generic Interface with NEurons (WIMAGINE) [Mestais et al., 2015].

WIMAGINE is an active implantable medical device designed to be implanted into the skull above the dura matter to record epidural ECoG brain signals. The recorded epidural ECoG signals are transmitted wirelessly to a custom designed base station connected to a computer [Mestais et al., 2015]. WIMAGINE implants are composed of

two main structures. The recording part is a 50-mm diameter silicone-coated titanium cylinder with a flat internal surface with 64 electrodes for ECoG recording. The 64 plane platinum iridium 90/10 electrodes have a 2.3 mm diameter and have a inter-electrodes distance of 4 and 4.5 mm on the lateral and antero-posterior directions, respectively [Sauter-Starace et al., 2019]. The digitized ECoG signals are low and high pass filtered in a bandwidth from 0.5Hz to 300Hz, amplified, cleaned thanks to an anti-aliasing filter and clustered into buffers [Robinet et al., 2011]. All these operations are performed using an ASIC CINESIC32 integrated into the implants (Figure 2-4). This integrated circuit presents a low noise amplifier (0.7 µVRMS) which requires low power alimentation (32µA per channel for a global consumption of 3mA with 32 active channels).

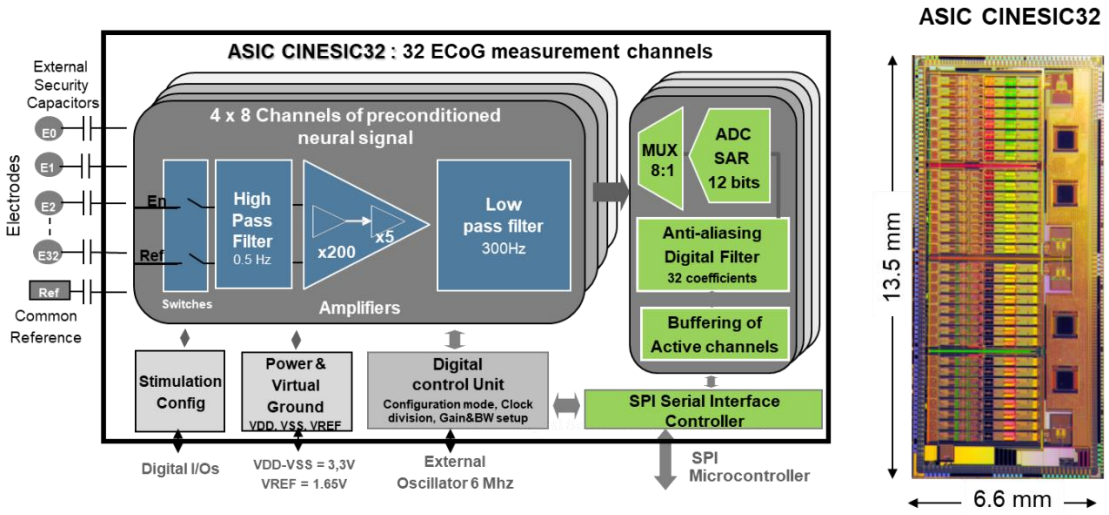



Figure 2-4: details of the ASIC CINESIC32 integrated into the wireless implant WIMAGINE[Robinet et al., 2011]. Signal to noise ratio in improved through the steps of High-pass filtering, amplification, low-pass filtering, etc.

The second part of the WIMAGINE implant is a silicon film containing high frequency (HF) antenna (13,56Mhz) for transcutaneous remote power supply and an ultra-high frequency (UHF) antenna (402-405 Mhz) for wireless data transfer. Remote power supply up to 100mW is provided to the implants through an inductive link with an external antenna integrated into a helmet worn by the patient (Figure 2-5).

Limited data rates, caused by restricted radio link (≤250 kb/s) narrowed the real-time transmission of the neural signals to a maximum of 32 electrodes by implant simultaneously recorded at a 586 Hz sampling frequency.

All the required European directive 2007/47/EC and ISO standards regulation for clinical trials applications have been obtained concerning biocompatibility (ISO10993), mechanical and electrical safety (ISO45502-1, ISO60601-1, ISO14708-1), software reliability, risk management of medical device (IEC62304, ISO14971), manufacturing

process (ISO13485) and electrical safety and electromagnetic compatibility of the external unit (NF EN 60601-1).

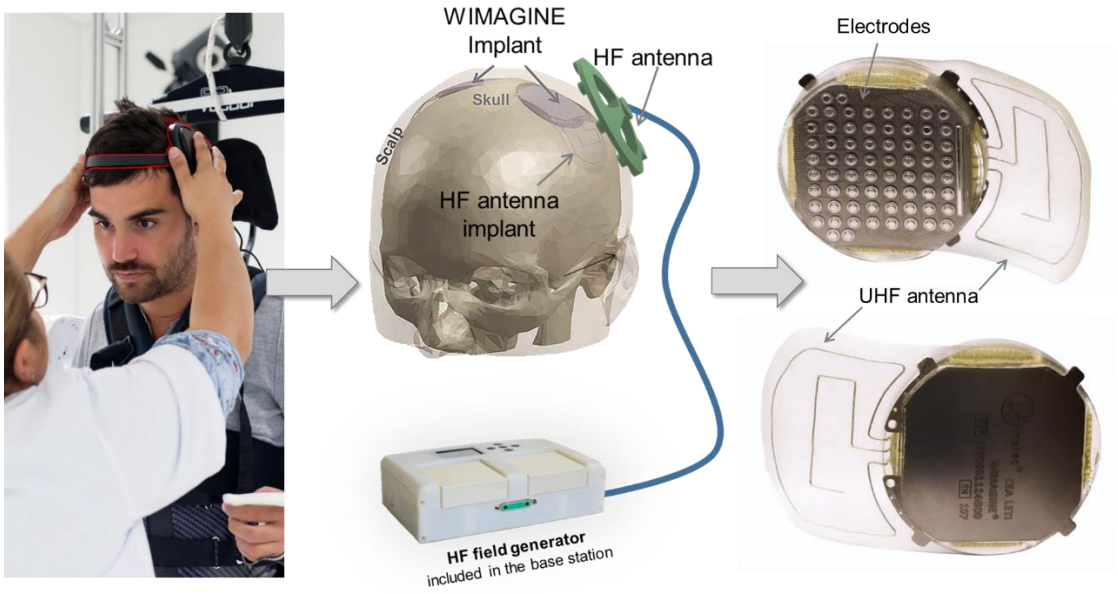



Figure 2-5: Global recording chain composed of the helmet worn by the patient. The helmet integrate a high frequency (HF) antenna for transcutaneous remote power supply and wireless data transfer with the WIMAGINE recording implants. The recorded neural signals are transmitted to a control base station which generates the HF field and sends the recorded neural signals to a computer to start the signal processing steps.

## 2.4.2. Software chain

The software chain is composed of four main components. The digitized neural signals from the WIMAGINE implants are sent to the Wireless Implant Software Control Interface (WISCI) which receives the ECoG neural signals, synchronizes the channels and formats the neural signals to real-time batch processing. In a second step, the formatted ECoG signals are treated by the Adaptive Brain Signal Decoder (ABSD) which processes the ECoG signals using an online adaptive model to estimate the patient's intended movements and control an effector. ABSD regroups the steps described as the classical signal processing block in the general architecture of BCI systems (pre-processing, feature extraction, etc.). ABSD was designed to support adaptive calibration procedures. ABSD updates an adaptive decoder using incremental batch learning in the background while decoding continues. However, ABSD requires a computationally light and optimized adaptive decoder to work. The orders generated by ABSD are sent to the EMY Motion Manager (EMM) and EMY Motion Controller (EMC) to generate appropriate motor commands for the exoskeleton. In the case of virtual effectors, an

adequate portal software is used to generate commands and control the virtual effector (Figure 2-6).

From the neural signals recorded at 586 Hz, the software chain extracts an order to send to the effector at a 10Hz decision rate. Moreover, the commands performed by the effector (limb activated, current hands positions, wrists angle, etc.) are recorded at 10 Hz and could be used for adaptive model calibration.

Every analysis and online experiment, including training and decoding, was performed with Matlab2017b® and Visual Studio 2015 using an Intel Xeon E5-2620v3 computer with 64 GB RAM.

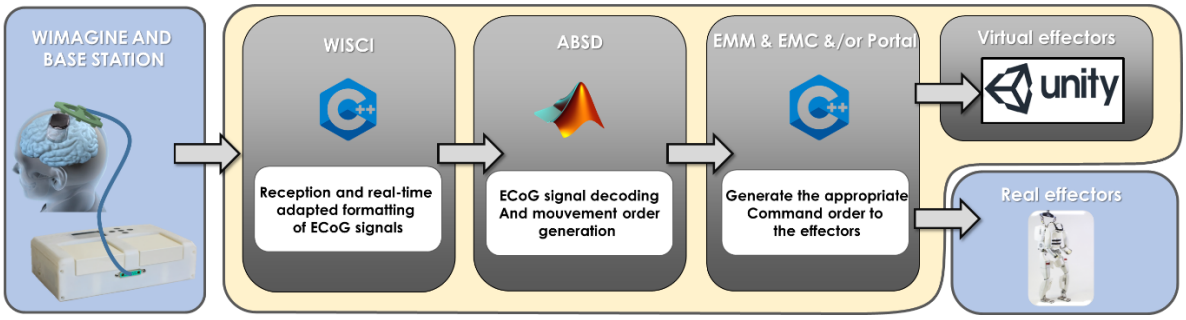



Figure 2-6: Clinatec BCI platform for real-time BCI experiments. Global Software chain (shaded in yellow) allowing to tranform ECoG neural signals into commands for the effectors.

## 2.4.3. Effectors

Several effectors were designed to be controlled by the patient during his training. The effectors can be clustered into the virtual and real effector categories.

### 2.1.1.1. Real effector

The purpose of the "BCI and Tetraplegia" clinical trial is to prove that a tetraplegic patient can control a complex multi-limb effector. The Enhancing MobilitY (EMY) exoskeleton adapted by the CEA/LIST is a wearable fully motorized four-limb robotic neuroprosthesis weighting 65 kg designed to be driven by the decoded ECoG brain signals[Benabid et al., 2019] during the clinical trial (ISO60601) [Benabid et al., 2019].

Upper limb control of both arms presents high movement amplitudes with 65, 105, 105, 100 degrees for shoulder rotation, shoulder and elbow flexion/extension and prono-supination respectively [Benabid et al., 2019] [Morinière et al., 2015]. Each arm can be controlled using angular or Cartesian end-point coordinates. Lower limbs of the EMY exoskeleton are controlled through a configurable walking cycle. Additional grasping/prehension systems were integrated to the EMY exoskeleton for object manipulation experiments.

Independent control of the 3D Cartesian endpoint trajectory of the arm, wrist rotation, open/close hand shape for both arms, walking, sitting, and rest state represent 13 DoF.

A battery and a computer station receiving ECoG radio-emitted signals are embedded in the back of the exoskeleton. The neural signals samples are decoded and translated into incremental velocity endpoint-control commands through the decoding software. The control commands are converted into joints movements by the exoskeleton control system activating the limbs and producing the appropriate movements.

To this date, EMY exoskeleton does not manage the balance of the prosthetic and it requires ceiling-mounted support to enable the patient to practice standing experiments.

To mention, the second patient controlled other real effectors such as the Kinova JACO assistive robotic arm and a wheelchair, nevertheless, these performances will not be precisely detailed in the manuscript.

#### 2.1.1.2. Virtual effectors

The virtual effectors used during the clinical trial were designed to train the patient to control from 1 to 13 DoF of the exoskeleton. Several video games such as walking simulators, car racing games, 1D and 2D cursor control were created to train the patients. Moreover, a virtual avatar reproducing the exoskeleton dimensions was integrated into the BCI platform. The virtual avatar is a replica of the exoskeleton and can perform every movement similarly to the exoskeleton. All the virtual environments were coded using UNITY® software.

In the entire Ph.D. manuscript, the experiments using virtual effectors were performed using the virtual avatar.

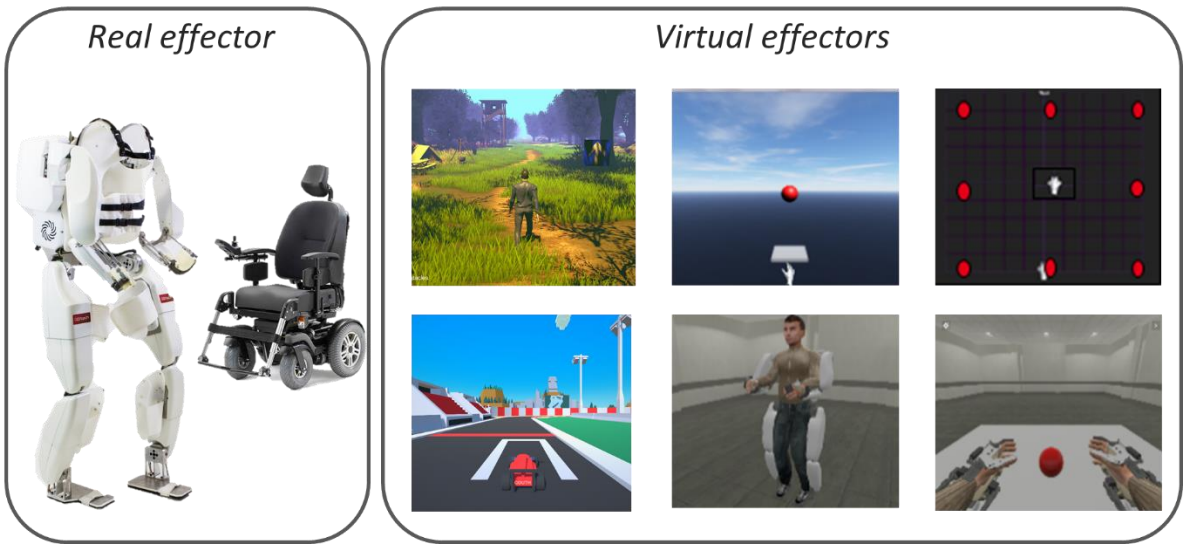



Figure 2-7: Real and virtual effectors available in the BCI and tetraplegia project. Real effectors cluster the exoskeleton, the wheelchair, etc. whereas the virtual effectors regroup the virtual avatar the car racing video game, etc.

## 2.5. Progressive patient Training and Timeline

The patient's training began in July 2017 and is ongoing to this date. During this period, the trainings in the laboratory and at the patient's home were performed in parallel. Patient's training at home was performed every week (two days per week). Home experiments were focused on virtual effector control. Experiments inside the laboratory occurred, on average, once a month (three days in a row) and focused on real effector control. During laboratory experiments, various tasks were proposed to the patient in order to control the different DoF of the exoskeleton such as walking, moving the arms in sitting position etc.

One of CLINATEC purposes is to control all the EMY exoskeleton DoF. The training strategy followed was to increase gradually the patient's control on the effectors, unlocking new DoFs when the patient achieved good control of the effector. The progress of the patient was investigated in terms of the number of DoFs controlled over time [Benabid et al., 2019]. To this point, numerous experimental paradigms were created to control the limbs of the exoskeleton (Table 1). With the exception of the "Walking" command, all the experimental paradigms can be executed with the exoskeleton in a standing or sitting position.

Table 1: Several control paradigms and associated controlled dimension used for patient's training during the clinical trial.

| Paradigm | Type of Control | Controlled dimension |
|---|---|---|
| **Cartesian left hand translation control** | Continuous | 1D or 2D or 3D |
| **Cartesian Right hand translation control** | Continuous | 1D or 2D or 3D |
| **Angular left wrist rotation control** | Continuous | 1D |
| **Angular right wrist rotation control** | Continuous | 1D |
| **Left grasping control** | Discrete | 1D |
| **Right grasping control** | Discrete | 1D |
| **Walking control** | Discrete | 1D |
| **Idle/rest control** | Discrete | 1D |

Once a DoF is controlled independently, it is added to the pool of controlled DoF and the patient is trained to control all the paradigm in the same experiment. As an example,

considering that the patient was able to control the 3D Cartesian left and right hand translation paradigm in different experiments (3D control experiments), a new paradigm with alternative control of both arm in the same experiment was created and tested (6D control), etc. This procedure was similar for both home and laboratory patient's training.

Home and laboratory experiments bring different feedback to the patient which may lead to different model parameters convergence. Therefore, the models calibrated during experiments inside the laboratory were only used during laboratory experiments. Similarly models created during home-based experiments were not used for laboratory experiments. The incremental training strategy as well as the number of continuous DoF controlled by the second patient across the two first years of experiments is presented in Figure 2-8.

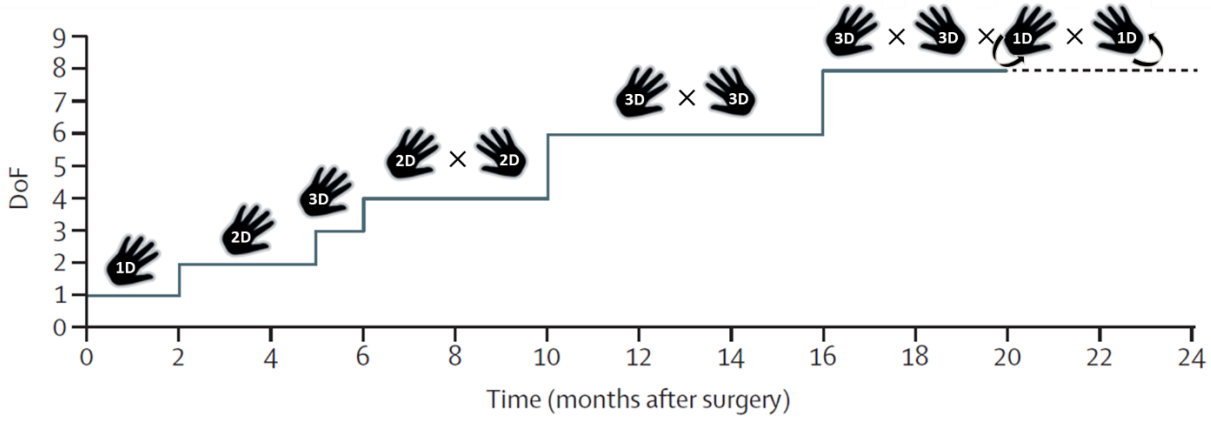



Figure 2-8: Evolution of the experiment paradigm and controlled DoF of the second patient across time. The figure is extracted from [Benabid et al., 2019].

## 2.6. Experimental session procedure

Experiments were divided into several steps. Firstly, the patient was settled in the exoskeleton or in front of the television where the virtual effectors were displayed. The recording helmet (including the HF antennas) was placed on the patient and signal quality was checked. Then, the BCI session started. A clinician used ABSD software environment to select an experiment with specific controllable DoFs. The BCI session was divided into two main phases. The first optional phase was the model calibration/training period. Calibration period aims to create a mathematical model which evaluates the correct command to send to the effector depending on the neural signals modulation of the patient. This phase was performed during the online control of the effectors by the patient and was optional as model created during previous experiments can be applied. The experimenter, depending on the

performance of the model stopped the calibration phase to start the second step of the experiment: the test phase. The test phase was used to evaluate the control performance of the patient with a fixed model. The tasks to complete were similar than the one proposed in the training phase. The only difference was that the decoding model was fixed and not updated anymore. To mention, contrarily to most of the state-of-the-art experiments, the patient could talk and interact with the experimenter without any restrictions during both phases.

A BCI session was composed of multiple tasks such as idle, left or right hand translation movements, left or right wrist rotations etc. The task to complete was selected by the experimenter. Each task was composed of several trials defined as a specific action to perform (e.g. reach the target on the left corner, rotate the wrist to a specific angle etc.).

During virtual effector based experiments, the targets to reach by the patient selected by the clinician were represented thanks to virtual spheres, gauges or icons depending on the task to complete. Using the exoskeleton, targets were represented thanks to lightened LED distributed on a panel. Fixed lightened LED corresponded to a hand reaching task (with left or right hand translation task depending on the side of the lightened led on the panel. Rotating flashing LEDs corresponded to a wrist rotation task (left or right wrist rotation depending on the side of the lightened LEDs on the panel). The direction of rotation to achieve was the same that the order of the flashing LEDs (clockwise or counterclockwise). Virtual avatar and exoskeleton left hand translation and right wrist rotation tasks are represented in the Figure 2-9.

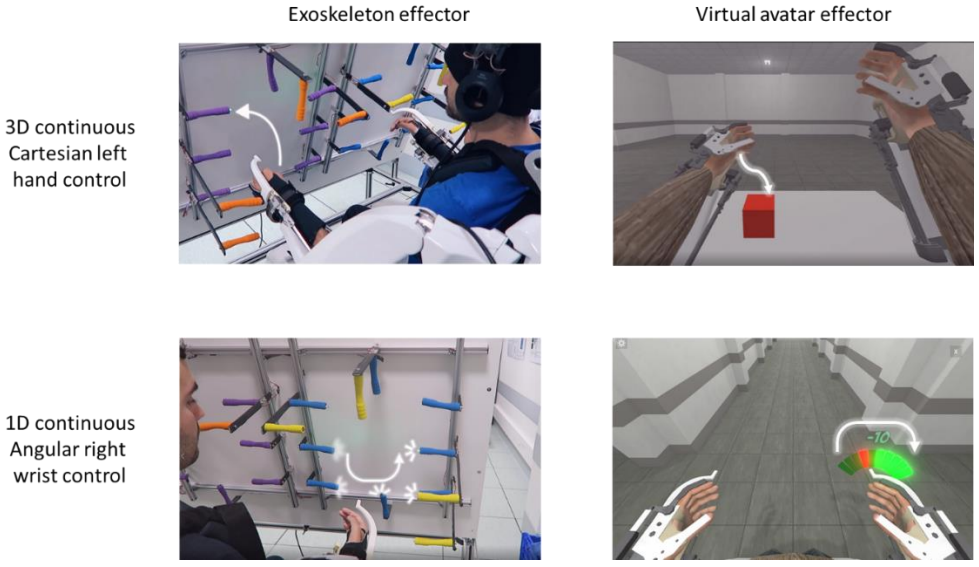



Figure 2-9: Example of a trial in a continuous left hand movement task and angular wrist movement task using exoskeleton or virtual effector. During a hand movement task with exoskeleton effector, the target to reach in the trial is showed to the patient using a lighted LED. Virtual avatar effector target of the left hand movement are represented with a red cube whereas it is represented with a blue sphere during right hand movement trial. Wrist rotation task is

ordered to the patient using flashing LEDs during experiments with the exoskeleton. Wrist rotation target is displayed with a gauge in the virtual environment.

For the control experiments based on exoskeleton effector, the instruction to the patient were provided through LEDs distributed on a panel. A total of 28 LEDs (14 for each arm) were placed into two 20cm x 26 x 31 cm cuboids. Some target locations were removed for the patient's comfort. Six LEDS located in the plane close to the patient were removed: the ones placed in the middle of the panel as well as the LEDs in the top left and right corner (Figure 2-10A). Therefore, the exoskeleton experiments presented in this manuscript were based on reaching tasks with 11 possible target locations per hand. As a safety precaution to avoid any patient's inconvenience and exoskeleton collision with the experimental environment, exoskeleton movements were limited. Theoretical physical limits as well as imposed movement restrictions of the exoskeleton are represented in Figure 2-10B.

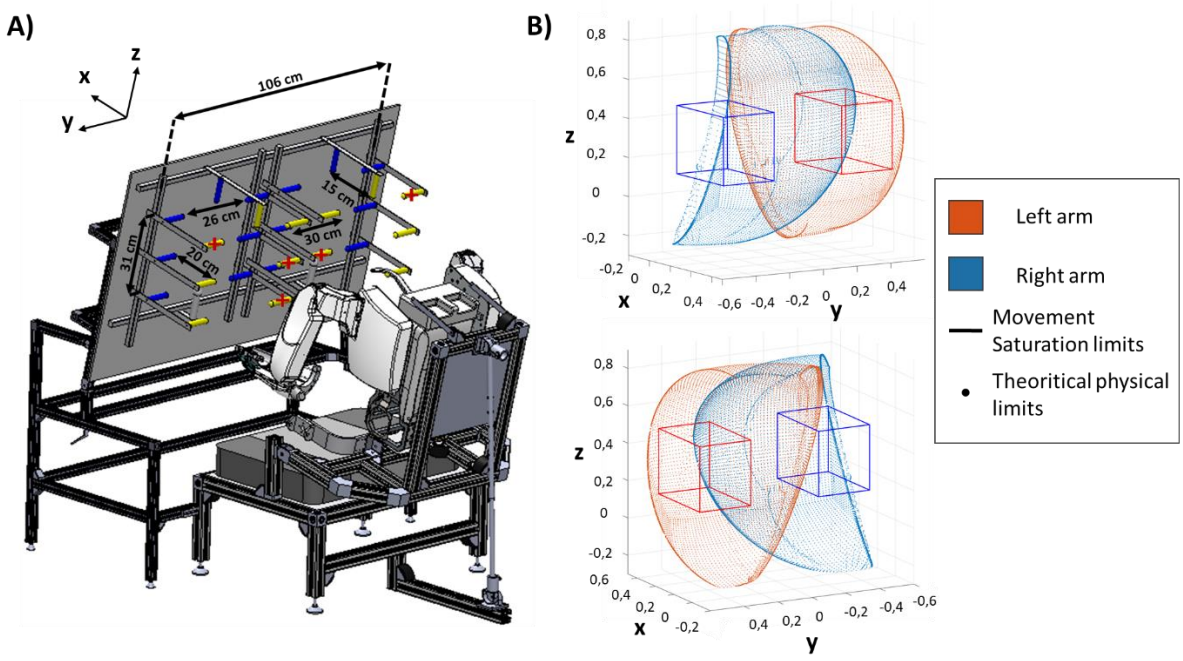



Figure 2-10: Experimental paradigm for online BCI control of the exoskeleton in a sitting position. A) Representation of the exoskeleton position in the case of 3D continuous left or right hand control tasks. The panel was placed in front of the exoskeleton while the LEDs were located at the end of the blue and yellow tubes displayed on the figure. B) Representation of the theoretical physical limits of the exoskeleton and the saturation limits imposed during the left and right hand translation tasks in order to avoid any uncomfortable arm position for the patient.

Chapter 3

# Transducers in Brain Computer Interfaces

**Contents**

As previously mentioned, numerous transducer strategies were designed for BCI applications. Given that the transducer is generally dependent on the inter-subject variability, the neuronal patterns, the recording system and the effector to control, abounding variations of transducer strategies were proposed through the past twenty years. Nevertheless, several classical strategies emerged as relevant techniques to extract brain neural signal information. The following chapter provides an overview of the most popular transducers pre-processing, feature extraction and model identification blocks used in the BCI field. Moreover, in regard to the CLINATEC clinical trial and thesis scope, special emphasis is placed on online processing methods and online adaptive transducer identification procedures.

## 3.1. Pre-processing

Pre-processing is generally the first signal processing step which follows the neural signal recording block. Pre-processing aims to enhance the signal quality and improve the signal-to-noise ratio to extract the most relevant features without undesired non-relevant artifacts [Bashashati et al., 2007a]. The neural signal may be amplified, downsampled to reduce the sampling frequency, band-pass filtered in neurologically interesting frequency band [Lotte et al., 2007] [Mutasim et al., 2018], etc. Subsequent filters may be applied to remove artifacts from several sources such as non-voluntary muscular activation, power line noise, etc. particularly for EEG recordings. Power line noise around 50-60 Hz and its harmonics were generally removed through notch, sharp edge, band pass or Butterworth filters in ECoG [Anderson et al., 2012] [Branco et al., 2018, 2017] [Bundy et al., 2016] [Choi et al., 2018] [Degenhart et al., 2018] [Elgharabawy and Wahed, 2016] [Jiang et al., 2017] [Salari et al., 2019] and EEG [Brunner et al., 2006] [Cantillo-Negrete et al., 2018] [Chowdhury et al., 2017] [Chu et al., 2015] [Cozza et al., 2020] [Delisle-Rodriguez et al., 2017] [Faller et al., 2012] [Mishra et al., 2018] [Mobaien and Boostani, 2016] [Nguyen et al., 2019] [Nicolas-Alonso et al., 2015] [Qibin Zhao et al., 2008] [Rong et al., 2018b] [Song and Yoon, 2015] [Wen and Huang, 2017] [Zhao et al., 2008] BCI experiments. For electrical muscle activities such as hand movements or eye blinking, algorithms were designed to detect these artifacts in an unsupervised manner, using for example independent component analysis (ICA) [Moro et al., 2017] [Seifzadeh et al., 2017], or algorithms dedicated to the processing of other types of biological signals such as electromyography (EMG) or electrooculography (EOG) [Hazrati and Erfanian, 2010] [D. Li et al., 2018] [Nguyen et al., 2019], etc.

Common average reference (CAR) is a spatial filter which estimates the average electrical activity measured across all electrodes. Re-referencing is achieved by creating an average of all electrodes and subtracting the resulting signal from each channel. CAR is a classic spatial filter frequently used in neural population recording systems such as LFP [Brandman et al., 2018] [Jarosiewicz et al., 2013] [Milekovic et al., 2018] ECoG [Branco et

al., 2017, 2018] [Chen et al., 2013] [Gunduz et al., 2016] [Hotson et al., 2016a] [Kubánek et al., 2009] [Miller et al., 2009] [Nakanishi et al., 2013, 2017] [Rembado et al., 2016] [P. T. Wang et al., 2016] [Wang et al., 2017] [Zhao et al., 2013a] and EEG [Chu et al., 2015] [Cincotti et al., 2008] [Cozza et al., 2020] [Kim et al., 2018] [Roijendijk et al., 2016] [Wen and Huang, 2017] BCIs. All the transducer steps are highly dependent on the type of recorded neural signals. Pre-processing algorithms specific to MUA and SUA neural signals were reported such as the spike sorting algorithm family which extracts the neuron action potentials from the recorded brain activity. Numerous spike sorting algorithms were designed for BCI experiments. For example, spike sorting algorithm based on amplitude-thresholding with a threshold defined as a multiple of the standard deviation of the neural signals was reported in [Achtman et al., 2007] [Fan et al., 2014] [Kim et al., 2008] [Li et al., 2011] [Orsborn et al., 2014] [Suminski et al., 2013].

Other algorithms were used to perform in the same time the artifact removal and the neuronal feature extraction steps. Principal component analysis (PCA) as well as independent component analysis (ICA) are two neural feature extraction algorithms which were reported for artifact signals rejection [Nicolas-Alonso and Gomez-Gil, 2012].

## 3.2. Neural feature extraction

Neural feature extraction block extracts the neural signal informative characteristics and discards irrelevant components to decode the mental task performed by the patient. Feature extraction can be divided into a feature generation step which extracts the relevant characteristics from the pre-processed neural signals and an optional feature dimension reduction step which selects among the computed features the best/most informative features [Bashashati et al., 2007a] [Nicolas-Alonso and Gomez-Gil, 2012].

### 3.2.1. Feature generation

The feature generation step is highly dependent on the recording system. Differentiation can be made between algorithms dedicated to the SUA/MUA neural signal processing and neural signal population processing.

#### 3.2.1.1. MEA single/multi neuron recording feature extraction

Spike counts is the general feature generation strategy for BCI system based on SUA and MUA recordings. After the spike sorting preprocessing step, spike count on small time windows (from 20ms to 100ms) is performed using a threshold generally defined as a multiple of the signal standard deviation to evaluate the threshold crossing rate [Collinger et al., 2013] [Dangi et al., 2014] [Fitzsimmons et al., 2009] [Gilja et al., 2012] [Hochberg et al., 2012, 2006] [Ifft et al., 2013] [Jarosiewicz et al., 2013] [Orsborn et al., 2014] [Perge et al., 2013] [Simeral et al., 2011] [Willett et al., 2018] [Wodlinger et al., 2015] [Young et al., 2019]. Nevertheless, other features can be evaluated to complete or replace the classic spike

counts features. Young and Willet in [Willett et al., 2018] [Young et al., 2019] added to spike count feature the high-frequency spike power (HFSP) by taking the root mean square of the filtered spike band voltages from 250 to 5000 Hz using a 8th order non-causal Butterworth filter. Another strategy was proposed by Shanechi [Shanechi et al., 2017, 2016] who directly performed BCI decoding on binary spike activity. The spikes were binned in small intervals containing at most one spike in each interval.

#### 3.2.1.2. Population recording: temporal, frequency and spatial feature extraction

**Time features** are the concatenation of sequential instantaneous neural signal samples to represent the neural signal amplitude variation across time. Time features are highly relevant for time-locked event such as ERPs and were commonly used in P300-based BCI [Cozza et al., 2020] [Lotte et al., 2018] [Vo et al., 2018]. Additionally, EEG neural signals decoding were reported based on Slow Cortical Potentials (SCPs) for virtual cursors control, communication-BCI and rehabilitation-BCI [Hou et al., 2019] [Lazarou et al., 2018] [Úbeda et al., 2018, 2017] etc. SCPs are EEG neural signals below 1-2 Hz which can be self-regulated by patients after training [Nicolas-Alonso and Gomez-Gil, 2012]. Finally, sensorimotor rhythm decoding was commonly reported in BCI experiments. In particular, local motor potentials (LMP) and high gamma neural signals modulation were reported to encode velocity information [Gunduz et al., 2016] [Hammer et al., 2013] [Pistohl et al., 2008] [Waldert et al., 2009]. Therefore, they were often used during offline and online BCI experiments such as ECoG based neural signals decoding of arms, grasping, fingers continuous movement preparation or execution and states classification [Flint et al., 2016] [Gunduz et al., 2016] [Kubánek et al., 2009] [Mehring et al., 2004] [Schalk et al., 2007] [Wang et al., 2011]. Times features were generally computed by averaging the signal amplitude in a defined time window in a specific frequency band to isolate precise patterns [Hotson et al., 2016a] [Wang et al., 2012]. Frequency bands were extracted based on finite or infinite impulse response (IIR or FIR) low or band pass filters such as Chebyshev [Ang et al., 2011] [Nagel and Spüler, 2019] [Vo et al., 2018], Butterworth [Cozza et al., 2020] [Gunduz et al., 2016], Savitsky-Golay [Brunner et al., 2006] [Bundy et al., 2016] [Flamary and Rakotomamonjy, 2012] [Pistohl et al., 2008] or Gaussian filters [Chen et al., 2013] [Cozza et al., 2020] [Hotson et al., 2016a], etc. Time features can be extracted from multiple frequency bands using a so-called filter bank strategy which evaluates in parallel sensorimotor rhythms from different frequency bands [Chen et al., 2013] [Nakanishi et al., 2013].

**Frequency and time-frequency features** are based on the variation of the neural signal power across one or several frequency bands [Lotte et al., 2018]. As previously mentioned, a mental task induces variations and specific patterns in low and high frequency bands of the neural signals acquired with population recording systems (LFP, ECoG, EEG, etc.) [Waldert et al., 2009]. Therefore, neural signals frequency and especially time-frequency features were commonly extracted in the BCI field. In the time-frequency

domain, both neural signal amplitude and phase features were reported in BCI and motor-BCI applications.

Neural signals decoding relying on the extraction of amplitude-based features such as instantaneous power or magnitude were extensively studied [Zeng et al., 2019] and is the most represented strategy for BCI applications. To extract relevant information from the neural signal in a specific frequency, the signal can be band pass filtered using single or multiple filters or transformed into the time-frequency domain using non-parametric transformation such as Short Time Fourier Transform (STFT), Wavelet transforms, Hilbert transform or parametric ones (e.g. autoregressive model) [Schaeffer and Aksenova, 2018] [Volkova et al., 2019]. All these technics aim to balance temporal and frequency resolution [Polikar, 1996] [Volkova et al., 2019].

Filter banks use a set of band-pass filters (Butterworth, Gabors, Savitsky-Golay, etc ) to obtain an amplitude representation of the neural signals on several frequency bands with a good trade-off between frequency distortion and temporal delay [Schaeffer and Aksenova, 2018].

STFT algorithm computes the Fourier transform of the neural signal into small segments where the signal is assumed to be stationary. The segment of signals is evaluated via the convolution of the neural signals and a predetermined window function [Polikar, 1996]. Fourier transform decomposes the signals into the linear summation of sinus signals at different frequencies and phases. Nevertheless, Fourier transform provides a power spectral representation of the signals without temporal definition. STFT overcomes this issue by dividing the signal into successive time windows and applies the Fourier transform in each epoch of the signal in time. Nevertheless, a trade-off between time and frequency must be found. STFT with small time windows lead to high temporal resolution but also to low frequency resolution at the same time. Both time and frequency resolutions are constant and fixed with the definition of the studied window length at the beginning of the experiments. STFT was frequently used in motor BCI research based on LFP [Milekovic et al., 2018], ECoG [Flint et al., 2016] [Jiang et al., 2017] [Miller et al., 2009] [P. T. Wang et al., 2016] [Yanagisawa et al., 2012], EEG [Mend and Kullmann, 2012] [Roijendijk et al., 2016] and MEG [Fukuma et al., 2016, 2015] neural signal recordings.

Wavelet transforms (WT) analyzes the signal at different frequencies with different resolutions. WT was designed to overcome the shortcomings of the STFT. Instead of sinus decomposition of the signals in fixed time window, WT uses a wavelet basic function named Mother Wavelet translated and scaled to obtained variable time resolution depending on the analyzed frequency. High frequencies have a better temporal resolution than low frequencies whereas low frequencies have a better frequency resolution than high frequencies [Polikar, 1996]. Continuous wavelet transform was frequently used in BCI studies to extract instantaneous power of MEA [Schwemmer et al., 2018], ECoG [Branco et al., 2017, 2018] [Choi et al., 2018] [Elgharabawy and Wahed,

2016] [Eliseyev et al., 2017] [Eliseyev and Aksenova, 2014, 2016] [Jubien et al., 2019] [Motrenko and Strijov, 2018] [Salari et al., 2019] [Schaeffer and Aksenova, 2016a] [Shimoda et al., 2012] [Zhao et al., 2013a] and EEG [Ebrahimpour et al., 2012] [López-Larraz et al., 2014] [Oliver et al., 2013] [Robinson et al., 2013] [Sreeja et al., 2019] neural signals. A comparative study based on ECoG neural signal processing highlighted that WT may provide better frequency resolution than STFT or autoregressive analysis [Motrenko and Strijov, 2018].

The Hilbert transform computes from the neural signals a so-called analytic signal defined in the complex domain. A spectro-temporal representation of the signal may be obtained by decomposing it into neighboring frequency components (through band-pass filters) and by computing the so-called analytic signal of each component via the Hilbert transform [Bruns, 2004]. Hilbert transform was applied in several ECoG [Hotson et al., 2016a] [Jiang et al., 2017] [Pistohl et al., 2008] [Saa et al., 2016] [Saa and Çetin, 2013] and EEG studies [Andreu-Perez et al., 2018] [Brunner et al., 2006] [Ma et al., 2020].

AutoRegressive (AR) method is a parametric model commonly used to extract frequency components from neural signals. AR models the signals as the random output signal of a linear time invariant filter [Nicolas-Alonso et al., 2015]. Since different tasks produce different brain activity the AR estimated filter coefficients between two tasks are different and can be used as features for BCI decoding or transform into an estimate of the signal power spectrum. Whereas AR had a superior resolution for small time-windows than STFT, it highlighted issues for non-stationary signals [Nicolas-Alonso et al., 2015]. AR was used in ECoG [Bundy et al., 2016] [Degenhart et al., 2018] [Farrokhi and Erfanian, 2018] [Gunduz et al., 2016] [Kubánek et al., 2009] [Schalk et al., 2008] [Vansteensel et al., 2016] [Wang et al., 2011] and EEG [Cincotti et al., 2008] [Hettiarachchi et al., 2015] [Iqbal and Aqil, 2016] [Spüler et al., 2012b] [Vidaurre et al., 2006b, 2007] [Wen and Huang, 2017] [Wolpaw and McFarland, 2004] BCI experiments.

BCIs based on phase features are less common than amplitude ones. However, the phase features were exploited in few offline and online BCI studies [Schaeffer and Aksenova, 2018] [Volkova et al., 2019]. Online and offline decoders based on phase features extracted with Hilbert transform were reported in [Brunner et al., 2006] [Gysels and Celka, 2004] [Hamner et al., 2011] [Pourbakhtiar et al., 2013] [Wang et al., 2006] [Wei et al., 2007]. Phase features were directly used as BCI decoder input variables, but other feature generation procedures were proposed. For example, the Phase Locking Value (PLV) was used to investigate the task-induced coupling in long-range synchronization of the neural activity between two electrodes from EEG neural signals [Gysels and Celka, 2004] [Hamner et al., 2011] [Pourbakhtiar et al., 2013] [Wang et al., 2006] [Wei et al., 2007]. Some EEG comparative studies highlighted better performance using phase features than amplitude ones [Sburlea et al., 2017] [Zeng et al., 2019]. In [Hammer et al., 2013], offline ECoG-based BCI results suggested that low frequency phases may be more informative for continuous motor decoding than magnitude-based features. Additionally, amplitude coupled with phase features showed better decoding performance than phase and

amplitude features separately in [Hammer et al., 2013] [Sburlea et al., 2017]. However, phase features were understudied compared to amplitude features and more investigation must be carried out to evaluate the potential of BCI decoding relying on phase or amplitude-phase features.

Commonly, the neural signals recorded from multiple electrodes are considered independently and concatenated to obtain a time-frequency description for each channel creating time-feature-space features. However, some strategies were designed to improve signal-to-noise ratio before computing time and time-frequency features.

**Spatial filters** combine the original recorded sources to create virtual ones with a higher signal-to-noise ratio than that of individual electrodes [Lotte et al., 2018]. Spatial filters can be independent of the recorded signals such as bipolar, Laplacian and surface Laplacian filters. A surface Laplacian filter estimates the radial current of the scalp [Andreu-Perez et al., 2018] using the recorded neural signals in order to enhance local sources contributions and reduce the contribution from distant sources [McFarland, 2015]. Laplacian filters were applied in numerous non-invasive BCI experiments using EEG [Huang et al., 2008] [Kheradpisheh et al., 2014] [López-Larraz et al., 2014] [Mobaien and Boostani, 2016] [Wolpaw and McFarland, 2004] and MEG [Spüler et al., 2012b] recording systems. Data-driven spatial filters were also applied during BCI experiments. Depending on the type of spatial filter, the filter weights were estimated using a supervised or unsupervised learning strategy.

The common spatial pattern (CSP) is a supervised algorithm created to optimize a spatial filter which discriminates two classes. The original multichannel neural signals filtered at the frequency of interest are projected into a subspace which maximizes the variance of the filtered neural signals for one class and minimizes it for the other class [Lotte and Congedo, 2016] [Nicolas-Alonso and Gomez-Gil, 2012]. The Projection (un-mixing) matrix is estimated by solving the simultaneous diagonalization of the covariance matrices of the two classes through Generalized Eigen Value Decomposition. The eigenvectors with the highest and lowest eigenvalues are the optimal projectors [Lotte and Congedo, 2016] [Nicolas-Alonso and Gomez-Gil, 2012]. Numerous variations of CSP were applied to motor BCI systems for binary and multi-state classification during offline and online EEG [Ang et al., 2011] [Bamdadian et al., 2013] [Cantillo-Negrete et al., 2018] [Chowdhury et al., 2017] [Dähne et al., 2014] [Khan et al., 2019] [Kheradpisheh et al., 2014] [D. Li et al., 2018] [Y. Li et al., 2009] [Lotte and Guan, 2011] [Mobaien and Boostani, 2016] [Nguyen et al., 2019] [Nicolas-Alonso et al., 2015] [Oliver et al., 2013] [Peterson et al., 2019] [Qibin Zhao et al., 2008] [Roijendijk et al., 2016] [Rong et al., 2018b] [Sannelli et al., 2016] [Scherer et al., 2015] [Seifzadeh et al., 2017] [Sharghian et al., 2019] [Shenoy et al., 2006] [Shin et al., 2015] [Song and Yoon, 2015] [Tan et al., 2020] [Vidaurre et al., 2011] [Zhao et al., 2008] and ECoG [Jiang et al., 2017] [Kapeller et al., 2015] [Y. Li et al., 2009] [Marathe and Taylor, 2013] [Morales-Flores et al., 2014] experiments. Among the CSP algorithm possible extensions, sparse CSP versions were designed to avoid overfitting troubles [Lotte and Guan, 2011] [Peterson et al., 2019] [Roijendijk et al., 2016]. Filter-bank CSP (FBCSP) algorithms which apply independent

CSP to several frequency bands were reported in [Ang et al., 2011] [Cantillo-Negrete et al., 2018] [Chowdhury et al., 2017] [Jiang et al., 2017] [Khan et al., 2019] [Nicolas-Alonso et al., 2015] [Oliver et al., 2013] [Tan et al., 2020]. Another supervised algorithm named xDAWN filter was designed for ERP classification during EEG experiments [Lotte et al., 2018] [Rivet et al., 2009]. Unsupervised spatial filters were designed for feature extractions and broadly used in BCI applications. The classic unsupervised filters are the principal component analysis (PCA) and the independent component analysis (ICA). PCA computes a linear combination of the neural signal features to create a new orthogonal basis maximizing the variance of the projected features. PCA algorithm was applied to MEA [Kao et al., 2017] [Velliste et al., 2014, 2008] [Wu and Swindlehurst, 2018], ECoG [Flint et al., 2016] [Miller et al., 2009] [Wang et al., 2009] and EEG [Cozza et al., 2020] neural signals. ICA method creates a linear combination of the input features in order to build a statistically independent basis. The estimation of the ICA filter weights commonly relies on the minimization of the mutual information or the maximization of the non-Gaussianity of the neural signals [Clerc et al., 2016a] [Naik, 2011] [Nordhausen and Oja, 2018]. ICA hypothesizes that the recorded neural signals are the linear combination of a finite number of independent sources. ICA aims to reconstruct the signals generated by the sources. As PCA, ICA was applied in numerous EEG [Chen and Fang, 2017] [Moro et al., 2017] [Seifzadeh et al., 2017] and ECoG [Palmer and Hirata, 2018] [Rembado et al., 2016] BCI experiments.

CSP, PCA and ICA algorithms were also used for feature dimension reduction procedure described in the next subsection.

**Multi-way neural signal features.** Since many years, time-frequency analysis highlighted attractive results to decode brain neural signals. Additionally, with the development of more sophisticated recording systems, electrodes/channels numbers highly increased to improve the recording spatial resolution. Therefore, modern BCIs rely generally on features with two (e.g. frequency-space) or three (e.g. time-frequency-space) multi-way array (also referred to as tensor) to decode brain signals [Cong et al., 2015]. The order of a tensor is the number of its dimensions. Vector and matrices of the neural signal features are specific cases of N=1 and N=2 way tensor[Cichocki et al., 2015]. Generally, BCI relied on neural signals described with a N=3 tensor of features (e.g. time-frequency-space features). Two main strategies were designed to process multi-way time-frequency-space features to decode the neural signals. The most usual procedure is named unfolding and considered the feature space as a vector or matrix feature space by concatenating supplementary dimensions. Unfolding procedure has the benefit to reduce the N-way tensor of neural signals features into a well-known domain and allows the application of generic BCI algorithms in clinical and preclinical EEG and ECoG classification and regression experiments [Chao et al., 2010] [Choi et al., 2018] [Cong et al., 2015] [Lotte et al., 2018] [Schaeffer and Aksenova, 2016a] [Shimoda et al., 2012]. However, unfolding strategy exponentially increases the dimension of the feature space and limits

the analysis of the features to standard pair-wise interactions which inevitably creates independent features and loses potentially existing interactions between/among the folded modes, such as time, frequency and space modes [Cichocki et al., 2015] [Cong et al., 2015]. Consequently, several approaches were designed to directly extract neural signal decoding information from high dimensional tensors using tensor factorization procedures or high dimensional tensor projectors to make the best use of the neural signal information. These procedures were reported in EEG and ECoG neural signal decoding studies [Cichocki et al., 2015] [Cong et al., 2015] [Eliseyev et al., 2017] [Eliseyev and Aksenova, 2013, 2014, 2016] [Krauss et al., 2018] [Lotte et al., 2018] [Onishi et al., 2012] [Washizawa et al., 2010] [Yu Zhang et al., 2013] [Zhang et al., 2015, 2017] [Zhao et al., 2013a, 2013b].

**Other features** were used in the context of BCI to decode neural signals [Schaeffer and Aksenova, 2018]. For example, features based on entropy measures or temporal sequence, etc. were evaluated during EEG experiments[Boostani et al., 2007] [Garcia et al., 2003] [Hsu, 2011] [Obermaier et al., 2001] [Thulasidas et al., 2006] [Vidaurre et al., 2009] [Zhang et al., 2008]. Additionally, a decoder using covariance matrices as input features were studied since the last few years. These decoders relied on the computation of symmetric positive-definite matrices and Riemannian geometry [Lotte et al., 2018]. Riemannian geometry using manifold of EEG neural signal covariance matrices highlighted promising results in the last few years.

**End-to-end decoding**. Deep learning is a specific method of the machine-learning field whose popularity is gradually increasing for BCI applications. Deep learning is usually implemented using a neural network architecture composed of several layers. These end-to-end trained decoders do not rely on a fixed feature extraction block to decode the brain neural signals as classic machine-learning algorithms. Instead, the feature extraction step is directly integrated into the model training to automatically extract features useful for decoding rather than hand-engineering them [Volkova et al., 2019]. Popular deep neural networks approaches reported in BCI studies clustered convolutional neural networks, recurrent neural networks, restricted Boltzmann machines [Lotte et al., 2018] [Volkova et al., 2019], etc.

**In conclusion**, motor BCI relying in population recording system commonly used amplitude-based features. Phase or phase-amplitude-based neural decoders highlighted better decoding performance than amplitude-based decoders in several studies with ECoG [Hammer et al., 2013] and EEG neural signal recording systems [Djemal et al., 2016] [Kumar et al., 2018] [Sburlea et al., 2017] [Sun et al., 2017] [Zeng et al., 2019]. Nevertheless, the benefits of phase-related features remain unclear due to the small number of reported offline studies and especially online experiments. In [Krusienski et al., 2012], EEG phase features did not lead to better classification accuracy. Phase features showed lower inter-session performance variability than amplitude-based decoder in [Sburlea et al., 2017]. However, decoders combining both phase and amplitude features showed better

inter-session performance than amplitude and phase decoders. Due to the lack of knowledge and unclear benefits of phase features, decoders based on amplitude features remain more widespread. Deeper investigations on phase features must be carried out.

While extensive studies on BCI using population recording system were conducted, there is no consensus on the best time-frequency features to select for BCI application. The efficiency of time-frequency and time-scale features for neural signal decoding depends on the analyzed datasets [Schaeffer and Aksenova, 2018]. On the one hand, parametric strategies generally outperformed non-parametric methods when the neural signals were well fitted by the selected parametric model. On the other hand, they were irrelevant if the signal was badly fitted [Schaeffer and Aksenova, 2018]. STFT requires finding a trade-off between time and frequency resolution. AR spectral estimation is preferred to Fourier Transform but it performs poorly when the signal is not stationary (which is problematic with non-stationary neural signals) and is also sensitive to the artifact [Kevric and Subasi, 2017] [Nicolas-Alonso and Gomez-Gil, 2012]. Some comparative studies highlighted better performance with parametric models [Herman et al., 2008] whereas it was outpaced by wavelet decomposition in others [Brodu et al., 2011] [Cabrera et al., 2010]. Additionally, the wavelet-based feature extraction procedure outperformed the FFT-based feature extraction procedure in P300, SSVEP and motor imagery EEG experiments [Rosas-Cholula et al., 2010] [Yeh et al., 2013] [Zhang et al., 2010] [Zhao and Wang, 2015].

### 3.2.2. Feature dimension reduction

As previously mentioned, with the multiplication of the studied frequency bands and the number of recorded channels, the feature space may have a high dimension. High dimensional feature space may be problematic for the neural signal decoders which require larger training dataset for the calibration procedure and require managing non-informative or correlated features. The application of dimensional reduction algorithms highlighted several benefits for neural signal processing. Feature space dimension reduction allows avoiding the curse of dimensionality [Bellman, 1961], improving decoding performances, and reducing the required computing time by allocating less computing resources to the feature extraction step (do not compute the irrelevant features). In the case of high-dimensional data flow processing and real-time decoding, all the mentioned aspects are relevant [Haufe et al., 2014]. Dimensionality reduction algorithms are dissociated into the projection methods and the feature selection algorithms.

Projection algorithms aim to project the original feature space into a lower dimensional subspace by linear or non-linear combinations of the initial feature space components. This family clusters the principal and independent component analysis (PCA and ICA), common spatial pattern (CSP) or partial least square (PLS). They were commonly used in BCI applications [Bousseta et al., 2018] [Bundy et al., 2016] [Choi et al., 2018] [Eliseyev et

al., 2017] [Haufe et al., 2014] [Hsu et al., 2016] [Jafarifarmand and Badamchizadeh, 2020] [Jiang et al., 2017] [Khan et al., 2019] [S. P. Kim et al., 2006] [Lotte et al., 2018] [Marathe and Taylor, 2013] [Palmer and Hirata, 2018] [Sannelli et al., 2016] [Schaeffer and Aksenova, 2016b] [Seifzadeh et al., 2017] [Sreenath and Ramana, 2017].

Principal component analysis (PCA) is a statistical unsupervised procedure used for dimensional reduction and feature extraction in the BCI field. PCA projectors are estimated to maximize the variance of the projected data [Bishop, 2006] [Nicolas-Alonso and Gomez-Gil, 2012]. In the vector space, it can be easily demonstrated that the variance is maximum if the projector is set to the eigenvector of the observation variable having the largest eigenvalue. For higher dimensional space, projectors are incrementally added by selecting the eigenvectors with the highest eigenvalues in descending order [Bishop, 2006]. Eigenvalues are representative of the information provided by each eigenvector to describe the dataset distribution. Therefore, the most informative PCA features can be used whereas less informative ones can be discarded to reduce the new feature space dimension with minimal loss of information (generally 5% to 30%). PCA feature projection algorithm was applied for the dimensional reduction of MEA [Kao et al., 2017] [Velliste et al., 2014] [Wu and Swindlehurst, 2018], ECoG [Flint et al., 2016] [Wang et al., 2009] and EEG [Cozza et al., 2020] [Lotte et al., 2018] neural feature space. The application of PCA does not always lead to performance improvements as the discriminative feature may not be relying on the selected principal components, nevertheless, PCA is a decent noise reduction method [Nicolas-Alonso and Gomez-Gil, 2012] [Volkova et al., 2019].

Independent component analysis (ICA) is a statistical unsupervised procedure which identifies a statistically independent basis composed of so-called "source" signals estimated from the combination (commonly linear) of multiple electrodes signals [Nicolas-Alonso and Gomez-Gil, 2012] [Stone, 2002]. The number of created sources is a hyperparameter fixed before the ICA computation. Generally, the number of sources is inferior to the number of original electrodes to perform the feature dimension reduction procedure. ICA was commonly used with low spatial resolution recording system such as EEG-based BCI to extract features from the neural signals in parallel with artifact rejection [Farooq et al., 2019] [Iversen and Makeig, 2019] [Kamousi et al., 2005] [Kim et al., 2019] [Moro et al., 2017] [Nicolas-Alonso and Gomez-Gil, 2012] [Ruan et al., 2017] [Senhadji et al., 2009] [Serby et al., 2005] [Wang et al., 2012] [Wu et al., 2020]. The application of ICA algorithm based on invasive recordings was poorly reported in the BCI field, probably because of the spatial resolution improvements related to this type of recording system. Only few BCI studies based on ECoG neural signal analysis employing ICA were reported [Bouchard et al., 2017] [Estrin et al., 2018] [Palmer and Hirata, 2018] and the benefits of this method were less obvious than its application to EEG neural signal [Hill et al., 2006] [Rembado et al., 2016]. Additionally, in [Wu et al., 2020], ICA algorithms highlighted worse decoding performance compared to other classical spatial filter algorithms such as CSP algorithms during online motor BCI experiments.

Partial least square (PLS) algorithm is the supervised counterpart of PCA. The input neural signals and output variables to predict are projected into a new low dimension subspace which maximizes the covariance between input and output variables projected into this lower subspace [Bro, 1998, 1996]. PLS was widely used for continuous decoding in the case of high dimensional features and small training dataset [Bundy et al., 2016] [Chen et al., 2013] [Choi et al., 2018] [Eliseyev et al., 2012] [Eliseyev and Aksenova, 2016] [Jubien et al., 2019] [Shimoda et al., 2012]. Nevertheless, PLS was also reported as a projection algorithm and was coupled to other classification or regression decoder [Farrokhi and Erfanian, 2018] [Schaeffer and Aksenova, 2016a].

The common spatial pattern (CSP) was commonly used in the BCI field for feature extraction and dimension reduction. As mentioned earlier, the eigenvectors with the highest and lowest eigenvalues are the optimal projectors [Lotte and Congedo, 2016] [Nicolas-Alonso and Gomez-Gil, 2012] and the variance of the neural signals is maximized for one class while minimized for the other class. Commonly only few eigenvectors maximizing the variance of each class are selected to create a new subspace with high discriminability between the two classes. Spatial filtering and dimension reduction procedure are always performed together with CSP algorithm. Numerous variation of CSP were designed such as filter bank CSP [Ang et al., 2011] [Cantillo-Negrete et al., 2018] [Khan et al., 2019] [Nicolas-Alonso et al., 2015] [Oliver et al., 2013] [Tan et al., 2020], common spatial pattern patches (CSPP) [Sannelli et al., 2016] or multiclass versions of the CSP [Jafarifarmand and Badamchizadeh, 2020] [Khan et al., 2019] [Kheradpisheh et al., 2014], penalized time-frequency band CSP (PTFBCSP) [Peterson et al., 2019], regularized CSP [Lotte and Guan, 2011] [Roijendijk et al., 2016]. Few experiments based on ECoG [Jiang et al., 2017] [Kapeller et al., 2015] [Xie et al., 2018] neural signals were reported but commonly, CSP is applied to EEG decoding problems.

Other projector algorithms were designed for BCI application such as Spatio-Spectral Decomposition (SSD) [Dähne et al., 2014], Neighborhood Component Analysis (NCA) [Cozza et al., 2020], or dimensionality reduction algorithms based on Riemannian geometry [Lotte et al., 2018]. Projection methods create a linear combination of the existing neural signal features to create new, more informative, features.

Another solution named feature selection estimates the informativeness of the features to select only a subset of the most relevant characteristics and discard the others. Feature selection family regroups filter-based, wrapper-based and embedded techniques (Figure 3-1) [Bolón-Canedo et al., 2013] [Khaire and Dhanalakshmi, 2019] [Lotte et al., 2018] [Rouhi and Nezamabadi-Pour, 2020]. They were applied in combination or instead of projection algorithms.

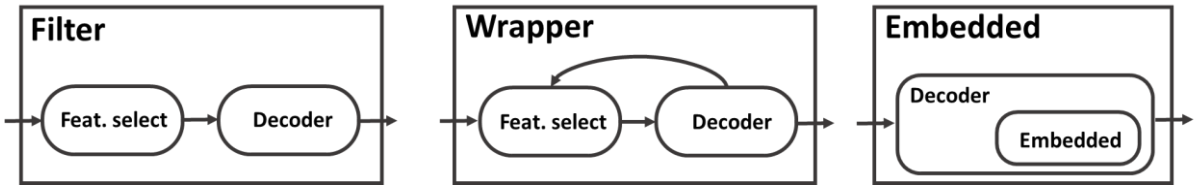



Figure 3-1: Feature selection family schematics. Feature selection family can be divided into filter (left), wrapper (middle) and embedded (right) methods. All of the strategies presents benefits and disadvantages depending on the application. Schematic extracted from [Bolón-Canedo et al., 2013].

Filter-based methods rely on statistical feature evaluation. They rank and select independently the features which cluster the most information without consideration of the trained decoder. Filter based algorithms are effective, have a low computation cost and a good generalization capacity (low risk of overfitting). They are based on simple strategies relying on the inherent characteristic of the features and do not take into account the decoding model (Figure 3-1) [Bolón-Canedo et al., 2013] [Khaire and Dhanalakshmi, 2019] [Rouhi and Nezamabadi-Pour, 2020]. However, these methods do not have any interaction with the decoder which may lead to sub-optimal solutions. They often tend to select highly correlated (redundant) features. [Bolón-Canedo et al., 2013] [Khaire and Dhanalakshmi, 2019] [Lotte et al., 2018] [Rouhi and Nezamabadi-Pour, 2020]. Filter-based methods commonly rely on correlation information or mutual information of the neural signal features. For example, Pearson correlation ranking [Lotte et al., 2018], Correlation-based r2-ranking [Spüler et al., 2012b], Fisher score [Long et al., 2011], Mutual Information-based Best Individual Feature (MIBIF) algorithm [Ang et al., 2011] [Oliver et al., 2013], Mutual Information-based best individual feature (BIF) [Robinson et al., 2013], Representation Entropy (RE) index, Maximal Information Compression Index (MICI) [Delisle-Rodriguez et al., 2017] or Multi-way quadratic programming feature selection (Multi-way QPFS) [Motrenko and Strijov, 2018] were commonly applied in offline BCI studies and online BCI experiments.

In the opposite, wrapper-based techniques rely on supervised learning algorithms to evaluate the possible interactions between the features and the decoder (Figure 3-1). These methods select the features depending on the performance of a decoder with specific feature subsets [Bolón-Canedo et al., 2013] [Khaire and Dhanalakshmi, 2019] [Lotte et al., 2018] [Rouhi and Nezamabadi-Pour, 2020]. Wrapper methods add or remove iteratively new features to the subset of selected feature space and evaluate the performance variation of the new subset combined with the trained decoder [Bolón-Canedo et al., 2013] [Khaire and Dhanalakshmi, 2019] [Lotte et al., 2018] [Rouhi and Nezamabadi-Pour, 2020]. Depending on the performance improvement or decay the added features are kept or removed. These methods are more efficient than Filter-based strategies. Nevertheless, they are costly in terms of computing time (high computational complexity), are

sensitive to overfitting and are decoder-dependent. Particle swarm optimization (PSO) [Cantillo-Negrete et al., 2018] [Kumar et al., 2017] [Remeseiro and Bolon-Canedo, 2019], sequential floating forward selection (SFFS) [Brunner et al., 2006], sequential backward floating selection (SBFS) [Khan et al., 2019], dimensionality reduction mechanism (called DimReM) [Tan et al., 2020], genetic algorithms [Corralejo et al., 2011] [Garrett et al., 2003] [Moro et al., 2017] [Remeseiro and Bolon-Canedo, 2019] [Schroder et al., 2003] and others strategies [Motrenko and Strijov, 2018] were used to optimize the feature selection step in BCI applications.

Embedded techniques regroups the strategies where the feature selection step is directly integrated into the model calibration procedure (Figure 3-1) [Bolón-Canedo et al., 2013] [Khaire and Dhanalakshmi, 2019] [Lotte et al., 2018] [Rouhi and Nezamabadi-Pour, 2020]. Embedded feature selection algorithms combine the benefits of both previously presented methods: keeping the advantages of wrapper strategies while decreasing the computational complexity [Khaire and Dhanalakshmi, 2019]. In the case of an embedded feature selection algorithm, the features selection procedure is an inseparable part of the model learning/training process. Therefore, the feature selection depends on the selected decoder. BCI Embedded techniques group the decision tree, regularization algorithms, etc. They were commonly applied during BCI experiments [Cincotti et al., 2008] [Eliseyev et al., 2012] [Eliseyev and Aksenova, 2016] [Flamary and Rakotomamonjy, 2012] [Foodeh et al., 2020] [Kim et al., 2018] [López-Larraz et al., 2014] [Lotte and Guan, 2011] [Mishra et al., 2018] [Nagel and Spüler, 2019] [Nakanishi et al., 2017] [Peterson et al., 2019] [Seifzadeh et al., 2017] [Sheikhattar et al., 2015] [Sreeja et al., 2019] [Wen et al., 2016] [Y. Zhang et al., 2013] and in other fields [Hervás et al., 2019] [Kalivas, 2012] [Khaire and Dhanalakshmi, 2019] [Muñoz-Romero et al., 2015].

**In summary**, Dimensional reduction algorithms highlighted benefits for neural signal decoding. They can remove correlated features, improve the signal-to-noise ratio, limit the risks of overfitting, speed up the decoder calibration and/or the neural signal decoding, reduce the computational loading, etc. Many strategies were tested to select the best features [Lotte et al., 2018]. However, there is no consensus on the best method to be followed as it is influenced by the analyzed signals properties (highly correlated features, low or high signal to noise ratio, signal stationarity across time, etc.), the problem to solve (binary or multi-class classification problem), the length of the training dataset, the restriction related to computational complexity (offline or online applications), etc. Projector methods were commonly applied for dimensional reduction of the neural signal feature dimension. Nevertheless, such procedures lead to less interpretable features, are computationally expensive and may be not optimal in the case of non-stationary signals. Feature selection algorithms find the best subset of features among all the computed ones. These methods can be dependent or not from the decoder training and present various advantages. Filter based-methods require less computing power than embedded methods which itself is less consuming than wrapper techniques.

Nevertheless, generally, wrapper and embedded methods outperformed filter-based strategies [Rouhi and Nezamabadi-Pour, 2020].

## 3.3. Effector control features

From the extracted neural signal features, the decoder evaluates a discrete or continuous output that will be converted into commands to control the effector.

Discrete output variables are commonly estimated to evaluate the discrete mental state of the subject. In motor BCI applications, the discrete states/classes were used to control various movement states such as the opening/closure of the hand [Cantillo-Negrete et al., 2018], the waking/gait cycle activation [López-Larraz et al., 2016], the movements of one specific finger [Schaeffer and Aksenova, 2016a] or limb [Choi et al., 2018], etc. Additionally, idle state classification was often integrated into asynchronous BCI systems [Kao et al., 2017] [Schaeffer and Aksenova, 2016a]. Discrete output variables were also reported to control the continuous movements of an effector. The continuous movements were discretized into a finite number of directions which are selected using a classifier [Huang et al., 2009] [Leeb et al., 2007] [Trejo et al., 2006] [Vidaurre et al., 2016].

Continuous dependent variables were commonly the end-point kinematic variables of the controlled effector such as the position, the velocity, the speed, the acceleration or several of these variables [Bundy et al., 2016] [Hammer et al., 2013] [Z. Li et al., 2009]. However, other less conventional output variables [Marathe and Taylor, 2011] such as joint velocity [Young et al., 2019], angle/orientation [Wodlinger et al., 2015], force [Carmena et al., 2003], muscle activation [Nakanishi et al., 2017] were reported.

Finally, PCA or other dimensionality reduction algoritms were applied to reduce the dependent variable dimension, "decorrelate" [Acharya et al., 2010] [Hotson et al., 2014] the output variables and represent the output variable space as a linear combination of the initial dependent variables. In this case, the model decoded the PCA coordinates from the neural signals before evaluating the original coordinates through inverse PCA transformation [Volkova et al., 2019]. Dependent variable dimension reduction via the PCA algorithm was reported in MEA preclinical [Mollazadeh et al., 2014] [Schaffelhofer et al., 2015] and ECoG clinical [Acharya et al., 2010] [Hotson et al., 2014] [Vinjamuri et al., 2011] experiments.

There is no consensus on the best continuous output variable to decode. Nevertheless, position and velocity are the most widespread output variables in the BCI field. In [Marathe and Taylor, 2011], the position-based decoder underperformed compared to the velocity-based decoder particularly in the case of large and numerous decoding errors.

## 3.4. Decoder/Model identification

After extracting the relevant information from the recorded neural signals, it is required to evaluate a function which transforms the input feature variables into discrete or continuous output variables. In the most general case, the assumption is made that an unknown linear or non-linear function is mapping the input variable space into a continuous or discrete variable space.

Generally, in the BCI field, the input and output variables at time $t$ are vector or matrix variables [Cong et al., 2015]. Let $\mathbf{x}_t \in \mathbb{R}^{I_1}$, $\mathbf{y}_t \in \mathbb{R}^{J_1}$ and $z_t \in \mathbb{N}$ denote the input neural signals variable, the continuous output variable and the discrete output variable respectively where $I_1$ is the input neural signal feature dimension and $I_2$ is the continuous dependent variable dimension. It is assumed that a linear or non-linear function $h$ exists such as $\mathbf{y}_t = h(\mathbf{x}_t) + \boldsymbol{\varepsilon}_t$ or $\mathrm{z}_t = h(\mathbf{x}_t) + \varepsilon_t$ in the cases of continuous and discrete variables respectively where $\boldsymbol{\varepsilon}_t$ and $\varepsilon_t$ are "noise" random variables generally, supposed to be independent and identically distributed.

The model identification, calibration or training consists in finding an estimation $\hat{h}$ of the unknown function $h$ which minimizes the differences between the output variables $\mathbf{y}_t$ or $\mathrm{z}_t$ and an estimated output variable $\hat{\mathbf{y}}_t = \hat{h}(\mathbf{x}_t)$ or $\hat{\mathrm{z}}_t = \hat{h}(\mathbf{x}_t)$ following specific criteria (e.g. minimum least square, maximum likelihood, etc.) and using machine-learning methods [Schaeffer and Aksenova, 2018]. Numerous models were tested to control various effectors and perform various tasks. Model parameter weights may be estimated using supervised, unsupervised and reinforcement learning strategies. This review is particularly focused on the supervised decoding of the neural signals in the case of BCI applications.

To translate clinical BCI experiments into daily life BCI applications, BCI system must be applied in real-time. Nevertheless, most BCI studies were carried out offline using a database with a finite number of samples. Translation from offline studies to real-time experiments is not trivial due to the computational complexity of the decoding algorithms used to estimate the output variables from input neural signal features. Numerous models applied in offline studies are not suitable for real-time decoding and data-flow processing. As an example, in [Cunningham et al., 2009], Cunningham argued that the computation time for spike rate decoding of a one-second spike train for the simplest methods was performed on a millisecond time scale whereas more complex strategies were limited to seconds or even minutes. In the case of real-time BCI application, algorithms with a computation time longer than the analyzed neural signal window can not be integrated into the online BCI system. Therefore, the computational complexity and the computation time for neural signal decoding are key characteristics to consider. Due to the temporal limitations of the complex algorithms and the optimization procedures, simple linear models are generally promoted for online neural signal decoding [Murphy et al., 2016].

Additionally, for online closed-loop BCI applications, several studies highlighted parameter weights and decoding performance differences between models trained using open-loop and closed-loop calibration procedures [Jarosiewicz et al., 2013] [Orsborn et al., 2014]. Closed-loop decoder adaptation (CLDA) can be achieved using an offline or online calibration procedure with different update frequencies (e.g. after each sample, second, trial, session, day, etc.). However, several benefits of online incremental calibration procedures were stressed in [Brandman et al., 2018] such as shorter calibration sessions, rapid feedback to the patient, etc. For example, a cross-validation procedure for model selection or hyperparameter optimization is a widespread calibration method applied in the BCI field which is computationally too heavy to be implemented for real-time model calibration. Therefore, the computational burden and frequency rate of the calibration procedure are important characteristics to consider.

The following section firstly introduces the most common strategies practiced for offline model calibration and their application in offline studies or online BCI experiments. Finally, a specific focus is granted to CLDA and incremental adaptive model calibration during real-time BCI experiments.

## 3.4.1. Offline decoder calibration

### 3.4.1.1. Discrete output variables decoding: Classifiers

A classification problem makes the assumption that observed neural signal $\mathbf{x}_t \in \mathbb{R}^{I_1}$ can be clustered into a finite number of class/label states $z_t \in \mathbb{N}$. These states are related to specific mental or motor imagery tasks which create specific neural signal patterns identified by the classifier. Offline classification experiments are the most common studies in the motor-BCI fields and numerous decoders were designed to enhance classification performance.

**Linear Discriminant Analysis (LDA)** decoder is a classic linear classifier for binary and multi-class problems. LDA is based on a multivariate Gaussian distribution estimation for each state with the assumption of equal covariance for each class. In the case of binary classification, LDA is looking for the hyperplane which maximizes the distance between the two projected classes while minimizing the interclass variance [Lotte et al., 2007]. LDA is one of the most popular types of classifiers in the BCI field [Lotte et al., 2018] and has been used in numerous experiments. LDA classifiers were employed for offline EEG neural signal analysis and performance comparison to other algorithms [Cozza et al., 2020] [Kim et al., 2018] [López-Larraz et al., 2014] [Lotte and Guan, 2011] [Scherer et al., 2015] [Seifzadeh et al., 2017] [Shin et al., 2015]. During the offline studies, stepwise LDA (SWLDA) and LDA were applied for preparation versus execution tasks and cursor direction classification from ECoG neural signals [Gunduz et al., 2016]. Additionally, the hand flexion and extension were classified during offline ECoG [Jiang et al., 2017]and EEG [Cantillo-Negrete et al., 2018] studies.

LDA classifiers were embedded in several online BCI systems based on both invasive and non-invasive recording systems. A BCI communication system controlled by an amyotrophic lateral sclerosis (ALS) patient and a locked-in syndrome patient was reported using LFP signals (MEA) and a LDA classifier [Milekovic et al., 2018]. Additionally, the real-time click detection of another BCI communication system was controlled by two tetraplegic and two ALS patients using SUA/MUA signals in [Jarosiewicz et al., 2015]. ECoG-based real-time decoding was performed to detect idle versus active state from epileptic patients using a LDA decoder [Kapeller et al., 2015] whereas a hierarchical LDA classifier was integrated into online asynchronous BCI experiments to detect the individual finger and idle states [Hotson et al., 2016a].

A tetraplegic patient [Pfurtscheller et al., 2000] and healthy subjects [Cantillo-Negrete et al., 2018] controlled the opening and closure of a hand orthosis during online experiments based on EEG recordings. Moreover, the lower limb exoskeleton control based on EEG recording and a LDA decoder was reported by a healthy subject during online BCI experiments [López-Larraz et al., 2016] and by eight paraplegic patients during a long-term neurorehabilitation study [Donati et al., 2016]. A LDA decoder was selected in [Khan et al., 2018] for real-time detection of the walking gait cycle from a fNIRS recording system because LDA provided a good trade-off between the time of execution and classification accuracy. Finally, online three-states classification experiments were reported based on the EEG neural signal decoding of 6 healthy subjects using LDA classifier [Brunner et al., 2006] [Chae et al., 2012] and 15 healthy subject using the shrinkage LDA algorithm [Schwarz et al., 2020].

**Quadratic Discriminant Analysis (QDA)** is a non-linear variant of LDA in which an individual covariance matrix is estimated for every class of observations. QDA is particularly useful if there is the prior knowledge that individual classes exhibit distinct covariance matrices. Due to higher computational complexity without obvious performance improvements, its simpler version, the LDA was often preferred for BCI application. Nevertheless, QDA was regularly used in offline studies for classifier performance comparison. QDA was compared to other algorithms based on ECoG [Jubien et al., 2019], EEG [Bhattacharyya et al., 2017b] [Eva and Lazar, 2019] [Faradji et al., 2009] [Javed et al., 2017] [Spinnato et al., 2015] [Vidaurre et al., 2006a] and fNIRS [Khan et al., 2018] [Naseer et al., 2016b] multi-class or binary classification problem.

**Support Vector Machine (SVM)** is one of the most common state of the art classifier applied in the BCI field. Linear or non-linear SVMs project data in a higher dimensional space where the classes are linearly separable by a hyperplane (kernel trick) built on the nearest training samples named support-vectors. The non-linearity is based on various kernel functions [Bishop, 2006]. In numerous BCI experiments, SVM outperformed other algorithms and highlighted robust classification with better generalization ability [Lotte et al., 2007]. Therefore, SVM was frequently applied to compare algorithm performance during offline studies using ECoG [Jubien et al., 2019], EEG [Cozza et al., 2020] [Eva and

Lazar, 2019] [Faradji et al., 2009] [Hettiarachchi et al., 2015] [Khan et al., 2019] [Mishra et al., 2018] [Schlögl et al., 2005] [Shin et al., 2015] [Tan et al., 2020] and fNIRS [Khan et al., 2018] [Naseer et al., 2016b] recording systems. Additionally, online experiments to control real and virtual effectors [AL-Quraishi et al., 2018] were performed using SVM algorithm for open/closure hand movement in MEG [Fukuma et al., 2016] as well as for P300 speller control [Thulasidas et al., 2006] [Woehrle et al., 2015] and online 4 state classification for 2D cursor control [Huang et al., 2009] or robot arm / prosthetic control [Hortal et al., 2015] [Yanagisawa et al., 2012]. Multiple SVM classifiers were trained in [Bhattacharyya et al., 2017a] to perform online 3D sequential control of a hardwired Jaco robot arm.

**Artificial Neural Networks (ANN)** classifier family regroups all the ANN possible architecture applied to BCI [Lotte et al., 2007]. Multilayer perceptron (MLP) is probably the most popular ANN for BCI classification. ANN are non-linear classifiers which apply successive weighted linear combination and non-linear functions to the input neural signal. ANN presents high flexibility to solve non-linear neural signal problems. ANN was generally applied to offline BCI analysis and algorithms comparison experiments in fNIRS, EEG and ECoG experiments [AL-Quraishi et al., 2018] [Faradji et al., 2009] [Javed et al., 2017] [Jubien et al., 2019] [Lotte et al., 2007] [Naseer et al., 2016b] [Nicolas-Alonso and Gomez-Gil, 2012] [Sakhavi et al., 2018] [Schaeffer and Aksenova, 2018] [Yang et al., 2015]. Due to the high computational burden of ANN algorithms, only few online BCI experiments were reported using ANN. A tetraplegic patient controlled in real-time a four-class functional electrical stimulation (FES) system to perform four hand shapes. The employed deep neural network was made up of Long Short Term Memory (LSTM), convolutional and fully connected layers [Schwemmer et al., 2018]. Online recognition of two mental states was performed using a fNIRS recording system and Deep ANN during driving simulation experiments [Huve et al., 2019]. Nowadays, ANN is not as prevalent in the real-time BCI application as in other domain due to the lack of interpretability of the ANN models (e.g. the ANN model parameters cannot be related to fundamental neurophysiological insights) [Volkova et al., 2019] and the high computational resources required to performed neural signal decoding.

**K-nearest neighbor (kNN)** is a non-parametric non-linear classifier which determines the unseen samples label depending on the label of its k-nearest neighbors evaluated within the training dataset. The nearest neighbors are found according to a distance metric (e.g. Euclidian distance). The unseen sample is clustered in the same class as its closest neighbors according to the distance metrics [Khan et al., 2018] [Lotte et al., 2007]. kNN algorithm was often tested during offline comparative studies of multiple algorithms as it is an easy non-linear decoder to implement. kNN experiments were predominantly reported for EEG recordings analysis. Binary classification performance comparison between kNN and other algorithms were reported based on EEG dataset [Cozza et al., 2020] [Hettiarachchi et al., 2015] [Khan et al., 2019] [Tan et al., 2020] as well as 3-state [Eva and Lazar, 2019] and 4-state [Schlögl et al., 2005] classification performance

comparison. Additionally, online control of a lower limb exoskeleton was reported from EEG neural signal decoding using a kNN classifier [Kwak et al., 2015]. The offline upper limb movement classification (3 classes) of two epileptic subjects was reported based on ECoG recordings using kNN classifier [Chin et al., 2007]. Finally, kNN was tested during offline binary classification comparative studies using a fNIRS recording system [Khan et al., 2018] [Naseer et al., 2016b]. This algorithm is efficient with low dimensional feature vectors. However, Euclidian distance-based kNN algorithms are highly sensitive to high dimensional feature space [Nicolas-Alonso and Gomez-Gil, 2012]. Due to its heavy computational resource requirements in the case of high dimensional data and its non-significant enhancement of the performance compared to other simpler methods, kNN was scarcely used for online BCI experiments.

**Other discrete algorithms** were applied in BCI experiments. Bayesian classifiers were reported during offline EEG and fNIRS studies [Khan et al., 2019, 2018] [Naseer et al., 2016b] as well as during online EEG experiments using LDA and Bayes rules [He et al., 2018] [King et al., 2014]. A logistic regression (LR) classifier was tested during EEG [Javed et al., 2017] and ECoG [Jubien et al., 2019] offline comparative studies as well as during EEG online comparative study [Bhattacharyya et al., 2017b], online binary classification experiments [Lehtonen et al., 2008] and online walking detection rehabilitation experiments [García-Cossio et al., 2015]. Additionally, supervised Gaussian Mixture Models (GMM) were integrated into a BCI system for real-time lower-body effector control from EEG neural signals [Kilicarslan et al., 2013]. Riemannian geometry-based classifiers were recently developed in the BCI field [Lotte et al., 2018] [Yger et al., 2017]. Few EEG-based offline [Mishra et al., 2018] [Roijendijk et al., 2016] and online [Kalunga et al., 2016] experiments highlighted the benefits of these new classifiers. Finally, random forest (RF) [AL-Quraishi et al., 2018] and Mahalanobis distance-based classifiers [Eva and Lazar, 2019] [Faradji et al., 2009] [Huang et al., 2009] [Lotte et al., 2007] [Schlögl et al., 2005] were applied to EEG BCI experiments to detect discrete mental states.

Previously presented decoders are static decoders. These decoders assume that the observed variables are independent in time. The possible temporal dependencies between successive observed variables are not considered. However, generally, this assumption is violated in BCI and particularly in motor BCI applications [Schaeffer and Aksenova, 2018]. Therefore, several BCI studies were led to investigate the classification enhancement related to the integration of temporal information into the BCI transducer.

One explored solution was to consider the temporal dependencies in the feature extraction or post-processing steps. Several BCI experiments were reported with neural features estimated using temporal/recurrent sliding windows [Dietterich, 2002] [Eliseyev et al., 2017] [Flamary and Rakotomamonjy, 2012] whereas the decoder output variables were post-processed using a moving average filter [Hotson et al., 2016a]. In [Fifer et al., 2014], LDA was coupled to a manually adjusted transition probability matrix to control in real-time the reaching and grasping movement (binary classification) of a robotic

prosthetic arm from EEG neural signals. Numerous models were designed to integrate directly the temporal dependencies into parameters estimation.

**Hidden Markov Model (HMM)** is the most common dynamic decoders in the BCI field. A HMM is a stochastic generative model that deals with observable and latent variables. It assumes that the observable variables at time $t$ (e.g. neural signals) are generated by hidden latent variables which follow a N-Markov chain model. At time $t$, the latent variable $z_t$ is dependent on the previous latent variables $z_{t-1}$, $z_{t-2}$ ... $z_{t-N}$. Generally, in the BCI field, latent variables are modeled with a N=1 Markov chain and are associated with a specific neural state (e.g. left arm activation, right wrist activation etc). HMM evaluates the dynamic transition of the states.

HMM classifiers were reported in offline [Antelis et al., 2017] [Dobiáš and Štastný, 2016] and online [Obermaier et al., 2001] EEG-based BCI experiments. Moreover, Markov models were integrated into more complex classifiers and tested in an offline study [Williams et al., 2018] and online experiment [Lisi et al., 2018] using EEG neural signal recordings. Dynamic models were also reported with invasive recording systems. ECoG neural signal classification with Markovian process or HMM were tested offline [Onaran et al., 2011b] [Pfeiffer et al., 2018] [Wang et al., 2011] [Wissel et al., 2013] and online [Hotson et al., 2016a] [Moses et al., 2018] clinical experiments whereas preclinical online [Kao et al., 2017] and offline [Darmanjian et al., 2003] MEA-based BCI studies applied HMM to detect idle versus active movement state activation.

Variants of HMM such as Hierarchical HMM (HHMM) [Saa and Çetin, 2013] [Suk and Lee, 2010], Input-Output HMM (IOHMM) [Chiappa, 2006] or Kernel-HMM [Xu et al., 2005] were tested offline using BCI EEG datasets. The HHMM generalized the HMM to a structured multi-level stochastic process where each hidden state of the HMM is composed of sub-states which are themselves modeled by a HMM and so on [Fine et al., 1998]. IOHMM is a discriminative decoder which models both the hidden state and observation variable succession. The Kernel-HMM combines the HMM and the maximum margin principle projected into a kernel space of a support vector machine (SVM) to enhance the classification performance [Xu et al., 2005]. Input-Output HMM (IOHMM) [Chiappa, 2006] and Kernel-HMM [Xu et al., 2005] algorithms outperformed HMM in EEG-based BCI classification offline studies. The hidden semi-Markov model (HSMM), another extension of HMM, was used for offline unsupervised fMRI mapping [Faisan et al., 2005] and EEG mental state classification [Oliver et al., 2012].

**Conditional random fields (CRF)** algorithm was reported in offline motor BCI experiments based on EEG [Delgado Saa and Çetin, 2011] [Hasan and Gan, 2011a, 2011b] [Saa and Çetin, 2013, 2012] and clinical ECoG [Saa et al., 2016] recordings. CRF is a discriminative model where the interaction of the latent variables with each other is determined by several past latent variables ($z_{t-1}$, ... $z_{t-\Delta t}$) and the sequence of observed variables ($x_{t-1}$, ... $x_{t-\Delta t}$). The HMM can be seen as the generative version of a linear-chain

CRF with a particular choice of feature function [Dietterich, 2002] [Sutton and McCallum, 2010]. Variants of CRF such as latent dynamic CRF [Saa and Çetin, 2013] or Hidden CRF [Delgado Saa and Çetin, 2011] [Saa and Çetin, 2012] were tested in EEG offline studies.

**Neural Networks (ANN)** taking into account temporal modeling of the observed variables were tested in offline BCI classification studies relying on EEG [Bashashati et al., 2017] [Bashashati and Ward, 2017] [J.-M. Cano-Izquierdo et al., 2012] [Haselsteiner and Pfurtscheller, 2000] or ECoG [Du et al., 2018] [Xie et al., 2018] neural signals. Additionally, online BCI experiments using EEG [Millan et al., 2004] and MEA [Schwemmer et al., 2018] recording system were reported. In the cited BCI experiments, various dynamic ANN algorithm such as Long-Short Term Memory (LSTM) networks [Schwemmer et al., 2018] [Xie et al., 2018], Neural Networks CRF [Bashashati et al., 2017] [Bashashati and Ward, 2017], time-dependent multi-layer perceptron algorithms [Haselsteiner and Pfurtscheller, 2000] [Millan et al., 2004], etc. were tested.

Whereas CRF and NN outperformed HMM in numerous articles [Bashashati et al., 2017] [Bashashati and Ward, 2017] [J.-M. Cano-Izquierdo et al., 2012] [Chiappa, 2006] [Delgado Saa and Çetin, 2011] [Hasan and Gan, 2011b] [Saa and Çetin, 2012] [Sakhavi et al., 2018] [Xu et al., 2005] [Yang et al., 2015], the decoders based on these two strategies are computationally expensive [Dietterich, 2002] [Sutton and McCallum, 2010] and may not be adapted to specific applications such as online decoding. HMM remains the most common dynamic model even though dynamic modeling remains underused for BCI state classification [Lotte et al., 2007].

**In summary**, various algorithms were applied to BCI mental state classification problems and few comparative studies intended to evaluate the best strategy to select [Boostani et al., 2007] [Cincotti et al., 2003] [Kanoga et al., 2018] [Lotte et al., 2007] [Oganesyan et al., 2018] [Saa et al., 2016] [Wissel et al., 2013]. No conclusion on the best classifier can be drawn from the comparative studies as the reported classifier performances seem to be highly dependent on, the studied dataset, the preprocessing, the feature extraction procedure, etc.

Among the static decoders, SVM, LDA and NN-based classifiers are likely the decoders with the most stable and robust performance across the studies [Boostani et al., 2007] [Kanoga et al., 2018] [Lotte et al., 2007] [Oganesyan et al., 2018]. Other algorithms such as kNN are not recommended in the case of high-dimensional space which is commonly the case in the neural signal processing field [Lotte et al., 2007]. Only few studies compared the performance of static versus dynamic decoders. With the exception of [Cincotti et al., 2003], dynamic decoders outperformed or at least performed as well as static decoders [Lotte et al., 2007] [Saa et al., 2016] [Wissel et al., 2013]. The discrete decoding experiments reviewed in [Lotte et al., 2007] highlighted that the HMM provided better performance than other algorithms during synchronous BCI experiments but performed similarly than static decoders in asynchronous BCI

experiments. However, asynchronous studies are lacking published experiments and results to bring out conclusions on the best classifier to employ.

#### 3.4.1.2. Continuous output variable decoding

Regression is a statistical approach often used in BCI and particularly motor BCI field which decodes from the observed neural signal $\mathbf{x}_t \in \mathbb{R}^{I_1}$ a continuous output variable $\mathbf{y}_t \in \mathbb{R}^{J_1}$. In the motor BCI field, the continuous dependent variables used to achieve end-point effector control (cursor, prosthetic hand, etc.) were often kinematic variables such as position, velocity, acceleration or speed. With the exception of some EEG experiments [Edelman et al., 2019] [McFarland et al., 2010] [Meng et al., 2016] [Waldert et al., 2009], continuous decoding was mostly performed with invasive recording systems (MEA and ECoG) [Schaeffer and Aksenova, 2018]. Numerous algorithms were designed to estimate linear or non-linear decoding models. A brief review of the most common strategies is described in this section.

**Population Vector Algorithm (PVA)** is a MEA-specific algorithm relying on the cosine tuning of the motor cortex neurons [Georgopoulos et al., 1986]. PVA estimates the cosine tuning of individual neuronal responses characterized by a single preferred direction in which the unit fires maximally. The vector direction is defined as the sum of the preferred directions of the recorded neuron population, weighted by the instantaneous firing rates of each cell [Chase et al., 2009] [Schwartz et al., 2001] [Velliste et al., 2008]. PVA assumes a uniform distribution of preferred directions. This algorithm was used for closed-loop BCI decoding of a prosthetic arm during monkey self-feeding experiments [Velliste et al., 2008] and center-out experiments [Koyama et al., 2010] [Taylor et al., 2002] [Wahnoun et al., 2006].

**Optimal linear estimation (OLE)** is a variant of PVA, without the assumption that the preferred directions are uniformly distributed, in order to avoid the bias introduced by PVA when the uniform distribution assumption is violated [Chase et al., 2009]. To this date, online BCI experiments which reported the highest number of controlled DoF integrated indirect OLE algorithm for SUA/MUA neural signal decoding. Wodlinger and Collinger highlighted respectively 10 DoF and 7 DoF real-time control of a robotic arm by a tetraplegic patient using MEA recordings. A BCI system using implanted FES, MEA neural signals, a OLE algorithm and a mobile arm support was designed to provide to a tetraplegic patient partially restored reaching and grasping movements during real-time experiments [Ajiboye et al., 2017]. Additionally, during closed-loop experiments, two tetraplegic patients implanted with MEA controlled the 3D Cartesian and joint velocity of a virtual arm effector [Young et al., 2019]. Finally, Based on indirect OLE algorithm [Salinas and Abbott, 1994] [Wang et al., 2007] adapted to the ECoG neural signal processing, Wang and Degenhart highlighted real-time 3D cursor control by a tetraplegic, an ALS and a brachial plexus injured patients using motor imagery strategy [Degenhart et al., 2018] [W. Wang et al., 2013].

In the BCI field, linear regression models were frequently trained using **Ordinary Least Squares (OLS)** to estimate the maximum likelihood with the assumption of Gaussian noise. Nevertheless, OLS is highly unstable in high-dimensional feature space or in the case of correlated features [Schaeffer and Aksenova, 2018] [Shanechi et al., 2013]. To overcome this issue, regularized/penalized linear regression were designed among which the pace regression [Kubánek et al., 2009] [Wang et al., 2010], the sparse regression [Nakanishi et al., 2017, 2013] [Rouse et al., 2016] [Williams et al., 2013] and the ridge regression [Kim et al., 2015] [Seifzadeh et al., 2017] [Shanechi et al., 2013] [Suminski et al., 2013, 2010] [Willett et al., 2013]. These regularized OLS algorithms were exploited in all the BCI field with EEG [Kim et al., 2015] [Seifzadeh et al., 2017], ECoG [Nakanishi et al., 2017, 2013] [Rouse et al., 2016] [Williams et al., 2013] and MEA [Shanechi et al., 2013] [Suminski et al., 2013, 2010] [Willett et al., 2013] neural signal decoding, for offline [Kim et al., 2015] [Nakanishi et al., 2017, 2013] [Seifzadeh et al., 2017] [Suminski et al., 2010] and online [Rouse et al., 2016] [Shanechi et al., 2013] [Suminski et al., 2013] [Willett et al., 2013] [Williams et al., 2013] clinical and preclinical experiments.

**Partial Least Square (PLS) regression** family algorithms were widely used in the BCI field due to their robustness to high dimensional feature space and multi-collinearity problems. Additionally, PLS algorithms provide reliable model estimation in the case of small training dataset [Cramer, 1993] [Geladi and Kowalski, 1986]. Several articles reported offline 3D movements decoding from ECoG neural signals during preclinical experiments in [Chao et al., 2010] [Chen et al., 2013] [Eliseyev and Aksenova, 2016] [Farrokhi and Erfanian, 2018] [Shimoda et al., 2012]. Alternative algorithms based on the PLS such as Multi-way PLS (NPLS), Sobolev NPLS (SNPLS), Polynomial Penalized NPLS (PNPLS) [Eliseyev and Aksenova, 2016], Higher Order Partial Least Squares (HOPLS2) [Zhao et al., 2013a], kernel tensor partial least squares (KTPLS) [Zhao et al., 2013b] and generalized additive model PLS (GAM-PLS) [Farrokhi and Erfanian, 2018] were designed and tested in offline 3D arm trajectory reconstruction preclinical BCI studies.

While the majority of the BCI algorithms assumed a linear relationship between continuous output variables and neural signal features, this assumption may be limited for accurate decoding. Therefore, several non-linear algorithms were evaluated [Schaeffer and Aksenova, 2018]. The application of non-linear models such as Wiener, cascade Wiener [Flint et al., 2016] [Schaeffer and Aksenova, 2016a] [Suminski et al., 2013], Support Vector Machine Regression (SVR) [K. H. Kim et al., 2006] [Mehring et al., 2003], Piecewise probabilistic decoding (PPD) [Farrokhi and Erfanian, 2018], Piecewise Linear Model (PLM) [Willett et al., 2018], neural networks [Pandarinath et al., 2018] [Schaeffer and Aksenova, 2018] [Schwemmer et al., 2018], Bayesian maximum-likelihood estimation [Ludwig et al., 2011], Gaussian process regression (GPR) [Wang et al., 2010] [Yanagisawa et al., 2012] were reported for offline ECoG [Farrokhi and Erfanian, 2018] [Flint et al., 2016] [Schaeffer and Aksenova, 2016a] [Wang et al., 2010] and online ECoG [Yanagisawa et al., 2012] or MEA neural signal decoding [Pandarinath et al., 2018] [Schwemmer et al., 2018] [Suminski

et al., 2013] [Willett et al., 2018]. The Additive Models (AMs), Generalized Linear Models (GLMs) and Generalized Additive Model (GAM) encompass the strategies dedicated to change a linear model into a non-linear one using specific well-known non-linear link functions: (e.g. Gaussian, binomial distribution, etc.) [Yun Gao et al., 2003]. The generalized models were tested in offline preclinical epidural and subdural ECoG studies on monkeys [Eliseyev and Aksenova, 2014] [Engel et al., 2017] [Farrokhi and Erfanian, 2018].

Commonly, non-linear models outperformed linear ones. The Non-linear SVM and MLP provided better decoding than linear algorithms in [K. H. Kim et al., 2006] whereas the Gaussian process models outperformed the pace regression algorithm in [Wang et al., 2010]. Additionally, PLS linear algorithms highlighted lower performance than PPD [Farrokhi and Erfanian, 2018], kernel PLS [Engel et al., 2017] and AM, GLM, GAM-PLS [Eliseyev and Aksenova, 2014] algorithms. The multiplicative recurrent neural network (MRNN) outperformed the state of the art closed-loop online continuous decoding algorithms REFIT-Kalman Filter during preclinical experiments [Sussillo et al., 2016].

The non-linear models may be more qualified to estimate the complex relationship between the neural signals and the decoded kinematic signals [Schaeffer and Aksenova, 2018] [Wang et al., 2010]. However, the non-linear model superiority does not make consensus. The PPD outperformed the PLS but the GAM-PLS highlighted poor decoding performance compared to PLS algorithms in [Farrokhi and Erfanian, 2018]. Additionally, the Kernel-PLS in [Engel et al., 2017] was slightly above PLS accuracy but did not show a significant improvement.

The non-linear decoders have several drawbacks to consider before their integration into real-life BCI applications. Firstly, they are more complex and sensitive to high-dimensional space and a small training dataset which may lead to overfitting issues [Schaeffer and Aksenova, 2018] [Wang et al., 2010]. Secondly, the non-linear model optimization algorithms are time-consuming and require generally more computational resources than linear models [Schaeffer and Aksenova, 2018] [Wang et al., 2010] which is a constrain to their use in BCI applications [Schaeffer and Aksenova, 2018] [Wang et al., 2010]. Linear models highlighted up to 10D and 3D control using MEA and ECoG recording systems respectively [Degenhart et al., 2018] [Wodlinger et al., 2015]. To introduce non-linear algorithms into online BCI applications, they need to stress far superior decoding performance than simple linear models to counter the significant disadvantages related to their complex non-linear model calibration procedure.

Previously presented regression algorithms are static decoders which assume that continuous output variable $\mathbf{y}_t \in \mathbb{R}^{J_1}$ at time t, is only related to the instantaneous neural signal activity $\mathbf{x}_t \in \mathbb{R}^{I_1}$. However, in the BCI field, this assumption is often violated and dependence of $\mathbf{x}_t$ with previous input $\mathbf{x}_{t-1}$ and output variables $\mathbf{y}_{t-1}$ is expected. Dynamic models consider the previously predicted variables to evaluate a more reliable

estimation of the desired variable. Like the HMM for the discrete models, dynamic models were applied during online BCI experiments and offline studies.

**Kalman filter (KF)** is the most widespread dynamic decoder in the BCI field. KF is a linear stochastic state-space model with a recursive Bayesian estimation procedure evaluating an unknown kinematic variable $\mathbf{y}_t$ as it evolves over time. Given noisy neural signal observations $\mathbf{x}_t$ and the previously observed cursor variable $\mathbf{y}_{t-1}$ [Dangi, 2015] [Schaeffer, 2017] following the equations:

$$\mathbf{y}_t = \mathbf{A}\mathbf{y}_{t-1} + \boldsymbol{w}_t, \qquad \boldsymbol{w}_t \sim \mathcal{N}(0, \mathbf{W})$$

$$\mathbf{x}_t = \mathbf{C}\mathbf{y}_{t-1} + \boldsymbol{v}_t, \qquad \boldsymbol{v}_t \sim \mathcal{N}(0, \mathbf{V})$$

where $\boldsymbol{w}_t$ and $\boldsymbol{v}_t$ are additive Gaussian noise terms with covariance matrices $\mathbf{W}$ and $\mathbf{V}$, respectively [Dangi, 2015]. The KF applications were reported in various MEA [Cunningham et al., 2010] [Gilja et al., 2012] [Hochberg et al., 2012] [Jarosiewicz et al., 2015] [Kim et al., 2011, 2008] [Perge et al., 2013] [Simeral et al., 2011] and ECoG [Kellis et al., 2012] [Marathe and Taylor, 2013] online BCI experiments as well as in MEA [Willett et al., 2018] and ECoG [Eliseyev and Aksenova, 2016] [Pistohl et al., 2008] [P. T. Wang et al., 2013] offline trajectory reconstruction studies. The Unscented Kalman Filter (UKF) is a non-linear version of the KF which uses a gaussian variable estimation with a deterministic sampling technique known as the unscented transformation [Haykin and John Wiley & Sons, 2001]. The MEA neural signal decoding using UKF was reported in preclinical online [Hotson et al., 2016b] [Ifft et al., 2013] [Z. Li et al., 2009] and offline [Ma et al., 2017] [Tseng et al., 2019] BCI studies. In [Ifft et al., 2013], Ifft used a UKF for the online decoding of bimanual movements from monkeys' neural signals recorded with MEA.

Other alternative dynamic decoding algorithms such as Particle, Point-Process or Laplace-Gaussian Filtering [Schaeffer and Aksenova, 2018], Long short-term memory (LSTM), Recurrent Neural Networks (RNN) [Du et al., 2018] [Tseng et al., 2019] [Xie et al., 2018] were applied to offline BCI experiments.

Dynamic decoders highlighted relevant results to decode continuous variables from brain neural signals in every type of recording and experimental paradigms. Only few static and dynamic algorithm comparative studies were reported. Nevertheless, some conclusions emerged from the state of the art. Over the last few years, Kalman filter families hogged the field of BCI continuous decoding based on MEA recording system [Cunningham et al., 2010] [Fan et al., 2014] [Gilja et al., 2012, 2012] [Hochberg et al., 2012] [Hotson et al., 2016b] [Ifft et al., 2013] [Jarosiewicz et al., 2015] [Kim et al., 2011, 2008] [Z. Li et al., 2009] [Ma et al., 2017] [Orsborn et al., 2012] [Perge et al., 2013] [Shenoy and Carmena, 2014] [Simeral et al., 2011] [Tseng et al., 2019] [Vaskov et al., 2018] [Willett et al., 2018]. Kalman filter algorithms highlighted good SUA and MUA spike decoding in offline and online BCI experiments, outperforming other algorithms in preclinical and clinical open-loop and closed-loop experiments [Kim et al., 2008] [Koyama et al., 2010] [Schaeffer and Aksenova, 2018]. They are suitable for online spike decoding and do not require large

computational burden. However, the overwhelming superiority of the Kalman filter family was not extended to other recording systems. The benefits of dynamic decoders compared to static ones are still ambiguous for ECoG decoding [Schaeffer and Aksenova, 2018]. The Kalman filter algorithm was outperformed by static decoders in [Eliseyev and Aksenova, 2016] whereas the Kalman filter provided better decoding performance in [Marathe and Taylor, 2013]. From epidural ECoG dataset, static based continuous decoders outperformed a Switching Kalman Filter (SKF) and provided similar results than a Wiener filter [Schaeffer and Aksenova, 2016a]. A deeper investigation on the relevance of dynamic decoders still needs to be led for BCI based on neural signal population recordings.

#### 3.4.1.3. Hybrid decoders

Regression decoder extracts from the neural signals kinematic output variables. However, classic continuous decoder presented limitations for asynchronous BCI experiments which require to switch between control and intended idle phases. Traditional decoders reported non-zero velocity during intended idle state resulting in inappropriate movements of the prosthesis [Suway et al., 2013] [Velliste et al., 2014]. In the case of multi-limb effector control, neuro-prostheses may benefit from distinguishing between periods of control with only part of the prostheses available (one limb) to avoid false activations of the other limbs of the effector which may be disturbing and stressful for the patient.

One possible approach is to evaluate a decoder for continuous kinematic variable decoding and another for mental state decoding to extract complementary information and combine their outputs. Hybrid discrete/continuous decoders involve a classifier that inhibits or enhances the continuous decoding output variables depending on the activated mental state [Schaeffer and Aksenova, 2016a].

Various hybrid models were integrated into preclinical experiments, commonly to detect idle versus active state during continuous effector control experiments or to discriminate finger activations during alternative finger movement experiments. Hybrid decoders using LDA classifier to detect idle versus active movement state combined with a Laplace Gaussian Filter [Velliste et al., 2014] or a Kalman filter [Aggarwal et al., 2013] for continuous movement decoding were reported in offline MEA preclinical experiments. A similar decoder using LDA coupled with a kinematic decoder was integrated into online MEA closed-loop preclinical experiments in [Suway et al., 2013]. Offline finger movement decoding from ECoG neural signals was performed based on a SVM classifier and a linear regression decoder [Elgharabawy and Wahed, 2016]. During an offline clinical ECoG study, a linear Bayesian classifier was applied to estimate the posterior probabilities of the idle and the movement states before applying a Kalman filter-based trajectory decoder if the movement state was more likely to be activated [P. T. Wang et al., 2013]

Among the hybrid models, the Mixture of experts (ME) architecture supposes that (multiple) intended idle and active states are associated to specific movements or actions that can be independently shaped by regression models called "experts". The ME output variables are estimated with the best expert or the combination of multiple experts. The expert selection/combination is performed by a so-called discrete "gating" model [Jacobs et al., 1991].

ME architecture was integrated into online finger decoding preclinical experiments to decode MEA neural signals using neural networks (NN) as gating and experts [Aggarwal et al., 2008]. Offline preclinical 3D reaching trajectory reconstruction from MEA neural signals using normalized Least Mean square regressions experts and MLP gating was reported in [Kim et al., 2003]. In ECoG offline study, Flamary et al mix linear ridge regression gating and linear regression experts to decode finger movements [Flamary and Rakotomamonjy, 2012]. Additionally, Bundy combined a Logistic Regression gating model with PLS experts for offline 3D trajectory reconstruction from ECoG recording of epileptic patients to take into account the intended idle state into the decoding process [Bundy et al., 2016]. Furthermore, Choi, in an offline preclinical study, mixes a LDA classifier and PLS experts for unimodal and bimanual 3D hand trajectory reconstructions from the ECoG recordings of a non-human primate [Choi et al., 2018]. Finally, a ME architecture, built with a mixture of multi-layer perceptrons, was tested in offline non-invasive EEG studies [Ebrahimpour et al., 2012] [Kheradpisheh et al., 2014].

Previously presented hybrid decoders employed the state of the art static and dynamic regression algorithms. However, all of them relied on static classifiers as gating models. The integration of dynamic classifiers into hybrid decoder as dynamic gating models was reported in offline and online clinical ECoG [Wang et al., 2011] and preclinical MEA [Achtman et al., 2007] [Darmanjian et al., 2003] [Kao et al., 2017] based experiments. HMM was combined with a moving average model [Darmanjian et al., 2003] and a REFIT-Kalman Filter [Kao et al., 2017] to discriminate active and idle states. Finally, a ME with dynamic gate estimation was reported by Schaeffer who combined a dynamic HMM classification with PLS experts to decode monkeys 3D arm trajectory [Schaeffer and Aksenova, 2016a] and Hotson who used a HMM with an Unscented Kalman Filter during preclinical experiments [Hotson et al., 2016b].

Hybrid decoder is a simple strategy to create an asynchronous BCI system. Instead of estimating a continuous model which must perform zero-velocity prediction during intended idle state, training a classifier dedicated to idle state detection to bypass the non-zero velocity outputs of the decoder is an easier strategy that highlighted good performance with invasive and non-invasive recording systems during preclinical and clinical online BCI experiments. Moreover, ME structure is a good way to introduce non-linearity based on linear models. As previously mentioned, non-linear models are commonly related to large computational burden and risk of overfitting. Non-linearity based on multiple linear models may be a solution to reduce the drawbacks of the non-

linear models. Nevertheless, only few online experiments with hybrid and/or ME architectures were reported.

### 3.4.2. Closed-loop decoder adaptation (CLDA) and incremental adaptive decoders

Neural signals are known to be non-stationary signals [Shenoy et al., 2006] with high variability. Brain signal non-stationarity is related to inter (subject to subject) or intra (sessions to session or trials to trials) variability [Clerc et al., 2016b]. Intra-variability clusters short-term variation related to attention, mood and muscle tension and long-term variation caused by patient's learning/adaptation (e.g. new MI strategy, skills, etc.) [Mladenović et al., 2017]. Additionally, neurons firing patterns continuously changed over hours, days and months requiring constant recalibration of the BCI decoders based on MEA recordings [Perge et al., 2013] [Simeral et al., 2011].

Several studies reported the poor decoding performance of models calibrated with passive subjects (open-loop procedure) when they were applied in online closed-loop experiments. These variations may be related to the modification of neuronal activity patterns with and without sensory feedback [Clerc et al., 2016b] [Schlögl et al., 2010]. The sensory feedbacks modify the neural signals to decode and therefore requires to be taken into account during the model calibration procedure. The calibration protocols integrating the patient's feedback into the model training are referred to as closed-loop decoder adaptation (CLDA) in opposite to open-loop decoder calibration [Dangi et al., 2014] [Orsborn et al., 2014]. The CLDA strategy with a "human-in-loop" training procedure highlighted drastic different decoder parameters compared to protocols with passive subjects (open loop) during the calibration phase [Cunningham et al., 2010] [Jarosiewicz et al., 2013]. Additionally, the CLDA strategy outperformed open-loop calibration in numerous studies [Lebedev and Nicolelis, 2017] [Murphy et al., 2016] [Orsborn et al., 2014] [Jarosiewicz et al., 2013]. CLDA provides a solution to adapt the model parameters depending on the neural signals related to the patient's feedback creating a closed-loop system where the patient is learning from the model and the model is adapting to the patients.

CLDA models the relationship between the neural features and the subject's motor intention inferred by the training data provided during the model calibration phase. CLDA can be dissociated into two main strategies, namely the offline and online CLDA. With the offline CLDA strategy, the model calibration is performed offline after the end of a closed-loop experiment whereas the calibration is achieved during the ongoing experiment with online CLDA procedures.

Generally, an offline or online CLDA procedure follows the same steps. The decoder is firstly initialized with a preliminary open-loop imagery task in which the subject imagines controlling a preprogrammed effector (cursor, robotic arm, etc.) which moves

automatically. From this non-perfect open-loop model calibration, closed-loop experiments can be performed and the model can be re-evaluated using the recorded closed-loop data during online or offline (model fixed during the online application) CLDA procedure. Additional recalibration of the model can be performed several times to optimize the closed-loop decoder using an online or offline CLDA procedure (Figure 3-2) [Brandman et al., 2018] [Shenoy and Carmena, 2014].

CLDA does not necessarily require online calibration even though online model re-calibration procedure benefits from many advantages. Online adaptive decoder shortens and streamlines the CLDA procedure which is a great asset considering that BCI is dedicated to patients who may have trouble keeping engaged and focused during long calibration phases [Brandman et al., 2018].

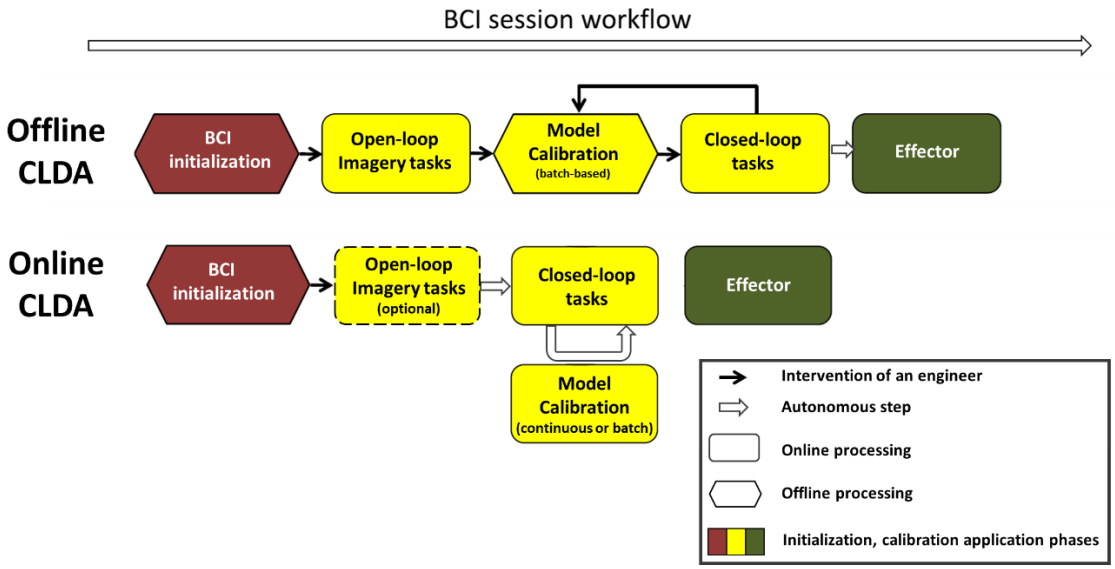



Figure 3-2: The different closed loop decoder adaptation (CLDA) procedure reported in the BCI field. Representation of typical BCI with CLDA calibration workflow from classic decoder using open-loop motor imagery model initialization to the entire system calibrated through closed-loop experiments. Hexagonal and rounded squares represents the offline and online steps respectively. No active involvement of the subjects during the offline steps. Black arrows represent the intervention of an engineer in the opposite to white arrows which represent self-managed step. Red, yellow and green color represent the initiation, the calibration and the use of the BCI system. The schematic is inspired and extracted from the study [Brandman et al., 2018].

Conventional offline and online CLDA procedures re-estimate the model parameters by concatenating all the recorded data and re-evaluating the model using a machine learning algorithm on the entire dataset. However, such procedures are not suited for long-term experiments with regular model recalibration and high-dimensional dataflow processing because the model recalibration requires increasing memory consumption and computational power as long as the model is re-estimated and input and output variables are stored.

Incremental adaptive decoders are causal algorithms that update or re-estimate the model parameters in an incremental manner with a continuous or batch learning procedure based on the new incoming data and previously computed models [Lotte et al., 2018] [Schlögl et al., 2010]. They generally rely on strategies which do not require saving all the previously recorded samples using basic statistic estimators. This constrain restricts the possible algorithms to more efficient and optimized methods with low computational cost than previously presented offline algorithms. CLDA algorithms are typically divided into two main components, the real-time application of the model to infer a subject's intended movement and the update block to incrementally adapt the model parameters [Dangi, 2015]. The updating rule can rely on batch-based algorithms which update the models during specific timed events occurring every second, every 10 minutes, after each trial, etc. or in a continuous manner after each decoder iteration [Shpigelman et al., 2009].

This section introduces the various adaptive BCI algorithms and strategies for online and offline CLDA.

#### 3.4.2.1. CLDA with classifiers

Model adaptation is commonly based on supervised learning, nevertheless, unsupervised strategies to update the classifier parameters are more frequently reported than previously presented study with fixed decoders.

In 2004, an offline open-loop study showed the possible improvements related to online adaptive calibration using means and covariance matrices update [Millan, 2004]. In 2006, Shenoy highlighted the benefits of an adaptive classifier to manage the EEG neural signals shifting in offline binary classification study [Shenoy et al., 2006]. The study proposed an offline CLDA procedure with two LDA adaptation strategies: RETRAIN and REBIAS. With the REBIAS strategy, Shenoy underlined the performance improvement of LDA classification just using a bias shifting approach. Since, numerous variations of the adaptive LDA algorithm were reviewed in EEG experiments [Lotte et al., 2018] [Mladenović et al., 2017] [Schlögl et al., 2010].

Li proposed a non-incremental adaptive decoder named Importance Weighted Linear Discriminant Analysis (IWLDA) with covariate shift adaptation [Y. Li et al., 2009] and tested this algorithm on EEG and ECoG dataset. Another simple approach presented in [Mend and Kullmann, 2012] was applied during online binary classification experiments. The BCI system is divided into two parallel processes. The main loop applies the classifier for neural signal decoding. The second loop concatenated the training data (to store the entire dataset) and performed the training procedure on all the cumulated data (the feature selection was performed through cross-validation and LDA re-training). This method had the benefit of being simple and straightforward. Nevertheless, this solution is restricted to small training dataset, simple decoding algorithms and is highly dependent on the computing power of the computer.

SVM algorithm is a popular algorithm for offline neural state classification. Numerous studies extended the SVM classifier to adaptive algorithms [Lotte et al., 2018] [Mladenović et al., 2017]. The Adaptive mixture of Relevance Vector Machine (a sparse version of the SVM algorithm) [Nguyen et al., 2019] was tested in an online 4 states classification experiments from online EEG neural signals. Online BCI application with offline CLDA using linear kernel SVM with covariate shift detection was reported to control an exoskeleton hand for rehabilitation application [Chowdhury et al., 2017]. Another offline study reported the application of adaptive Kernel Fisher SVM (KF-SVM) [Yang et al., 2017].

Other less common classifiers were reported in the BCI field to handle non-stationary neural signals. Among these decoders, an incremental adaptive probabilistic neural network was designed for online binary classification [Hazrati and Erfanian, 2010], the Extended Kalman filter (EKF) was tested during offline binary classification studies [Ji Won Yoon et al., 2008] [Yoon et al., 2008] and the semi-supervised adaptive Naïve Bayesian Parzen window (NBPW) classifier was tested offline in an EEG study [Ang et al., 2011]. The Online Dictionary Learning using Correlation-Based Least Squares Update (CBLSU) was reported for the offline discrimination of 3 classes based on EEG dataset [Sharghian et al., 2019].

Finally, offline CLDA integrating the patient feedback was performed with a neural network combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to decode various hand shapes based on LFP recordings (MEA). The decoder allows a tetraplegic patient to control his forearm thanks to functional electrical stimulation (FES). The network parameters were re-trained during an offline supervised or unsupervised procedure using the past training dataset concatenated with new closed-loop data [Schwemmer et al., 2018].

### Incremental CLDA classifiers

The previously presented adaptive decoder generally concatenated the stored training dataset to recalibrate the model with the entire dataset and improve the model performance. However, this strategy might be limited if the model recalibration requires a long training procedure (for example to control complex effector with high DoF) as it requires storing all the data to re-evaluate the model without losing information. Another solution tested during BCI applications is to incrementally update the model with new data or incrementally update necessary and sufficient variables to re-evaluate the model. The algorithms referred to as an incremental adaptive decoders perform a CLDA procedure with a reduced computational time and computer memory which suit them for online BCI application.

Vidaurre tested offline four supervised and unsupervised incremental adaptive LDA algorithms (adaptive mean, adaptive covariance matrices, etc.) using EEG neural signals from four different dataset [Vidaurre et al., 2011]. Additionally, incremental adaptive

Kalman-LDA was tested in offline EEG studies [Hsu, 2011]. Incremental adaptive Kalman-LDA was compared to an incremental adaptive version of the QDA algorithm referred to as the ADaptive Information Matrix (ADIM) which performed incremental adaptive non-linear classification [Vidaurre et al., 2007, 2006b]. Online adaptive binary classification from EEG neural signals was reported using incremental LDA\QR [Wen and Huang, 2017] and incremental Kalman adaptive LDA [Vidaurre et al., 2007] [Woehrle et al., 2015]. During online EEG based-experiments, an adaptive SVM classifier was applied for BCI speller applications using incremental SVM [Ma et al., 2020] [Vo et al., 2018] [Woehrle et al., 2015] and iterative semi-supervised SVM [Long et al., 2011].

Finally, less conventional algorithms achieving incremental adaptive model calibration procedures were reported. An incremental adaptive classifier named adaptive Extreme machine learning was tested in an offline EEG-based study for binary classification [Bamdadian et al., 2013]. Incremental adaptive fuzzy models were reported for offline 2-class motor imagery task discrimination and offline 4-state classification from EEG neural signal using first-order Takagi-Sugeno fuzzy model [Rong et al., 2018b] and self-regularized supervised Gaussian fuzzy adaptive system adaptive resonance theory [Jafarifarmand and Badamchizadeh, 2020] respectively. Additionally, several incremental adaptive Riemannian classifiers based on minimum distance to the mean (MDM) were designed and tested in offline 3 and 4-states classification studies in [Kumar et al., 2019].

### Unsupervised CLDA classifiers

Unsupervised learning is a more common strategy for model re-estimation than for model calibration. With unsupervised adaptation, the labels associated with the recorded neural signals are unknown. Unsupervised model adaptation which only required neural signals features were reported in BCI experiments.

For example, the global mean of neural signal features was adapted to overcome the shift between the distributions estimated during the calibration procedure and the test period in offline EEG studies [Vidaurre et al., 2011] and in online EEG experiments for upper-limb control through functional electrical stimulations [Vidaurre et al., 2016]. A similar strategy with global mean and global covariance matrix adaptations was tested in [Vidaurre et al., 2011].

Another unsupervised learning strategy referred to as a semi-supervised adaptation in [Lotte et al., 2018] combined both the model initially calibrated with labeled data and the new incoming neural signals with unknown labels to perform the model re-training/adaptation.

One solution is to use the classification outputs of the current decoder and the associated neural signals as new training data for model re-estimation. This easy to implement strategy was reported in [Hasan and Gan, 2009] for offline EEG classification based on incremental adaptive unsupervised Gaussian Mixture Models with a sequential expectation-maximization procedure. Similar learning procedures using unsupervised

incremental adaptive SVM were reported for offline and online MEG experiments [Spüler et al., 2012b]. Spüler introduced a threshold criterion to only select the samples which were most likely well classified. As a wrong prediction could lead to degrading the classifier performance, only the samples with high probabilities of being correct were stored for incremental learning [Spüler et al., 2012b].

Another proposed solution was to integrate other stereotyped neurophysiological signals to infer the neural signals labels. Some semi-supervised strategies relied on Error Potentials (ERP) detection [Lotte et al., 2018]. ERP detection provides a good estimation of the correctly and badly classified unlabeled data. This strategy was tested offline in EEG based classification study using Weighted Majority Voting (WMV) based on adaptive SVM [Oliver et al., 2013] and was reported several times for offline and online P300 BCI applications [Lotte et al., 2018].

**CLDA with classifiers: conclusion**

The integration of a CLDA procedure in decoders dedicated to BCI application is a poorly explored area where most of the proposed adaptive classifiers were tested during offline and online EEG experiments or in offline studies based on dataset acquired with invasive recording systems. Moreover, the reported closed-loop decoder was not all based on incremental adaptive strategies which might be an obstacle for real-time classier updates due to hardware limitations (increasing memory, computing time and resources required, etc.). Clinical and pre-clinical experiments using invasive recording systems with real-time incremental CLDA remained unexplored in the BCI domain. Additionally, to our knowledge, dynamic adaptive classification was weakly studied. Whereas the application of HMM was reported in BCI experiments for reaching and grasping control of a robotic arm [Fifer et al., 2014], classification of finger movements [Hotson et al., 2016a], detection of evoked neural signals [Lisi et al., 2018] [Moses et al., 2018] or prediction of the idle state during trajectory decoding [Kao et al., 2017], no study on the online calibration of HMM in the BCI domain was reported. In other research fields, a HMM with low computational cost and potential for adaptive classifier calibration was reported in [Chan and Englehart, 2005] to command prostheses through myoelectric control. Additionally, several theoretical articles proposed strategies to extended online HMM to the adaptive classifier domain based on adaptive expectation-maximization procedure, gradient descent techniques [Cappé, 2011] [Chis and Harrison, 2015] [Digalakis, 1999] [Khreich et al., 2012] [Mongillo and Deneve, 2008] [Stiller and Radons, 1999], etc. Adaptive dynamic classifier exploitation for real-time BCI decoding is an interesting research field to explore.

### 3.4.2.2. CLDA for continuous outputs

This section introduced the CLDA procedure and the adaptive algorithms designed to decode continuous variables and control continuous effectors.

The first offline CLDA approach to control a 2D cursor movement was reported by Wolpaw [Wolpaw and McFarland, 2004] during online EEG experiments. The control of the cursor was provided to four healthy subjects by a linear equation which combined the weighted amplitude of the μ and β frequency bands (one for the vertical and one for the horizontal displacement). After each trial, the weights were adapted based on the past trials using an offline least-mean squares algorithm [Wolpaw and McFarland, 2004]. In other experiments, based on Kalman filter algorithms, two tetraplegic patients controlled a 2D cursor through MEA neural signals decoding [Kim et al., 2008] where the model was updated at the end of each closed-loop block (a group of several trials).

Since, other algorithms with offline CLDA were proposed for communication BCI applications to tetraplegic and ALS patients based on MEA neural signals [Jarosiewicz et al., 2015, 2013] [Kim et al., 2011]. Jarosiewicz proposed a real-time point and click cursor control for virtual typing applications [Jarosiewicz et al., 2015, 2013]. They used a Kalman filter to control the cursor movement whereas the LDA classifier detected the click activation. Both decoders were recalibrated between blocks of trials. Participants were able to pause the BCI application by their brain signals. During those breaks, the last N blocks were used to recalibrate the BCI decoder which lasts between 1-3 min before restoring the cursor control to the patient [Jarosiewicz et al., 2015]. Similar clinical experiments with real-time MEA neural signal decoding based on offline CLDA procedure using Kalman filter and univariate Gaussian classifier with dynamic/temporal post-processing were reported in [Kim et al., 2011].

Another offline CLDA procedure was tested during preclinical MEA experiments by Gilja in [Gilja et al., 2012]. This study proposed a decoder named the recalibrated feedback intention Kalman filter (ReFit-KF) which performed a discrete batch-based calibration algorithm to update the decoder 10–15 min after the initial seeding [Gilja et al., 2012] [Orsborn et al., 2012]. Additionally, Gilja proposed a recursive procedure to enhance the model decoding accuracy which re-calibrated the ReFit-KF based on new neural signals and shifted closed-loop decoder outputs. After the estimation of the initial parameters through an open-loop calibration, the non-optimal model was used for closed-loop cursor control. The neural signal recorded during the closed-loop experiment and an "intended estimate" of the cursor velocity was used for the model recalibration (Figure 3-2). The "intended estimate" of the cursor velocity was generated by rotating the decoded velocity vectors such that they pointed straight towards the targets. This velocity estimation assumed that the monkey was constantly intending to move directly towards the target [Gilja et al., 2012] [Shenoy and Carmena, 2014]. ReFit-KF performed a batch-based CLDA applying one discrete decoder update 10–15 min after the initial seeding [Orsborn et al., 2012]. A single ReFit-KF CLDA update was sufficient to achieve a significant improvement in reach kinematics [Dangi, 2015]. However, the procedure could be repeated several times to improve the decoding performance. This innovation highlighted significant continuous decoding performance improvements. Refit-KF was

used in many applications to decode MEA neural signals and highlighted good continuous performance in many online preclinical [Fan et al., 2014] [Vaskov et al., 2018] and clinical [Willett et al., 2018] experiments based on continuous center-out-back or finger flexion tasks.

Kalman filter and Support vector Regression were outperformed by a non-linear model named Kernel-AutoRegressive Moving Average (Kernel-ARMA) in [Shpigelman et al., 2009] during open-loop hand tracking study. Moreover, Kernel-ARMA was applied in an online preclinical 3D cursor experiment using MEA neural signals. Kernel-ARMA performed an online re-estimation of the model based on a limited number of training examples: the oldest training examples are removed from the training dataset.

### Incremental CLDA continuous decoders

Other decoders inspired by the previously presented "intended estimate" ("cursor-goal") procedure of Gilja were designed. They relied on incremental batch-online CLDA with a medium update frequency ("adaptation on intermediate Time-Scales") [Orsborn et al., 2012, 2011]. For example, an alternative of the ReFit-KF calibration procedure named "SmoothBatch" CLDA was proposed to incrementally update the Kalman filter decoder during online batch recalibration (1-2 min time scale) using exponentially weighted sliding average and recursive maximum likelihood algorithm [Dangi, 2015] [Dangi et al., 2014] [Orsborn et al., 2014, 2012]. The observed neural activity and intended cursor kinematics were collected over one batch period before to compute a weighted average of the current parameters with those estimated from the new batch of neural activity [Dangi, 2015]. SmoothBatch-KF showed good results in online 2D cursor control based on monkeys implanted with MEA recording devices [Dangi, 2015] [Dangi et al., 2014] [Orsborn et al., 2014, 2012].

Finally, CLDA can be performed online in shorter time scales going from a recalibration at each iteration (at every new sample) or in the order of few seconds. An incremental adaptive version of the Kalman algorithm was designed such as adaptive Kalman filter (AKF) [Dangi et al., 2011], Unscented KF with unsupervised Bayesian regression updater [Li et al., 2011] or Gaussian Process regression Discriminative KF (GP-DKF) [Brandman et al., 2018]. AKF performed the model parameter recalibration at each decoder iteration. AKF was only tested on simulated data [Dangi et al., 2011] and was compared to SmoothBatch-KF during online preclinical MEA neural signals decoding in [Dangi, 2015]. The incremental adaptive Unscented KF and the GP-DKF algorithms updated their parameters every few seconds and performed, based on MEA neural signals, online 2D pursuit task using monkey [Li et al., 2011] and 2D cursor center-out real-time decoding from three tetraplegic patients [Brandman et al., 2018], respectively. Other dynamic decoders were adapted for online incremental model calibration. Online preclinical 2D random pursuit task experiments were reported in [Suminski et al., 2013] using an

adaptive Wiener filter with incremental gradient descent for MEA neural signal decoding.

Based on MEA recordings of non-human primates, Shanechi designed the Point Process Filter (PPF) which updated its parameters at every spike event and used an optimal feedback-control model to infer the velocity intention during adaptation [Shanechi et al., 2017, 2016, 2014]. This decoder outperformed the SmoothBatch-Kalman filter and highlighted robust and high online decoding performance in 2D cursor control preclinical experiments [Shanechi et al., 2016, 2014]. Additionally, PPF allowed studying the impact of the decision and feedback rates to control neuroprosthetic devices [Shanechi et al., 2017].

With the exception of Wolpaw's study [Wolpaw and McFarland, 2004], all the presented articles were performed based on MEA recording devices. Other MEA-based CLDA procedures were reviewed in [Dangi, 2015]. Eliseyev presented an incremental adaptive algorithm referred to as the Recursive Exponentially Weighted N-way Partial Least Square (REW-NPLS) [Eliseyev et al., 2017] for ECoG neural signal decoding. REW-NPLS is a PLS algorithm extended to input and output tensor variables that perform model recalibration in few seconds (10 seconds). This algorithm highlighted good results in offline 3D arm monkey movement reconstruction study based on ECoG neural signals decoding as well as during offline fingers movement reconstruction study from four healthy subjects using MEG recordings [Eliseyev et al., 2017]. Moreover, REW-NPLS, in the CLINATEC "BCI and Tetraplegia" clinical trial, provided to a tetraplegic patient, based on wireless ECoG recordings, the 3D hand translation control of an exoskeleton during real-time point-to-point pursuit task experiments [Benabid et al., 2019].

### CLDA for continuous outputs: Conclusion

CLDA procedure with adaptive decoders offers numerous advantages compared to open-loop calibration procedure and fixed (non-adaptive) decoders to decode brain neural signals during real-time BCI applications. Firstly, adaptive decoders are more robust to brain neural signal non-stationarity. In the case of neural signal patterns changes across time, fixed model performance may degrade whereas adaptive decoder can modify the model parameters to fit the brain neural signals variations and remains effective [Li et al., 2011] [Lotte et al., 2018]. Secondly, numerous articles highlighted that neural signal patterns during open-loop (without feedback) and closed-loop (with feedback) BCI sessions were different and lead to different model parameters [Jarosiewicz et al., 2013]. This difference may be related to the neural signals generated by the sensory feedbacks added during closed-loop experiments or the co-adaptation of the user and the decoder. The CLDA with "human-in-loop" training highlighted better decoding performances than open-loop calibrated models.

However, as depicted, numerous CLDA procedures were reported with diverse "time-scale of adaptation" [Dangi, 2015]. The update might be performed at the end of every

session [Jarosiewicz et al., 2015, 2013], trials [Wolpaw and McFarland, 2004], 15 min [Gilja et al., 2012] [Orsborn et al., 2012], 1 min [Dangi, 2015], few seconds [Brandman et al., 2018] [Eliseyev et al., 2017], every sample [Dangi et al., 2011], etc. Fast update calibration provides quicker feedback to the patient and decreases the time required to converge to an adequate neural control [Brandman et al., 2018]. A balance should be found between model accuracy, model complexity and decoder update rate.

Most of the reported adaptive regression decoders were based on clinical or preclinical MEA neural signals processing or offline studies using EEG dataset. With the exception of [Eliseyev et al., 2017], ECoG-based adaptive decoders were not reported. Further investigation on the use of CLDA using an incremental adaptive decoder should be carried out.

#### 3.4.2.3. Adaptive feature selection

Commonly, non-stationarity adaptation and neural signals related to patient's feedback are integrated into the BCI system through the BCI decoder adaptation with an adaptive classifier or an adaptive regression model. Another proposed solution was to modify the features extraction step in order to modify the computed feature depending on the brain variability [Mladenović et al., 2017]. As an example of simple feature non-stationarity tracking, Jarosiewicz designed a communication BCI for a tetraplegic patient using MEA neural signals which recomputed the raw average firing rate of the neurons using an offline batch update procedure to maintain an accurate baseline [Jarosiewicz et al., 2015] and avoid bias.

Similarly to the adaptive decoders, adaptive feature selection algorithms generally relied on state of the art feature selection algorithms. Several independent component algorithms were reported in [Hsu et al., 2016] to manage brain signal non-stationarity or to perform online adaptive denoising/artifact detection. For example, the online recursive independent component analysis (ORICA) [Chen and Fang, 2017] [Ho et al., 2019] [Hsu et al., 2016, 2015] [Wang et al., 2018] was tested during offline simulations based on artificially noised EEG neural signals. Online calibrated common spatial pattern algorithms were tested offline in EEG motor imagery studies using an incremental or an adaptive common spatial pattern (ICSP and ACSP) [Costa et al., 2018] [Mobaien and Boostani, 2016] [Song and Yoon, 2015] [Zhao et al., 2008]. Vidaurre in [Vidaurre et al., 2006b] proposed a fully online adaptive BCI using adaptive autoregressive (AAR) feature selection and an ADIM classifier (the incremental adaptive version of the non-linear classifier QDA) during offline and online EEG neural signals experiments. An extension of the AAR to the multivariate case was tested during an offline analysis of EEG recordings in [Hettiarachchi et al., 2015]. Incremental learning of axDawn spatial filter algorithm was applied online during single-trial detection of ERPs [Woehrle et al., 2015]. Finally, other less conventional feature selection algorithms such as adaptive spatial

filter [Morales-Flores et al., 2014] or brute force algorithm [Mend and Kullmann, 2012] were tested offline through ECoG and EEG neural signal decoding studies respectively.

Adaptive feature selection is a poorly explored area and was mainly evaluated during offline EEG studies. This strategy highlighted interesting results to handle brain non-stationarity, enhance signal-to-noise ratio and remove artifacts. However, deeper investigations on the impact of feature adaptation on the decoder performance should be carried out. Indeed, with a fixed decoder, modifying the input feature space can deteriorate the decoding accuracy. Vidaurre in [Vidaurre et al., 2006b] adapted both the feature extraction block and the model parameters nevertheless more investigations on this type of "fully" adaptive decoder must be achieved.

#### 3.4.2.4. Alternative strategies

It must be mentioned that Sussillo proposed an alternative or complementary algorithm to the presented adaptive decoders in order to handle brain non-stationarity across time. The algorithm, named Multiplicative Recurrent Neural Network (MRNN) [Sussillo et al., 2016] handled brain signal non-stationarity by concatenating multiple models calibrated with diverse "recording conditions". RNN "remembered" the state over time to handle dynamic complex and time-varying relationship between neural signals and movements. The "multiplicative" structure allowed the neural signal features to influence the recurrent weights. MRNN learned a "library" of various neural-kinematic mappings representing the natural dynamic of the RNN and various neural recording conditions [292]. Combining the "library" architecture to handle various neural conditions and the adaptive decoder to manage the natural drifting of the neural signals could be an interesting solution to test for future BCI experiments. This algorithm was evaluated during online closed-loop center-out experiments with MEA neural signal decoding from a monkey and outperformed the state-of-the-art REFIT-KF algorithm.

## 3.5. Post-processing

Post-processing techniques reported in BCI applications relied on prior knowledge about the desired output variables, the restrictions related to the effector or the subject's safety. They were integrated into BCI systems to smooth the output of the decoder or to apply modifications to the decoder output such as output thresholding, enable state transition for a defined time period [Schaeffer and Aksenova, 2018], etc. Jarosciewicz applied an adaptive bias correction to the decoded cursor velocity using an exponentially weighted mean of the velocity outputs exceeding a predefined speed threshold during online BCI communication experiments based on the MEA neural signals of tetraplegic patients [Jarosiewicz et al., 2015]. Similarly, to reduce click errors related to noisy classification from MEA neural signals of a tetraplegic patient, Kim averaged the click state predictions over a period of time (typically 500 ms) [Kim et al.,

2011]. To enable a monkey to control in real-time a robotic arm from intracortical recordings, Velliste applied to the continuous output variables a 5 to 11-sample temporal filter to smooth the decoded trajectory [Velliste et al., 2008]. The Graz-BCI system integrated a post-processing step based on a dwell time [Pfurtscheller et al., 2010] [Townsend et al., 2004] and refractory period [Townsend et al., 2004] for online synchronous 2D classification [Townsend et al., 2004] and SSVEP-Based Orthosis control [Pfurtscheller et al., 2010] experiments from EEG neural signals. The dwell time was a fixed period of time in which the decoder output variables should be above a defined threshold to be considered as a valid event detection [Pfurtscheller et al., 2010] [Townsend et al., 2004]. The refractory period was the duration after a detected valid event during which new valid events were ignored [Townsend et al., 2004].

Other possible post-processing strategies based on the detection of neurophysiological patterns related to BCI errors were reported. The detection of decoding errors from the neural signals was used to automatically correct the decoded output variables. This strategy was commonly used for EEG-based BCI applications with Error Related Neuronal Response (ERNR) detection algorithms to correct decoder mistakes. For example, the incorrect decoded letters from an EEG-based BCI P300 speller were automatically deleted if an ERNR was detected in [Spüler et al., 2012a]. An extensive review of the integration of ERNR-based correction for BCI application, particularly focused on EEG neural signals decoding, was proposed in [Chavarriaga et al., 2014]. Additionally, few studies highlighted error-related neural signal modulations from invasive recording systems. Shenoy's team proved that error-related neural signals could be extracted from the premotor and primary motor cortices using MEA recordings and could be evaluated to improve the BCI decoding performances [Even-Chen et al., 2018, 2017]. They stressed relevant results in preclinical online experiments [Even-Chen et al., 2017] and clinical offline study [Even-Chen et al., 2018]. ERNR patterns were detected from ECoG motor cortex neural signals in [Milekovic et al., 2013] but were not integrated in ECoG-based BCI systems to improve the decoding performance.

Other post-processing strategies based on moving average, confidence measures of the classification outputs and blocking state transitions approaches were reported in EEG experiments [Bashashati et al., 2007a] [Schaeffer and Aksenova, 2018].

Prediction filtering across time is often applied as a post-processing step to smooth the decoder outputs [King et al., 2015]. The impact of error magnitude, smoothness, prediction delay and velocity on the closed-loop continuous movement decoding accuracy (Figure 3-3) was investigated in [Marathe and Taylor, 2015]. The conclusion of the study stressed that smoothing the predictions reduced the prediction errors but added supplementary delays. Additionally, they highlighted that minimizing the prediction delay was highly relevant especially when decoding accuracy was poor [Marathe and Taylor, 2015]. This study is relevant for post-processing filtering tuning, as

higher temporal filtering may improve the decoder prediction but also increase the prediction delay.

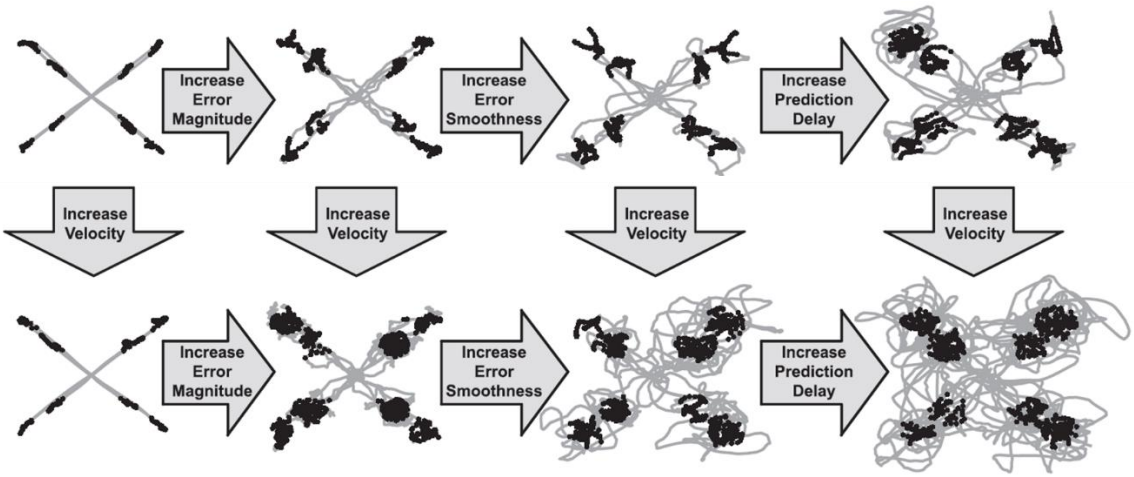



Figure 3-3: Impact of the error magnitude, error smoothness, prediction velocity and prediction delay on the closed-loop continuous decoding performance. Figure extracted from the closed-loop BCI study [Marathe and Taylor, 2015]

Generally, during BCI online closed-loop experiments, for the control of real effectors, physical boundaries were added via a post-processing step to restrict the possible position, movements, speed, etc. of the effector and assure the patient's safety. For example, in [Hochberg et al., 2012], a tetraplegic patient controlled in real-time a DEKA prosthetic arm through the decoding of her MEA neural signals. The workspace of the prosthetic arm was limited with virtual boundaries to avoid any collision with the tabletop, support stand and the patient. This type of post-processing is highly dependent on the controlled effector.

Post-processing is a powerful tool to reduce the weaknesses of the online decoders using post-processing techniques/rules that would be too complex to integrate into the model calibration procedure. Additionally, post-processing allows integrating prior knowledge on the desired task. However, post-processing must be carefully integrated into a closed-loop BCI system. A trade-off between performance improvement related to the post-processing step and delay added to the prediction must be considered to not disrupt patient's feedback and inevitably deteriorate the decoded predictions.

## 3.6. Conclusion

In this chapter, the principal blocks of a BCI transducer were described and the main approaches of the state-of-the-art BCI studies were presented. After a preprocessing step, relevant information is extracted from the brain neural signals to allow a decoder

to estimate a specific neural state (classifier) or the patient's intended continuous outputs (continuous decoder) via a neural signal decoder. Exhaustive reviews referring to less common methods used in BCI are available in [Bashashati et al., 2007a] [Lotte et al., 2018] [Schaeffer and Aksenova, 2018].

The most common studies are classification experiments based on EEG recording systems. Continuous decoding is generally performed with invasive recording systems which highlighted higher decoding performance and a higher number of controlled DoF than non-invasive ones. For online application, linear models are favored by their ease of use and calibration compared to more complex models.

Asynchronous BCIs are more realistic and representative of daily life applications than synchronous BCIs. Asynchronous BCIs introduce an "idle" state detection in which no command/action is performed by the effector [Han et al., 2020]. However, asynchronous BCIs are more complex than synchronous ones and present higher false-positive detection rates (detection of an active state instead of idle one) which is highly problematic for many applications and may be disturbing for the users. BCI state of the art with the highest number of controlled DoF are synchronous [Collinger et al., 2013] [Degenhart et al., 2018] [W. Wang et al., 2013] [Wodlinger et al., 2015]. Asynchronous BCI experiments were mainly tested based on EEG neural signals [Chae et al., 2012] [Kalunga et al., 2016] [Li et al., 2013] [Mason and Birch, 2000] [Müller-Putz et al., 2006] [Nagel and Spüler, 2019] [Ortner et al., 2011] [Saa and Çetin, 2013] [Yousefi et al., 2019]. Additionally, EEG studies were generally limited to simple asynchronous switch between active and idle states [Mason and Birch, 2000] [Müller-Putz et al., 2006] [Yousefi et al., 2019] or idle state detection for SSVEP or P300 BCI applications (speller, etc.) [Nagel and Spüler, 2019] [Ortner et al., 2011]. Only a few EEG studies integrated idle detection in more complex tasks [Chae et al., 2012] [Li et al., 2013] [Saa and Çetin, 2013]. Asynchronous decoding, while more prevalent in the EEG studies, were reported in some preclinical ECoG [Chao et al., 2010] [Choi et al., 2018] [Schaeffer and Aksenova, 2016b], preclinical MEA [Achtman et al., 2007] [Ludwig et al., 2011] [Suway et al., 2013] [Velliste et al., 2014] and clinical [Bundy et al., 2016] [Elgharabawy and Wahed, 2016] [Flamary and Rakotomamonjy, 2012] [P. T. Wang et al., 2013] ECoG studies which combined idle state detection and continuous movement predictions. Nevertheless, with the exception of the MEA preclinical experiments [Suway et al., 2013], all the invasive asynchronous BCI studies were tested offline. Asynchronous BCIs is a major condition for the majority of the BCI daily life applications and should always be integrated for the control of complex effectors.

Hybrid (discrete/continuous) is one of the solutions tested for asynchronous BCI application as well as mixing discrete and continuous outputs such as moving a cursor to a target and activate a "clicking" state. Additionally, hybrid decoders highlighted promising results for multi-limbs control experiments in ECoG online experiments [Hotson et al., 2016a] and offline studies [Choi et al., 2018] [Elgharabawy and Wahed, 2016] [Flamary and Rakotomamonjy, 2012] [Schaeffer, 2017]. However, the multi-limbs studies

with hybrid decoders were mainly restricted to individual finger movement detection [Elgharabawy and Wahed, 2016] [Flamary and Rakotomamonjy, 2012] [Hotson et al., 2016a] [Schaeffer, 2017]. Further investigation on the benefits of hybrid (discrete/continuous) decoders in the case of complex multi-limb or bimanual effector control should be carried out.

Dynamic classifiers and regression decoders were only restricted to few studies. With the exception of Kalman filters which are prevalent for continuous decoding, only a few other dynamic continuous decoders were employed in online applications. Time dependency is often integrated through post-processing. Dynamic classifiers were tested using complex algorithms such as LSTM and CRF during offline studies and outperformed HMM classifiers. However, the computational burden of these algorithms is superior to HMM decoders which is problematic for online BCI applications. HMM performance improvements compared to static decoders do not reach a consensus in the BCI field and are highly application dependent [Lotte et al., 2007]. Few offline and online preclinical experiments [Kao et al., 2017] [Schaeffer and Aksenova, 2016a] stressed the benefits of HMM for idle state detection in asynchronous BCI applications. Further investigation on the advantages of dynamic models for online closed-loop BCI applications must be achieved.

CLDA integrates the neural signal modulations related to the patient's feedback in the model calibration procedure. CLDA highlighted benefits for closed-loop BCI decoding performance as well as robustness to brain the neural signal non-stationarity. CLDA procedure should be achieved to enhance closed-loop BCI applications. However, offline CLDA algorithms require numerous training and calibration sequences to estimate a model which can be time-consuming and increase patient's mental load. Incremental online CLDA algorithms train and apply the model at the same time which presents various advantages for real-time BCI applications. However, to this date, incremental online CLDA algorithms were understudied. On the one hand, EEG-based adaptive decoders were mainly focused on adaptive feature extraction algorithms and adaptive classifiers. On the other hand, MEA-based decoders were more focused on continuous movement decoding and highlighted interesting approaches to re-calibrate the models at various time scales. Nonetheless, incremental adaptive EEG-based-classifiers were mainly tested during offline studies whereas incremental adaptive MEA based-continuous decoder performance evaluations were generally limited to online preclinical center-out experiments. Moreover, with the exception of the REW-NPLS algorithm [Eliseyev et al., 2017], no adaptive algorithms were reported with ECoG recordings. Deeper investigations on such adaptive incremental decoders and experiments should be carried out.

Chapter 4

# Online incremental adaptive multilinear switching model

## Contents

This chapter introduces the new Recursive Exponentially Weighted Markov Switching multi-Linear Model (REW-MSLM) designed for online closed-loop adaptive decoder calibration and asynchronous multi-limb effector control based on ECoG recordings. To control a multi-limb effector and handle stable idle state decoding, this algorithm relies on a Mixture of Experts architecture. This new fully adaptive decoder was derived from two algorithms named recursive exponentially weighted n-way partial least squares (REW-NPLS) [Eliseyev et al., 2017] and Markov-Switching linear model (MSLM) [Schaeffer and Aksenova, 2016b]. The first one is an adaptive tensor based decoder for closed-loop adaptive calibration. The second is a Mixture of Experts algorithm which integrates a dynamic classifier to enhance the discrimination between the active states and improve the idle state detection. Before further description of the REW-MSLM, the Mixture of Experts, the MSLM and the REW-NPLS algorithms are described.

## 4.1. Mixture of experts

Mixture of Experts (ME) can be described as the parallel computation of several predictions from different regression models named "experts" that are weighted (enhanced or inhibited) according to the input variables using a classifier referred as "gate" model [Jacobs et al., 1991] [Yuksel et al., 2012]. Gate and experts terminologies have been firstly introduced by Jacobs in [Jacobs et al., 1991].

Let $\mathbf{x}_t \in X, X \subset \mathbb{R}^m$ and $\mathbf{y}_t \in Y, Y \subset \mathbb{R}^n$ be the explanatory and the response variables, respectively, indexed by $t \in \mathbb{N}$, where $m$ and $n$ are the feature space dimensions. ME assumes [Jacobs et al., 1991] [Waterhouse and Robinson, 1994] [Yuksel et al., 2012] that the feature space of independent variables is sub-divided into $K$ (possibly overlapped) regions $X = \bigcup_{k=1}^{K} X_k$, and that $X$ is mapped to $Y$ with the set of $K \in \mathbb{N}$ local linear or nonlinear functions $\Phi = \{\varphi_k : X_k \to Y,\ k \in [1; K]\}$ called experts:

$$\mathbf{y}_t = \sum_{k=1}^{K} \delta_{k,z_t}\ \varphi_k(\mathbf{x}_t) + \boldsymbol{\varepsilon}_{k,t}.$$

Here, $z_t$ represents the selected expert at time $t$, $\delta_{k,z_t}$ is the Kronecker delta (where $\delta_{k,z_t} = 1$ if $k = z_t$ and $\delta_{k,z_t} = 0$ otherwise) and $\varepsilon_{k,t}$ is the observation noise (possibly) related to the k[th] expert, generally, supposed to be independent and identically distributed (iid). The vector $\mathbf{y}_t$ is predicted from the input variable $\mathbf{x}_t$, using the estimated expert $\hat{\varphi}_k$ fitted at the corresponding region of the neural space:

$$\hat{\mathbf{y}}_t = \sum_{k=1}^{K} \gamma_{k,t}(\mathbf{x}_t)\ \hat{\varphi}_k(\mathbf{x}_t).$$

Generally, $\hat{\mathbf{y}}_t = E(\mathbf{y}_t|\mathbf{x}_t)$ is the Bayes estimate of the response variables [Bishop, 2006] [Waterhouse and Robinson, 1994], $\hat{\varphi}_k(\mathbf{x}_t) = E(\mathbf{y}_t|\mathbf{x}_t, z_t = k)$ is the estimate issued by expert $k$, and $\gamma_{k,t}(\mathbf{x}_t) = p(z_t = k|\mathbf{x}_t)$ is the gating probability distribution of the $k^{th}$ expert at time $t$, with $\gamma_{k,t} \in [0; 1]$ and $\sum_{k=1}^{K} \gamma_{k,t} = 1$ [Bishop, 2006] [Yuksel et al., 2012].

ME is suited for naturally sub-divided dataset due to its ability to train each expert independently to a specific pattern [Jacobs et al., 1991] [Yuksel et al., 2012] which fit with multi-limb effectors. It introduces non-linearity to the model by mixing the multiple linear (or not) regression models [Yuksel et al., 2012]. ME architecture is commonly used across divers fields [Carvalho and Tanner, 2003] [Yuksel et al., 2012] with many applications in finance [Carvalho and Tanner, 2003] [Hoang and Williamson, 1998] [Yu and Cho, 2016], weather study [Jeffries and Pfeiffer, 2001], bioinformatics [Lê Cao et al., 2010] [Qi et al., 2007], facial recognition [Ebrahimpour et al., 2011] [Gutta et al., 2000], etc.

Various regressions and classification algorithms were developed to improve ME gate and experts performance [Yuksel et al., 2012]. ME was built based on Support Vector Machine (SVM) [Elgharabawy and Wahed, 2016] [Gutta et al., 2000], multilayer perceptrons (MLP) [Kim et al., 2003], Gaussian mixture model (GMM) [Yuksel et al., 2012], etc. Finally, ME was modified to have specific features for each expert [Kheradpisheh et al., 2014] enhancing the feature space sub-division of the original algorithm [Jacobs et al., 1991] [Waterhouse and Robinson, 1994] [Yuksel et al., 2012].

In motor BCI research, hybrid decoders, defined in the manuscript as a combination of discrete (e.g. classifiers) and continuous (e.g. regression) decoders, were widely spread in the case of asynchronous experiments particularly to distinguish the idle and control periods [Choi et al., 2018] [Suway et al., 2013] [Velliste et al., 2014] [P. T. Wang et al., 2013]. Hybrid models were reported for MEA-based real-time 3D reaching task preclinical experiments in which the discrete predictions of a Linear discriminant analysis (LDA) classifier were cascaded with a Population Vector Algorithm (PVA) [Velliste et al., 2014] or a Gaussian Filter to predict zero velocity movements during idle state periods [Suway et al., 2013]. Similar offline studies were reported in ECoG-driven clinical and preclinical BCI [Choi et al., 2018] [P. T. Wang et al., 2013]. A LDA classifier was mixed with a PLS decoder to decode idle state, alternative or simultaneous bimanual movements from Non-Human Primate (NHP) [Choi et al., 2018]. The ECoG neural signals of able-bodied patients were decoded offline using the discrete predictions of a linear Bayesian classifier combined to a Kalman filter in order to inhibit the trajectory predictions during idle state [P. T. Wang et al., 2013]. Among the hybrid decoders, ME architecture was applied for several offline studies. EEG offline studies based on motor imagery control strategy decoded thanks to MLPs for both gating and experts decoders were reported in [Ebrahimpour et al., 2012] [Kheradpisheh et al., 2014]. MEA-based 3D reaching tasks was completed on monkeys using a linear filter and MLP [Kim et al., 2003] whereas finger movements decoding from subdural ECoG recordings was achieved thanks to SVM classifier and Simple Linear regression models [Elgharabawy and Wahed, 2016] [Flamary and Rakotomamonjy, 2012]. Reaching movement trajectory reconstruction in the 3D space was performed offline based on able-bodied epileptic patients' ECoG neural signals

using logistic regression (LR) gating coupled with Partial Least Square (PLS) experts [Bundy et al., 2016].

While many of motor BCI articles stressed the interest of dynamic modeling to take into account the temporal dependencies in the data to enhance the prediction performances [Saa et al., 2016] [Schaeffer and Aksenova, 2018] [P. T. Wang et al., 2013] [Williams et al., 2013] [Wu et al., 2004], the majority of the mentioned ME decoders applied in the BCI fields were static. To provide temporal information to the model, static ME decoders can be coupled with pre-processing as smoothed auto-regressive features [Flamary and Rakotomamonjy, 2012] or post-processing moving average [Hotson et al., 2016a]. Additionnaly, Dynamic models can be applied to integrate directly temporal dependencies in ME gating.

## 4.2. Sequential/Dynamic hybrid decoders

Dynamic gating of hybrid models or mixture of experts algorithm is not a broad area of research in the BCI field. Only few examples were applied to BCI preclinical and clinical experiments. Achtman [Achtman et al., 2007] designed a free-paced system which discriminated the idle, preparation and action states using the instantaneous state estimates and past classification to estimate the state transition whereas another classifier evaluated the reached target. The results of the study were obtained with offline MEA preclinical experiments analysis but the system was suitable to online application [Achtman et al., 2007]. A dynamic hybrid decoder using a pace-regression combined with a switching non-linear dynamic system was reported for offline finger trajectory reconstruction from ECoG signals of five epileptic patients [Wang et al., 2011]. Additionaly, a first-order Markovian transition probability was coupled to LDA classifier to discriminate active versus idle state during clinical offline finger trajectory reconstruction from ECoG signals [Hotson et al., 2016a].

HMM is a well know dynamic classifier used in many fields. This algorithm had already been integrated into hybrid decoder as dynamic gating algorithm and commonly used for asynchronous BCI system. Hybrid decoder coupling moving average models with HMMs was designed to detect idle versus active states for 3D arm trajectory offline reconstruction experiments from MEA signals of a monkey [Darmanjian et al., 2003]. A ReFIT-KF and a Hidden Markov Model (HMM) were coupled to decode 2D cursor trajectory during asynchronous online preclinical experiments with MEA recording system implanted on three rhesus macaques [Kao et al., 2017]. The ReFIT-KF controlled the velocity of the cursor while the HMM indicated whether or not to move the cursor (classification between idle and movement state) [Kao et al., 2017]. A dynamic gating ME referred as Hidden Markov Mixture of Experts (HMME) integrated a HMM gating model for sleep phases analysis based on offline EEG recordings [Liehr et al., 1999]. Finally, a new dynamic decoder was designed in [Schaeffer and Aksenova, 2016b] called

Markov Switching Linear Model (MSLM). MSLM is a particular ME which mixed continuous decoder and dynamic gating model based on HMM reducing false activations and deactivations by increasing current state stability [Schaeffer and Aksenova, 2016b]. Evaluated offline with subdural or epidural ECoG non-human primates recordings, MSLM outperformed other models with dynamic gating such as Switching Kalman Filter (SKF) and Markov post processed Wiener filters [Schaeffer and Aksenova, 2016b].

## 4.3. Markov Switching Linear Models

Among the algorithms based on ME architecture, dynamic gating is an extension which already provided promising results in motor BCI studies. M.C Schaeffer developed the Markov Switching Linear Model (MSLM) [Schaeffer, 2017] [Schaeffer and Aksenova, 2016b], a variant of the ME, which uses linear experts and a first-order Hidden Markov Model (HMM) for dynamic gating (Figure 4-1).

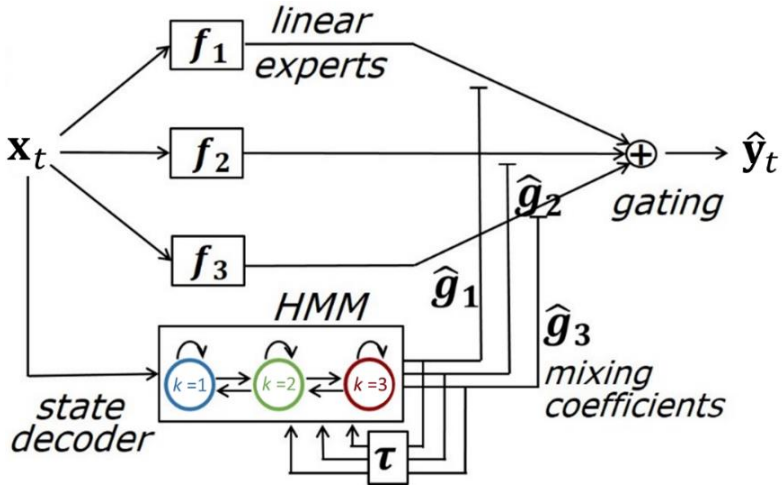



Figure 4-1: Markov Switching Linear Model (MSLM) structure [Schaeffer, 2017].

This method assumes that the expert selection (among the $K \in \mathbb{N}$ expert) is conditioned on unobserved discrete state $z_t$, which depends exclusively on the past state $z_{t-1}$, $p(z_t|z_{1:t}) = p(z_t|z_{t-1})$. The neural features $\mathbf{x}_t \in X$ are only dependent on the current state $p(\mathbf{x}_t|z_{1:t}) = p(\mathbf{x}_t|z_t)$ [Schaeffer, 2017] [Schaeffer and Aksenova, 2016b].

Thus, HMM state model is governed by the parameter bunch $\theta_g = \{\mathbf{A}, \{c_i\}_{k=1}^K, \boldsymbol{\pi}\}$, where $\mathbf{A}$ is the transition matrix, $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{K\times K}$, $a_{ij} = p(z_t = j|z_{t-1} = i)$; $\{c_k\}_{k=1}^K$ is the set of parameters characterizing the conditional distributions of the observed variables $p(\mathbf{x}_t|z_t)$, and $\boldsymbol{\pi} \in \mathbb{R}^K$ is the initial state probability vector at $t = 0$ [Bishop, 2006] [Rabiner, 1989].

$\mathbf{y}_t \in Y$ is estimated using gating coefficients $\gamma_{k,t}(\mathbf{x}_t) = p(z_t = k|\mathbf{x}_{1:t})$ and $k$ experts:

$$\hat{\mathbf{y}}_t = \sum_{k=1}^{K} \gamma_{k,t}(\mathbf{x}_t)\ (\mathbf{Beta}_k.\mathbf{x}_t + \mathbf{bias}_k),$$

where $\mathbf{Beta}_k \in \mathbb{R}^{m\times n}$ is the matrix of parameters of the $k^{th}$ linear expert and $\mathbf{bias}_k \in \mathbb{R}^n$ its associated bias. Thus, MSLM model is entirely defined by the gating $\theta_g$ and the set of experts $\theta_e = \{\theta_e^k\}_{k=1}^K = \{\mathbf{Beta}_k, \boldsymbol{bias}_k\}_{k=1}^K$ parameters: $\Theta = \{\theta_g, \theta_e\}$.

In [Schaeffer, 2017], the estimation of MSLM parameters was considered for fully supervised and unsupervised (according to the states sequence) training strategies. The PhD research was only focused on the fully supervised training.

The Maximum Likelihood (ML) estimate of MSLM parameters (experts and gate) using training data $\{\mathbf{X}, \mathbf{Y}, \mathbf{z}\}$ was considered [Schaeffer and Aksenova, 2016b]. Here, $\mathbf{X} \in \mathbb{R}^{L\times m}$, $\mathbf{Y} \in \mathbb{R}^{L\times n}$, $\mathbf{z} \in \mathbb{N}^L$ are the observation matrices of explanatory, response and state sequence variables respectively, $L$ is the number of samples.

The linear experts were independently trained with subsets of $\mathbf{X}$ and $\mathbf{Y}$ generated according to the state labels: expert k was trained with a cluster of training samples belonging to the state $z_t = k$. Gate parameters were estimated using $\mathbf{X}$ and $\mathbf{z}$ variables. The transition matrix $\mathbf{A}$ was computed by counting transition between $z_{t-1}$ and $z_t$.

In the BCI experiments, HMM emission probability $p(\mathbf{x}_t|z_t)$ was usually estimated using generative state decoders based on classical distributions, e.g. Gaussian models parameterized by an approximation of mean and covariance based on observation samples $\{c_k\}_{k=1}^K = \{\mu, \Sigma\}$ [Kao et al., 2017] [Pfeiffer et al., 2018] [Wissel et al., 2013]. Nevertheless, discriminative classifiers can also allow to estimating $p(z_t|\mathbf{x}_t)$ and infer the emission probability $p(\mathbf{x}_t|z_t)$ relying on the Bayes' theorem [Bishop, 2006]. This alternative highlighted relevant results in many applications (using e.g. support vector machine [Antelis et al., 2017] or logistic regression [Schaeffer and Aksenova, 2016b]).

After a comparative study, gate model was evaluated based on a discriminative classifier before to be integrated to HMM state decoder for gating emission probability estimation. Discriminative parameters of the classifier were estimated using PLS-based dimension reduction and Logistic Regression algorithm. Parameters of the experts were identified using Partial least Square (PLS) regressions. Hyperparameter optimization of PLS algorithm was performed using 6-fold cross-validation procedure and Wold's R criterion on the cross-validated PRESS statistic [Schaeffer, 2017].

MSLM was tested using asynchronous pre-clinical and clinical dataset based on ECoG recordings. MSLM performance was evaluated with 3D arm trajectory reconstruction of monkey arm and finger movements from epileptic patients. The MSLM results were compared to a Switching Kalman Filter and a Markovian post-processed Wiener Filter [Schaeffer, 2017]. MSLM outperformed both algorithms in term of state detection

accuracy and highlighted better trajectory reconstruction than the Switching Kalman Filter [Schaeffer, 2017] [Schaeffer and Aksenova, 2016b].

Even though MSLM comparative study presented relevant results in subdural and epidural ECoG signal decoding in preclinical (3D reaching tasks in non-human primates) and clinical (finger movements decoding) data demonstrating, in particular, strong idle state support, offline trajectory decoding was only reported. Additionally, the experts and gate algorithms as well as the cross-validation optimization method employed to evaluate the models were computationally demanding in term of required computation time and resources. Therefore, MSLM might not be suited for online adaptive close-loop decoding.

## 4.4. Recursive Exponentially Weighted N-way Partial Least Squares (REW-NPLS)

### 4.4.1. Partial Least Squares algorithm family

Due to the robustness in the computation of high dimensional data, algorithms of Partial Least Squares (PLS) family were frequently used in continuous and discrete BCI decoding. Numerous publications which reported offline ECoG-based hand trajectory decoding [Bundy et al., 2016] [Chao et al., 2010] [Chen et al., 2013] [Choi et al., 2018] [Eliseyev and Aksenova, 2014] [Schaeffer and Aksenova, 2016b], and EEG-based classification or cursor decoding [Maleki et al., 2018] [Trejo et al., 2006] sustained the interest of such algorithms.

The classical PLS regression algorithm is an offline procedure based on the iterative projection of input $\mathbf{x}_t \in \mathbb{R}^m$ and output $\mathbf{y}_t \in \mathbb{R}^n$ variables into latent variables spaces of dimension $f$ ($f$ is referred as the PLS "hyperparameter"). Projectors are estimated by maximizing the covariance between the input and the output latent variables [Wold et al., 1984]. Subspace dimension $f$ is typically determined through cross-validation.

Generally, iterative algorithms are not well adapted to large number of training samples requiring heavy computations. To overcome this limitation, Lindgren [Lindgren et al., 1993] and Dayal [Dayal and MacGregor, 1997a, 1997b] developed the kernel PLS algorithms. These methods, based on kernel matrices, allowed memory saving and faster computations in the estimation of the model parameters.

Conventional PLS [Wold et al., 1984] and kernel PLS [Dayal and MacGregor, 1997a] [Lindgren et al., 1993] are offline algorithms. Recursive PLS (RPLS) algorithms for online modeling of data flow were firstly introduced by Helland [Helland et al., 1992] and Qin [Qin, 1998, 1993]. These algorithms update the loading matrices of the previous model using online incoming data [Qin, 1998]. The Recursive Exponentially Weighted PLS

(REW-PLS), proposed by Dayal [Dayal and MacGregor, 1997b], is based on a more efficient kernel algorithm. Although these algorithms performed adaptive decoding, the hyperparameter $f$ still needed to be optimized offline in a preliminary study using cross-validation algorithm.

A generalization of the previously described conventional PLS algorithms to tensor data, N-way Partial Least Square (NPLS) algorithm, was proposed by Bro [Bro, 1998, 1996]. A tensor is a generalization of a matrix to higher order dimensions, also known as ways or modes. Vector and matrices are special cases of tensors with one and two modes respectively [Bro, 1998] [Kolda and Bader, 2009]. Tensor-based algorithms emerged as a promising strategy for brain signal processing. In the BCI field, the method allowed simultaneous treatments of high-dimensional data in the temporal, frequency and spatial domains [Cichocki et al., 2015] [Eliseyev et al., 2017]. NPLS algorithm projects input and output tensors into low dimensional space of latent variables using low rank tensor decomposition. This offline method improved the model stability and robustness compared to the classic unfold PLS leading to more accurate and interpretable predictions [Bro, 1998, 1996]. NPLS combines the robustness of PLS regression with the ability to preserve the structure of the data, which is lost in vector-oriented approaches [Eliseyev et al., 2017]. Additionally, the preservation of the data structure without unfolding optimizes the data information to provide a more robust estimate of the loading vector in the case of small training dataset [Bro, 1996].

Dimensional reduction strategies are solutions to handle high dimensional feature space. Embedded feature selection techniques were developed for NPLS algorithm. $L_1$-penalized NPLS algorithm generates more generalized and interpretable models by creating sparse model with the less relevant and noisy parameters fixed to zero. $L_1$-penalized NPLS sparse solution can improve the decoding performance and reduce the computational burden. This algorithm was used during offline preclinical BCI analysis [Eliseyev et al., 2012] [Foodeh et al., 2020] and in other domain [Hervás et al., 2019]. Nevertheless, the solution proposed in [Eliseyev et al., 2012] was not suited for online real-time model update.

For online modeling of tensor data flow, Recursive N-way PLS (RNPLS), a generalization of RPLS to tensor variables, was proposed [Eliseyev and Aksenova, 2013]. Similar to generic RPLS, RNPLS still required fixing the hyperparameter $f$ from offline preliminary study. Based on more computationally efficient kernel REW-PLS, REW-NPLS algorithm was proposed by Eliseyev [Eliseyev et al., 2017]. In addition to recursive online tensor-based linear regression identification, the Recursive-Validation (RV) procedure for online hyperparameter $f$ optimization was introduced. RV allowed the REW-NPLS algorithm to be a fully adaptive algorithm entirely tuned in real-time.

### 4.4.2. REW-NPLS algorithms

REW-NPLS algorithm adaptively update a set of $F$ ($F \in \mathbb{N}^*$ is the fixed upper bound latent space dimension) models $\left\{\underline{\mathbf{Beta}}_u^f, \underline{\mathbf{bias}}_u^f\right\}_{f=1}^F$ using the current block tensor of observation $\left\{\underline{\mathbf{X}}_u, \underline{\mathbf{Y}}_u\right\}$ and previously computed models weighted with the forgetting factor $\mu_1$. Here $u \in \mathbb{N}$ is the update iteration number, $\underline{\mathbf{Beta}}_u^f \in \mathbb{R}^{(I_1 \times \ldots \times I_M) \times (J_1 \times \ldots \times J_N)}$, $\underline{\mathbf{bias}}_u^f \in \mathbb{R}^{J_1 \times \ldots \times J_N}$ are the current update of models' coefficients, and $\underline{\mathbf{X}}_u \in \mathbb{R}^{\Delta L \times I_1 \times \ldots \times I_M}$, $\underline{\mathbf{Y}}_u \in \mathbb{R}^{\Delta L \times J_1 \times \ldots \times J_N}$ are the input and output $M \in \mathbb{N}$ and $N \in \mathbb{N}$ order tensors of observations with $\Delta L \in \mathbb{N}^*$ the number of samples recorded between the two update blocks $u-1$ and $u$.

REW-NPLS models are adaptively updated using covariance tensors $\underline{\mathbf{XX}}_u \in \mathbb{R}^{(I_1 \times \ldots \times I_M) \times (I_1 \times \ldots \times I_M)}$ and $\underline{\mathbf{XY}}_u \in \mathbb{R}^{(I_1 \times \ldots \times I_M) \times (J_1 \times \ldots \times J_N)}$ following the equations:

$$\underline{\mathbf{XX}}_u = \mu_1 \underline{\mathbf{XX}}_{u-1} + \underline{\mathbf{X}}_u \times_1 \underline{\mathbf{X}}_u,$$

$$\underline{\mathbf{XY}}_u = \mu_1 \underline{\mathbf{XY}}_{u-1} + \underline{\mathbf{X}}_u \times_1 \underline{\mathbf{Y}}_u,$$

Where "$\times_k$" is the k-mode tensor product and $\mu_1$ is a forgetting factor, and $u \in \mathbb{N}$ is the update iteration number.

PARAllel FACtor analysis (PARAFAC) tensor decomposition algorithm is used to extract a set of projectors $\left\{\mathbf{w}_f^1 \in \mathbb{R}^{I_1}, \ldots, \mathbf{w}_f^M \in \mathbb{R}^{I_M}\right\}_{f=1}^F$ from $\underline{\mathbf{XY}}_u$. The new $\left\{\underline{\mathbf{Beta}}_u^f\right\}_{f=1}^F$ are estimated based on these projectors and the couple scores/loadings from the models estimated with lower latent space dimensions $\{1, \ldots, f-1\}$ [Eliseyev et al., 2017]. Finally, the information decoded by $\underline{\mathbf{Beta}}_u^f$ are removed from $\underline{\mathbf{XY}}_u$ using deflation procedure to evaluate $\underline{\mathbf{Beta}}_u^{f+1}$ in the next loop (4-2). PARAFAC based tensor decomposition is a key element in the evaluation of the REW-NPLS models. This method is described in detail in the next sections. As a brief description, PARAFAC aims to decompose a tensor into a linear combination of vectors (the projectors) outer products.

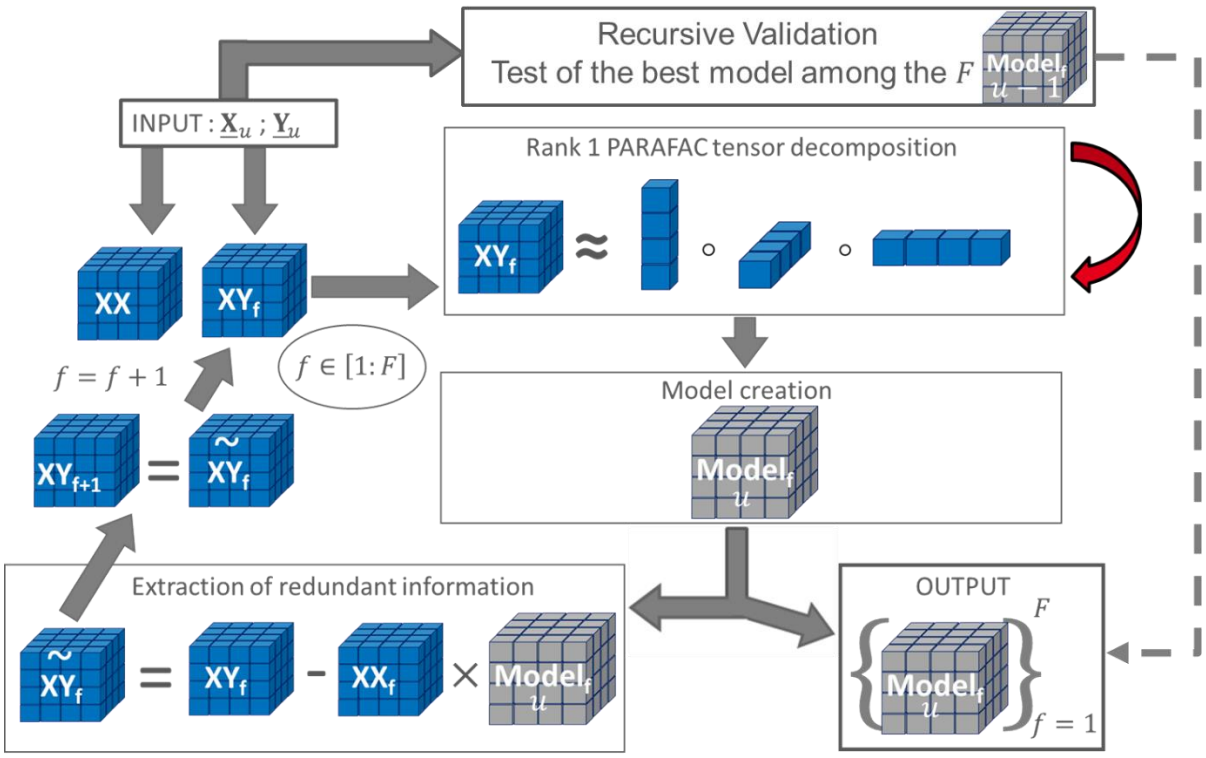



Figure 4-2: REW-NPLS main loop principles.

In parallel to parameters update, Recursive-Validation (RV) procedure estimates the optimal dimension of the space of latent variables based on incoming data and previously computed models [Eliseyev et al., 2017]. During the update, new observation tensors $\underline{\mathbf{X}}_u$ and $\underline{\mathbf{Y}}_u$ are available. The models previously computed $\left\{\underline{\mathbf{Beta}}_{u-1}^{f}, \underline{\mathbf{bias}}_{u-1}^{f}\right\}_{f=1}^{F}$ are used to compute a set of $F$ predictions $\left\{\underline{\hat{\mathbf{Y}}}_u^f\right\}_{f=1}^{F}$ to evaluate each model performance based on new incoming tensor: $e_u^f = \mu_2\, e_{u-1}^f + \varepsilon\left(\underline{\mathbf{Y}}_u, \underline{\hat{\mathbf{Y}}}_u^f\right)$, where $\mu_2$ is the forgetting coefficient ($\mu_2 \in [0; 1]$) and $\varepsilon$ is a function evaluating the accuracy of the prediction to maximise. The estimated optimal hyperparameter for current update is defined as $f_u^* = argmax_f\{e_u^f\}$. RV procedure is represented in Figure 4-3.

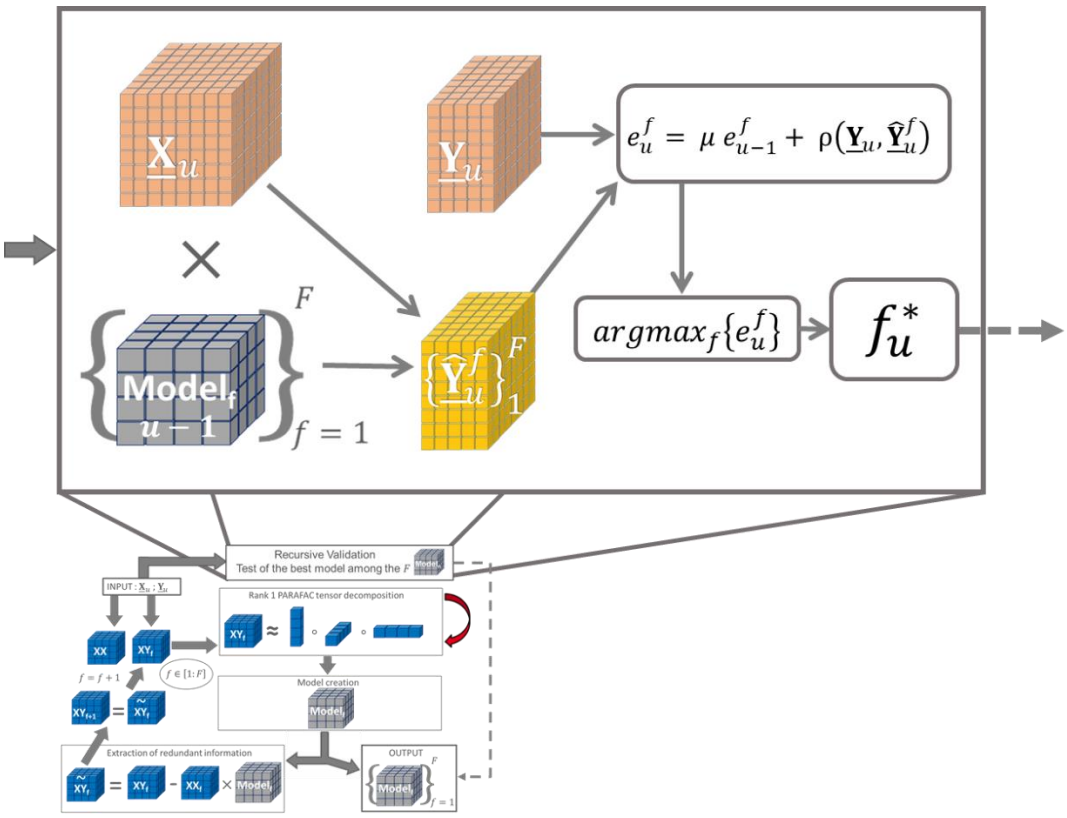



Figure 4-3: Recursive Validation procedure is performed to evaluate to optimal latent space dimension.

Recursive algorithms are a promising tool in motor BCI applications to integrate the subject's feedback directly in the model training phase [Brandman et al., 2018] [Orsborn et al., 2014] and improve the model robustness to the brain neural signal non-stationarity [Clerc et al., 2016b]. The consequences of such brain variability can lead to suboptimal offline-tuned models and thus lower decoding performances than expected during online decoding experiments compared to adaptive models [Jarosiewicz et al., 2013]. On the contrary, closed-loop decoders highlighted relevant improvements in trajectory decoding and robustness over time [Lebedev and Nicolelis, 2017] [Murphy et al., 2016] [Orsborn et al., 2014]. The REW-NPLS algorithm was designed for real-time adaptive incremental modelling and computation. The model showed relevant results in offline BCI trajectory decoding using epidural ECoG recordings from NHP and using MEG recordings from able-bodied subjects during finger tapping experiments [Eliseyev et al., 2017]. However, such regression model is not consistent to control complex or multi-limb effectors which required strong idle state support. In the case of bimanual, and even more, whole body effector, independent control of limbs with zero-velocity movement decoding of the non-activated limbs is mandatory. These tasks are not specially addressed by single model such as REW-NPLS.

## 4.5. Recursive Exponentially Weighted Markov Switching multi-Linear Model (REW-MSLM)

The Recursive Exponentially Weighted Markov Switching multi-Linear Model (REW-MSLM) is a piecewise linear model based on mixture of experts (ME) architecture. This new fully adaptive decoder was derived from two algorithms. On the one hand, the REW-MSLM has a hybrid Mixture of Expert (ME) structure similar to that of the Markov-Switching linear model (MSLM) [Schaeffer and Aksenova, 2016b], mixing discrete and continuous decoding. On the other hand, the REW-MSLM integrates the recursive exponentially weighted n-way partial least squares (REW-NPLS) [Eliseyev et al., 2017] for experts and gating model estimations. REW-NPLS is an online adaptive algorithm that handles tensor high-dimensional data flow and handle closed-loop decoder adaptation.

### 4.5.1. REW-MSLM description

The recursive exponentially weighted Markov-switching multi-linear model (REW-MSLM) is an online tensor-based fully adaptive mixture of multi-linear expert algorithm. The REW-MSLM inherits the Markov-switching linear model (MSLM) [Schaeffer and Aksenova, 2016a] mixture of experts (ME) structure, generalizing the MSLM model to the online incremental CLDA algorithm family, tensor-input-tensor-output variables and introduces the recursive model parameter identification procedure inspired by the REW-NPLS method [Eliseyev et al., 2017].

In our specific case of application, ME is an approach which mixes or switches independent decoders, called “experts”. Experts might be associated with a particular control tasks. For example, an independent expert decoder may be related to the control of specific movements of an effector, such as continuous left hand translation, right hand translation, left wrist or right wrist rotation. Such mixture of expert structure was implemented in the present study. Expert models, determined using the online adaptive REW-NPLS algorithm, are linearly mixed according to the gating model which estimates the probability of an expert to be active. This probability is used to compute gating coefficients to weight all the experts’ outputs. REW-MSLM output is the linear combination of all the weighted expert outputs. State equiprobability leads to the activation/mixing of all the experts, e.g. both hands translation and wrists rotation. Whereas if the probability for one limb is 1, the probability of the others is 0, this limb will be the only activated. The gating procedure applied in the article is referred as “soft gating” in contrast to “hard gating” which only selects the limb with the highest gating probability (limits the movement to one limb). In addition, fully adaptive REW-MSLM algorithm inherits the dynamic gating of the generic MSLM using hidden Markov model (HMM) for state sequence estimation in order to improve the decoder classification robustness (Figure 4-4).

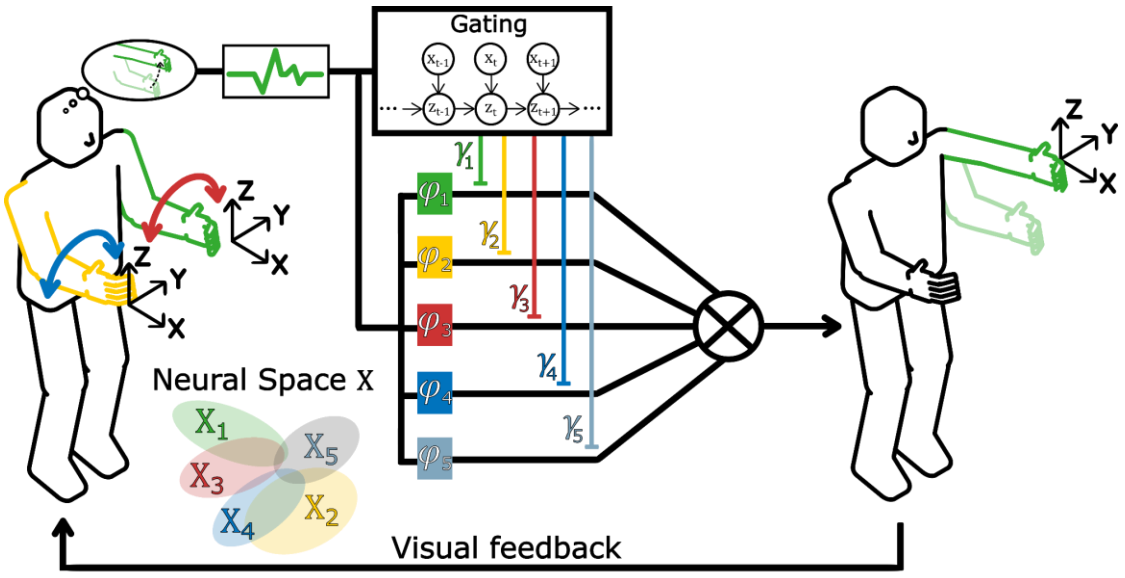



Figure 4-4: Recursive exponentially weighted Markov-switching linear model (REW-MSLM) architecture. The REW-MSLM includes a mixture of experts model, which can be described as the parallel computation of several predictions from different regression models (experts) that are weighted (enhanced or inhibited) according to the input variables using a classifier (gate). We hypothesize that the input feature space $X$ can be divided into several specific local regions $X_k$ and that each sub-space can be fitted using local multilinear functions $\varphi_k$ associated with an expert. Multilinear functions $\varphi_k$ are estimated using $k$ independent REW-NPLS models. The selected expert is determined based on the dynamic gating model. The gating model is a hidden Markov model (HMM) which computes the probability $\gamma_k$ for each expert to be activated. Commands are decoded by the REW-MSLM and sent to the effector to provide visual feedback to the patient.

The REW-NPLS discriminative decoder is also embedded into the HMM-based gating process to evaluate state probability. REW-NPLS was used because of its online adaptive characteristics and relevance for high dimensional input variable decoding. Moreover, discriminative decoders were selected instead of generative one due to their benefits for high dimensional and complex dependencies of features [Schaeffer, 2017] [Sutton and McCallum, 2012].

Basic assumption of ME approach is that each expert decodes his own specific region of feature space. Given $\underline{\mathbf{X}}_t \in \underline{X} \subset \mathbb{R}^{I_1 \times \ldots \times I_M}$ and $\underline{\mathbf{Y}}_t \in \underline{Y} \subset \mathbb{R}^{J_1 \times \ldots \times J_N}$ the independent and dependent $M$ and $N$ order tensor variables at time $t$ respectively. The feature space of independent variables is supposed to be partitioned into $K$ possibly overlapping regions $\underline{X} = \bigcup_{k=1}^{K} \underline{X}_k$. It is assumed that the space of input variables is mapped to the space of output variables using $K$ local multilinear functions $\Phi = \{\varphi_k : \underline{X}_k \rightarrow \underline{Y},\ k = 1,2,\ldots,K\}$. Let $z_t \in [1;K] \subset \mathbb{N}^*$ be a latent state variable defining the selected local multilinear function at time $t$ such as $\underline{\mathbf{Y}}_t = \varphi_{z_t}(\underline{\mathbf{X}}_t)$.

Dynamic gating is introduced using a first-order HMM [Schaeffer and Aksenova, 2016a]. Let $z_t$ be a latent state variable following the first-order Markovian assumption, which states that the dependence of $z_t$ is limited to the past state $z_{t-1}$. $\underline{\mathbf{Y}}_t$ is estimated as follows:

$$\widehat{\underline{\mathbf{Y}}}_t = \sum_{k=1}^{K} \gamma_{k,t} \left(\underline{\mathbf{Beta}}_k \underline{\mathbf{X}}_t + \underline{\mathbf{bias}}_k\right).$$

Here, $\underline{\mathbf{Beta}}_k \in \mathbb{R}^{I_1 \times \ldots \times I_M \times J_1 \times \ldots \times J_N}$ and $\underline{\mathbf{bias}}_k \in \mathbb{R}^{J_1 \times \ldots \times J_N}$ are the $k^{th}$ expert tensor parameters and its associated bias. $\gamma_{k,t} = p\left(z_t = k \middle| \underline{\mathbf{X}}_{1:t}\right)$ is the dynamic gating weight coefficient associated with the $k^{th}$ expert at time $t$. REW-MSLM sub-models are entirely defined through the experts' parameters $\theta_e = \{\theta_e^k\}_{k=1}^K = \left\{\underline{\mathbf{Beta}}_k, \underline{\mathbf{bias}}_k\right\}_{k=1}^K$ and HMM parameters $\theta_g = \left\{\mathbf{A}, \{c_k\}_{k=1}^K, \boldsymbol{\pi}\right\}$, where $\mathbf{A}$ is the transition matrix, $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{K \times K}$, $a_{ij} = p(z_t = j | z_{t-1} = i)$, $\{c_k\}_{k=1}^K$ is the set of parameters employed to estimate the conditional emission probability of the observed variables $p\left(\underline{\mathbf{X}}_t \middle| z_t\right)$, and $\boldsymbol{\pi} \in \mathbb{R}^K$ is the initial state probability vector at $t = 0$.

## 4.5.2. REW-MSLM online incremental training

The proposed REW-MSLM algorithm recursively estimates $\boldsymbol{\Theta} = \{\theta_g, \theta_e\}$ with a supervised training procedure. At each update $u$, the corresponding batch of training dataset $\left\{\underline{\mathbf{X}}_u, \underline{\mathbf{Y}}_u, \mathbf{z}_u\right\}$ is given with $\underline{\mathbf{X}}_u \in \mathbb{R}^{\Delta L \times I_1 \times \ldots \times I_M}$, $\mathbf{Y}_u \in \mathbb{R}^{\Delta L \times J_1 \times \ldots \times J_N}$, $\mathbf{z}_u = \left(z_{t_1}, \ldots, z_{t_1 + \Delta L}\right)^T \subset \mathbb{N}^{*\Delta L}$ and $\Delta L$ the update block size.

The $K$ local multilinear functions $\varphi_k$ mapping the input variable space to the response variable space are estimated using expert specific samples. The $k^{th}$ expert model parameter update is performed on the training dataset $\left\{\underline{\mathbf{X}}_u^k, \underline{\mathbf{Y}}_u^k\right\}$. $\underline{\mathbf{X}}_u^k$ and $\underline{\mathbf{Y}}_u^k$ are sub-tensors of $\underline{\mathbf{X}}_u$ and $\underline{\mathbf{Y}}_u$ formed by samples labelled as belonging to state $k$. The $k^{th}$ expert model parameters are updated using REW-NPLS algorithm REW-NPLS$_e$ = REW-NPLS$\left(\underline{\mathbf{X}}_u^k, \underline{\mathbf{Y}}_u^k\right)$ with the forgetting factor $\mu_k$, $0 \leq \mu_k \leq 1$.

For online optimization latent variable space dimension (hyperparameter $f$), the REW-NPLS$_e$ algorithm estimates a set of $F$ models for each expert $\left\{\underline{\mathbf{Beta}}_{u,k}^f, \underline{\mathbf{bias}}_{u,k}^f\right\}_{k,f=1}^{K,F}$. $F \in \mathbb{N}^*$ is the fixed highest latent space dimension. The optimal hyperparameter of the $k^{th}$ expert $f_k^* \leq F$ is selected following the Recursive-Validation procedure [Eliseyev et al., 2017]. Recursive-Validation exploits the newly available dataset $\left\{\underline{\mathbf{X}}_u^k, \underline{\mathbf{Y}}_u^k\right\}$ as testing data for the currently available models to evaluate the best hyperperparameters before to use them as training dataset for the models updating procedure. The best models are chosen independently for each expert and are used for real-time decoding of the neural signals: $\left\{\underline{\mathbf{Beta}}_k, \underline{\mathbf{bias}}_k\right\}_{k=1}^K = \left\{\underline{\mathbf{Beta}}_{u,k}^{f_k^*}, \underline{\mathbf{bias}}_{u,k}^{f_k^*}\right\}_{k=1}^K$.

Similarly, at each update $u$, the HMM gating parameters estimation is updated based on the update block dataset $\left\{\underline{\mathbf{X}}_u, \mathbf{z}_u\right\}$. The HMM transition matrix $\mathbf{A}$ is approximated by

counting the successive transition of states in $\mathbf{z}_u$ and the transition matrix estimated during the previous updates weighted with the forgetting factor $\mu_g$ , $0 \leq \mu_g \leq 1$. HMM conditional emission probabilities $p(\underline{\mathbf{X}}_t|z_t)$ is inferred through the combination of $p(z_t|\underline{\mathbf{X}}_t)$ and class prior $p(z_t)$ using Bayes' theorem [Bishop, 2006]. The REW-NPLS discriminative decoder is embedded into the HMM-based gating process to evaluate $p(z_t|\underline{\mathbf{X}}_{1:t})$. It is trained on the observation tensor of input variables $\underline{\mathbf{X}}_u$ and the latent state dummy variable matrix $\mathbf{Z}_u \in \{0,1\}^{K\times\Delta L}$ where the column-wise (single) non-zero element depicts the activated state for each sample.

The REW-NPLS discriminative algorithm computes a set of $F$ multilinear models $\{\underline{\mathbf{B}}_u^f, \mathbf{b}_u^f\}_{f=1}^F$, where $\underline{\mathbf{B}}_u^f \in \mathbb{R}^{I_1\times\ldots\times I_M\times K}$ and $\mathbf{b}_u^f \in \mathbb{R}^K$ are the gating tensor of the parameters and its related bias. The Recursive-Validation procedure selects the best model based on the estimated gating optimal hyperparameter $f_g^* \leq F$ and defines the optimal gating model as $\{\underline{\mathbf{B}}, \mathbf{b}\} = \{\underline{\mathbf{B}}_u^{f_g^*}, \mathbf{b}_u^{f_g^*}\}$ for dynamic gating weight $\gamma_{k,t}$ estimation. The output variable $\hat{\boldsymbol{z}}_t \in \mathbb{R}^K$ determines how likely each hidden state is generated based on $\underline{\mathbf{X}}_t$. From the REW-NPLS discriminative decoder prediction $\hat{\boldsymbol{z}}_t$, $p(z_t|\underline{\mathbf{X}}_t)$ is evaluated with the softmax function before to compute $\gamma_{k,t} = p(z_t = k|\underline{\mathbf{X}}_{1:t})$ using HMM forward algorithm.

As mentioned, REW-MSLM is using dynamic HMM gating. The equivalent mixture of expert algorithm using static gating (without HMM) is referred as REW-SLM in the manuscript. REW-SLM is used to highlight the performance enhancement related the implementation of HMM (further details on the comparative study between REW-MSLM and REW-SLM are available in the Chapter 9). REW-SLM gating is computed with the REW-NPLS algorithm calibrated on the explanatory variables and the latent states followed by the softmax function $p(z_t|\underline{\mathbf{X}}_t)$. In the static REW-SLM classifier, the HMM forward algorithm is not applied. In the opposite, the REW-MSLM algorithm, in order to integrate the Markov process for the prediction of the discrete state, applies the forward algorithm.

### 4.5.3. Decoder application

The proposed REW-MSLM algorithm is used in closed-loop multi-limb experiments. New input data $\underline{\mathbf{X}}_t$ are recorded, and each expert model is applied based on the decoding model $\{\underline{\mathbf{Beta}}_k, \underline{\mathbf{bias}}_k\}_{k=1}^K$. Finally, the dynamic gating coefficients $\gamma_{k,t}$ are estimated using the latent state variable estimator $\hat{\boldsymbol{z}}_t$ (equation (4.1)) post-processed with a softmax function (equation (4.2)) [Yuksel et al., 2012] (equivalent to REW-SLM classifier output) followed with the HMM forward algorithm [Rabiner, 1989] (equations (4.3) and (4.4)). The forward algorithm evaluates $\gamma_{k,t}$ by considering the past and current observations:

$$\hat{\boldsymbol{z}}_t = \underline{\mathbf{B}}\,\underline{\mathbf{X}}_t + \mathbf{b}, \tag{4.1}$$

$$p(z_t = k|\underline{\mathbf{X}}_t) = \frac{\exp(\hat{z}_{k,t})}{\sum_{i=1}^{K} \exp(\hat{z}_{i,t})}, \quad (4.2)$$

$$p(z_t = k, \underline{\mathbf{X}}_{1:t}) = p(\underline{\mathbf{X}}_t|z_t = k) \sum_{j=1}^{K} a_{kj}\, \gamma_{k,t-1}, \quad (4.3)$$

$$\gamma_{k,t} = p(z_t = k|\underline{\mathbf{X}}_{1:t}) = \frac{p(z_t = k, \underline{\mathbf{X}}_{1:t})}{\sum_{j=1}^{K} p(z_t = k, \underline{\mathbf{X}}_{1:t})}\ . \quad (4.4)$$

## 4.6. Conclusion

In this chapter, a new decoder is introduced for online adaptive decoding of multi-limb effector control from ECoG neuronal activity recording. The REW-MSLM conjugates the benefits of generic REW-NPLS and MSLM algorithms. On the one hand, REW-MSLM uses the MSLM mixture of expert structure to control complex effector such as exoskeleton or it avatar. In the case of multi-limb effector, an expert model can be associated to each limb allowing to control several limbs and complete complex actions with a single REW-MSLM decoder. A dynamic gating is introduced to improve expert classification as well as strong idle state discrimination. On the other hand, the experts and gate models are calibrated with several independent REW-NPLS algorithms to incrementally update the REW-MSLM during online closed-loop experiments.

REW-MSLM online adaptive multi-limb decoder was designed to integrate quickly the patient's feedback neural activity in a "human-in-loop" calibration procedure to create an adaptive closed-loop decoder where the patient and model fits together.

Chapter 5

# Online incremental group-wise sparse REW-MSLM

## Contents

BCI experiments highlighted high intra and inter-subjects variability in the BCI decoders. Although BCI model is generally relying on neurological markers generalizable on the majority of subjects [Clerc et al., 2016b], it requires to generate a wide range of neural features to include possible neurophysiological patterns and select among them the most relevant subject-specific features. Thus, high dimensional feature space is commonly used in BCI to decode patient's action from the brain neural signals.

However, the processing of noisy and high dimensional features, such as brain signals, brings several challenges to overcome. They can be grouped as: the model calibration issue, the model generalization and interpretation problems and the hardware related obstacles [Bellman, 1961] [Bishop, 2006] [Nicolas-Alonso and Gomez-Gil, 2012] [Remeseiro and Bolon-Canedo, 2019]. Firstly, with the exception of algorithms specifically oriented to this problematic, in machine learning, higher dimensional models require more training data set. Nevertheless, real-time BCI experiments are performed during rare and brief sessions due to the reduced ability of disabled patients to remain focused in prolonged calibration sequences [Brandman et al., 2018]. Therefore, generally, calibration sessions are too short for complex high dimensional parameter identification and may lead to the classical "curse of dimensionality" problem related to uninformative or correlated features and small training dataset compared to feature space dimension. Additionally, high dimensional feature spaces and related models with high number of parameters are more complicated to interpret than low dimensional one. Finally, high dimensional feature space computation and high dimensional model evaluation require high computational power and time for neural signal processing, model calibration and application. These hardware considerations are key characteristics in the case of real-time embedded/portable BCI application which have limited computational resources.

In this chapter, two online adaptive group-wise sparse decoders are presented to reduce the feature space dimension employed for BCI decoding and improve the model interpretability. The proposed decoders were designed to be integrated in the REW-MSLM algorithms as sparse gating and/or experts models and create BCI systems with low computational cost, suited for portable applications. Firstly a brief review on the different feature dimension reduction strategies is introduced, then, in order to understand the proposed algorithms, the PARAFAC procedure employed in the REW-NPLS algorithm is detailed. Next, the new $L_p$-Penalized REW-NPLS algorithm (PREW-NPLS) for online sparse model identification is proposed. Finally, an extension of the PREW-NPLS algorithm, including online automatic regularization parameter tuning named Automatic $L_p$-Penalized REW-NPLS (APREW-NPLS) is presented.

## 5.1. Context related feature dimension reduction

High dimensional feature space may lead to numerous issues such as high computing power requirements, high computing time, and "curse of dimensionality" problem in

the decoder training and application steps [Bellman, 1961] [Bishop, 2006] [Nicolas-Alonso and Gomez-Gil, 2012] [Remeseiro and Bolon-Canedo, 2019]. Furthermore, high dimension feature space generally presents useless features with irrelevant and/or redundant information which disrupt the model training and application. The presented high dimensional feature space limitations are recurrent complications in the BCI field, especially in the case of real-time BCI experiments.

To prevent these issues, dimensional reduction algorithms decreasing the feature space dimension were employed to create the BCI model. Reduction of the feature space dimension may improve the decoding performance and drastically reduce the computing time and resources required. In the case of daily life BCI applications with high dimensional data flow processing, computing time and resources management is a crucial aspect to consider [Haufe et al., 2014]. Dimensional reduction algorithms are dissociated into projection and feature selection algorithm families. Both dimensional reduction algorithm types were applied to online BCI experiments and offline studies.

**Projections algorithms** were often used in BCI studies [Bousseta et al., 2018] [Bundy et al., 2016] [Choi et al., 2018] [Eliseyev et al., 2017] [Haufe et al., 2014] [Hsu et al., 2016] [Jafarifarmand and Badamchizadeh, 2020] [Jiang et al., 2017] [Khan et al., 2019] [S. P. Kim et al., 2006] [Lotte et al., 2018] [Marathe and Taylor, 2013] [Palmer and Hirata, 2018] [Sannelli et al., 2016] [Schaeffer and Aksenova, 2016b] [Seifzadeh et al., 2017] [Sreenath and Ramana, 2017]. They project the feature space into a subspace of lower dimension by linear or non-linear combination of the initial feature space components (more details are available in the Chapter 3). This family clusters the principal and independent component analysis (PCA and ICA), spatio-spectral decomposition (SSD), common spatial pattern (CSP) or partial least squares (PLS) [Bousseta et al., 2018] [Bundy et al., 2016] [Choi et al., 2018] [Eliseyev et al., 2017] [Haufe et al., 2014] [Hsu et al., 2016] [Jafarifarmand and Badamchizadeh, 2020] [Jiang et al., 2017] [Khan et al., 2019] [S. P. Kim et al., 2006] [Lotte et al., 2018] [Marathe and Taylor, 2013] [Palmer and Hirata, 2018] [Sannelli et al., 2016] [Schaeffer and Aksenova, 2016b] [Seifzadeh et al., 2017] [Sreenath and Ramana, 2017], etc. However, such methods may not improve the computing time as they does not optimize feature extraction step. The irrelevant features are still computed.

**Feature selection family** regroups filter-based, wrapper-based and embedded techniques [Bolón-Canedo et al., 2013] [Khaire and Dhanalakshmi, 2019]. Filter-based methods rank and select independently the features which cluster the most information without consideration of the trained decoder. This method is effective in computation time and have good generalization capacity. However, these methods tend to select highly correlated (redundant) features.

In the opposite, wrapper-based techniques incorporate supervised learning algorithms to evaluate the possible interactions between the features. Wrapper methods add iteratively new features to the subset of selected features space and evaluate the

performance of the selected subset combined with the trained decoder [Lotte et al., 2018]. These methods are efficient, nevertheless, they are costly in terms of computing time and may lead to overfitting.

Embedded techniques regroups the strategies were the feature selection steps is directly integrated into the decoding algorithm to combined the benefits of both previous methods: keeping the advantages of wrapper while decreasing computational complexity [Khaire and Dhanalakshmi, 2019]. Features selection is performed directly within the model learning process. For example, BCI embedded techniques regroups decision tree, and regularization algorithms. Regularization strategies add penalty on the model parameter optimization to reduce the freedom of the model. Numerous regularization are named depending on the applied penalization norm/pseudo-norm: $L_0$, $L_1$ (Lasso), $L_2$ (Ridge) or elastic net regularization algorithms etc. The $L_1$ regularization adds a penalty term equal to the sum of the absolute value of the coefficients whereas $L_2$ regularization integrates a penalty equal to the sum of the squared value of the coefficients and elastic net regularization is defined as the combination of both $L_1$ and $L_2$ penalization [Bishop, 2006]. $L_p$ regularization algorithms with $0 \leq p \leq 1$ discard irrelevant features promoting sparse solution [Bishop, 2006] [Hastie et al., 2015]. Sparse solution is efficient to avoid overfitting and may lead to reduction in computing time.

Regularization algorithms were commonly applied in the BCI field for feature selection or to improve neural signal decoding such as $L_0$ [Sreeja et al., 2019], $L_1$ [Eliseyev et al., 2012] [Flamary and Rakotomamonjy, 2012] [López-Larraz et al., 2014] [Lotte and Guan, 2011] [Y. Zhang et al., 2013], $L_2$ [Cincotti et al., 2008] [Flamary and Rakotomamonjy, 2012] [Nagel and Spüler, 2019] [Seifzadeh et al., 2017], elastic net [Kim et al., 2018] [Peterson et al., 2019] norm penalization or other regularization strategies such as regularization algorithms using, polynomial regression [Eliseyev and Aksenova, 2016], sparse regularization based on automatic relevance determination (ARD) parameters [Nakanishi et al., 2017] [Toda et al., 2011], Kullback-Leibler regularization in the Riemannian mean [Mishra et al., 2018], etc.

Generally, regularization algorithms is performed in single-wise manner, they evaluate independently the contribution of each model parameter before to apply a constrain in order to regulate the amplitude of each parameter weight. Each feature is regularized independently and is not evaluated as belonging to a group of features to be penalized. Therefore, in the case of tensor input features, each tensor component is set to zero independently to each other. Such element-wise component regularization of tensor features may lead to more complicated interpretation of the results and extraction of the relevant features (Figure 5-1A). However, there are many applications with structurally grouped input features where it may be of interest to set simultaneously to zero or non-zero value the features within a pre-determined group [Hastie et al., 2015].

Group-wise regularization performs selection by grouping the relevant features and applying the penalization to the groups of features at once (Figure 5-1B) [Eliseyev et al.,

2012] [Giordani and Rocci, 2013] [Hastie et al., 2015] [Martínez-Montes et al., 2008] [Y. Zhang et al., 2013]. Grouping can cluster features over different modalities such as the electrodes, the frequency bands [van Gerven et al., 2009], etc. Group-wise sparse regularization promotes the model convergence to sparse solution (in a group-wise level), simplifies the model interpretation and is suited to naturally structured features. Moreover, group-wise selection discards group of variables from the signal processing workflow (electrode or frequency) reducing the computational cost and the required computing time for real-time applications. Group-wise penalization was rarely applied to the BCI field [Eliseyev et al., 2012] [Motrenko and Strijov, 2018] [van Gerven et al., 2009] [Wu et al., 2019]. Regularized PARAFAC and Tucker decomposition are two algorithms designed for group-wise tensor penalization. In these algorithms, tensors are expressed as a linear combination of vectors which are independently regularized. Regularized tensor decomposition leads to a slice-wise tensor penalization creating more easily interpretable solution than element-wise regularization strategy (Figure 5-1). These approaches were used in few offline BCI studies [Eliseyev et al., 2012] [Martínez-Montes et al., 2008] and in other research fields [Giordani and Rocci, 2013] [Hervás et al., 2019] [Kim et al., 2014, 2013].

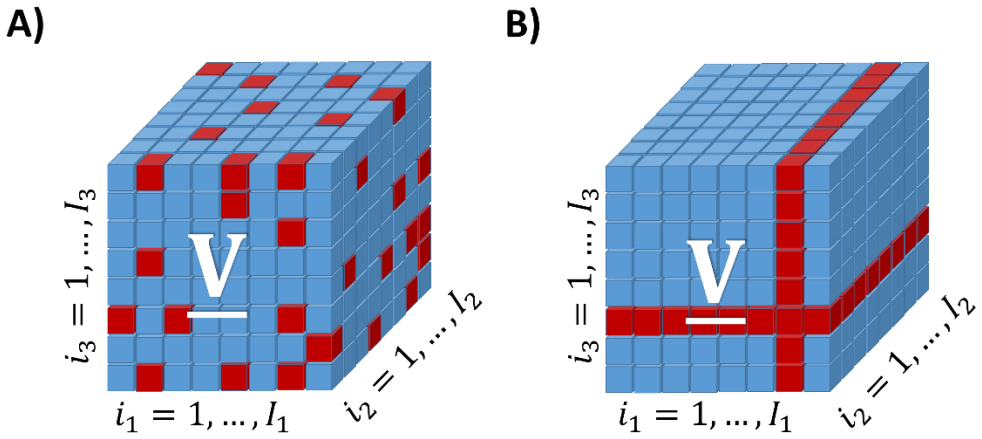



Figure 5-1: Difference between element-wise and group-wise regularization. Example of third order tensor sparse evaluation using element-wise regularization (A) and group-wise regularization (B).

In BCI studies, most of the presented feature dimensional reduction algorithms were tested during offline studies [Bundy et al., 2016] [Choi et al., 2018] [Eliseyev et al., 2012] [Flamary and Rakotomamonjy, 2012] [Garrett et al., 2003] [Jafarifarmand and Badamchizadeh, 2020] [Jiang et al., 2017] [Khan et al., 2019] [S. P. Kim et al., 2006] [Kumar et al., 2017] [López-Larraz et al., 2014] [Lotte and Guan, 2011] [Mishra et al., 2018] [Motrenko and Strijov, 2018] [Nakanishi et al., 2017] [Oliver et al., 2013] [Palmer and Hirata, 2018] [Peterson et al., 2019] [Robinson et al., 2013] [Schaeffer and Aksenova, 2016b] [Schroder et al., 2003] [Seifzadeh et al., 2017] [Sreeja et al., 2019] [Toda et al., 2011] [van Gerven et al., 2009] [Wu et al., 2019] [Y. Zhang

et al., 2013]. Nevertheless, some of them were applied in online applications. Generally, feature selection was performed in offline preliminary studies before to apply the set of selected feature during online clinical or preclinical BCI experiments [Bousseta et al., 2018] [Brunner et al., 2006] [Cantillo-Negrete et al., 2018] [Huang et al., 2009] [Kim et al., 2018] [Marathe and Taylor, 2013] [Nagel and Spüler, 2019] [Spüler et al., 2012b].

Online adaptive dimensional reduction strategies have several advantages for online adaptive BCI. Majority of decoders trained in real-time are sensitive to overfitting due to the lack of training data. Moreover, reduced feature space dimensions may reduce the required computing resources to apply the model in real-time with faster data flow analysis.

The adaptive dimensionality reduction algorithms commonly applied in the BCI [Ang et al., 2011] [Chen and Fang, 2017] [Hsu et al., 2016] [Lotte et al., 2018] [Mobaien and Boostani, 2016] [Sannelli et al., 2016] [Song and Yoon, 2015] [Woehrle et al., 2015] [Zhao et al., 2008] were based on projection strategies such as adaptive CSP, PCA, ICA or xDAWN algorithms. However, all of them were only tested during offline studies. Few of the dimensional reduction algorithms [Vidaurre et al., 2006b] were integrated into a BCI software made of an adaptive dimensional reduction procedure and an adaptive classifier/regression decoder.

Few adaptive feature selection algorithms were applied in the motor BCI field during online experiments. Filter methods were tested on BCI simulation using Mutual Information [Oliver et al., 2013] or during online BCI experiments based on Fisher score [Faller et al., 2012]. Wrapper strategy was optimized using parallel computation for online BCI classifier [Mend and Kullmann, 2012] whereas embedded methods using semi-supervised feature selection [Long et al., 2011] and weighting features algorithm [Andreu-Perez et al., 2018] were designed and used during online BCI applications. Adaptive Genetic Algorithm was proposed for adaptive channel selection in [Moro et al., 2017]. Nevertheless, all these algorithms were applied to simple online binary classification BCI experiments.

Regularized algorithms trained offline were applied during online BCI experiments in [Cincotti et al., 2008] [Ma et al., 2020] [Shin et al., 2015]. Adaptive regularized algorithms with fixed penalization hyperaparemeter were tested using offline dataset but none of these algorithms have been applied to real-time BCI experiments [Mishra et al., 2018] [Roijendijk et al., 2016] [Sharghian et al., 2019]. Adaptive algorithm with a $L_1$-norm regularization strategy were reported in other domains with an adaptive logistic regression [Sheikhattar et al., 2015], a Kernel least squares [Yang et al., 2019] and a recursive least squares algorithms [B. Chen et al., 2012].

Only few dimensional reduction methods were integrated into adaptive algorithms for online incremental calibration during real-time BCI experiments and were generally restricted to EEG-based experiments [Andreu-Perez et al., 2018] [Faller et al., 2012] [Long et

al., 2011] [Mend and Kullmann, 2012] [Moro et al., 2017]. Computational complexity and difficulty to integrate dimensional reduction methods into real-time algorithms may explain the lack of proposed solutions. Moreover, dimensional reduction methods often rely on hyperparameters which required to be tuned to optimize the decoding performances. This hyperparameter optimization problem may be another explanation of the lack of regularized adaptive decoder in the BCI fields.

In motor BCI field, the $L_1$-Regularized N-way PLS algorithm developed by Eliseyev [Eliseyev et al., 2012] and the Regularized PLS proposed by Foodeh [Foodeh et al., 2020] outperformed their non-penalized version thanks to the suppression of noisy/non-relevant electrodes. However, these algorithms were not adapted to online adaptive decoding, required preliminary studies to fixed the hyperparameters and were only tested offline on NHP using ECoG [Eliseyev et al., 2012] [Foodeh et al., 2020] and rats using LFP [Foodeh et al., 2020].

In the next section, the new $L_p$-Penalized REW-NPLS (PREW-NPLS) is proposed. PREW-NPLS is a new regularized recursive exponentially weighted N-way PLS designed for online adaptive decoding promoting group-wise (slice-wise) sparsity generalized to $L_0$, $L_{0,5}$ and $L_1$ norm regularization. PREW-NPLS rely on the REW-NPLS algorithms. The crucial REW-NPLS tensor decomposition procedure inspired from PARAFAC algorithm is modified to estimate a sparse $L_0$, $L_{0,5}$ or $L_1$ PARAFAC tensor decomposition. PREW-NPLS is an incremental adaptive regression algorithm which incrementally estimates a sparse $L_0$, $L_{0,5}$ and $L_1$ solution with a fixed penalization hyperparameter. As mentioned, hyperparameter generally requires preliminary offline studies to be optimized which is counterintuitive for algorithm dedicated to incremental online closed-loop calibration. Therefore, a new upgraded version referred as Automatic $L_p$-Penalized REW-NPLS (APREW-NPLS) is introduced in order to automatically optimize the penalization hyperparameter during the online incremental calibration of the model using a reinforcement learning strategy. The PREW-NPLS and APREW-NPLS algorithms were designed to be integrated in the REW-MSLM algorithms as sparse gating and/or experts models for portable applications. Both algorithms are described in the next sections but firstly require to introduce in details the PARAFAC decomposition procedure.

## 5.2. PARAFAC procedure

REW-NPLS algorithm evaluates a set of projectors from the covariance matrix $\underline{\mathbf{XY}}_u$ using a rank one decomposition to evaluate the model parameters. Several tensor decomposition strategies were designed such as the PARAFAC, Tucker and multilinear SVD decomposition. The tensor decomposition employed in REW-NPLS algorithm is based on PARAllel FACtor analysis (PARAFAC) tensor decomposition procedure. It is described in further detail in the next section.

Parallel Factor analysis (PARAFAC) or CANDECOMP/PARAFAC (CP) also known as polyadic decomposition (PD) can be considered as the generalization of principal component analysis (PCA) and singular value decomposition (SVD) to the tensor case [Cichocki et al., 2015] [Kolda and Bader, 2009]. This method represents a $M$-order tensor $\underline{\mathbf{V}} \in \mathbb{R}^{I_1 \times \ldots \times I_M}$ as the linear combination of vector outer products (rank-one tensors) such as :

$$\underline{\mathbf{V}} = \sum_{r=1}^{R} \rho_r \mathbf{w}_r^1 \circ \mathbf{w}_r^2 \circ \ldots \circ \mathbf{w}_r^M + \underline{\mathbf{E}}, \quad \text{with } r, m : \|\mathbf{w}_r^m\| = 1.$$

Here, $1 \leq m \leq M$ corresponds to the m[th] mode/dimension of the tensor variable, "$\circ$" is the (vector) outer product of the decomposition projectors $\mathbf{w}_r^m \in \mathbb{R}^{I_m}$, $R \in \mathbb{N}$ is the fixed number of rank-one tensors used to decompose the original tensor variable, $\rho_r$ is the weight associated to each rank-one tensor of the decomposition and $\underline{\mathbf{E}} \in \mathbb{R}^{I_1 \times \ldots \times I_M}$ is the tensor of residuals [Kolda and Bader, 2009]. An example of three-order tensor decomposition based on the linear combination of $R$ outer products of three vectors is showed in the Figure 5-2.

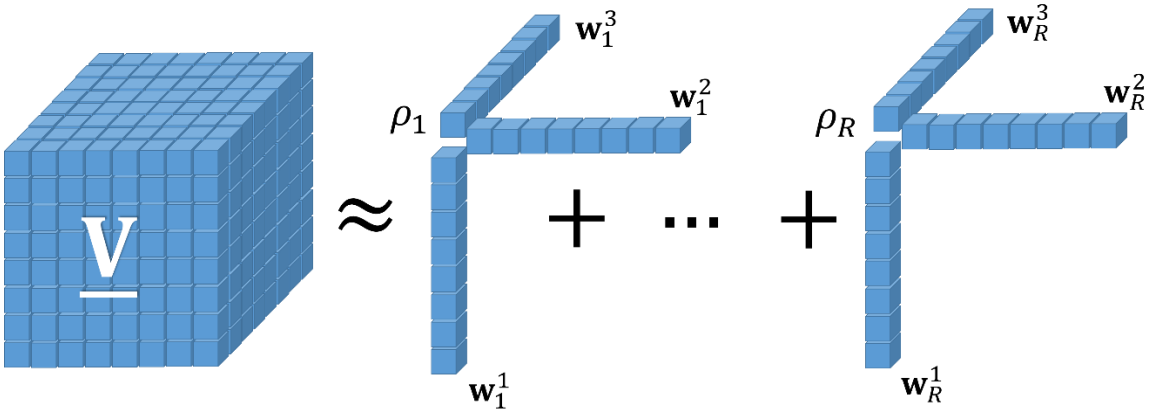



Figure 5-2 Example of a PARAFAC decomposition of a 3-order tensor. The tensor is decomposed in linear combination of $R$ vectors outer products.

## 5.2.1. PARAFAC decomposition computation

Tensor decomposition is an appealing tool since the last twenty years in various fields (audio, image, video processing, biomedical applications, etc.) due to the rising of high dimensional data [Cichocki et al., 2015]. Nevertheless, no specific algorithm determining the rank of tensor has been defined [Kolda and Bader, 2009]. Consequently, the number of rank-one tensor decomposition $R$ is set to a sub-optimal value [Kolda and Bader, 2009]. Fixing $R$ leads to solve a low-rank approximation problem which is an ill-posed problem [Pereira Da Silva et al., 2015]. Numerous algorithms has been designed to locally solve this problem.

Most of the solutions can be grouped into two families: direct methods regrouping Alternating least square (ALS), direct tri-linear decomposition (DTLD) and iterative non-least square methods such as self-weighted alternating tri-linear decomposition (SWATLD) or alternating slice-wise diagonalization (ASD). Hessian and gradient based methods regroup Newton-based algorithms, damped Gauss-Newton with compression (dGNc), positive matrix factorization for 3-way arrays (PMF3) and high-order singular value decomposition (HOSVD) [Kolda and Bader, 2009] [Tomasi, 2006]. No agreements on the best solution has been found on the literature but ALS seems to generally leads to good quality decomposition even though it is slower than numerous algorithms such as ASD [Faber et al., 2003] [Tomasi, 2006] [Tomasi and Bro, 2006].

Alternating least square (ALS) method is the most popular algorithm for PARAFAC decomposition [Faber et al., 2003] [Kolda and Bader, 2009] due to the ease of implementation. Nevertheless, this algorithm has many drawbacks. ALS method can be long to converge without guarantee of finding a global minimum [Bilian. Chen et al., 2012] [Cichocki et al., 2015] [Kolda and Bader, 2009] [Silva et al., 2015]and is dependent on the initialization of the decomposition vectors [Kolda and Bader, 2009]. Several methods have been design to improve ALS performances depending on the decomposition quality, computing resources, computation time [Cichocki et al., 2015] [Faber et al., 2003] such as Tikhonov regularization, maximum block improvement method [Bilian. Chen et al., 2012], coupled-eigenvalue (CE) post-processing [Pereira Da Silva et al., 2015], etc.

The dGNc and PFM3 algorithm showed better results than ALS in [Tomasi, 2006] but were more computationally expensive. CE post-processing improved the decomposition of truncated HOSVD whereas the Sequential rank-one approximation (SeROA), presented in [Silva et al., 2015], highlighted good results that should be compared to ALS. An interesting solution proposed in [Tomasi, 2006] was to combine the different algorithms in order to exploit the benefits of each one. SWATLD algorithm could be used to initialize the decomposition factors of the rank-one tensor decomposition for PMF3, dGN or ALS algorithms before to apply CE post-processing [Pereira Da Silva et al., 2015] [Tomasi, 2006]. However, there is no consensus on the advantages of the proposed alternative compared to ALS algorithms [Kolda and Bader, 2009] [Tomasi and Bro, 2006]

Next section is mainly focused on the most widespread ALS algorithm employed in the REW-NPLS for the PARAFAC tensor decomposition.

## 5.2.2. ALS based PARAFAC decomposition

Alternating least square (ALS) method optimizes one projector ($\mathbf{w}_r^m \in \mathbb{R}^{I_m}$) at a time and fixes the others [Cichocki et al., 2015] [Kolda and Bader, 2009] [Pereira Da Silva et al., 2015]. In the next section, PARAFAC decomposition is considered in the specific case of three-order tensor decomposition to simplify the notation and to be closer to the BCI

application presented in the next chapters. However, all the presented equations are generalizable to N-order tensor decomposition procedure.

Let $\underline{\mathbf{V}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ be a third order tensor which undergoes PARAFAC decomposition. The aim is to find a tensor $\underline{\widehat{\mathbf{V}}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ equal to the linear combination of $R \in \mathbb{N}$ outer product of three normalized projectors $\mathbf{w}_r^1 \in \mathbb{R}^{I_1}$,$\mathbf{w}_r^2 \in \mathbb{R}^{I_2}$, $\mathbf{w}_r^3 \in \mathbb{R}^{I_3}$ weighted with the coefficient $\rho_r \in \mathbb{R}$:

$$\min_{\underline{\widehat{\mathbf{X}}}} \left\| \underline{\mathbf{V}} - \underline{\widehat{\mathbf{V}}} \right\|,$$
$$\underline{\widehat{\mathbf{V}}} = \sum_{r=1}^{R} \rho_r \mathbf{w}_r^1 \circ \mathbf{w}_r^2 \circ \mathbf{w}_r^3 ,$$
$$\|\mathbf{w}_r^1\| = \|\mathbf{w}_r^2\| = \|\mathbf{w}_r^3\| = 1.$$

The factor matrices refers to the concatenation of the decomposition factors $\mathbf{W}^1 \in \mathbb{R}^{I_1 \times R}$,$\mathbf{W}^2 \in \mathbb{R}^{I_2 \times R}$, $\mathbf{W}^3 \in \mathbb{R}^{I_3 \times R}$ with $\mathbf{W}^i = [\mathbf{w}_1^i \; \mathbf{w}_2^i \; \ldots \; \mathbf{w}_R^i]$ and $i = 1,2,3$. From the factor matrices and the weighting vector $\boldsymbol{\rho} \in \mathbb{R}^R$, PARAFAC decomposition can be expressed with the unfolded tensor shape [Kolda and Bader, 2009]:

$$\widehat{\mathbf{V}}_{(1)} = \mathbf{W}^1 \boldsymbol{\rho} (\mathbf{W}^3 \odot \mathbf{W}^2)^{\mathrm{T}},$$
$$\widehat{\mathbf{V}}_{(2)} = \mathbf{W}^2 \boldsymbol{\rho} (\mathbf{W}^3 \odot \mathbf{W}^1)^{\mathrm{T}},$$
$$\widehat{\mathbf{V}}_{(3)} = \mathbf{W}^3 \boldsymbol{\rho} (\mathbf{W}^2 \odot \mathbf{W}^1)^{\mathrm{T}}.$$

The ALS is an iterative procedure which reduces the optimization problem to smaller sub-problems [Tomasi and Bro, 2006]. Each step of the ALS solves a linear regression problem with one vector feature. At each step ALS fixes two of the three matrices $\mathbf{W}^1$, $\mathbf{W}^2$ and $\mathbf{W}^3$ and reduce the problem to a linear least-squares optimization. For example, in an ALS algorithm iteration, $\mathbf{W}^2$ and $\mathbf{W}^3$ are fixed to solve $\mathbf{W}^1$ then $\mathbf{W}^2$ is solved by fixing $\mathbf{W}^1$ and $\mathbf{W}^3$ and, finally, the same operation is realized for $\mathbf{W}^3$. Firstly, $\mathbf{W}^2$ and $\mathbf{W}^3$ are fixed which leads to

$$\min_{\widehat{\mathbf{W}}^1} \left\| \mathbf{V}_{(1)} - \widehat{\mathbf{W}}_{\boldsymbol{\rho}}^1 (\mathbf{W}^3 \odot \mathbf{W}^2)^{\mathrm{T}} \right\|,$$

where $\widehat{\mathbf{W}}^1 \in \mathbb{R}^{I_1 \times R}$ is the estimated factor matrix following the first decomposition dimension with [Kolda and Bader, 2009]:

$$\widehat{\mathbf{W}}_{\boldsymbol{\rho}}^1 = \widehat{\mathbf{W}}^1 \boldsymbol{\rho}. \tag{5.2.1}$$

Minimum is achieved for

$$\widehat{\mathbf{W}}_{\boldsymbol{\rho}}^{\mathbf{1}} = \mathbf{V}_{(1)} [(\mathbf{W}^3 \odot \mathbf{W}^2)^{\mathrm{T}}]^{\dagger},$$

which simplifies due to the Khatri-Rao pseudoinverse properties ((0.1) to

$$\widehat{\mathbf{W}}_{\boldsymbol{\rho}}^1 = \mathbf{V}_{(1)} (\mathbf{W}^3 \odot \mathbf{W}^2) (\mathbf{W}^{3\mathrm{T}} \mathbf{W}^3 * \mathbf{W}^{2\mathrm{T}} \mathbf{W}^2)^{\dagger}.$$

$\mathbf{W}^2$ and $\mathbf{W}^3$ are estimated following the same steps by fixing $\mathbf{W}^1 = \widehat{\mathbf{W}}^1$ using column-wise normalization with (5.2.1) resulting in:

$$\widehat{\mathbf{W}}^1 = \mathbf{V}_{(1)}(\mathbf{W}^3 \odot \mathbf{W}^2)\left(\mathbf{W}^{3\mathrm{T}}\mathbf{W}^3 * \mathbf{W}^{2\mathrm{T}}\mathbf{W}^2\right)^{\dagger},$$
$$\widehat{\mathbf{W}}^2 = \mathbf{V}_{(2)}(\mathbf{W}^3 \odot \mathbf{W}^1)\left(\mathbf{W}^{3\mathrm{T}}\mathbf{W}^3 * \mathbf{W}^{1\mathrm{T}}\mathbf{W}^1\right)^{\dagger},$$
$$\widehat{\mathbf{W}}^3 = \mathbf{V}_{(3)}(\mathbf{W}^2 \odot \mathbf{W}^1)\left(\mathbf{W}^{2\mathrm{T}}\mathbf{W}^2 * \mathbf{W}^{1\mathrm{T}}\mathbf{W}^1\right)^{\dagger}.$$

This procedure is repeated until a specified condition is reached (fixed number of iteration, convergence criterion, etc.). The projection matrices can be initialized with random values, values estimated in previous iteration of the ALS algorithm or values determined using another algorithm such as DTLD [Eliseyev et al., 2017] [Faber et al., 2003] [Kolda and Bader, 2009] [Tomasi and Bro, 2006].

### 5.2.3. PARAFAC decomposition in the REW-NPLS algorithm.

REW-NPLS algorithm integrated a PARAFAC-based decomposition algorithm to extract the set of projectors of $\underline{\mathbf{XY}}_u$. In the next section, the PARAFAC decomposition problem of the REW-NPLS algorithm is considered in the specific case of three order tensor decomposition $\underline{\mathbf{XY}}_u \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, $\|\underline{\mathbf{XY}}_u\| = 1$ to simplify the notation and to be closer to the BCI application considered in the PhD thesis. Nevertheless, all the presented results can be generalized to the n order tensor decomposition.

At each iteration ($f$ is current iteration number) of REW-NPLS algorithm, one iteration of PARAFAC algorithm is used (with a rank one approximation, $R = 1$) to decompose tensor $\underline{\mathbf{XY}}_u$ and to estimate the projectors $\mathbf{w}_f^1$, $\mathbf{w}_f^2$, $\mathbf{w}_f^3$:

$$\min_{\widehat{\underline{\mathbf{XY}}}_u} \|\underline{\mathbf{XY}}_u - \widehat{\underline{\mathbf{XY}}}_u\|,$$
$$\widehat{\underline{\mathbf{XY}}}_u = \rho_f \mathbf{w}_f^1 \circ \mathbf{w}_f^2 \circ \mathbf{w}_f^3,$$
$$\|\mathbf{w}_f^1\| = \|\mathbf{w}_f^2\| = \|\mathbf{w}_f^3\| = 1.$$

Here, $\|\cdot\|$, as a reminder, always referred to the $L_2$ norm (Frobenius, Euclidian norm depending on the variable dimensions). Equally:

$$\min_{\widehat{\underline{\mathbf{XY}}}_u} \|\underline{\mathbf{XY}}_u - \widehat{\underline{\mathbf{XY}}}_u\|^2 \qquad (5.2.2)$$
$$\widehat{\underline{\mathbf{XY}}}_u = \rho_f \mathbf{w}_f^1 \circ \mathbf{w}_f^2 \circ \mathbf{w}_f^3,$$
$$\|\mathbf{w}_f^1\| = \|\mathbf{w}_f^2\| = \|\mathbf{w}_f^3\| = 1.$$

As only one iteration of REW-NPLS algorithm is considered, iteration index $f$ is discarded in the section for the purpose of simplification.All the presented steps are repeated for each latent space dimension $f$ with $f = 1, \ldots, F$.

This problem is no longer an ill-posed problem [Silva et al., 2015]. ALS algorithm guarantees to converge [Wang and Chu, 2014]. In the REW-NPLS algorithm, PARAFAC decomposition is solved using ALS algorithm [Eliseyev et al., 2017]. It optimizes sequentially

$$\min_{\mathbf{w}^1} \left\| \underline{\mathbf{XY}}_{u(1)} - \mathbf{w}^1(\mathbf{w}^3 \otimes \mathbf{w}^2)^{\mathrm{T}} \right\|^2, \tag{5.2.3}$$

$$\min_{\mathbf{w}^2} \left\| \underline{\mathbf{XY}}_{u(2)} - \mathbf{w}^2(\mathbf{w}^3 \otimes \mathbf{w}^1)^{\mathrm{T}} \right\|^2, \tag{5.2.4}$$

$$\min_{\mathbf{w}^3} \left\| \underline{\mathbf{XY}}_{u(3)} - \mathbf{w}^3(\mathbf{w}^2 \otimes \mathbf{w}^1)^{\mathrm{T}} \right\|^2 \tag{5.2.5}$$

until convergence [Uschmajew, 2015]. In the case of three-order tensor, Least Square (LS) solutions for each step are expressed:

$$\mathbf{w}_\rho^1 = \underline{\mathbf{XY}}_{u(1)}(\mathbf{w}^3 \odot \mathbf{w}^2)(\mathbf{w}^{3\mathrm{T}}\mathbf{w}^3 * \mathbf{w}^{2\mathrm{T}}\mathbf{w}^2)^{\dagger}.$$

As $\mathbf{w}^i \in \mathbb{R}^{I_i}$, the solution can be simplified using:

$$\left(\mathbf{w}_f^{3\mathrm{T}}\mathbf{w}_f^3 * \mathbf{w}_f^{2\mathrm{T}}\mathbf{w}_f^2\right) = \left\|\hat{\mathbf{w}}_f^2\right\|^2 * \left\|\hat{\mathbf{w}}_f^3\right\|^2 = \left\|\mathbf{w}_f^3 \otimes \mathbf{w}_f^2\right\|^2 \in \mathbb{R},$$
$$\text{and } \left(\mathbf{w}_f^3 \odot \mathbf{w}_f^2\right) = \left(\mathbf{w}_f^3 \otimes \mathbf{w}_f^2\right).$$

To obtain the least square solution:

$$\mathbf{w}_\rho^1 = \frac{\underline{\mathbf{XY}}_{u(1)}(\mathbf{w}^3 \otimes \mathbf{w}^2)}{\|\mathbf{w}^3 \otimes \mathbf{w}^2\|^2}. \tag{5.2.6}$$

Normalization allows the estimation of parameter $\rho_f$ and $\mathbf{w}^1$ with $\|\mathbf{w}^1\| = 1$. The same procedure is repeated to evaluate both $\mathbf{w}^2$and $\mathbf{w}^3$ :

$$\mathbf{w}_\rho^2 = \frac{\underline{\mathbf{XY}}_{u(2)}(\mathbf{w}^3 \otimes \mathbf{w}^1)}{\|\mathbf{w}^3 \otimes \mathbf{w}^1\|^2}, \tag{5.2.7}$$

$$\mathbf{w}_\rho^3 = \frac{\underline{\mathbf{XY}}_{u(3)}(\mathbf{w}^2 \otimes \mathbf{w}^1)}{\|\mathbf{w}^2 \otimes \mathbf{w}^1\|^2}. \tag{5.2.8}$$

Each one is normalized to evaluate $\rho$ and $\mathbf{w}^1$, $\mathbf{w}^2$, $\mathbf{w}^3$ with $\|\mathbf{w}^1\| = \|\mathbf{w}^2\| = \|\mathbf{w}^3\| = 1$. These three solutions are successively computed until a convergence or maximum iteration number criterion is reached (Figure 5-3).

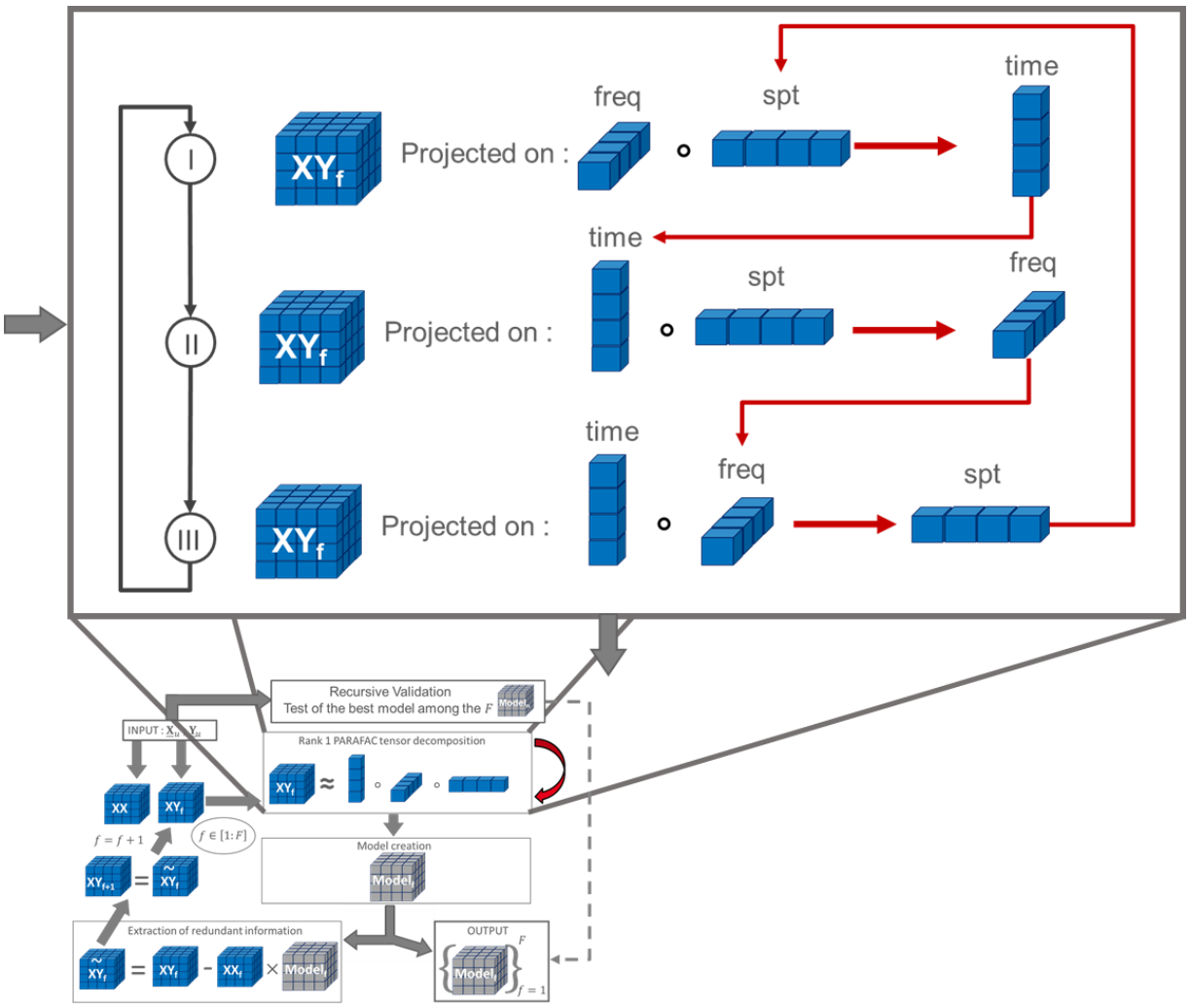



Figure 5-3: PARAFAC-inspired tensor decomposition used in the REW-NPLS algorithm using alternative least square (ALS) algorithm. At each iteration ($f$ is current iteration number), the tensor $\underline{\boldsymbol{XY}}$ is decomposed into three vectors. In the considered case of this PhD manuscript, the decomposition factor $\boldsymbol{w}^1$, $\boldsymbol{w}^2$, $\boldsymbol{w}^3$, are attributed to the time, frequency and spatial domain. Each of the decomposition factor is evaluated alternatively by fixing two of them to estimate the third. This operation is repeated for each factor until a convergence criterion is reached.

## 5.3. Lp-Penalized REW-NPLS (PREW-NPLS)

PREW-NPLS algorithm exploited a penalized version of the PARAFAC algorithm to create group-wise sparse solution. This algorithm is an online adaptive algorithm which introduced $L_p$ penalization with p being the classic norm regularization ($L_1$) or less conventional norm and pseudo-norm penalization type such as $L_0$ and $L_{0.5}$. This section describes the penalized PARAFAC procedure and its integration into the REW-NPLS algorithm to build the new online adaptive sparse PREW-NPLS algorithm.

## 5.3.1. Penalized PARAFAC procedure

In the PARAFAC-based algorithm used in REW-NPLS, ALS strategy fixes all projectors except one at each step of the algorithm. Consequently, each step of the ALS solved a linear regression with one vector feature. In this section, $L_0$, $L_{0,5}$ and $L_1$ regularized linear regression are simplified to be applied in online PARAFAC subroutine of REW-NPLS. The following equation will be presented in the case of three-order tensor and rank one ($R = 1$) PARAFAC decomposition to simplify the notations but can be generalized to N-order tensor.

Given a three order-tensor $\underline{\mathbf{V}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ to decompose using regularized PARAFAC with ALS strategy and $\mathbf{w}^i \in \mathbb{R}^{*I_i}$ with $i = 1,2,3$ the decomposition factors estimated by the PARAFAC. Let us consider the unfolded tensor $\underline{\mathbf{V}}_{(i)}$ with $\underline{\mathbf{V}}_{(i)} = \left(\mathbf{v}_1^1 | \dots | \mathbf{v}_1^{I_1}\right) \in \boldsymbol{R}^{I_1 \times I_2 I_3}$ where $\mathbf{v}_i^j$ are the rows of matrix $\underline{\mathbf{V}}_{(i)}$. Taking into account that $(\mathbf{w}^2 \otimes \mathbf{w}^1)^{\mathrm{T}} \in \boldsymbol{R}^{I_1 I_2}$, $(\mathbf{w}^3 \otimes \mathbf{w}^1)^{\mathrm{T}} \in \boldsymbol{R}^{I_1 I_3}$ and $(\mathbf{w}^3 \otimes \mathbf{w}^2)^{\mathrm{T}} \in \boldsymbol{R}^{I_2 I_3}$ are vectors, the optimization tasks (5.2.3)-(5.2.5) are separated into element-wise optimization:

$$\min_{\mathrm{w}_j^1} \left\| \mathbf{v}_1^j - \mathrm{w}_j^1 (\mathbf{w}^3 \otimes \mathbf{w}^2)^{\mathrm{T}} \right\|^2 \quad j = 1, \dots, I_1, \tag{5.3.1}$$

$$\min_{\mathrm{w}_j^2} \left\| \mathbf{v}_2^j - \mathrm{w}_j^2 (\mathbf{w}^3 \otimes \mathbf{w}^1)^{\mathrm{T}} \right\|^2 \quad j = 1, \dots, I_2, \tag{5.3.2}$$

$$\min_{\mathrm{w}_j^3} \left\| \mathbf{v}_3^j - \mathrm{w}_j^3 (\mathbf{w}^2 \otimes \mathbf{w}^1)^{\mathrm{T}} \right\|^2 \quad j = 1, \dots, I_3. \tag{5.3.3}$$

Here $\mathrm{w}_j^i$ are the projector elements of vectors $\mathbf{w}^1 = \left(\mathrm{w}_1^1, \dots, \mathrm{w}_{I_1}^1\right)^T \in \mathbb{R}^{*I_1}$, $\mathbf{w}^2 = \left(\mathrm{w}_1^2, \dots, \mathrm{w}_{I_2}^2\right)^T \in \mathbb{R}^{*I_2}$, and $\mathbf{w}^3 = \left(\mathrm{w}_1^3, \dots, \mathrm{w}_{I_3}^3\right)^T \in \mathbb{R}^{*I_3}$ estimated by the PARAFAC. The (5.2.6)-(5.2.8) least squares (LS) solutions may be written as:

$$\left(\mathrm{w}_j^1\right)_{LS} = \frac{\mathbf{v}_1^j (\mathbf{w}^3 \otimes \mathbf{w}^2)}{\|\mathbf{w}^3 \otimes \mathbf{w}^2\|^2}, \qquad j = 1, \dots, I_1, \tag{5.3.4}$$

$$\left(\mathrm{w}_j^2\right)_{LS} = \frac{\mathbf{v}_2^j (\mathbf{w}^3 \otimes \mathbf{w}^1)}{\|\mathbf{w}^3 \otimes \mathbf{w}^1\|^2}, \qquad j = 1, \dots, I_2 \tag{5.3.5}$$

$$\left(\mathrm{w}_j^3\right)_{LS} = \frac{\mathbf{v}_3^j (\mathbf{w}^2 \otimes \mathbf{w}^1)}{\|\mathbf{w}^2 \otimes \mathbf{w}^1\|^2}, \qquad j = 1, \dots, I_3. \tag{5.3.6}$$

Sparse $L_p$ ($p = 0, \frac{1}{2}, 1$) norm/pseudo norm penalization including a protection variable characteristic is proposed to be integrated to the cost function of REW-NPLS procedure to provide a group-wise sparse solutions, namely, solutions sparse by slices. Optimization task (5.2.2) is replaced by the optimization of the cost function penalized with $L_p$ ($p = 0, \frac{1}{2}, 1$) norm/pseudo norms:

$$\min\left\|\underline{\mathbf{V}} - \underline{\widehat{\mathbf{V}}}\right\|^2 + \mathrm{P}(\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3), \tag{5.3.7}$$

$$\mathrm{P}(\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3) = \lambda_1 \|\mathbf{w}^1\|_{q,\mathcal{L}_1} + \lambda_2 \|\mathbf{w}^2\|_{q,\mathcal{L}_2} + \lambda_3 \|\mathbf{w}^3\|_{q,\mathcal{L}_3},$$

$$\|\mathbf{w}^1\| = \|\mathbf{w}^2\| = \|\mathbf{w}^3\| = 1.$$

Where $\left\|\mathbf{w}^i\right\|_{p,\mathcal{L}_i}$ for $p = 0, \frac{1}{2}, 1$ and $i = 1, 2, 3$ is denoted as :

$$\left\|\mathbf{w}^i\right\|_{0,\mathcal{L}_i} = \sum_{k \in \mathcal{L}_i} \left(1 - \delta_{0,\mathrm{w}_k^i}\right),$$

$$\left\|\mathbf{w}^i\right\|_{1,\mathcal{L}_i} = \sum_{k \in \mathcal{L}_i} \left|\mathrm{w}_k^i\right|,$$

$$\left\|\mathbf{w}^i\right\|_{\frac{1}{2},\mathcal{L}_i} = \sum_{k \in \mathcal{L}_i} \sqrt{\left|\mathrm{w}_k^i\right|}.$$

Here, the regularization functions may only regularize a part of the indices (projector elements). The indices of the potentially penalized projector elements are defined by a set $\mathcal{L}_i \subset \{1,2,\dots,I_i\}$ with $i = 1, 2, 3$ while the other elements not included in $\mathcal{L}_i$ are "protected" and cannot be penalized. $\mathcal{L}_i$ may vary depending on the REW-NPLS iteration. $0 < \lambda_i \leq 1$ are the penalization hyperparameters. The Kronecker delta $\delta_{0,\mathrm{w}_k^i} = 1$ if $\mathrm{w}_k^i = 0$, $\delta_{0,\mathrm{w}_k^i} = 0$ otherwise.

The same ALS strategy (5.2.3)-(5.2.5) than the procedure used in conventional REW-NPLS is proposed to be applied for the optimization of (5.3.7). ALS fixed all projectors except one at each step of the algorithm, leading to the three successive optimization tasks:

$$\min_{\mathbf{w}^1}\left(\left\|\underline{\mathbf{V}}_{(1)} - \mathbf{w}^1(\mathbf{w}^3 \otimes \mathbf{w}^2)^{\mathrm{T}}\right\|^2 + \lambda_1 \|\mathbf{w}^1\|_{q,\mathcal{L}_1}\right),$$

$$\min_{\mathbf{w}^2}\left(\left\|\underline{\mathbf{V}}_{(2)} - \mathbf{w}^2(\mathbf{w}^3 \otimes \mathbf{w}^1)^{\mathrm{T}}\right\|^2 + \lambda_2 \|\mathbf{w}^2\|_{q,\mathcal{L}_2}\right),$$

$$\min_{\mathbf{w}^3}\left(\left\|\underline{\mathbf{V}}_{(3)} - \mathbf{w}^3(\mathbf{w}^2 \otimes \mathbf{w}^1)^{\mathrm{T}}\right\|^2 + \lambda_3 \|\mathbf{w}^3\|_{q,\mathcal{L}_3}\right).$$

The solutions of non-regularized problem (5.3.4)-(5.3.6) are used as initial approximation and are referred as the Least Square (LS) solution noted $\mathbf{w}_{LS}^i$.

Previously, similar penalized ALS was considered in [Eliseyev et al., 2012]. However the study was limited to $\mathrm{L}_1$-norm and did not consider additional protection variables $\mathcal{L}_i$. Moreover, the problem was solved using non-adaptive NPLS regression for offline classification preclinical experiments and highlighted non-viable solution for real-time processing if more than 14 electrodes were considered [Eliseyev et al., 2012]. More general case of $\mathrm{L}_\mathrm{p}$ ($\mathrm{p} = 0, \frac{1}{2}, 1$)-norm/pseudo-norm penalization with possible variable protection

procedure in considered in [Moly et al., 2020] and an efficient integration to REW-NPLS algorithm is proposed in the manuscript.

Unlike the non-regularized ALS algorithm (5.2.3)-(5.2.5), the norms of projectors are not arbitrary parameters anymore due to penalization terms. Therefore, the normalization of current estimate is added into ALS optimization cycle [Moly et al., 2020].

$$\min_{\widetilde{\mathbf{w}}^1}\left(\left\|\underline{\mathbf{V}}_{(1)}-\widetilde{\mathbf{w}}^1(\mathbf{w}^3\otimes\mathbf{w}^2)^{\mathrm{T}}\right\|^2+\lambda_1\|\widetilde{\mathbf{w}}^1\|_{q,\mathcal{L}_1}\right)\ with\ \mathbf{w}^1=\frac{\widetilde{\mathbf{w}}^1}{\|\widetilde{\mathbf{w}}^1\|},\tag{5.3.8}$$

$$\min_{\widetilde{\mathbf{w}}^2}\left(\left\|\underline{\mathbf{V}}_{(2)}-\widetilde{\mathbf{w}}^2(\mathbf{w}^3\otimes\mathbf{w}^1)^{\mathrm{T}}\right\|^2+\lambda_2\|\widetilde{\mathbf{w}}^2\|_{q,\mathcal{L}_2}\right)\ with\ \mathbf{w}^2=\frac{\widetilde{\mathbf{w}}^2}{\|\widetilde{\mathbf{w}}^2\|},\tag{5.3.9}$$

$$\min_{\widetilde{\mathbf{w}}^3}\left(\left\|\underline{\mathbf{V}}_{(3)}-\widetilde{\mathbf{w}}^3(\mathbf{w}^2\otimes\mathbf{w}^1)^{\mathrm{T}}\right\|^2+\lambda_3\|\widetilde{\mathbf{w}}^3\|_{q,\mathcal{L}_3}\right)\ with\ \mathbf{w}^3=\frac{\widetilde{\mathbf{w}}^3}{\|\widetilde{\mathbf{w}}^3\|}.\tag{5.3.10}$$

Next, for faster computing, it can be noted that all considered regularization functions are decomposed as a sum of element-wise functions. Consequently, similarly to (5.3.2)-(5.3.4) optimization tasks (5.3.8)-(5.3.10) are split into element-wise optimization:

$$\min_{\mathrm{w}_j^1}\left(\left\|\mathbf{v}_1^j-\mathrm{w}_j^1(\mathbf{w}^3\otimes\mathbf{w}^2)^{\mathrm{T}}\right\|^2+\lambda_1 g_p(\mathrm{w}_j^1)\right), j=1,\dots,I_1,\tag{5.3.11}$$

$$\min_{\mathrm{w}_j^2}\left(\left\|\mathbf{v}_2^j-\mathrm{w}_j^2(\mathbf{w}^3\otimes\mathbf{w}^1)^{\mathrm{T}}\right\|^2+\lambda_2 g_p(\mathrm{w}_j^2)\right), j=1,\dots,I_2,\tag{5.3.12}$$

$$\min_{\mathrm{w}_j^3}\left(\left\|\mathbf{v}_3^j-\mathrm{w}_j^3(\mathbf{w}^2\otimes\mathbf{w}^1)^{\mathrm{T}}\right\|^2+\lambda_3 g_p(\mathrm{w}_j^3)\right), j=1,\dots,I_3,\tag{5.3.13}$$

$$with$$

$$g_p(\mathrm{w}_j^i)=\begin{cases}1-\delta_{0,\mathrm{w}_j^i}, & \text{if } p=0 \text{ and } \mathrm{w}_j^i\in\mathcal{L}_i\\ \left|\mathrm{w}_j^i\right|, & \text{if } p=1 \text{ and } \mathrm{w}_j^i\in\mathcal{L}_i\\ \sqrt{\left|\mathrm{w}_j^i\right|}, & \text{if } p=1/2 \text{ and } \mathrm{w}_j^i\in\mathcal{L}_i\\ 0 & \text{otherwise}\end{cases}.\tag{5.3.14}$$

In the next subsections, the particular cases of $L_0$, $L_1$, $L_{0.5}$ penalizations are presented. Details of the demonstration are available in [Moly et al., 2020].

**In the case of $L_0$ penalization** which penalized the parameter weights depending on the number of non-zero coefficients, and considering one of the optimization step, e.g. (5.3.11) of ALS optimization [Moly et al., 2020] The solution turns out to be an element-wise hard thresholding of the least square solution $\left(\mathrm{w}_j^1\right)_{LS}\ j=1,\dots,I_1$ leading to [Moly et al., 2020]:

$$\left(\mathrm{w}_j^1\right)_{L_0} = \begin{cases} 0 & \text{if } j \in \mathcal{L}_1 \text{ and } \left(\mathrm{w}_j^1\right)_{LS} \leq Threshold_{L_0} \\ \left(\mathrm{w}_j^1\right)_{LS} & \text{otherwise} \end{cases},$$

$$Threshold_{L_0} = \frac{\sqrt{\lambda_1}}{\|\mathbf{w}^3 \otimes \mathbf{w}^2\|}.$$

**In the case of $L_{0.5}$ penalization,** and considering one of the optimization step, e.g. (5.3.11) of ALS optimization. Based on (5.3.11) and (5.3.14), the function to minimize $FopL_{0.5}$ takes the form:

$$FopL_{0.5}\left(\mathrm{w}_j^1\right) = \left\|\mathbf{v}_1^j - \mathrm{w}_j^1(\mathbf{w}^3 \otimes \mathbf{w}^2)^{\mathrm{T}}\right\|^2 + \lambda_1\sqrt{\left|\mathrm{w}_j^1\right|}, \tag{5.3.15}$$

or equivalently:

$$FopL_{0.5}\left(\mathrm{w}_j^1\right) = \|\mathbf{w}^3 \otimes \mathbf{w}^2\|^2\left(\left(\mathrm{w}_j^1\right)_{LS} - \mathrm{w}_j^1\right)^2 + \lambda_1\sqrt{\left|\mathrm{w}_j^1\right|}. \tag{5.3.16}$$

The solution to this minimization problem is:

$$\left(\mathrm{w}_j^1\right)_{L_{0.5}} = \begin{cases} 0, & \text{if } j \in \mathcal{L}_1 \text{ and } \left(\mathrm{w}_j^1\right)_{LS} \leq ThresholdL_{0.5} \\ argmin\left(FopL_{0.5}(0), FopL_{0.5}\left(\mathcal{B}.\left(\mathrm{w}_j^1\right)_{LS}\right)\right), & \text{if } i \in \mathcal{L}_1 \text{ and } \left(\mathrm{w}_j^1\right)_{LS} > ThresholdL_{0.5} \\ \left(\mathrm{w}_j^1\right)_{LS} & \text{otherwise} \end{cases},$$

where

$$ThresholdL_{0.5} = \frac{3}{4}\left(\frac{\lambda_1}{\|\mathbf{w}^3 \otimes \mathbf{w}^2\|^2}\right)^{\frac{2}{3}},$$

and $\mathcal{B}$ is the solution of the cubic polynomial function (Figure 5-4):

$$x(1-x)^2 = C \tag{5.3.17}$$

$$with\ x = \frac{\mathrm{w}_j^1}{\left(\mathrm{w}_j^1\right)_{LS}}\ and\ C = \frac{{\lambda_1}^2}{16\|\mathbf{w}^3 \otimes \mathbf{w}^2\|^4\left(\left(\mathrm{w}_j^1\right)_{LS}\right)^3}.$$

To summarize, in the case $C > \frac{4}{27}$, $\left(\mathrm{w}_j^1\right)_{L_{0.5}} = 0$ whereas in the case $C \in \left[0, \frac{4}{27}\right]$, by the properties of the cubic polynomial function (Figure 5-4), the biggest root of (5.3.17) in the interval $[0; 1]$ is in the interval $\left[\frac{1}{3}; 1\right]$ which allow to easily compute $\mathcal{B}\left(\mathrm{w}_j^1\right)_{LS}$ and have a straightforward solution between 0 and $\mathcal{B}\left(\mathrm{w}_j^1\right)_{LS}$.

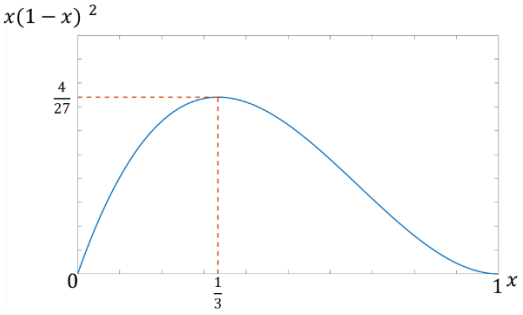



Figure 5-4 : Cubic polynomial function related to the evaluation of $\left(w_j^1\right)_{L_{0.5}}$.

Finally**, in the case of $L_1$ penalization**, considering one of the optimization step, e.g. (5.3.11) of ALS optimization, the solution turns out to be an element-wise soft-thresholding of the least square solution $\left(\mathrm{w}_j^1\right)_{LS}\ j = 1, \dots, I_1$ leading to [Moly et al., 2020]:

$$\left(\mathrm{w}_j^1\right)_{L_1} = \begin{cases} 0 & \text{, if } j \in \mathcal{L}_1 \text{ and } \left(\mathrm{w}_j^1\right)_{LS} \leq ThresholdL_1 \\ sign\left(\left(\mathrm{w}_j^1\right)_{LS}\right)\left(\left|\left(\mathrm{w}_j^1\right)_{LS}\right| - ThresholdL_1\right) & \text{, if } i \in \mathcal{L}_1 \text{ and } \left(\mathrm{w}_j^1\right)_{LS} > ThresholdL_1, \\ \left(\mathrm{w}_j^1\right)_{LS} & \text{otherwise} \end{cases}$$

$$ThresholdL_1 = \frac{\lambda_1}{\left\|\mathbf{w}^3 \otimes \mathbf{w}^2\right\|^2}.$$

## 5.3.2. Penalized PARAFAC in the PREW-NPLS algorithm

Penalized PARAFAC based tensor decomposition is integrated into REW-NPLS algorithm to extract iteratively the set of penalized projectors $\left\{\mathbf{w}_f^1 \in \mathbb{R}^{I_1}, \mathbf{w}_f^2 \in \mathbb{R}^{I_2}, \mathbf{w}_f^3 \in \mathbb{R}^{I_3}\right\}_{f=1}^{F}$ from $\underline{\mathbf{XY}}_u$ for each latent space dimension $f \subset \{1,2,\dots,F\}$.

For $f = 1$, all the projector elements can be potentially penalized. Therefore, the protection set is initialized to $\mathcal{L}_{i,1} \subset \{1,2,\dots,I_j\}$ as each projector element can be penalized. For any $f$, after that the PARAFAC convergence criteria are reached, indices with non-zero elements of $\mathbf{w}_f^i$ (non-penalized projector elements) are removed from $\mathcal{L}_{i,f}$ resulting in the protection set for the next iteration $\mathcal{L}_{i,f+1} \subset \mathcal{L}_{i,f}$. Therefore, the non-penalized elements at the iteration $f$ cannot be penalized anymore for the next iterations $f+1, f+2, \dots, F$.

The protection variable is introduced because REW-NPLS model is estimated via an incremental procedure, the model at iteration $f+1$ contains information extracted at iteration $f$. Therefore, if a decomposition factor has a non-zero value at iteration $f$, it must be considered at iteration $f+1$. A scheme representing the basic steps of the PREW-NPLS main loop for a specific $f$ is represented in the case of spatial $L_1$ penalization with a penalization hyperparameter $\lambda$ in Figure 5-5A whereas one loop of

the penalized PARAFAC estimated with the ALS algorithm integrated in the PREW-NPLS algorithm is shown in Figure 5-5B. With the exception of the penalized PARAFAC decomposition, PREW-NPLS model calibration is similar to the REW-NPLS algorithm (presented in the Chapter 4). At each iteration $u$, a set of $F$ models are evaluated with a penalization hyperparameter $\lambda$ and is noted $\theta_{u,\lambda} = \left\{\underline{\mathbf{Beta}}_u^{f,\lambda}, \underline{\mathbf{bias}}_u^{f,\lambda}\right\}_{f=1}^{F}$.

The previously presented PREW-NPLS based on the regularized PARAFAC procedure allows to perform group-wise parameter penalization for a fixed penalization hyperparameter $\lambda$. The selection of this hyperparameter influences greatly the sparsity of the solution and the global performance of the algorithm. The selection of the $\lambda$ hyperparameter may be a complex task and is often optimized based on random or grid search using cross-validation strategy. However such strategy cannot be applied for online decoding because they require high computing resources, too long computing time and are not suited to data-flow processing. Therefore, during online experiments, penalization hyperparameter $\lambda$ is commonly fixed using prior knowledge or preliminary offline studies whereas the optimal penalization hyperparameter might be different in offline and online closed-loop experiments.

To overpass this drawback, an upgrade of the PREW-NPLS algorithm named Automatic Lp-PREW-NPLS (APREW-NPLS) is proposed. APREW-NPLS is an adaptive penalized REW-NPLS which automatically evaluate the performance of several penalization hyperparameters.

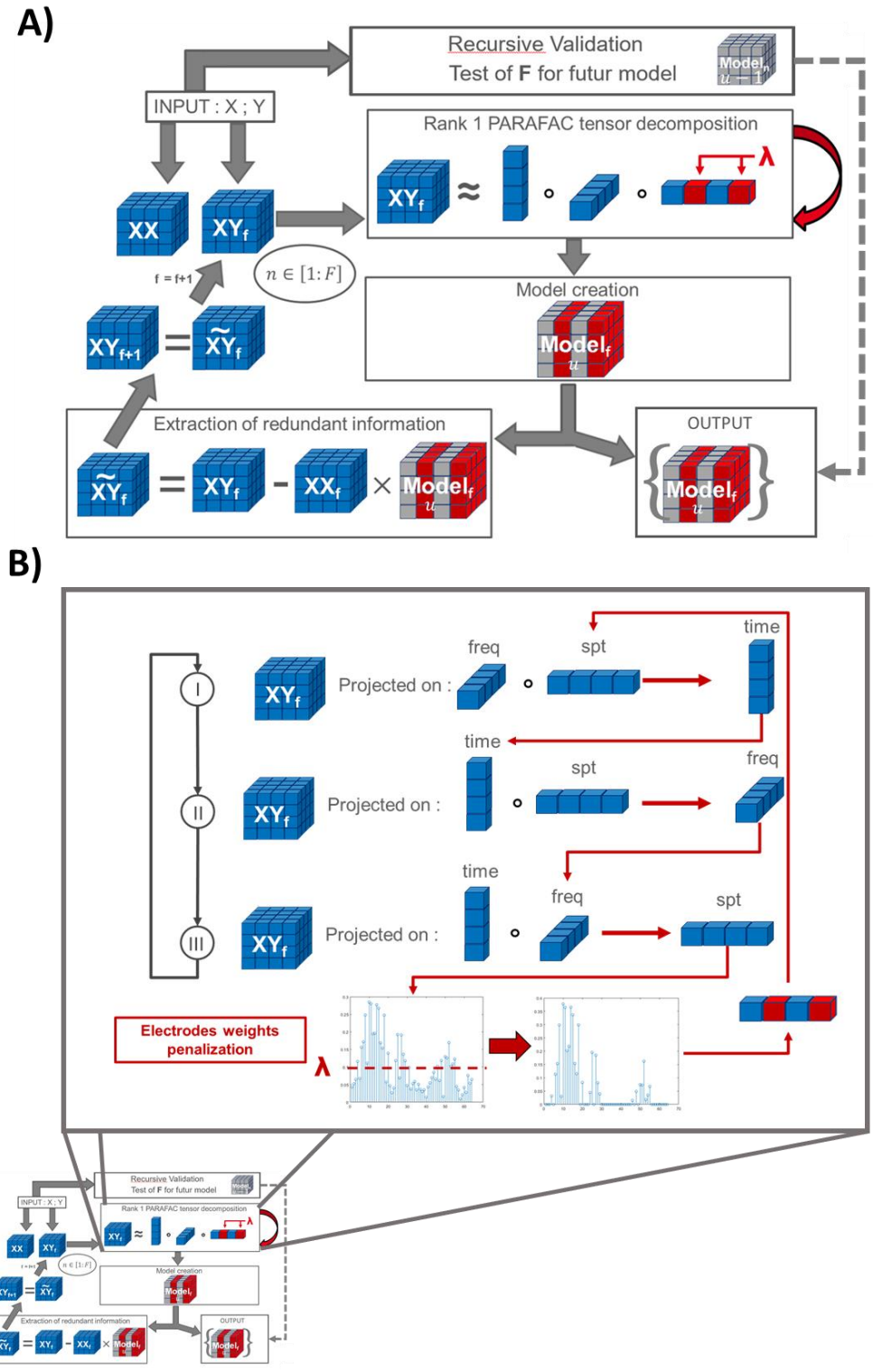



Figure 5-5 : Penalized REW-NPLS (PREW-NPLS) algorithm. (A) PREW-NPLS algorithm main steps with penalized PARAFAC decomposition leading to slice-wise sparse model. (B) Example of the $L_1$-PARAFAC decomposition performed in the case of $L_1$-PREW-NPLS penalization on the space (electrodes) domain with the hyperparameter $\lambda$. ALS algorithm is used for decomposition factor estimation.

## 5.4. Automatic $\lambda$ penalized REW-NPLS (APREW-NPLS)

Although machine learning aims to optimize numerous parameters to automatically find a model that fits a problem, some specific parameters need to be fixed before the learning process begin. These parameters are named "hyperparameters" and can be found in all the machine learning strategies (supervised, reinforcement, unsupervised). Hyperparameters govern numerous aspects of machine learning algorithms [Bishop, 2006] [Jia Wu and Jia Wu, n.d.] [Kuhn and Johnson, 2013]. They can define the model architecture (neural networks, mixture of experts structure), the applied kernel functions (nonlinear SVM), the regularization term ($L_1$ or $L_2$ norm penalization), the learning rates (KNN clustering convergence speed), the number of decomposition factors (PCA, PARAFAC, ICA, etc.), the dimension of the latent variable space (PLS) [Bishop, 2006] [Flamary et al., 2016] [Hutter et al., 2011a] [Kuhn and Johnson, 2013] [L. Li et al., 2018] etc. Hyperparameters highly influence the generalization of a model. Consequently, it is required to find the optimal hyperparameters. However, hyperparameter dynamic is not well understand and may be difficult to optimize [Jia Wu and Jia Wu, n.d.] [Keerthi et al., 2006] [L. Li et al., 2018].

APREW-NPLS is an incremental adaptive penalized algorithm which aims to select the best penalization hyperparameter $\lambda$, introduced by the PREW-NPLS algorithm. Before to introduce the APREW-NPLS method to evaluate the algorithm performance with various penalization hyperparameter $\lambda$, a brief review on the state of the art of the hyperparameter optimization strategies is proposed.

### 5.4.1. Hyperparameter optimization

Hyperparameter automatic optimization is an important field of a more general research domain named Automated Machine Learning (AutoML) gathering various techniques and strategies for algorithm selection, model selection, hyperparameter optimization, etc. [Feurer et al., 2015] [Hutter et al., 2014]. Nevertheless, most of the reported studies were theoretical and needed to be tested in real life applications.

With the exception of some EEG studies oriented on feature selection [Andreu-Perez et al., 2018] [Corralejo et al., 2011] [Faller et al., 2012] [Flamary et al., 2016] [Garrett et al., 2003] [Long et al., 2011] [Mend and Kullmann, 2012] [Oliver et al., 2013] [Schroder et al., 2003], a deep neural network stochastic gradient descent optimization [Shojaedini et al., 2018] or a dynamic stopping calibration criterion procedure [Schreuder et al., 2013], AutoML is highly uncommon in the BCI field. The only hyperparameter optimization procedure reported in online adaptive experiments were limited to adaptive feature selection algorithms for P300 or binary classification [Andreu-Perez et al., 2018] [Faller et al., 2012] [Long et al., 2011] [Mend and Kullmann, 2012] [Oliver et al., 2013]. To our knowledge, in the BCI field, no algorithm has been designed for real-time automatic selection of an optimal

penalization hyperparameter for online adaptive model. AutoML is a poorly explored field in the domain of motor BCI. However, an overview of AutoML state of the art in the BCI field and other domains is detailed in the next section.

Strategies and algorithms focused on hyperparamer optimization can be clustered in different groups. Firstly, the most common reported strategies were grid (exhaustive) and random search in the hyperparameter space [Flamary et al., 2016] [Schreuder et al., 2013]. Grid search tests numerous hyperparameter configurations linearly (or not) distributed over the hyperparameter space whereas random search randomly selects hyperparameter configurations. Among these strategies, studies demonstrated the superiority of random search providing better results with smaller computation time [Bergstra and Bengio, 2012] [Hutter et al., 2014]. Although these strategies are simple to understand and implement, they are restricted to a low number of hyperparameter optimization due to the exponentially increasing possible settings with growing dimension of the hyperparameter space. These methods do not find optimal solution, are computationally expensive and time consuming. To handle complex algorithm configuration optimization, subtler methods were designed.

Several studies reported optimization strategies based on Model-free methods. These strategies are quite simple and efficient because they do not have to alternate between fitting and testing a model to find a solution. They are based on stochastic search of optimal hyperparameter configuration. For example, classical gradient descent based strategies were used for hyperparameter optimization [Bakhteev and Strijov, 2019] [Chapelle et al., 2002] [Keerthi et al., 2006]. Racing algorithm family regroups numerous algorithms (ROAR, F-race family, etc) based on competitive learning where bad configurations are iteratively removed from the configuration space [Hutter et al., 2011a]. F-Race algorithms family evaluates the performance of numerous hyperparameter configurations and use a nonparametric Friedman test to identify statistically less efficient configurations before to remove them from the space of possible configurations [Hutter et al., 2011a]. PaRaMILS employs local search procedure to find the best configuration before to undergo perturbations and repeat this procedure [Hutter et al., 2009, 2011a].

Another group of Model-free algorithms named population-based methods clusters the genetic algorithms (GA) such as Tabu_GA, or gender GA [Guo et al., 2019] [Hutter et al., 2011a, 2014]. In a first step, these algorithms starts with a finite set of possible hyperparameter configurations and evaluate the performance of each of them. Next, the set of hyperparameter configurations are mixed ("reproduce") to form a new set of configurations. The hyperparameter configurations which presents the best performance are more likely to be used to create the next generation of hyperparameter configurations [Whitley, 1994]. GAs have been reported in BCI research for EEG optimal feature selection [Corralejo et al., 2011] [Garrett et al., 2003] [Schroder et al., 2003]. Another

reported Model-free algorithm was based on the common multi-armed bandit problem from the reinforcement-learning field. Multi-Armed Bandit (MAB) problem can be defined as the selection of an action among $N$. For each action, a reward from a probability distribution is obtained. MAB solution aims to maximize the expected reward over numerous successive actions or a time period [L. Li et al., 2018] [Slivkins, 2019] [Sutton and Barto, 2017]. In the PhD studied case, the selection of an action is associated to the selection of a hyperparameter configuration and the reward is an evaluation of the performance of the selected configuration. Several MAB algorithms were designed to handle the autoML problems for hyperparameter optimization such as HYPERBAND algorithm (based on pure-exploration non-stochastic infinite-armed bandit problem and racing algorithms) [Dôres et al., 2018] [L. Li et al., 2018], algorithms for simultaneous model and hyperparameter selection [Efimova et al., 2019], MASSCAH algorithm [Shalamov et al., 2018], Extreme-Region Upper Confidence Bound Bandit (ER-UCB bandit) [Hu et al., 2019], TUPAQ [Sparks et al., 2015], etc. Q-learning algorithm (reinforcement learning method) was also used for dynamic model selection [Feng and Zhang, 2018].

Finally, Model-based approach regroups several methods which recently highlighted interesting results. Strategies based on Sequential Model-Based Optimization (SMBO) alternatively fit the models and apply them in order to select the next hyperparameter configuration to evaluate [Hutter et al., 2011a]. Bayesian optimization methods outperformed or at least provided equivalent performance than other SMBO algorithms [Z. Wang et al., 2016]. Bayesian optimization algorithms estimating the probability $p(y|\lambda)$ with $y$ the performance evaluation and $\lambda$ a set of hyperparameter configuration were reported in several studies [L. Li et al., 2018] [Z. Wang et al., 2016]. Bayesian optimization methods aim to optimize the hyperparameter selection in an adaptive procedure using exploration-exploitation strategy to fit $p(y|\lambda)$. State of the art algorithms employed Gaussian distribution to model $p(y|\lambda)$ using various algorithms such as random forest algorithms in Sequential Model-based Algorithm Configuration (SMAC) [Hutter et al., 2011a] [Thornton et al., 2013], random matrix to reduce the hyperparameter configuration space dimension in Random Embedding Bayesian Optimization (REMBO) algorithm [Z. Wang et al., 2016], Tree-structure Parzen Estimator (TPE) [Thornton et al., 2013], efficient global optimization (EGO) [Hutter et al., 2011a] [Jones et al., 1998], Entropy Search algorithm for fast Bayesian hyperparameter optimization (Fabolas) [Klein et al., 2017], etc.

Model-based approach can lead to better hyperparameter configuration optimization [Klein et al., 2017]. However, these algorithms are computationally expensive and are more complicated to implement than model-free approach. Additionally, some limitations of Model-based approach compared to model-free approach were reported [Hutter et al., 2011a] [Klein et al., 2017] [L. Li et al., 2018].

Model-based and Model-free strategies can be clustered into two different categories. The algorithms which are optimizing hyperparameter selection in adaptive manner

trying to find the optimal configuration from a specified hyperparameter set of possible configurations (GA, SMBO algorithm) and the ones which are adaptive in computation by eliminating the bad configuration without uniformly training all the configuration (HYPERBAND, racing algorithms etc.).

The presented methods performed the hyperparameter optimizations. Nevertheless, the majority of the proposed algorithms were only tested on simulated data [Bakhteev and Strijov, 2019] [Hu et al., 2019] [Hutter et al., 2009, 2011a] [Klein et al., 2017] [Z. Wang et al., 2016] or from various online available dataset repository [Chapelle et al., 2002] [Dôres et al., 2018] [Efimova et al., 2019] [Feng and Zhang, 2018] [Keerthi et al., 2006] [L. Li et al., 2018] [Shalamov et al., 2018] [Sparks et al., 2015] [Thornton et al., 2013]. With the exception of the Genetic algorithms and the Particle Swarn Optimization (PSO) for feature selection optimization [Atyabi et al., 2016] [Corralejo et al., 2011] [Moro et al., 2017] [Park et al., 2018] [Schroder et al., 2003], none of these methods were evaluated in the BCI fields. Additionally numerous strategies are computationally expensive and/or are optimized using cross-validation procedure [Chapelle et al., 2002] [Flamary et al., 2016] [Hutter et al., 2014] [Keerthi et al., 2006]. These methods were not tested in the scope of online closed-loop decoders hyperparameter optimization (even though ROAR may be tested). An online closed-loop automatic decoder should train the decoder, optimize the hyperparameters and apply the decoder with the best hyperparameters in the same time.

The previously proposed Penalized-REW-NPLS algorithm brings penalized solution to online adaptive decoders. However, PREW-NPLS algorithm requires to set a fixed regularization hyperparameter $\lambda$ which modulates the sparsity of the model. Selecting a good penalization hypermarameter which preserves the neural signal decoding performance while increasing the solution sparsity is a complicated problem. PREW-NPLS algorithm requires prior offline study to select the most efficient $\lambda$ before to apply the selected penalization hyperparameter during online adaptive closed-loop experiments.

We present an Automatic $\lambda$ Penalized-REW-NPLS algorithm using model-free algorithm configuration methods inspired of multi-arm bandit problem to train and optimize automatically several models with different penalization hyperparameters. Model-free algorithm configuration methods inspired of multi-arm bandit problem was selected because it is a suited to online adaptive algorithm as it requires less computational resources than model-based algorithms.

### 5.4.2. Automatic $\lambda$ Penalized-REW-NPLS algorithm

Multi-arm bandit problem is a well-known optimization problem. Given an unknown environment in which numerous action can be performed. Each action on the environment emits a reward chosen from an unknown probability distribution which is dependent on the selected action. The objective is to perform the best actions to

maximize the expected reward over a time period [Slivkins, 2019] [Sutton and Barto, 2017]. Multi-arm bandit is an exploitation versus exploration algorithm that can be integrated into the decoder calibration procedure to optimize the hyperparameter settings. In the case of APREW-NPLS, multi-arm bandit optimization algorithm was integrated into the Recursive Validation procedure of APREW-NPLS to optimize the penalization hyperparamter $\lambda$ during the model calibration steps.

### 5.4.2.1. APREW-NPLS principles

Given an update iteration number $u \in \mathbb{N}$, APREW-NPLS considers a set of penalization hyperparameter $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \dots, \lambda_i, \dots, \lambda_\Lambda\}$, $\Lambda \in \mathbb{N}^+$, related to a set of penalized models $\boldsymbol{\Theta}_u(\boldsymbol{\lambda}) = \{\theta_{u,\lambda_i}\}_{\lambda_i \in \boldsymbol{\lambda}}$ with $\theta_{u,\lambda_i} = \left\{\underline{\mathbf{Beta}}_u^{f,\lambda_i}, \underline{\mathbf{bias}}_u^{f,\lambda_i}\right\}_{f=1}^{F}$, $\lambda_i \in \boldsymbol{\lambda}$. $F \in \mathbb{N}$ is the fixed upper bound latent space dimension. $\underline{\mathbf{Beta}}_u^{f,\lambda_i} \in \mathbb{R}^{(I_1 \times \dots \times I_M) \times (J_1 \times \dots \times J_N)}$, $\underline{\mathbf{bias}}_u^{f,\lambda_i} \in \mathbb{R}^{J_1 \times \dots \times J_N}$ are the current models' parameters and associated bias with the penalization hyperparameter $\lambda_i$ (see section 4.4) (Figure 5-6). APREW-NPLS algorithm only updates at each iteration $u$ the models that maximizes the expected rewards among all the models. The expected reward set is defined as $\mathbf{Q}_u = \{Q_u(1), Q_u(2), \dots, Q_u(i), \dots, Q_u(\Lambda)\}$, where $Q_u(i)$ is the i$^{\text{th}}$ expected reward associated to the model with the penalization hyperparameter $\lambda_i$, $1 \leq i \leq \Lambda$, at the update iteration $u$. From the point of view of multi-arm bandit optimization, the environment is the neural signal input variables. The selection of the models to update is considered as the action and the decoding performance of the updated models on the incoming new batch of data are considered as the reward used to select the models to recalibrate at the next update iteration.

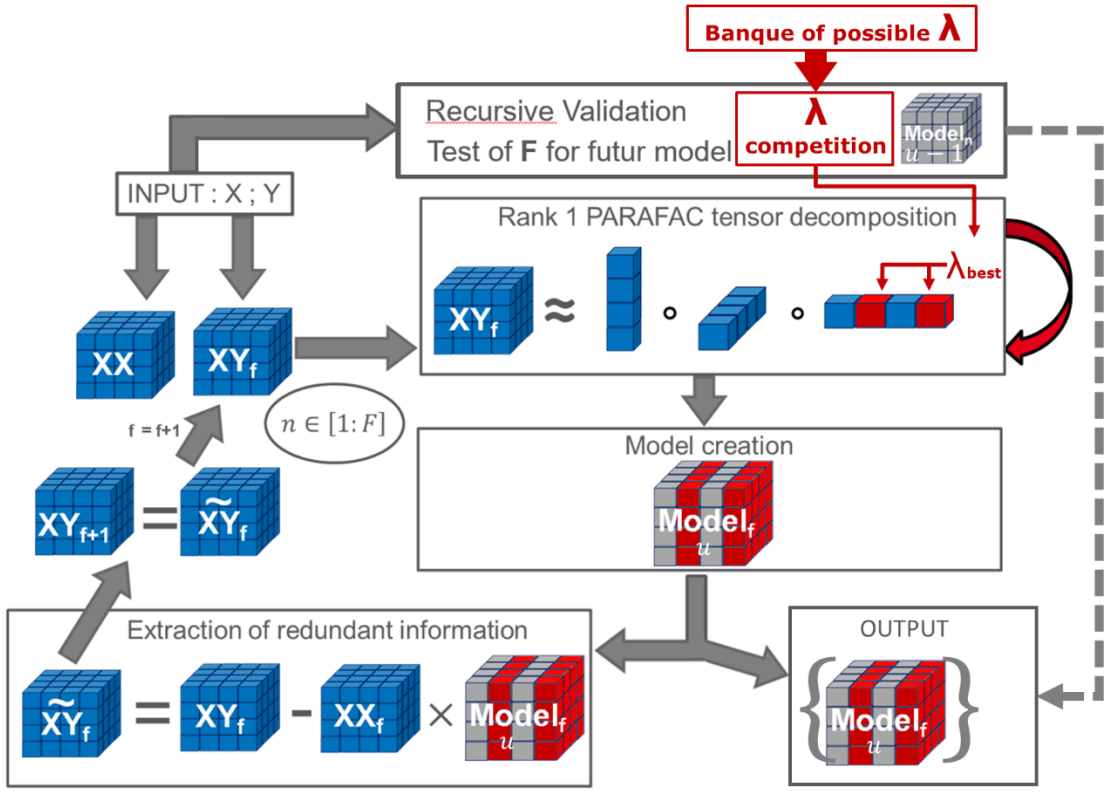



Figure 5-6: Automatic $\lambda$ Penalized REW-NPLS (APREW-NPLS) algorithm. APREW-NPLS algorithm main steps with penalized PARAFAC decomposition leading to slice-wise sparse model with incremental optimization of the best penalization hyperparameter.

### 5.4.2.2. APREW-NPLS model calibration

At each calibration increment $u$, a set of $\Lambda_{updt} \in \mathbb{N}^+$ penalized models $\boldsymbol{\Theta}_u(\boldsymbol{\lambda}_{updt})$ with the associated set of penalization hyperparameters $\boldsymbol{\lambda}_{updt} \subset \boldsymbol{\lambda}$ is selected among the set $\boldsymbol{\Theta}_u(\boldsymbol{\lambda})$. Then, the models contained in the set $\boldsymbol{\Theta}_u(\boldsymbol{\lambda}_{updt})$ are updated. The training procedure of each selected model is similar to the calibration procedure of the PREW-NPLS algorithm and is based on the current tensor block of observation $\{\underline{\mathbf{X}}_u, \underline{\mathbf{Y}}_u\}$ and previously computed models $\boldsymbol{\Theta}_{u-1}(\boldsymbol{\lambda})$ weighted with the forgetting factor $\mu_1$. APREW-NPLS models calibration follows the same strategy than PREW-NPLS algorithm with the exception that, at each iteration $u$, $\Lambda_{updt}$ models with different penalization hyperparameters are updated instead of only one in PREW-NPLS algorithm.The models not belonging to $\boldsymbol{\Theta}_u(\boldsymbol{\lambda}_{updt})$ are not updated and fixed to the same parameter weights than the models in $\boldsymbol{\Theta}_{u-1}(\boldsymbol{\lambda})$. The selection of the models to update, contained in the set $\boldsymbol{\Theta}_u(\boldsymbol{\lambda}_{updt})$, is performed during the Recursive Validation (RV) procedure using the observation tensors $\underline{\mathbf{X}}_u$ and $\underline{\mathbf{Y}}_u$ in parallel to the selection of the optimal latent space

dimension $f^*$ previously explained. The main RV steps selecting the set of models to update at the next calibration increment are represented in Figure 5-7.

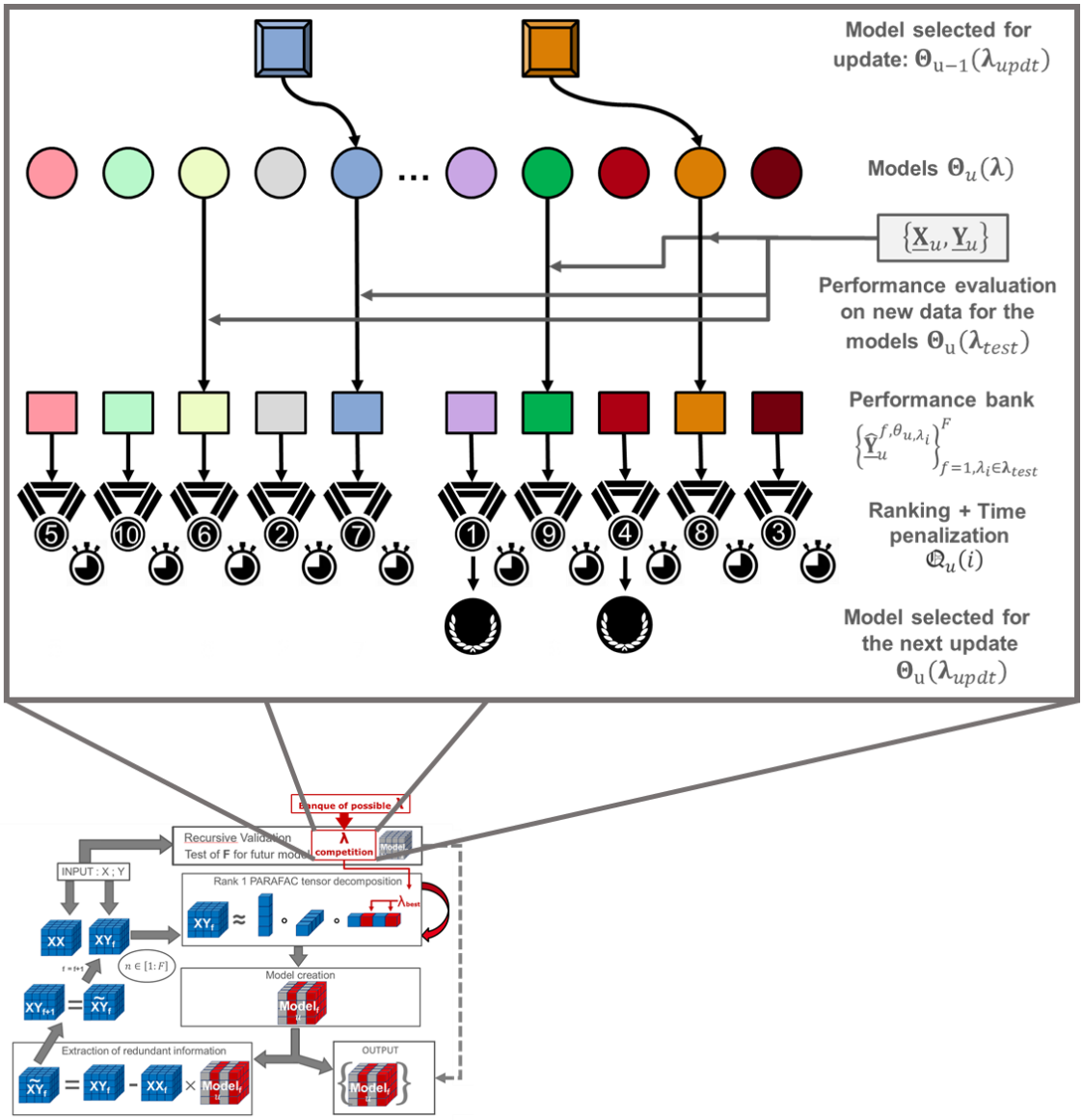



Figure 5-7: Recursive-Validation procedure based on multi-arm bandit problem for penalized model selection optimization. A bank of model performance depending of the penalization hyperparameter is incrementally updated using the Recursive-Validation procedure include in the REW-NPLS algorithm [Eliseyev et al., 2017]. From the performance of each model a ranking of the best models is evaluated depending on a criteria (sparsity, decoding performance, etc.). This model ranking is weighted depending on how long model has not been updated. The models with the best weighted ranks are updated during the next APREW-NPLS algorithm iteration.

### 5.4.2.3. APREW-NPLS Recursive Validation strategy

At each iteration $u$, before the APREW-NPLS model calibration phase, RV procedure evaluates the performance of a set of penalized models referred as $\boldsymbol{\Theta}_{\mathrm{u}}(\boldsymbol{\lambda}_{test})$. A set $\boldsymbol{\lambda}_{test} \subset \boldsymbol{\lambda}$ of penalization hyperparameters groups the $\Lambda_{updt}$ models updated at the previous calibration increment $\boldsymbol{\Theta}_{u-1}(\boldsymbol{\lambda})$ and $\Lambda_{test} \in \mathbb{N}^{+}$ other models selected randomly or based on the expected rewards $\mathbf{Q}_{u-1}$. The performance of the $\boldsymbol{\Theta}_{\mathrm{u}}(\boldsymbol{\lambda}_{test})$ models are evaluated for each latent space dimension similarly to REW-NPLS and PREW-NPLS (see section 4.4.2). The prediction of each penalized model for a specific latent space dimension $f$ is noted $\left\{\underline{\widehat{\mathbf{Y}}}_u^{f,\theta_{u,\lambda_i}}\right\}_{f=1,\lambda_i\in\boldsymbol{\lambda}_{test}}^{F}$. The penalized models prediction for each specific latent space dimension $f$ and each $\lambda_i \in \boldsymbol{\lambda}_{test}$ are compared to the block of training dataset of output observations $\underline{\mathbf{Y}}_u$ to compute an estimated performance index defined as:

$$e_u^{f,\theta_{u,\lambda_i}} = \mu_2\, e_{u-1}^{f,\theta_{u,\lambda_i}} + \varepsilon\left(\underline{\mathbf{Y}}_u, \underline{\widehat{\mathbf{Y}}}_u^{f,\theta_{u,\lambda_i}}\right), \lambda_i \in \boldsymbol{\lambda}_{test},$$

where $\mu_2$ is the forgetting coefficient ($\mu_2 \in [0;1]$) and $\varepsilon$ is a function evaluating the accuracy of the prediction to maximize. For the penalized models that do not belong to the set of tested models $\boldsymbol{\Theta}_{\mathrm{u}}(\boldsymbol{\lambda}_{test})$, $e_u^{f,\theta_{u,\lambda_i}} = e_{u-1}^{f,\theta_{u,\lambda_i}}$, $\lambda_i \notin \boldsymbol{\lambda}_{test}$. For $\lambda_i \in \boldsymbol{\lambda}$, the estimated optimal latent space dimension $f_{\theta_{u,\lambda_i}}^{*}$ of each penalized model $\theta_{u,\lambda_i}$ is defined as $f^{*} = \underset{f}{\operatorname{argmax}}\left(e_u^{f,\theta_{u,\lambda_i}}\right)$ and the associated optimal performance of the models is noted $\mathrm{p}_u(\lambda_i) = e_u^{f^{*},\theta_{u,\lambda_i}}$. From the optimal performance of each model, a reward is granted to each model following the reward function [Feng and Zhang, 2018]:

$$\mathcal{R}_u(i) = ranking\left(\underset{i}{\operatorname{argmax}}\left(\mathrm{p}_u(\lambda_i)\right)\right) - ranking\left(\mathrm{p}_u(\lambda_i)\right), \quad \lambda_i \in \boldsymbol{\lambda}, \mathrm{i} = \{1, \dots, \Lambda\}.$$

Other reward function can be selected depending on the objective of the hyperparameter optimization. The presented reward function is focused on the decoding performance of each penalized model without considering the degree of sparsity of each solution. In the case of embedded BCI system where the model sparsity is a key characteristic, a sparsity indicator may be added to the reward function to select the models depending on this criterion.

Most of the time, the multi-arm bandit problem is applied to stationary data where reward probability distributions do not change over time. Nevertheless, numerous studies highlighted the non-stationarity property of the neural signals and the intra-subject variability due to inattention, habituation, etc. [Clerc et al., 2016b] [Nicolas-Alonso et al., 2015]. Recent rewards are more representative of the current environment (probability distribution) than past rewards. Given $\mu_3 \in [0;1]$ a constant forgetting factor, the expected reward $Q_u(i)$ obtained if the penalized model with a regularization hyperparameter $\lambda_i$ and $\mathrm{i} = \{1, \dots, \Lambda\}$ is updated, is defined as:

$$Q_u(i) = (1-\mu_3)Q_{u-1}(i) + \alpha\mathcal{R}_u(i) = (1-\mu_3)^u Q_1(i) + \sum_{i=1}^{u} \alpha(1-\mu_3)^{u-i}\mathcal{R}_u(i).$$

Multi-arm bandit is an exploitation versus exploration algorithm. The expected reward estimation algorithm $Q_u(i)$ is purely exploitation. Nevertheless, it is possible that less-explored actions lead to better results than the current selected best action. This assumption is even more valid in the case of non-stationary data where each action-reward probability distribution changes. Numerous strategies to include exploration properties were designed. The best-known method is the ε-greedy algorithm which forces the selection of an action among the sub-optimal solution according to $Q_u(i)$ with a probability ε. The ε-greedy algorithm considers the sub-optimal actions as equivalent and selects one of them randomly without difference between nearly greedy actions and unlikely ones [Sutton and Barto, 2017]. The exploitation-exploration expected reward $\mathfrak{Q}_u$ selects a sub-optimal action considering the current expected reward as well as the number of iterations since the last time this action has been selected. Therefore, in the current PhD research, the APREW-NPLS expect reward is modified using this upper-confidence bound following the equation:

$$\mathfrak{Q}_u(i) = Q_u(i) + \upsilon\frac{N_{it} - N_u(i)}{N_{it}}, \quad \text{with i} = \{1, \dots, \Lambda\}.$$

Here, $\upsilon \in \mathbb{R}$ weights the degree of exploration, $N_{it} \in \mathbb{N}^*$ is the number of update iteration realized and $N_u(i) \in \mathbb{N}$ is the index of the last update where the model $i$ has been selected. The selected models that will be updated in the next update $\boldsymbol{\Theta}_u(\boldsymbol{\lambda}_{updt})$ are the models which maximize $\mathfrak{Q}_u$. The exploitation-exploration expected reward $\mathfrak{Q}_u$ is used to select the models to update at each iteration but numerous other strategies could be implemented to choose the models to update depending on various criteria.

## 5.5. Conclusion

In this chapter adaptive incremental penalized decoders were proposed to estimate a group-wise sparse solution for continuous ECoG decoding namely $L_p$-Penalized REW-NPLS (PREW-NPLS) and its extension Automatic $L_p$-Penalized REW-NPLS (APREW-NPLS). PREW-NPLS proposed three different types of norm penalization to estimate a group-wise sparse model during online closed-loop experiments. The created sparse model is more interpretable which is an important feature in the BCI field. Moreover, sparse solution is suited for the integration of models into a system with lower computing power and resources. Sparse models may reduce the calibration period required.

PREW-NPLS requires to fix the type of penalization norm and penalization hyperparameter before the start of the experiment. These mandatory fixed settings are problematic in the case of BCI applications with unknown penalization hyperpamameter and high inter and intra patient variability. Based on autoML strategies, APREW-NPLS algorithm was designed to address the issues raised by PREW-NPLS. Automatic selection of the penalization hyperparameter is performed using a reinforcement learning strategy inspired by the Multi-arm bandit problem. Moreover, APREW-NPLS calibration procedure trains several models in parallel allowing to save time and adapt the optimal penalized model through time/experiments depending on the brain signals changes during online closed-loop experiments.

PREW-NPLS and APREW-NPLS can be integrated in the REW-MSLM algorithms to evaluate the experts and gate models. Integrating such algorithms in the multi-limb REW-MSLM decoder may be interesting to enhance the specificity of each expert model. Indeed, to control a complex multi-limb effector, REW-MSLM decoder uses one REW-NPLS expert per limb to decode the intended movements from the neural signals. Although these models converge independently, they are all based on the same neural signal feature input variables. APREW-NPLS expert creates more specific model which only takes into accounts the neural features which provide relevant decoding information for the concerned limb.

Chapter 6

# Hierarchical HMM-inspired gating model for low-latency state switching

## Contents

The Hidden Markov Model was integrated into the REW-MSLM to sustain strong idle state detection and to ensure the patient's asynchronous control of a multi-limb effector. However, with increasing number of states/classes and, potentially, class imbalance, classifier may struggle to detect all the states with high accuracy. Few studies subjected the benefits of hierarchical static classifiers compared to conventional single-layer decoders [Abascal et al., 2020] [Dong et al., 2017] [Salazar-Ramirez et al., 2019]. While Hierarchical decoders were poorly tested in the BCI field, they presented some advantages in the case of complex multi-class problem. Hierarchical classifiers highlighted benefits in the case of naturally structured state sequences such as idle versus active state classification before to discriminate the neural signals between multiple active states [Abascal et al., 2020] [Bashashati et al., 2007b] [Dong et al., 2017] [Gundelakh et al., 2018] [Gupta et al., 2020] [Hotson et al., 2016a] [Jeong et al., 2020] [Kee et al., 2017] [Murguialday et al., 2007] [Omedes et al., 2017] [Onaran et al., 2011a] [Salazar-Ramirez et al., 2019]. Moreover, as mentioned in previous chapter, dynamic decoding classifiers highlighted interesting properties for EEG, ECoG and MEA neural signals decoding in several online [Darmanjian et al., 2003] [Hotson et al., 2016a] [Kao et al., 2017] [Lisi et al., 2018] [Millan et al., 2004] [Moses et al., 2018] [Obermaier et al., 2001] [Schwemmer et al., 2018] and offline [Antelis et al., 2017] [Bashashati et al., 2017] [Bashashati and Ward, 2017] [J. Cano-Izquierdo et al., 2012] [Delgado Saa and Çetin, 2011] [Dobiáš and Štastný, 2016] [Du et al., 2018] [Hasan and Gan, 2011a, 2011b] [Haselsteiner and Pfurtscheller, 2000] [Onaran et al., 2011a] [Pfeiffer et al., 2018] [Saa and Çetin, 2013, 2012] [Wang et al., 2011] [Williams et al., 2018] [Wissel et al., 2013] [Xie et al., 2018] BCI experiments.

In this chapter, a new gating model, named H2M2, inspired by Hierarchical Hidden Markov Model (HHMM) is introduced in order to improve the state transition responsiveness of the classifier during complex tasks and to design a lower-latency classifier than HMM classifier. This decoder is inspired by the natural structure of the movement. The first section reminds the state of the art of dynamic hybrid models and hierarchical classifiers in the BCI research field, whereas the rest of the chapter focuses on the HHMM definition and the description of the new H2M2 gating algorithm.

## 6.1. Hierarchical approach in BCI, and motor BCI

The hierarchical decoders were poorly studied in the BCI field. The few reported hierarchical classifiers were mainly focused on idle state detection for asynchronous BCIs. Indeed, hierarchical BCI decoders were typically organized in a two layer structure which firstly isolated the idle state from the active states and then applied another classifier to select one of the available active state [Abascal et al., 2020] [Bashashati et al., 2007b] [Hotson et al., 2016a] [Jeong et al., 2020] [Kee et al., 2017] [Murguialday et al., 2007] [Omedes et al., 2017] [Salazar-Ramirez et al., 2019].

In particular, a hierarchical linear classifier designed to control the closure of a robotic hand based on EEG mu-band power modulation was reported [Murguialday et al., 2007]. In [Murguialday et al., 2007], a first classifier discriminated the active and idle states whereas a second one selected between three states (release, maintain, or crush an object in the robotic hand). Similarly, hierarchical architecture was considered for a 5-class problem based on offline EEG neural signal analysis to cluster firstly the idle and action states using unsupervised K-mean algorithm and supervised SVM to distinguish left hand, right hand, tongue or foot imaginary movements [Salazar-Ramirez et al., 2019].

Other less conventional hierarchical decoder architecture were reported in BCI applications in order to combine binary classifier predictions for multi-states classification problems [Lotte et al., 2018]. In [Dong et al., 2017], hierarchical SVM algorithm was designed for offline 4-state classification from EEG neural signals. The first layer of the classifier was composed of four "one versus all" SVM algorithms. If no dominant state was apparent, a second layer composed of six "one versus one" SVMs was activated for thinner classification. In [Gundelakh et al., 2018], online 4-class decoding was reported using EEG recording system. A first layer based on two ANN and two SVM classifiers was applied before to compute a second-classification layer composed of ANN which made the final classification. Finally, 3-class, 4-class, and 5-class offline mental task classifications from EEG signals were tested using Optimal Decision Tree based Support Vector Machine (ODT-SVM) classifier [Gupta et al., 2020]. Based on ECoG dataset, offline classification of finger movements was performed using 10 pair-wise SVM decoders in parallel with hierarchic classification rule [Onaran et al., 2011a].

## 6.2. Dynamic Hierarchical decoders

Dynamic hierarchical decoder family is underrepresented in the BCI field. Only few studies reported the combination of both hierarchical structure and dynamic decoding. Hotsons' study discriminated flexion of individual finger to control in real-time a hand prosthetic effector from ECoG neural signals of an epileptic patient [Hotson et al., 2016a]. The hierarchical classification was performed with two LDA decoders which classified idle versus movement states and individual finger state detection respectively. The binary movement versus idle LDA classifier was given a first-order Markov chain. Additionally, dynamic hierarchical classifier such as HHMM [Saa and Çetin, 2013] [Sugiura et al., 2007] [Suk and Lee, 2010] and hierarchical CRF [Sugiura et al., 2007] were explored during offline EEG-based motor imagery experiments.

Dynamic hierarchical classifier are more widespread in other fields with more evident data structure such as medical applications with electrocardiography [Hu et al., 2012] [Liang et al., 2014] or muscles recordings classification [Malešević et al., 2017], motion/gesture/activity recognition [Aarno and Kragic, 2006] [Asghari et al., 2019] [He et al., 2012] [Kabir et al., 2016] [Kulić and Nakamura, 2010] [Lee and Cho, 2016] [Nguyen et al., 2005] [Solaimanpour and Doshi, 2017] [Wei et al., 2011] [Zhu and Sheng, 2009], handwriting

recognition [Lee et al., 1998], etc. Some applications rely on the modelling of structured data with different stochastic levels, temporal length scales, complexity and behaviors. Taking the example of text recognition, punctuation marks, frequent combination of letters, frequent combination of words and endings of phrases may have different time scales and be represented by different temporal models [Fine et al., 1998].

Taking the example of a disabled patient performing a reach-and-grasp task composed of reaching and grasping movements. To complete the task, the following sequential actions from the same body side should be realized: reach the object, open the hand, turn the wrist to match the hand aperture with the object shape and close the hand. During this sequence, the arm, wrist and grasp states of the same body side are more likely to be activated than the arm, wrist and grasp states of the other arm. Such movement behavior may be represented using a classifier with a hierarchical architecture. Such decoder may reduce the number of misclassifications and improve the decoder responsiveness.

Motor cortex activity on the contraparietal side of the intended movement is higher than ipsilateral side which can be considered as a pseudo neurological architecture to exploit. To our knowledge, there is no study which reported an online dynamic adaptive incremental hierarchical decoder.

Based on these considerations, a HHMM-inspired gating classifier referred as H2M2 was designed. H2M2 classifies the left, right body side intended movements and idle states in a first layer before to cluster finer movements in deeper layers. This gating model was created to improve the classification and speed up the decoding transition.

## 6.3. REW-MSLM with HHMM-inspired gating algorithm

### 6.3.1. Hierarchical Hidden Markov Model (HHMM)

Dynamic decoders (as HMM) describe the extrinsic dynamics of data allowing to model transitions between classes. Hierarchical dynamic decoders represent at the same time the intrinsic structure of each class and their extrinsic dynamic. Hierarchical hidden markov models (HHMM) generalized HMM to a structured multi-level stochastic process [Fine et al., 1998]. Each hidden state is considered to be a self-contained sequential probabilistic model which might generate a sequence of sub-states activation. In other words, each state might activate a sub-HMM with sub-states that might generate another sub-HMM etc. Only specific states or sub-states named production states emit output observations whereas hidden intermediary states (named internal states) are not visible [Fine et al., 1998]. Each sub-HMM has a final state whose activation results in a return to the parent state which activated the sub-model. Transition between states of the same

sub-model is referred as horizontal transition whereas diving into lower sub-state or turning back to a parent state is called vertical transition.

Let $D \in \mathbb{N}^*$ be the number of hierarchical layers of HHMM, layers are indexed by $d \in \{1, \dots, D\}$, $H^d \in \mathbb{N}^*$ be the numbers of sub-HMMs of the hierarchical level $d$, sub-HMMs of the hierarchical level $d$ are indexed by $h \in \{1, \dots, H^d\}$: $\text{HMM}^{h,d}$, $K^{h,d} \in \mathbb{N}^*$ be the number of state in each $\text{HMM}^{h,d}$, $d \in \{1, \dots, D\}$, $h \in \{1, \dots, H^d\}$. States of the HHMM at a hierarchical level $d$ in a sub-model $h$ with a sub-index $k \in \{1, \dots, K^{h,d}\}$ are noted as $s^{k,h,d}$. The set of HMMM states $S = S_{int} \cup S_{prod}$ is the union of the set of production state $S_{prod}$, and the set of internal states $S_{int}$.

An example of HHMM is shown in Figure 6-1 with 3 layers, $D = 3$, single sub-HMM at the two first layers $H^1 = H^2 = 1$, and two sub-HMMs at the third layer, $H^3 = 2$. In this example, the set of production states is $S_{prod} = \{s^{1,1,1}, s^{4,1,2}, s^{3,1,3}, s^{1,2,3}\}$ whereas others states are grouped in the set of internal states $S_{int}$.

The next section introduced the variant of Hierarchical HMM structure, referred as H2M2, proposed for hierarchical gating in REW-MSLM algorithm.

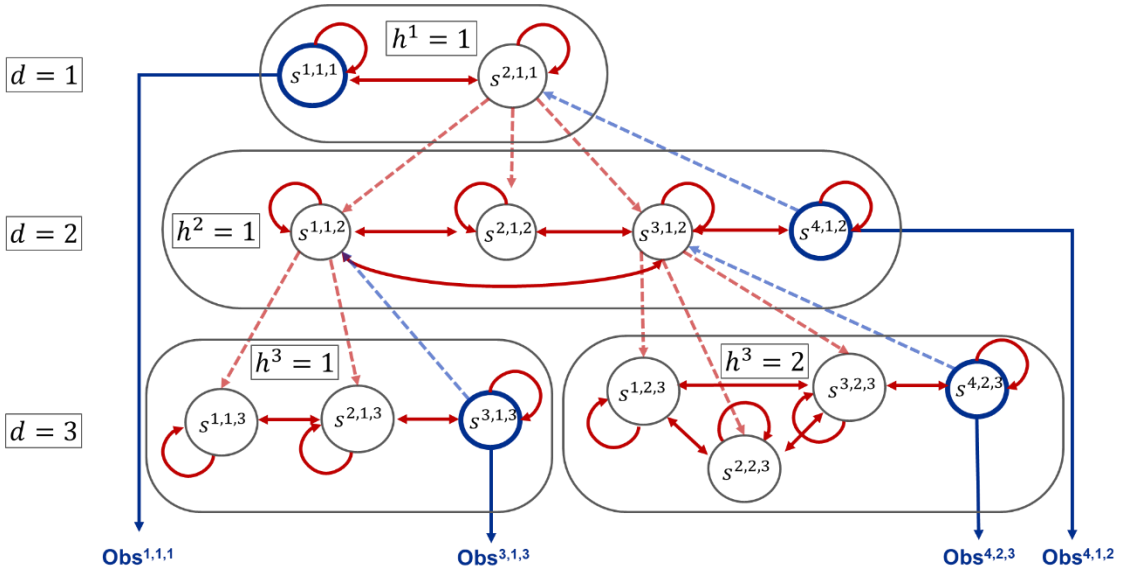



Figure 6-1: Hierarchic hidden markov model structure. Each state from a layer $d$ is a self-contained sequential probabilistic model. The activation of an internal state leads to a vertical transition toward a lower layer. In a layer, horizontal transition between the state from a sub-HMM are performed similarly as state with HMM. If a production state is activated, as observation is generated and an upward vertical transition is performed to the related internal state. Sub-HMMs are represented with the grey ovoid shapes. Internal states are presented with grey circles whereas production states are in blue. Horizontal and downward vertical transitions are colored in red whereas upward vertical transition are in blue.

## 6.3.2. General H2M2 parameters description

Similarly to conventional HHMM, each state of H2M2 is considered to be a self-contained sequential probabilistic model which might generate a sequence of sub-state activations. Each state might activate a sub-HMM with sub-states that might generate another sub-HMM etc. Output observations are only emitted by production states and internal states do not emit visible observation.

Contrary to conventional HHMM, in H2M2, each sub-HMM is independent and change with the brain neural signals. Additionally, every state from the last layer is considered as a production state. Therefore, at each time step, vertical and horizontal transition probabilities inside each sub-HMM are evaluated.

At each time step, one of the production states is activated and emits the output observations. Similarly to conventional single level HMM, active production state is defined by latent variable $z_t \in S_{prod}$.

For each sub-HMM, denoted as $\text{HMM}^{h,d}$, $d$ is hierarchical level, $h$ is the index of sub-HMM at a given hierarchical level, a state transition probability matrix is noted as $\mathbf{A}^{h,d} = (a_{ij}^{h,d}) \in \mathbb{R}^{K^{h,d} \times K^{h,d}}$. Here $a_{ij}^{h,d} = p\left(s^{i,h,d} \middle| s^{j,h,d}\right)$ is the probability of making a horizontal transition from the $i^{th}$ state to the $j^{th}$ [Fine et al., 1998]. Equivalently to HMM, the initial state probability of each state is defined as $\pi^{k,h,d}$. Finally, for the production states, $\{c_s\}_{s \in S_{prod}}$is the set of parameters employed to estimate the conditional emission probability of the observed variables $p\left(\underline{\mathbf{X}}_t \middle| z_t\right)$.

At time $t$, the active production state is defined by the variable $z_t \in S_{prod}$. Other states (intern) may be active at precedent layers. A function $\Psi: S \to S$ designed to determine the active inner states in the upper layers leading to the active production state is defined as follows. For every active state $s^{k,h,d}$ at the layers $d > 1$, $\Psi(s^{k,h,d})$ is the active state at the precedent layer $d-1$ considered in the $\text{HMM}^{h,d-1}$ leading to $s^{k,h,d}$. For the first layer states: $\Psi\left(s^{k,h,1}\right) = s^{k,h,1}$ if $d = 1$. For active production state $z_t \in S_{prod}$ at time $t$, the states $\Psi(z_t)$, $\Psi^2(z_t) = \Psi(\Psi(z_t))$, $\Psi^3(z_t)$, etc. form a whole set of active sates at time $t$: $\mathcal{Z}_t = \cup_{j=0}^{D}\{\Psi^j(z_t)\} = \{z_t, \Psi(z_t), \Psi^2(z_t), \dots, \Psi^D(z_t)\}$. Here $\Psi^0(z_t) = z_t$.

The set $\mathcal{Z}_t$ describes the path of all the activated intern states leading to the production states $z_t \in S_{prod}$ at time $t$. For example, for a production state $z_t$ at the third layer, $d = 3$, $\mathcal{Z}_t = \{z_t, \Psi(z_t), \Psi^2(z_t)\}$, $card(\mathcal{Z}_t) = 3$. For a production state $z_t$ at the first layer, $d = 1$, $\mathcal{Z}_t = \{z_t\}$, $card(\mathcal{Z}_t) = 1$. The function $card()$ is the function estimating the cardinality of the set $\mathcal{Z}$. The cardinality is defined as the "number of elements" for a finite set.

For H2M2 the activation probability of the production state $z_t = s$ is defined as :

$$\gamma_{s,t} = p(z_t = s|\underline{\mathbf{X}}_{1:t}) = \prod_{j=0}^{card(Z_t)} p(\Psi^j(z_t)|\underline{\mathbf{X}}_{1:t}), \qquad s \in S_{prod}. \tag{6.3.1}$$

To simplify the notation and to be closer to the BCI application presented in future chapters, in the next section, the H2M2 is considered in the specific case where $z_t \in S_{prod}$ with $card(S_{prod}) = 5$ production states. The architecture of the considered H2M2 algorithm is represented in the Figure 6-2. Nevertheless, all the presented methods, algorithms and strategies can be generalized to any H2M2 architecture.

In this particular case, the H2M2 architecture is split in $D = 2$ layers composed of 3 sub-HMMs with the number of sub-HMM at the layer $d = 2$ is $H^d = 2$. This example is made of five production states $S_{prod} = \{s^{1,1,1}, s^{1,1,2}, s^{2,1,2}, s^{1,2,2}, s^{2,2,2}\}$ and two internal states $S_{int} = \{s^{2,1,1}, s^{3,1,1}\}$. The transition matrix $\mathbf{A}^{1,1} \in \mathbb{R}^{3\times3}$ describes the transition between the states $s^{1,1,1}, s^{2,1,1}, s^{3,1,1}$ whereas $\mathbf{A}^{1,2} \in \mathbb{R}^{2\times2}$ and $\mathbf{A}^{2,2} \in \mathbb{R}^{2\times2}$ evaluate the transitions between $s^{1,1,2}, s^{2,1,2}$ and $s^{1,2,2}, s^{2,2,2}$ respectively. A schematic of the considered HHMM in represented in Figure 6-2.

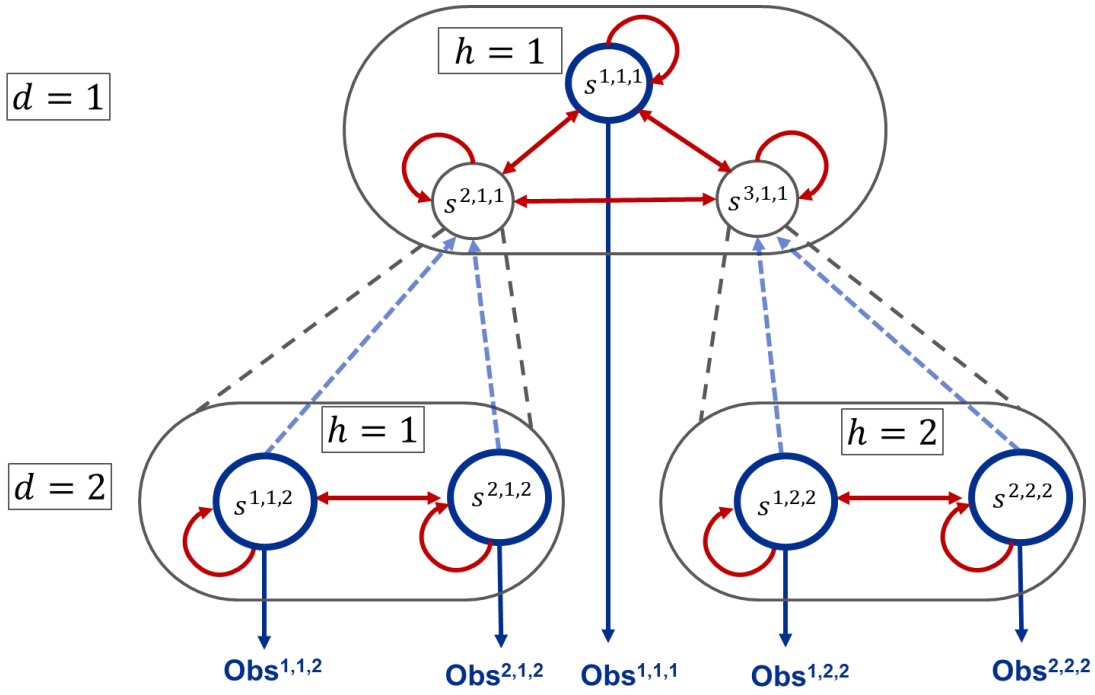



Figure 6-2: Structure of the proposed H2M2 algorithm. Example of H2M2 architecture for the specific application case of two layers, three sub-models with two of them in the second layer, five observation states and two internal states. Sub-HMMs are represented with the grey ovoid shapes. Internal states are presented with grey circles whereas production states are in blue. Horizontal and downward vertical transitions are colored in red whereas upward vertical transition are in blue.

### 6.3.3. H2M2 online incremental training

Similarly than the HMM gating parameters training in the section 4.5.2, at each update $u$, H2M2 gating parameter estimation is updated based on the update block dataset $\{\underline{\mathbf{X}}_u, \mathbf{z}_u\}$ where $\underline{\mathbf{X}}_u \in \mathbb{R}^{\Delta L \times I_1 \times \ldots \times I_M}$, $\mathbf{z}_u = \left(z_{t_1}, \ldots, z_{t_1+\Delta L}\right)^T \subset \mathbb{N}^{*\Delta L}$ and $\Delta L$ the update block size. H2M2 transition matrix $\mathbf{A}^{1,1}$, $\mathbf{A}^{1,2}$ and $\mathbf{A}^{2,2}$ are approximated by counting the successive transition of states in $\mathbf{z}_u$ and the transition matrix estimated during the previous updates weighted with the forgetting factor $\mu_g$ , $0 \leq \mu_g \leq 1$.

For the matrix $\mathbf{A}^{1,1}$ which contains internal (non-production) states $S_{int} = \{s^{2,1,1}, s^{3,1,1}\}$, it is considered that the internal states $s^{2,1,1}$ or $s^{3,1,1}$ are activated and considered for the gating parameter estimation if $s^{2,1,1} \in \mathcal{Z}_t$ or $s^{3,1,1} \in \mathcal{Z}_t$ respectively.

H2M2 conditional emission probability $p(\underline{\mathbf{X}}_t|z_t)$ is inferred through the combination of $p(\Psi^j(z_t)|\underline{\mathbf{X}}_t)$ and their class prior $p\left(\Psi^j(z_t)\right)$ using Bayes' theorem [Bishop, 2006], with $j = 0 \ldots D$. Three REW-NPLS discriminative decoders are embedded into the H2M2-based gating process to evaluate each sub-model probability . Each sub-model is trained independently on the observation tensor of input variables $\underline{\mathbf{X}}_u$ and the latent state dummy variable matrix $\mathbf{Z}_u \in \{0,1\}^{card(S_{prod}) \times \Delta L}$ where the column-wise (single) non-zero element depicts the activated state for each sample.

H2M2 conditional emission probability is inferred similarly than HMM in the section 4.5.2. However, instead of calibrating one set of $F$ multilinear models $\theta_g^f = \{\underline{\mathbf{B}}_u^f, \mathbf{b}_u^f\}_{f=1}^F$ to evaluate the conditional emission probabilities $p(\underline{\mathbf{X}}_t|z_t)$, a model is evaluated for each sub-HMM: $\text{HMM}^{h,d}$. In the presented case, a group of three multilinear models are calibrated $\theta_g = \{\theta_g^{f,1,1}, \theta_g^{f,1,2}, \theta_g^{f,2,2}\}$ (Figure 6-3). The REW-NPLS discriminative algorithm computes the three set of $F$ multilinear models $\theta_g^{f,h,d} = \{\underline{\mathbf{B}}_u^{f,h,d}, \mathbf{b}_u^{f,h,d}\}_{f=1}^F$, where $\underline{\mathbf{B}}_u^{f,h,d}$ and $\mathbf{b}_u^{f,h,d}$ are the tensor of parameters and related bias of the sub-model with index $h$ in the $d$ layer.

The Recursive-Validation procedure selects the best models based on the estimated optimal gating hyperparameter (the latent variable space dimension $f$) for each sub-model $f_g^{h,d*} \leq F$ and defines the optimal gating sub-models as $\theta_g^{h,d} = \{\underline{\mathbf{B}}^{h,d}, \mathbf{b}^{h,d}\} = \left\{\underline{\mathbf{B}}_u^{f_g^{h,d*},h,d}, \mathbf{b}_u^{f_g^{h,d*},h,d}\right\}$ for dynamic gating weight $\gamma_{s,t}$ estimation.

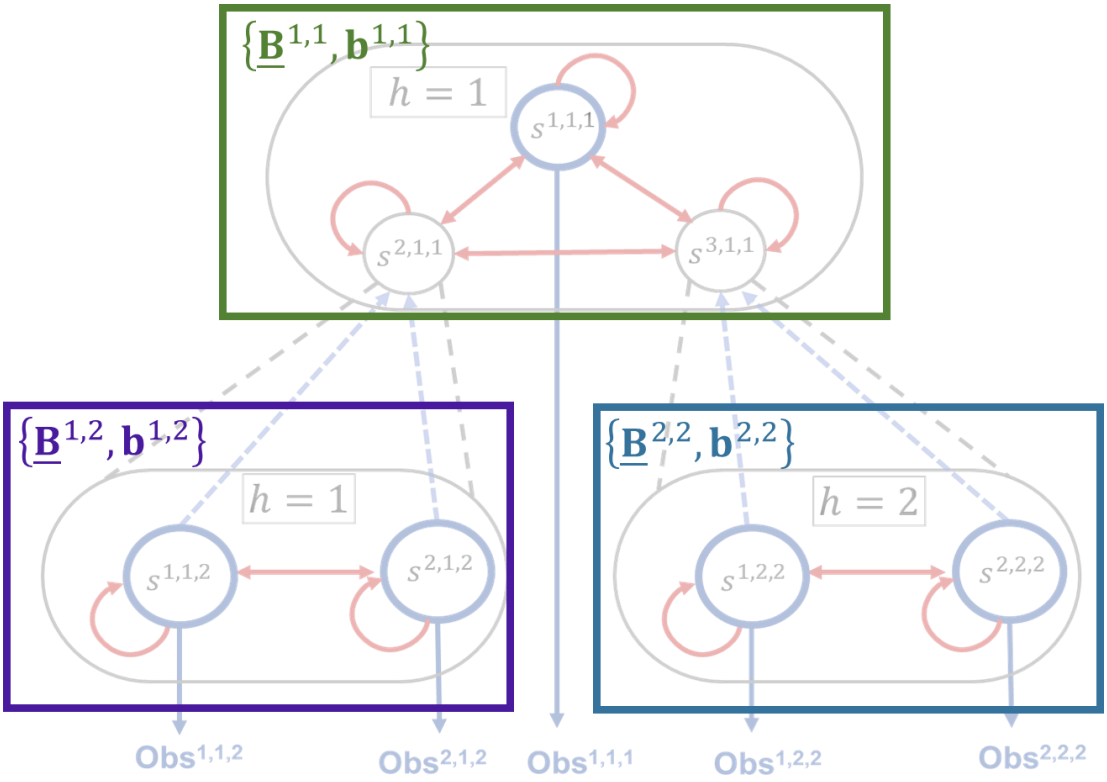



Figure 6-3: Illustration of the H2M2 sub-models. Each sub HMM: $HMM^{h,d}$ is considered as independent model to evaluate the conditional emission probability for each state of the sub-models. The $HMM^{h,d}$ conditional emission probability are estimated based on REW-NPLS algorithms. The optimal latent space dimensions are found for each sub-HMM independently using the Recursive Validation procedure of REW-NPLS algorithm.

## 6.3.4. Online H2M2 application

The variable $\gamma_{s,t} \in \mathbb{R}$ determines how likely the production state $s \in S_{prod}$ is generated based on the current observation variable $\underline{\mathbf{X}}_t$. Let define $\hat{\boldsymbol{s}}^{k,h,d}$ the prediction of the REW-NPLS discriminative sub-models $\theta_g^{h,d}$ and $p(\Psi^d(z_t)|\underline{\mathbf{X}}_t)$ the activation probability of each sub-state in the considered sub-HMM: $\text{HMM}^{h,d}$. $p(\Psi^d(z_t)|\underline{\mathbf{X}}_t)$ is evaluated independently for the three sub-HMM $\text{HMM}^{h,d}$ before to compute $p(\Psi^d(z_t)|\underline{\mathbf{X}}_{1:t})$ using H2M2 forward algorithm:

$$j = 0, \dots, D,$$

$$\hat{\boldsymbol{s}}^{h,d} = \underline{\mathbf{B}}^{h,d}\, \underline{\mathbf{X}}_t + \mathbf{b}^{h,d},$$

$$p(\Psi^j(z_t) = s^{k,h,d-j}|\underline{\mathbf{X}}_t) = \frac{\exp(\hat{s}^{k,h,d-j})}{\sum_{i=1}^{K^h} \exp(\hat{s}^{i,h,d-j})},$$

$$p(\Psi^j(z_t) = s^{k,h,d-j}, \underline{\mathbf{X}}_{1:t}) = p(\underline{\mathbf{X}}_t|\Psi^j(z_t) = s^{k,h,d-j}) \sum_{i=1}^{K^h} a_{k,i}^{h,d}\, p(\Psi^j(z_{t-1})|\underline{\mathbf{X}}_{1:t-1}),$$

$$p\left(\Psi^j(z_t) = s^{k,h,d-j} \middle| \underline{\mathbf{X}}_{1:t}\right) = \frac{p\left(\Psi^j(z_t) = s^{k,h,d-j}, \underline{\mathbf{X}}_{1:t}\right)}{\sum_{i=1}^{K^h} p\left(\Psi^j(z_t) = s^{i,h,d-j}, \underline{\mathbf{X}}_{1:t}\right)}.$$

Finally, the production state probability $\gamma_{s,t} = p\left(z_t = s^{k,h,d} \middle| \underline{\mathbf{X}}_{1:t}\right)$ with $s^{k,h,d} \in S_{prod}$ is evaluated by mixing the sub-state probabilities of the internal states $S_{int} = \{s^{2,1,1}, s^{3,1,1}\}$ with the related production states probabilities estimated in the sub-models. Following the equation (6.3.1), $\gamma_{s,t}$ can be expressed for each production state as:

$$\gamma_{s^{1,1,1},t} = p\left(\Psi^0(z_t) = s^{1,1,1} \middle| \underline{\mathbf{X}}_{1:t}\right),$$

$$\gamma_{s^{1,1,2},t} = p\left(z_t = s^{1,1,2} \middle| \underline{\mathbf{X}}_{1:t}\right) = p\left(\Psi^0(z_t) = s^{1,1,2} \middle| \underline{\mathbf{X}}_{1:t}\right) p\left(\Psi^1(z_t) = s^{2,1,1} \middle| \underline{\mathbf{X}}_{1:t}\right),$$

$$\gamma_{s^{2,1,2},t} = p\left(z_t = s^{2,1,2} \middle| \underline{\mathbf{X}}_{1:t}\right) = p\left(\Psi^0(z_t) = s^{2,1,2} \middle| \underline{\mathbf{X}}_{1:t}\right) p\left(\Psi^1(z_t) == s^{2,1,1} \middle| \underline{\mathbf{X}}_{1:t}\right),$$

$$\gamma_{s^{1,2,2},t} = p\left(sz_t =^{1,2,2} \middle| \underline{\mathbf{X}}_{1:t}\right) = p\left(\Psi^0(z_t) = s^{1,2,2} \middle| \underline{\mathbf{X}}_{1:t}\right) p\left(\Psi^1(z_t) = s^{3,1,1} \middle| \underline{\mathbf{X}}_{1:t}\right),$$

$$\gamma_{s^{2,2,2},t} = p\left(z_t = s^{2,2,2} \middle| \underline{\mathbf{X}}_{1:t}\right) = p\left(\Psi^0(z_t) = s^{2,2,2} \middle| \underline{\mathbf{X}}_{1:t}\right) p\left(\Psi^1(z_t) = s^{3,1,1} \middle| \underline{\mathbf{X}}_{1:t}\right).$$

### 6.3.5. H2M2 gating integration in REW-MSLM

The dynamic gating introduced with REW-MSLM is replaced by a dynamic gating evaluated with the H2M2 algorithm. Therefore, $\underline{\mathbf{Y}}_t$ is estimated as follows:

$$\underline{\hat{\mathbf{Y}}}_t = \sum_{s \in S_{prod}} \gamma_{s,t} \left(\underline{\mathbf{Beta}}_s \, \underline{\mathbf{X}}_t + \underline{\mathbf{bias}}_s\right).$$

Here, $\underline{\mathbf{Beta}}_s$ and $\underline{\mathbf{bias}}_s$ are the expert tensor parameters related to the production state $s \in S_{prod}$ and its associated bias. $\gamma_{s,t} = p\left(z_t = s \middle| \underline{\mathbf{X}}_{1:t}\right)$ is the dynamic gating weight coefficient at time $t$ of the expert assigned to the production state $s \in S_{prod}$. REW-MSLM models are entirely defined through the experts' parameters $\theta_e = \left\{\underline{\mathbf{Beta}}_s, \underline{\mathbf{bias}}_s\right\}_{s \in S_{prod}}$ and the H2M2 models' parameters $\theta_g$.

As the application and the incremental training of the expert models are strictly identical than the training procedure described in section 4.5, experts evaluation is not detailed in the following section which are more focus on the evaluation of the HHMM parameters.

## 6.4. Conclusion

In this chapter, a new gating decoder has been introduced to improve the decoding performance of the REW-MSLM gate in the case of asynchronous complex state experiments using the natural prior knowledge related to movement discrete state sequences. The new gating model inspired by the Hierarchic Hidden Markov Model (HHMM) relies on the natural movement structure to improve the state classification and recognition. Instead of modelling the state probability with one model, state probability estimation is divided into sub-models to represent more accurately the

different stochastic levels and time scales of the brain neural signals. This new gating model is based on the combination of adaptive incremental linear models. H2M2 was designed for real-time calibration and application during online closed-loop experiments.

Chapter 7

# Experimental set-up and data

## Contents

The performance of the algorithms were evaluated online with multiple closed-loop experiments performed during the CLINATEC clinical trial or using pseudo-online simulations using the dataset recorded during previous closed-loop experiments. All the experiments were designed to stress the asynchronous multi-limb decoding performance and stability of the designed algorithms compared to other decoders based on chronic ECoG neural signals. This chapter firstly presents the BCI platform set up used during the experiments. Then it introduces the data analyzed in this study.

## 7.1. Recording set up

WIMAGINE is an active implantable medical device able to record epidural ECoG on 64 electrodes. The digitized epidural ECoG data were radiotransmitted to a custom designed base station connected to a computer [Mestais et al., 2015]. During the experimental sessions, 32 electrodes for each implant were selected in a checkerboard pattern because of limited data rates, caused by restricted radio link [Benabid et al., 2019] (Figure 7-1). Epidural ECoG signals were recorded at a sampling rate of fs= 586 Hz.

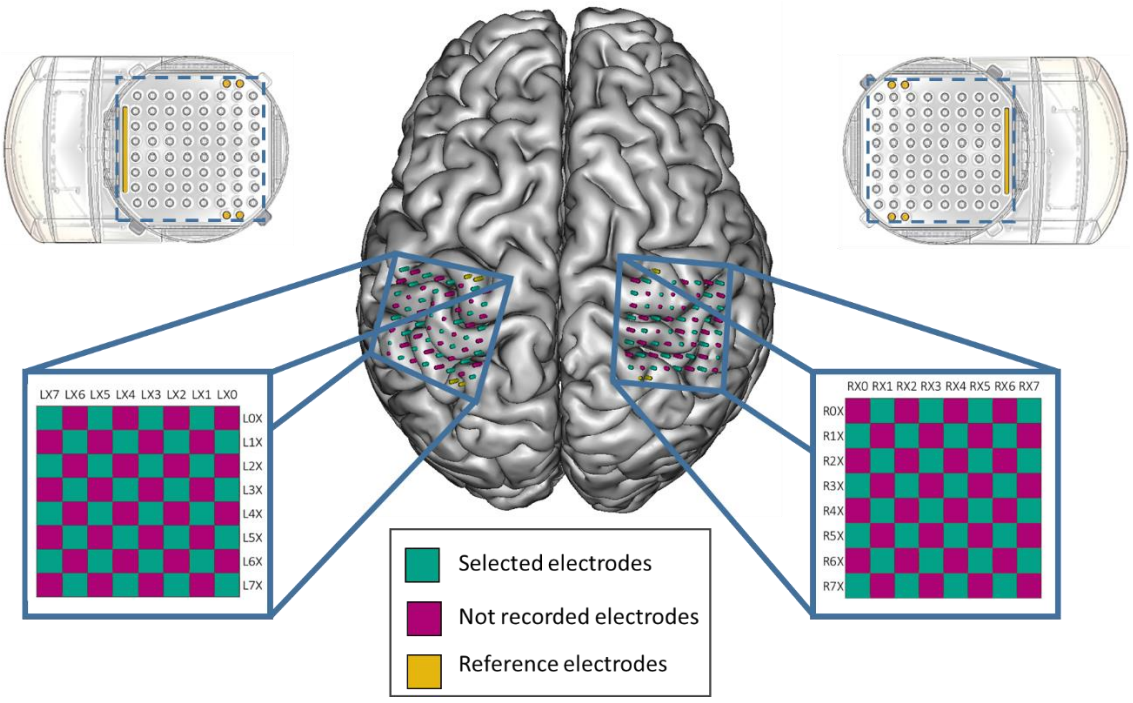



Figure 7-1: WIMAGINE is an active implantable epidural ECoG recording system composed of 64 electrodes. Two WIMAGINE were implanted above specific region of the patient's motor cortex. Due to limited data rate, half of the electrodes for each implant were selected in a checkerboard pattern to cover a large brain area. The selected electrodes for the neural signal decoding experiments are colored in green whereas not recorded ones are shown in purple. The same electrodes were selected for all the experiments presented in this manuscript.

## 7.2. Effector control

During laboratory experiments, the patient was strapped into the enhancing mobility (EMY) exoskeleton. A computer station receiving ECoG radio-emitted signals was embedded in the back of the exoskeleton. The neural signals samples were decoded and translated into incremental end-point-control commands through the decoding software. Finally, the control commands were converted into joints movement by the exoskeleton control system activating the limbs and producing the appropriate movements. The virtual avatar was a virtual replica of the exoskeleton and was used for home-based training. For both laboratory and home-based experimental sessions, the patient was sitting down in the exoskeleton or his wheelchair respectively. During the laboratory experiments, the LED panel was placed in front of the patient to provide him the task instructions. At home, a television broadcasted the instructions and the virtual avatar movements. The virtual environment and avatar were presented with a first person view. The task success feedback was provided to the patient. During laboratory experiments, the lightened LEDs showing the target to reach were manually switched off by the experimenter when the task was completed. For wrist rotation tasks, clockwise or counter clockwise successive flashing LEDs informed the patient with the task to achieve. Flashing LEDs were manually turned off when the task was completed. During the experiments with the virtual environment, the target to reach during an arm translation task automatically turned from red or blue colour (left or right arm) to green colour when the task was completed. During a wrist rotation task continuous feedback was provided to the patient through a gauge which turned green for successful tasks. The patient was allowed to move and talk freely during the training and test experiments in order to create models robust artefacts related to muscular activities.

## 7.3. Experiments design

Numerous experiments with various tasks were carried out during the "BCI and Tetraplegia" clinical trial. This section is only focused on the experiments performed within the framework of the PhD research.

The experiments studied in the manuscript were performed between March 5$^{th}$, 2018 and January 17$^{th}$, 2020. Each experiment/session was composed of successive tasks decided by an experimenter. All the experiments related to the PhD thesis were asynchronous alternative multi-limb/bimanual experiments and relied on the same experimental structure. Each task corresponded to one of the available state between the idle state (IS) and the active states (AS). During IS, no target was presented to the patient. The patient had to remain in a non-active state until a new task started. AS tasks regroup all the intended movement tasks that should be performed by the user. Depending on the experiments, diverse AS tasks with various complexity (controlled dimension) were

proposed and executed. The AS tasks controlled in the experiments presented in this manuscript were the translation of the left ($AS_{LH}$) and right ($AS_{RH}$) hand in the 3D space and the 1D angular rotation of the left ($AS_{LW}$) and right ($AS_{RW}$) wrist (Figure 7-2).

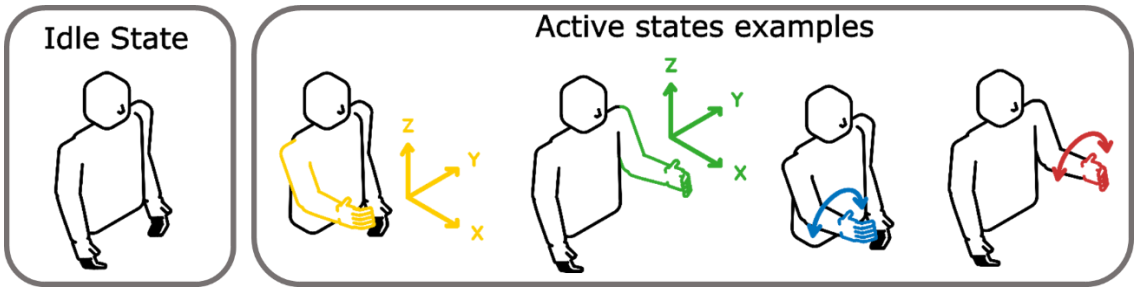



Figure 7-2: Available degree of freedom controlled by the patient in the experiments. The possible movements are translation of the left and right hand in the 3D space and the 1D angular rotation of the left and right wrist. An additional idle state is always available in the experiments. All the experiments performed and analyzed in the manuscript are asynchronous alternative multi-limb/bimanual experiments relying on the combination of several of the presented available movements.

Each task was made of several successive trials where the patient attempted to reach a target location set sequentially with the left or right hand or to rotate the left or right wrist until a given angle. During a session, the hand position was not reset by the system between state, task and trials. For a given AS, the starting position of the hand for a trial was the position of the hand at the end of the previous trial of the same AS. An example of a session with three tasks IS, $AS_{LH}$ and $AS_{RH}$ and 4D continuous decoding (2D for each AS) is shown in Figure 7-3. In the Figure 7-3, the AS tasks are represented in 2D space during an alternative 2D left and right hand translation experiment for ease of understanding but the same experimental paradigm can be generalized to 3D space environment and tasks requiring 3D movements. Seven trials from two non-consecutive $AS_{LH}$ are represented with the position of the cursor at the beginning of the second $AS_{LH}$ task equal to the position of the cursor at the end of the first $AS_{LH}$ task.

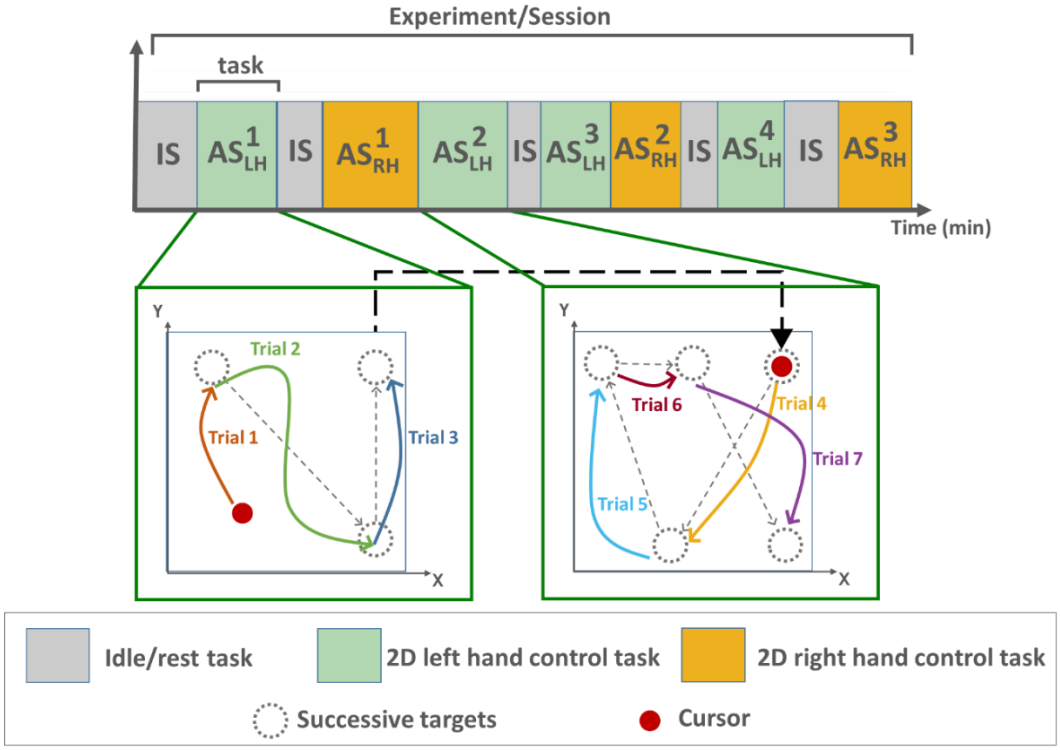



Figure 7-3: Examples of 4D alternative multi-limb pursuit tasks. One session is composed of successive tasks. In this 4D session example, the experiments is composed of three different tasks: 2D left and right hands translation task (referred as $AS_{LH}$ and $AS_{RH}$) and idle task (IS). Each active task is composed of several trials in which the 2D cursor must reach the proposed targets. The index of the AS tasks is noted with a superscript index, the first $AS_{LH}$ task is noted $AS_{LH}^{1}$ whereas the second is referred as $AS_{LH}^{2}$. The cursor position is not reset between tasks, during task and during idle state. The last cursor position is not reset and correspond to the position of the cursor before the patient changes its mental state and the model decodes the state transition.

The session example depicted in Figure 7-3 is referred as an asynchronous alternative bimanual point-to-point pursuit experiments. A point-to-point pursuit task is more complicated than the commonly reported center-out experiment which reset the cursor to an initial stating position at the end of each trial. Pursuit task is characterised by a better exploration of the control space with multiple possible starting points.

All the closed-loop experiments presented in the manuscript were based on this experimental paradigm (in the 3D space). During a session, the patient aimed to reach the proposed targets or rotate the wrist to specific angle by controlling an exoskeleton or a virtual avatar. 22 targets were 3D symmetrically distributed in two cubes in front of the patient with 11 targets per hand for the exoskeleton reaching tasks and were virtually reproduced for the avatar training and testing sessions.

## 7.4. Dataset specification

All the experiments were based on multiple alternative pursuit tasks sessions. However, several specific experiments were designed to highlight the benefits of the proposed algorithms. This section introduced all the experiments used for offline and online performance evaluation. During these experiments, an additional specification to the experimental paradigm was added. Before any transition from an AS task to another AS task (not between trials from the same AS task), an IS task was always imposed. This paradigm forced the patient to control with high accuracy the idle state and enhanced the asynchronous characteristic of proposed BCI experiments. All the achieved BCI experiments were in closed-loop sessions. Therefore for each dataset, the online decoder used to decode the patient's neural signals during the online closed-loop experiments is specified.

### 7.4.1. Online closed-loop 6D experiments using REW-NPLS decoder

Alternative bimanual 3D left and right hand translation pursuit tasks using the virtual avatar was achieved during the intermediate stage of the clinical trial. During these experiments, the patient controlled in real-time 6 dimensions (6D) clustered in $z = 3$ states: idle (IS), left ($AS_{LH}$) and right ($AS_{RH}$) hand control states using the REW-NPLS incremental adaptive decoder. All the experiments were closed-loop sessions recorded between March and June 2018. Three different training/testing paradigms were appraised during these online closed-loop experiments.

First, the series of sessions titled A ($n = 5$) was carried out to evaluate the performance of the algorithm with a small training dataset. The sessions from the series A were self-contained experiments. The decoding models were independently created (initialized to zero), trained and tested during the same experiment. The sessions of the series B ($n = 4$) were used to evaluate the importance of cross-session training. The models were initialized to zero in the first session. Then, the models created during the previous sessions were used to initialize the model parameters of the next experiment. Finally, the experiments of the C series ($n = 5$) were performed to evaluate the robustness of a model calibrated using cross-session training. The experiments from C series were carried out from 9 days to 28 days after the last experiments of the series B and thus, after the last model calibration. The model estimated during the last experiment of the B series was used as neural signal decoder in the C series. A representation of the three training/testing paradigms is shown in Figure 7-4.

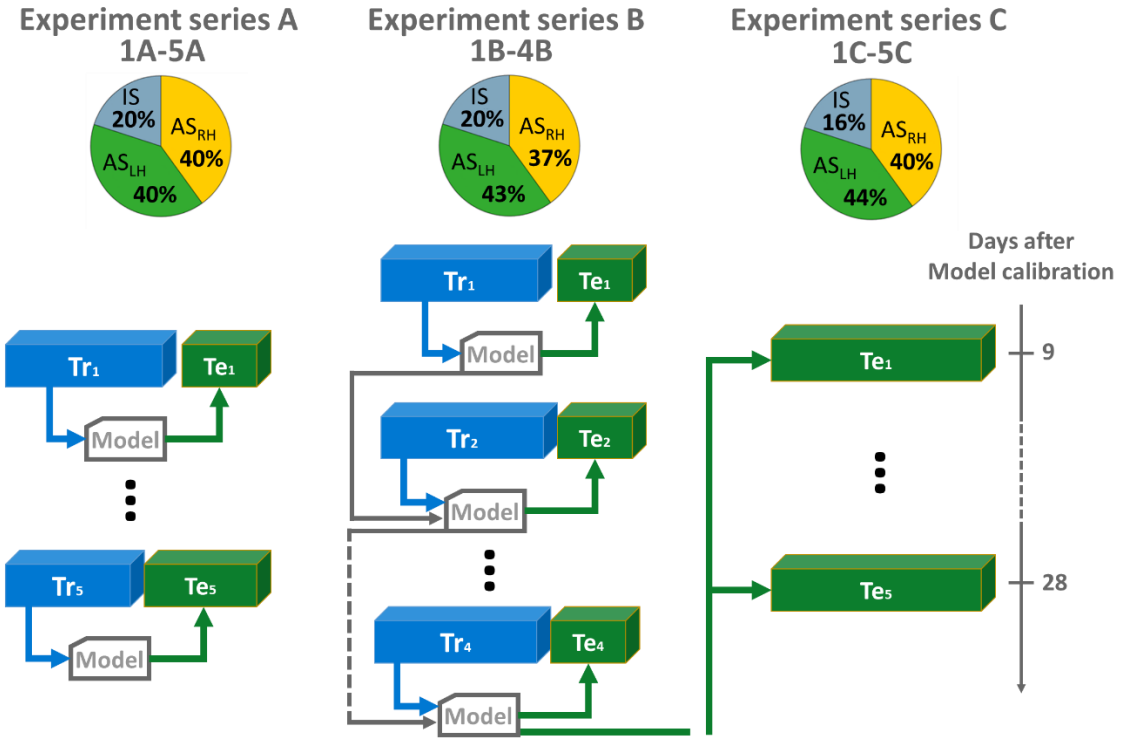



Figure 7-4: Representation of the three paradigms designed for the pseudo-online REW-MSLM decoder evaluation. REW-MSLM decoder was evaluated on three different experiment paradigms: model calibrated from scratch at the beginning of each session with small training dataset (paradigm A), model adaptation with multiple recalibration sessions (paradigm B), and fixed model without adaptation using model created with paradigm B (paradigm C). The three paradigms provided indication on the online closed loop behaviour of the model at the beginning (A), during (B) and after (C) model calibration period. The dataset related to the paradigms A, B and C are composed of 5, 4 and 5 experiments respectively.

The registered dataset of the three series A, B, C were used in order to perform further algorithms performance comparison in pseudo-online studies. Particularly, these series were used to compare the decoding performance of the REW-NPLS algorithm and REW-MSLM.

## 7.4.2. Online closed-loop 8D experiments using REW-MSLM decoder

The patient performed real-time asynchronous closed-loop 8D experiments using the REW-MSLM incremental adaptive closed-loop decoder. The session clustered 3D alternative two-handed reaching tasks ($AS_{LH}$ and $AS_{RH}$), 1D wrist rotation movements for each hand ($AS_{LW}$ and $AS_{RW}$) and idle state (IS) for a total of $z = 5$ states and 8 continuous dimensions [Benabid et al., 2019]. The number of experts was fixed to N=2 with one expert associated to left body-side limb decoding whereas the other estimated the right body-side

limb model. The hand and wrist continuous movements from the same body side were decoded in the same expert.

This 8D experiment paradigm was achieved using both the virtual avatar and the exoskeleton effector for the sessions at home and inside the laboratory, respectively. 8D experiments with virtual avatar or exoskeleton control were performed independently. The models created with one effector were not used to control the other one. The dataset obtained from the virtual avatar control are referred as series D of experiments whereas dataset with exoskeleton effector control is named series E of experiments.

During a session, the patient aimed to reach the proposed targets or rotate the wrist to specific angle. 22 targets were 3D symmetrically distributed in two cubes in front of the patient (11 targets per hand) for the exoskeleton reaching tasks and were virtually reproduced for the avatar training sessions (Figure 7-5). Sessions last in average $29 \pm 8$ min and $20 \pm 6$ min using the virtual avatar or the exoskeleton effector, respectively.

For the exoskeleton, a REW-MSLM decoder was recursively trained in real-time during 6 closed-loop experiments distributed over 2 months and was not reupdated since. For the virtual avatar control, 6 closed-loop experiments were achieved in late September for incremental real-time REW-MSLM adaptation. The total calibration time of the models for virtual avatar was 3 hours and 37 minutes with a total of 189, 194, 181 and 218 trials for the left and right hand translation and left and right wrist rotation tasks, respectively. 3 hours and 33 minutes calibration time was performed to train the model dedicated to the exoskeleton control with a total of 180, 184, 188 and 226 trials for the left and right hand translation and left and right hand rotation control.

The performance of the models were evaluated during 37 avatar experiments distributed over 5 to 203 days after the last model update session and 468 to 666 days after the recording system implantation. For the exoskeleton control sessions, 15 test experiments distributed over 0 to 167 days after the last model update session and 531 to 698 days after implantation were performed. Five exoskeleton experiments conducted between the 62$^{nd}$ and 63$^{rd}$ days were excluded due to patient health issues unrelated to the study. Timeline representing the model calibration and test during the virtual avatar and exoskeleton based experiments across time is shown in Figure 7-5.

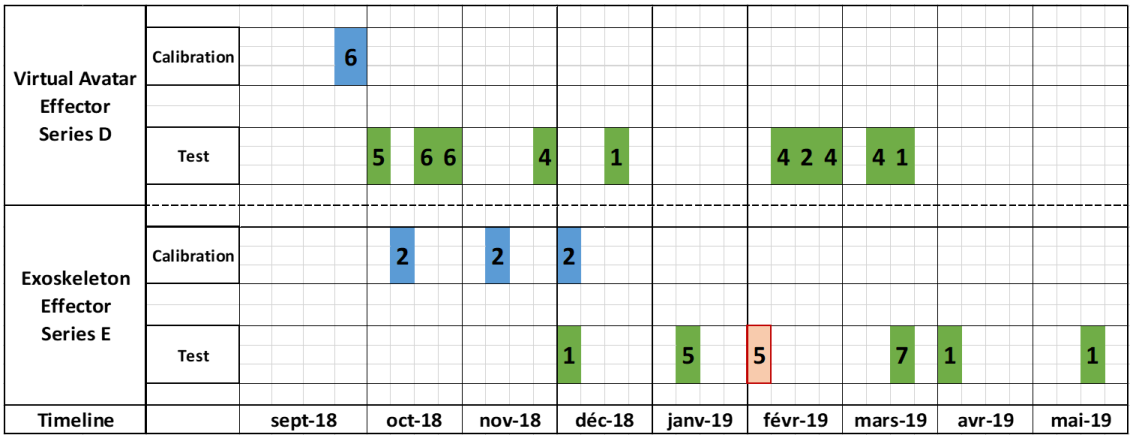



Figure 7-5: Timeline of the calibration and test sessions. Chronology of the calibration and tests sessions for the 8D online closed-loop experiments based on the virtual avatar or the exoskeleton effector. The models created for the control of the virtual avatar and the exoskeleton were calibrated independently during six real-time closed-loop experiments (in a raw for the virtual avatar control and distributed in two months for the exoskeleton) colored in blue. Test sessions are stressed through the green colored boxes whereas sessions not considered in the evaluation of the decoder performance are shaded in orange and surrounded with red color. The number inside the boxes represents the number the experiments performed weekly.

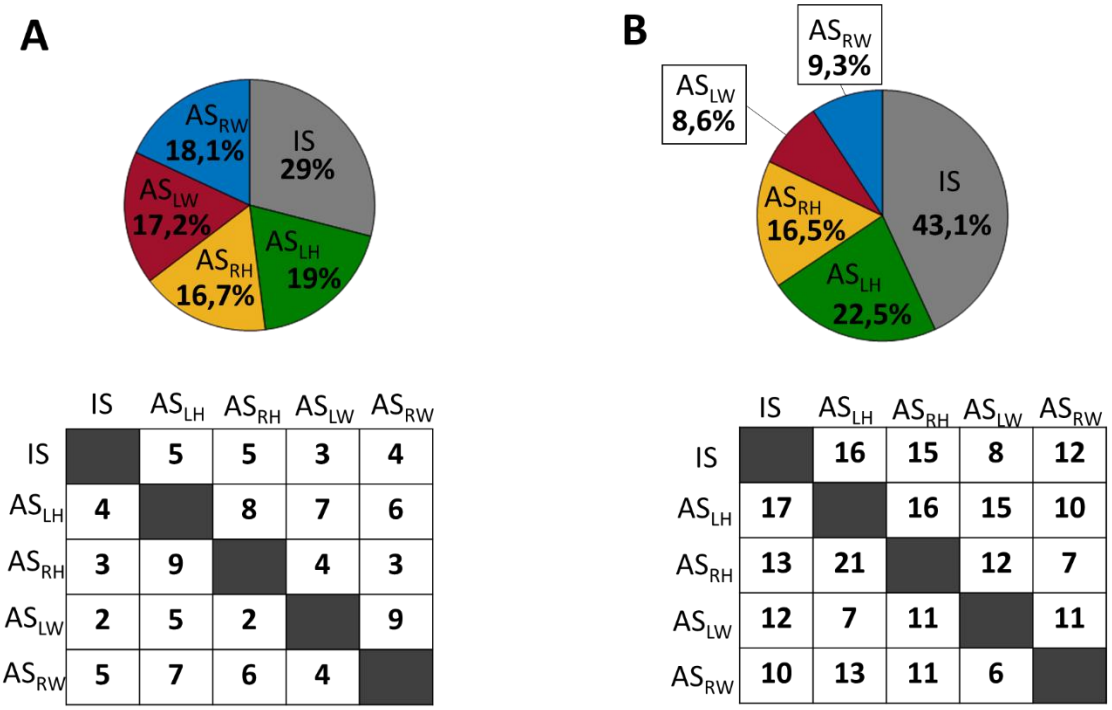
The real-time closed-loop experiments with the avatar and the exoskeleton were achieved to evaluate the performance of the REW-MSLM algorithm with an online CLDA protocol during real-time experiments. The dataset D using the virtual avatar effector is composed of 43 experiments including the calibration and test sessions. Pseudo-online studies using the same procedure than during the online closed-loop 8D virtual avatar control experiments (Pre-processing, buffer size, batch training, number of training experiments, etc.) were carried out to evaluate the performance of the PREW-NPLS and APREW-NPLS algorithms.

### 7.4.3. Online closed-loop full-state transition 4D experiments using REW-MSLM decoder

The real-time closed-loop experiments described in 7.4.1 and 7.4.2 were alternative pursuit tasks from AS states with mandatory IS transition between each AS states. Nevertheless, for daily life application, the subject can sequentially switch from one active state to another without idle state transition multiplying the possible state transitions.

New online adaptive closed-loop experiments were recorded using the virtual avatar effector between late October 2019 and mid-January 2020. The patient controlled 1D continuous movements of the left and right hands translation (vertical displacement) and 1D left and right wrists rotation tasks. This paradigm lead to a 4D continuous and $z = 5$ discrete states control problem (IS, $AS_{LH}$, $AS_{RH}$, $AS_{LW}$, $AS_{RW}$). The model was

trained and tested using the online closed-loop adaptive decoder REW-MSLM during 10 experiments (titled series F of experiments). REW-MSLM was calibrated during the first five experiments before to be tested. Model calibration lasted in total 81 min whereas all the test dataset represented 105min of experiments. In the opposite to the previously reported online experiments, during the calibration and the test phase, all the possible state transitions were experimented (including AS to AS transitions). Figure 7-6 represents the data distribution and the state transitions achieved by the experimenter during the calibration (Figure 7-6A) and test sessions (Figure 7-6B).




| | IS | $AS_{LH}$ | $AS_{RH}$ | $AS_{LW}$ | $AS_{RW}$ |
|---|---|---|---|---|---|
| IS | | 5 | 5 | 3 | 4 |
| $AS_{LH}$ | 4 | | 8 | 7 | 6 |
| $AS_{RH}$ | 3 | 9 | | 4 | 3 |
| $AS_{LW}$ | 2 | 5 | 2 | | 9 |
| $AS_{RW}$ | 5 | 7 | 6 | 4 | |

| | IS | $AS_{LH}$ | $AS_{RH}$ | $AS_{LW}$ | $AS_{RW}$ |
|---|---|---|---|---|---|
| IS | | 16 | 15 | 8 | 12 |
| $AS_{LH}$ | 17 | | 16 | 15 | 10 |
| $AS_{RH}$ | 13 | 21 | | 12 | 7 |
| $AS_{LW}$ | 12 | 7 | 11 | | 11 |
| $AS_{RW}$ | 10 | 13 | 11 | 6 | |

Figure 7-6: Description of the series F of experiments during the calibration and the test phases . The state distribution as well as the transition proposed by the experimenter between each state is represented for the calibration (A) and test experiments (B). The matrices represent the transition from the current state (column) to the following state (line) for each state.

Chapter 8

# Algorithms integration into BCI Adaptive platform

## Contents

To evaluate the performance of the proposed online closed-loop adaptive REW-MSLM decoder, it is mandatory to perform online closed-loop BCI experiments. Therefore, REW-MSLM was integrated into the software chain inside the Adaptive Brain Signal Decoder (ABSD) software for real-time decoding and online incremental CLDA. The BCI platform is presented in Figure 8-1. The ABSD software processes the ECoG neural signals in order to create a command to an effector (virtual avatar, exoskeleton, etc.). ABSD performs the classic signal processing steps (pre-processing, feature extraction, decoding, post-processing). The main loop achieved ECoG neural signal decoding at a 10 Hz frequency rate. Additionally, in parallel to the decoding main loop, ABSD was designed to run a second loop which executes the incremental batch update of the decoder at an update frequency rate fixed between 0.066Hz and 0.1Hz (every 10 to 15 seconds).

The following section introduced the signal processing steps performed in the main loop of the ABSD software before to describe the integration of REW-MSLM decoder and the interaction between the application and update loops to carry out incremental online CLDA. In the PhD research, ABSD and REW-MSLM algorithm were integrated into a DELL 7810 computer with Windows 10 operating system, Intel Xeon E5-2637 V3, 3,5GHz (2 processors) and 64 Gb of RAM.

## 8.1. Pre-processing

The pre-processing steps are directly integrated inside the implantable wireless recording system WIMAGINE. WIMAGINE is an active implantable medical device able to record epidural ECoG on 64 electrodes.

The ECoG signals were low and high pass filtered in a bandwidth from 0.5Hz to 300Hz, amplified and cleaned thanks to filters embedded into the implant hardware [Mestais et al., 2015]. The digitized epidural ECoG data were radiotransmitted to a custom designed base station connected to a computer [Mestais et al., 2015]. epidural ECoG neural signals from 32 electrodes were recorded at a sampling rate of fs= 586 Hz. Data were sent to the decoding software in batch of 100 ms. Finally, before the feature extraction step, aberrant values were removed from the dataflow at time $t$ and replaced by the recorded value at the precedent time step $t-1$. Aberrant values clustered non-numeric values or artefacts. If the neural signal variation between the time step $t-1$ and $t$ were above a fixed threshold, the neural signals at time $t$ were labeled as artefacts and replaced by the neural signal at time $t-1$.

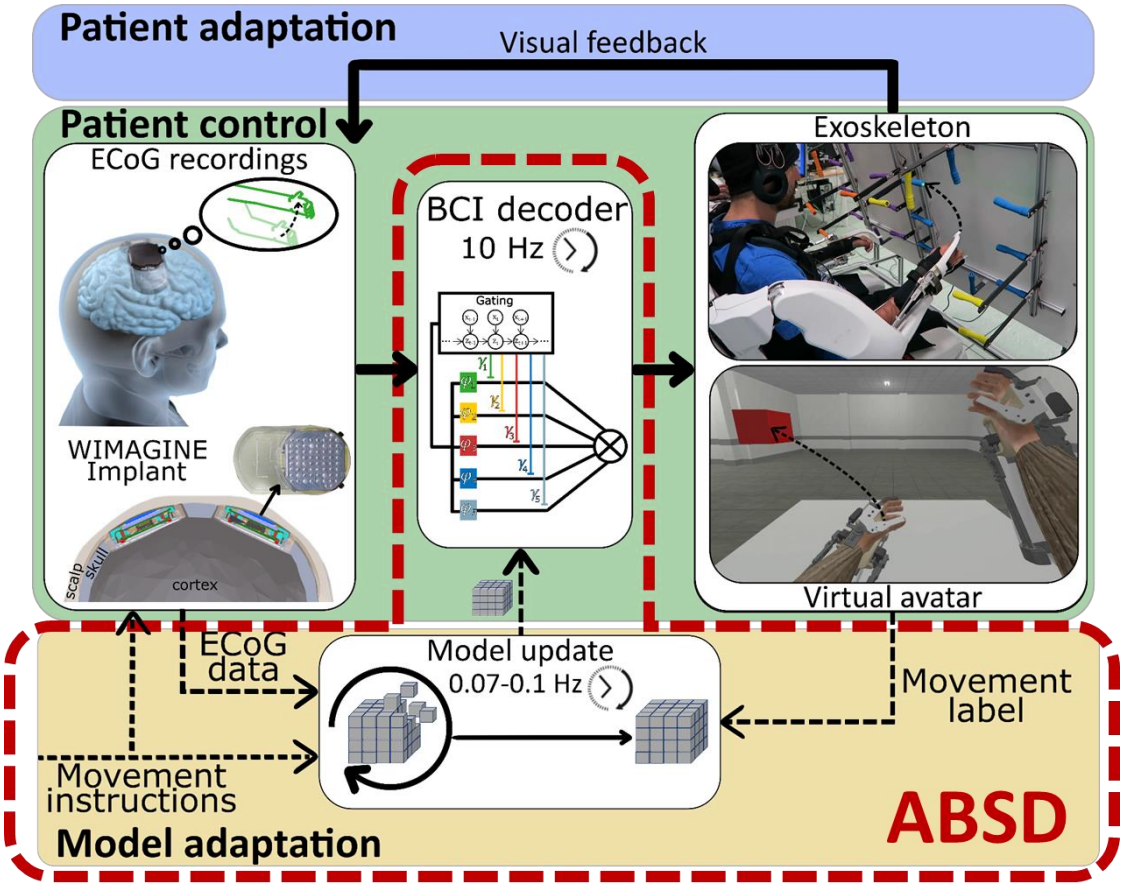



Figure 8-1: ABSD real-time adaptive BCI platform of the clinical research protocol "BCI and tetraplegia" of CLINATEC®. Two epidural ECoG recording WIMAGINE implants with a 64-electrode array [Mestais et al., 2015] are used to record brain signals. Each array provides wirelessly radiotransmitted electrical brain activity to an external processing unit. Implants were placed into the skull in contact with the dura mater above the motor cortex by a craniotomy. ECoG recordings are sent to the BCI decoders that translates neural signals into order (at 10Hz frequency) to control various effectors. Virtual avatar effector is used for patient's training at home whereas exoskeleton effector is used for training in CLINATEC® Both effectors provides visual feedback to the patient that allows him to adapt and respond in closed-loop to model predictions. Meanwhile, the model is updated (at a 0.07-1Hz frequency) based on supervised learning using ECoG data, movement instructions and movement labels. The ABSD BCI platform allows to generate "humain-in-loop" models where the neural signals related to the patient's feedback are directly integrated into the model calibration procedure.

## 8.2. Feature extraction

During the experimental sessions, at each time step $t$, the epidural ECoG neural signal epochs for all the electrodes, $\mathbf{X}_t \in \mathbb{R}^{586x64}$, were generated using a $\Delta t = 1$ s window with a 100 ms sliding step [Eliseyev et al., 2017]. ECoG epochs were mapped to the temporal frequency space using complex continuous wavelet transform (CCWT) (Morlet) with a

frequency range from 10 to 150 Hz (10 Hz step) for all the electrodes. CCWT is a feature extraction strategy that was widely used in the field of BCIs. Its efficiency has previously been demonstrated [Chao et al., 2010] [Choi et al., 2018] [Eliseyev et al., 2017] [Schaeffer and Aksenova, 2016a] [Shimoda et al., 2012]. The absolute value of CCWT was decimated along the temporal modality (by averaging the samples) to obtain a 10-point description of 1s time epoch for each frequency band and for each channel, resulting in the temporal-frequency-spatial neural feature tensor $\underline{\mathbf{X}}_t \in \mathbb{R}^{10x15x64}$. All the feature extraction steps are represented in the Figure 8-2.

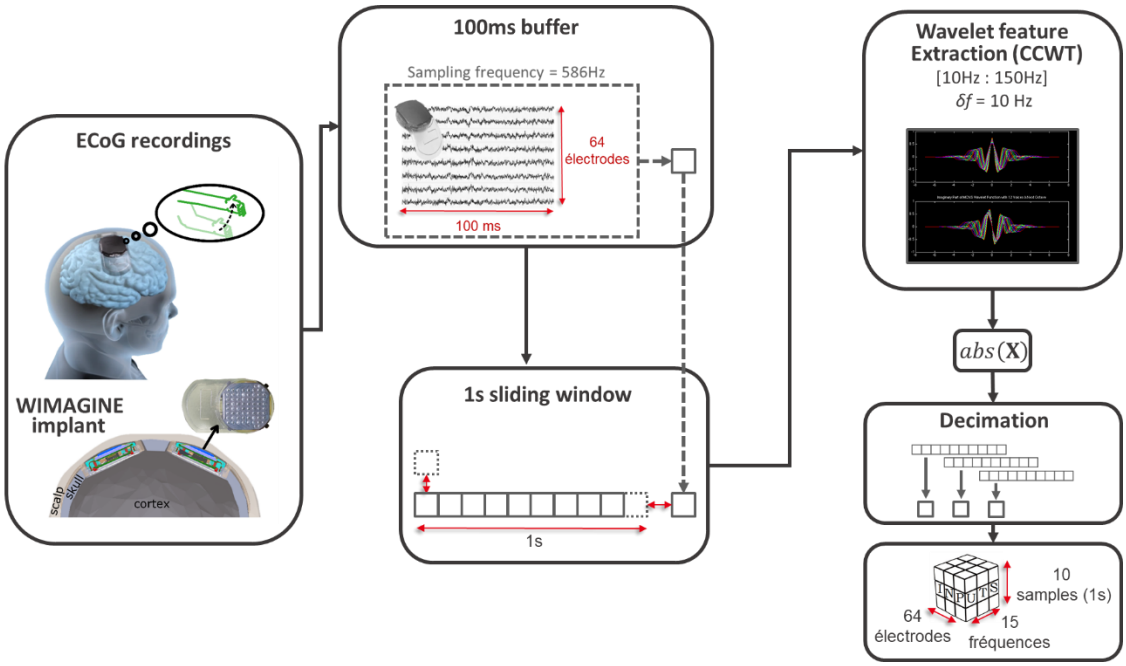



Figure 8-2: Feature extraction procedure. ECoG neural signals were recorded at a 586 Hz sampling rate. For each electrode, a 100ms batch was extracted and concatenated to previous signals to form a 1 second ECoG epoch. From the raw signals, ECoG epoch spatial-temporal-frequency characteristics of the signals were extracted through continuous complex Morlet wavelet transform between 10 Hz and 150 Hz with a 10Hz step. Absolute values and decimation of the wavelet coefficient were computed to extract the final tensor of observation variables.

The REW-MSLM gate and expert models were estimated using a supervised learning strategy. Supervised learning CLDA required the neural signals and associated labels for the estimation of both the gate and expert models. Therefore, continuous and discrete output features were estimated.

The movement output features dedicated to asynchronous multi-limb control at the time $t$ were characterized by the optimal continuous movement $\mathbf{y}_t$ and the discrete state label $z_t \in [1; K] \subset \mathbb{N}^*$, where $K$ is the number of state.

Considering the experiments analysed in the PhD research, the optimal continuous movement is defined as $\mathbf{y}_t = \left((\mathbf{y}_t^{Ltr})^{\boldsymbol{T}}, (\mathbf{y}_t^{Rtr})^{\boldsymbol{T}}\right)^{\boldsymbol{T}}$ with $\mathbf{y}_t \in \mathbb{R}^6$ for alternative 3D left and

right hand translation tasks (experiment series A, B and C) or $\mathbf{y}_t = \left((\mathbf{y}_t^{Ltr})^{\boldsymbol{T}}, y_t^{Lr}, (\mathbf{y}_t^{Rtr})^{\boldsymbol{T}}, y_t^{Rr}\right)^{\boldsymbol{T}}$ with $\mathbf{y}_t \in \mathbb{R}^8$ if 1D left and right wrist rotations were added (experiment series D and E). Here $\mathbf{y}_t^{Ltr} \in \mathbb{R}^3$ and $\mathbf{y}_t^{Rtr} \in \mathbb{R}^3$ are the 3D left and right hand translation components of $\mathbf{y}_t$ whereas $y_t^{Lr} \in \mathbb{R}$ and $y_t^{Rr} \in \mathbb{R}$ are left and right wrist rotation components of $\mathbf{y}_t$.

Similarly to the calibration hypothesis introduced by Gilja [Gilja et al., 2012], the assumption that the intended movement of the patient always followed the most efficient trajectory towards the target was made. Thus, $\mathbf{y}_t^{Ltr}$ and $\mathbf{y}_t^{Rtr}$ were defined as the 3D Cartesian vector between the current hand position at the time moment $t$ and the target position. $y_t^{Lr} \in \mathbb{R}$ and $y_t^{Rr} \in \mathbb{R}$ were left and right wrist rotation components of $\mathbf{y}_t$, defined as a 1D angle between the current angle position and the target angle position (Figure 8-3). The discrete state $z_t$ labels were determined by the task instruction with $K = 3$ in the 6D control experiments (IS, $AS_{LH}$ and $AS_{RH}$) and $K = 5$ (IS, $AS_{LH}$, $AS_{RH}$, $AS_{LW}$ and $AS_{RW}$,) in the 8D experiments. Output movement features were recorded during experiments at 10 Hz and were used during the model calibration phase.

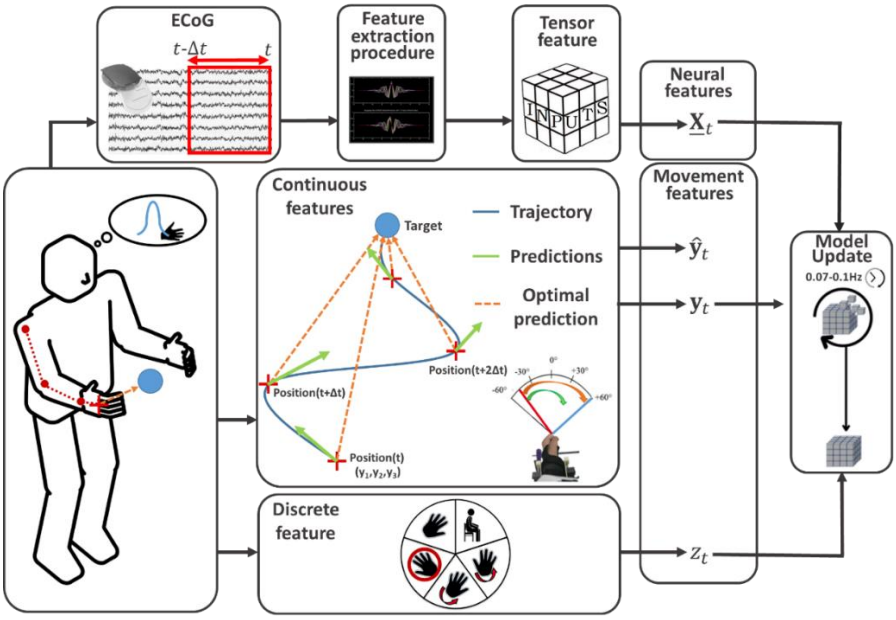



Figure 8-3: Feature extraction for supervised CLDA procedure. Neural and movement features recorded during the closed-loop experiments were used for the adaptive supervised training procedure based on the temporal-frequency-spatial neural feature tensor $\underline{\boldsymbol{X}}_t$ (computed through the steps detailed in Figure 8-2) and the optimal kinematic features $\boldsymbol{y}_t$. The optimal kinematic features $\boldsymbol{y}_t$ were defined as the 3D Cartesian vector between the current position and the target position for the 3D hand translation and as 1D angular vector between the current angle and the target angle for 1D wrist rotation. The discrete state labels was noted $z_t$. The prediction from the current model $\hat{\boldsymbol{y}}_t$, the optimal prediction $\boldsymbol{y}_t$ according to the current position and the associated state $\boldsymbol{z}_t$ were recorded as movement features. $\underline{\mathbf{X}}_t, \boldsymbol{y}_t$ and $\boldsymbol{z}_t$were stored in a buffer until the next update (every 15s) to update the REW-MSLM decoder.

## 8.3. Post-processing

Speed limit post-processing was applied to $\mathbf{y}_t^{assist}$. If the Euclidian norm of $\mathbf{y}_t^{assist}$ was above a defined speed limit threshold, the command sent to the effector was limited to:

$$\mathbf{y}_t^{eff} = min(SpeedLimit, \|\mathbf{y}_t^{as}\|) \frac{\mathbf{y}_t^{as}}{\|\mathbf{y}_t^{eff}\|}.$$

$SpeedLimit$ is an experiment fixed threshold and $\|.\|$ is the Euclidian norm.

Finally, the decoded incremental endpoint-control commands $\mathbf{y}_t^{eff}$ were converted into joints movement by exoskeleton control system activating the limbs and producing the appropriate movements through the activation of the motors of the exoskeleton or the displacement of the virtual avatar.

## 8.4. CLDA procedure integration in the BCI adaptive platform

### 8.4.1. Features labelling for CLDA.

In order to perform online decoding with online CLDA, the main application loop for the online decoding and the adaptation loop for the update of the REW-MSLM sub-models were split and implemented in two independent processes while communicating through shared memory.

The REW-MSLM decoder is structured with a discrete gating model and several continuous expert models. All the models are independently and incrementally updated in real-time on different batch of data.

During the online closed-loop experiments, in order to incrementally update the REW-MSLM decoder, the input and output features were stacked in buffers before to be sent to the ABSD second loop for gate and expert models update. To perform the incremental batch learning of the gate and expert models, the data were stored in buffers. In the considered application case, the buffer was defined as a memory storage used to temporarily store data while it is being moved from the application loop to the adaptation loop. The process is represented in Figure 8-4.

In order to update the gate model, all the neural signals samples $\underline{\mathbf{X}}_t$ as well as all the discrete state samples $z_t$ since the last update $u_{t-1}$were saved in a gate buffer to create the HMM gating parameters update block dataset $\{\underline{\mathbf{X}}_u, \mathbf{z}_u\}$. For the $k^{th}$ expert model update, only the neural signals and movements features related to the expert $k$ were stored in the k$^{th}$ expert buffer in order to create the dataset $\{\underline{\mathbf{X}}_u^k, \underline{\mathbf{Y}}_u^k\}$. $\underline{\mathbf{X}}_u^k$ and $\underline{\mathbf{Y}}_u^k$ are sub-tensors of $\underline{\mathbf{X}}_u$ and $\underline{\mathbf{Y}}_u$ formed by samples labelled as belonging to state $k$ (Figure 8-4A).

The size of the gate buffer was fixed to cluster 150 samples before to launch the model calibration procedure.

The filling of the gate buffer operated as an updating threshold to start the calibration procedure. If the $k^{th}$ expert buffer stacked more than 150 samples when the gate buffer was full, the corresponding expert model was updated in the same time than the gate model (Figure 8-4B). The gate model buffer acted as a computer clock cycle, the buffer was full every 150 samples and was updated with an expert if the expert buffer collected a sufficient amount of data. The full buffers (the buffer gate and optionally the dataset recorded for an expert if enough data were collected) were sent to the ABSD calibration loop in order to achieve the gate model update and optionally one of the expert model update. After the update of the models on the new batch of data, the calibration loop transferred the updated models to the ABSD main loop in order to apply the updated gate and optionally expert models to the incoming neural signals.

## 8.4.2. REW-MSLM initialization.

REW-MSLM has a specific architecture with independent experts which estimate continuous outputs before to be enhanced or inhibited by the gating classifier predictions. As mentioned in previous chapters, each REW-MSLM (gate and experts) can be initialized from scratch or initialized from a previously trained model. However, a REW-MSLM dedicated to a specific experimental paradigm can also be initialized from several REW-MSLMs previously trained on other different experimental paradigms. For ease of understanding, an example is shown in Figure 8-5.

Let's consider a first REW-MSLM model trained on an asynchronous multi-limb experimental paradigm referred as $Prdg_1$ composed of three tasks (IS, $AS_{LH}$, $AS_{LW}$) with one expert associated to each task and a second REW-MSLM calibrated with another experimental paradigm, titled $Prdg_2$, including four tasks (IS, $AS_{RH}$, $AS_{LW}$, $AS_{RW}$) with one expert trained for each task.

In order to create a REW-MSLM for a new paradigm $Prdg_3$ clustering previously trained tasks (IS, $AS_{LH}$, $AS_{RH}$, $AS_{LW}$, $AS_{RW}$), the expert sub-models of the REW-MSLM dedicated to $Prdg_3$ can be initialized using the expert models from the $Prdg_1$ and $Prdg_2$.

In the example presented in Figure 8-5, expert model parameters associated to the IS and $AS_{LH}$ tasks of $Prdg_3$ are initialized with the experts of REW-MSLM from $Prdg_1$ whereas other experts are initialized from $Prdg_2$ model. It should be noted, that both $Prdg_1$ and $Prdg_2$ trained an expert model for $AS_{LW}$ task. The $AS_{LW}$ expert model parameters from $Prdg_1$ or $Prdg_2$ can be used (the selection of the model can be carried out depending on various criterion such performance, stability, etc.). Additionally, new task (and expert model) never trained on any experimental paradigm can be added to the REW-MSLM architecture. This new expert is trained from scratch. Concerning the gate model, the

gate model from $Prdg_1$ and $Prdg_2$ can be exploited to initialize the gate model of the REW-MSLM $Prdg_3$. However, the gate model will require a calibration period to learn the transition from the already trained states to the state associated to the new expert.

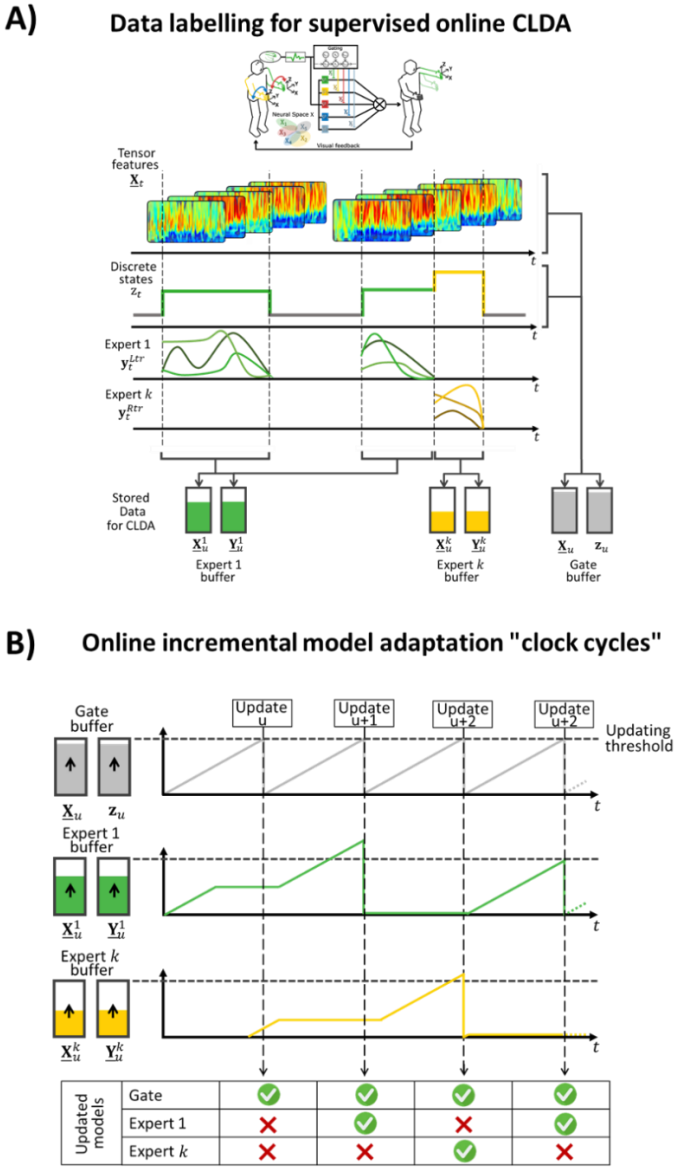



Figure 8-4: Data labelling for CLDA. (A) The data labeling procedure to stack the data for each expert and prepare the gate and expert models update. The buffer of data dedicated to the gate model stacks all the neural signals $\underline{\boldsymbol{X}}_t$ and the associated discrete label $z_t$ in the application loop before sending the batch of data to the update loop. The buffer of the expert $k$ only stacks the neural signals and the optimal kinematic features ($\boldsymbol{y}_t^{Ltr}$ or $\boldsymbol{y}_t^{Rtr}$ or $y_t^{Lr}$ or $y_t^{Rr}$ etc.) related to one specific discrete state $z_t = k$. (B) The update threshold defined the amount of data stored in the buffer of the gate model before starting the update procedure. If an expert buffer stacks more data than the gate buffer when the gate buffer is full, the gate and the expert buffers are sent to the calibration loop to update the gate and expert models.

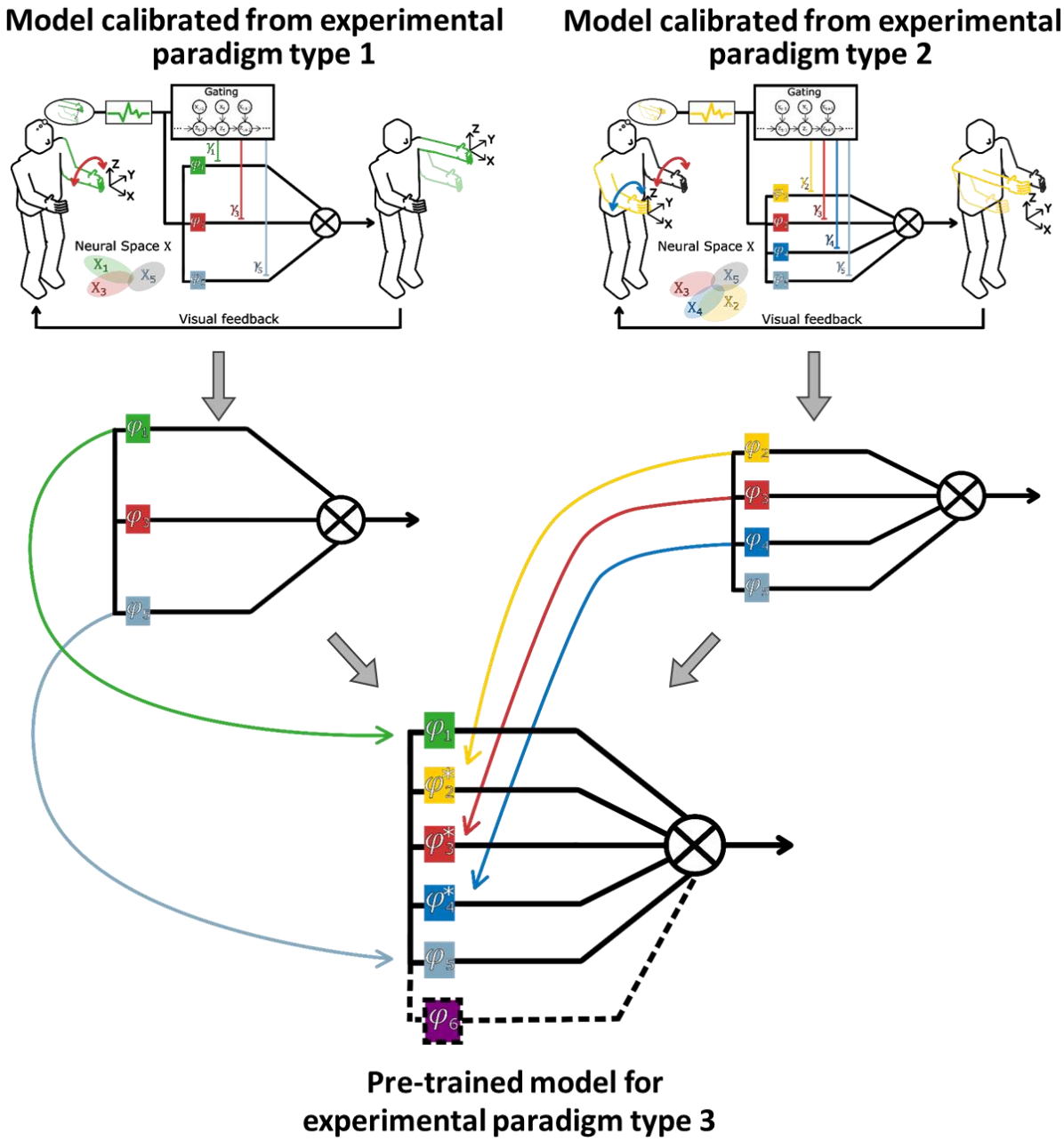



Figure 8-5: REW-MSLM architecture exploitation for model initialization. REW-MSLM has a mixture of expert architecture with experts trained independently. It is possible to initialize the expert models of a REW-MSLM with the the expert models from two previously trained models. For example, a REW-MSLM trained on $Prdg_1$ with three tasks (IS, $AS_{LH}$, $AS_{LW}$) can be mixed with a second REW-MSLM trained on $Prdg_2$ with four tasks (IS, $AS_{RH}$, $AS_{LW}$, $AS_{RW}$) to initialize the experts of a new model for a $Prdg_3$ with five tasks (IS, $AS_{LH}$, $AS_{RH}$, $AS_{LW}$, $AS_{RW}$). Additionally, new task with new expert model not calibrated in $Prdg_1$ or $Prdg_2$ can be added and trained from scratch (represented in purple in the figure).

Chapter 9

# Performance evaluation and analysis

## Contents

Before to integrate REW-MSLM into the clinical trial for online closed-loop experiments, it is mandatory to stress the interest of such algorithms for asynchronous multi-limb decoding. Several series of online closed-loop sessions are time and labour consuming and may produce stress and mental load to the patient. Additionally, the comparison of several online algorithms during closed-loop experiments is a complicated task. During closed-loop online experiments, the predicted trajectories are related to the current decoding model and patient's feedback. Consequently, during online closed-loop experiments, it is not possible to achieve algorithms comparison which produce different predictions and feedbacks.

In order to compare the decoder performance, an offline pseudo-online comparative study was undertaken before to integrate REW-MSLM into the BCI platform. Pseudo-online experiments are offline simulations conducted using the same parameters as those used for the online experiments. Pre-processing, buffer size, batch-wise training and application of the model are performed following the same procedure as that used for online real-time experiments to reproduce the online experiment conditions. Pseudo-online comparison is not fully generalizable for the online case. Nevertheless, it allows characterising the studied algorithms before an integration into the clinical BCI decoding platform. The REW-MSLM is a hybrid decoder which mixes discrete and continuous decoding. Therefore, evaluations of the performance related to the classifier (the gate) and the regression algorithms (the experts) must be carried out.

The first section of this chapter introduces the performance evaluation procedure followed to evaluate the REW-MSLM performance during offline studies and online closed-loop experiments. Then, the performance evaluation procedures of the new decoders PREW-NPLS, APREW-NPLS and the new classifier H2M2 are described.

## 9.1. REW-MSLM performance evaluation

### 9.1.1. REW-MSLM offline comparative study and online evaluation

Before to be evaluated during real-time closed loop experiments, REW-MSLM performance evaluation was achieved during offline studies. The datasets used for the pseudo-online comparison were recorded during online closed-loop experiments using REW-NPLS algorithms previously performed in the clinical trial (dataset A, B and C). REW-MSLM and the decoders used for performance comparison were recomputed in pseudo-online manner. Three key features of the REW-MSLM were evaluated.

Firstly, REW-MSLM integrates a gate which discriminates several neural states to weight the experts' output. We first highlighted the benefit of integrating discrete multi-state classifier for an asynchronous multi-limb control paradigm to switch between active states (AS) and handle robust idle state (IS) support.

Secondly, the gate model is based on dynamic decoding using HMM. Therefore, we demonstrated that HMM dynamic gating enhanced classification performance compared to the classic sample-wise gating.

Finally, the REW-MSLM algorithm benefits from the ME structure which splits the neural space into state related subsets associated to independent expert decoders (left arm translation expert, right arm translation expert, etc.). As a result, the training data are divided into subsets associated with particular experts allowing independent expert learning. However, the continuous experts were trained on a smaller specific subset of the training dataset. This may affect regression performance. The expert-specific subset training strategy was evaluated.

To evaluate the first key feature presented above, the REW-MSLM gating (Figure 9-1C) is compared to the continuous REW-NPLS model [Eliseyev et al., 2017] thresholded in post-processing to label the continuous decoding results as discrete IS and AS states. Such a comparison stresses the importance of the ME structure which dedicates a specific model to discrete state decoding (Figure 9-1A). The REW-NPLS with discretized output evaluated thanks to the post-processing threshold is named REW-NPLS$_T$.

Next, to determine the benefits of the dynamic HMM gating, the REW-MSLM gating model was compared to its own variant without dynamic HMM (Figure 9-1B). The version of the REW-MSLM algorithm with a static gating model referred as the REW-SLM algorithm (see section 4.5.2) was evaluated.

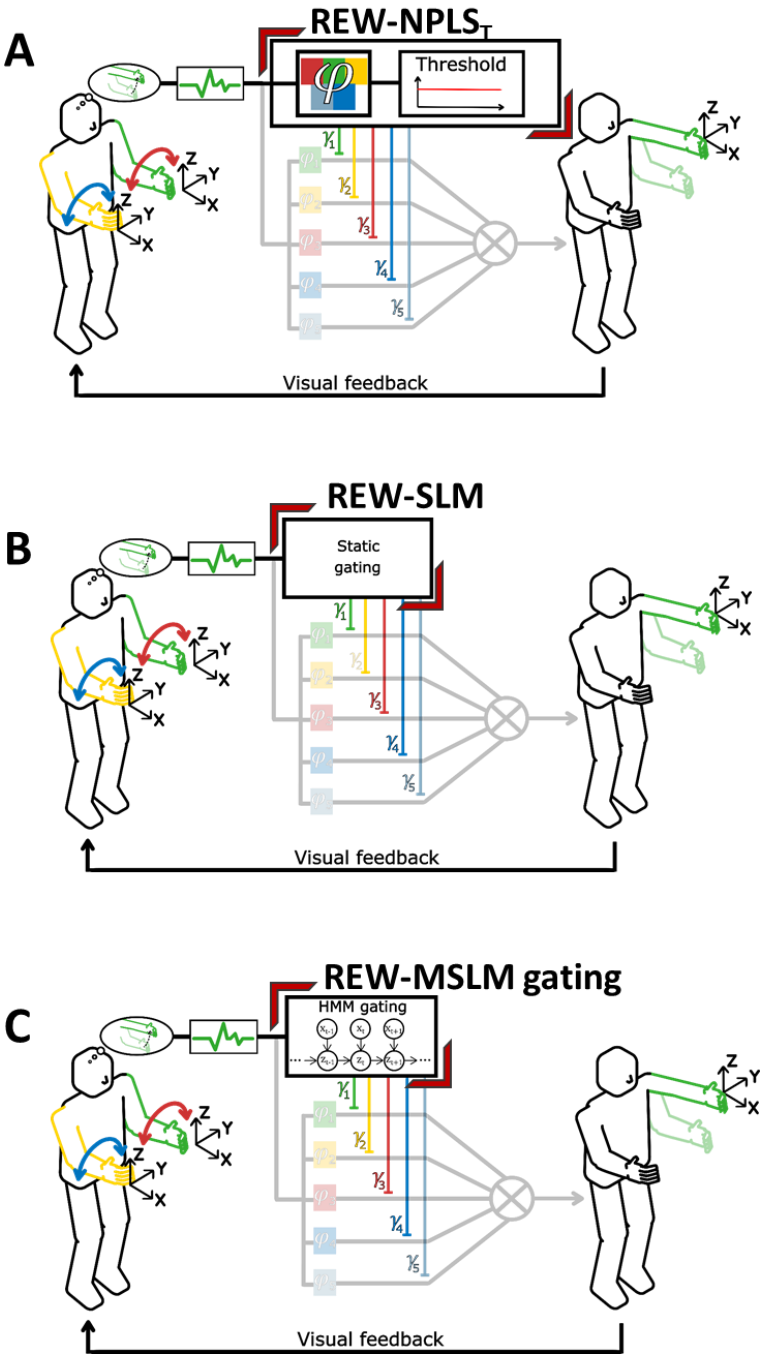



Figure 9-1: Algorithms compared to evaluate the impact of dynamic gating algorithms for online closed-loop multi-limb experiments. (A) REW-NPLS$_T$ is the model evaluated to estimate the benefits of adding a specific model dedicated to state detection. REW-NPLS model is evaluated and continuous model outputs are thresholded to define the active state. (B) REW-SLM is a mixture of expert algorithm similar to REW-MSLM with the exception that the gating model is static. REW-SLM and REW-MSLM comparison is achieved to highlight the benefits of the HMM dynamic gating. (C) Schematic of the REW-MSLM. Red brackets show the specific characteristic in which the analysis was focused on (here, the gating models).

Finally, the 3D continuous decoding performance of the REW-MSLM experts were tested to evaluate the impact of expert-specific subset training strategy on the decoder accuracy (Figure 9-2). The continuous decoding performance of the REW-MSLM experts (Figure 9-2A) were compared to a REW-NPLS model trained on the entire dataset (Figure 9-2B).

REW-MSLM performance comparison was focused on the REW-NPLS algorithm because this algorithm is a state-of-the-art online adaptive tensor-input tensor-output algorithm which have been previously employed during online and offline clinical and preclinical ECoG-based BCI experiments [Benabid et al., 2019] [Eliseyev et al., 2017]. Furthermore, as described in the previous chapters, REW-MSLM was designed based on REW-NPLS decoder. Therefore, it is relevant to evaluate the performance differences between both algorithms.

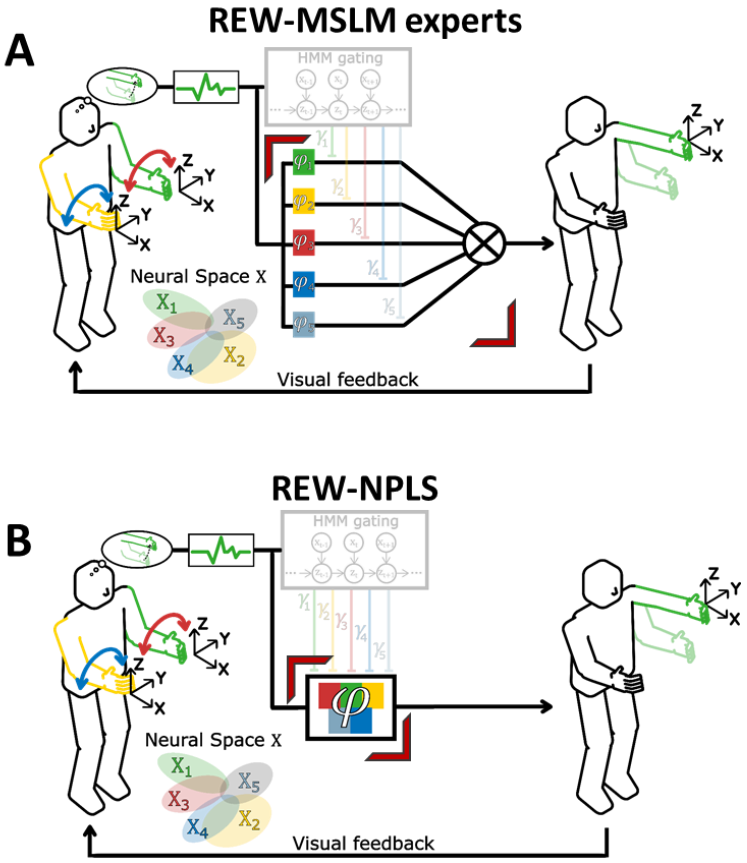



Figure 9-2 : Algorithms compared to evaluate the impact of expert-specific subset training strategy for online closed-loop multi-limb experiments. The REW-MSLM (A) experts' parameter are evaluated on specific subset of data (one for each limb to control) whereas the REW-NPLS continuous model parameters (B) are estimated on the entire dataset. Red brackets show the specific characteristic the analysis is focused on. In this case, the gating model.

To clarify the importance of the REW-MSLM ME model structure, which combines a classifier (state classification) with a continuous decoder, and the importance of dynamic vs. static gating, the REW-MSLM was compared to the state-of-the-art adaptive algorithms with three database using simulated pseudo-online experiments. The series A evaluated the performance of the algorithms with short training dataset. Then, the series B was carried out to stress the cross-session training. Finally, the C series provided information on the models robustness across time.

Although offline pseudo-online studies give an initial overview of the potential REW-MSLM decoding performance and benefits, they are not generalizable. No definitive conclusion can be extracted from these studies. Online experiment is the only solution to appraise the model robustness and to analyse the neural signal patterns modulation of the patient. Therefore, online closed-loop experiments integrating REW-MSLM as neural signal decoder were achieved.

## 9.1.2. Performance indicators

### 9.1.2.1. Offline performance indicators

#### Discrete performance indicators

Discrete performance were evaluated based on accuracy ($acc$) and F-score ($fsc$) indicators. These indicators were computed using the confusion matrix, which summarizes the number of correctly classified samples from one state (true positives, $tp$), incorrectly labelled samples in one state (false negatives, $fn$), correctly classified samples not belonging to the state (true negatives, $tn$) and incorrectly labelled samples not belonging to the state (false positives, $fp$):

$$Accuracy = \frac{1}{K}\sum_{k=1}^{K}\frac{tp_k+tn_k}{tp_k+tn_k+fp_k+fn_k}, \quad (9.1)$$

$$Fscore = \frac{1}{K}.\sum_{k=1}^{K}\frac{(\beta^2+1)\,Precision_k\,Recall_k}{\beta^2\,Precision_k+Recall_k}, \quad (9.2)$$

$$Precision_k = \frac{tp_k}{tp_k+fp_k},\ Recall_k = \frac{tp_k}{tp_k+fn_k}. \quad (9.3)$$

Here, the weighting coefficient $\beta$ was set to one, the true positives $tp_k$ were considered for samples labelled as belonging to state $k$, and the true negatives $tn_k$ included those from all the other states (one versus all analysis). $K$ was set to $K = 3$ for the pseudo-online comparative study, and $K = 5$ for the 8D online experiments.

Accuracy is a commonly reported indicator in the BCI for binary and multi-state classification [Bundy et al., 2016] [Hotson et al., 2016a] [Nguyen et al., 2019] [Schaeffer and Aksenova, 2016b] [Vidaurre et al., 2006b] and is useful for performance comparison due to its ease of computation and interpretation. Nevertheless, as accuracy presents weaknesses in

the case of highly unbalanced class, F1-score was also computed to evaluate classification performance.

The previously described state decoding indicators are sample-based performance estimators. They do not reflect the dynamic behaviour of the misclassified samples. Therefore, supplementary indicators were introduced to quantify the performance of the multi-state classification (Figure 9-3). First, the latency between the instruction and the estimated state transition was computed to evaluate the combined response time of the patient and the model. The estimated state transition was considered valid only when the decoded state was stable for 1s (10 samples). The transition had to be achieved in the 5s following the instruction state transition for it to not be counted as an incorrectly labelled state. Samples belonging to the transition/latency period were not considered in the other discrete performance indicators. Finally, the block of errors defined as consecutive misclassified samples were counted (Figure 9-3B) to evaluate the block error rate determined as the number of error blocks divided by the length of the experiments. The block error rate was then converted in averaged error blocks per minute. The averaged duration of the block of errors (Figure 9-3C) was also evaluated. In this PhD study, it was considered that several consecutive misclassification of the same class was potentially less disturbing than badly labelled samples switching at a high frequency which may lead to jerky effector command.

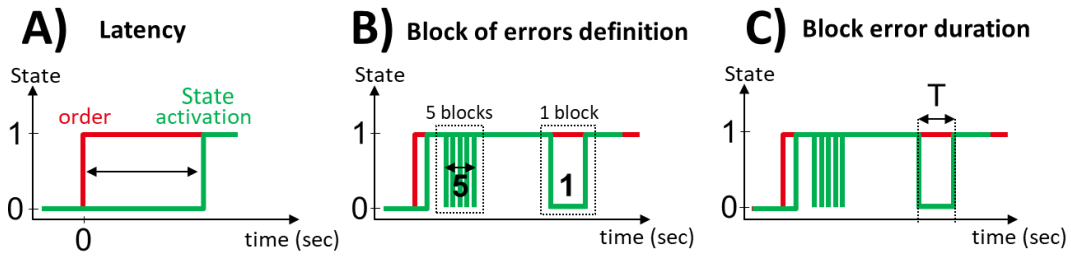



Figure 9-3 Three dynamic performance indicators were evaluated. A) The latency was defined as the delay between the task instruction provided by the experimenter and the correct state activation. This latency encompassed the reaction time of the patient, the system latency as well as the decoding latency. B) Block of errors were defined as the consecutive misclassified samples. From the number of block of error per experiments, a number of block of error per minute was evaluated. C) Block error duration computed the duration T of the block of errors.

Significance of the differences between the three decoders were computed for datasets A and C. Significance analysis on the B series was excluded because of the low number of sessions (n=4). The Mann-Whitney U test with Bonferroni corrections ($\alpha_{multi-class} = 0.0167$) was computed in the multi-class comparisons. Otherwise, $\alpha = 0.05$.

### Continuous performance indicators

As mentioned above, the predicted trajectories performed during the online closed-loop experiments are related to the decoding model currently used during the experiments and the patient's feedback. Therefore, trajectory decoding performance indicators cannot be used to evaluate the performance of different algorithms in pseudo-online experiments. A sample-based indicator is introduced to compare the continuous predictions of several algorithms (Figure 9-4A). The dot product indicator $DotP$, known in other field as the cosine similarity, is based on the comparison between the predicted directions $\hat{\mathbf{y}}_t$ and the optimal prediction defined as the 3D Cartesian vector between the current position and the target $\mathbf{y}_t$ for 3D translation tasks using the scalar product. After normalization:

$$DotP = \frac{1}{T}\sum_{t=1}^{T}\frac{\mathbf{y}_t \cdot \hat{\mathbf{y}}_t}{\|\mathbf{y}_t\|\|\hat{\mathbf{y}}_t\|},$$

where " $\cdot$ "defined the dot product, $DotP \in [-1,1]$, $T$ is the number of samples recorded for a specific limb (right or left hand). The average dot product over time provided an indicator of the algorithm global static prediction. To our knowledge, this indicator was only referenced in three articles in which EEG neural signals were analysed [Olcay and Karaçalı, 2019] [Rashid et al., 2018] [Xu et al., 2019]. This indicator was often used in the information retrieval, text mining and data mining fields [Rani and S, 2017] [Schenker et al., 2003] [Umakanth and Santhi, 2020].

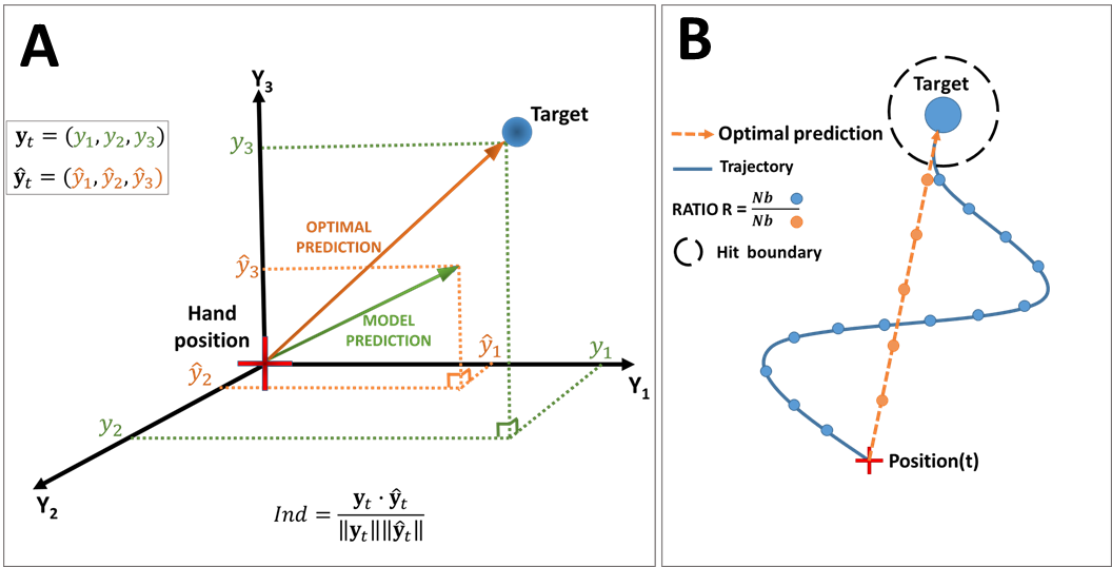



Figure 9-4 : A: online experiment performance indicators. A) Definiton of the scalar product indicator used to quantify continuous decoding performances in the pseudo-online studies. B) A target is considered as reached if the cursor is inside a sphere with the target coordinates as center and 2.5cm as radius. The R-ratio is the ratio between the distance travelled by the cursor during the task and the minimal distance travelled to reach the target.

#### 9.1.2.2. Online performance indicators

##### Discrete performance indicators

For the online discrete performance, the same static indicators than the one computed for the pseudo-online studies were computed. The accuracy ($acc$) and F-score ($fsc$) indicators are defined by the equations (9.1 and (9.2 respectively.

##### Continuous performance indicators

For the evaluation of the online closed-loop experiment performance, the success rate (SR) [Benabid et al., 2019] [Hochberg et al., 2012] [Wodlinger et al., 2015] set as the percentage of targets hit, and the R-ratio [Benabid et al., 2019], defined as the ratio between the distance travelled by the effector to reach a target and the distance from the initial position of the effector to the target location were computed (Figure 9-4B). R-ratio [Benabid et al., 2019] was also named distance ratio in [Degenhart et al., 2018] and was equivalent to the inverse of the individual path efficiency defined in [Collinger et al., 2013] [Wodlinger et al., 2015] for each task. The SR and R-ratio performance indicators were defined in the same way for the evaluation of the wrist rotation performance. The target was considered to have been hit when the 1D angular vector between the wrist position and the target was null.

##### Stability indicators

Additionally, the evolution of the performance indicators across experiments (across time) was evaluated. The linear fit with a 95% confidence interval was estimated for each indicator testing the zero slope hypothesis and evaluating the performance stability of the REW-MSLM decoder across time.

### 9.1.3. Chance level comparative study

An additional study was performed to evaluate the discrete and continuous performance indicators chance level during online closed-loop 8D experiments. Discrete states are not uniformly distributed, with a higher prior probability for idle and hand translation tasks than wrist rotations tasks. During exoskeleton-based experiments, idle, left and right hand, left and right wrist states represented 26%, 36%, 27%, 6%, 5% of the discrete state distribution, respectively. For the SR and R-ratio, $n = 100$ random hit simulations were repeated. Simulation of random movement reaching tasks were performed with the same target locations as those used during the exoskeleton-based experiments. A 3D randomly moving cursor must reach a randomly selected target within a fixed duration (defined as 99% of the cumulative distribution of the experimental time used during the exoskeleton-based experiments). At each time step, the cursor moved in a random direction with a speed fixed to the maximal speed of the exoskeleton. The target was considered to have been reached if the distance between the cursor and the target was less than 5 cm. These random sessions highlighted an averaged

SR of 7.1 ± 5.5% (R-ratio: 24 ± 14) for the left hand translation, 9.5 ± 6.6% (R-ratio: 33 ± 19) for the right hand translation, 40 ± 7.1% (R-ratio: 15 ± 4.6) for the left hand rotation and 33 ± 4.9% (R-ratio: 12 ± 2.7) for the right hand rotation tasks.

### 9.1.4. Decoding model influence analysis

In order to evaluate the parameter estimated by the REW-MSLM algorithm which had the most influence on the neural signal decoding, an analysis of the experts and gating models was carried out.

The gating emission probability model is defined by the couple $\{\underline{\mathbf{B}}, \mathbf{b}\}$ where $\underline{\mathbf{B}} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times K}$ and $\mathbf{b} \in \mathbb{R}^K$ are the tensor of the model parameters and its related bias, $I_i \in \mathbb{N}^*$ with $i \in \{1,2,3\}$ are the tensor dimensions and $K \in \mathbb{N}^*$ is the number of possible states to predict. In the specific case of the experiments proposed in the PhD manuscript, $I_1 = 10$, $I_2 = 15$ and $I_3 = 64$ are the temporal, frequency and spatial neural feature dimensions and, generally, $K = 5$ (IS, $AS_{LH}$, $AS_{LW}$, $AS_{RH}$, $AS_{RW}$). The influence of the gating model parameter weights for the activation of each state on the temporal, frequency or spatial modality was estimated following the equation:

$$Proj(I_1, i, k) = \frac{\sum_j^{I_2} \sum_l^{I_3} abs(b_{ijlk})}{\sum_i^{I_1} \sum_j^{I_2} \sum_l^{I_3} abs(b_{ijlk})}.$$

Here, $Proj(I_1, i, k)$ is considered as the influence of the i$^{th}$ ($i \in \{1, \dots, I_1\}$) parameter weight on the first dimension (the size of the dimension is $I_1$) for the $k^{th}$ expert.

Similarly, the influence of the expert model parameters on the continuous predictions of the REW-MSLM was estimated for each expert on each modality. The proposed decoding model influence analysis allowed to provide an interpretation of the REW-MLSM parameter weights. However, this analysis had some limitations. $Proj(I_1, i, k)$ is computed based on summation of the absolute value of the REW-MLSM parameter weights which does not allow to conclude on the positive or negative influence of each parameter weight on the predictions.

### 9.1.5. Neural signal modulation analysis

Different mental tasks may lead to different neural activation. The averaged time-frequency responses during the attempted discrete state activation was evaluated for each electrode using the online 8D closed-loop experiments performed with the exoskeleton. The neural activity modulation related to left and right hand translation state activations and left and right wrist rotation state activations ($AS_{LH}$, $AS_{LW}$, $AS_{RH}$, $AS_{RW}$) were compared to the averaged time-frequency response during idle state (IS) to highlight the neural signal time-frequency modulations after a state instruction and during the task activation.

The neural signals modulation was evaluated on the series of experiments E composed of 20 experiments where 8D exoskeleton control was achieved (training and testing

sessions were considered). These sessions were online closed-loop experiments performed with the REW-MSLM decoder.

In all the experiments, the neural signals related to an active states ($AS_{LH}$,$AS_{LW}$,$AS_{RH}$,$AS_{RW}$) were extracted in an 8 sec window around the AS activation corresponding to the decoder correct classification of the active state (Figure 9-5A). Additionally, the latency (delay) between the task instruction and the activation of an AS was computed.

Neural signals related to idle state (IS) period were also extracted. All the neural signals labeled as IS were stacked with the exception of the 2 seconds before or after any state transition with an AS (AS towards IS and IS towards AS).

For the neural signals windows related to IS, the 2 sec before or after any activation/transition from or to an AS were not considered (Figure 9-5B).

To obtain more relevant and cleaner neural signal modulations, the extracted AS and IS windows containing false activations ($fp$ and $fn$ classification) of the non-selected state as well as non-stable activation of the intended state were excluded. Additionally, extracted AS windows which presented latency state activation considered as outliers (latency above 10 seconds) were removed.

Both AS and IS remaining neural signal windows were mapped to the time-frequency space based on the complex continuous wavelet transform (Morlet) similarly as for feature extraction step of the ABSD BCI system previously described (details in the Chapter 8). The absolute value of the IS and AS windows was computed before to be log-transformed (Figure 9-5).

All the remaining IS neural time-frequency windows were concatenated and averaged to obtain an IS baseline. Then, the IS baseline was subtracted to every remaining AS time-frequency window. Finally, the AS vs IS time-frequency windows were averaged. This procedure was achieved independently for the 64 electrodes (Figure 9-5).

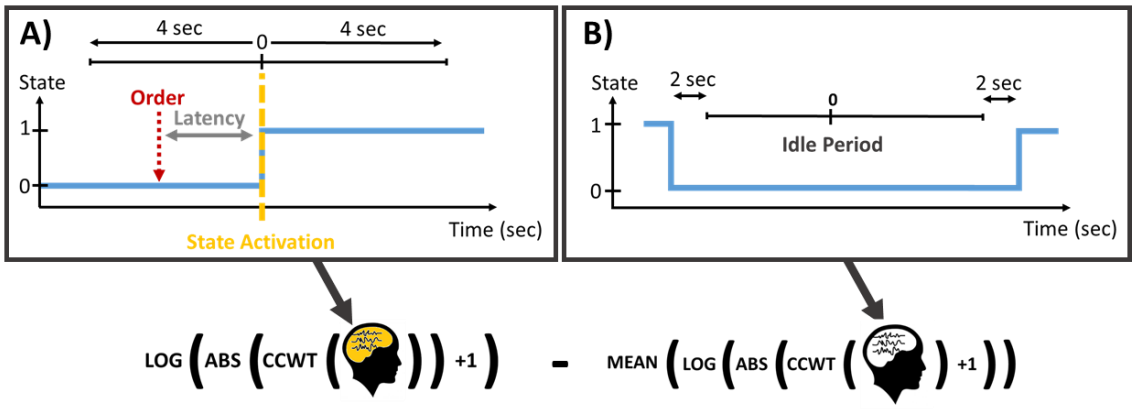



Figure 9-5: Averaged time-frequency responses during intended state activation. The averaged time-frequency responses were evaluated for each electrode. A) The active states (AS) were extracted in an 8 sec window centered on the AS activation. Additionally, the latency between the task instruction and the AS activation was computed. B) The idle state (IS) periods were extracted with the exception of the 2 seconds before or after any state transition with an AS.

## 9.2. PREW-NPLS and APREW-NPLS performance evaluation

### 9.2.1. Algorithm comparison

The PREW-NPLS and APREW-NPLS were designed to integrate the REW-MSLM algorithm as sparse expert or gate sub-models. PREW-NPLS and APREW-NPLS algorithms performance were evaluated during pseudo-online experiments based on the 8D dataset D. Both algorithms are a penalized version of the REW-NPLS algorithms which may be integrated into the REW-MSLM as regularized experts. Therefore the performance of these algorithms were compared to their non-penalized version (REW-NPLS) trained with the same expert-subset calibration strategy than REW-MSLM. PREW-NPLS and APREW-NPLS algorithms were particularly evaluated on the left and right 3D hand translation tasks of the D series of experiments. Similarly as during the online closed-loop experiments of the D series, the PREW-NPLS and APREW-NPLS models were calibrated on the first 6 sessions and were tested on the remaining 37 experiments.

$L_p$-PREW-NPLS was presented in the previous section for $p = 0, \frac{1}{2}, 1$. The three type of penalization were tested during the pseudo-online studies. PREW-NPLS penalized models required to fixed a supplementary hyperparameter named the penalization hyperparameter $\lambda$. To evaluate the influence of $\lambda$ on the model performance and on the model sparsity, 31 models were evaluated with the penalization hyperparameter $\lambda$ going from 0 to 0.6 with a 0.02 steps. In the case of the $L_0$-PREW-NPLS studies, preliminary

results highlighted that the studied $\lambda$ range was not relevant. Therefore, additional models with the penalization hyperparameter $\lambda$ going from 0 to 0.06 with a 0.002 steps were estimated (Figure 9-6A).

APREW-NPLS similarly than PREW-NPLS can be evaluated with three different types of penalization. However, the preliminary pseudo online study was only focused on the $L_1$ norm penalization. The study was limited to $L_1$ norm penalization due to the limitations of the $L_0$ and $L_{0.5}$ norm/pseudo norm penalization which were highlighted in the results chapter. APREW-NPLS selected the most relevant penalization coefficients among a list of possible $\lambda$ value. All the lambda of the set $\boldsymbol{\lambda}$ were tested and evaluated during the same pseudo-online experiments contrarily to PREW-NPLS which required one pseudo-online experiment per penalization coefficient. Six models with $\lambda$ going from 0.1 to 0.6 with a 0.1 steps were estimated and compared during the pseudo-online experiments (Figure 9-6B).

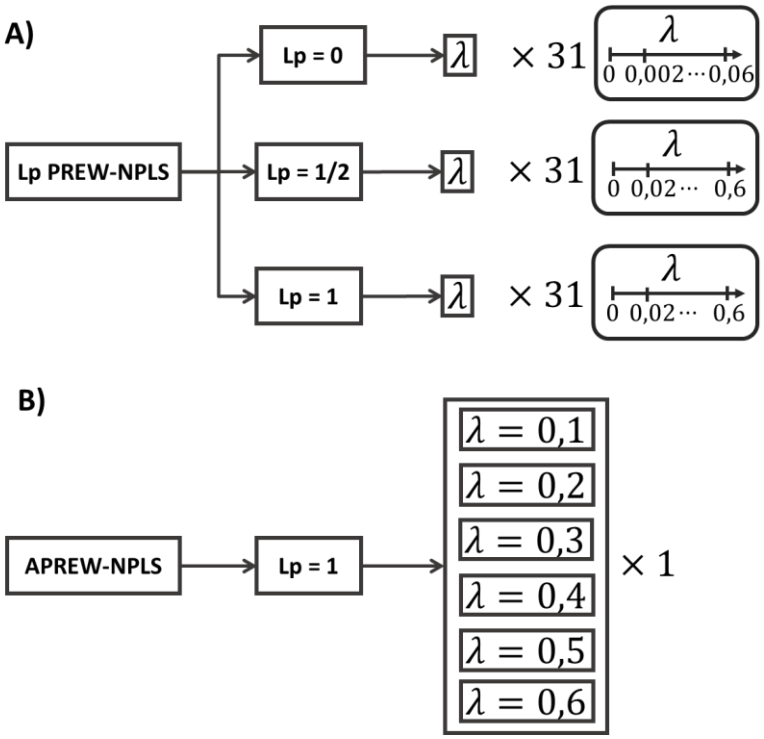



Figure 9-6: Pseudo-online PREW-NPLS and APREW-NPLS performance studies. (A) PREW-NPLS is evaluated for the three penalization type $p = 0, \frac{1}{2}, 1$. PREW-NPLS model need to fix the penalization hyperparameter $\lambda$. To evaluate the performance the three type of penalization depending on the penalization hyperparameter, 31 $\lambda$ values (from 0 to 0.6 with a 0.02 steps) were calibrated and tested for each penalization type. (B) APREW-NPLS model was only calibrated ones with the penalization type $p = 1$. As the model optimize the penalization hyperparameter, the calibration was only carried out once but for a smaller subset of possible $\lambda$ going from 0.1 to 0.6 with a 0.1 steps.

### 9.2.2. Performance indicators

Similarly to the pseudo-online evaluation of the REW-MSLM performance, the scalar product (cosine similarity) $DotP(t)$ was computed. The median, 95% confidence interval of the median, 25th and 75th percentiles of the $DotP(t)$ were estimated for each model.

The PREW-NPLS and APREW-NPLS algorithms converged into sparse solutions by fixing non-relevant (non-informative / noisy) electrodes to exactly 0. Direct decoding performance was therefore not the only relevant indicator. A sparse decoder with the same performance than a "classic" decoder might lead to faster model application and better generalization of the decoded neural signals.

Considering a penalized model $\theta_{u,\lambda_i} = \left\{\underline{\mathbf{Beta}}_u^{f,\lambda_i}, \underline{\mathbf{bias}}_u^{f,\lambda_i}\right\}_{f=1}^{F}$ with $\underline{\mathbf{Beta}}_u^{f,\lambda_i} \in \mathbb{R}^{(I_1\times\ldots\times I_M)\times(J_1\times\ldots\times J_N)}$ estimated using the PREW-NPLS or APREW-NPLS algorithms with the group-wise penalization restricted to the $m^{th}$ dimension of size $I_m$. This model was computed from the set of penalized projectors $\left\{\mathbf{w}_f^1 \in \mathbb{R}^{I_1}, \ldots, \mathbf{w}_f^M \in \mathbb{R}^{I_M}\right\}_{f=1}^{F}$ evaluated with the penalized PARAFAC decomposition.

The model sparsity indicator was defined by the number of element $\mathrm{w}_{j,f}^m$ of $\mathbf{w}_f^m \in \mathbb{R}^{I_m}$ fixed to zero. The $SparseIdx$ of the model $\theta_{u,\lambda_i}$ following the $m^{th}$ dimension of size $I_m$is defined as:

$$SparseIdx\left(\theta_{u,\lambda_i}, m\right) = \frac{\sum_{j=1}^{I_m} \delta_{\mathrm{w}_{j,f}^m,0}}{I_m}.$$

Here, $\delta$ is the Kronecker symbol.

For the PREW-NPLS algorithm, significance of the differences between the cosine similarity of REW-NPLS and PREW-NPLS algorithm was computed for the left and right hand translation studies and for each penalization hyperparameter $\lambda$. The statistical analysis was performed with the non-parametric paired Wilcoxon signed rank test with ($\alpha_{multi-class} = 0.00161$) and without ($\alpha = 0.05$) the Bonferroni correction.

## 9.3. H2M2 gating performance evaluation

The H2M2 algorithms was designed to integrate the REW-MSLM algorithm as gating model. The performance improvement related to H2M2 gating model was evaluated using pseudo-online simulations.

### 9.3.1. Algorithm comparison

H2M2 is a generalization of the HMM to a hierarchical structure and sub-HMM models to decode the discrete latent variables. To stress the benefits of H2M2, the classifier was compared to two HMMs with specific prior paradigm.

Firstly, H2M2 was compared to a HMM with a limited number of available transition. The architecture was similar to the HMM computed during the online closed-loop 8D experiments with left and right hand translation and wrist rotation states (series D and E of experiments). During the 8D experiments, the consecutive pursuit tasks were always separated by short or long period of idle state. Therefore, the created HMM had limited number of transition (Figure 9-7A). This HMM architecture, referred as $HMM_{limited}$, had a limited number of transition which induced delay to the transition between two AS. However, $HMM_{limited}$ architecture reduced the number of false activation. Additionally, H2M2 was compared to $HMM_{full}$ with all the transition available (Figure 9-7B).

The H2M2 required to fix a prior architecture of the state. Knowing the number and the possible transition between the state is useful to reduce the calibration procedure. Figure 9-7C shows the H2M2 architecture tested during the pseudo-online study and compared to $HMM_{limited}$, and $HMM_{full}$. During the pseudo-online simulations, the H2M2 was structured into two layers $D = 2$. The first layer clustered three states defined as idle state (IS), the left body side movement state ($AS_L$) and the right body side movement state ($AS_R$). IS was a production state (emit output observations) whereas both AS states were defined as autonomous sub-HMM. $AS_L$ and $AS_R$ were both composed of two states defined as the left and right hand translation and wrist rotations states respectively ($AS_{LH}$, $AS_{RH}$, $AS_{LW}$, $AS_{RW}$). Therefore the set of production state was $S_{prod} = \{\mathrm{IS}, \mathrm{AS_{LH}}, \mathrm{AS_{RH}}, \mathrm{AS_{LW}}, \mathrm{AS_{RW}}\}$ and the set of internal state was $S_{int} = \{\mathrm{AS_L}, \mathrm{AS_R}\}$.

The pseudo-online analysis was performed on the F series of experiments where the REW-MSLM with $HMM_{full}$ gating model was calibrated and applied during online closed-loop 4D continuous and 5 discrete state experiments with the virtual avatar effector. All the presented HMMs (H2M2, $HMM_{limited}$, $HMM_{full}$) were calibrated and tested offline using pseudo-online simulations. During the pseudo-online simulations, each HMM (H2M2, $HMM_{limited}$, and $HMM_{full}$) was integrated as REW-MSLM gating model. Each HMM calibration was performed on the same sessions as during the online experiments.

## 9.3.2. Performance indicators

Similarly, as for the discrete REW-MSLM gating performance evaluation, static (sample-based) and dynamic performance indicators were evaluated. Latency, frequency rate and length or error blocks were computed as well as numerous static multi-class indicators. The samples labeled as belonging to transition/latency period were not considered for the performance indicator evaluation (with the exception of the latency score). For further investigation on the state classification, other static performance indicators in addition to the Accuracy ((9.1), F-score ((9.2), Precision and Recall ((9.3), were evaluated.

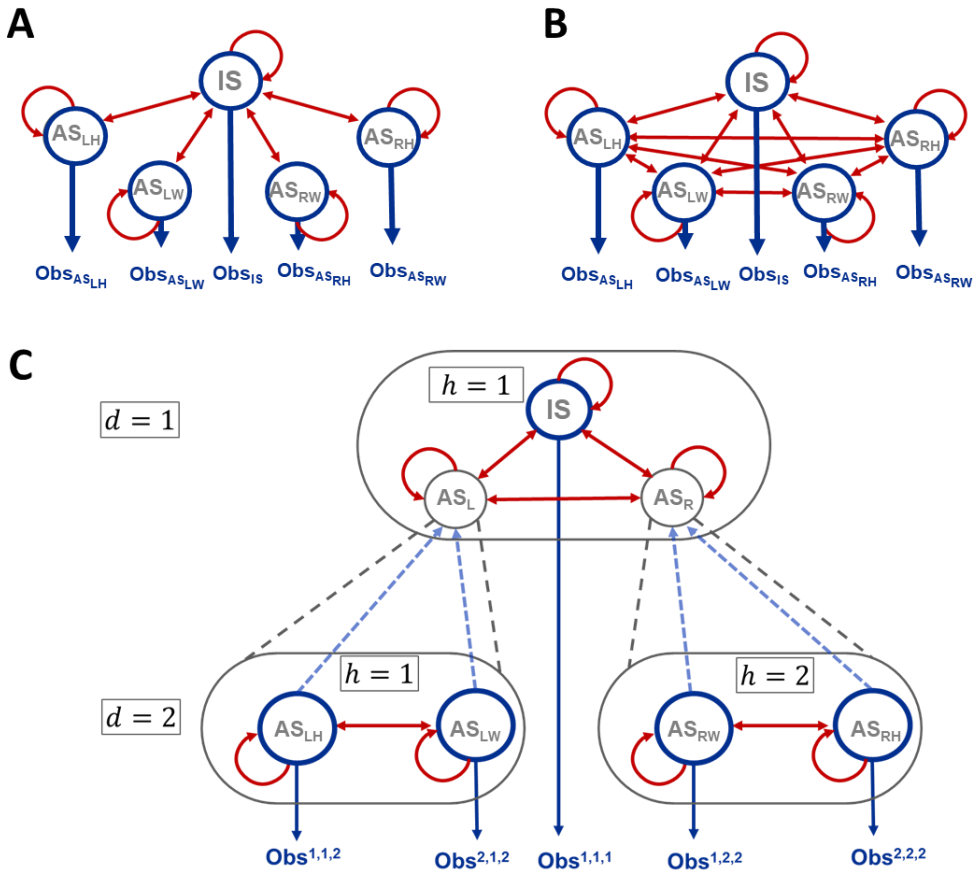



Figure 9-7: Pseudo-online evaluated HMMs with different architectures. Three HMMs were integrated into REW-MSLM as gating model and tested to evaluate the performance improvements related to each HMM specificity. (A) The HMM gating of REW-MSLM calibrated during the 8D online closed-loop experiments of the clinical trial (series D and E) was restricted in the achievable state transition. The experimental paradigm of series D and E imposed to return to IS between different AS tasks (not between trials from the same AS task). Therefore, the trained transition were limited to transitions from IS to one of the AS (and vice versa). This HMM is titled $HMM_{limited}$ in this study. (B) Conversely to the $HMM_{limited}$, a HMM was calibrated with all the transition available (including transition from one AS to another). This HMM is referred as $HMM_{full}$. (C) The proposed H2M2 gating model is based on a hierarchical architecture. The first layer is composed of three states defined as the IS, the left body side movement state ($AS_L$) and the right body side movement state ($AS_R$) with IS being a production state whereas $AS_L$ and $AS_R$ states being internal states. $AS_L$ is composed of two production states which are the left hand translation and wrist rotations states respectively ($AS_{LH}$, $AS_{LW}$). Similarly, $AS_R$ lead to two production states which are the right hand translation and wrist rotations states respectively ($AS_{RH}$, $AS_{RW}$).

Let's defined the indicators of the production state $s \in S_{prod}$ with $card(S_{prod})$ the number of production states, computed using the number of correctly classified samples from one state (true positives, $tp_s$), the incorrectly labelled samples in one state (false negatives, $fn_s$), the correctly classified samples not belonging to the state (true negatives,

$tn_s$) and the incorrectly labelled samples not belonging to the state (false positives, $fp_s$). Then,

$$Specificity = \frac{1}{card(S_{prod})} \sum_{s \in S_{prod}} \frac{tn_s}{tn_s + fp_s},$$

$$Bookmaker = \frac{1}{card(S_{prod})} \sum_{s \in S_{prod}} Recall_s + Specificity_s - 1,$$

$$Jaccard\ Coefficient = \frac{1}{card(S_{prod})} \sum_{s \in S_{prod}} \frac{tp_s}{tp_s + fn_s + fp_s},$$

$$HF\ difference = \frac{1}{card(S_{prod})} \sum_{s \in S_{prod}} Recall_s + Precision_s - 1,$$

$$Class\ Balance\ Accuracy = \frac{1}{card(S_{prod})} \sum_{s \in S_{prod}} \frac{tp_s}{tp_s + \max(fn_s, fp_s)},$$

$$Matthews\ Correlation = \frac{1}{card(S_{prod})} \sum_{s \in S_{prod}} \frac{tp_s \times tn_s - fp_s \times fn_s}{\sqrt{(tp_s + fp_s)(tp_s + fn_s)(tn_s + fp_s)(tn_s + fn_s)}},$$

$$Kappa = \frac{1}{card(S_{prod})} \sum_{s \in S_{prod}} \frac{Accuracy_s - ChanceLevel_s}{1 - ChanceLevel_s}\ with$$

$$ChanceLevel_k = \frac{(tp_s + fn_s)(tp_s + fp_s) + (tn_s + fp_s)(tn_s + fn_s)}{(tp_s + fn_s + tn_s + fp_s)^2},$$

$$Gmean = \sqrt[card(S_{prod})]{\prod_{s \in S_{prod}} Recall_s}.$$

All the described indicators were used to evaluate the pseudo-online performance of the three REW-MSLM gating models. Multi-class classification performance description is a highly complex task. Each indicator provides different information of the classification performance. For example, Kappa score measures the agreement between the accuracy and the chance level whereas Matthews correlation coefficient is a discretized version of the classic Pearson correlation which computes the balance ratios of the $tp$, $tf$, $fp$ and $fn$ in one formula. Bookmaker mixes the probability to correctly classified the selected state (recall score) and the one related to the other classes (specificity score) etc. A comparative study on each indicator was proposed in [Martel et al., 2020].

Bookmaker, Gmean, Kappa and Matthews Correlation Coefficient were reported to be good performance indicators especially for imbalanced dataset. However, Accuracy and F-score were comparison indicators commonly reported in BCI studies. In order to

compare the classification performance of the algorithms with other reported articles, accuracy and F-scores indicators were still evaluated.

## 9.4. Conclusion

In this chapter, it was proposed to compare the new mixture of expert decoder REW-MSLM with other online incremental state of the art decoder. Before to integrate the REW-MSLM into the online adaptive BCI platform of the clinical trial, pseudo-online study was carried out based on previously recorded online closed-loop BCI experiments. However, although pseudo-online studies were mandatory, they were not fully representative of the possible benefits of the new designed adaptive incremental decoders during online closed-loop experiments as the patient's feedback and adaptation were not considered in such offline simulations. Therefore, in a second step, online closed-loop experiments integrating REW-MSLM were performed for 8D exoskeleton or virtual avatar control.

In order to improve the continuous and discrete performance of the REW-MSLM gating and expert models, new continuous and discrete decoders were proposed. The recorded 8D experiments using REW-MSLM to control the virtual avatar were exploited for pseudo-online simulations of sparse PREW-NPLS and APREW-NPLS algorithms. Additionally, new online experiments were performed with REW-MSLM in order to compare the new dynamic classifier to the current gating model of the REW-MSLM.

Various performance indicators were selected to evaluate the effectiveness of the algorithms. Static and dynamic indicators were estimated for discrete performance evaluation whereas an indicator based on the dot product was computed in the pseudo-online study. During the online experiments, more straightforward indicators were presented such as the number of hit targets. PREW-NPLS and APREW-NPLS required to evaluate the "classic" decoding performance as well as the sparsity of the estimated model. Finally, in order to fully describe H2M2 classification differences with other dynamic classifiers, numerous discrete indicators were estimated.

Chapitre 10

# Results

## Contents

## 10.1. REW-MSLM

To validate the REW-MSLM algorithm benefits before an integration into the clinical trial BCI platform (ABSD), it was compared to the state-of-the-art adaptive algorithms by testing their discrete and continuous decoding performance. To this aim, series of closed loop 6D alternative two-handed reaching experiments where offline analysed in pseudo-online manner. We next present online closed-loop 8D alternative two-handed reaching and rotating clinical trial performances using the exoskeleton and virtual avatar effectors over 6 months.

### 10.1.1. Offline comparative study

The REW-MSLM mixes discrete and continuous decoding. The discrete multi-state decoding performance for an asynchronous control paradigm evaluating the accuracy of switching between all active states (AS) and, especially, the robustness of idle state (IS) support was firstly evaluated. We demonstrated that HMM dynamic gating enhanced classification performance compared to classic sample-wise gating. Supplementary latency which may be induced by HMM classifier is evaluated. To be made, REW-MSLM was firstly compared to the continuous REW-NPLS model [Eliseyev et al., 2017] thresholded in post-processing (referred to as REW-NPLS$_T$) to stress the interest of the ME structure which dedicates a specific model to discrete state decoding. Next, the REW-MSLM was compared to its own variant without HMM (called REW-SLM) to determine the benefits of dynamic HMM gating. The performances of the three algorithms were evaluated using three different experimental paradigms. The session of experiments A was designed to test the models in all-in-one experiments with small training dataset whereas the sessions of experiments B and C were achieved to evaluate the performance of the model with cross-session training and the stability of the neural signal decoding without recalibration.

#### 10.1.1.1. REW-MSLM discrete pseudo-online performance.

The REW-MSLM demonstrated strong discriminative abilities (Figure 10-1A) between all states ($acc = 93 \pm 1.8\%$, $fsc = 86 \pm 3\%$), between IS and AS ($acc = 91 \pm 3\%$,$fsc = 84 \pm 5\%$) and between AS$_{LH}$ and AS$_{RH}$ ($acc = 99 \pm 0.8\%$, $fsc = 99 \pm 0.8\%$) regardless of the experimental paradigm. The same performance indicators lead to $acc = 87 \pm 2\%$, $fsc = 76 \pm 3\%$ between all states, $acc = 86 \pm 2\%$, $fsc = 75 \pm 3\%$ between IS and AS and $acc = 93 \pm 0.3\%$, $fsc = 93 \pm 0.2\%$ between AS$_{LH}$ and AS$_{RH}$ for REW-SLM algorithm whereas REW-NPLS performs $acc = 62 \pm 2\%$, $fsc = 36 \pm 5\%$ between all states, $acc = 70 \pm 7\%$,$fsc = 49 \pm 0.6\%$ between IS and AS and $acc = 59 \pm 8\%$, $fsc = 57 \pm 9\%$ between AS$_{LH}$ and AS$_{RH}$ (Figure 10-1A). The REW-MSLM strongly discriminated each state with a particularly robust distinction between the left and right hand states. Significant improvements compared to REW-NPLS$_T$ and REW-SLM were found in the majority of the performance indicators (Figure 10-1A). No significant differences between the performance in the experimental sessions B and C were found ($p > 0.1$), indicating the model stability in session C, even though the model was not recalibrated in these experiments.

The latency of the switching state averaged over the three experimental paradigms (A, B and C) was higher for the REW-MSLM than for the REW-SLM: $lat = 2.05 \pm 0.059$ s versus $lat = 1.46 \pm 0.31$ (Figure 10-1B). Similarly, the block error duration increased with the REW-MSLM decoders. The HMM state decoder error lasted $ErrB_{time} = 4.31 \pm 0.88$ s, whereas the discrete static decoder error duration of the REW-SLM was $ErrB_{time} = 0.49 \pm 0.024$ s. However, the error block frequency decreased considerably with the REW-MSLM decoders: the error block frequency for the REW-SLM was high ($ErrB_{rate} = 20.7 \pm 1.95$ error blocks per minute), whereas the REW-MSLM error block frequency was reduced to $ErrB_{rate} = 1.6 \pm 0.26$ blocks per minute (Figure 10-1B).

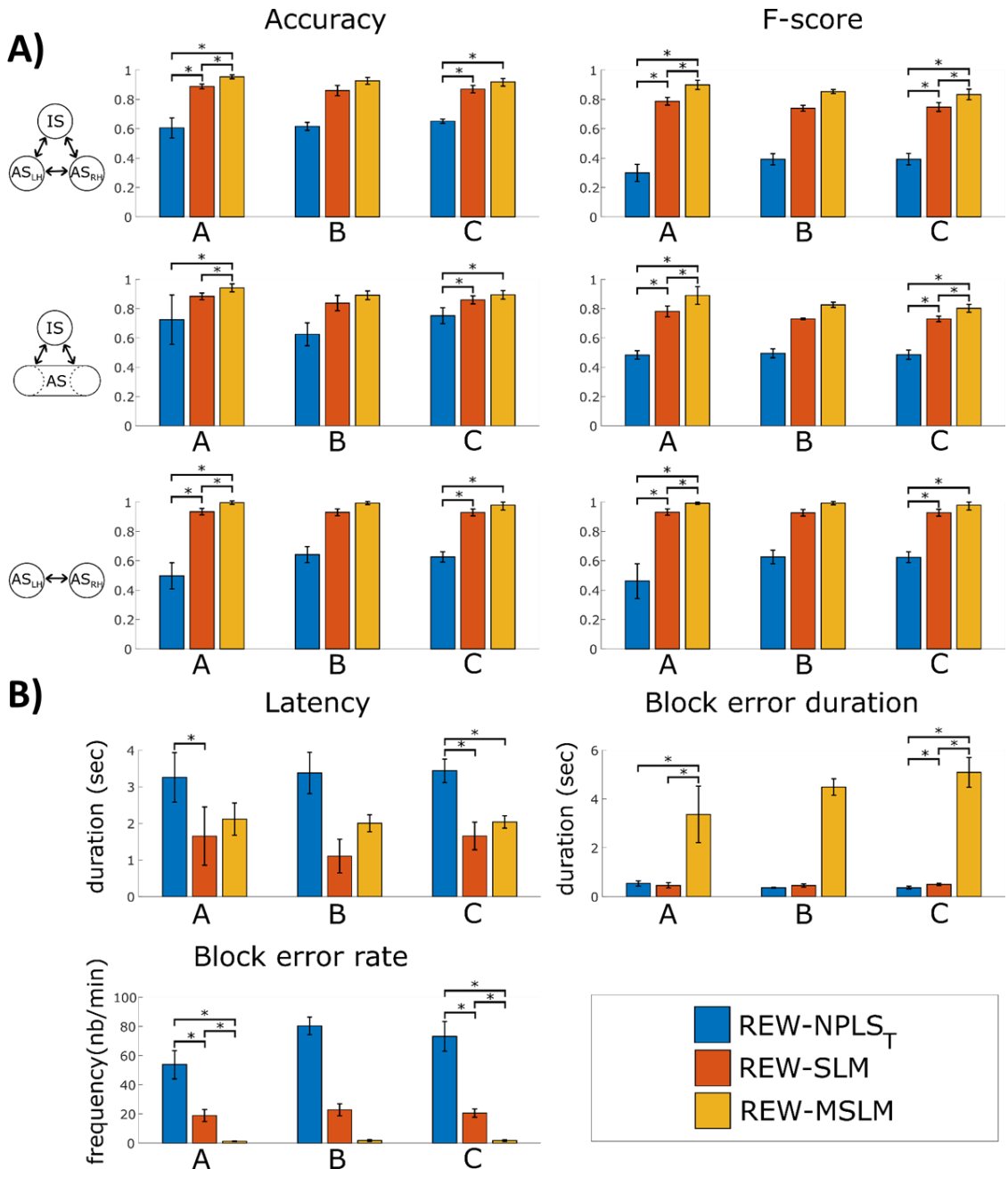



Figure 10-1 : State decoding results obtained during pseudo online experiments. A) Average accuracy and F-score over datasets A, B and C for 3 different analyses: all states (idle state IS, left hand translation active state $AS_{LH}$ and right hand translation active state $AS_{RH}$) considered independently, IS versus AS (both hand translation states merged) performance and $AS_{LH}$ versus $AS_{RH}$. B) Time dynamic performance indicators: Latency duration is evaluated as the time required to reach the desired state. Block of error durations shows the average time that last an error block of consecutive misclassified samples. The block error rate represents the occurrence of blocks of wrong detections per minute. Standard deviation is represented for each algorithm and each dataset using a vertical bar. Significance of the differences between the three decoders were computed for datasets A and C (B is excluded because of the sample size) using the Mann-Whitney U test with Bonferroni correction ($\alpha_{multi-class} = 0.0167$) in the multi-class comparisons. Otherwise, $\alpha$=0.05. Significant values are indicated by an asterisk.

Next, the continuous decoding of the REW-MSLM experts are compared to the REW-NPLS model trained on the entire dataset to evaluate expert-specific subset training strategy. The decoding performances was compared to REW-NPLS algorithm because this algorithm is a state-of-the-art online adaptive tensor-input tensor-output algorithm which has been previously employed for closed-loop ECoG-based BCI [Benabid et al., 2019] [Eliseyev et al., 2017].

#### 10.1.1.2. REW-MSLM continuous pseudo-online performance.

To evaluate the expert-specific subset training strategy piece-wise linear continuous REW-MSLM predictions were compared to those of the REW-NPLS decoder trained on the entire data set. Continuous performance median, 25th and 75th percentiles are represented in the Figure 10-2 for the three experimental paradigms. The results show similar performance in all the paradigms for both hands reaching task in average with ($DotP_{LH} = 0.095 \pm 0.05$,$DotP_{RH} = -0.03 \pm 0.16$) compared to the REW-NPLS model ($DotP_{LH} = -0.03 \pm 0.14$ and $DotP_{RH} = -0.04 \pm 0.1$) for paradigm A.

REW-MSLM left hand translation decoding of experimental sessions B and C (Figure 10-2B and C) demonstrated similar average decoding performance (B: $DotP_{LH} = 0.21 \pm 0.06$ and C: $DotP_{LH} = 0.23 \pm 0.13$) compared to the REW-NPLS model (B: $DotP_{LH} = 0.18 \pm 0.05$ and C: $DotP_{LH} = 0.18 \pm 0.11$). Similar results were observed for the right hand translation decoding with a REW-MSLM average performance of B: $DotP_{RH} = 0.15 \pm 0.07$ and C: $DotP_{RH} = 0.2 \pm 0.03$ compared to the REW-NPLS model performance of B: $DotP_{RH} = 0.14 \pm 0.09$ and C: $DotP_{RH} = 0.19 \pm 0.03$. No significant difference was evaluated between the REW-NPLS and REW-MSLM performance (Figure 10-2). However, significant enhancement of the results were computed between the decoding performance of the left hand translation models of the dataset A and dataset B ($p = 0.0159$) and the decoding performance of the right hand translation models of the dataset A and dataset C ($p = 0.0079$). The small number of experiments does not allow to confirm other significant changes. The performance improvements between dataset A and datasets B and C highlighted the benefits of cross-session training for increasing both the training data length and robustness to signal variability. No performance differences were visible between datasets B and C, stressing the model robustness.

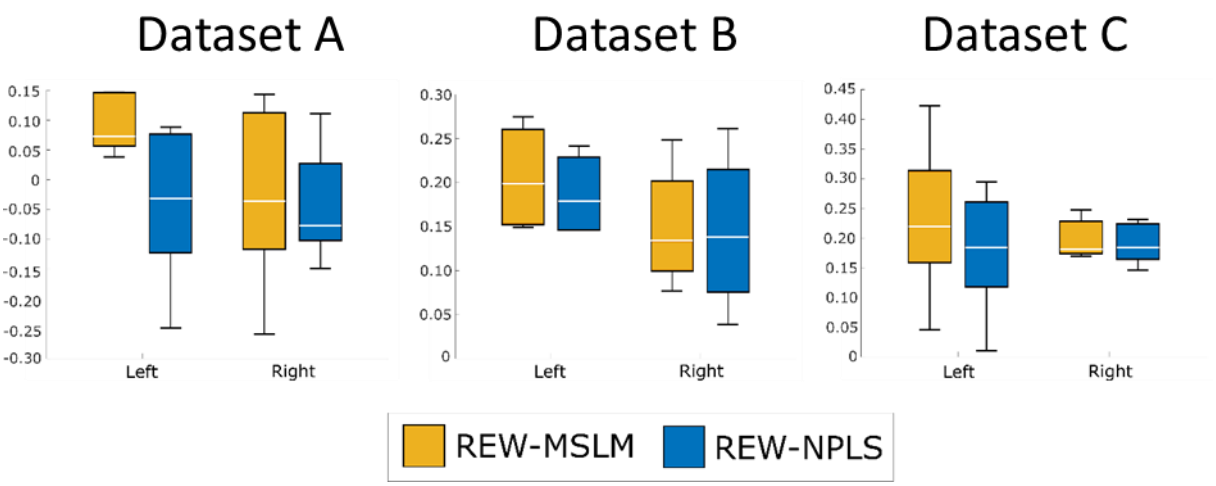



Figure 10-2: Continuous decoding performance for each hand for datasets A, B and C. Statistics of the scalar product between the predicted hand directions and the optimal prediction (defined as the target-cursor oriented distance) averaged over time and the experiments were evaluated for each dataset. The performance indicators are shown in blue for the state-of-the-art REW-NPLS model and in yellow for the new REW-MSLM. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the extreme data. Significance of the differences between the decoders and the dataset were computed using the Mann-Whitney U test ($\alpha = 0.05$).

Although pseudo-online experiments allow appraising the proposed decoder performance and stability, these results cannot be generalized to online closed-loop experiments due to the lack of patient's feedback. The purpose of CLDA procedure is to integrate the patient's feedback and related neural signals into the model calibration to perform both patient and model learning/adaption simultaneously. This behaviour cannot to be computed and evaluated in pseudo-online studies. Therefore, the performances of the REW-MSLM algorithm for 8D real-time closed-loop experiments using a virtual avatar or exoskeleton effectors over several months were appraised.

## 10.1.2. Online closed-loop 8D results

The patient achieved online asynchronous closed-loop 8D control of the effectors performing 3D alternative two-handed reaching tasks and 1D wrist rotation movements [Benabid et al., 2019] using the virtual avatar or the exoskeleton for the sessions at home and inside the laboratory, respectively. Sessions were composed of successive tasks selected by the experimenter (idle state IS or the left $AS_{LH}$, right hand $AS_{RH}$, right wrist $AS_{LW}$, and left wrist $AS_{LW}$ activation states). Each task was made of several successive trials where the patient attempted to reach target locations set sequentially with the left or right hand or to rotate the left or right wrist until a given angle. The following section presents the results obtained firstly during the 37 home sessions controlling the virtual avatar and then the 15 exoskeleton experiments. The experiments were performed over 5 to 203 days and 0 to 167 after the last model recalibration for the virtual avatar and exoskeleton experiments respectively. The

experiments were carried out 468 to 666 days and 531 to 698 days after patient's surgery for the virtual avatar and exoskeleton experiments respectively

### 10.1.2.1. Online closed-loop virtual avatar experiments

Classification decoding performance between the five states (idle, left and right hands translation and left and right wrists rotation) was demonstrated with the REW-MSLM algorithm across all the experiments (Figure 10-3A) with an average (averaged across states and experiments) F-score of $fsc = 76 \pm 9\%$ and accuracy of $acc = 93 \pm 3\%$ (Figure 10-4A). The hit performance demonstrated a right hand translation SR of $53 \pm 15\%$ (R-ratio:$5.4 \pm 3.5$) and a left hand translation SR of $55 \pm 18\%$ (R-ratio: $5.2 \pm 3.1$), whereas the average wrist rotation SR was $95 \pm 8.2\%$ (R-ratio: $3.6 \pm 3.3$) across all the experiments. Considering the prior probability of the idle, hand translations and wrist rotations states as the chance level to activate each state. The discrete state chance level of IS, $AS_{LH}$, $AS_{RH}$, $AS_{LW}$, $AS_{RW}$ were estimated at 26%, 36%, 27%, 6%, 5% respectively. The classification performances of REW-MSLM were estimated higher than the chance level for every state and every experiment across the 203 days. Similarly, the continuous decoding performance were above the averaged SR of $7.1 \pm 5.5\%$ (R-ratio: $24 \pm 14$) for the left hand translation, $9.5 \pm 6.6\%$ (R-ratio: $33 \pm 19$) for the right hand translation, $40 \pm 7.1\%$ (R-ratio: $15 \pm 4.6$) for the left hand rotation and $33 \pm 4.9\%$ (R-ratio: $12 \pm 2.7$) for the right hand rotation tasks estimated during the chance level study.

An analysis on the model decoding performance stability across time and experiments without model recalibration was carried out. The zero slope hypothesis was not rejected for 16 of the 18 indicators. It was rejected for the left wrist rotation R-ratio, which increased by 0.014 daily, and the right hand translation SR, which reduced daily by 0.07%. The zero slope analysis of the performance indicators are summarized in Table 2. These results highlighted the stability of the REW-MSLM over 6 months using a virtual avatar effector during 8D experiments, even though the model did not undergo a long training/calibration period and no model re-calibration was carried out.

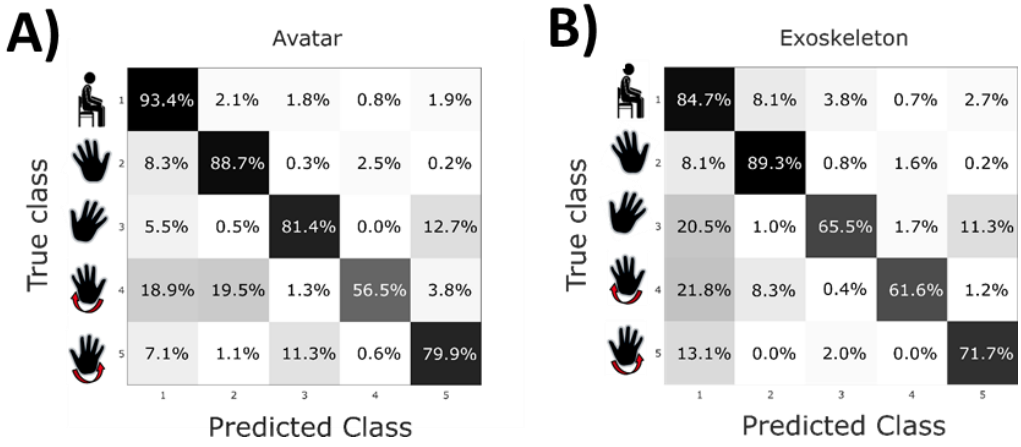



Figure 10-3: Discrete decoding performance during the online experiments for the virtual avatar or the exoskeleton effectors. Average confusion matrices for online test sessions using the virtual avatar (A) or exoskeleton (B) effectors including idle state IS or the left $AS_{LH}$, right hand $AS_{RH}$, right wrist $AS_{LW}$, and left wrist $AS_{LW}$ control states. The diagonals display the correct classification rate for each class.

Table 2: Linear approximation of the performance indicators across time estimated for the 8D virtual avatar experiments. In order to evaluate the evolution of the performance indicators across time, linear approximations of the performance across time were computed. The estimated slope and associated bias are shown in the table. The p-values indicating the significance of the results are also presented.

| | | $ax + b$ | | |
|---|---|---|---|---|
| | | $a$ | $b$ | $p - value$ |
| F-score $fsc$ | IS | 0,0285 | 81,7 | 0,353 |
| | $AS_{LH}$ | −0,0189 | 90,7 | 0,255 |
| | $AS_{RH}$ | −0,0197 | 88,8 | 0,547 |
| | $AS_{LW}$ | −0,0747 | 68,2 | 0,0819 |
| | $AS_{RW}$ | −0,0280 | 62,5 | 0,524 |
| Accuracy $acc$ | IS | −0,00683 | 92,5 | 0,542 |
| | $AS_{LH}$ | −0,00396 | 93,9 | 0,692 |
| | $AS_{RH}$ | −0,00303 | 93,2 | 0,801 |
| | $AS_{LW}$ | −0,00630 | 95,1 | 0,426 |
| | $AS_{RW}$ | −0,00327 | 94,1 | 0,760 |
| $Hit - score$ | LH translation | −0,0161 | 56,6 | 0,723 |
| | RH translation | −0,0747 | 58,7 | **0,0496** |
| | LW rotation | −0,0196 | 96,3 | 0,440 |
| | RW rotation | 0,00869 | 94,9 | 0,622 |
| $R - Ratio$ | LH translation | 0,00417 | 4,91 | 0,583 |
| | RH translation | −0,00708 | 6,03 | 0,0715 |
| | LW rotation | 0,0138 | 3,39 | **0,0141** |
| | RW rotation | 0,00308 | 2,59 | 0,386 |

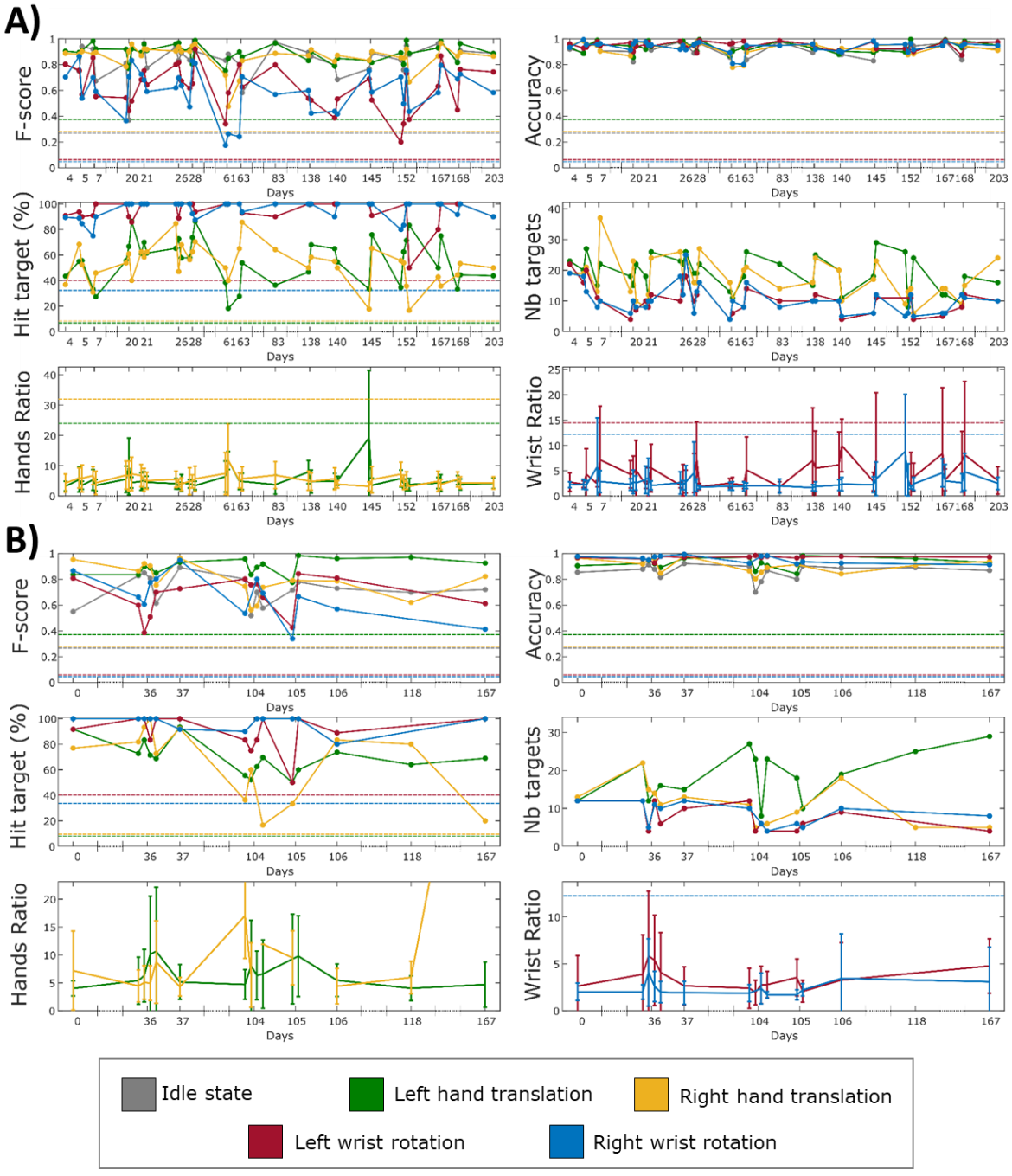



Figure 10-4: Online experiment performance across several months for the virtual avatar and exoskeleton effectors, correspondently. Online 8D experiment performance (for each state: idle, left and right hand translation and rotation) using the virtual avatar effector across 203 days after last model calibration (A) or using exoskeleton effector across 167 days after last model calibration (B). F-score and accuracy discrete performance indicators were evaluated for each state. Continuous performances were computed using the success rate (SR) (percentage of targets hit) and the R-ratio (ratio between the distance travelled by the effector to reach a target and the distance from the initial position of the effector to target location). Standard deviation is shown for each algorithm and each dataset using a vertical bar. If the chance levels are not outside the scale of the figure, they are represented for each state or task.

#### 10.1.2.2. Online closed-loop exoskeleton experiments

The Discrete decoding performances of 8D experiments yielded relevant and stable discrete decoding performance results across the 167 days (Figure 10-3B). The REW-MSLM gating yielded an average F-score of $75 \pm 12\%$ and accuracy of $92 \pm 4\%$ with high distinctiveness between the classification of the left and right sides of the body (less than 1% misclassified samples) and strong idle state decoding with an average of 85% accurately classified idle state samples.

Left hand translation demonstrated an average SR of $69 \pm 13\%$ with an R-ratio of $6.7 \pm 5.4$. Right hand translation showed similar SR but higher standard deviation than left hand translation, with an average SR of $65 \pm 29\%$ and an R-ratio of $13 \pm 4.5$ (Figure 10-4B). The decoding for both wrist rotation tasks showed an average right and left wrist rotation task completion rate of $93 \pm 12\%$ with a low R-ratio ($2.9 \pm 2.4$).

It is worth to note, that for the period 0 to 37 days after the last decoder calibration session, the online sessions using the exoskeleton yielded a decoding accuracy of 94% averaged across the five classes. Additionally, on the same period, 8D control with an average SR (for both hands) of 83% and 97% with an average R-ratio of $6.4 \pm 2.3$ and $3.3 \pm 1.7$ for the 3D hand translation and 1D wrist rotation was reported. This 0 to 37 days period corresponds or overpasses the time interval reported generally in ECoG-based BCI studies. Commonly, ECoG based clinical trials last from several days to 1 or 2 weeks (less than 28 days) of research with an implantation from 3 to 35 days [Bundy et al., 2016] [Degenhart et al., 2018] [Leuthardt et al., 2004] [Nakanishi et al., 2017, 2013] [Schalk et al., 2008, 2007] [Schalk and Leuthardt, 2011] [Volkova et al., 2019] [W. Wang et al., 2013] [Yanagisawa et al., 2012].

All the 18 performance indicators had values higher than those obtained by our chance level studies for all the experiments. The discrete state chance level of IS, $AS_{LH}$, $AS_{RH}$, $AS_{LW}$, $AS_{RW}$ were estimated at 26%, 36%, 27%, 6%, 5% whereas the continuous decoding performance of the chance level study were evaluated at an averaged SR of $7.1 \pm 5.5\%$ (R-ratio: $24 \pm 14$) for the left hand translation, $9.5 \pm 6.6\%$ (R-ratio: $33 \pm 19$) for the right hand translation, $40 \pm 7.1\%$ (R-ratio: $15 \pm 4.6$) for the left hand rotation and $33 \pm 4.9\%$ (R-ratio: $12 \pm 2.7$) for the right hand rotation tasks.

Similarly to the 8D experiments with the virtual avatar, the decoding stability was evaluated with linear regression fitting analysis (Table 3). The zero slope hypothesis was not rejected for 12 of the 18 indicators. The right side of the body seemed to have a slow performance decrease across experiments, gathering 5 of the 6 diminishing indicators. The linear fits demonstrated significant reduction in the right limb performance for the discrete ($-0.25\%$ F-score and $-0.04\%$ accuracy per day) right wrist rotation indicators and for the right hand translation F-score ($-0.17\%$), SR ($-0.42\%$) and R-ratio ($+0.24$). However, the outliers of the right hand translation R-ratio indicator of the day 167 might bias the analysis. Significant decreases were found in the left hand SR ($-0.18\%$ per day). The left hand SR seemed to decay in the first experiments before stabilizing.

Table 3: Linear approximation of the performance indicators across time estimated for the 8D exoskeleton experiments. In order to evaluate the evolution of the performance indicators across time, linear regression fitting of the performance across time was achieved. The estimated slope and associated bias are shown in the table. The p-values indicating the significance and reliability of the results are also presented.

| | | $ax + b$ | | |
|---|---|---|---|---|
| | | $a$ | $b$ | $p - value$ |
| F-score $fsc$ | IS | $-0{,}0328$ | 74,5 | 0,637 |
| | $AS_{LH}$ | 0,0510 | 85,6 | 0,158 |
| | $AS_{RH}$ | $-0{,}174$ | 92,4 | $\mathbf{0,0162}$ |
| | $AS_{LW}$ | 0,0435 | 63,8 | 0,642 |
| | $AS_{RW}$ | $-0{,}253$ | 85,8 | $\mathbf{0,0106}$ |
| Accuracy $acc$ | IS | $-0{,}0227$ | 87,6 | 0,538 |
| | $AS_{LH}$ | 0,00827 | 92,1 | 0,746 |
| | $AS_{RH}$ | $-0{,}0492$ | 94,1 | 0,127 |
| | $AS_{LW}$ | 0,0143 | 95,6 | 0,104 |
| | $AS_{RW}$ | $-0{,}0398$ | 98,5 | $\mathbf{0,0140}$ |
| $Hit - score$ | LH translation | $-0{,}185$ | 84,0 | $\mathbf{0,00808}$ |
| | RH translation | $-0{,}419$ | 97,1 | $\mathbf{0,00810}$ |
| | LW rotation | $-0{,}0702$ | 95,1 | 0,437 |
| | RW rotation | $-0{,}0216$ | 98,7 | 0,589 |
| $R - Ratio$ | LH translation | $-0{,}00529$ | 7,13 | 0,709 |
| | RH translation | 0,245 | $-6{,}03$ | $\mathbf{0,0243}$ |
| | LW rotation | $-0{,}00557$ | 3,86 | 0,470 |
| | RW rotation | 0,00142 | 2,28 | 0,766 |

Examples of hand trajectories using the exoskeleton on a session carried out 106 days after the model calibration is presented in Figure 10-5A and Figure 10-5B for the left and right hand translation, respectively. The entire session of the 106$^{th}$ days is represented in Figure 10-5C. This session is composed of successive tasks with two right hand translation tasks and three idle, left hand translation, left and right hand rotation tasks. Each task was composed of several trials. Trajectories represented in the Figure *10-5*A and Figure 10-5B are trials form the first left hand and second right hand translation tasks.

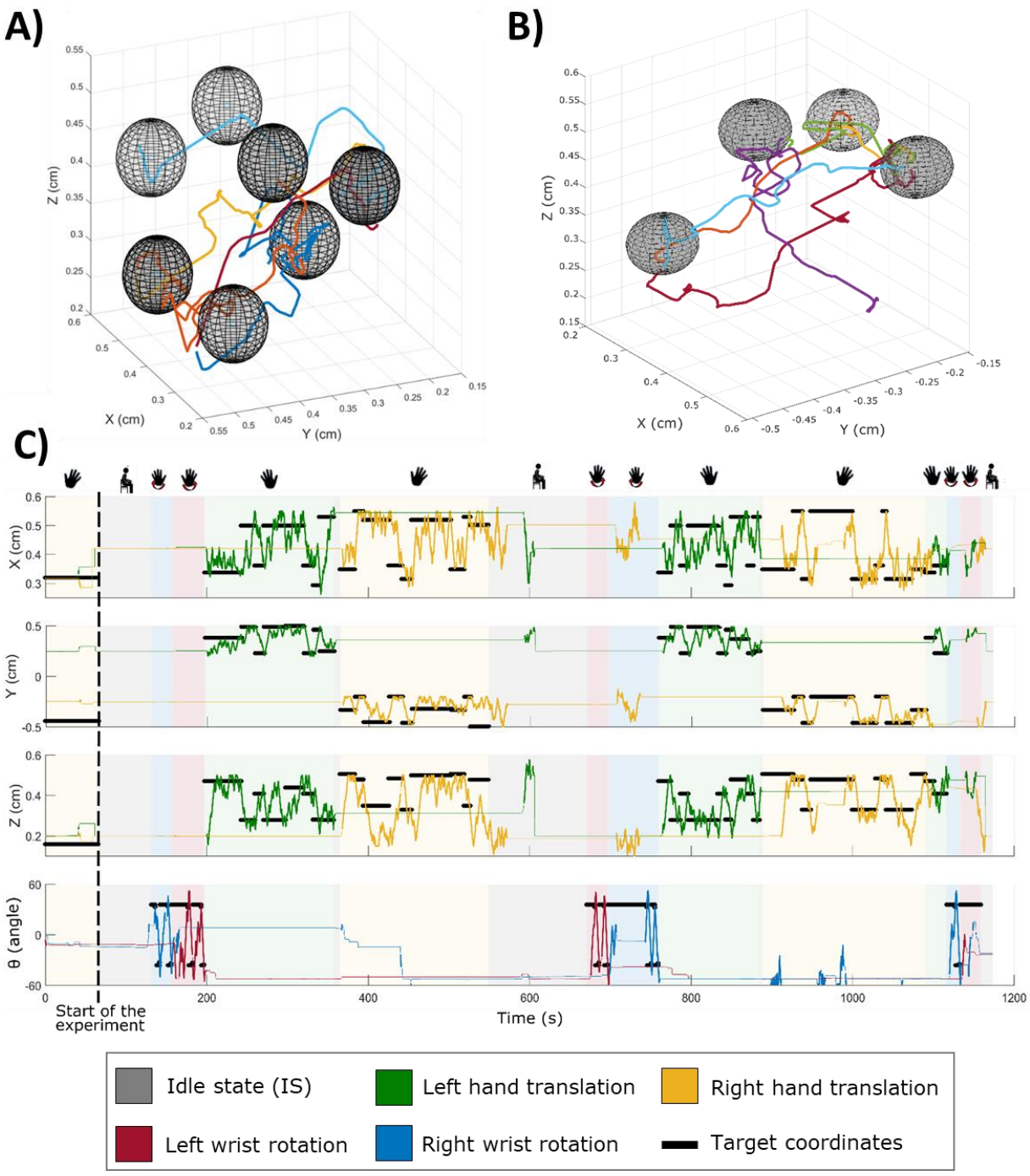



Figure 10-5: Example of session realized 106 days after the last model calibration using exoskeleton effector. A) left hand trajectory across time and trials. These trajectories are extracted from the first left hand task of the session. B) right hand trajectory across time and trials. These trajectories are extracted from the second right hand translation task of the session. C) Movement on X, Y, Z and θ (angle for wrist rotation) across the sessions performed 106 days after the last model calibration. Shaded area color correspond to the task that patient must perform. Colored Lines represent left and right hand coordinates for X, Y and Z-axis and left and right wrist angle for θ axis. Thick line width underline the state controlled by the patient.

#### 10.1.2.3. REW-MSLM parameters visualization

The gating model of REW-MSLM used for exoskeleton control is represented on the spatial, frequency and temporal modality in Figure 10-6.

On the spatial modality, the contralateral electrodes presented a higher influence on the left and right hand (translation and rotation) state estimation. Spatial modality presented higher parameter weights on the contralateral electrode array for the left and right hand (translation and rotation) state estimation. Additionally, translation and rotation from the same hand seemed to activate nearby but distinct electrodes. The parameters weights of the frequency band between 20-30Hz (β-band) and 80Hz-120Hz (γ-band) showed higher influence for the left and right hand translation state discrimination. The same frequency bands were relevant for rotation and idle state classification, nevertheless, lower frequency bands (<20 Hz) significantly contributed to the decoding, especially for idle state decoding. Finally, parameter weights in the temporal modalities between -0.1s and -0.8s before the event were dominant. Temporal parameter weights were similar for all states.

Expert models used for exoskeleton control are represented on the spatial, frequency and temporal modality in Figure 10-7. Spatial modality presented heavy parameter weights on the contralateral electrode array for left hand (translation and rotation) continuous model. All the left hand 3D translation parameters presented similar model with dominant frequency band between 80Hz-120Hz (γ-band). Left wrist rotation parameters showed dominant frequency band between 20-30Hz (β-band). Right hand continuous parameters were more complex to analyse. However, β and γ frequency bands were dominant in each model. Finally, parameter weights in the temporal modalities increased with temporal parameters closer to the movements.

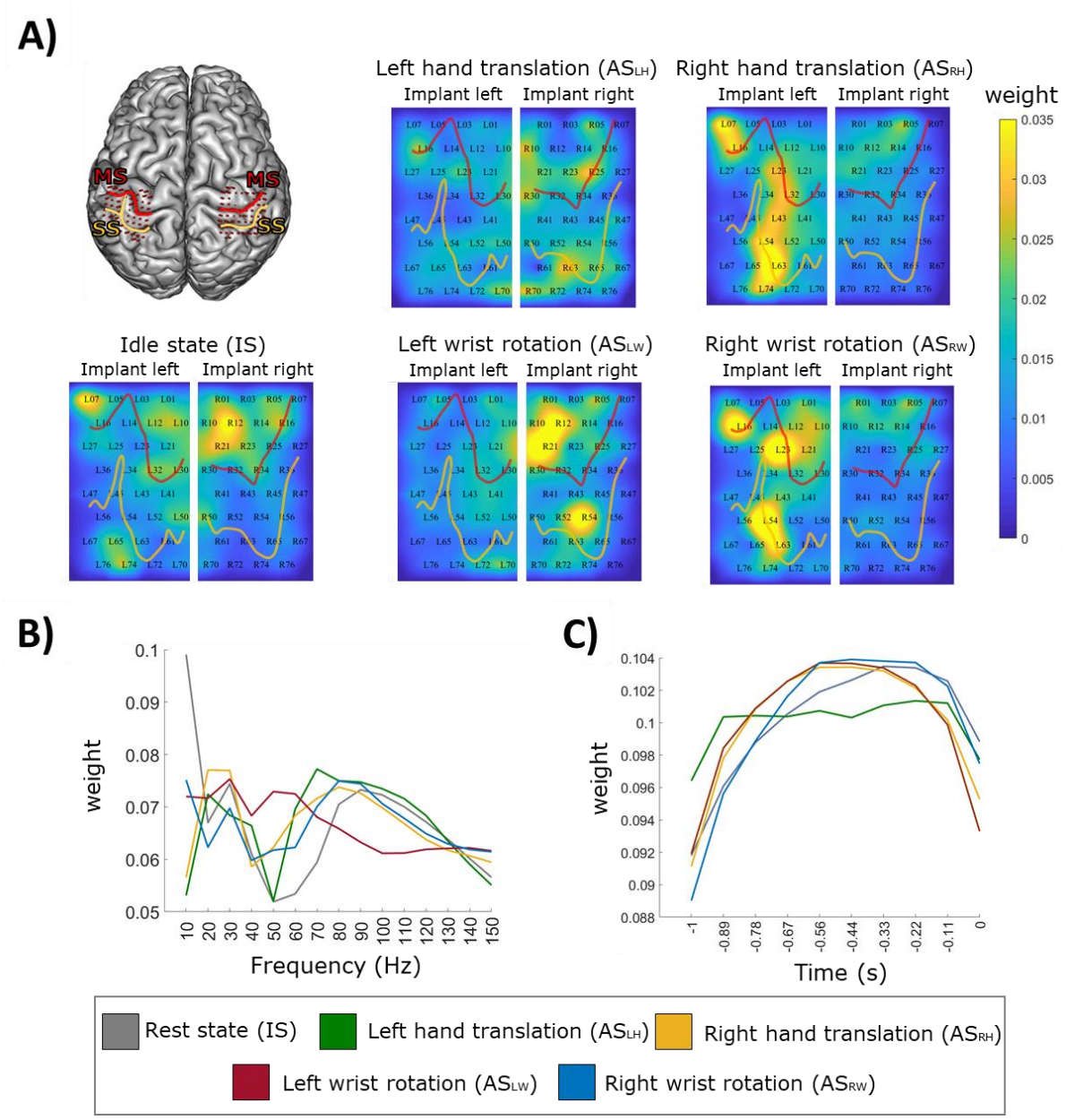



Figure 10-6: Example of a gating model. Gating parameter weights (discrete decoding) of the REW-MSLM created using an exoskeleton effector according to the (A) spatial, frequency (B) or temporal (C) modalities for each state: rest state (IS), left hand 3D translation and rotation states ($AS_{LH}$ and $AS_{LW}$ ) or right hand 3D translation and rotation states ($AS_{RH}$ and $AS_{RW}$). The sensory sulcus (SS) and motor sulcus (MS) are represented in the spatial domain in yellow and red curves respectively.

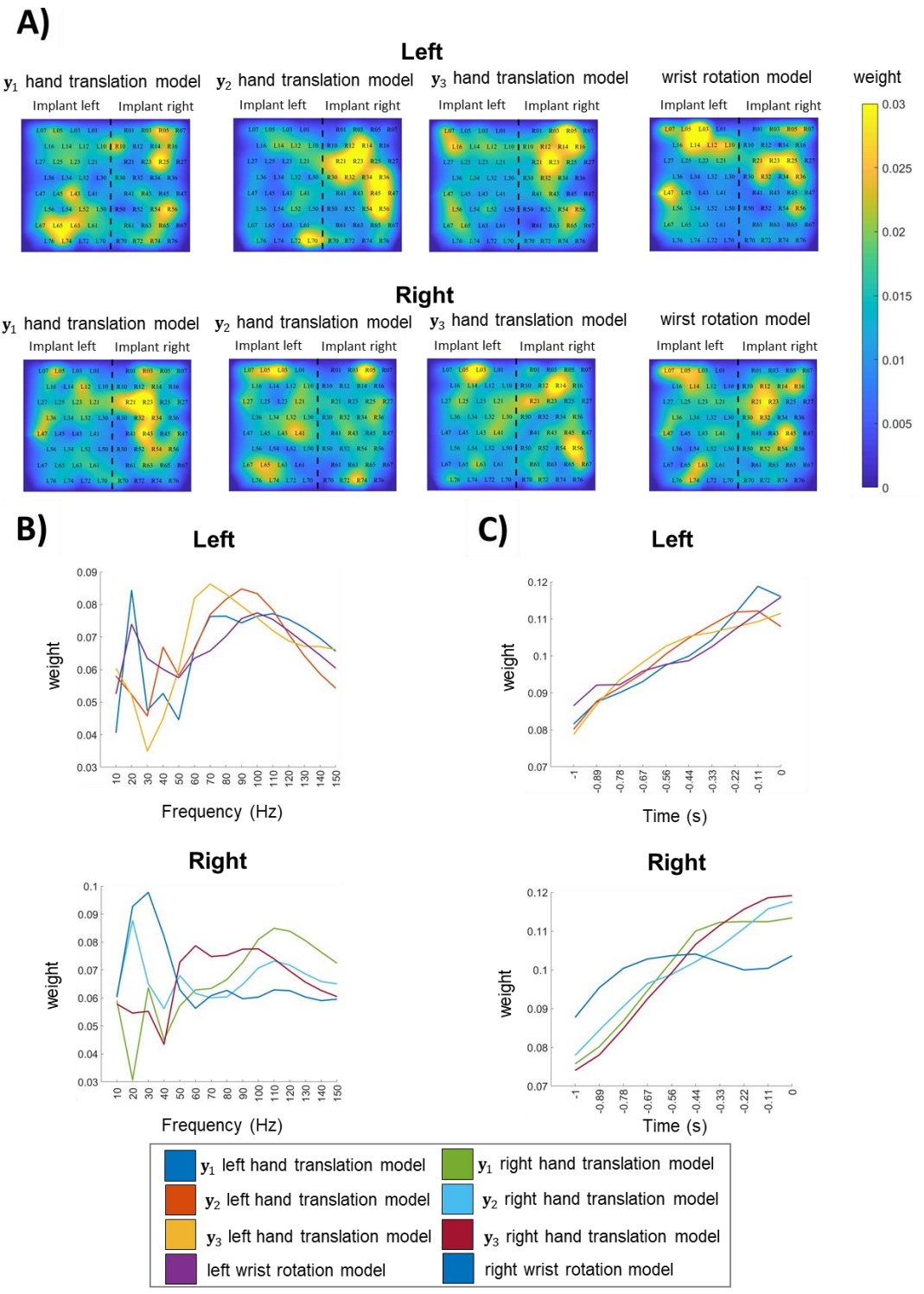



Figure 10-7: Expert model estimated during online experiments with CLDA**.** Experts parameter weights (continuous decoding) of the REW-MSLM decoder used during online clinical 8D alternative bimanual experiments achieved with the exoskeleton according to the (A) spatial, frequency (B) or temporal (C) modalities for the left or right arm continuous movements (hand translation and wrist rotation models). The sensory sulcus (SS) and motor sulcus (MS) are represented in the spatial domain in yellow and red curves respectively.

### 10.1.2.4. Neural signal modulation analysis in online 8 Dimensional control

The analysis on the neural signal modulation occurring during the activation of an active state was performed based on the online closed-loop decoding experiments using REW-MSLM with $z = 5$ states and 8D continuous decoding. During the experiments, the patient controlled the exoskeleton. The modulation between a neural signal baseline computed based on the neural signal activity during the IS periods and the neural signal activity during an active state activation are represented in Figure 10-8A, Figure 10-8B, Figure 10-9A and Figure 10-9B for $AS_{LH}$, $AS_{LW}$, $AS_{RH}$, $AS_{RW}$ respectively. A 8 seconds time-frequency map centred on the state activation time moment was computed for each of the 64 electrodes. Additionally, the latency between the order to perform the task and the state activation was evaluated. The latency indicator clustered the reaction time of the patient and the processing time of the BCI platform (transfer of data, processing, decoding, transmission of the command, etc.). The median latency of $AS_{LH}$, was $2.8 \pm 2.1$ seconds (Figure 10-8A) whereas left wrist rotation state latency was estimated at $3.2 \pm 1.4$ seconds (Figure 10-9A). The right hand translation state activation highlighted a median latency of $1.6 \pm 3.0$ seconds (Figure 10-8B) and the median latency of $AS_{RW}$ was computed at $2.6 \pm 1.0$ seconds (Figure 10-9B).

For the left and right hand translation states ($AS_{LH}$, $AS_{RH}$) presented in Figure 10-8, contralateral electrodes highlighted high modulation in the frequency bands below 40 Hz and above 70 Hz after the task order was given. Enlargements of the neural signal modulations in the electrodes L52 and R52 for the $AS_{LH}$ and $AS_{RH}$ states were represented in Figure 10-10A and Figure 10-10B respectively. A power drop in the frequency bands lower than 40 Hz is shown for almost all the electrodes of the contralateral side. For the electrodes represented in the Figure 10-10A and Figure 10-10B, the negative modulation in the low frequency bands are centered on the 20 Hz frequency band (likely the β frequency band). Additionally, a rise of power amplitude in the high frequency bands above 70 Hz was stressed around the state activation time moment of contralateral electrodes. The positive modulation of high frequency bands seems larger between the bottom of the sensory sulcus (SS) and motor sulcus (MS) (Figure 10-8).

Similarly to the hand translation states, for the left and right wrist rotation state ($AS_{LW}$, $AS_{RW}$) modulation shown in Figure 10-9, a modulation in the frequency bands below 40 Hz and above 60 Hz after than the task order was stressed. Enlargements of the neural signal modulations in the electrodes L54 and R54 for the $AS_{LW}$ and $AS_{RW}$ states were represented in Figure 10-10C and Figure 10-10D respectively. A negative modulation, centred on the 20 Hz frequency band (likely the β frequency band) for the electrodes L54 and R54 was visible on a part of the contralateral electrodes in the frequency bands below 40 Hz. This modulation was less generalized than hand translation states variations with no evident modulation of the contralateral electrodes located at the bottom of the implants near the brain longitudinal fissure. Contrarily to the hand translation states, a positive modulation of the neural signals around the 20 Hz frequency bands was noticeable on the time-frequency maps of some ipsilateral electrodes (e.g. the electrodes of Figure 10-10C and D). For the high frequency bands above 60 Hz, a positive variation was visible on the contralateral electrodes. On the electrodes

represented in the Figure 10-10D, for the $AS_{RW}$ neural signal modulation, the position variations seems stronger around 80 Hz. Additionally, Figure 10-8B and Figure 10-9B highlighted different activated electrodes for right hand translation and right wrist rotations. $AS_{RW}$ presented a higher modulation on the upper electrodes whereas for $AS_{RH}$ the main activated electrodes were located in the lower electrodes of the implants.

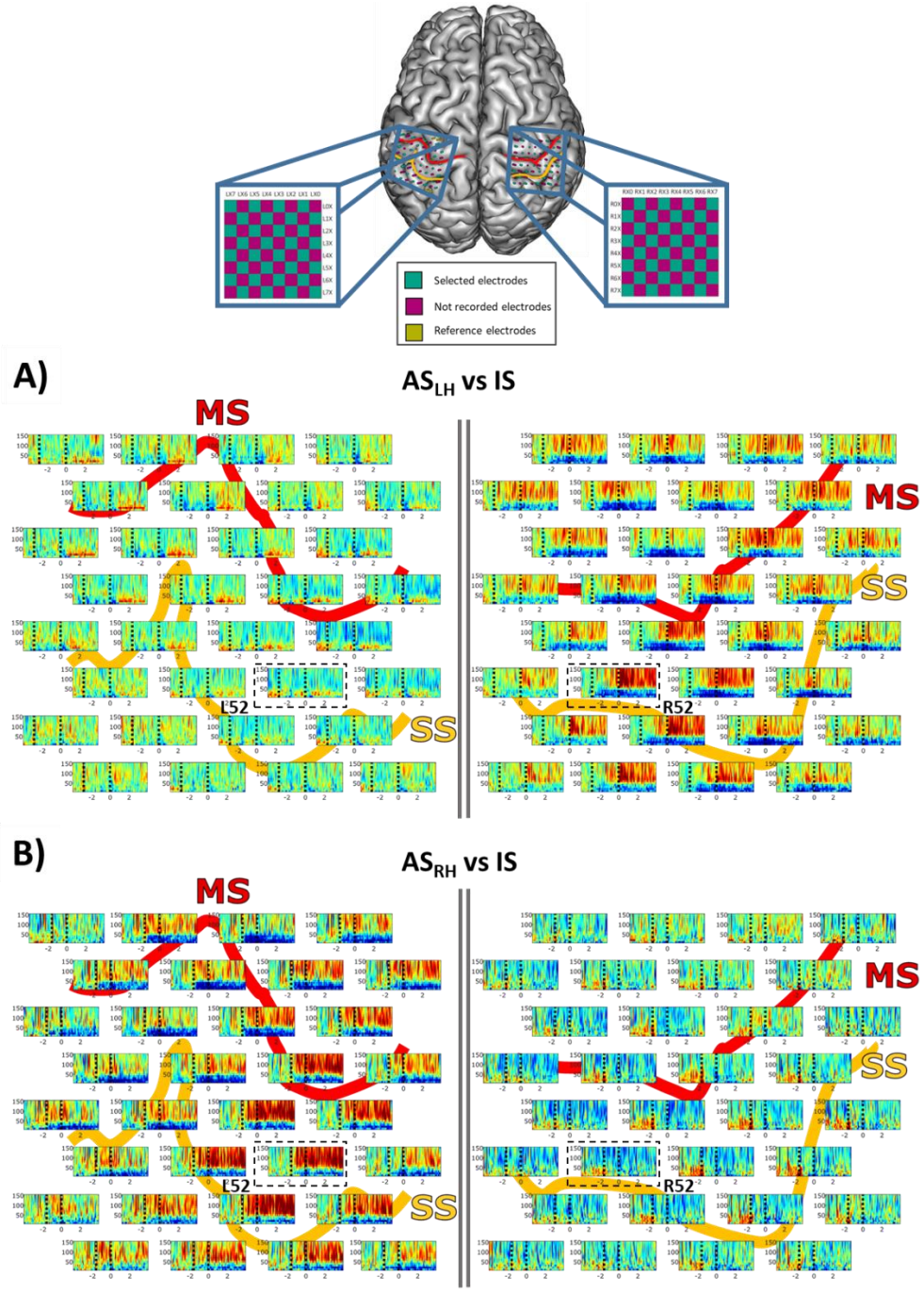



Figure 10-8: Modulation of the neural signal activity between idle state and Left or right hand translation active states . (A) Neural signal modulation between left hand translation state and idle state for the electrodes located on the left and right implants selected with a checkerboard pattern. The location of the selected electrodes are represented in the brain schematic at the top. (B) The neural signal modulation between right hand translation state and idle state for the electrodes located on the left and right implants The sensory sulcus (SS) and motor sulcus (MS) are represented in the spatial domain in yellow and red curves respectively. Dotted lines represent the median latency for each active task and the centre of studied window respectively. The dotted squares highlight the electrodes enlarged on the Figure 10-10.

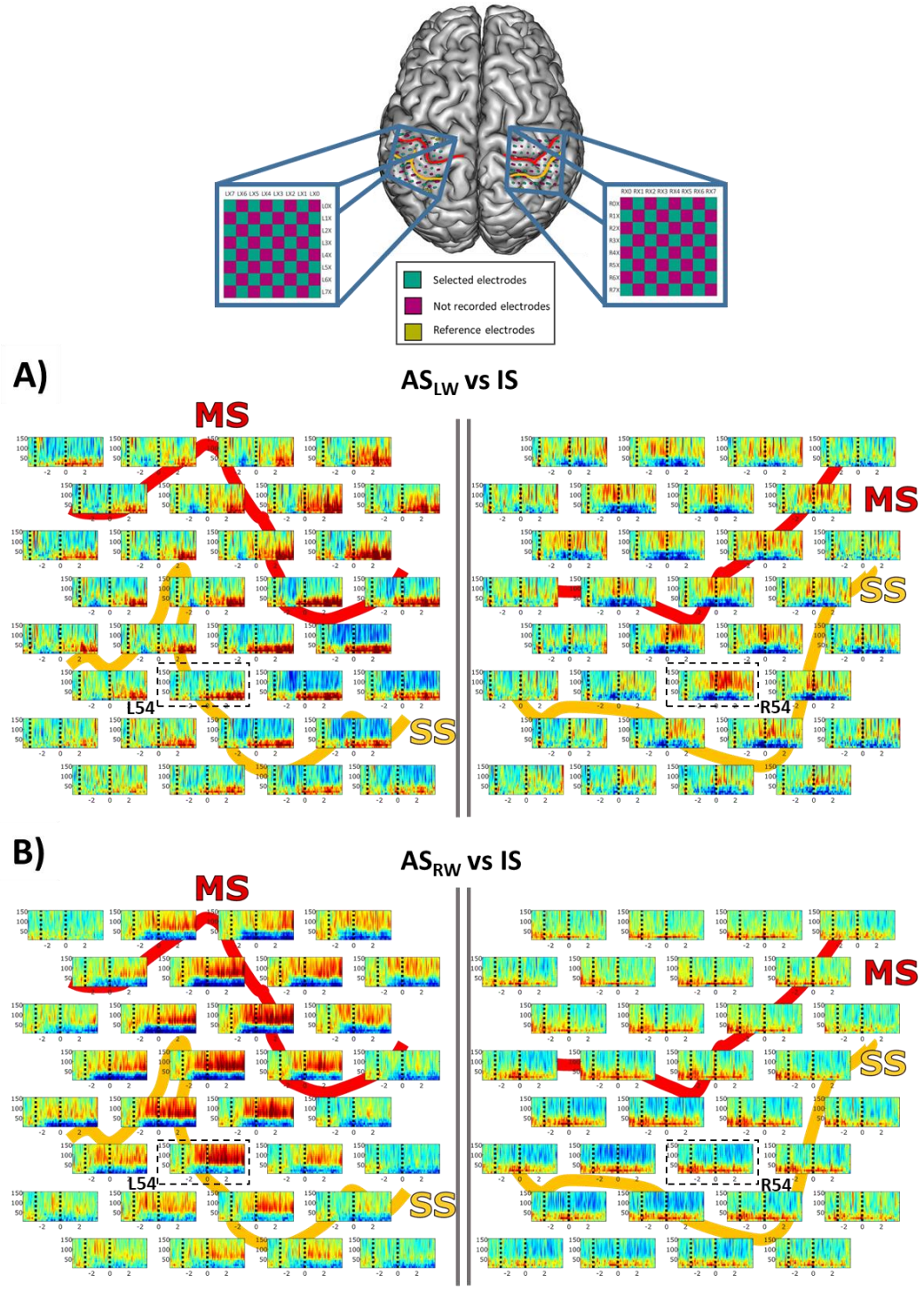



Figure 10-9: Modulation of the neural signal activity between idle state and Left and right wrist rotation active states . (A) Neural signal modulation between left wrist rotation state and idle state for the electrodes located on the left and right implants selected with a checkerboard pattern. The location of the selected electrodes are represented in the brain schematic at the top. (B) The neural signal modulation between right wrist rotation state and idle state for the electrodes located on the left and right implants The sensory sulcus (SS) and motor sulcus (MS) are represented in the spatial domain in yellow and red curves respectively. Dotted lines represent the median latency for each active task and the centre of studied window respectively. The dotted squares highlight the electrodes enlarged on the Figure 10-10.

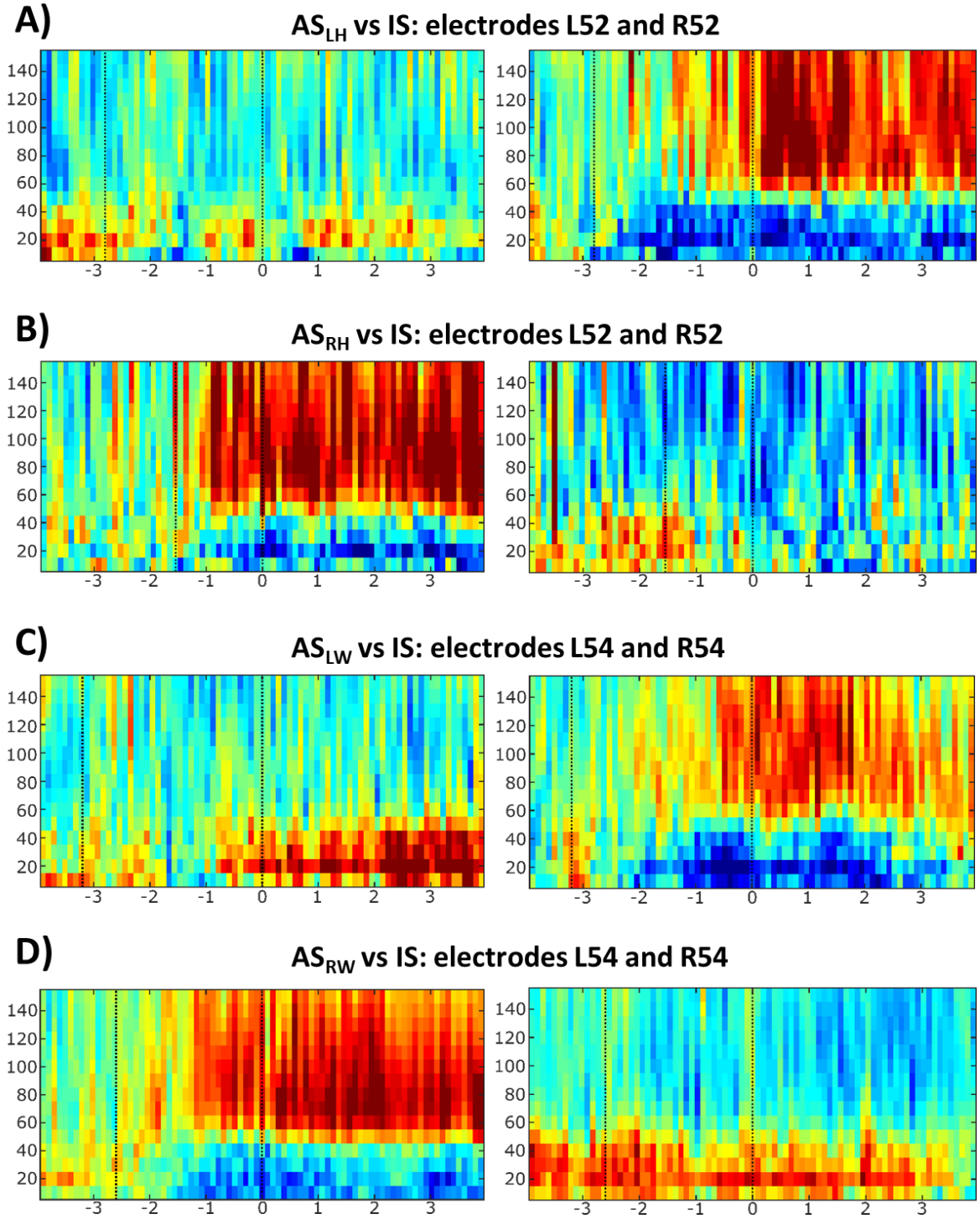



Figure 10-10 : Modulation of the neural signal activity between idle state and active states focused on specific electrodes. (A) Modulation of the neural signal activity between idle state and left hand translation state ($AS_{LH}$) focused on the electrodes of the left and right implants L52 and R52. (B) Modulation of the neural signal activity between idle state and right hand translation state ($AS_{RH}$) focused on the electrodes of the left and right implants L52 and R52. (C) Modulation of the neural signal activity between idle state and left wrist rotation state ($AS_{LW}$) focused on the electrodes of the left and right implants L54 and R54. (C) Modulation of the neural signal activity between idle state and right wrist rotation state ($AS_{RW}$) focused on the electrodes of the left and right implants L54 and R54.

### 10.1.3. Discussion

The REW-MSLM decoder was proposed to address the poorly explored field of asynchronous multi-limb effector control. ME architecture was employed to handle numerous dimensions and to decode the robust idle state. To allow cross-session training of the decoder with multiple recording conditions during closed loop BCI effectors control experiments directly, an adaptive/incremental learning algorithm was designed. Dynamic expert gating using a HMM was added to ME decoder to ensure the robustness of idle state support.

To clarify the importance of the ME model structure, which combines discrete decoding (state classification) and continuous trajectory decoding, and the importance of dynamic vs. static gating, REW-MSLM was compared to the the state-of-the-art adaptive algorithms with 3 databases using simulated pseudo-online experiments.

For discrete decoding, the REW-MSLM outperformed alternative approaches in discrete classification regardless of the dataset and the paradigm (all states decoding, IS against AS, and $AS_{LH}$-$AS_{RH}$ switching) with an averaged F-score improvement across all paradigms of 39±4% and 8.3±2% compared to REW-NPLS and REW-SLM respectively. These results sustain the benefits to train a specific model dedicated to state classification and the improvements related to dynamic classification. The switching state latency study related to the state transition delay between the instruction and the discrete decoding response demonstrated an average increase in duration by 0.45 s, 0.87 s and 0.38 s (over 3 datasets) between the discrete decoder with and without dynamic HMM processing. However, the REW-MSLM results show a drastic 92% decrease in the block error rate between the discrete decoder with and without dynamic HMM processing, overcoming the high frequency misclassified sample issue of static classifier. For physical effectors, such as an exoskeleton, which are in direct contact with the patient and has a latency of mechanical activation/deactivation of up to a few seconds, it is mandatory that false activation blocks remain exceptionally rare events.

For continuous pseudo-online experiments, REW-MSLM experts highlighted slight improvement or similar performance compared to REW-NPLS whereas the training datasets were different. REW-MSLM allows experts training using independent data sets. This may be highly profitable for progressive BCI decoder training increasing tasks complexity. In addition, considering a specific task (e.g. left hand translation), the developed REW-MSLM and the state of the art REW-NPLS performed similar left hand movement decoding. However, numerous non-desired movements (non-zero velocity predictions) was decoded for the other limb (e.g. right hand translation) with the REW-NPLS algorithm. In contrast, REW-MSLM performed similar trajectory decoding performance for the required limb without unintentional movements from the other limbs thanks to accurate state classification provided by the gate. Unintended movements of the not-intentionally controlled limb impede the control of complex effectors such as exoskeleton and all the more in the case of asynchronous control with idle state decoding. The suppression of the unintended movements leads to better visual feedbacks and concentration of the patient which may induce better model calibration.

Finally, the REW-MSLM was integrated into the homemade BCI adaptive brain signal decoder (ABSD) software platform and was used in the "BCI and tetraplegia" clinical trial. This algorithm provided to a tetraplegic patient the control of a virtual avatar and an exoskeleton in real-time with alternating rotation and translation movements of both hands on his own intention, which corresponds to 8D continuous control and supporting 5 discrete states and preserve good decoding performance for 6 months.

Figure 10-6 illustrates the gating model weights in the frequency, temporal and spatial modalities. In the frequency modality the model coefficients are consistent with the previous studies which highlighted the significance of $\beta$ and high $\gamma$-band to decode movements from direct neural signals [Bundy et al., 2016] [Volkova et al., 2019] [Waldert et al., 2009]. As expected, spatial weights were higher in the contralateral electrodes of the realized movement for both left and right hand translation and rotation which is corroborated by previous studies [Fukuma et al., 2015] [Jerbi et al., 2011] [Waldert et al., 2009].

Finally, the neural signal modulations computed between the idle state and the active states represented in the Figure 10-8, Figure 10-9 and Figure 10-10 were consistent with knowledge of the state of the art.

## 10.2. Lp-Penalized REW-NPLS offline comparative study

### 10.2.1. Sparsity study

$L_p$-Penalized REW-NPLS (PREW-NPLS) is a penalized version of the REW-NPLS algorithm which estimates group-wised sparse solution. Groups were formed following the feature modalities: grouped by electrode and or frequencies and or time. PREW-NPLS was designed in order to be employed as expert or gate model of REW-MSLM. This section highlighted the performance of $L_p$-PREW-NPLS algorithms during a pseudo-online study testing the online adaptive decoder with p being the classic $L_1$ norm regularization or less conventional $L_0$ and $L_{0.5}$ norm penalization. The PREW-NPLS models were tested in a pseudo-online study using the 3D left and right hand translations data of the 8D online closed-loop experiments (sessions from dataset D).

The PREW-NPLS algorithm has two hyperparameters, the dimension of the latent space $f \subset \{1,2,\dots,F\}$ and the penalization parameter $\lambda \in [0;1]$. Even though the latent space dimension hyperparameter is evaluated online during the model calibration using the recursive validation procedure, it is relevant to evaluate the evolution of the models sparsity depending on the two hyperparameter values. The sparsity evolutions of the $L_0$ REW-NPLS, $L_{0.5}$ REW-NPLS and $L_1$ REW-NPLS models for the left and right hand translation tasks are shown in the Figure 10-11A, B and C respectively.

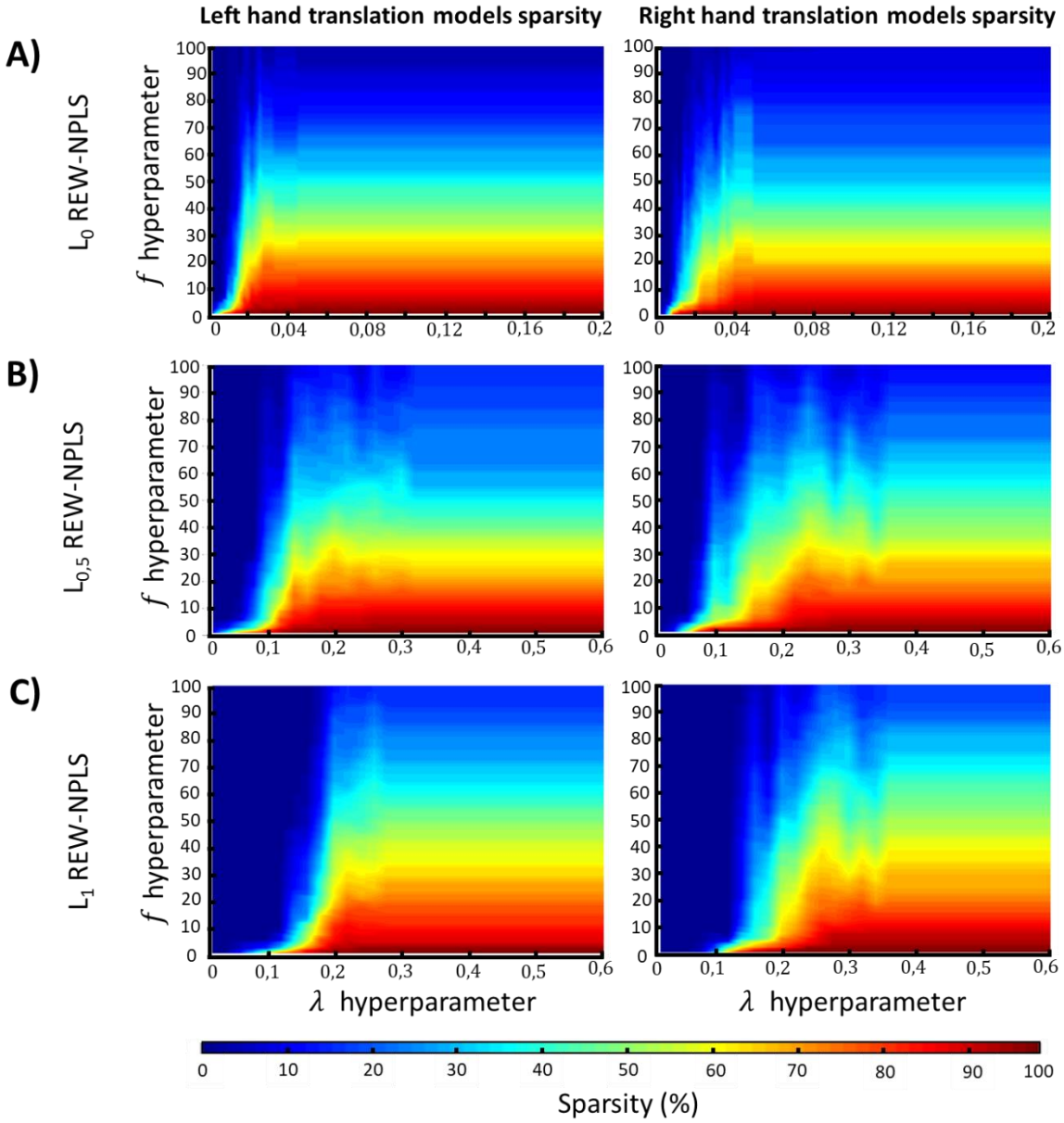



Figure 10-11 : Sparsity evolution depending on the hyperparameters $\lambda$ and $f$. Sparsity of the solution is estimated for the $L_0$ REW-NPLS (A), $L_{0.5}$ REW-NPLS (B) and $L_1$ REW-NPLS (C) algorithms depending on the $\lambda$ and $f$ where $\lambda$ is the penalization coefficient of the model and $f$ is the latent space dimension. The sparsity evolution is presented for the decoders of the left or right hand translation continuous movements.

The $SparseIdx$ of the $L_0$, $L_{0.5}$ and $L_1$ models had similar patterns. Three distinct phases can be extracted. If the penalization coefficient $\lambda$ was small, the sparsity index $SparseIdx$ was close to 0. This behavior is shown in the Figure 10-11 with $\lambda < 0.01, 0.1$ and $0.1$ for the $L_0$, $L_{0.5}$ and $L_1$ penalization norm respectively. In the opposite, for a high penalization coefficient $\lambda$, the models seemed to reach a maximum of sparsity for $\lambda$ above $0.05, 0.3$ and $0.3$ for the $L_0$, $L_{0.5}$ and $L_1$ penalization left hand translation models and $\lambda > 0.06, 0.36$ and $0.36$ for the $L_0$, $L_{0.5}$ and $L_1$

penalization right hand translation models respectively. This models from this phase are referred as "converged" models in the next sections. Between these two phases, the $SparseIdx$ value increased with the penalization hyperparameter $\lambda$ whatever the tested penalization norm and the controlled hand.

Higher was the penalization hyperparameter $\lambda$, sparser was the estimated models whereas high latent space dimension led to reduce the model sparsity. As an example, in the "stationary phase" with high penalization coefficient $\lambda$, 50% of the electrodes were set to zero with a dimension of the latent space fixed to $f = 34, 40$ and 53 whereas only 25% of the electrodes were removed from the models for $f = 60, 72$ and 85 for the $L_0$, $L_{0.5}$ and $L_1$ penalization right hand translation models respectively.

The models decoding performance and sparsity were highly dependent on $f$ and $\lambda$ hyperparameters. In the following sections the latent space dimension $f$ was set to the optimal value estimated during the pseudo-online model calibration using the conventional Recursive Validation procedure designed in the REW-NPLS algorithm [Eliseyev et al., 2017]. As mentioned, for high penalization hyperparameter $\lambda$, the sparsity was stable and the calibration led to equivalent models. Therefore, all the models penalized with a coefficient $\lambda$ going from 0 to 0.6 (0.02 steps) are not represented in the next studies for high $\lambda$ values.

## 10.2.2. PREW-NPLS decoding performance

The dot product performance and the sparsity index of the $L_0$, $L_{0.5}$ and $L_1$ models for the left and the right hand movement tasks are presented depending on the penalization coefficient $\lambda$ in the Figure 10-12 and Figure 10-13 respectively. The results are presented using the median, the 25$^{th}$ ($Q_1$) and the 75$^{th}$ ($Q_3$) percentiles using the notation: median ($Q_1 - Q_3$).

**The state of the art REW-NPLS ($\lambda = 0$)** algorithm performance in the left hand decoding study, presented in the first position of each sub-figure (Figure 10-12A, B and C), highlighted a median $= 0.223$, a $Q_1 = 0.158$ and a $Q_3 = 0.266$ which is noted $0.223$ $(0.158 - 0.266)$.

**$L_0$ REW-NPLS algorithms** (Figure 10-12A) showed relevant performance for different penalization coefficient $\lambda$ value. Obviously the $SparseIdx$ indicator increased with higher penalization hyperparameter value. However, the dot product (cosine similarity) highlighted better performance than REW-NPLS algorithm with various $\lambda$ value. For $\lambda = 0.01$, the $SparseIdx = 0\%$ but the dot product (cosine similarity) was evaluated at $0.252$ $(0.165 - 0.296)$. For $\lambda = 0.026$, the cosine similarity was estimated at $0.248$ $(0.173 - 0.288)$ with a $SparseIdx = 56{,}25\%$. For $\lambda = 0.04$ to $\lambda = 0.046$ and for $\lambda > 0.046$ PREW-NPLS demonstrated a $0.248$ $(0.162 - 0.294)$ and $0.236$ $(0.146 - 0.268)$ continuous decoding performance with 40 ($SparseIdx = 62.5\%$) and 41 ($SparseIdx = 64.06\%$) electrode parameter weights over 64 set to zero value.

**$L_{0.5}$ REW-NPLS algorithm** (Figure 10-12B), similarly to $L_0$ REW-NPLS algorithms presented equal decoding performance than REW-NPLS algorithm with small decoding performance improvements for some models. For $\lambda = 0.22$ with 18 electrodes parameter weights set to zero ($SparseIdx = 28.13\%$) the PREW-NPLS model highlighted higher cosine similarity

performance 0.253 $(0.189 - 0.301)$ than the REW-NPLS model. Additionally, for $\lambda = 0.3$, the dot product was estimated at $0.245$ $(0.156 - 0.2838)$ with a sparsity index $SparseIdx = 35.94\%$. Finally, for $\lambda > 0.32$ the models converged to a sparse solution with $SparseIdx = 28.13\%$ and continuous decoding performances similar to REW-NPLS model: $0.217$ $(0.143 - 0.261)$.

**$L_1$ REW-NPLS algorithm** (Figure 10-12C) highlighted similar results than $L_0$ and $L_{0.5}$ REW-NPLS algorithms. For $\lambda = 0.12$, with a sparsity of $SparseIdx = 0\%$, the PREW-NPLS model highlighted a $0.253$ $(0.151 - 0.286)$ cosine similarity. A sparsity index of $SparseIdx = 29.69\%$ is reached for $\lambda = 0.20$ with a decoding performance of $0.249$ $(0.162 - 0.295)$. Finally, for $\lambda > 0.34$, 41 electrodes parameter weights are set to zero value leading to a $0.245$ $(0.173 - 0.283)$.

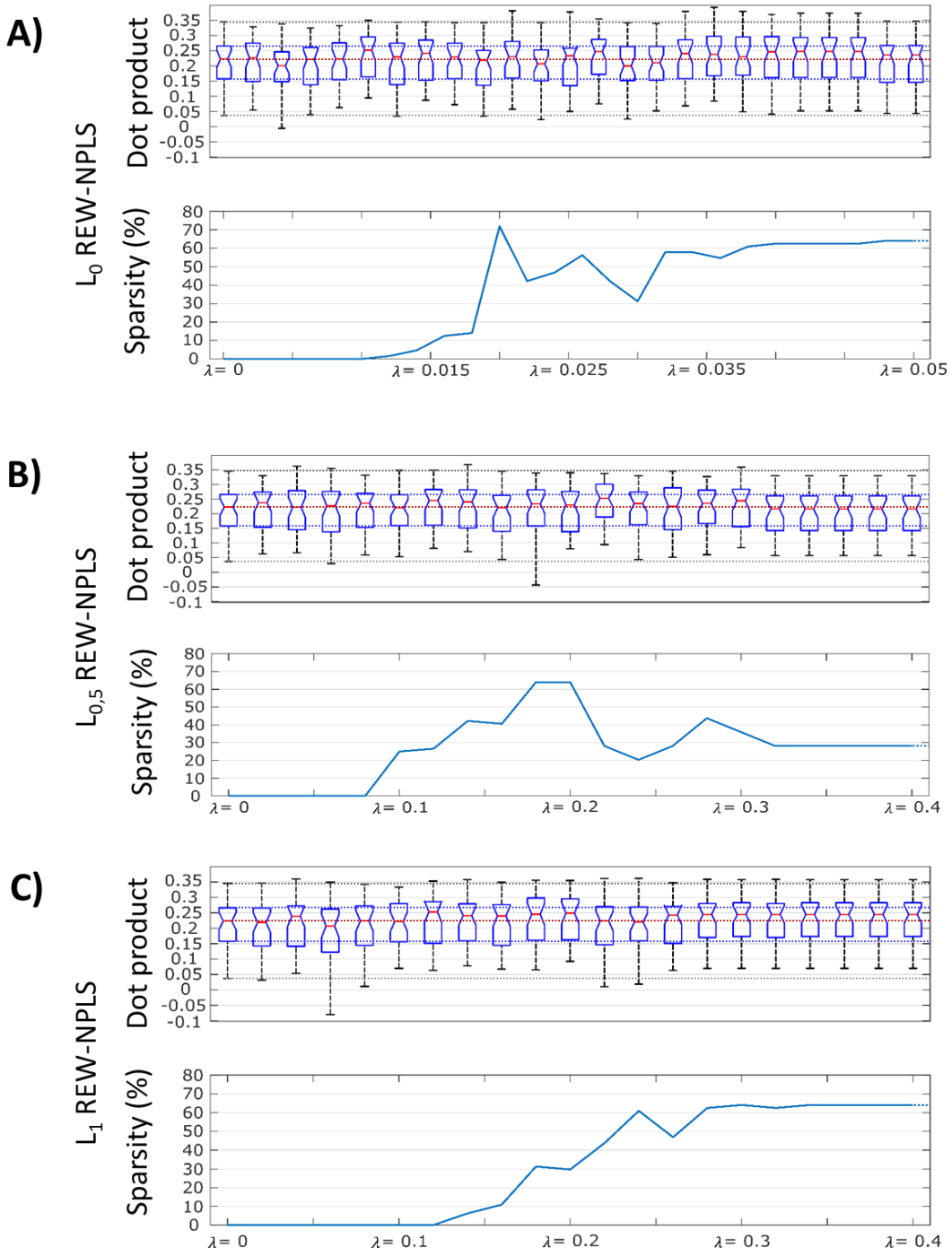



Figure 10-12 : The model performance indicators of the $L_P$ REW-NPLS algorithm for left hand movement decoding. The cosine similarity and the model sparsity were computed for the $L_0$ REW-NPLS (A), $L_{0.5}$ REW-NPLS (B) and $L_1$ REW-NPLS (C) algorithms. The cosine similarly performance on each session was summarized using a box plot representation where the red line is the median the blue lines indicate the 25th and 75th percentiles ($Q_1$ and $Q_3$). Additionally, the black boundaries show the upper and lower extreme cosine similarity obtained for the experiments. The performance of the REW-NPLS algorithms is presented in the first box plot of each $L_P$ REW-NPLS algorithm sub-plot. The median, $Q_1$ and $Q_3$ of the REW-NPLS models are extended using horizontal dotted lines for easier performance comparison. Additionally, the sparsity of each solution is depicted.

A similar pseudo-online study was performed to decode the right hand translation movements. The results are presented in the Figure 10-13. The right arm decoding study stressed worse cosine similarity than the left arm decoding study.

**The state of the art REW-NPLS ($\lambda = 0$) model** performance for the right arm translation decoding are showed in the first position of Figure 10-13A, B and C. REW-NPLS algorithm highlighted a cosine similarity of $0.127$ $(0.0468 - 0.155)$.

**$L_0$ REW-NPLS algorithm** (Figure 10-13A) shows performance improvements with sparse solutions for different penalization hyperparameter $\lambda$. For $\lambda = 0.01$, the $SparseIdx = 4.68\%$ corresponding to only 3 electrode parameter weights set to zero value but the dot product (cosine similarity) was evaluated at $0.157$ $(0.1018 - 0.203$. These performance represent a cosine similarity enhancement of 24%, 117% and 30% for the median, the $Q_1$ and $Q_3$ metrics respectively. For $\lambda = 0.018$, the cosine similarity was estimated at $0.157$ $(0.0989 - 0.185)$ with a $SparseIdx = 37.5\%$. For $\lambda = 0.024$, sparser solution was obtained with a sparsity index of $SparseIdx = 45.31\%$ and a cosine similarity estimated at $0.153$ $(0.0786 - 0.198)$. For $\lambda > 0.04$ $L_0$ PREW-NPLS models converged to sparse solution with 48 electrodes over 64 ($SparseIdx = 75\%$) removed from the model which highlighted decoding performance similar to the state of the art REW-NPLS. The best performance of the models with $\lambda > 0.04$ was estimated at $0.128$ $(0.058 - 0.168)$.

**$L_{0.5}$ REW-NPLS algorithm** (Figure 10-13B), similarly to $L_0$ REW-NPLS decoder, highlighted better decoding performance than REW-NPLS with sparser solutions for some penalization parameter $\lambda$. For $\lambda = 0.1$, 23 electrodes were removed from the model ($SparseIdx = 35.94\%$) and the cosined similarity was estimated at $0.136$ $(0.100 - 0.177)$. With higher penalization parameter $\lambda = 0.16$ sparser model was computed with $SparseIdx = 54.69\%$ without decreasing the decoding performance $DotP = 0.150$ $(0.0881 - 0.176)$. The sparsest models were obtained for $\lambda = 0.26$ and $\lambda = 0.28$ showing a sparsity index of $SparseIdx = 79.69\%$ and $SparseIdx = 78.13\%$ respectively. Finally, the models converged to the same solutions for $\lambda > 0.36$ with a sparsity of $SparseIdx = 68.75\%$ (44 electrodes removed from the final solution) and a cosine similarity of $DotP = 0.131$ $(0.0835 - 0.186)$.

**$L_1$ REW-NPLS algorithm** (Figure 10-13C) results show better decoding performance than REW-NPLS algorithm for numerous penalization parameter $\lambda$. Several models with small penalization parameter $\lambda = 0.04, 0.06$ and $0.1$ without setting any electrode to zero ($SparseIdx = 0\%$) highlighted a cosine similarity of $DotP = 0.154$ $(0.0915 - 0.202)$, $DotP = 0.158$ $(0.0791 - 0.184)$ and $DotP = 0.164$ $(0.0959 - 0.191)$ representing a median improvements of 21%, 24% and 29% respectively. Similar decoding performance were obtained for higher penalization parameter $\lambda = 0.22$ and $\lambda = 0.26$ with a dot product indicator of $DotP = 0.154$ $(0.101 - 0.192)$ and $DotP = 0.152$ $(0.0872 - 0.197)$ but with 33 (51.56%) and 44 (68.75%) electrodes parameters weights set to zero, respectively. Finally, for a penalization parameter $\lambda > 0.38$, the models calibration stabilized to a solution with a sparsity indicator of $SparseIdx = 68.75\%$ with a decoding performance of $DotP = 0.131$ $(0.0835 - 0.186)$.

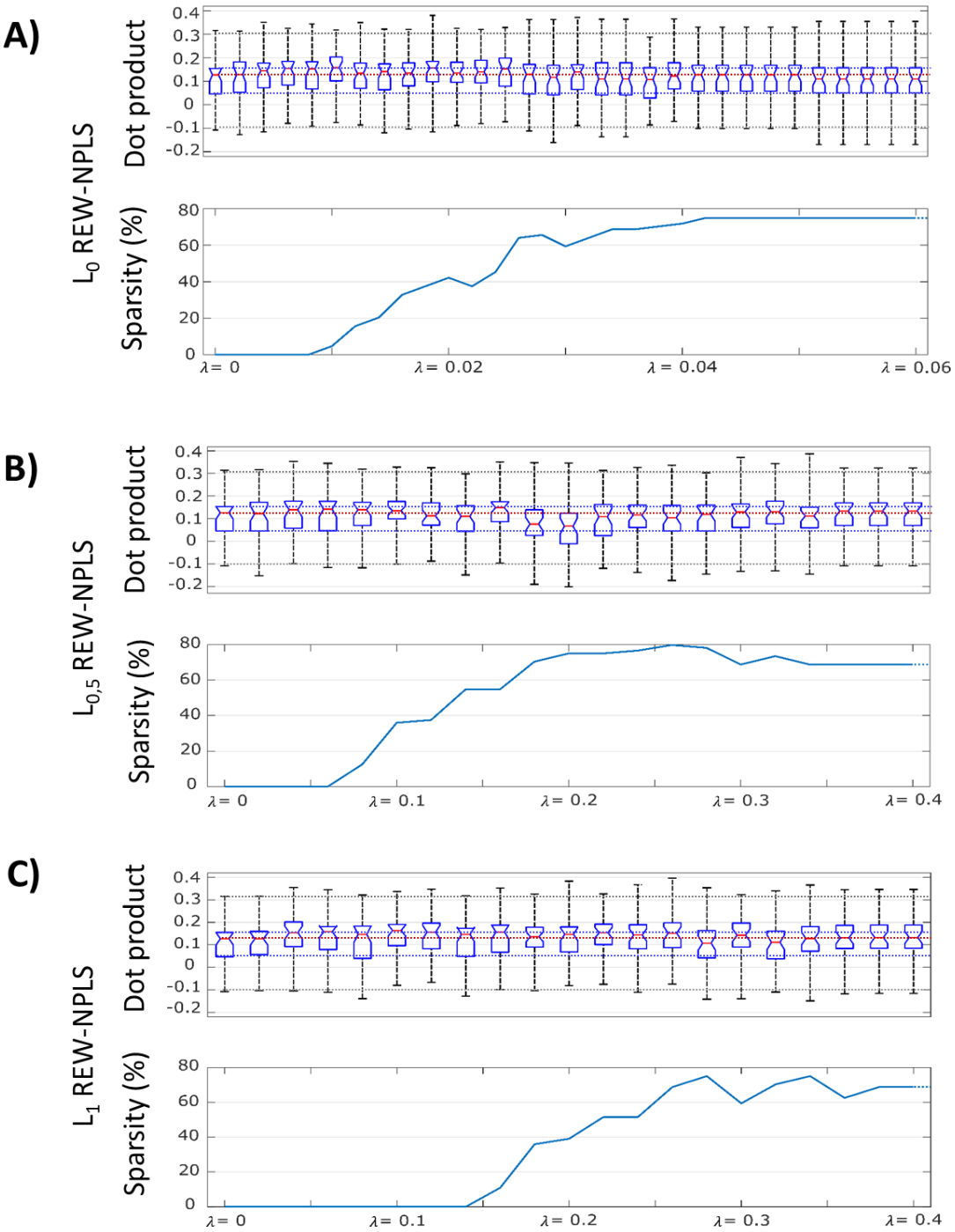



Figure 10-13: The model performance indicators of the $L_p$ REW-NPLS algorithm (p=0,0.5,1) for right hand movement decoding. The cosine similarity and the model sparsity is computed for the $L_0$ REW-NPLS (A), $L_{0.5}$ REW-NPLS (B) and $L_1$ REW-NPLS (C) algorithms. The cosine similarly performance on each session is summarized using a box plot representation where the red line is the median the blue lines indicate the 25th and 75th percentiles ($Q_1$ and $Q_3$). Additionally, the black boundaries show the upper and lower extreme cosine similarity obtained for the experiments. The performance of the REW-NPLS algorithms is presented in the first box plot of each $L_p$ REW-NPLS algorithm sub-plot. The median, $Q_1$ and $Q_3$ of the REW-NPLS models are extended using horizontal dotted lines for easier performance comparison. Additionally, the sparsity of each solution is depicted.

Statistical differences in the cosine similarity indicator were highlighted between the state of the art REW-NPLS algorithm and the $L_0$, $L_{0.5}$ and $L_1$ REW-NPLS models estimated for the left and right hand translation tasks. The statistical analysis performed with the non-parametric paired Wilcoxon signed rank test with and without the Bonferroni correction are represented for the left hand translation decoding in the Table 4 and in the Table 5 for the right hand translation decoding. Numerous models highlighted statistical performance difference with the REW-NPLS model. As examples of statistical performance difference among the previously mentioned models, the $L_0$-PREW-NPLS performance highlighted statistical differences with $\lambda \in \{0.01,0.026,0.04\}$ for the left hand translation study and $\lambda \in \{0.01,0.018,0.24\}$ for the right hand translation decoding study. For $L_{0.5}$-PREW-NPLS differences were stressed for $\lambda \in \{0.22,0.03\}$ and $\lambda \in \{0.1,0.16,0.04\}$ for the left and right hand translation study. Finally, $L_1$-PREW-NPLS showed statistical performance differences with REW-NPLS algorithm for the left hand translation tasks with $\lambda \in \{0.12,0.34,0.36\}$ as well as for the right hand translation decoding study with $\lambda \in \{0.04,0.06,0.1, 0.22, 0.26, 0.38\}$.

Table 4: Significance of the differences between the REW-NPLS decoder and the L0, L0.5 or L1 PREW-NPLS algorithm in the pseudo-online left hand translation decoding study using the non-parametric paired Wilcoxon signed rank test with and without the Bonferroni correction ($\alpha = 0.05$ and $\alpha_{Bonferroni} = 0.00161$).The bolded P-values represent the statistical differences with $\alpha = 0.05$ whereas the bolded underlined P-values highlight the statistical difference with the Bonferroni correction.

| $\lambda$ | 0.02 | 0.04 | 0.06 | 0.08 | 0.10 | 0.12 | 0.14 | 0.16 | 0.18 | 0.2 | 0.22 | 0.24 | 0.26 | 0.28 | 0.30 | 0.32 | 0.34 | 0.36 | 0.38 | 0.40 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $L_1$ | 0.73 | 0.15 | **0.021** | **0.018** | 0.53 | **0.0016** | **0.0005** | 0.24 | **0.0012** | 0.5 | 0.15 | **0.027** | 0.24 | **0.0023** | 0.24 | 0.21 | 0.24 | 0.21 | 0.21 | 0.21 |
| $L_{0.5}$ | 0.21 | 0.84 | 0.39 | 0.99 | 0.21 | **0.025** | 0.07 | 0.06 | 0.47 | 0.96 | **0.0003** | 0.59 | 0.83 | 0.18 | **0.007** | **0.0005** | **0.0005** | **0.0005** | **0.0005** | **0.0005** |

| $\lambda$ | 0.002 | 0.004 | 0.006 | 0.008 | 0.010 | 0.012 | 0.014 | 0.016 | 00.18 | 0.02 | 0.022 | 0.024 | 0.026 | 0.028 | 0.030 | 0.032 | 0.034 | 0.036 | 0.038 | 0.040 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $L_0$ | 0.22 | $\mathbf{10^{-4}}$ | **0.012** | 0.45 | **0.0006** | 0.34 | **0.01** | 0.11 | **0.009** | 0.40 | $\mathbf{10^{-5}}$ | 0.10 | **0.04** | **0.0017** | 0.13 | 0.47 | **0.012** | **0.044** | **0.013** | **0.008** |

| $\lambda$ | 0.042 | 0.044 | 0.46 | 0.48 | 0.50 |
|---|---|---|---|---|---|
| $L_0$ | **0.008** | **0.008** | 0.67 | 0.67 | 0.67 |

Table 5: Significance of the differences between the REW-NPLS decoder and the L0, L0.5 or L1 PREW-NPLS algorithm in the pseudo-online right hand translation decoding study using the non-parametric paired Wilcoxon signed rank test with and without the Bonferroni correction ($\alpha = 0.05$ and $\alpha_{Bonferroni} = 0.00161$).The bolded P-values represent the statistical differences with $\alpha = 0.05$ whereas the bolded underlined P-values highlight the statistical difference with the Bonferroni correction.

| $\lambda$ | 0.02 | 0.04 | 0.06 | 0.08 | 0.10 | 0.12 | 0.14 | 0.16 | 0.18 | 0.2 | 0.22 | 0.24 | 0.26 | 0.28 | 0.30 | 0.32 | 0.34 | 0.36 | 0.38 | 0.40 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $L_1$ | **0.005** | **$\underline{10^{-5}}$** | **$\underline{0.0002}$** | 0.25 | **$\underline{10^{-6}}$** | **$\underline{10^{-5}}$** | **0.038** | **$\underline{0.0004}$** | **$\underline{10^{-5}}$** | **$\underline{0.0016}$** | **$\underline{10^{-6}}$** | **$\underline{0.0004}$** | **$\underline{10^{-6}}$** | 0.22 | **0.01** | **0.05** | **0.01** | **0.005** | **0.003** | **0.003** |
| $L_{0.5}$ | 0.26 | 0.08 | 0.07 | **0.012** | **$\underline{0.00018}$** | 0.16 | **0.013** | **$\underline{10^{-4}}$** | **$\underline{0.0001}$** | **$\underline{10^{-5}}$** | **0.007** | 0.72 | 0.27 | 0.15 | 0.36 | 0.17 | 0.455 | 0.17 | 0.17 | 0.17 |

| $\lambda$ | 0.002 | 0.004 | 0.006 | 0.008 | 0.010 | 0.012 | 0.014 | 0.016 | 00.18 | 0.02 | 0.022 | 0.024 | 0.026 | 0.028 | 0.030 | 0.032 | 0.034 | 0.036 | 0.038 | 0.040 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $L_0$ | 0.88 | **0.015** | **$\underline{0.001}$** | **$\underline{10^{-6}}$** | **$\underline{10^{-6}}$** | **0.009** | 0.13 | **0.006** | **$\underline{10^{-6}}$** | **$\underline{10^{-5}}$** | **$\underline{0.0001}$** | **$\underline{10^{-5}}$** | 0.77 | **0.05** | **0.03** | 0.41 | 0.41 | 0.12 | 0.11 | 0.11 |

| $\lambda$ | 0.042 | 0.044 | 0.46 | 0.48 | 0.50 | 0.052 | 0.054 | 0.56 | 0.58 | 0.60 |
|---|---|---|---|---|---|---|---|---|---|---|
| $L_0$ | 0.11 | 0.11 | 0.11 | 0.11 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 | 0.18 |

### 10.2.3. $L_p$ REW-NPLS parameters visualization

The REW-NPLS and the $L_p$ REW-NPLS model parameter weights are illustrated on the temporal, frequency and spatial domain in the Figure 10-14 for the left hand translation models and in the Figure 10-15 for the right hand translation models. For easier comparison and selection the presented models are the ones with "converged" penalization hypeparameter $\lambda > 0.06, 0.4$ and $0.4$ for $L_0$, $L_{0.5}$ and $L_1$ REW-NPLS algorithms respectively whereas the latent space dimension $f$ was fixed using the Recursive-Validation (RV) of PREW-NPLS algorithm similar to the Recursive-Validation procedure of the REW-NPLS algorithm. $L_p$ REW-NPLS model parameter weights of the left hand translation models (Figure 10-14) in the frequency domain were higher in the low frequency bands than REW-NPLS model. In the opposite, the models estimated on the offline right hand translation study showed dominant parameter weights in the high frequency bands. The temporal parameter weights were similar for all the algorithms. As previously mentioned, the sparsity of the solution was highly dependent on the latent space dimension hyperparameter $f$. The $L_{0.5}$ and $L_1$ REW-NPLS algorithms, with small latent space dimensions ($f = 26$ and $f = 32$), led to sparser solution than the $L_0$ REW-NPLS model with a latent space dimension $f = 52$. However, similarities between the dominant electrodes of the $L_p$ REW-NPLS solutions and between the $L_p$ REW-NPLS and REW-NPLS are visible.

For easier visualization, the spatial parameter weights are presented for the left (Figure 10-16) and right (Figure 10-17) hand translation models on a map with the electrode locations relative to the sensory (SS) and motor (MS) sulci. As most of the parameter weights were fixed to zero value in $L_p$ REW-NPLS algorithms, the amplitude of the parameters are much more important than the REW-NPLS model weights which are more balanced between all the electrodes. For both, left and right hand models, $L_p$, $L_{0.5}$ and $L_1$ REW-NPLS solutions used electrodes located in similar regions.

For the left hand models (Figure 10-16), the contralateral electrodes (right implant) highlighted strong influence on the movement decoding. In particular, the region located around the electrodes R21, R23, R24, R25, R36 and R47 above the lower part of the MS and the upper part of the SS show large parameter weights for the $L_0$, $L_{0.5}$ and $L_1$ REW-NPLS algorithms. Additionally, the electrodes positioned on the upper left side of the right MS stressed major influenced in the left arm decoding.

Similarly, for the right hand models (Figure 10-17), the prominence of the contralateral electrodes (left implant) was visible for the REW-NPLS and all the penalized version of REW-NPLS algorithms. Important spatial parameter weights were stressed for the electrodes located in the upper part of the MS named electrodes L05, L03 and L14 as well as the electrodes positioned in the lower part of the MS referred as electrodes L41, L43 and L50. Similarly to left hand models, the important parameter weights were noticeable on the ipsilateral implant.

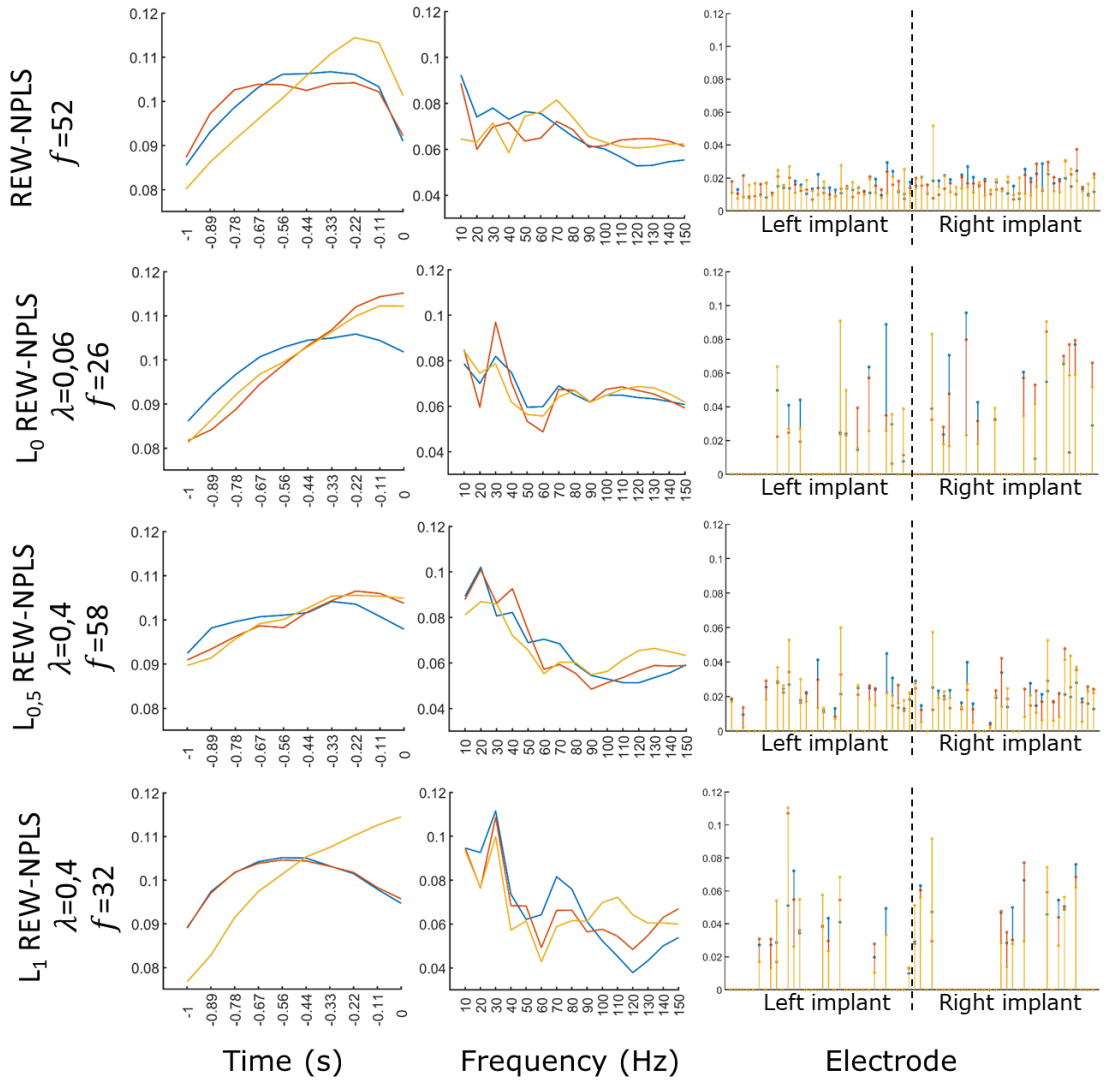



Figure 10-14: Parameter weights of the Lp REW-NPLS and REW-NPLS models estimated offline in the left arm decoding study. Model parameter weights of the tested algorithms for 3D left hand translation movements decoding from the D dataset according to the spatial, frequency or temporal modalities. The parameter weights related to the $y_1$, $y_2$ and $y_3$ axis are represented using blue, orange and yellow lines respectively.

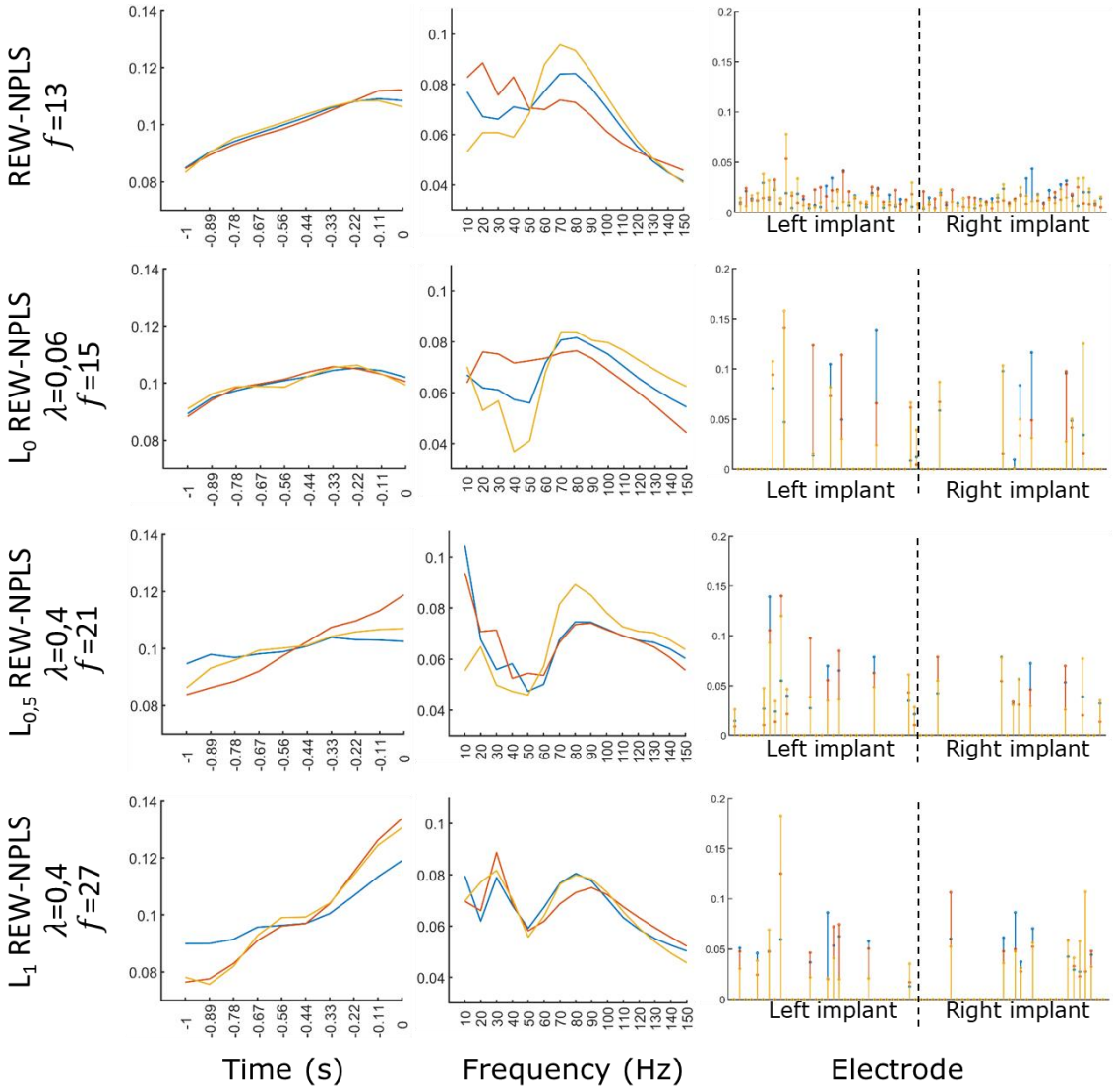



Figure 10-15: Parameter weights of the Lp REW-NPLS and REW-NPLS models estimated offline in the right arm decoding study. Model parameter weights of the tested algorithms for 3D right hand translation movements decoding from the D dataset according to the spatial, frequency or temporal modalities. The parameter weights related to the $y_4$, $y_5$ and $y_6$ axis are represented using blue, orange and yellow lines respectively.

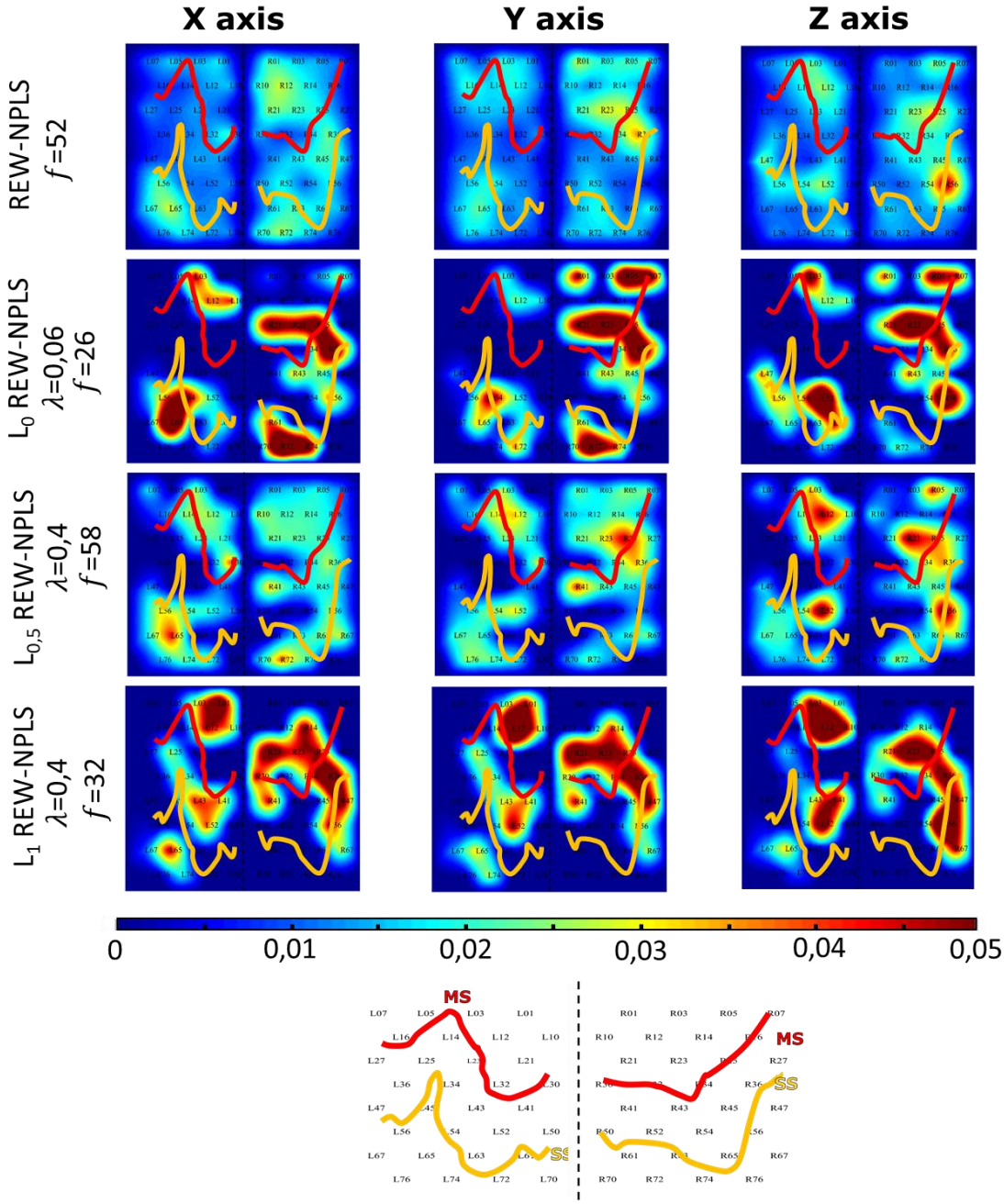



Figure 10-16 : 3D left hand decoding parameter weights of the three PREW-NPLS models projected on the spatial modality depending on the electrode location on the implant. The optimal latent space dimension $f$ estimated using the Recursive-Validation procedure. The sensory sulcus (SS) and motor sulcus (MS) are represented in the spatial domain in yellow and red curves respectively.

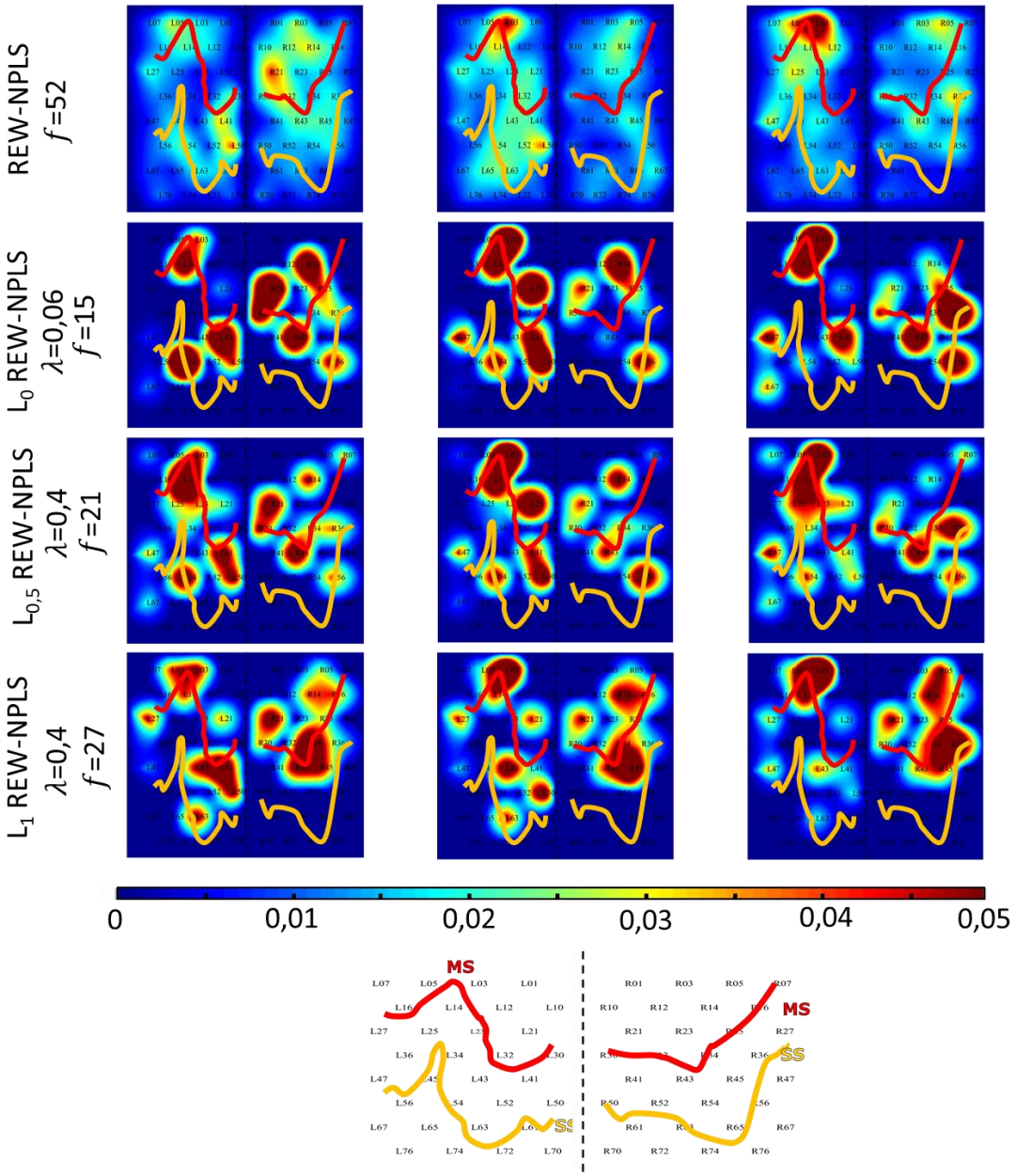



Figure 10-17 : 3D right hand decoding parameter weights of the three PREW-NPLS models projected on the spatial modality depending on the electrode location on the implant. The optimal latent space dimension $f$ estimated using the Recursive-Validation procedure is used. The sensory sulcus (SS) and motor sulcus (MS) are represented in the spatial domain in yellow and red curves respectively.

### 10.2.4. Discussion

The study was based on pseudo-online decoding of the left or right hand translation movements recorded during the online closed-loop experiments. The dataset D is composed of 43 experiments. The tested models were calibrated during the offline study using the first 6 experiments (recorded in late September 2018) and was tested based on the experiments recorded between early October 2018 and mid-March 2019.

To be as close as possible to the online experiment settings, the penalized models were calibrated on the same experiments which were used during the online-closed loop experiments. The number of training session was small (14%) and focused at the beginning of the series of experiments (no re-calibration period). This may explain the high inter-session variability of the decoding performance for both REW-NPLS and PREW-NPLS algorithms.

However, the $L_0$, $L_{0.5}$ and $L_1$-PREW-NPLS algorithms highlighted equivalent or better decoding performance than REW-NPLS decoder using sparse solutions with up to 41 and 48 of the electrode parameter weights set to zero value for the left and right hand translation $L_1$ norm penalized models. Decoding performance improvements were more evident for the 3D right hand translation models than the left hand translation model. Sparse solutions allow removing the majority of the electrodes which may reduce the required computational burden for the model recalibration.

Additionally, $L_0$, $L_{0.5}$ and $L_1$ REW-NPLS algorithms converged to similar solutions with comparable decoding performance. However, the $L_{0.5}$ REW-NPLS algorithm is looking for a solution of a cubic equation which requires higher computational load to be solved than the calibration procedure with the $L_0$ and $L_1$ norm penalization. Indeed, the equation solving required for the $L_{0.5}$ pseudo-norm penalization is applied at every loop of the PARAFAC algorithm which is repeated several times for every latent space dimension ($f$) of the PREW-NPLS model. The repeated equation solving procedure of the $L_{0.5}$ pseudo-norm penalization is more complicated than the thresholding procedure applied with the $L_0$ and $L_1$ norm penalization. Consequently, $L_{0.5}$ REW-NPLS algorithm may not be adapted to online CLDA. $L_0$ and $L_1$ REW-NPLS algorithms highlighted decoding performance and computational requirements more adapted for an integration into REW-MSLM experts or gating algorithms.

## 10.3. Automatic PREW-MSLM offline comparative study

### 10.3.1. Sparsity study

Automatic $L_1$-Penalized REW-NPLS is a penalized version of the REW-NPLS algorithm which estimates several PREW-NPLS models during the calibration phase, ranks the model depending on the estimated performance and automatically selects the best

model with the adequate penalization hyperparameter. Similarly to the PREW-NPLS algorithm, the APREW-NPLS algorithm was tested in a pseudo-online study using the 3D left and right hand translation data of the 8D online closed-loop experiments (sessions from dataset D). The APREW-NPLS calibration procedure was performed on the same experiments than the REW-MSLM during the online closed-loop experiments. The study was focused on the $L_1$ norm penalization but can be adapted to the $L_0$ and $L_{0.5}$ norm penalization type.

The Figure 10-18 shows the sparsity of all the models estimated by the APREW-NPLS algorithm during the pseudo-online study depending on the set of available penalization hyperparameter $\lambda$ and the latent space dimension hyperparameter $f$ without considering the estimated ranking of the models. Similarly to the PREW-NPLS algorithms, the sparsity is small for $\lambda = 0.1$ and is increasing with the penalization factor. For $\lambda > 0.3$ the models did not converge to similar solutions as the PREW-NPLS algorithms because all the models were not trained with the same number of update increments.

For the left hand decoders with $\lambda > 0.3$, the models converged to a solution with 50% of the electrode parameter weights set to zero with a dimension of the latent space close to $f = 50$ whereas 25% of the electrodes were removed for $f \approx 83$. The right hand translation models show higher sparsity variability than left hand models for $\lambda > 0.3$.

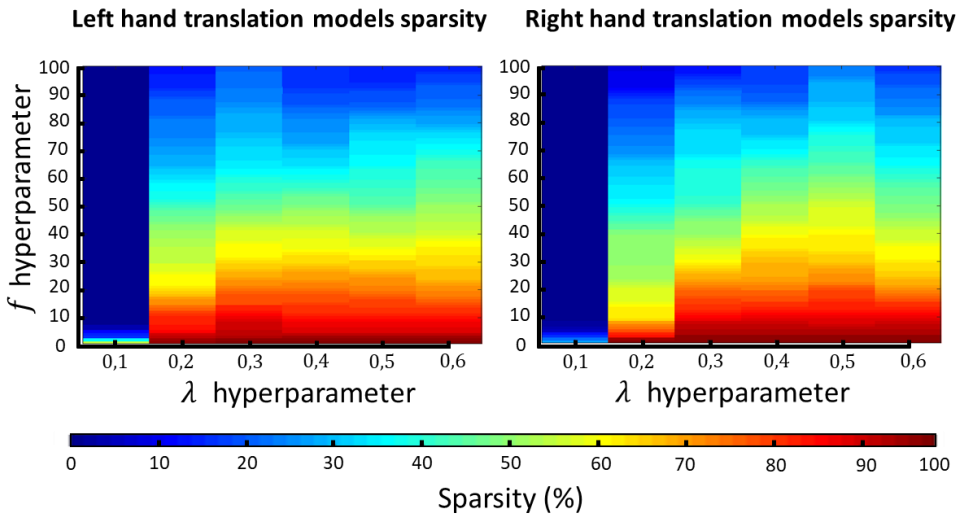



Figure 10-18 : Sparsity evolution of the APREW-NPLS models depending on the selected hyperparameters . Sparsity of the solution estimated using the $L_1$ APREW-NPLS algorithm depending on the two hyperparameters $\lambda$ and $f$ where $\lambda$ is the penalization coefficient of the model and $f$ is the latent space dimension. The sparsity evolution is presented for the estimated decoders of the left or right hand continuous movements.

## 10.3.2. APREW-NPLS decoding performance

The results of the 3D left and right hand translation decoding studies are displayed in the Figure 10-19A and Figure 10-19B respectively. For each study, the decoding performance and the sparsity indicators were evaluated for the six models obtained using the APREW-NPLS model calibration procedure. In addition to the performance of the models on the test dataset, the rank of each model estimated during the last model calibration iteration is tagged. The performance of the models is compared to the state of the art REW-NPLS algorithm. The REW-NPLS decoder highlighted a cosine similarity of $DotP = 0.223\ (0.158 - 0.266)$ and $DotP = 0.127\ (0.0468 - 0.155)$ in the pseudo-online left and right hand translation decoding studies respectively.

**For the left hand translation models** (Figure 10-19A), the APREW-NPLS model with $\lambda = 0.1$ was ranked last during the model calibration procedure and achieved the worst decoding performance with a cosine similarity of $DotP = 0.168\ (0.106 - 0.245)$ without setting any parameter weight to zero ($SparseIdx = 0\%$). For $\lambda = 0.2$, the model was ranked 3rd . The sparsity index reached $SparseIdx = 60.94\%$. The decoding performance stressed small improvements compared to REW-NPLS algorithm with a cosine similarity of $0.241\ (0.151 - 0.300)$. The models characterized with a penalization coefficient of $\lambda = 0.3$ ranked penultimate models according to APREW-NPLS algorithm highlighted similar results than REW-NPLS algorithms with a dot product indicator of $0.212\ (0.160 - 0.270)$ but with multiple electrodes parameter weights set to zero: $SparseIdx = 78.13\%$. The APREW-NPLS algorithm assigned the best ranked to the model with $\lambda = 0.4$ which highlighted good decoding performance $0.243\ (0.160 - 0.290)$ with a sparse model $SparseIdx = 73.44\%$. The last two models with the penalization coefficient $\lambda = 0.5$ and 0.6 showed a decoding performance of $0.244\ (0.142 - 0.270)$ and $0.237\ (0.149 - 0.290)$ with a sparsity index of $SparseIdx = 62.5\%$ and $71.88\%$ respectively. These models were rated as fourth and second best models.

**For the right hand models** (Figure 10-19B), the APREW-NPLS models highlighted better decoding performance than REW-NPLS algorithms for $\lambda = 0.1, 0.2$ and 0.6 which were ranked 6th,5th and 2nd best models according to APREW-NPLS calibration procedure. These models exhibited a decoding performance of $0.160\ (0.0766 - 0.194)$ for $\lambda = 0.1$, $0.155\ (0.0898 - 0.174)$ for $\lambda = 0.2$ and $0.145\ (0.0841 - 0.197)$ for $\lambda = 0.6$ with a sparsity index of $SparseIdx = 0\%, 56.35$ and $60.94\%$ respectively. These APREW-NPLS models exhibited a median cosine similarity improvement of $26\%, 22\%$ and $14\%$ respectively compared to the median REW-NPLS decoding performance. The APREW-NPLS with a penalization hyperparameter equal to $\lambda = 0.3$, 0.4 and 0.5 presented similar results than REW-NPLS algorithm with a sparsity index between $SparseIdx = 39.06\%$ and $SparseIdx = 71.88\%$. These models were classified 3rd, 1st, and 4th best models respectively.

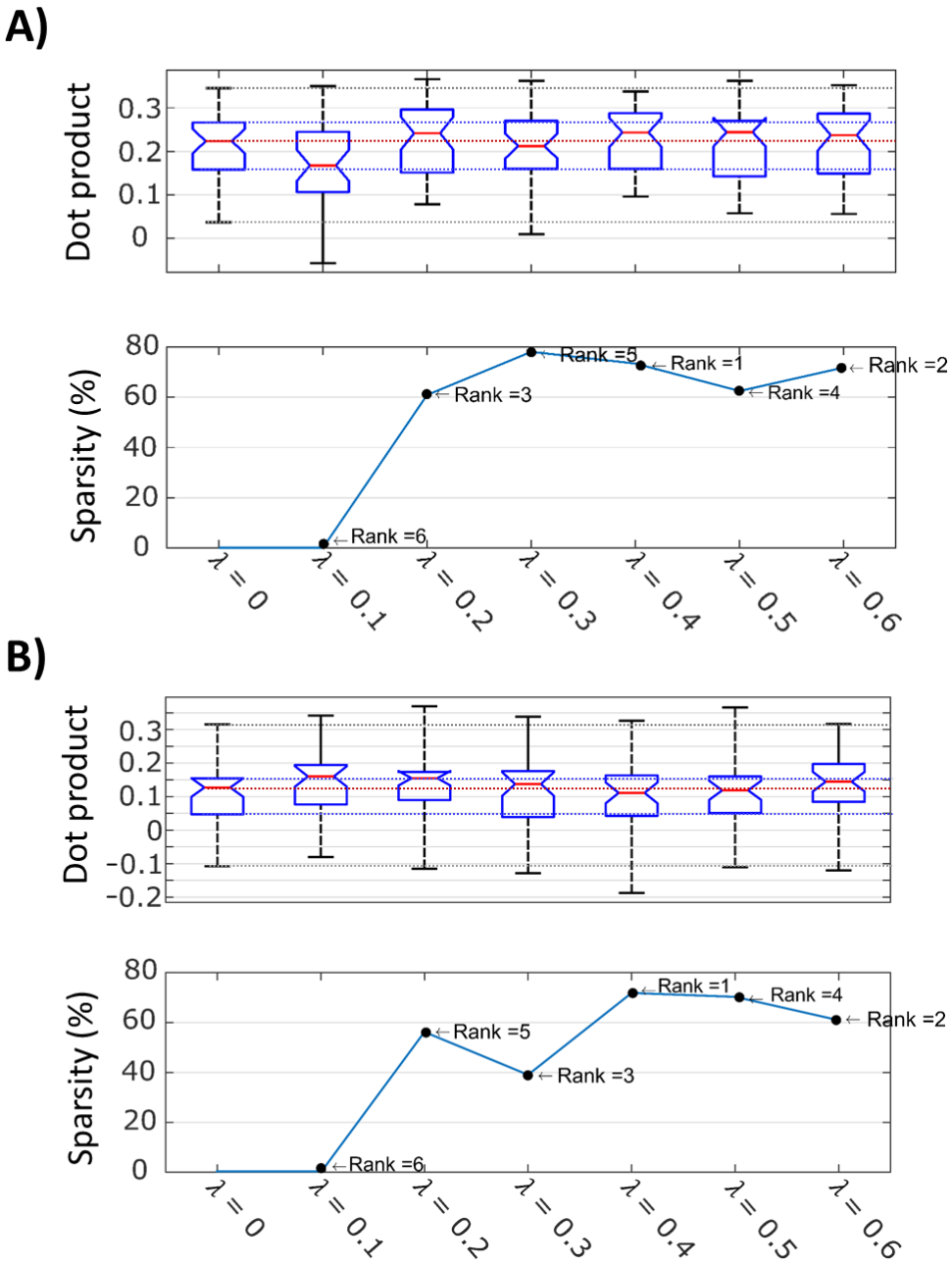



Figure 10-19: APREW-NPLS model performance. The model performance indicators of the AREW-NPLS algorithm for left hand translation movement decoding (A) and right hand translation movement decoding (B). The cosine similarity and the model sparsity were computed. The cosine similarly performance on the 37 test sessions were summarized using a box plot representation where the red line is the median the blue lines indicate the 25th and 75th percentiles ($Q_1$ and $Q_3$). Additionally, the black boundaries show the upper and lower extreme cosine similarity obtained for the experiments. The performance of the REW-NPLS algorithms is presented in the first box plot for left and right hand decoding. The median, $Q_1$ and $Q_3$ of the REW-NPLS models are extended with horizontal dotted lines for easier performance comparison. Additionally, the sparsity of each solution is depicted. The penalized models were ranked during the APREW-NPLS model calibration procedure from the worst to the best model. The estimated rank at the end of the incremental model training is displayed for each model.

### 10.3.3. Example of APREW-NPLS calibration procedure across experiments

The APREW-NPLS algorithm performed the calibration of six penalized models in the same session. To select which models should be re-estimated at each update increment, the Recursive-Validation procedure estimated the rank of each model. The rank was used to evaluate the expected reward of each model. An example of the APREW-NPLS calibration procedure for the left and right hand translation models are shown in the Figure 10-20. More precisely, the dot product and expected reward evolution across update iteration during the pseudo-online incremental model calibration procedure of the left and right hand translation studies are displayed in the Figure 10-20A and Figure 10-20B respectively.

The rank of the models displayed on the Figure 10-19 was extracted from the expected reward of each model at the last update iteration exhibited in the Figure 10-20. The dot product increased with the update iteration whereas the expected reward stabilized for all the models.

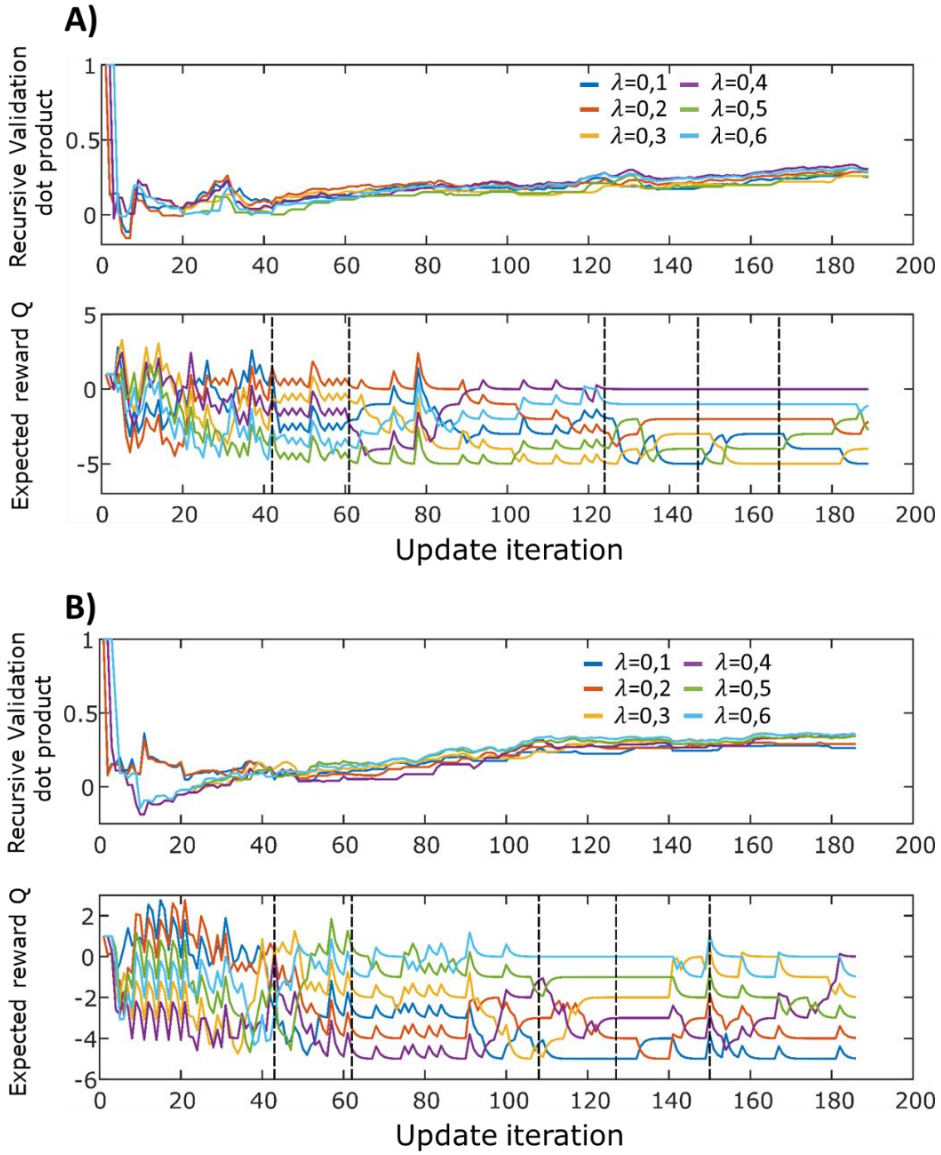



Figure 10-20: APREW-NPLS model calibration. The dot product and expected reward evolution across update iteration during the pseudo-online incremental model calibration procedure of the left (A) and right (B) hand translation studies. The two indicators were computed using the Recursive Validation algorithm. The dot product indicator as well as the expected reward are displayed for the set of possible penalization hyperparameters. The vertical dotted lines stressed the transition between two pseudo-online calibration sessions.

## 10.3.4. AREW-NPLS parameters visualization

The REW-NPLS and the AREW-NPLS model parameter weights are illustrated on the temporal, frequency and spatial domains in the Figure 10-21 for the left hand translation models and in the Figure 10-22 for the right hand translation models.

Similarly than the $L_p$ REW-NPLS model parameter weights, for easier visualization and comparison, the spatial parameter weights of the APREW-NPLS models are displayed

for the left (Figure 10-23) and right (Figure 10-24) hand translation models on a map with the electrode locations relative to the sensory (SS) and motor (MS) sulci.

The model parameter weights estimated with APREW-NPLS algorithms (Figure 10-21 and Figure 10-22) were similar to the parameter weights observed in the PREW-NPLS studies (Figure 10-14 and Figure 10-15). Low frequency band parameters were prevalent for the left hand models whereas both low and high frequency bands are visible for the right hand models.

For the left hand translation spatial parameter weights (Figure 10-23), the electrodes R21, R23, R24, R25, R36 and R47 above the lower part of the MS and the upper part of the SS strongly impact the neural signals decoding similarly to the PREW-NPLS models analyzed previously (Figure 10-16). Similarities between PREW-NPLS (Figure 10-17) and APREW-NPLS (Figure 10-24) models for the right hand translation models were also discernable for the L05, L03, L14, L41, L43 and L50 electrodes parameter weights.

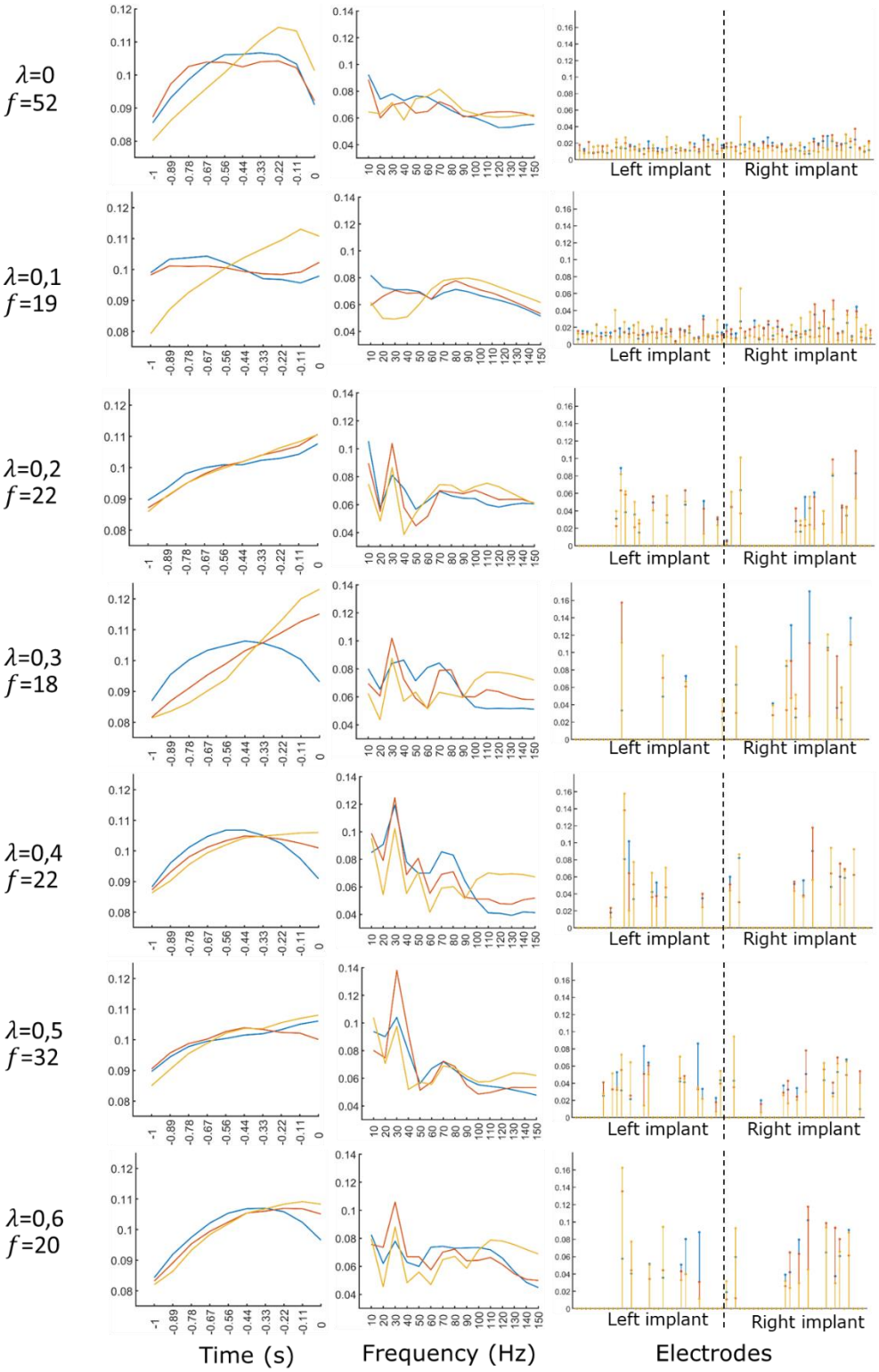



Figure 10-21: Parameter weights of the REW-NPLS and APREW-NPLS models estimated offline in the left hand translation decoding study. Model parameter weights of the tested algorithms for 3D left hand translation decoding from the D dataset according to the spatial, frequency or temporal modalities. The parameter weights related to the $y_1$, $y_2$ and $y_3$ axis are represented using blue, orange and yellow lines respectively.

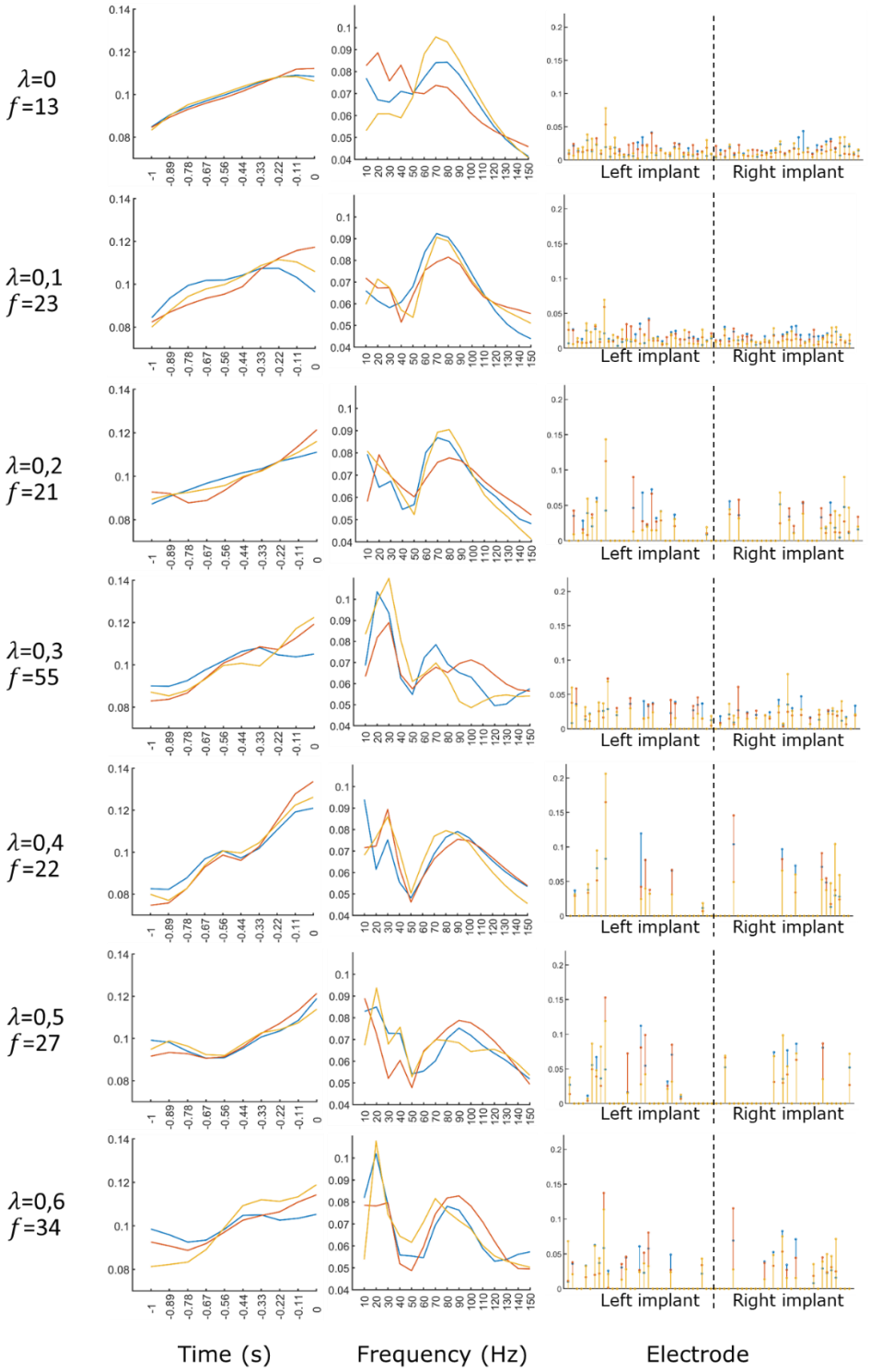



Figure 10-22: Parameter weights of the REW-NPLS and APREW-NPLS models estimated offline in the right hand translation decoding study. Model parameter weights of the tested algorithms for 3D right hand translation decoding from the D dataset according to the spatial, frequency or temporal modalities. The parameter weights related to the $y_1$, $y_2$ and $y_3$ axis are represented using blue, orange and yellow lines respectively.

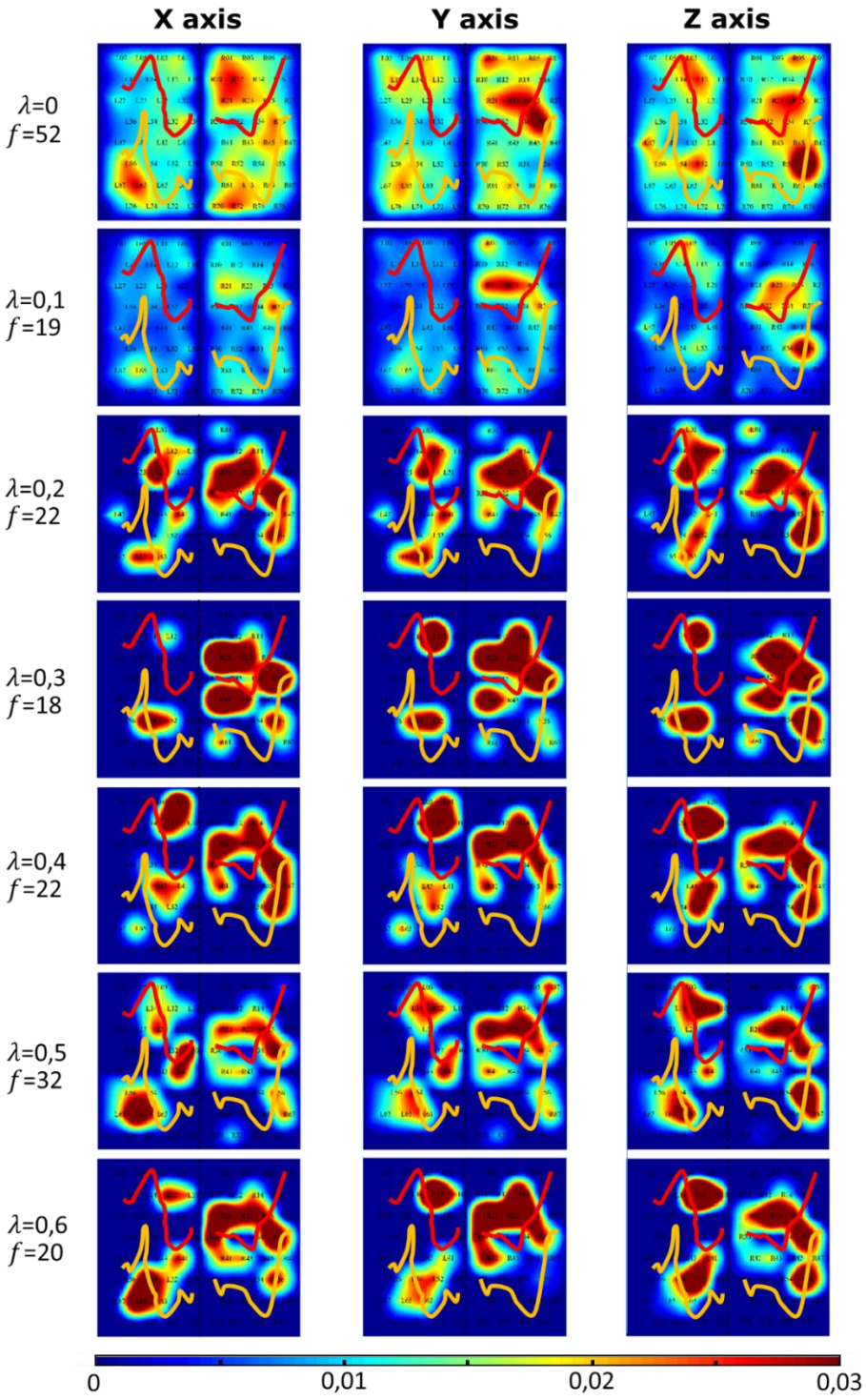



Figure 10-23 : 3D left hand translation decoding parameter weights of the APREW-NPLS models projected on the spatial modality depending on the electrode location on the implant. All the estimated models with different penalization hyperparameter are represented with the optimal latent space dimension estimated during the Recursive-Validation procedure. The sensory sulcus (SS) and motor sulcus (MS) are represented in the spatial domain in yellow and red curves respectively.

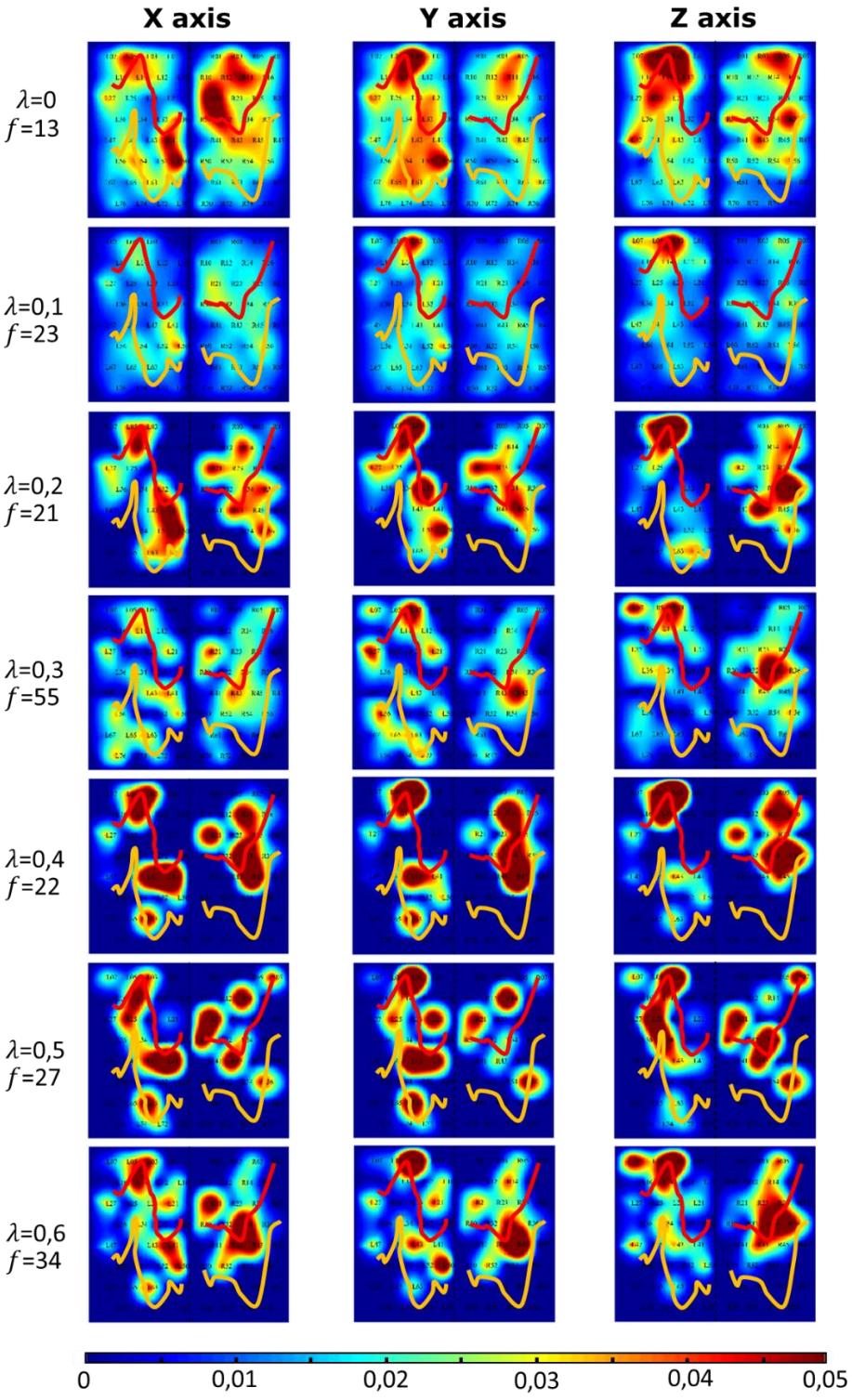



Figure 10-24: 3D right hand translation decoding parameter weights of the APREW-NPLS models projected on the spatial modality represented depending on the electrode location on the implant. All the estimated models with different penalization hyperparameter are represented with the optimal latent space dimension estimated during the Recursive-Validation procedure. The

sensory sulcus (SS) and motor sulcus (MS) are represented in the spatial domain in yellow and red curves respectively.

### 10.3.5. Discussion

Numerous algorithms applied in the BCI field require to tune a hyperparameter. Generally, the hyperparameter is optimized using a validation dataset, a cross-validation procedure, a preliminary study etc. However, these strategies are time consuming, and are commonly performed offline. In the case of online closed-loop experiments, the optimal hyperparameter may be different that the one found during offline studies and may vary across time. In order to go beyond these limitations, a decoder which automatically determines the best penalization hyperparameter was proposed and tested in a preliminary pseudo-online study.

The designed APREW-NPLS trained incrementally multiple models with different penalization hyperparameters all at once. In this pseudo-online study, the number of possible penalization hyperparameter was set to 6 but a larger set of hyperparameter can be selected. The models converged to similar parameter weights and performance than REW-NPLS and PREW-NPLS. Even though some models highlighted higher median cosine similarity than the REW-NPLS algorithm.

The APREW-NPLS algorithm selected sparse models as the optimal solutions for both left and right hand translation studies. Similar decoding performance than the REW-NPLS decoder was highlighted by the APREW-NPLS models with up to 47 and 46 electrodes parameter weights set to zero for the left and right hand translation models ranked 1 by the APREW-NPLS models.

The estimated ranking estimated during the calibration procedure highlighted similarities with the ranking of the decoding performance on the test dataset for the left hand translation study. However, the ranking similarities in the right hand translation study were less evident. Across all the analyzed experiments, left hand translation decoding always highlighted better and more stable decoding performance than the decoding of the right hand translation movements. It is likely that the recorded data for the calibration of the left hand translation models are of better quality than the recorded data for the right hand translation models. Longer calibration periods for the estimation of the right hand translation models may lead to better performance.

As mentioned, APREW-NPLS models converged to similar models than PREW-NPLS but trained multiple models in one calibration procedure. APREW-NPLS decoder highlighted slightly smaller average median cosine similarity on the estimated models than PREW-NPLS models with a 6.3% and 2.6% cosine similarity reduction for the left and right hand translation decoding respectively. However, the highest decoding performance obtained with APREW-NPLS models on the test dataset were similar to the

best decoding performance of the PREW-NPLS models with 3.6% and 2.15% decrease for the left and right hand translation decoding performance.

The presented pseudo-online APREW-NPLS results were limited to a preliminary study. APREW-NPLS algorithm requires deeper investigation. In particular, the PREW-NPLS studies highlighted that models with a $L_1$ norm penalization factor $\lambda > 0.4$ lead to "converged" models with equal parameter weights. Therefore, the APREW-NPLS models with $\lambda = 0.4, 0.5$ and 0.6 should lead to the same models. APREW-NPLS calibration with lower penalization hyperparameter (between 0.1 and 0.4) may lead to better cosine similarity and discrimination between the best and worse models.

Additionally, the model rank estimation procedure used in the APREW-NPLS algorithm to select the next models to update was only based on the model decoding performance. In the case of portable application, a balance between sparsity and model accuracy may be a more suited solution. Moreover, the model ranks tagged in the Figure 10-19 were only estimated on the last expected reward increment. A better ranking algorithm taking into account other model characteristics may lead to a better estimation of the model to select.

Finally, the APREW-NPLS decoder was tested for continuous 3D left and right hand translation decoding. Nevertheless, as the REW-NPLS algorithm, the APREW-NPLS algorithm can be integrated to estimate the gating model used for discrete classification. Therefore, future research will also test the decoding performance and sparsity of the APREW-NPLS algorithm for discrete decoding.

## 10.4. H2M2 gating algorithm offline comparative study

In order to improve the dynamic gating algorithm used in the REW-MSLM experiments, the H2M2 classifier was designed. The H2M2 dynamic decoder was compared to the $HMM_{limited}$ classifier originally integrated in the REW-MSLM algorithm and the $HMM_{full}$ algorithm. The three dynamic classifiers were tested during pseudo-online study with the series F of experiments. In this dataset, the patient controlled the virtual avatar in 4D continuous and $z = 5$ discrete states (IS, $AS_{LH}$, $AS_{RH}$, $AS_{LW}$, $AS_{RW}$). In contrast to the series of experiments D and E, all the state transitions were achieved during the experiments.

### 10.4.1. H2M2 classification performance comparison

The discrete decoding performance of the $HMM_{limited}$, $HMM_{full}$ and H2M2 algorithm are presented in the Figure 10-25 and Figure 10-26. The static indicators are shown in the Figure 10-25 whereas dynamic indicators are displayed in the Figure 10-26.

**For the static indicators** (Figure 10-25), it should be mentioned that the classical F-score and accuracy indicators highlighted high scores for all the models with a Fscore of $91.6\% \pm 1.7$, $92.1\% \pm 1.8$ and $92.4\% \pm 1.9$ and a accuracy of $88.7\% \pm 3.2$, $89.3\% \pm 3.2$

and 89.5% ± 3.2 for the $HMM_{limited}$ and $HMM_{full}$ and H2M2 classifier respectively. With the exception of the precision indicator, they all increased with H2M2 classifier compared to $HMM_{limited}$ and compared to $HMM_{full}$. However the performance variation between $HMM_{full}$ and the H2M2 classifier were small with an average improvements of 1.5% ± 0.83. Compared to the $HMM_{limited}$, H2M2 showed an average improvements of 4.9% ± 2.9. The Gmean performance indicator of the H2M2 algorithm was estimated at 73.5% ± 4.4 whereas $HMM_{limited}$ and $HMM_{full}$ reached a Gmean classification score of 65.6% ± 5.4 and 70.9% ± 4.8. The highest performance improvements were estimated for Recall, HF difference, Bookmaker and Gmean indicators with a classification enhancement of 6.9%, 8.2%, 7.8% and 7.9% compared to $HMM_{limited}$, and 2.4%, 1.5%, 2.7% and 2.6% compared to $HMM_{full}$ respectively.

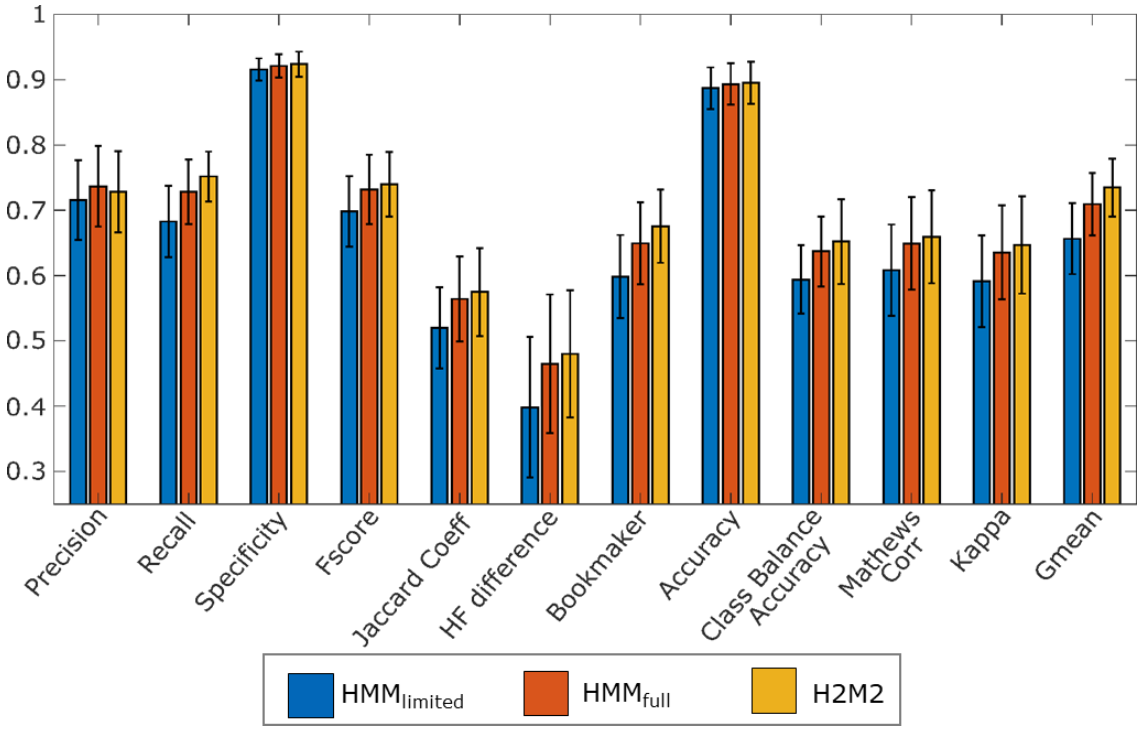



Figure 10-25 : Average static indicator performance for the three tested algorithms: $HMM_{limited}$, $HMM_{full}$ and H2M2 tested on the pseudo online series F of experiments. The standard deviation is represented for each algorithm using a black vertical bar.

**The dynamic indicators** are shown in the Figure 10-26. Firstly, the latency histograms of the three decoders, represented in the Figure 10-26A, were approximated by gamma distributions with the (shape; scale) characteristics of (1.46; 4.47), (1.41; 3.75) and (1.77; 2.24) for the $HMM_{limited}$ and $HMM_{full}$ and H2M2 algorithm respectively. The estimated gamma distributions of the latency indicator for the three dynamic decoders are represented in the Figure 10-26B. The highest values reached for the latency distribution are similar with a latency duration of 2.1 sec, 1.6 sec and 1.8 sec for the $HMM_{limited}$ and $HMM_{full}$ and H2M2 algorithm. However, the distribution shape are quite different.

The estimated median latency of the three decoders are represented in Figure 10-26C. H2M2 showed lower decoding latency than $HMM_{limited}$ and $HMM_{full}$. Using the notation: median $(Q_1 - Q_3)$, the H2M2 latency was computed at $2.6\ (1.9 - 4.9)$ whereas the $HMM_{limited}$ and $HMM_{full}$ algorithms showed a median latency of $4.1\ (3.0 - 7.25)$ and $3.2\ (2.3 - 5.9)$ respectively. The median latency was reduced by 37% and 19% compared to $HMM_{limited}$ and $HMM_{full}$.

H2M2 highlighted an error length block duration of $2.2\ (0.9 - 5.35)$ whereas almost no difference was estimated between the block error length indicators of $HMM_{limited}$ and $HMM_{full}$ with a block error length of $3.1\ (1.0 - 7.45)$ and $2.7\ (1.1 - 7.1)$ respectively. H2M2 highlighted a median error block duration reduction of 29% and 19% compared to $HMM_{limited}$ and $HMM_{full}$ respectively.

Finally, H2M2 showed more frequent errors than the two other algorithms. The error rate was estimated at $2.0\ (1.9 - 2.0)$, $1.8\ (1.7 - 2.3)$ and $2.6\ (2.3 - 2.8)$ for the $HMM_{limited}$ and the $HMM_{full}$ and the H2M2 algorithms.

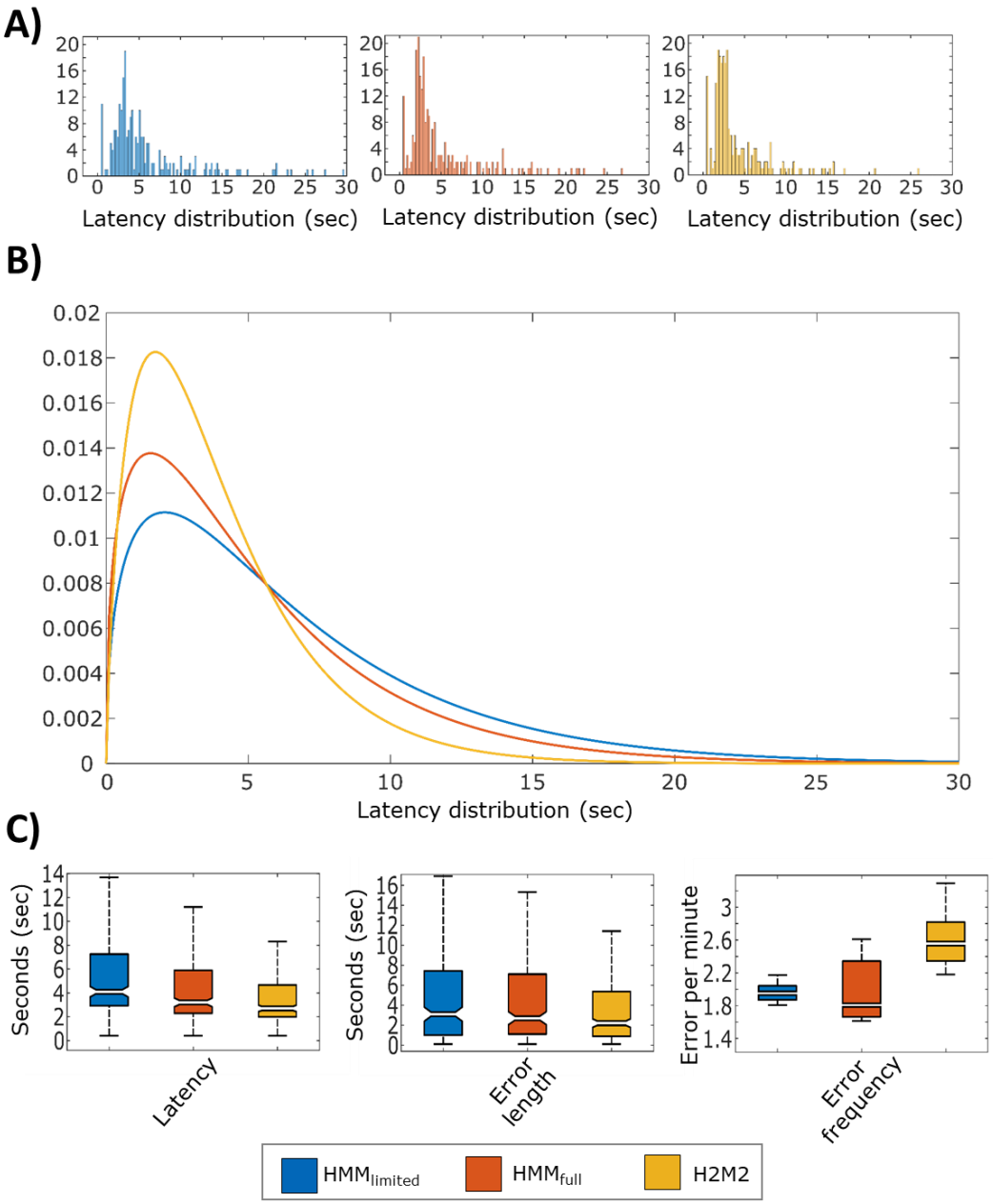



Figure 10-26: Estimated dynamic indicators for the three decoder in the pseudo online series F of experiments. A) Latency histograms of the $HMM_{limited}$, $HMM_{full}$ and H2M2 algorithms. B) Estimated latency distribution of the three algorithms: $HMM_{limited}$, $HMM_{full}$ and H2M2. C) The median, the 25th ($Q_1$) and the 75th ($Q_3$) percentiles of the latency, error length and error frequency indicators. The black vertical bars represent the maximum value not considered as outliers. A sample is considered as outlier if the value of the indicator is more than 1.5 times the interquartile range from the 25th or the 75th percentile. All the indicators were estimated using all the recorded data (including outliers). Significance of the differences between the three decoders are computed for the three decoders using the Student t-test with a Bonferroni correction ($\alpha_{multi-class} = 0.0167$). Significant values are indicated by an asterisk.

## 10.4.2. H2M2 parameter weights visualization

H2M2 splits the classification problem into several sub-problems estimated by sub-models. This section presents the H2M2 parameter weights projected on the frequency, temporal and spatial modalities (Figure 10-27) compared to the parameter weights of the $HMM_{limited}$ or $HMM_{full}$ (Figure 10-28).

As previously mentioned, a HMM model is defined by an initial state probability $\boldsymbol{\pi}$, an emission model $\{\underline{\mathbf{B}}, \mathbf{b}\}$ and a transition matrix $\mathbf{A}$. $HMM_{limited}$ and $HMM_{full}$ models have the same initial state probability $\boldsymbol{\pi}$ and emission model $\{\underline{\mathbf{B}}, \mathbf{b}\}$. However, they are defined with different transition matrices $\mathbf{A}_{limited}$ and $\mathbf{A}_{full}$. Therefore, the parameter weights represented in the Figure 10-28 are the parameter weights of the $HMM_{limited}$ and the $HMM_{full}$ algorithms: $\underline{\mathbf{B}} = \underline{\mathbf{B}}_{limited} = \underline{\mathbf{B}}_{full}$.

The projection of the H2M2 in the first line of Figure 10-27 showed the model estimated for IS state and the two internal states: the left body side movement state and the right body side movement state noted $AS_L$ and $AS_R$ respectively. Frequency bands around 10Hz and more particularly around 20Hz exhibited great influence on the predictions of the two internal states $AS_L$ and $AS_R$. IS state was not as clear as active states but showed important influence of the 10Hz frequency band. Similarly to the HMM models, H2M2 exhibited large value for the parameter weights related to high frequency bands above 130Hz. The parameter weights projected on the temporal modalities highlighted the typical curved shape that was found with the online closed-loop 8D REW-MSLM gating model (Figure 10-6).

The projection of the H2M2 in the second and third line of Figure 10-27 show the models estimated to discriminate the hand and wrist states from the same body side. The projection of the frequency modality exhibited large parameter weights below 20Hz, between 50Hz and 90Hz and above 130Hz. The H2M2 frequency parameter weights were similar to the HMM parameters.

For ease of visualization and interpretation, the H2M2 and HMM parameter weights projected on the spatial modalities are presented in the Figure 10-29A and Figure 10-29B respectively with the location of the electrodes compared to MS and SS. For H2M2, whereas the parameter weights discriminating the $AS_L$ and $AS_R$ seems more diffused on all the electrodes with a small superiority of contralateral electrodes, the models classifying the binary problem hand versus wrist are more focused on specific areas. For the left hand versus wrist model, the higher parameter weights were located on the electrodes R14, R16, R25 and R32 around the MS. For the right hand versus wrist model, the electrodes L03, L05, L12, L14 and L21 showed stronger parameter weights.

The H2M2 and HMM parameter weights were likely to be similar especially the model related to right hand versus wrist model of H2M2 compared to the $AS_{RH}$ and $AS_{RW}$

models estimated by HMM. For both models, majority of the most impactful electrodes are located above the MS and the upper part of the SS.

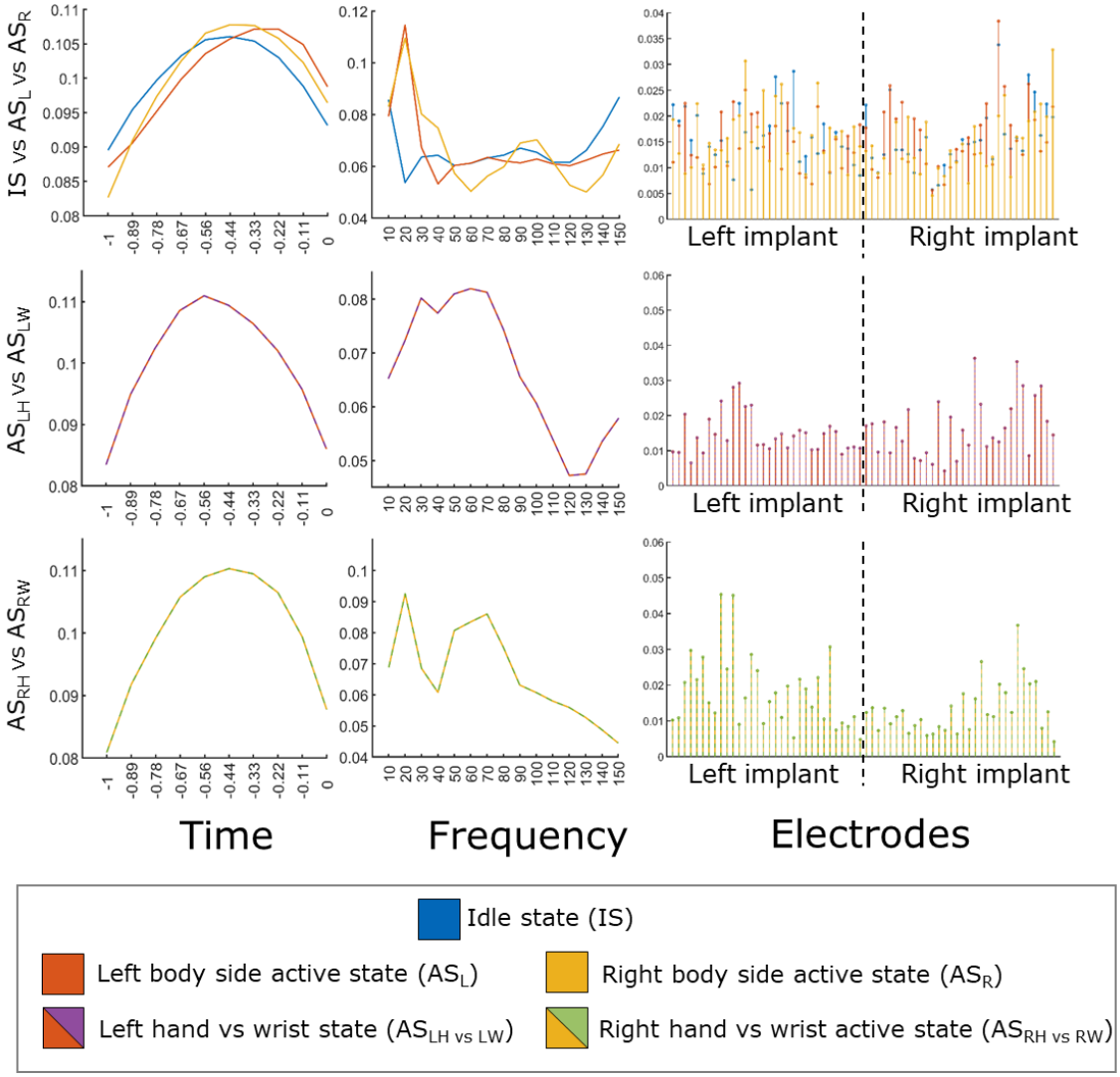



Figure 10-27: Parameter weights of the H2M2 gating model estimated offline with the series F of experiments. Model parameter weights of the H2M2 algorithms for the decoding of five states according to the spatial, frequency or temporal modalities. The first line represents the sub-model parameter weights discriminating the Idle state (IS), the left body side movement state referred as ($AS_L$) and the right body side movement state named ($AS_R$). The second line highlights the parameters weights of the sub-model which classifies the states between left hand movements ($AS_{LH}$) and left wrist rotation state ($AS_{LW}$). The last sub-model is acting similarly than the second one with the right hand movements ($AS_{RH}$) and right wrist rotation state ($AS_{RW}$).

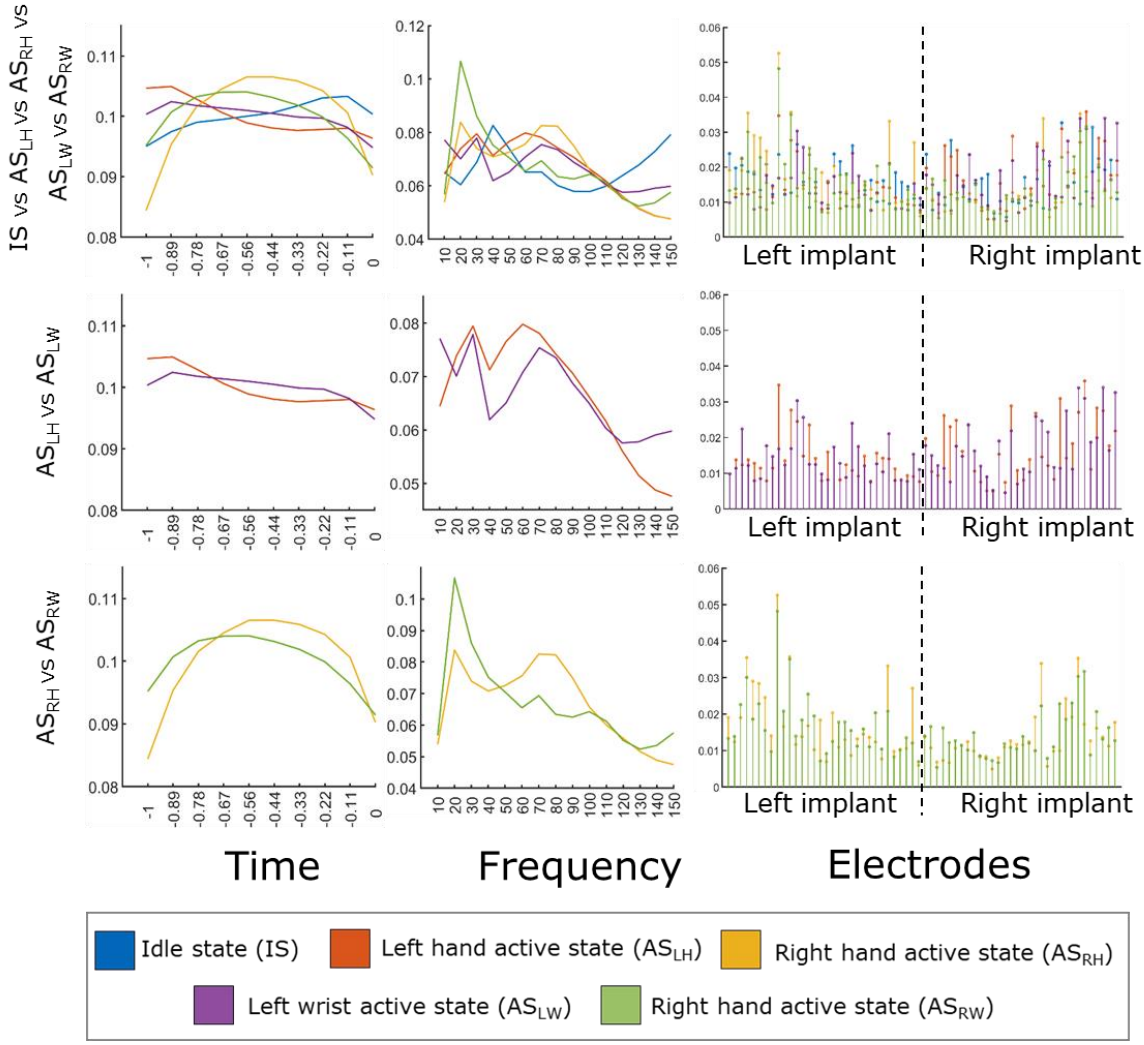



Figure 10-28 : Gating parameter weights of the $HMM_{limited}$ and $HMM_{full}$ according to the spatial, frequency or temporal modalities for each state: rest state (IS), left hand 3D translation and rotation states ($AS_{LH}$ and $AS_{LW}$) or right hand 3D translation and rotation states ($AS_{RH}$ and $AS_{RW}$). The only difference between both models is the transition matrix. The parameter weights $AS_{LH}$, $AS_{LW}$ and $AS_{RH}$, $AS_{RW}$ are represented on the second and third lines respectively for easier comparison with the H2M2 parameter weights.

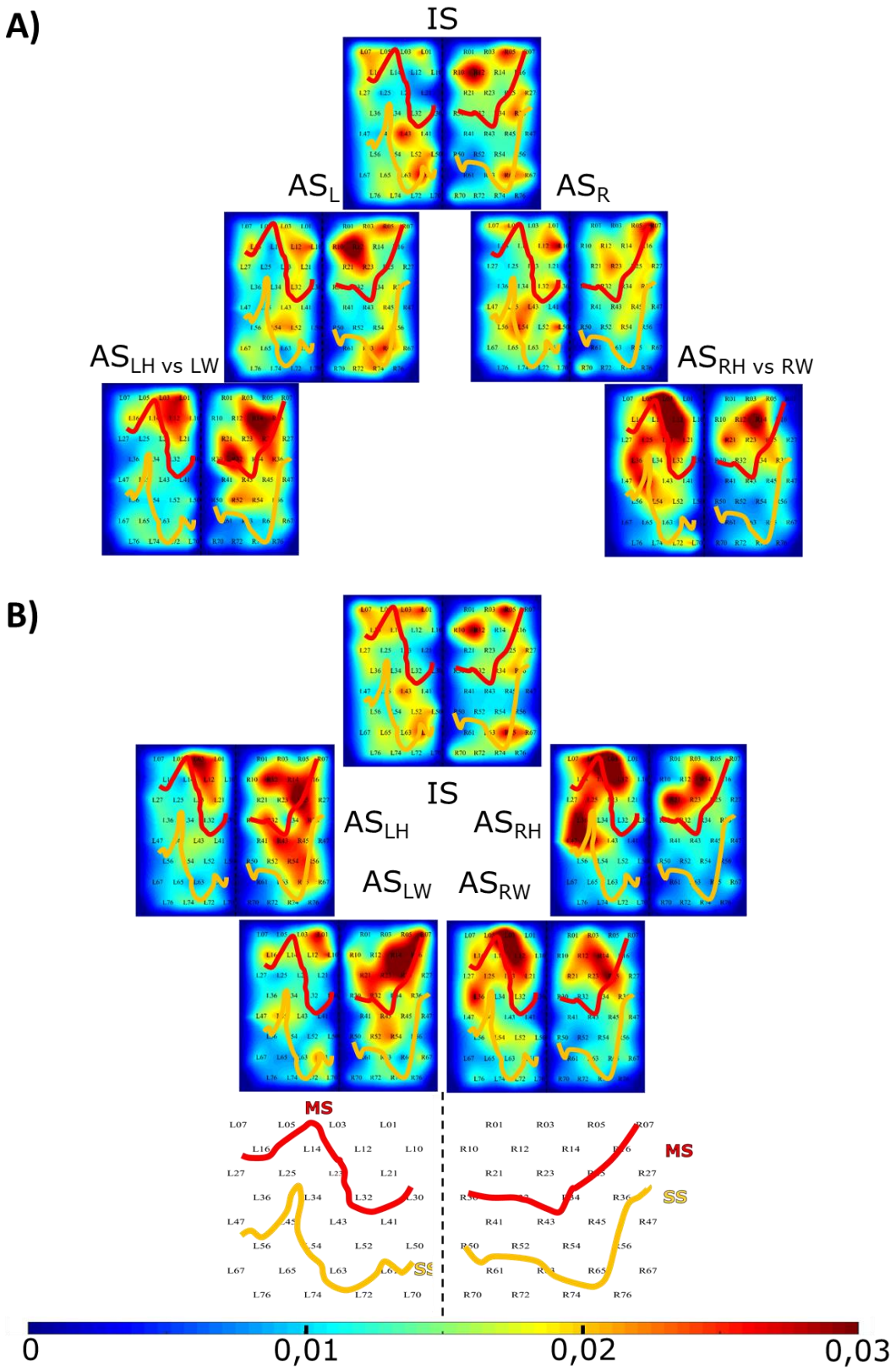



Figure 10-29 : Parameter weights of the H2M2 (A) and HMM$_{limited}$ orHMM$_{full}$ (B) estimated models following the spatial modality. The sensory sulcus (SS) and motor sulcus (MS) are represented in the spatial domain in yellow and red curves respectively.

### 10.4.3. Discussion

The aim of the H2M2 was to exploit the hierarchical structure of the proposed control tasks to improve the gating classifier. The pseudo-online study using the 4D continuous and $z = 5$ discrete states experiments highlighted better static and dynamic performance indicator using the H2M2 decoder.

Highest static decoder improvements were stressed on the Recall-based indicators (HF Difference, Bookmaker and Gmean indicators). From these results, it can be concluded that H2M2 algorithm enhanced the true positive ($t_p$)-false negative ($f_n$) ratio. In the same time, the precision and specificity indicators remained constant highlighting stable true positive ($t_p$)-false positive ($f_p$) and true negative ($t_n$)-false positive ($f_p$) ratio. H2M2 presented lower transition latency between states and shorter error blocks but more frequent error blocks.

The H2M2 decoding performance were evaluated in a preliminary pseudo-online study with small training and testing dataset. New online closed-loop experiments integrating directly H2M2 algorithm as gating classifier of the REW-MSLM mixture of expert algorithm must be performed to confirm the results stressed during this pseudo-online study.

The incremental adaptive H2M2 classifier may be a relevant solution to improve the responsiveness of the REW-MSLM. Nevertheless, it is important to notice that in our application case, the H2M2 trained three classifiers instead of one which increase the computational loadings. H2M2 algorithm may be more relevant in the case of more complicated hierarchical state classification task with higher number of mental state to discriminate. A trade-off between the responsiveness, the block error rate and the computational loading must be considered for the integration of H2M2 in the REW-MSLM depending on the BCI application and the control task to realize.

## 10.5. Conclusion

The Recursive Exponentially Weighted Markov Switching multi-Linear Model was designed to control multi-limb effectors during online closed-loop experiments using an incremental CLDA procedure. The decoding performance of REW-MSLM were firstly evaluated during a pseudo-online study where the gating benefits and the interest of cross-session training were stressed. Finally, the REW-MSLM was integrated in the online clinical BCI and tested during online closed-loop 8D experiments with 3D left and right hand translation and 1D left and right wrist rotation tasks. The REW-MSLM decoder highlighted good performance and stability across time with a good decoding performance without any model recalibration during 6 months.

The pseudo-online studies of the incremental adaptive PREW-NPLS and APREW-NPLS algorithms highlighted the benefits of group-wised penalized solutions promoting the

sparsity in the case of small calibration dataset. PREW-NPLS decoders highlighted at least similar decoding performance than REW-NPLS algorithm with decreasing dimension of the features in the spatial modality. APREW-NPLS was designed to automatically set the penalization hyperparameter during the online closed-loop experiments. APREW-NPLS showed promising results but require deeper test, training and investigations.

H2M2 was designed as new adaptive gating model for the REW-MSLM algorithm. In a pseudo-online study, the H2M2 algorithm presented a better responsiveness and shorter error blocks than HMM models at the cost of higher error block frequency. A trade-off between false positive detection and decoding latency must be considered depending on the BCI application.

PREW-NPLS, APREW-NPLS and H2M2 are supplementary algorithms which were designed to be easily integrated into the REW-MSLM algorithm in order to adapt the mixture of expert decoder depending on the BCI application. All the algorithms were evaluated during pseudo-online experiments. While the offline studies were mandatories in order to evaluate the performance of the algorithms, the results cannot be fully generalized to the online closed-loop experiments due to the lack of patient's feedback. Therefore new online closed-loop experiments integrating the new decoders into the REW-MSLM algorithm must be carried out.

Chapter 11

# Discussion
# Limitations
# Perspectives

## Contents

## 11.1. Discussion

The clinical trial of Clinatec provided the proof of concept that a tetraplegic patient can control a four-limb neuroprosthetic exoskeleton through ECoG neural signal decoding. The current Ph.D. research focused on several technical challenges namely, online decoding, direct neural decoding control strategy from neural population recordings, asynchronous multi-limb decoding, complex pursuit task completion and closed-loop decoder adaptation.

In order to control an effector, the BCI decoder must be sufficiently optimized and efficient to decode neural signals in real-time. However, most of the algorithms tested in offline studies do not take into consideration the computational requirements of the proposed solutions and generally required high computational loading to estimate a decoding model and to apply it.

In order to make more transparent control of an effector performing complex tasks, the direct neural decoding control strategy seemed more adapted than the motor imagery (MI) control strategy. Somatotopic remapping control strategy may lead to easier neural signal decoding, nevertheless, EEG studies reported that 10 to 30% of the users were unable to control MI-BCIs [Jeunet et al., 2016]. The control of complex effectors required numerous different MI strategies which are by definition in limited number. Moreover, MI may create a high mental load to the user. Besides, direct neural decoding is mandatory for other applications such as rehabilitation applications.

Additionally, in order to translate the BCIs from the laboratory to real-life applications, common challenges to overcome were reported, namely, the high-dimensional control of effectors, experiments closer to real life behavior, and the ability of the asynchronous BCI system to act as a stand-alone device. Daily life actions often require multi-limb and/or more complex actions than the one tested during clinical trial experiments. Generally, BCI performance are evaluated through center-out tasks [Bundy et al., 2016] [Degenhart et al., 2018] [Hochberg et al., 2006] [McFarland et al., 2010] [W. Wang et al., 2013] [Young et al., 2019]. However, these tasks are simple and do not look like the complex everyday life actions that should be performed by the patient. Pursuit tasks were used for decoding performance evaluation as they seem more similar to the daily-life patient-environment interactions than center-out tasks. Patients must be able to control freely an effector to perform several tasks from the same decoder and have strong idle state control.

Finally, the decoding performance of the patient should be optimized and not degrade across time. This challenge is particularly difficult knowing the brain signal variability across time. Moreover, during online closed-loop experiments, neural signals are highly dependent on the sensory feedback provided to the patient through the control of the effector. Therefore, the patient should be integrated into the model calibration procedure (“human-in-loop”) in order to improve the decoding performance. A strategy which already highlighted good results for stable and accurate neural signal decoding is the adaptation of the decoder during closed-loop experiments. Closed-loop decoder adaptation (CLDA) leads to different model parameter convergence, better performance compared to decoders trained offline during open-loop

experiments [Lebedev and Nicolelis, 2017, 2006] [Murphy et al., 2016] [Orsborn et al., 2014], easier/faster training procedure [Brandman et al., 2018] and allows the model adaptation to the variations of the neural signals across time.

Based on the first successful long-term (more than 36 months) chronic exploitation of bilateral epidural ECoG recordings in a tetraplegic individual, new BCI decoders were designed. The Recursive Exponentially Weighted Markov Switching Model (REW-MSLM), the Penalized Recursive Exponentially Weighted n-way Partial Least Square (PREW-NPLS), the Automatic $\lambda$ PREW-NPLS (APREW-NPLS) and the Hierarchical structured Hidden Markov Model (H2M2) are incremental adaptive decoders created in order to control in real-time an asynchronous multi-limb effector.

The REW-MSLM is a piecewise linear model based on a mixture of experts (ME) architecture composed of several continuous "expert" models decoding continuous movements from the neural signals and a dynamic "gating" model activating or inhibiting the expert continuous outputs. The REW-MSLM algorithm is an adaptive mixture of experts algorithm where every expert and gate models are independently and incrementally updated to perform an online closed-loop decoder adaptation and allow the decoder and the patient to learn from each other (human-in-loop calibration).

The REW-MLSM has a specific architecture, each expert calibration procedure is independent. This highly flexible REW-MSLM structure allowed to perform the expert parameter weight estimation with different algorithms for each expert in order to fit the best model to each task. Therefore, new algorithms were designed in order to be integrated into the REW-MSLM algorithm and provide new properties to the gate and expert models of REW-MSLM.

PREW-NPLS and APREW-NPLS are two adaptive group-wise sparse decoders designed to reduce the feature space dimension, improve the model interpretability, create low computational cost models suited for portable applications and be integrated in the REW-MSLM algorithms as sparse gating and/or expert models. The $L_p$-PREW-NPLS is a new regularized recursive exponentially weighted N-way PLS designed for online adaptive decoding promoting group-wise (slice-wise) sparsity generalized to $L_0$, $L_{0,5}$ and $L_1$ norm/pseudo-norm regularization. $L_p$-PREW-NPLS group-wise sparse regularization was proposed to prevent overfitting, to improve the decoding performance and to simplify the model interpretation compared to the REW-NPLS algorithms. The Automatic PREW-NPLS (APREW-NPLS) was designed in order to estimate the penalization hyperparameter during the incremental online calibration. APREW-NPLS overcame the drawbacks of PREW-NPLS which required to determine the best penalization hyperparameter during offline preliminary studies before its use into online closed-loop experiments.

Finally, H2M2 dynamic classifier was designed to create a gating model with a high decoding responsiveness and low latency state transitions.

### 11.1.1. Offline evaluation of the developed algorithms

Before to integrate the new decoders into the ABSD software chain of the clinical trial to perform online adaptive multi-limb experiments, the developed algorithms were evaluated in several pseudo-online comparative studies.

#### REW-MSLM offline evaluation

In the REW-MSLM offline pseudo-online study, the interests of adding a dynamic gating model in order to inhibit or enhance the output variables of the continuous decoders were demonstrated. Continuous REW-NPLS decoder was not able to perform non-zero velocity predictions as numerous regression algorithms. The integration of a gating model allowed avoiding the non-intended movements from the non-controlled limb which should significantly reduce the stress and mental task complexity of the subject. Moreover, the significant result variations between static and dynamic gating in term of pure decoding performance (accuracy and F-score) and in term of misclassified sample distribution (the frequency rate of the error blocks) attested the needs to integrate a dynamic classifier into the decoder in order to perform asynchronous multi-limb experiments.

In the REW-MSLM offline study, the results induced the benefits of cross-session training in order to obtain a better decoder more robust to the variation of the neural signals and the experimental condition variability. Indeed, the results from dataset C highlighted stable performance whereas the model was trained based on cross-session calibration procedure from dataset B recorded 9 to 28 days before.

Previous ECoG-based direct neural decoding state-of-the-art studies highlighted a correlation between the neural signals and the cursor velocity of $0.48 \pm 0.09$ [Schalk et al., 2007], and $0.41 \pm 0.14$ [Bundy et al., 2016]. Nevertheless, these results are not fully comparable to the present study. Firstly, the reported experiments were based on subdural ECoG recordings [Bundy et al., 2016] [Schalk et al., 2007] while the presented Ph.D. study was based on epidural ECoG. Epidural ECoG reduces invasiveness and the potential impact of surgical site infection. However, a significant decrease in the decoding performance was highlighted in experiments with non-human primates (NHP) [Farrokhi and Erfanian, 2018] [Schaeffer and Aksenova, 2016a] using epidural ECoG recordings compared to subdural ECoG. Finally, state-of-the-art experiments were based on able-bodied epileptic patients performing 2D/3D center-out experiments using actual arm movements [Bundy et al., 2016] or 2D circular movements based on actual joystick control [Schalk et al., 2007]. The recordings were processed offline. In the present study, a tetraplegic patient performed online control of a real effector to achieve alternative multiple point-to-point pursuit tasks. Pursuit tasks explore the entire 3D space and are harder than generic center-out task.

The benefits of the REW-MSLM induced during the pseudo-online studies were confirmed during the online-adaptive closed-loop experiments using the online incremental adaptive REW-MSLM decoder.

**PREW-NPLS offline evaluation**

The PREW-NPLS was tested offline in a pseudo-online study using dataset recorded during the online closed-loop 8D experiments performed with the REW-MSLM algorithms. In order to stick to the online closed-loop experiments restrictions, the pseudo-online Lp-PREW-NPLS calibration dataset was restricted to the same 6 calibration experiments which were used for the online REW-MSLM experts and gating model estimations. Lp-PREW-NPLS performance were only estimated for the 3D left and right hand translation tasks as the REW-NPLS expert of REW-MSLM already highlighted good decoding performance for the left and right wrist rotation tasks. The model was evaluated with $L_0$, $L_{0,5}$ and $L_1$ norm regularization and multiple penalization hyperparameters in order to estimate the impact of the sparsity on the decoding performance. In the presented pseudo-online study, the group-wise regularization was focused on the spatial modality.

The PREW-NPLS decoders highlighted equivalent or better decoding performance than the REW-NPLS algorithm for the majority of the penalized hyperparameters. As an example, for the right hand decoding study, the $L_1$-PREW-NPLS model with a penalization hyperparameter of $\lambda = 0.26$, highlighted a significant ($p - value = 10^{-6}$) cosine similarity improvement of $21\%, 116\%$ and $24\%$ of the median, $25^{th}$ and $75^{th}$ percentiles respectively with less than half of the electrodes maintained to a non-zero value (33 electrodes set to zero). In the case of small penalization hyperparameters such as $\lambda = 0.1$, the model converged to a non-sparse solution with significant cosine similarity differences between REW-NPLS and $L_1$-PREW-NPLS models ($p - value = 10^{-6}$) leading to a median, $25^{th}$ and $75^{th}$ percentile enhancement of $24\%, 104\%$ and $23\%$ respectively. The sparsest solution with the $L_1$-PREW-NPLS algorithm removed $75\%$ of the electrodes without decreasing the cosine similarity indicator reducing the features space from $10 \times 15 \times 64 = 9600$ features to $10 \times 15 \times 16 = 2400$. If the sparse models with a limited number of electrodes turns out to be stable, the feature space dimension can be definitively reduced.

Significant indicator differences between REW-NPLS and $L_1$-PREW-NPLS models were more evident for the 3D right hand translation models than for the 3D left hand translation ones. In all the manuscript, the 3D right hand translation models highlighted lower decoding performance than the left hand translation models. Group-wise sparse regularization may be a solution to enhance the neural signal decoding for 3D right hand translation tasks. Although not all the $L_1$-PREW-NPLS models highlighted better cosine similarity than the REW-NPLS model, for most of the left and right $L_1$-PREW-NPLS models, similar results were obtained with only a small number of spatial features compared to REW-NPLS.

Finally, reducing the feature space dimension led to more interpretable models. Several activation patterns were discernable. In the spatial domain, the electrodes closed to the motor and sensory sulci exhibit important parameter weights whereas, in the frequency domains, frequency bands below 30Hz and between 60Hz and 110Hz were dominant.

The PREW-NPLS offline pseudo-online study exhibited the potential benefits of penalized models to decode the neural signals using sparse solutions.

### APREW-NPLS offline evaluation

To evaluate the $L_p$-PREW-NPLS model performance compared to REW-NPLS algorithm, a grid search was performed to create lots of penalized models using three penalization norms/pseudo-norm ($L_0$, $L_{0.5}$ and $L_1$) and increasing penalization hyperparameter $\lambda$. This $L_p$-PREW-NPLS offline grid analysis results in a total of 558 pseudo-online calibration procedures to achieve for each 3D hand translation study (left and right). This study was highly time consuming and required high computational resources. Moreover, there is no evidence demonstrating that the optimal hyperparameter extracted from the offline pseudo-online study of PREW-NPLS does not change over time and is not influenced by the patient's feedback during the online closed-loop experiments.

To go beyond the presented limitations, the Automatic PREW-NPLS (APREW-NPLS) algorithm was designed. The APREW-NPLS algorithm tests online a set of different PREW-NPLS model configurations (with different hyperparameters) and incrementally updates the models that are most likely to be the best decoding models. With this competition procedure, models with poor decoding performance are not often updated whereas decoders with good performance are put forward. To evaluate the behavior and decoding performance of the APREW-NPLS, a preliminary study was carried out. APREW-NPLS was tested offline for the neural signal decoding of the 3D left and right hand translation task with a set of six possible hyperparameter values. All the six models were trained in the same time on the six training dataset recorded from the online closed-loop calibration procedure achieved for the REW-MSLM estimation. APREW-NPLS models showed similar decoding performance than the original REW-NPLS algorithm with a sparsity index up to 78.13% and 71.88% for the left and right hand models respectively. Additionally, the APREW-NPLS models converged to sparse solutions with parameter weights similar to the one estimated in the offline PREW-NPLS study. The optimization of the model hyperparameters during the online experiments is a powerful tool which can be applied in various BCI applications with numerous different hyperparameters.

The presented APREW-NPLS performance evaluation relies on a preliminary study which requires deeper investigation on, the set of tested penalization hyperparameter, the model ranking algorithm, the calibration procedure, etc. Nevertheless, it is likely that a larger number of hyperparameters to tune may require larger training dataset. A trade-off between the number of penalization hyperparameters and the length of the training dataset must be reached.

#### 11.1.1.1. H2M2 offline evaluation

Finally, in order to integrate a low latency state transition gating model in REW-MSLM, a new dynamic decoder, named H2M2, inspired by HHMM, was proposed. This algorithm breaks down the classification problem into sub-problems with one classifier dedicated to each sub-

problem. A preliminary pseudo-online study based on online closed-loop 4D continuous movements and 5 discrete state virtual avatar experiments were carried out. H2M2 highlighted better responsiveness with a lower latency between the task instruction and the decoding than the HMM gating model. However, the block error rate increased. The H2M2 gating is a model to consider in the case of complex classification problems where the system responsiveness is a high priority characteristic. The benefits of H2M2 algorithm integrated in REW-MSLM as a gating model is highly dependent on the classification problem requirements. A trade-off between the responsiveness, the block error rate and the computational loading must be considered.

### 11.1.2. Real-time closed-loop BCI experiments

The REW-MSLM decoder highlighted promising results in multiple offline pseudo-online studies. Therefore, the REW-MSLM decoder was integrated into the online neural signal decoding platform ABSD to control complex virtual or real effectors.

During the clinical trial, different versions of the REW-MSLM algorithm were integrated on the online closed-loop BCI platform ABSD. The evolution of the REW-MSLM was linked to the patient's skills improvements and difficulties of the proposed experiments. Before the first REW-MLSLM version, the REW-NPLS algorithm was used for online closed-loop BCI experiments. During this period, experiments from switching control (one discrete state controlled) to alternative 2D two-hand translation control (4D continuous dimensions and 3 discrete states) were attempted. During these experiments, a strong idle state and independent control of limbs were not achieved. The first integration of REW-MSLM was performed in mid-February 2018 using a dynamic gating model and one expert. This first model achieved good state discrimination and was tested during 6D alternative left and right hand translation (3 discrete states) experiments. The upgraded version of REW-MSLM with the dynamic gating and two experts dedicated to left and right body side decoding was integrated 3 months later. This REW-MSLM version was used to perform the online closed-loop 8D experiments with 5 discrete states presented in this manuscript. Finally, a generalized version of the REW-MSLM using a dynamic gating and one expert for each task (left hand translation, left wrist rotation, right hand translation, etc.) was integrated into the clinical trial BCI platform. Experiments clustering 8D continuous decoding and 6 to 8 discrete state classification tasks were trained. The upgrade milestone of the REW-MSLM algorithm integrated in the BCI platform as well as the evolution of the proposed experimental paradigm are shown in Figure 11-1.

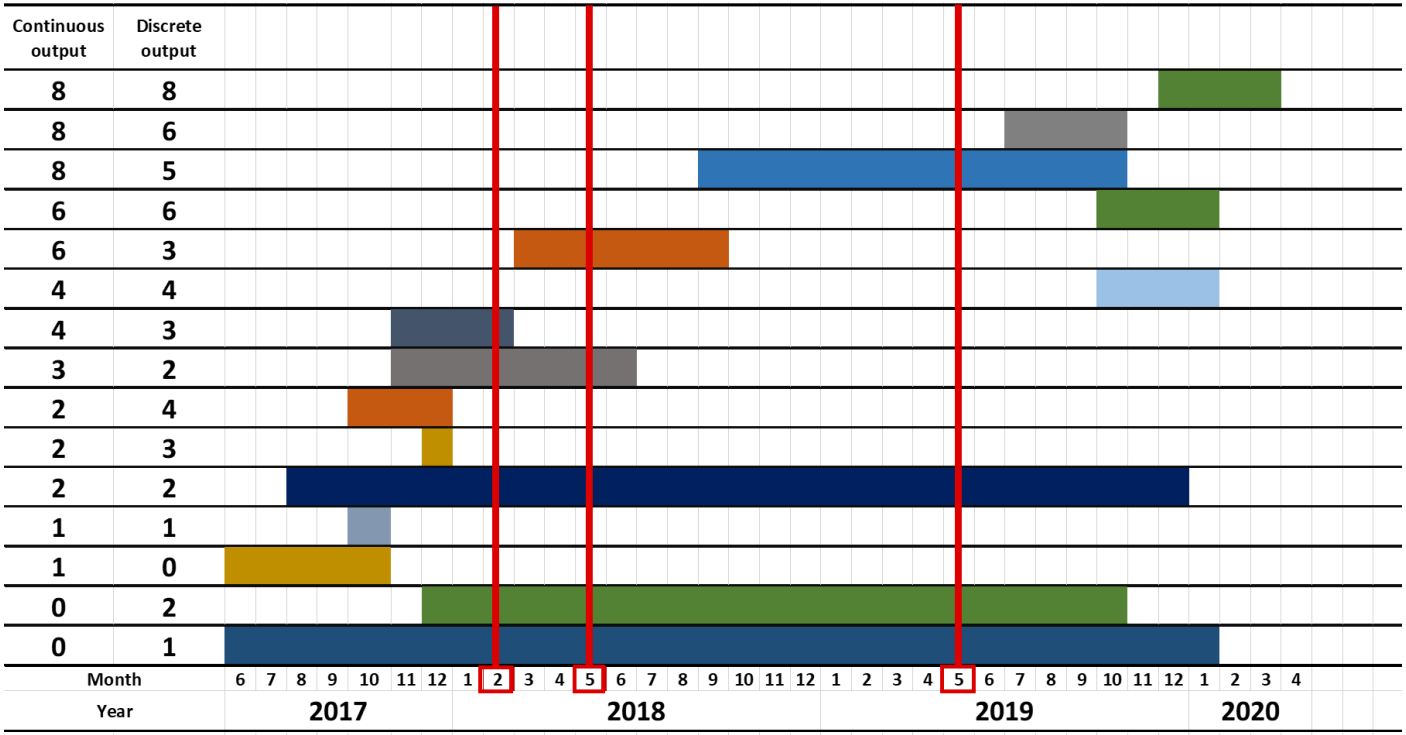



Figure 11-1: Evolution of the DoF controlled by the patient across the Clinical trial timeline during the online closed-loop experiments. During the clinical trial, more and more complicated tasks and experiments were proposed to the patient depending on the experimental results and the ease of the patient to control the effectors. The red lines represented the modification of the decoder integrated into the online adaptive ABSD platform for online closed-loop experiments. Each experiment paradigm is presented with a colored area showing the period in which the experiments were achieved. The discrete dimension correspond to the number of states $K$ which are discriminated by the REW-MSLM gating model whereas the continuous dimension indicator show the sum of independent continuous dimensions controlled during the alternative tasks. For example, an asynchronous experiment characterized by alternative 3D left hand translation, 3D right hand translation, 1D left wrist rotation, 1D right wrist rotations tasks is displayed on the figure in the line corresponding to $K = 5$ states (the four active states and the idle state) and $3 + 3 + 1 + 1 = 8$ continuous ouputs

Using the REW-MSLM algorithm, a tetraplegic patient performed the online high-dimensional control of an exoskeleton and a virtual avatar. The patient achieved 8D continuous control including alternative 3D hand reaching tasks and 1D wrist rotation for each hand with 5 discrete states: idle state, left and right hands translation and left and right wrists rotation. The discrete and continuous decoding performance highlighted stable results over 6 months of clinical experiments after the last model recalibration for both effectors.

During the online closed-loop 8D experiments, the REW-MSLM 8D models were trained for each effector based on cross-session calibration procedure during 6 experiments for approximately 3.5 h (with an average of 195 training trials per task). These models training periods and the global number of trials seem moderate considering the number of dimensions to control and the performance obtained compared to those in similar studies [Degenhart et al., 2018] [W. Wang et al., 2013]. More training data may lead to a more generalized model and thus, better results.

The online alternative multi-limb 8D pursuit tasks proposed to the patient to control a complex effector are more complicated tasks than the usual state-of-the-art. Generally, 3D arm control is evaluated based on classical center-out experiments [Bundy et al., 2016] [Degenhart et al., 2018] [W. Wang et al., 2013] [Young et al., 2019]. Center-out tasks request to go from the center of a workspace to one of the targets localized at equal distance from the initial hand location. During center-out tasks, after each trial (succeeded or failed), the hand position is reset to the initial position after few seconds of rest. In the alternative multi-limb point-to-point pursuit task experiments proposed in the CLINATEC clinical trial, the patient controlled the effector all along the session and no takeover was provided by the system to reset the hand position or to propose a rest period to the patient. Consequently, the control task is more complex because the initial position of the hand in a trial changes constantly, the entire 3D control space is explored and decoding mistake/drifting of the hand from one trial is affecting the following ones.

In the presented online 8D (5 discrete states) asynchronous alternative bimanual experiments, the control of both virtual avatar and exoskeleton was maintained far above the chance level, without recalibration, over 167 days and 203 days for the exoskeleton and virtual avatar effectors, respectively. The 8D experiments were carried out between 468 and 698 days after the implantation of the WIMAGINE electrodes. These results highlighted the stability of both the REW-MSLM decoder and the neural activity recording method with the two WIMAGINE epidural ECoG recording implants [Mestais et al., 2015]. Additionally, these results demonstrated that CLINATEC epidural ECoG-driven BCI outperformed the state-of-the-art ECoG-based BCIs, gets closer to MEAs in terms of decoding performance and outpaced both the state-of-the-art ECoG and MEAs-based BCIs in terms of decoder stability.

Generally ECoG-driven BCI studies were mainly performed on temporary ECoG grid implantation from 3 to 35 days [Bundy et al., 2016] [Degenhart et al., 2018] [Leuthardt et al., 2004] [Nakanishi et al., 2017, 2013] [Schalk et al., 2008, 2007] [Schalk and Leuthardt, 2011] [Volkova et al., 2019] [W. Wang et al., 2013] [Yanagisawa et al., 2012]. The online closed-loop success rate (SR) for both effectors from 0 to 37 days after the last model calibration highlighted an averaged hit score of 83% and 97% for the 3D hand translation and 1D wrist rotation tasks (averaged on both hands). The 3D hand translation results are similar to the best patient's 3D decoding performance of the current ECoG-driven BCI state of the art [Degenhart et al., 2018]. However, the decoding performance of the referred ECoG-driven BCIs state of the art were evaluated during online 3D center-out experiments which are easier to complete than the alternative point-to-point pursuit tasks used to evaluate the REW-MSLM decoding performance. Additionally, after the completion of the calibration experiments, the REW-MSLM models used to control the virtual avatar and the exoskeleton were fixed for 167 days and 203 days without any model recalibration. In the opposite, in the mentioned state-of-the-art BCI experiments [Degenhart et al., 2018], the model was often recalibrated between test experiments.

The online closed-loop results presented a high stability level and was far above the realized chance level study across all experiments for both effectors. For the exoskeleton experiments, the left hand translation SR seemed to decay between the 37$^{th}$ and the 104$^{th}$ day and stabilize until the end, whereas the right hand translation SR showed higher variability in the performance (between 17% and 100%). For discrete decoding, it was noticeable that switching from the left arm control to the right arm control (and vice versa) represented less than 1% of the errors. Most of the decoding misclassifications were related to two issues. First, the majority of the mistakes were related to false positive idle state activation. Secondly, the decoders struggled to differentiate rotation and translation from the same body side. This difficulty may be related to the similarity of both tasks and consequently lead to brain neural signal pattern activations within a close proximity.

The results seem to demonstrate higher average performance using the exoskeleton than using a virtual avatar effector. Nevertheless, it is worth mentioning that the calibration and test procedures of the models dedicated to the virtual avatar and the exoskeleton control were not performed in the same manner. The 6 calibration experiments of the virtual avatar model were performed during the same week whereas test experiments of the virtual avatar model were carried out weekly at a high frequency. Conversely, the calibration experiments of the model dedicated to the exoskeleton control were distributed in 2 months and the test experiments were less common than virtual avatar test experiments. The lower frequency of exoskeleton experiments compared to virtual avatar experiments may explain the higher variability in the performance using the exoskeleton compared to the virtual avatar. It is likely that a higher frequency of experiments is beneficial for patient's training and control.

Finally, all the pseudo offline and online closed-loop BCI experiments previously presented confirmed in a long term study that direct neural decoding is not limited to individual neuron action potential driven (MEA-based) BCIs and can be achieved from population neuron recordings, particularly from epidural ECoG neural signals. This result challenges the empirical evidence that population neuron recordings are limited to the control of fewer dimensions partly due to lower spatio-temporal resolution and the restricted number of possible somatotopic remapping combination [Degenhart et al., 2018].

## 11.2. Limitations

While the presented Ph.D. results relied on the proof of concept that a tetraplegic patient can control a complex multi-limb effector using a direct neural decoding strategy, several limitations of the Ph.D. research should be mentioned.

Firstly, all the experiments were carried out on a single patient. The results obtained with one patient must be generalized to other disabled subjects.

The current manuscript reports long-term stable performance of high dimensional control of multi-limb exoskeleton and its avatar. While the experiments demonstrated encouraging results, only alternative bimanual control was performed due to the experimental paradigm.

While the current study reported an experimental paradigm with a better exploration of the 3D control space and less restrictive experimental conditions than the traditional center-out tasks, experiments closer to domestic, urban, and professional environments should be designed to move further the technology from clinical trials toward to daily life applications.

Although the Ph.D. research was focused on online closed-loop brain signals decoding, PREW-NPLS, APREW-NPLS and H2M2 algorithms were only tested offline using pseudo-online procedure. Pseudo-online procedures are a good way to test algorithms dedicated to incremental closed loop decoder adaptation, nevertheless, the results obtained are not fully generalizable and the algorithms must be tested during online experiments to bring definitive conclusions of the decoding performance. Indeed, algorithms designed for “human-in-loop” integration in the calibration process will always be difficult to analyze offline.

Finally, it is worth mentioning that the majority of the Ph.D. manuscript was focused on one specific algorithm family. Indeed, The REW-MSLM gating and expert models were estimated with the REW-NPLS algorithm. Other online adaptive decoders reported in BCI studies may be evaluated to be integrated into the REW-MSLM sub-models such as adaptive SVM or LDA for the gate and online adaptive Kalman filter for the experts.

Moreover, the pseudo-online comparative studies performed to appraise the decoding performance improvements of the new decoders were limited to performance comparison with decoders from the NPLS algorithm family. In order to stress the benefits of the PREW-NPLS, APREW-NPLS and H2M2 algorithms, a more representative comparative study with other adaptive real-time decoders should be considered.

During the 8D online closed-loop experiments, the REW-MSLM calibration duration was empirically determined. The calibration phase ended when the experimenter decided that the decoding performance was visually sufficiently high. Therefore, the calibration time was fixed based on subjective criteria and was likely not optimal. The performance differences between 8D virtual avatar and exoskeleton control might be related to the variation in the model calibration procedure. It is likely that the model's calibration during the online experiments and during the pseudo-online experiments were undertrained.

While long-term stable performance of high dimensional (8D) control of multi-limb exoskeleton and its avatar were obtained with REW-MSLM decoder, the 3D left and right hand translation tasks highlighted slow and curved reaching trajectories. These results matched the pseudo-online studies where the cosine similarities of both hand translation tasks were low.

Considering the pseudo-online PREW-NPLS study, the best penalization hyperparameter and electrodes to penalized are probably different during offline and online calibration. Therefore, it is difficult to conclude on the best penalization hyperparameter to select. Moreover, the study was only focused on the spatial feature regularization whereas time and frequency features could be also evaluated.

The APREW-NPLS evaluation was limited to a preliminary study with a reduced number of penalization hyperparameters. Moreover, a better selection of the tested penalization hyperparameters could have been done. Indeed, the $L_1$-PREW-NPLS study highlighted that the penalized models with $\lambda = 0.4, 0.5$ and $0.6$ should lead to identical models. Therefore, the ranking estimated between the models in APREW-NPLS might be biased.

Similarly to APREW-NPLS, H2M2 was limited to a pseudo-online preliminary study.The study was carried out with small training and testing dataset limiting the possible interpretation of the results. Moreover, the state distribution in the training and testing dataset was very different which might affect the decoding performance.

## 11.3. Perspectives

From the hereby-reported studies, several investigations should be carried out to address the previously presented limits and respond to the questions raised in the study.

While the presented results were focused on only one patient, the CLINATEC clinical trial "BCI and Tetraplegia" is still ongoing. The "BCI and Tetraplegia" protocol planned the inclusion in the clinical trial of a total of five patients. A new subject was included in the clinical trial protocol in late 2019 and was implanted the November 19th 2019. Since, the new patient started training and his control performance will be evaluated in future studies.

The 8D multi-limb experiments performed with the REW-MSLM algorithm highlighted encouraging results but were limited to alternative bimanual control as previously mentioned. However, simultaneous bimanual control is theoretically possible thanks to REW-MSLM soft gating strategy: the gating is not a selection of one limb among the others but the mixing of all of them depending on the limb activation probability computed by the HMM gating. REW-MSLM is not limiting to perform simultaneous bimanual effector control. This bimanual experimental paradigm is the nearest perspective of the study. In a more general guideline perspective, the future experimental paradigms should attempt to get closer to domestic, urban, and professional applications with new experiments including bimanual control, grasping control etc.

In the near future, PREW-NPLS, APREW-NPLS and H2M2 algorithms should be tested during online closed-loop experiments to evaluate their performance with an incremental CLDA procedure.

Offline comparative study between PLS algorithm and other state-of-the-art decoders were reported based on able-bodied subjects' and NHPs' ECoG neural signals [Schaeffer, 2017]. For the continuous decoding, PLS was compared to Principal Component Regressions (PCR) with different settings (a hyperparameter defined the percentage of input variable variance explained 20%, 40%, 60% and 80%) and penalization regression algorithms (LASSO) [Schaeffer, 2017]. The comparative study highlighted that PLS and LASSO algorithms outperformed PCR and that LASSO training was computationally more expensive than the PLS algorithm [Schaeffer, 2017]. Therefore, in the reported study [Schaeffer, 2017], PLS was preferred for continuous output variable decoding. For the discrete decoder comparative study, LDA, QDA, Logistic Regression (LR), linear and non-linear SVM coupled with PCA or PLS dimension reduction algorithms were tested. PLS-based decoders outperformed the decoders using PCA dimensional reduction algorithms. In the preclinical dataset, PLS coupled with LR outperformed the other decoders. In the clinical dataset based on able-bodied subjects' ECoG neural signals, PLS-LR decoder was outpaced by the LASSO-LR algorithm but required less

computational resources. Although the comparative study [Schaeffer, 2017] provides an initial overview on the expected decoding performance of the PLS algorithm family, it cannot be fully transposed to the Ph.D. application case. Therefore, new comparative studies between state-of-the-art decoders should be performed [Schaeffer, 2017].

Considering the current REW-MSLM algorithm integrated in the online ECoG-drive BCI of the clinical trial, several opportunities for improvements should be investigated. As mentioned, the previously trained models were fixed without determining an optimal training time and were probably undertrained. The model should be trained for a longer time in the future to accumulate more information and evaluate the impact of a larger dataset on online closed-loop performance. Additionally, studies evaluating the impact of experiment frequency on the decoding performance, the model stability and the patients' adaptations should be performed for a better understanding of the model calibration procedure. Model interpretation and convergence will be further investigated too.

Additionally, each expert is trained with independent dataset which allow removing, changing or adding new experts to the REW-MSLM structure. This model structure enable adding the experts from multiple models calibrated with different experiments in order to simplify the model initialization and the initial control provided to the patient. Gathering the experts from different experiments and models may shorten the model calibration procedure, as it only requires updating the gating model to enable the switch to the newly added expert. Additionally, gate or expert models calibrated to perform neural signal decoding for the same task may be merged to create a new more general model with higher stability. The benefits that the REW-MSLM flexible structure could bring to the calibration procedure using strategies such as grouping models from different tasks or merging sub-models calibrated on the same task should be evaluated.

To improve the decoding performance with faster and less curved reaching trajectories, the integration of other algorithms in the REW-MSLM may be investigated. Various post-processing strategies could be integrated in ABSD to achieved smoother and more straightforward trajectory decoding. Additionally, variations of the current REW-NPLS algorithm integrating penalization or non-linear kernels should be tested in order to improve the decoding performance.

Several articles highlighted in preclinical experiments the benefits of using several linear models calibrated on different movement phases to improve the reaching decoding performance [Kang et al., 2012] [Kim et al., 2003] [Yu et al., 2007]. As an example of future REW-MSLM improvements, an alternative Mixture of Expert architecture with states associated to movement phases could be explored [Schaeffer, 2017].

PREW-NPLS pseudo online study was limited to spatial feature penalization. In order to evaluate the potential decoding performance improvements of the decoder with a norm penalization on other modalities, PREW-NPLS algorithm with group-wised

regularizations applied to the frequency and temporal modalities should be studied. Additionally, the created models should be interpreted in order to extract relevant information on the prevalent frequency bands for neural signal decoding.

New settings as well as the ranking and exploitation-exploration algorithms must be explored to enhance the APREW-NPLS performance. The APREW-NPLS algorithm methodology can be modified depending on the BCI system requirements. The presented preliminary study focused on the decoding performance to evaluate the best model of the set of hyperparameters, nevertheless, other indicators such as the sparsity index can be integrated as criteria in the model ranking algorithm. Moreover, the exploitation-exploration algorithm selecting the models to calibrate during the next update increment can be tuned or totally changed to obtain more or less greedy algorithms. Another possibility is to remove from the possible set of tested hyperparameters the worse configurations to evaluate new settings. Such procedure had already been reported with racing/competitive algorithms (ROAR, F-race family, PaRaMILS, etc.) in other fields than the BCI research area [Hutter et al., 2011b, 2009]. With the discarding procedure, the set of tested hyperparameters is not limited to a finite number of configurations and can explore continuously different hyperparameter settings. Indeed, the algorithm can be extended to an infinite number of penalization hyperparameter with the only restriction than $\lambda \in [0,1]$. A set of finite penalization hyperparameter could be selected at the model initialization. Using the presented performance/ranking procedure, a new penalization hyperparameter set can be computed from the best penalization hyperparameters through optimization strategy (e.g. gradient descent). The best penalized models could be preserved whereas penalized models with low performance could be discarded from the explored set of possible penalization hyperparameters. Additionally, models with $L_1$, $L_{0.5}$ and $L_0$ norm penalization can be calibrated and compared in the same APREW-NPLS algorithm as the penalization type is not constrained during the selection. Finally, at the end of the calibration phase, the APREW-NPLS algorithm disposed of several models with different performance and rank. Instead of using only the best model to decode the neural signals, a prediction of the intended movements can be estimated from all the penalized models. The output variables of each model could be weighted depending on their rank in order to create a prediction based on soft voting ensemble.

Finally, a deeper examination of the benefits of H2M2 classifier will be conducted. Larger training and testing dataset will be acquired in order to perform more reliable performance comparison in offline pseudo-online studies before integrating the H2M2 classifier in REW-MSLM as the gating model. Additionally, H2M2 should be evaluated on more complex tasks with numerous possible state transitions and on applications where the classifier responsiveness has priority compared to the error block frequency rate.

In three years of experiments with chronic ECoG neural signals, CLINATEC built a unique database of BCI experiments. During these experiments, numerous decoding models were calibrated. All the previously trained REW-MSLM models should be analyzed in order to extract information of the general convergence of the models which could be used for prior knowledge initialization of the models.

While the somatotopic remapping (MI) approach may be sufficient for BCI functional compensation, direct neural decoder is mandatory for BCIs dedicated to the rehabilitation of individuals suffering from severe motor disabilities. Direct neural control based on semi-invasive recording systems (epidural ECoG recordings) may open new perspectives for medical BCI applications.

Finally, the REW-MSLM decoder translates the motor cortex activity into commands based on ECoG neural signals recorded with the WIMAGINE implants. The application of the decoder could be considered for other applications possibly requiring the recording of the neural signals from other brain regions. The decoder could also be applied in other medical fields such as, for instance, seizure detection for epileptic patients.

# Bibliography

Aarno, D., Kragic, D., 2006. Layered HMM for motion intention recognition, in: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference On. IEEE, pp. 5130–5135.

Abascal, J., Arruti, A., Martín, J.I., Muguerza, J., 2020. A Hierarchical BCI System Able to Discriminate between Non Intentional Control State and Four Intentional Control Activities. Presented at the International Conference on Physiological Computing Systems, pp. 91–97.

Acharya, S., Fifer, M.S., Benz, H.L., Crone, N.E., Thakor, N.V., 2010. Electrocorticographic amplitude predicts finger positions during slow grasping motions of the hand. J. Neural Eng. 7, 046002. https://doi.org/10.1088/1741-2560/7/4/046002

Achtman, N., Afshar, A., Santhanam, G., Yu, B.M., Ryu, S.I., Shenoy, K.V., 2007. Free-paced high-performance brain–computer interfaces. J. Neural Eng. 4, 336–347. https://doi.org/10.1088/1741-2560/4/3/018

Aggarwal, V., Acharya, S., Tenore, F., Shin, H.-C., Etienne-Cummings, R., Schieber, M.H., Thakor, N.V., 2008. Asynchronous Decoding of Dexterous Finger Movements Using M1 Neurons. IEEE Transactions on Neural Systems and Rehabilitation Engineering 16, 3–14. https://doi.org/10.1109/TNSRE.2007.916289

Aggarwal, V., Mollazadeh, M., Davidson, A.G., Schieber, M.H., Thakor, N.V., 2013. State-based decoding of hand and finger kinematics using neuronal ensemble and LFP activity during dexterous reach-to-grasp movements. J Neurophysiol 109, 3067–3081. https://doi.org/10.1152/jn.01038.2011

Ajiboye, A.B., Willett, F.R., Young, D.R., Memberg, W.D., Murphy, B.A., Miller, J.P., Walter, B.L., Sweet, J.A., Hoyen, H.A., Keith, M.W., Peckham, P.H., Simeral, J.D., Donoghue, J.P., Hochberg, L.R., Kirsch, R.F., 2017. Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. The Lancet 389, 1821–1830. https://doi.org/10.1016/S0140-6736(17)30601-3

AL-Quraishi, M.S., Elamvazuthi, I., Daud, S.A., Parasuraman, S., Borboni, A., 2018. EEG-Based Control for Upper and Lower Limb Exoskeletons and Prostheses: A Systematic Review. Sensors 18, 3342. https://doi.org/10.3390/s18103342

Anderson, N.R., Blakely, T., Schalk, G., Leuthardt, E.C., Moran, D.W., 2012. Electrocorticographic (ECoG) correlates of human arm movements. Exp Brain Res 223, 1–10. https://doi.org/10.1007/s00221-012-3226-1

Andreu-Perez, J., Cao, F., Hagras, H., Yang, G., 2018. A Self-Adaptive Online Brain–Machine Interface of a Humanoid Robot Through a General Type-2 Fuzzy Inference System. IEEE Transactions on Fuzzy Systems 26, 101–116. https://doi.org/10.1109/TFUZZ.2016.2637403

Ang, K.K., Chin, Z.Y., Zhang, H., Guan, C., 2011. Filter Bank Common Spatial Pattern (FBCSP) algorithm using online adaptive and semi-supervised learning, in: The 2011 International Joint Conference on Neural Networks. Presented at the The 2011 International Joint Conference on Neural Networks, pp. 392–396. https://doi.org/10.1109/IJCNN.2011.6033248

Anitha, T., Shanthi, N., Sathiyasheelan, R., Emayavaramban, G., Rajendran, T., 2019. Brain-Computer Interface for Persons with Motor Disabilities - A Review. The Open Biomedical Engineering Journal 13. https://doi.org/10.2174/1874120701913010127

Annese, V.F., Mezzina, G., Venuto, D.D., 2017. Wireless Brain-Computer Interface for Wheelchair Control by Using Fast Machine Learning and Real-Time Hyper-Dimensional Classification, in: Current Trends in Web Engineering, Lecture Notes in Computer Science. Presented at the International Conference on Web Engineering, Springer, Cham, pp. 61–74. https://doi.org/10.1007/978-3-319-74433-9_5

Antelis, J.M., Montesano, L., Ramos-Murguialday, A., Birbaumer, N., Minguez, J., 2017. Decoding Upper Limb Movement Attempt From EEG Measurements of the Contralesional Motor Cortex in Chronic Stroke Patients. IEEE Transactions on Biomedical Engineering 64, 99–111. https://doi.org/10.1109/TBME.2016.2541084

Asghari, P., Soleimani, E., Nazerfard, E., 2019. Online human activity recognition employing hierarchical hidden Markov models. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-019-01380-5

Atyabi, A., Shic, F., Naples, A., 2016. Mixture of autoregressive modeling orders and its implication on single trial EEG classification. Expert Systems with Applications 65, 164–180. https://doi.org/10.1016/j.eswa.2016.08.044

Ayodele, T.O., 2010. Types of Machine Learning Algorithms. New Advances in Machine Learning. https://doi.org/10.5772/9385

Bakhteev, O.Y., Strijov, V.V., 2019. Comprehensive analysis of gradient-based hyperparameter optimization algorithms. Ann Oper Res. https://doi.org/10.1007/s10479-019-03286-z

Ball, T., Schulze-Bonhage, A., Aertsen, A., Mehring, C., 2009. Differential representation of arm movement direction in relation to cortical anatomy and function. J. Neural Eng. 6, 016006. https://doi.org/10.1088/1741-2560/6/1/016006

Bamdadian, A., Guan, C., Ang, K.K., Xu, J., 2013. Improving session-to-session transfer performance of motor imagery-based BCI using adaptive extreme learning machine, in: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2188–2191. https://doi.org/10.1109/EMBC.2013.6609969

Barrese, J.C., Rao, N., Paroo, K., Triebwasser, C., Vargas-Irwin, C., Franquemont, L., Donoghue, J.P., 2013. Failure mode analysis of silicon-based intracortical microelectrode arrays in non-human primates. J. Neural Eng. 10, 066014. https://doi.org/10.1088/1741-2560/10/6/066014

Bashashati, A., Fatourechi, M., Ward, R.K., Birch, G.E., 2007a. A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. J. Neural Eng. 4, R32–R57. https://doi.org/10.1088/1741-2560/4/2/R03

Bashashati, A., Ward, R.K., Birch, G.E., 2007b. Towards Development of a 3-State Self-Paced Brain-Computer Interface. Computational Intelligence and Neuroscience 2007, e84386. https://doi.org/10.1155/2007/84386

Bashashati, H., Ward, R.K., 2017. Ensemble of Neural Network Conditional Random Fields for self-paced Brain Computer Interfaces. Advances in Science, Technology and Engineering Systems 2, 996–1005. https://doi.org/10.25046/aj0203126

Bashashati, H., Ward, R.K., Bashashati, A., Mohamed, A., 2017. Neural network conditional random fields for self-paced brain computer interfaces. Presented at the Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016, pp. 939–943. https://doi.org/10.1109/ICMLA.2016.129

Baxter, B.S., Decker, A., He, B., 2013. Noninvasive control of a robotic arm in multiple dimensions using scalp electroencephalogram, in: 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER). Presented at the 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER), pp. 45–47. https://doi.org/10.1109/NER.2013.6695867

Bellman, R.E., 1961. Adaptive Control Processes: A Guided Tour. Princeton University Press.

Benabid, A.L., Costecalde, T., Eliseyev, A., Charvet, G., Verney, A., Karakas, S., Foerster, M., Lambert, A., Morinière, B., Abroug, N., Schaeffer, M.-C., Moly, A., Sauter-Starace, F., Ratel, D., Moro, C., Torres-Martinez, N., Langar, L., Oddoux, M., Polosan, M., Pezzani, S., Auboiroux, V., Aksenova, T., Mestais, C., Chabardes, S., 2019. An exoskeleton controlled by an epidural wireless brain–machine

interface in a tetraplegic patient: a proof-of-concept demonstration. The Lancet Neurology 0. https://doi.org/10.1016/S1474-4422(19)30321-7

Bergstra, J., Bengio, Y., 2012. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research 13, 281–305.

Bhattacharyya, S., Basu, D., Konar, A., Tibarewala, D.N., 2015. Interval type-2 fuzzy logic based multiclass ANFIS algorithm for real-time EEG based movement control of a robot arm. Robotics and Autonomous Systems 68, 104–115. https://doi.org/10.1016/j.robot.2015.01.007

Bhattacharyya, S., Konar, A., Tibarewala, D.N., 2017a. Motor imagery and error related potential induced position control of a robotic arm. IEEE/CAA Journal of Automatica Sinica 4, 639–650. https://doi.org/10.1109/JAS.2017.7510616

Bhattacharyya, S., Konar, A., Tibarewala, D.N., Hayashibe, M., 2017b. A Generic Transferable EEG Decoder for Online Detection of Error Potential in Target Selection. Front. Neurosci. 11. https://doi.org/10.3389/fnins.2017.00226

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer New York.

Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., 2013. A review of feature selection methods on synthetic data. Knowl Inf Syst 34, 483–519. https://doi.org/10.1007/s10115-012-0487-8

Boostani, R., Graimann, B., Moradi, M.H., Pfurtscheller, G., 2007. A comparison approach toward finding the best feature and classifier in cue-based BCI. Med Bio Eng Comput 45, 403. https://doi.org/10.1007/s11517-007-0169-y

Bouchard, K.E., Bujan, A.F., Chang, E.F., Sommer, F.T., 2017. Sparse coding of ECoG signals identifies interpretable components for speech control in human sensorimotor cortex, in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 3636–3639. https://doi.org/10.1109/EMBC.2017.8037645

Bousseta, R., El Ouakouak, I., Gharbi, M., Regragui, F., 2018. EEG Based Brain Computer Interface for Controlling a Robot Arm Movement Through Thought. IRBM 39, 129–135. https://doi.org/10.1016/j.irbm.2018.02.001

Brain Computer Interface: Neuroprosthetic Control of a Motorized Exoskeleton ClinicalTrials.gov [WWW Document], n.d. URL https://clinicaltrials.gov/ct2/show/NCT02550522 [accessed 8.27.18].

Branco, M.P., Freudenburg, Z.V., Aarnoutse, E.J., Bleichner, M.G., Vansteensel, M.J., Ramsey, N.F., 2017. Decoding hand gestures from primary somatosensory cortex using high-density ECoG. NeuroImage 147, 130–142. https://doi.org/10.1016/j.neuroimage.2016.12.004

Branco, M.P., Freudenburg, Z.V., Aarnoutse, E.J., Vansteensel, M.J., Ramsey, N.F., 2018. Optimization of sampling rate and smoothing improves classification of high frequency power in electrocorticographic brain signals. Biomed. Phys. Eng. Express 4, 045012. https://doi.org/10.1088/2057-1976/aac3ac

Brandman, D.M., Hosman, T., Saab, J., Burkhart, M.C., Shanahan, B.E., Ciancibello, J.G., Sarma, A.A., Milstein, D.J., Vargas-Irwin, C.E., Franco, B., Jessica Kelemen, Blabe, C., Murphy, B.A., Young, D.R., Willett, F.R., Pandarinath, C., Stavisky, S.D., Kirsch, R.F., Walter, B.L., Ajiboye, A.B., Cash, S.S., Eskandar, E.N., Miller, J.P., Sweet, J.A., Shenoy, K.V., Henderson, J.M., Jarosiewicz, B., Harrison, M.T., Simeral, J.D., Hochberg, L.R., 2018. Rapid calibration of an intracortical brain–computer interface for people with tetraplegia. J. Neural Eng. 15, 026007. https://doi.org/10.1088/1741-2552/aa9ee7

Bro, R., 1998. Multi-way analysis in the food industry: models, algorithms, and applications. Københavns UniversitetKøbenhavns Universitet, LUKKET: 2012 Det Biovidenskabelige Fakultet for Fødevarer, Veterin\a ermedicin og NaturressourcerFaculty of Life Sciences, LUKKET: 2012 Institut for

FødevarevidenskabDepartment of Food Science, LUKKET: 2012 Kvalitet og TeknologiQuality & Technology.

Bro, R., 1996. Multiway calibration. Multilinear PLS. J. Chemometrics 10, 47–61. https://doi.org/10.1002/(SICI)1099-128X(199601)10:1<47::AID-CEM400>3.0.CO;2-C

Brodu, N., Lotte, F., Lécuyer, A., 2011. Comparative study of band-power extraction techniques for Motor Imagery classification, in: 2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB). Presented at the 2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), pp. 1–6. https://doi.org/10.1109/CCMB.2011.5952105

Brunner, C., Scherer, R., Graimann, B., Supp, G., Pfurtscheller, G., 2006. Online control of a brain-computer interface using phase synchronization. IEEE Transactions on Biomedical Engineering 53, 2501–2506. https://doi.org/10.1109/TBME.2006.881775

Bruns, A., 2004. Fourier-, Hilbert- and wavelet-based signal analysis: are they really different approaches? Journal of Neuroscience Methods 137, 321–332. https://doi.org/10.1016/j.jneumeth.2004.03.002

Bundy David T., Souders Lauren, Baranyai Kelly, Leonard Laura, Schalk Gerwin, Coker Robert, Moran Daniel W., Huskey Thy, Leuthardt Eric C., 2017. Contralesional Brain–Computer Interface Control of a Powered Exoskeleton for Motor Recovery in Chronic Stroke Survivors. Stroke 48, 1908–1915. https://doi.org/10.1161/STROKEAHA.116.016304

Bundy, D.T., Pahwa, M., Szrama, N., Leuthardt, E.C., 2016. Decoding three-dimensional reaching movements using electrocorticographic signals in humans. J. Neural Eng. 13, 026021. https://doi.org/10.1088/1741-2560/13/2/026021

Buzsáki, G., Anastassiou, C.A., Koch, C., 2012. The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes. Nat Rev Neurosci 13, 407–420. https://doi.org/10.1038/nrn3241

Cabrera, A.F., Farina, D., Dremstrup, K., 2010. Comparison of feature selection and classification methods for a brain–computer interface driven by non-motor imagery. Med Biol Eng Comput 48, 123–132. https://doi.org/10.1007/s11517-009-0569-2

Cano-Izquierdo, J., Ibarrola, J., Almonacid, M., 2012. Improving Motor Imagery Classification With a New BCI Design Using Neuro-Fuzzy S-dFasArt. IEEE Transactions on Neural Systems and Rehabilitation Engineering 20, 2–7. https://doi.org/10.1109/TNSRE.2011.2169991

Cano-Izquierdo, J.-M., Ibarrola, J., Almonacid, M., 2012. Improving Motor Imagery Classification With a New BCI Design Using Neuro-Fuzzy S-dFasArt. IEEE Transactions on Neural Systems and Rehabilitation Engineering 20, 2–7. https://doi.org/10.1109/TNSRE.2011.2169991

Cantillo-Negrete, J., Carino-Escobar, R.I., Carrillo-Mora, P., Elias-Vinas, D., Gutierrez-Martinez, J., 2018. Motor imagery-based brain-computer interface coupled to a robotic hand orthosis aimed for neurorehabilitation of stroke patients. Journal of Healthcare Engineering 2018. https://doi.org/10.1155/2018/1624637

Cappé, O., 2011. Online EM Algorithm for Hidden Markov Models. J. Comput. Graph. Statist 22.

Carmena, J.M., Lebedev, M.A., Crist, R.E., O'Doherty, J.E., Santucci, D.M., Dimitrov, D.F., Patil, P.G., Henriquez, C.S., Nicolelis, M.A.L., 2003. Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates. PLOS Biology 1, e42. https://doi.org/10.1371/journal.pbio.0000042

Carvalho, A.X., Tanner, M.A., 2003. Hypothesis testing in mixtures-of-experts of generalized linear time series, in: 2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings. Presented at the 2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings., pp. 285–292. https://doi.org/10.1109/CIFER.2003.1196273

Carvalho, R., Dias, N., Cerqueira, J.J., 2019. Brain-machine interface of upper limb recovery in stroke patients rehabilitation: A systematic review. Physiotherapy Research International 24, e1764. https://doi.org/10.1002/pri.1764

Chae, Y., Jeong, J., Jo, S., 2012. Toward brain-actuated humanoid robots: Asynchronous direct control using an EEG-Based BCI. IEEE Transactions on Robotics 28, 1131–1144. https://doi.org/10.1109/TRO.2012.2201310

Chan, A.D.C., Englehart, K.B., 2005. Continuous myoelectric control for powered prostheses using hidden Markov models. IEEE Transactions on Biomedical Engineering 52, 121–124. https://doi.org/10.1109/TBME.2004.836492

Chan, A.T., Quiroz, J.C., Dascalu, S., Harris, F.C., Jr., 2015. An overview of brain computer interfaces. Presented at the Proceedings of the 30th International Conference on Computers and Their Applications, CATA 2015, pp. 327–333.

Chao, Z.C., Nagasaka, Y., Fujii, N., 2010. Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey. Front. Neuroeng. 3. https://doi.org/10.3389/fneng.2010.00003

Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S., 2002. Choosing Multiple Parameters for Support Vector Machines. Machine Learning 46, 131–159. https://doi.org/10.1023/A:1012450327387

Chase, S.M., Schwartz, A.B., Kass, R.E., 2009. Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain–computer interface algorithms. Neural Networks, Brain-Machine Interface 22, 1203–1213. https://doi.org/10.1016/j.neunet.2009.05.005

Chavarriaga, R., Sobolewski, A., Millán, J. del R., 2014. Errare machinale est: the use of error-related potentials in brain-machine interfaces. Front. Neurosci. 8. https://doi.org/10.3389/fnins.2014.00208

Chen, B., Zhao, S., Seth, S., Principe, J.C., 2012. Online efficient learning with quantized KLMS and L1 regularization, in: The 2012 International Joint Conference on Neural Networks (IJCNN). Presented at the The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. https://doi.org/10.1109/IJCNN.2012.6252455

Chen, Bilian., He, Simai., Li, Zhening., Zhang, Shuzhong., 2012. Maximum Block Improvement and Polynomial Optimization. SIAM J. Optim. 22, 87–107. https://doi.org/10.1137/110834524

Chen, C., Shin, D., Watanabe, H., Nakanishi, Y., Kambara, H., Yoshimura, N., Nambu, A., Isa, T., Nishimura, Y., Koike, Y., 2013. Prediction of Hand Trajectory from Electrocorticography Signals in Primary Motor Cortex. PLOS ONE 8, e83534. https://doi.org/10.1371/journal.pone.0083534

Chen, C.K., Fang, W.C., 2017. A reliable brain-computer interface based on SSVEP using online recursive independent component analysis, in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2798–2801. https://doi.org/10.1109/EMBC.2017.8037438

Chiappa, S., 2006. Analysis and Classification of EEG Signals using Probabilistic Models for Brain Computer Interfaces. Ecole Polytechnique Fédérale de Lausanne, Lausane.

Chin, C.M., Popovic, M.R., Thrasher, A., Cameron, T., Lozano, A., Chen, R., 2007. Identification of arm movements using correlation of electrocorticographic spectral components and kinematic recordings. J. Neural Eng. 4, 146. https://doi.org/10.1088/1741-2560/4/2/014

Chis, T., Harrison, P.G., 2015. Adapting Hidden Markov Models for Online Learning. Electronic Notes in Theoretical Computer Science 318, 109–127. https://doi.org/10.1016/j.entcs.2015.10.022

Choi, H., Lee, J., Park, J., Lee, S., Ahn, K., Kim, I.Y., Lee, K.-M., Jang, D.P., 2018. Improved prediction of bimanual movements by a two-staged (effector-then-trajectory) decoder with epidural ECoG in nonhuman primates. J. Neural Eng. 15, 016011. https://doi.org/10.1088/1741-2552/aa8a83

Chowdhury, A., Raza, H., Meena, Y.K., Dutta, A., Prasad, G., 2017. Online Covariate Shift Detection based Adaptive Brain-Computer Interface to Trigger Hand Exoskeleton Feedback for Neuro-Rehabilitation. IEEE Transactions on Cognitive and Developmental Systems 1–1. https://doi.org/10.1109/TCDS.2017.2787040

Chu, D., Liao, L.-Z., Ng, M.K.-P., Wang, X., 2015. Incremental Linear Discriminant Analysis: A Fast Algorithm and Comparisons. IEEE Transactions on Neural Networks and Learning Systems 26, 2716–2735. https://doi.org/10.1109/TNNLS.2015.2391201

Cichocki, A., Mandic, D., Lathauwer, L.D., Zhou, G., Zhao, Q., Caiafa, C., PHAN, H.A., 2015. Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis. IEEE Signal Processing Magazine 32, 145–163. https://doi.org/10.1109/MSP.2013.2297439

Cincotti, F., Mattia, D., Aloise, F., Bufalari, S., Astolfi, L., De Vico Fallani, F., Tocci, A., Bianchi, L., Marciani, M.G., Gao, S., Millan, J., Babiloni, F., 2008. High-resolution EEG techniques for brain–computer interface applications. Journal of Neuroscience Methods, Brain-Computer Interfaces (BCIs) 167, 31–42. https://doi.org/10.1016/j.jneumeth.2007.06.031

Cincotti, F., Scipione, A., Timperi, A., Mattia, D., Marciani, A.G., Millan, J., Salinari, S., Bianchi, L., Bablioni, F., 2003. Comparison of different feature classifiers for brain computer interfaces, in: First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings. Presented at the First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings., pp. 645–647. https://doi.org/10.1109/CNE.2003.1196911

Clerc, M., Bougrain, L., Lotte, F., 2016a. Brain-Computer Interfaces 1. Wiley-ISTE.

Clerc, M., Daucé, E., Mattout, J., 2016b. Adaptive Methods in Machine Learning, in: Brain–Computer Interfaces 1. Wiley-Blackwell, pp. 207–232. https://doi.org/10.1002/9781119144977.ch10

Collinger, J.L., Wodlinger, B., Downey, J.E., Wang, W., Tyler-Kabara, E.C., Weber, D.J., McMorland, A.J., Velliste, M., Boninger, M.L., Schwartz, A.B., 2013. High-performance neuroprosthetic control by an individual with tetraplegia. The Lancet 381, 557–564. https://doi.org/10.1016/S0140-6736(12)61816-9

Cong, F., Lin, Q.-H., Kuang, L.-D., Gong, X.-F., Astikainen, P., Ristaniemi, T., 2015. Tensor decomposition of EEG signals: A brief review. Journal of Neuroscience Methods 248, 59–69. https://doi.org/10.1016/j.jneumeth.2015.03.018

Corralejo, R., Hornero, R., Álvarez, D., 2011. Feature selection using a genetic algorithm in a motor imagery-based Brain Computer Interface, in: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Presented at the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 7703–7706. https://doi.org/10.1109/IEMBS.2011.6091898

Costa, Ana.P., Møller, Jakob.S., Iversen, Helle.K., Puthusserypady, S., 2018. An adaptive CSP filter to investigate user independence in a 3-class MI-BCI paradigm. Computers in Biology and Medicine 103, 24–33. https://doi.org/10.1016/j.compbiomed.2018.09.021

Costecalde, T., Aksenova, T., Torres-Martinez, N., Eliseyev, A., Mestais, C., Moro, C., Benabid, A.L., 2017. A Long-Term BCI Study With ECoG Recordings in Freely Moving Rats. Neuromodulation: Technology at the Neural Interface 21, 149–159. https://doi.org/10.1111/ner.12628

Cozza, F., Galdi, P., Serra, A., Pasqua, G., Pavone, L., Tagliaferri, R., 2020. Dimension Reduction Techniques in a Brain–Computer Interface Application, in: Esposito, A., Faundez-Zanuy, M., Morabito, F.C., Pasero, E. [Eds.], Neural Approaches to Dynamics of Signal Exchanges, Smart Innovation, Systems and Technologies. Springer Singapore, Singapore, pp. 107–118. https://doi.org/10.1007/978-981-13-8950-4_11

Cramer, R.D., 1993. Partial Least Squares (PLS): Its strengths and limitations. Perspectives in Drug Discovery and Design 1, 269–278. https://doi.org/10.1007/BF02174528

Cunningham, J.P., Gilja, V., Ryu, S.I., Shenoy, K.V., 2009. Methods for estimating neural firing rates, and their application to brain–machine interfaces. Neural Networks, Brain-Machine Interface 22, 1235–1246. https://doi.org/10.1016/j.neunet.2009.02.004

Cunningham, J.P., Nuyujukian, P., Gilja, V., Chestek, C.A., Ryu, S.I., Shenoy, K.V., 2010. A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. Journal of Neurophysiology 105, 1932–1949. https://doi.org/10.1152/jn.00503.2010

Dähne, S., Meinecke, F.C., Haufe, S., Höhne, J., Tangermann, M., Müller, K.-R., Nikulin, V.V., 2014. SPoC: A novel framework for relating the amplitude of neuronal oscillations to behaviorally relevant parameters. NeuroImage 86, 111–122. https://doi.org/10.1016/j.neuroimage.2013.07.079

Dangi, S., 2015. Closed-loop decoder adaptation algorithms for brain-machine interface systems.

Dangi, S., Gowda, S., Héliot, R., Carmena, J.M., 2011. Adaptive Kalman filtering for closed-loop Brain-Machine Interface systems, in: 2011 5th International IEEE/EMBS Conference on Neural Engineering. Presented at the 2011 5th International IEEE/EMBS Conference on Neural Engineering, pp. 609–612. https://doi.org/10.1109/NER.2011.5910622

Dangi, S., Gowda, S., Moorman, H.G., Orsborn, A.L., So, K., Shanechi, M., Carmena, J.M., 2014. Continuous Closed-Loop Decoder Adaptation with a Recursive Maximum Likelihood Algorithm Allows for Rapid Performance Acquisition in Brain-Machine Interfaces. Neural Computation 26, 1811–1839. https://doi.org/10.1162/NECO_a_00632

Darmanjian, S., Sung Phil Kim, Nechyba, M.C., Morrison, S., Principe, J., Wessberg, J., Nicolelis, M.A.L., 2003. Bimodal brain-machine interface for motor control of robotic prosthetic, in: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453). Presented at the Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), pp. 3612–3617 vol.3. https://doi.org/10.1109/IROS.2003.1249716

Dayal, B.S., MacGregor, J.F., 1997a. Improved PLS algorithms. J. Chemometrics 11, 73–85. https://doi.org/10.1002/(SICI)1099-128X(199701)11:1<73::AID-CEM435>3.0.CO;2-#

Dayal, B.S., MacGregor, J.F., 1997b. Recursive exponentially weighted PLS and its applications to adaptive control and prediction. Journal of Process Control 7, 169–179.

Degenhart, A.D., Hiremath, S.V., Yang, Y., Foldes, S., Collinger, J.L., Michael Boninger, Tyler-Kabara, E.C., Wang, W., 2018. Remapping cortical modulation for electrocorticographic brain–computer interfaces: a somatotopy-based approach in individuals with upper-limb paralysis. J. Neural Eng. 15, 026021. https://doi.org/10.1088/1741-2552/aa9bfb

Delgado Saa, J.F., Çetin, M., 2011. Hidden conditional random fields for classification of imaginary motor tasks from EEG data. Presented at the European Signal Processing Conference, pp. 1377–1381.

Delisle-Rodriguez, D., Villa-Parra, A.C., Bastos-Filho, T., López-Delis, A., Frizera-Neto, A., Krishnan, S., Rocon, E., 2017. Adaptive Spatial Filter Based on Similarity Indices to Preserve the Neural Information on EEG Signals during On-Line Processing. Sensors 17, 2725. https://doi.org/10.3390/s17122725

Dietterich, T.G., 2002. Machine Learning for Sequential Data: A Review, in: Caelli, T., Amin, A., Duin, R.P.W., de Ridder, D., Kamel, M. [Eds.], Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 15–30.

Digalakis, V.V., 1999. Online adaptation of hidden Markov models using incremental estimation algorithms. IEEE Transactions on Speech and Audio Processing 7, 253–261.

Djemal, R., Bazyed, A.G., Belwafi, K., Gannouni, S., Kaaniche, W., 2016. Three-Class EEG-Based Motor Imagery Classification Using Phase-Space Reconstruction Technique. Brain Sciences 6, 36. https://doi.org/10.3390/brainsci6030036

Dobiáš, M., Štastný, J., 2016. Movement EEG classification using parallel Hidden Markov Models, in: 2016 International Conference on Applied Electronics (AE). Presented at the 2016 International Conference on Applied Electronics (AE), pp. 65–68. https://doi.org/10.1109/AE.2016.7577243

Donati, A.R.C., Shokur, S., Morya, E., Campos, D.S.F., Moioli, R.C., Gitti, C.M., Augusto, P.B., Tripodi, S., Pires, C.G., Pereira, G.A., Brasil, F.L., Gallo, S., Lin, A.A., Takigami, A.K., Aratanha, M.A., Joshi, S., Bleuler, H., Cheng, G., Rudolph, A., Nicolelis, M.A.L., 2016. Long-Term Training with a Brain-Machine Interface-Based Gait Protocol Induces Partial Neurological Recovery in Paraplegic Patients. Scientific Reports 6, 30383. https://doi.org/10.1038/srep30383

Donchin, O., Gribova, A., Steinberg, O., Bergman, H., Vaadia, E., 1998. Primary motor cortex is involved in bimanual coordination. Nature 395, 274–278. https://doi.org/10.1038/26220

Dong, E., Li, C., Li, L., Du, S., Belkacem, A.N., Chen, C., 2017. Classification of multi-class motor imagery with a novel hierarchical SVM algorithm for brain–computer interfaces. Med Biol Eng Comput 55, 1809–1818. https://doi.org/10.1007/s11517-017-1611-4

Dôres, S.C.N. das, Soares, C., Ruiz, D., 2018. Bandit-Based Automated Machine Learning, in: 2018 7th Brazilian Conference on Intelligent Systems (BRACIS). Presented at the 2018 7th Brazilian Conference on Intelligent Systems (BRACIS), pp. 121–126. https://doi.org/10.1109/BRACIS.2018.00029

Dornhege, G., Blankertz, B., Curio, G., Muller, K.R., 2004. Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multiclass paradigms. IEEE Transactions on Biomedical Engineering 51, 993–1002. https://doi.org/10.1109/TBME.2004.827088

Du, A., Yang, S., Liu, W., Huang, H., 2018. Decoding ECoG Signal with Deep Learning Model Based on LSTM, in: TENCON 2018 - 2018 IEEE Region 10 Conference. Presented at the TENCON 2018 - 2018 IEEE Region 10 Conference, pp. 0430–0435. https://doi.org/10.1109/TENCON.2018.8650433

Ebrahimpour, R., Babakhani, K., Mohammad-Noori, M., 2012. EEG-based motor imagery classification using wavelet coefficients and ensemble classifiers, in: The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012). Presented at the The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012), pp. 458–463. https://doi.org/10.1109/AISP.2012.6313791

Ebrahimpour, R., Kabir, E., Yousefi, M.R., 2011. Improving mixture of experts for view-independent face recognition using teacher-directed learning. Machine Vision and Applications 22, 421–432. https://doi.org/10.1007/s00138-009-0232-9

Eckert, M.J., Martin, M.J., 2017. Trauma: Spinal Cord Injury. Surgical Clinics of North America, Contemporary Management of Civilian Trauma 97, 1031–1045. https://doi.org/10.1016/j.suc.2017.06.008

Edelman, B.J., Meng, J., Suma, D., Zurn, C., Nagarajan, E., Baxter, B.S., Cline, C.C., He, B., 2019. Noninvasive neuroimaging enhances continuous neural tracking for robotic device control. Science Robotics 4, eaaw6844. https://doi.org/10.1126/scirobotics.aaw6844

Efimova, V., Filchenkov, A., Shalyto, A., 2019. Reinforcement-Based Simultaneous Algorithm and Its Hyperparameters Selection, in: Strijov, V.V., Ignatov, D.I., Vorontsov, K.V. [Eds.], Intelligent Data Processing, Communications in Computer and Information Science. Springer International Publishing, pp. 15–27.

Elgharabawy, A., Wahed, M.A., 2016. Decoding of finger movement using kinematic model classification and regression model switching, in: 2016 8th Cairo International Biomedical Engineering Conference (CIBEC). Presented at the 2016 8th Cairo International Biomedical Engineering Conference (CIBEC), pp. 84–89. https://doi.org/10.1109/CIBEC.2016.7836126

Eliseyev, A., Aksenova, T., 2016. Penalized Multi-Way Partial Least Squares for Smooth Trajectory Decoding from Electrocorticographic (ECoG) Recording. PLOS ONE 11, e0154878. https://doi.org/10.1371/journal.pone.0154878

Eliseyev, A., Aksenova, T., 2014. Stable and artifact-resistant decoding of 3D hand trajectories from ECoG signals using the generalized additive model. J. Neural Eng. 11, 066005. https://doi.org/10.1088/1741-2560/11/6/066005

Eliseyev, A., Aksenova, T., 2013. Recursive N-way partial least squares for brain-computer interface. PloS one 8, e69962.

Eliseyev, A., Auboiroux, V., Costecalde, T., Langar, L., Charvet, G., Mestais, C., Aksenova, T., Benabid, A.-L., 2017. Recursive Exponentially Weighted N-way Partial Least Squares Regression with Recursive-Validation of Hyper-Parameters in Brain-Computer Interface Applications. Scientific Reports 7, 16281. https://doi.org/10.1038/s41598-017-16579-9

Eliseyev, A., Mestais, C., Charvet, G., Sauter, F., Abroug, N., Arizumi, N., Cokgungor, S., Costecalde, T., Foerster, M., Korczowski, L., Morinière, B., Porcherot, J., Pradal, J., Ratel, D., Tarrin, N., Torres-Martinez, N., Verney, A., Aksenova, T., Benabid, A.L., 2014. CLINATEC #x00AE; BCI platform based on the ECoG-recording implant WIMAGINE #x00AE; and the innovative signal-processing: Preclinical results, in: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Presented at the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1222–1225. https://doi.org/10.1109/EMBC.2014.6943817

Eliseyev, A., Moro, C., Faber, J., Wyss, A., Torres, N., Mestais, C., Benabid, A.L., Aksenova, T., 2012. L1-Penalized N-way PLS for subset of electrodes selection in BCI experiments. J. Neural Eng. 9, 045010. https://doi.org/10.1088/1741-2560/9/4/045010

Engel, S., Aksenova, T., Eliseyev, A., 2017. Kernel-Based NPLS for Continuous Trajectory Decoding from ECoG Data for BCI Applications, in: Tichavský, P., Babaie-Zadeh, M., Michel, O.J.J., Thirion-Moreau, N. [Eds.], Latent Variable Analysis and Signal Separation, Lecture Notes in Computer Science. Springer International Publishing, pp. 417–426. https://doi.org/10.1007/978-3-319-53547-0_39

Estrin, S., Martinez-Cancino, R., Makeig, S., Gilja, V., 2018. Improving Classification Accuracy in Cortical Surface Recordings Using ICA-Based Features, in: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Presented at the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3285–3288. https://doi.org/10.1109/SMC.2018.00556

Eva, O., Lazar, A., 2019. Amplitude Modulation Index as Feature in a Brain Computer Interface. TS 36, 201–207. https://doi.org/10.18280/ts.360301

Even-Chen, N., Stavisky, S.D., Kao, J.C., Ryu, S.I., Shenoy, K.V., 2017. Augmenting intracortical brain-machine interface with neurally driven error detectors. J Neural Eng 14, 066007. https://doi.org/10.1088/1741-2552/aa8dc1

Even-Chen, N., Stavisky, S.D., Pandarinath, C., Nuyujukian, P., Blabe, C.H., Hochberg, L.R., Henderson, J.M., Shenoy, K.V., 2018. Feasibility of Automatic Error Detect-and-Undo System in Human Intracortical Brain–Computer Interfaces. IEEE Transactions on Biomedical Engineering 65, 1771–1784. https://doi.org/10.1109/TBME.2017.2776204

Faber, N. (Klaas) M., Bro, R., Hopke, P.K., 2003. Recent developments in CANDECOMP/PARAFAC algorithms: a critical review. Chemometrics and Intelligent Laboratory Systems 65, 119–137. https://doi.org/10.1016/S0169-7439(02)00089-8

Faisan, S., Thoraval, L., Armspach, J.-, Metz-Lutz, M.-, Heitz, F., 2005. Unsupervised learning and mapping of active brain functional MRI signals based on hidden semi-Markov event sequence models. IEEE Transactions on Medical Imaging 24, 263–276. https://doi.org/10.1109/TMI.2004.841225

Faller, J., Vidaurre, C., Solis-Escalante, T., Neuper, C., Scherer, R., 2012. Autocalibration and Recurrent Adaptation: Towards a Plug and Play Online ERD-BCI. IEEE Transactions on Neural Systems and Rehabilitation Engineering 20, 313–319. https://doi.org/10.1109/TNSRE.2012.2189584

Fan, J.M., Nuyujukian, P., Kao, J.C., Chestek, C.A., Ryu, S.I., Shenoy, K.V., 2014. Intention estimation in brain–machine interfaces. J. Neural Eng. 11, 016004. https://doi.org/10.1088/1741-2560/11/1/016004

Faradji, F., Ward, R.K., Birch, G.E., 2009. Plausibility assessment of a 2-state self-paced mental task-based BCI using the no-control performance analysis. Journal of Neuroscience Methods 180, 330–339. https://doi.org/10.1016/j.jneumeth.2009.03.011

Farooq, F., Rashid, N., Farooq, A., Ahmed, M., Zeb, A., Iqbal, J., 2019. Motor Imagery based Multivariate EEG Signal Classification for Brain Controlled Interface Applications, in: 2019 7th International Conference on Mechatronics Engineering (ICOM). Presented at the 2019 7th International Conference on Mechatronics Engineering (ICOM), pp. 1–6. https://doi.org/10.1109/ICOM47790.2019.8952008

Farrokhi, B., Erfanian, A., 2018. A piecewise probabilistic regression model to decode hand movement trajectories from epidural and subdural ECoG signals. J. Neural Eng. 15, 036020. https://doi.org/10.1088/1741-2552/aab290

Fatourechi, M., Bashashati, A., Ward, R.K., Birch, G.E., 2007. EMG and EOG artifacts in brain computer interface systems: A survey. Clinical Neurophysiology 118, 480–494. https://doi.org/10.1016/j.clinph.2006.10.019

Feng, C., Zhang, J., 2018. Reinforcement Learning based Dynamic Model Selection for Short-Term Load Forecasting.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F., 2015. Efficient and Robust Automated Machine Learning, in: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. [Eds.], Advances in Neural Information Processing Systems 28. Curran Associates, Inc., pp. 2962–2970.

Fifer, M.S., Hotson, G., Wester, B.A., McMullen, D.P., Wang, Y., Johannes, M.S., Katyal, K.D., Helder, J.B., Para, M.P., Vogelstein, R.J., Anderson, W.S., Thakor, N.V., Crone, N.E., 2014. Simultaneous Neural Control of Simple Reaching and Grasping With the Modular Prosthetic Limb Using Intracranial EEG. IEEE Transactions on Neural Systems and Rehabilitation Engineering 22, 695–705. https://doi.org/10.1109/TNSRE.2013.2286955

Fine, S., Singer, Y., Tishby, N., 1998. The Hierarchical Hidden Markov Model: Analysis and Applications. Machine Learning 32, 41–62. https://doi.org/10.1023/A:1007469218079

Fitzsimmons, N.A., Lebedev, M.A., Peikon, I.D., Nicolelis, M.A.L., 2009. Extracting Kinematic Parameters for Monkey Bipedal Walking from Cortical Neuronal Ensemble Activity. Front Integr Neurosci 3. https://doi.org/10.3389/neuro.07.003.2009

Flamary, R., Rakotomamonjy, A., 2012. Decoding Finger Movements from ECoG Signals Using Switching Linear Models. Front Neurosci 6. https://doi.org/10.3389/fnins.2012.00029

Flamary, R., Rakotomamonjy, A., Sebag, M., 2016. Statistical Learning for BCIs, in: Brain–Computer Interfaces 1. John Wiley & Sons, Ltd, pp. 185–205. https://doi.org/10.1002/9781119144977.ch9

Flint, R.D., Rosenow, J.M., Tate, M.C., Slutzky, M.W., 2016. Continuous decoding of human grasp kinematics using epidural and subdural signals. J. Neural Eng. 14, 016005. https://doi.org/10.1088/1741-2560/14/1/016005

Foodeh, R., Ebadollahi, S., Daliri, M.R., 2020. Regularized Partial Least Square Regression for Continuous Decoding in Brain-Computer Interfaces. Neuroinform. https://doi.org/10.1007/s12021-020-09455-x

Frolov, A.A., Mokienko, O., Lyukmanov, R., Biryukova, E., Kotov, S., Turbina, L., Nadareyshvily, G., Bushkova, Y., 2017. Post-stroke Rehabilitation Training with a Motor-Imagery-Based Brain-Computer Interface (BCI)-Controlled Hand Exoskeleton: A Randomized Controlled Multicenter Trial. Front. Neurosci. 11. https://doi.org/10.3389/fnins.2017.00400

Fukuma, R., Yanagisawa, T., Saitoh, Y., Hosomi, K., Kishima, H., Shimizu, T., Sugata, H., Yokoi, H., Hirata, M., Kamitani, Y., Yoshimine, T., 2016. Real-Time Control of a Neuroprosthetic Hand by

Magnetoencephalographic Signals from Paralysed Patients. Scientific Reports 6, 1–14. https://doi.org/10.1038/srep21781

Fukuma, R., Yanagisawa, T., Yorifuji, S., Kato, R., Yokoi, H., Hirata, M., Saitoh, Y., Kishima, H., Kamitani, Y., Yoshimine, T., 2015. Closed-Loop Control of a Neuroprosthetic Hand by Magnetoencephalographic Signals. PLOS ONE 10, e0131547. https://doi.org/10.1371/journal.pone.0131547

Garcia, G.N., Ebrahimi, T., Vesin, J.-M., 2003. Support vector EEG classification in the Fourier and time-frequency correlation domains, in: First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings. Presented at the First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings., pp. 591–594. https://doi.org/10.1109/CNE.2003.1196897

García-Cossio, E., Severens, M., Nienhuis, B., Duysens, J., Desain, P., Keijsers, N., Farquhar, J., 2015. Decoding Sensorimotor Rhythms during Robotic-Assisted Treadmill Walking for Brain Computer Interface (BCI) Applications. PLoS One 10. https://doi.org/10.1371/journal.pone.0137910

Garrett, D., Peterson, D.A., Anderson, C.W., Thaut, M.H., 2003. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11, 141–144. https://doi.org/10.1109/TNSRE.2003.814441

Geladi, P., Kowalski, B.R., 1986. Partial least-squares regression: a tutorial. Analytica Chimica Acta 185, 1–17. https://doi.org/10.1016/0003-2670(86)80028-9

Georgopoulos, A.P., Carpenter, A.F., 2015. Coding of movements in the motor cortex. Current Opinion in Neurobiology, Motor circuits and action 33, 34–39. https://doi.org/10.1016/j.conb.2015.01.012

Georgopoulos, A.P., Schwartz, A.B., Kettner, R.E., 1986. Neuronal population coding of movement direction. Science 233, 1416–1419. https://doi.org/10.1126/science.3749885

Gilja, V., Nuyujukian, P., Chestek, C.A., Cunningham, J.P., Yu, B.M., Fan, J.M., Churchland, M.M., Kaufman, M.T., Kao, J.C., Ryu, S.I., Shenoy, K.V., 2012. A high-performance neural prosthesis enabled by control algorithm design. Nature Neuroscience 15, 1752. https://doi.org/10.1038/nn.3265

Giordani, P., Rocci, R., 2013. Constrained Candecomp/Parafac via the Lasso. Psychometrika 78, 669–684. https://doi.org/10.1007/s11336-013-9321-9

Gunasekera, B., Saxena, T., Bellamkonda, R., Karumbaiah, L., 2015. Intracortical Recording Interfaces: Current Challenges to Chronic Recording Function. ACS Chem. Neurosci. 6, 68–83. https://doi.org/10.1021/cn5002864

Gundelakh, F., Stankevich, L., Sonkin, K., 2018. Mobile robot control based on noninvasive brain-computer interface using hierarchical classifier of imagined motor commands. MATEC Web Conf. 161, 03003. https://doi.org/10.1051/matecconf/201816103003

Gunduz, A., Brunner, P., Sharma, M., Leuthardt, E.C., Ritaccio, A.L., Pesaran, B., Schalk, G., 2016. Differential roles of high gamma and local motor potentials for movement preparation and execution. Brain-Computer Interfaces 3, 88–102. https://doi.org/10.1080/2326263X.2016.1179087

Guo, B., Hu, J., Wu, W., Peng, Q., Wu, F., 2019. The Tabu_Genetic Algorithm: A Novel Method for Hyper-Parameter Optimization of Learning Algorithms. Electronics 8, 579. https://doi.org/10.3390/electronics8050579

Gupta, A., Agrawal, R.K., Kirar, J.S., Kaur, B., Ding, W., Lin, C.-T., Andreu-Perez, J., Prasad, M., 2020. A hierarchical meta-model for multi-class mental task based brain-computer interfaces. Neurocomputing 389, 207–217. https://doi.org/10.1016/j.neucom.2018.07.094

Gutta, S., Huang, J.R.J., Jonathon, P., Wechsler, H., 2000. Mixture of experts for classification of gender, ethnic origin, and pose of human faces. IEEE Transactions on Neural Networks 11, 948–960. https://doi.org/10.1109/72.857774

Gysels, E., Celka, P., 2004. Phase synchronization for the recognition of mental tasks in a brain-computer interface. IEEE Transactions on Neural Systems and Rehabilitation Engineering 12, 406–415. https://doi.org/10.1109/TNSRE.2004.838443

Hämäläinen, M., Hari, R., Ilmoniemi, R.J., Knuutila, J., Lounasmaa, O.V., 1993. Magnetoencephalography---theory, instrumentation, and applications to noninvasive studies of the working human brain. Rev. Mod. Phys. 65, 413–497. https://doi.org/10.1103/RevModPhys.65.413

Hammer, J., Fischer, J., Ruescher, J., Schulze-Bonhage, A., Aertsen, A., Ball, T., 2013. The role of ECoG magnitude and phase in decoding position, velocity, and acceleration during continuous motor behavior. Front. Neurosci. 7. https://doi.org/10.3389/fnins.2013.00200

Hamner, B., Leeb, R., Tavella, M., del R. Millán, J., 2011. Phase-based features for motor imagery brain-computer interfaces, in: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Presented at the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2578–2581. https://doi.org/10.1109/IEMBS.2011.6090712

Han, C.-H., Müller, K.-R., Hwang, H.-J., 2020. Brain-Switches for Asynchronous Brain–Computer Interfaces: A Systematic Review. Electronics 9, 422. https://doi.org/10.3390/electronics9030422

Hasan, B.A.S., Gan, J.Q., 2011a. Temporal modeling of EEG during self-paced hand movement and its application in onset detection. J. Neural Eng. 8, 056015. https://doi.org/10.1088/1741-2560/8/5/056015

Hasan, B.A.S., Gan, J.Q., 2011b. Conditional random fields as classifiers for three-class motor-imagery brain–computer interfaces. J. Neural Eng. 8, 025013. https://doi.org/10.1088/1741-2560/8/2/025013

Hasan, B.A.S., Gan, J.Q., 2009. Unsupervised adaptive GMM for BCI, in: 2009 4th International IEEE/EMBS Conference on Neural Engineering. Presented at the 2009 4th International IEEE/EMBS Conference on Neural Engineering, pp. 295–298. https://doi.org/10.1109/NER.2009.5109291

Haselsteiner, E., Pfurtscheller, G., 2000. Using time-dependent neural networks for EEG classification. IEEE Transactions on Rehabilitation Engineering 8, 457–463. https://doi.org/10.1109/86.895948

Hassan, M., Rabiul Islam, S.Md., 2019. Design and Implementation of Pre-processing Chip for Brain Computer Interface Machine, in: 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST). Presented at the 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST), pp. 428–433. https://doi.org/10.1109/ICREST.2019.8644230

Hastie, T., Tibshirani, R., Wainwright, M., Tibshirani, R., Wainwright, M., 2015. Statistical Learning with Sparsity : The Lasso and Generalizations. Chapman and Hall/CRC. https://doi.org/10.1201/b18401

Haufe, S., Dähne, S., Nikulin, V.V., 2014. Dimensionality reduction for the analysis of brain oscillations. NeuroImage 101, 583–597. https://doi.org/10.1016/j.neuroimage.2014.06.073

Haykin, S., John Wiley & Sons, I., 2001. The Unscented Kalman filter, in: Kalman Filtering and Neural Networks. s.n., s.l., p. 50.

Hazrati, M.Kh., Erfanian, A., 2010. An online EEG-based brain–computer interface for controlling hand grasp using an adaptive probabilistic neural network. Medical Engineering & Physics 32, 730–739. https://doi.org/10.1016/j.medengphy.2010.04.016

He, L., Zong, C., Wang, C., 2012. Driving intention recognition and behaviour prediction based on a double-layer hidden Markov model. J. Zhejiang Univ. - Sci. C 13, 208–217. https://doi.org/10.1631/jzus.C11a0195

He, Y., Eguren, D., Azorín, J.M., Grossman, R.G., Luu, T.P., Contreras-Vidal, J.L., 2018. Brain–machine interfaces for controlling lower-limb powered robotic systems. J. Neural Eng. 15, 021004. https://doi.org/10.1088/1741-2552/aaa8c0

Helland, K., Berntsen, H.E., Borgen, O.S., Martens, H., 1992. Recursive algorithm for partial least squares regression. Chemometrics and Intelligent Laboratory Systems, Proceedings of the 2nd Scandinavian Symposium on Chemometrics 14, 129–137. https://doi.org/10.1016/0169-7439(92)80098-O

Herman, P., Prasad, G., McGinnity, T.M., Coyle, D., 2008. Comparative Analysis of Spectral Approaches to Feature Extraction for EEG-Based Motor Imagery Classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering 16, 317–326. https://doi.org/10.1109/TNSRE.2008.926694

Hervás, D., Prats-Montalbán, J.M., García-Cañaveras, J.C., Lahoz, A., Ferrer, A., 2019. Sparse N-way partial least squares by L1-penalization. Chemometrics and Intelligent Laboratory Systems 185, 85–91. https://doi.org/10.1016/j.chemolab.2019.01.004

Hettiarachchi, I.T., Nguyen, T.T., Nahavandi, S., 2015. Multivariate Adaptive Autoregressive Modeling and Kalman Filtering for Motor Imagery BCI, in: 2015 IEEE International Conference on Systems, Man, and Cybernetics. Presented at the 2015 IEEE International Conference on Systems, Man, and Cybernetics, pp. 3164–3168. https://doi.org/10.1109/SMC.2015.549

Hill, N.J., Lal, T.N., Schröder, M., Hinterberger, T., Widman, G., Elger, C.E., Schölkopf, B., Birbaumer, N., 2006. Classifying Event-Related Desynchronization in EEG, ECoG and MEG Signals, in: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. [Eds.], Pattern Recognition, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 404–413. https://doi.org/10.1007/11861898_41

Ho, Y.-L., Huang, Y.-D., Wang, K.-Y., Fang, W.-C., 2019. A SOC Design of ORICA-based Highly Effective Real-time Multi-channel EEG System, in: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 664–667. https://doi.org/10.1109/EMBC.2019.8856322

Hoang, D., Williamson, G., 1998. A mixture of global and local gated experts for the prediction of high frequency foreign exchange rates, in: Lee, H.-Y., Motoda, H. [Eds.], PRICAI'98: Topics in Artificial Intelligence, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 329–340.

Hochberg, L.R., Bacher, D., Jarosiewicz, B., Masse, N.Y., Simeral, J.D., Vogel, J., Haddadin, S., Liu, J., Cash, S.S., Smagt, P. van der, Donoghue, J.P., 2012. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature 485, 372–375. https://doi.org/10.1038/nature11076

Hochberg, L.R., Serruya, M.D., Friehs, G.M., Mukand, J.A., Saleh, M., Caplan, A.H., Branner, A., Chen, D., Penn, R.D., Donoghue, J.P., 2006. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature 442, 164–171. https://doi.org/10.1038/nature04970

Hortal, E., Planelles, D., Costa, A., Iáñez, E., Úbeda, A., Azorín, J.M., Fernández, E., 2015. SVM-based Brain–Machine Interface for controlling a robot arm through four mental tasks. Neurocomputing 151, 116–121. https://doi.org/10.1016/j.neucom.2014.09.078

Hotson, G., Fifer, M.S., Acharya, S., Benz, H.L., Anderson, W.S., Thakor, N.V., Crone, N.E., 2014. Coarse Electrocorticographic Decoding of Ipsilateral Reach in Patients with Brain Lesions. PLOS ONE 9, e115236. https://doi.org/10.1371/journal.pone.0115236

Hotson, G., McMullen, D.P., Fifer, M.S., Johannes, M.S., Katyal, K.D., Para, M.P., Robert Armiger, Anderson, W.S., Thakor, N.V., Wester, B.A., Crone, N.E., 2016a. Individual finger control of a modular prosthetic limb using high-density electrocorticography in a human subject. J. Neural Eng. 13, 026017. https://doi.org/10.1088/1741-2560/13/2/026017

Hotson, G., Smith, R.J., Rouse, A.G., Schieber, M.H., Thakor, N.V., Wester, B.A., 2016b. High Precision Neural Decoding of Complex Movement Trajectories Using Recursive Bayesian Estimation With Dynamic Movement Primitives. IEEE Robotics and Automation Letters 1, 676–683. https://doi.org/10.1109/LRA.2016.2516590

Hou, H., Sun, B., Meng, Q., 2019. Slow cortical potential signal classification using concave–convex feature. Journal of Neuroscience Methods 324, 108303. https://doi.org/10.1016/j.jneumeth.2019.05.012

Hsu, S.H., Mullen, T.R., Jung, T.P., Cauwenberghs, G., 2016. Real-Time Adaptive EEG Source Separation Using Online Recursive Independent Component Analysis. IEEE Transactions on Neural Systems and Rehabilitation Engineering 24, 309–319. https://doi.org/10.1109/TNSRE.2015.2508759

Hsu, S.H., Pion-Tonachini, L., Jung, T.P., Cauwenberghs, G., 2015. Tracking non-stationary EEG sources using adaptive online recursive independent component analysis, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 4106–4109. https://doi.org/10.1109/EMBC.2015.7319297

Hsu, W.-Y., 2011. EEG-based motor imagery classification using enhanced active segment selection and adaptive classifier. Computers in Biology and Medicine 41, 633–639. https://doi.org/10.1016/j.compbiomed.2011.05.014

Hu, S., Wei, H., Chen, Y., Tan, J., 2012. A Real-Time Cardiac Arrhythmia Classification System with Wearable Sensor Networks. Sensors 12, 12844–12869. https://doi.org/10.3390/s120912844

Hu, Y.-Q., Yu, Y., Liao, J.-D., 2019. Cascaded Algorithm-Selection and Hyper-Parameter Optimization with Extreme-Region Upper Confidence Bound Bandit 2528–2534.

Huang, D., Lin, P., Fei, D., Chen, X., Bai, O., 2009. EEG-based online two-dimensional cursor control, in: 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Presented at the 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4547–4550. https://doi.org/10.1109/IEMBS.2009.5332722

Huang, D., Qian, K., Fei, D., Jia, W., Chen, X., Bai, O., 2012. Electroencephalography (EEG)-Based Brain–Computer Interface (BCI): A 2-D Virtual Wheelchair Control Based on Event-Related Desynchronization/Synchronization and State Control. IEEE Transactions on Neural Systems and Rehabilitation Engineering 20, 379–388. https://doi.org/10.1109/TNSRE.2012.2190299

Huang, J., Ma, S., Zhang, C.-H., 2008. Adaptive Lasso for sparse high-dimensional regression models. Statistica Sinica 18, 1603–1618.

Huang, Q., Zhang, Z., Yu, T., He, S., Li, Y., 2019. An EEG-/EOG-Based Hybrid Brain-Computer Interface: Application on Controlling an Integrated Wheelchair Robotic Arm System. Front. Neurosci. 13. https://doi.org/10.3389/fnins.2019.01243

Hüebner, D., Verhoeven, T., Müeller, K.-R., Kindermans, P.-J., Tangermann, M., 2018. Unsupervised Learning for Brain-Computer Interfaces Based on Event-Related Potentials: Review and Online Comparison [Research Frontier]. IEEE Computational Intelligence Magazine 13, 66–77. https://doi.org/10.1109/MCI.2018.2807039

Hutter, F., Caruana, R., Bardenet, R., Bilenko, M., Guyon, I., Kegl, B., Larochelle, H., 2014. Automatic Machine Learning: Methods, Systems, Challenges, AutoML 2014@ ICML. AutoML 2014 Workshop@ ICML.

Hutter, F., Hoos, H.H., Leyton-Brown, K., 2011a. Sequential Model-Based Optimization for General Algorithm Configuration (Extended Version), in: Coello, C.A.C. [Ed.], Learning and Intelligent Optimization. Springer Berlin Heidelberg, pp. 507–523.

Hutter, F., Hoos, H.H., Leyton-Brown, K., 2011b. Sequential Model-Based Optimization for General Algorithm Configuration, in: Coello, C.A.C. [Ed.], Learning and Intelligent Optimization. Springer Berlin Heidelberg, pp. 507–523.

Hutter, F., Hoos, H.H., Leyton-Brown, K., Stützle, T., 2009. ParamILS: An automatic algorithm configuration framework. Journal of Artificial Intelligence Research 36, 267–306. https://doi.org/10.1613/jair.2808

Huve, G., Takahashi, K., Hashimoto, M., 2019. Online Recognition of the Mental States of Drivers with an fNIRS-Based Brain-Computer Interface Using Deep Neural Network, in: 2019 IEEE International Conference on Mechatronics (ICM). Presented at the 2019 IEEE International Conference on Mechatronics (ICM), pp. 238–242. https://doi.org/10.1109/ICMECH.2019.8722936

ICTRP clinical trial NCT02550522 [WWW Document], n.d. URL https://apps.who.int/trialsearch/Trial2.aspx?TrialID=NCT02550522 [accessed 6.29.20].

Ifft, P.J., Shokur, S., Li, Z., Lebedev, M.A., Nicolelis, M.A.L., 2013. A Brain-Machine Interface Enables Bimanual Arm Movements in Monkeys. Science Translational Medicine 5, 210ra154-210ra154. https://doi.org/10.1126/scitranslmed.3006159

Iqbal, H., Aqil, M., 2016. A QR decomposition based RLS algorithm with forgetting factor for adaptation of AR EEG features, in: 2016 International Conference on Emerging Technologies (ICET). Presented at the 2016 International Conference on Emerging Technologies (ICET), pp. 1–5. https://doi.org/10.1109/ICET.2016.7813250

Iturrate, I., Antelis, J., Minguez, J., 2009. Synchronous EEG brain-actuated wheelchair with automated navigation, in: 2009 IEEE International Conference on Robotics and Automation. Presented at the 2009 IEEE International Conference on Robotics and Automation, pp. 2318–2325. https://doi.org/10.1109/ROBOT.2009.5152580

Iversen, J.R., Makeig, S., 2019. MEG/EEG Data Analysis Using EEGLAB, in: Supek, S., Aine, C.J. [Eds.], Magnetoencephalography: From Signals to Dynamic Cortical Networks. Springer International Publishing, Cham, pp. 391–406. https://doi.org/10.1007/978-3-030-00087-5_8

Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E., 1991. Adaptive Mixtures of Local Experts. Neural Comput. 3, 79–87. https://doi.org/10.1162/neco.1991.3.1.79

Jafarifarmand, A., Badamchizadeh, M.A., 2020. Real-time multiclass motor imagery brain-computer interface by modified common spatial patterns and adaptive neuro-fuzzy classifier. Biomedical Signal Processing and Control 57, 101749. https://doi.org/10.1016/j.bspc.2019.101749

Jarosiewicz, B., Masse, N.Y., Bacher, D., Cash, S.S., Eskandar, E., Friehs, G., Donoghue, J.P., Hochberg, L.R., 2013. Advantages of closed-loop calibration in intracortical brain–computer interfaces for people with tetraplegia. J. Neural Eng. 10, 046012. https://doi.org/10.1088/1741-2560/10/4/046012

Jarosiewicz, B., Sarma, A.A., Bacher, D., Masse, N.Y., Simeral, J.D., Sorice, B., Oakley, E.M., Blabe, C., Pandarinath, C., Gilja, V., Cash, S.S., Eskandar, E.N., Friehs, G., Henderson, J.M., Shenoy, K.V., Donoghue, J.P., Hochberg, L.R., 2015. Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. Science Translational Medicine 7, 313ra179-313ra179. https://doi.org/10.1126/scitranslmed.aac7328

Javed, A., Tiwana, Mohsin I., Tiwana, Moazzam I., Rashid, N., Iqbal, J., Khan, U.S., 2017. Recognition of finger movements using EEG signals for control of upper limb prosthesis using logistic regression. Biomedical Research 28, 7361–7369.

Jeffries, N., Pfeiffer, R., 2001. A mixture model for the probability distribution of rain rate. Environmetrics 12, 1–10. https://doi.org/10.1002/1099-095X(200102)12:1<1::AID-ENV425>3.0.CO;2-N

Jeong, J.-H., Lee, B.-H., Lee, D.-H., Yun, Y.-D., Lee, S.-W., 2020. EEG Classification of Forearm Movement Imagery Using a Hierarchical Flow Convolutional Neural Network. IEEE Access 8, 66941–66950. https://doi.org/10.1109/ACCESS.2020.2983182

Jerbi, K., Vidal, J.R., Mattout, J., Maby, E., Lecaignard, F., Ossandon, T., Hamamé, C.M., Dalal, S.S., Bouet, R., Lachaux, J.-P., Leahy, R.M., Baillet, S., Garnero, L., Delpuech, C., Bertrand, O., 2011. Inferring hand movement kinematics from MEG, EEG and intracranial EEG: From brain-machine interfaces to motor rehabilitation. IRBM, NUMÉRO SPÉCIAL : LE CERVEAU DANS TOUS SES ÉTATS 32, 8–18. https://doi.org/10.1016/j.irbm.2010.12.004

Jeunet, C., N'Kaoua, B., Lotte, F., 2016. Chapter 1 - Advances in user-training for mental-imagery-based BCI control: Psychological and cognitive factors and their neural correlates, in: Coyle, D. [Ed.], Progress in Brain Research, Brain-Computer Interfaces: Lab Experiments to Real-World Applications. Elsevier, pp. 3–35. https://doi.org/10.1016/bs.pbr.2016.04.002

Ji Won Yoon, Roberts, S.J., Dyson, M., Gan, J.Q., 2008. Sequential Bayesian estimation for adaptive classification, in: 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. Presented at the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 601–605. https://doi.org/10.1109/MFI.2008.4648010

Jia Wu, X.-Y.C., Jia Wu, X.-Y.C., n.d. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. Journal of Electronic Science and Technology 17, 26–40. https://doi.org/10.11989/JEST.1674-862X.80904120

Jiang, T., Jiang, T., Wang, T., Mei, S., Liu, Q., Li, Y., Wang, X., Prabhu, S., Sha, Z., Ince, N.F., 2017. Characterization and Decoding the Spatial Patterns of Hand Extension/Flexion using High-Density ECoG. IEEE Transactions on Neural Systems and Rehabilitation Engineering 25, 370–379. https://doi.org/10.1109/TNSRE.2016.2647255

Johannes, M.S., Bigelow, J.D., Burck, J.M., Harshbarger, S.D., Kozlowski, M.V., Van Doren, T., 2011. An overview of the developmental process for the modular prosthetic limb. Johns Hopkins APL Technical Digest (Applied Physics Laboratory) 30, 207–216.

Jones, D.R., Schonlau, M., Welch, W.J., 1998. Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization 13, 455–492. https://doi.org/10.1023/A:1008306431147

Jorfi, M., Skousen, J.L., Weder, C., Capadona, J.R., 2015. Progress Towards Biocompatible Intracortical Microelectrodes for Neural Interfacing Applications. J Neural Eng 12, 011001. https://doi.org/10.1088/1741-2560/12/1/011001

Jubien, G., Schaeffer, M., Bonnet, S., Aksenova, T., 2019. Decoding of finger activation from ECoG data: a comparative study, in: 2019 International Joint Conference on Neural Networks (IJCNN). Presented at the 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. https://doi.org/10.1109/IJCNN.2019.8852214

Kabir, M.H., Hoque, M.R., Thapa, K., Yang, S.-H., 2016. Two-Layer Hidden Markov Model for Human Activity Recognition in Home Environments. International Journal of Distributed Sensor Networks 12, 4560365. https://doi.org/10.1155/2016/4560365

Kalivas, J.H., 2012. Overview of two-norm (L2) and one-norm (L1) Tikhonov regularization variants for full wavelength or sparse spectral multivariate calibration models or maintenance. Journal of Chemometrics 26, 218–230. https://doi.org/10.1002/cem.2429

Kalunga, E.K., Chevallier, S., Barthélemy, Q., Djouani, K., Monacelli, E., Hamam, Y., 2016. Online SSVEP-based BCI using Riemannian geometry. Neurocomputing 191, 55–68. https://doi.org/10.1016/j.neucom.2016.01.007

Kamousi, B., Zhongming Liu, Bin He, 2005. Classification of motor imagery tasks for brain-computer interface applications by means of two equivalent dipoles analysis. IEEE Transactions on Neural Systems and Rehabilitation Engineering 13, 166–171. https://doi.org/10.1109/TNSRE.2005.847386

Kandel, M., Tollet, M., 2016. Anatomy of the Nervous System, in: Brain–Computer Interfaces 1. John Wiley & Sons, Ltd, pp. 1–24. https://doi.org/10.1002/9781119144977.ch1

Kang, X., Schieber, M.H., Thakor, N.V., 2012. Decoding of finger, hand and arm kinematics using switching linear dynamical systems with pre-motor cortical ensembles. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference 2012, 1732–1735. https://doi.org/10.1109/EMBC.2012.6346283

Kanoga, S., Kanemura, A., Asoh, H., 2018. A COMPARATIVE STUDY OF FEATURES AND CLASSIFIERS IN SINGLE-CHANNEL EEG-BASED MOTOR IMAGERY BCI, in: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP). Presented at the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 474–478. https://doi.org/10.1109/GlobalSIP.2018.8646636

Kao, J.C., Nuyujukian, P., Ryu, S.I., Shenoy, K.V., 2017. A High-Performance Neural Prosthesis Incorporating Discrete State Selection With Hidden Markov Models. IEEE Transactions on Biomedical Engineering 64, 935–945. https://doi.org/10.1109/TBME.2016.2582691

Kapeller, C., Gergondet, P., Kamada, K., Ogawa, H., Takeuchi, F., Ortner, R., Prückl, R., Kheddar, A., Scharinger, J., Guger, C., 2015. Online control of a humanoid robot through hand movement imagination using CSP and ECoG based features, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1765–1768. https://doi.org/10.1109/EMBC.2015.7318720

Kee, Y.-J., Lee, M.-H., Williamson, J., Lee, S.-W., 2017. A Hierarchical Classification Strategy for Robust Detection of Passive/Active Mental State Using User-Voluntary Pitch Imagery Task, in: 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR). Presented at the 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), pp. 906–910. https://doi.org/10.1109/ACPR.2017.133

Keerthi, S.S., Sindhwani, V., Chapelle, O., 2006. An Efficient Method for Gradient-Based Adaptation of Hyperparameters in SVM Models, in: NIPS. https://doi.org/10.7551/mitpress/7503.003.0089

Kellis, S., Hanrahan, S., Davis, T., House, P.A., Brown, R., Greger, B., 2012. Decoding hand trajectories from micro-electrocorticography in human patients, in: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Presented at the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4091–4094. https://doi.org/10.1109/EMBC.2012.6346866

Kevric, J., Subasi, A., 2017. Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system. Biomedical Signal Processing and Control 31, 398–406. https://doi.org/10.1016/j.bspc.2016.09.007

Khaire, U.M., Dhanalakshmi, R., 2019. Stability of feature selection algorithm: A review. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2019.06.012

Khan, J., Bhatti, M.H., Khan, U.G., Iqbal, R., 2019. Multiclass EEG motor-imagery classification with sub-band common spatial patterns. EURASIP Journal on Wireless Communications and Networking 2019, 174. https://doi.org/10.1186/s13638-019-1497-y

Khan, R.A., Naseer, N., Qureshi, N.K., Noori, F.M., Nazeer, H., Khan, M.U., 2018. fNIRS-based Neurorobotic Interface for gait rehabilitation. Journal of NeuroEngineering and Rehabilitation 15, 7. https://doi.org/10.1186/s12984-018-0346-2

Kheradpisheh, S.R., Sharifizadeh, F., Nowzari-Dalini, A., Ganjtabesh, M., Ebrahimpour, R., 2014. Mixture of feature specified experts. Information Fusion 20, 242–251. https://doi.org/10.1016/j.inffus.2014.02.006

Khreich, W., Granger, E., Miri, A., Sabourin, R., 2012. A survey of techniques for incremental learning of HMM parameters. Information Sciences 197, 105–130. https://doi.org/10.1016/j.ins.2012.02.017

Kilicarslan, A., Prasad, S., Grossman, R.G., Contreras-Vidal, J.L., 2013. High accuracy decoding of user intentions using EEG to control a lower-body exoskeleton, in: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 5606–5609. https://doi.org/10.1109/EMBC.2013.6610821

Kim, H., Yoshimura, N., Koike, Y., 2019. Classification of Movement Intention Using Independent Components of Premovement EEG. Front. Hum. Neurosci. 13. https://doi.org/10.3389/fnhum.2019.00063

Kim, H.J., Ollila, E., Koivunen, V., Croux, C., 2013. Robust and sparse estimation of tensor decompositions, in: 2013 IEEE Global Conference on Signal and Information Processing. Presented at the 2013 IEEE Global Conference on Signal and Information Processing, pp. 965–968. https://doi.org/10.1109/GlobalSIP.2013.6737053

Kim, H.J., Ollila, E., Koivunen, V., Poor, H.V., 2014. Robust iteratively reweighted Lasso for sparse tensor factorizations, in: 2014 IEEE Workshop on Statistical Signal Processing (SSP). Presented at the 2014 IEEE Workshop on Statistical Signal Processing (SSP), pp. 420–423. https://doi.org/10.1109/SSP.2014.6884665

Kim, K.H., Kim, S.S., Kim, S.J., 2006. Superiority of nonlinear mapping in decoding multiple single-unit neuronal spike trains: A simulation study. Journal of Neuroscience Methods 150, 202–211. https://doi.org/10.1016/j.jneumeth.2005.06.015

Kim, S., Simeral, J.D., Hochberg, L.R., Donoghue, J.P., Friehs, G.M., Black, M.J., 2011. Point-and-Click Cursor Control With an Intracortical Neural Interface System by Humans With Tetraplegia. IEEE Transactions on Neural Systems and Rehabilitation Engineering 19, 193–203. https://doi.org/10.1109/TNSRE.2011.2107750

Kim, S., White, A., Scalzo, F., Collier, D., 2018. Elastic net ensemble classifier for event-related potential based automatic spelling. Biomedical Signal Processing and Control 46, 166–173. https://doi.org/10.1016/j.bspc.2018.06.005

Kim, S.-P., Sanchez, J.C., Erdogmus, D., Rao, Y.N., Wessberg, J., Principe, J.C., Nicolelis, M., 2003. Divide-and-conquer approach for brain machine interfaces: nonlinear mixture of competitive linear models. Neural Netw 16, 865–871. https://doi.org/10.1016/S0893-6080(03)00108-4

Kim, S.P., Sanchez, J.C., Rao, Y.N., Erdogmus, D., Carmena, J.M., Lebedev, M.A., Nicolelis, M. a. L., Principe, J.C., 2006. A comparison of optimal MIMO linear and nonlinear models for brain-machine interfaces. J Neural Eng 3, 145–161. https://doi.org/10.1088/1741-2560/3/2/009

Kim, S.-P., Simeral, J.D., Hochberg, L.R., Donoghue, J.P., Black, M.J., 2008. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. J. Neural Eng. 5, 455–476. https://doi.org/10.1088/1741-2560/5/4/010

Kim, Y.J., Park, S.W., Yeom, H.G., Bang, M.S., Kim, J.S., Chung, C.K., Kim, S., 2015. A study on a robot arm driven by three-dimensional trajectories predicted from non-invasive neural signals. BioMed Eng OnLine 14, 81. https://doi.org/10.1186/s12938-015-0075-8

Kindermans, P.-J., Schreuder, M., Schrauwen, B., Müller, K.-R., Tangermann, M., 2014. True Zero-Training Brain-Computer Interfacing – An Online Study. PLOS ONE 9, e102504. https://doi.org/10.1371/journal.pone.0102504

King, C.E., Dave, K.R., Wang, P.T., Mizuta, M., Reinkensmeyer, D.J., Do, A.H., Moromugi, S., Nenadic, Z., 2014. Performance Assessment of a Brain–Computer Interface Driven Hand Orthosis. Ann Biomed Eng 42, 2095–2105. https://doi.org/10.1007/s10439-014-1066-9

King, C.E., Wang, P.T., McCrimmon, C.M., Chou, C.C., Do, A.H., Nenadic, Z., 2015. The feasibility of a brain-computer interface functional electrical stimulation system for the restoration of overground walking after paraplegia. Journal of NeuroEngineering and Rehabilitation 12, 80. https://doi.org/10.1186/s12984-015-0068-7

Klein, A., Falkner, S., Bartels, S., Hennig, P., Hutter, F., 2017. Fast Bayesian hyperparameter optimization on large datasets. Electron. J. Statist. 11, 4945–4968. https://doi.org/10.1214/17-EJS1335SI

Kolda, T., Bader, B., 2009. Tensor Decompositions and Applications. SIAM Rev. 51, 455–500. https://doi.org/10.1137/07070111X

Koyama, S., Chase, S.M., Whitford, A.S., Velliste, M., Schwartz, A.B., Kass, R.E., 2010. Comparison of brain–computer interface decoding algorithms in open-loop and closed-loop control. J Comput Neurosci 29, 73–87. https://doi.org/10.1007/s10827-009-0196-9

Kozai, T.D.Y., Jaquins-Gerstl, A.S., Vazquez, A.L., Michael, A.C., Cui, X.T., 2015. Brain Tissue Responses to Neural Implants Impact Signal Sensitivity and Intervention Strategies. ACS Chem. Neurosci. 6, 48–67. https://doi.org/10.1021/cn500256e

Krauss, P., Metzner, C., Schilling, A., Tziridis, K., Traxdorf, M., Wollbrink, A., Rampp, S., Pantev, C., Schulze, H., 2018. A statistical method for analyzing and comparing spatiotemporal cortical activation patterns. Scientific Reports 8, 5433. https://doi.org/10.1038/s41598-018-23765-w

Krusienski, D.J., McFarland, D.J., Wolpaw, J.R., 2012. Value of amplitude, phase, and coherence features for a sensorimotor rhythm-based brain–computer interface. Brain Research Bulletin 87, 130–134. https://doi.org/10.1016/j.brainresbull.2011.09.019

Kubánek, J., Miller, K.J., Ojemann, J.G., Wolpaw, J.R., Schalk, G., 2009. Decoding flexion of individual fingers using electrocorticographic signals in humans. J. Neural Eng. 6, 066001. https://doi.org/10.1088/1741-2560/6/6/066001

Kuhn, M., Johnson, K., 2013. Applied Predictive Modeling, 1st ed. 2013, Corr. 2nd printing 2018 edition. ed. Springer, New York.

Kulić, D., Nakamura, Y., 2010. Incremental learning of human behaviors using hierarchical hidden Markov models, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. Presented at the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4649–4655. https://doi.org/10.1109/IROS.2010.5650813

Kumar, S., Reddy, T., Behera, L., 2018. EEG Based Motor Imagery Classification Using Instantaneous Phase Difference Sequence, in: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Presented at the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 499–504. https://doi.org/10.1109/SMC.2018.00094

Kumar, S., Sharma, A., Tsunoda, T., 2017. An improved discriminative filter bank selection approach for motor imagery EEG signal classification using mutual information. BMC Bioinformatics 18, 545. https://doi.org/10.1186/s12859-017-1964-6

Kumar, S., Yger, F., Lotte, F., 2019. Towards Adaptive Classification using Riemannian Geometry approaches in Brain-Computer Interfaces, in: 2019 7th International Winter Conference on Brain-Computer Interface (BCI). Presented at the 2019 7th International Winter Conference on Brain-Computer Interface (BCI), pp. 1–6. https://doi.org/10.1109/IWW-BCI.2019.8737349

Kwak, N.-S., Müller, K.-R., Lee, S.-W., 2015. A lower limb exoskeleton control system based on steady state visual evoked potentials. J. Neural Eng. 12, 056009. https://doi.org/10.1088/1741-2560/12/5/056009

LaFleur, K., Cassady, K., Doud, A., Shades, K., Rogin, E., He, B., 2013. Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface. J. Neural Eng. 10, 046003. https://doi.org/10.1088/1741-2560/10/4/046003

Langhorne, P., Bernhardt, J., Kwakkel, G., 2011. Stroke rehabilitation. The Lancet 377, 1693–1702. https://doi.org/10.1016/S0140-6736(11)60325-5

Lazarou, I., Nikolopoulos, S., Petrantonakis, P.C., Kompatsiaris, I., Tsolaki, M., 2018. EEG-Based Brain–Computer Interfaces for Communication and Rehabilitation of People with Motor Impairment: A Novel Approach of the 21st Century. Front. Hum. Neurosci. 12. https://doi.org/10.3389/fnhum.2018.00014

Lê Cao, K.-A., Meugnier, E., McLachlan, G.J., 2010. Integrative mixture of experts to combine clinical factors and gene markers. Bioinformatics 26, 1192–1198. https://doi.org/10.1093/bioinformatics/btq107

Lebedev, M.A., Nicolelis, M.A.L., 2017. Brain-Machine Interfaces: From Basic Science to Neuroprostheses and Neurorehabilitation. Physiological Reviews 97, 767–837. https://doi.org/10.1152/physrev.00027.2016

Lebedev, M.A., Nicolelis, M.A.L., 2011. Chapter 3 - Toward a whole-body neuroprosthetic, in: Schouenborg, J., Garwicz, M., Danielsen, N. [Eds.], Progress in Brain Research, Brain Machine Interfaces: Implications for Science, Clinical Practice and Society. Elsevier, pp. 47–60. https://doi.org/10.1016/B978-0-444-53815-4.00018-2

Lebedev, M.A., Nicolelis, M.A.L., 2006. Brain–machine interfaces: past, present and future. Trends in Neurosciences 29, 536–546. https://doi.org/10.1016/j.tins.2006.07.004

Lee, J.J., Kim, J.H., Nakajima, M., 1998. A Hierarchical HMM Network-Based Approach for On-Line Recognition of Multi-Lingual Cursive Handwritings. IEICE TRANSACTIONS on Information and Systems E81-D, 881–888.

Lee, M.B., Kramer, D.R., Peng, T., Barbaro, M.F., Liu, C.Y., Kellis, S., Lee, B., 2019. Clinical neuroprosthetics: Today and tomorrow. Journal of Clinical Neuroscience 68, 13–19. https://doi.org/10.1016/j.jocn.2019.07.056

Lee, Y.-S., Cho, S.-B., 2016. Layered hidden Markov models to recognize activity with built-in sensors on Android smartphone. Pattern Anal Applic 19, 1181–1193. https://doi.org/10.1007/s10044-016-0549-8

Leeb, R., Friedman, D., Müller-Putz, G.R., Scherer, R., Slater, M., Pfurtscheller, G., 2007. Self-Paced (Asynchronous) BCI Control of a Wheelchair in Virtual Environments: A Case Study with a Tetraplegic. Computational Intelligence and Neuroscience. https://doi.org/10.1155/2007/79642

Lehtonen, J., Jylanki, P., Kauhanen, L., Sams, M., 2008. Online Classification of Single EEG Trials During Finger Movements. IEEE Transactions on Biomedical Engineering 55, 713–720. https://doi.org/10.1109/TBME.2007.912653

Leuthardt, E.C., Miller, K.J., Schalk, G., Rao, R.P.N., Ojemann, J.G., 2006. Electrocorticography-based brain computer Interface-the seattle experience. IEEE Transactions on Neural Systems and Rehabilitation Engineering 14, 194–198. https://doi.org/10.1109/TNSRE.2006.875536

Leuthardt, E.C., Schalk, G., Wolpaw, J.R., Ojemann, J.G., Moran, D.W., 2004. A brain–computer interface using electrocorticographic signals in humans. J. Neural Eng. 1, 63. https://doi.org/10.1088/1741-2560/1/2/001

Li, D., Zhang, H., Khan, M.S., Mi, F., 2018. A self-adaptive frequency selection common spatial pattern and least squares twin support vector machine for motor imagery electroencephalography recognition. Biomedical Signal Processing and Control 41, 222–232. https://doi.org/10.1016/j.bspc.2017.11.014

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A., 2018. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. Journal of Machine Learning Research 18, 1–52.

Li, Y., Koike, Y., Sugiyama, M., 2009. A Framework of Adaptive Brain Computer Interfaces, in: 2009 2nd International Conference on Biomedical Engineering and Informatics. Presented at the 2009 2nd International Conference on Biomedical Engineering and Informatics, pp. 1–5. https://doi.org/10.1109/BMEI.2009.5305646

Li, Y., Pan, J., Wang, F., Yu, Z., 2013. A Hybrid BCI System Combining P300 and SSVEP and Its Application to Wheelchair Control. IEEE Transactions on Biomedical Engineering 60, 3156–3166. https://doi.org/10.1109/TBME.2013.2270283

Li, Z., O'Doherty, J.E., Hanson, T.L., Levedev, M.A., Henriquez, C.S., Nicolelis, M.A.L., 2009. Unscented Kalman filter for brain-machine interfaces. PLoS ONE 4. https://doi.org/10.1371/journal.pone.0006243

Li, Z., O'Doherty, J.E., Lebedev, M.A., Nicolelis, M.A.L., 2011. Adaptive Decoding for Brain-Machine Interfaces Through Bayesian Parameter Updates. Neural Computation 23, 3162–3204. https://doi.org/10.1162/NECO_a_00207

Liang, W., Zhang, Y., Tan, J., Li, Y., 2014. A Novel Approach to ECG Classification Based upon Two-Layered HMMs in Body Sensor Networks. Sensors 14, 5994–6011. https://doi.org/10.3390/s140405994

Liehr, S., Pawelzik, K., Kohlmorgen, J., Müller, K.-R., 1999. Hidden Markov mixtures of experts with an application to EEG recordings from sleep. Theory in Biosciences 118, 246–260.

Lin, Z., Zhang, C., Zeng, Y., Tong, L., Yan, B., 2018. A novel P300 BCI speller based on the Triple RSVP paradigm. Scientific Reports 8, 3350. https://doi.org/10.1038/s41598-018-21717-y

Lindgren, F., Geladi, P., Wold, S., 1993. The kernel algorithm for PLS. J. Chemometrics 7, 45–59. https://doi.org/10.1002/cem.1180070104

Lisi, G., Rivela, D., Takai, A., Morimoto, J., 2018. Markov Switching Model for Quick Detection of Event Related Desynchronization in EEG. Front. Neurosci. 12. https://doi.org/10.3389/fnins.2018.00024

Long, J., Gu, Z., Li, Y., Yu, T., Li, F., Fu, M., 2011. Semi-supervised joint spatio-temporal feature selection for P300-based BCI speller. Cogn Neurodyn 5, 387. https://doi.org/10.1007/s11571-011-9167-8

Lopes da Silva, F., 2013. EEG and MEG: Relevance to Neuroscience. Neuron 80, 1112–1128. https://doi.org/10.1016/j.neuron.2013.10.017

López-Larraz, E., Montesano, L., Gil-Agudo, Á., Minguez, J., 2014. Continuous decoding of movement intention of upper limb self-initiated analytic movements from pre-movement EEG correlates. Journal of NeuroEngineering and Rehabilitation 11, 153. https://doi.org/10.1186/1743-0003-11-153

López-Larraz, E., Sarasola-Sanz, A., Irastorza-Landa, N., Birbaumer, N., Ramos-Murguialday, A., 2018. Brain-machine interfaces for rehabilitation in stroke: A review. NeuroRehabilitation 43, 77–97. https://doi.org/10.3233/NRE-172394

López-Larraz, E., Trincado-Alonso, F., Rajasekaran, V., Pérez-Nombela, S., del-Ama, A.J., Aranda, J., Minguez, J., Gil-Agudo, A., Montesano, L., 2016. Control of an Ambulatory Exoskeleton with a Brain–Machine Interface for Spinal Cord Injury Gait Rehabilitation. Front Neurosci 10. https://doi.org/10.3389/fnins.2016.00359

Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., Yger, F., 2018. A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. J. Neural Eng. 15, 031005. https://doi.org/10.1088/1741-2552/aab2f2

Lotte, F., Congedo, M., 2016. EEG Feature Extraction, in: Brain–Computer Interfaces 1. John Wiley & Sons, Ltd, pp. 127–143. https://doi.org/10.1002/9781119144977.ch7

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B., 2007. A review of classification algorithms for EEG-based brain–computer interfaces. J. Neural Eng. 4, R1–R13. https://doi.org/10.1088/1741-2560/4/2/R01

Lotte, F., Guan, C., 2011. Regularizing Common Spatial Patterns to Improve BCI Designs: Unified Theory and New Algorithms. IEEE Transactions on Biomedical Engineering 58, 355–362. https://doi.org/10.1109/TBME.2010.2082539

Ludwig, K.A., Miriani, R.M., Langhals, N.B., Marzullo, T.C., Kipke, D.R., 2011. Use of a Bayesian maximum-likelihood classifier to generate training data for brain–machine interfaces. J. Neural Eng. 8, 046009. https://doi.org/10.1088/1741-2560/8/4/046009

Ma, X., Ma, C., Huang, J., Zhang, P., Xu, J., He, J., 2017. Decoding Lower Limb Muscle Activity and Kinematics from Cortical Neural Spike Trains during Monkey Performing Stand and Squat Movements. Frontiers in Neuroscience 11. https://doi.org/10.3389/fnins.2017.00044

Ma, Z., Cheng, J., Tao, D., 2020. Online learning using projections onto shrinkage closed balls for adaptive brain-computer interface. Pattern Recognition 97, 107017. https://doi.org/10.1016/j.patcog.2019.107017

Mak, J.N., Wolpaw, J.R., 2009. Clinical Applications of Brain-Computer Interfaces: Current State and Future Prospects. IEEE Reviews in Biomedical Engineering 2, 187–199. https://doi.org/10.1109/RBME.2009.2035356

Maleki, M., Manshouri, N., Kayikçioğlu, T., 2018. Fast and accurate classifier-based brain-computer interface system using single channel EEG data, in: 2018 26th Signal Processing and Communications

Applications Conference (SIU). Presented at the 2018 26th Signal Processing and Communications Applications Conference (SIU), pp. 1–4. https://doi.org/10.1109/SIU.2018.8404376

Malešević, N., Marković, D., Kanitz, G., Controzzi, M., Cipriani, C., Antfolk, C., 2017. Decoding of individual finger movements from surface EMG signals using vector autoregressive hierarchical hidden Markov models (VARHHMM), in: 2017 International Conference on Rehabilitation Robotics (ICORR). Presented at the 2017 International Conference on Rehabilitation Robotics (ICORR), pp. 1518–1523. https://doi.org/10.1109/ICORR.2017.8009463

Marathe, A.R., Taylor, D.M., 2015. The impact of command signal power distribution, processing delays, and speed scaling on neurally-controlled devices. J. Neural Eng. 12, 046031. https://doi.org/10.1088/1741-2560/12/4/046031

Marathe, A.R., Taylor, D.M., 2013. Decoding continuous limb movements from high-density epidural electrode arrays using custom spatial filters. J. Neural Eng. 10, 036015. https://doi.org/10.1088/1741-2560/10/3/036015

Marathe, A.R., Taylor, D.M., 2011. Decoding position, velocity, or goal: Does it matter for brain–machine interfaces? J. Neural Eng. 8, 025016. https://doi.org/10.1088/1741-2560/8/2/025016

Marin, C., Fernandez, E., 2010. Biocompatibility of intracortical microelectrodes: current status and future prospects. Front. Neuroeng. 3. https://doi.org/10.3389/fneng.2010.00008

Martel, F., Dupuy, T., Moly, A., Chabardès, S., Aksenova, T., 2020. Evaluation criteria for closed-loop adaptive dynamic discrete-continuous brain-computer interfaces: clinical study case with tetraplegic patient., in: 2020 International Joint Conference on Neural Networks (IJCNN). Presented at the 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–10. https://doi.org/10.1109/IJCNN48605.2020.9207243

Martínez-Montes, E., Sánchez-Bornot, J.M., Valdés-Sosa, P.A., 2008. PENALIZED PARAFAC ANALYSIS OF SPONTANEOUS EEG RECORDINGS. Statistica Sinica 18, 1449–1464.

Mason, S.G., Birch, G.E., 2000. A brain-controlled switch for asynchronous control applications. IEEE Transactions on Biomedical Engineering 47, 1297–1307. https://doi.org/10.1109/10.871402

Mattia, D., Pichiorri, F., Molinari, M., Rupp, R., 2013. Brain Computer Interface for Hand Motor Function Restoration and Rehabilitation, in: Allison, B.Z., Dunne, S., Leeb, R., Del R. Millán, J., Nijholt, A. [Eds.], Towards Practical Brain-Computer Interfaces: Bridging the Gap from Research to Real-World Applications, Biological and Medical Physics, Biomedical Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 131–153. https://doi.org/10.1007/978-3-642-29746-5_7

McCrimmon, C.M., Wang, P.T., Heydari, P., Nguyen, A., Shaw, S.J., Gong, H., Chui, L.A., Liu, C.Y., Nenadic, Z., Do, A.H., 2018. Electrocorticographic Encoding of Human Gait in the Leg Primary Motor Cortex. Cereb Cortex 28, 2752–2762. https://doi.org/10.1093/cercor/bhx155

McFarland, D.J., 2015. The Advantages of the Surface Laplacian in Brain-Computer Interface Research. Int J Psychophysiol 97, 271–276. https://doi.org/10.1016/j.ijpsycho.2014.07.009

McFarland, D.J., McCane, L.M., David, S.V., Wolpaw, J.R., 1997. Spatial filter selection for EEG-based communication. Electroencephalography and Clinical Neurophysiology 103, 386–394. https://doi.org/10.1016/S0013-4694(97)00022-2

McFarland, D.J., Sarnacki, W.A., Wolpaw, J.R., 2010. ELECTROENCEPHALOGRAPHIC (EEG) CONTROL OF THREE-DIMENSIONAL MOVEMENT. J Neural Eng 7, 036007. https://doi.org/10.1088/1741-2560/7/3/036007

Mehring, C., Nawrot, M.P., de Oliveira, S.C., Vaadia, E., Schulze-Bonhage, A., Aertsen, A., Ball, T., 2004. Comparing information about arm movement direction in single channels of local and epicortical field potentials from monkey and human motor cortex. Journal of Physiology-Paris, Decoding and interfacing the brain: from neuronal assemblies to cyborgs 98, 498–506. https://doi.org/10.1016/j.jphysparis.2005.09.016

Mehring, C., Rickert, J., Vaadia, E., de Oliveira, S.C., Aertsen, A., Rotter, S., 2003. Inference of hand movements from local field potentials in monkey motor cortex. Nature Neuroscience 6, 1253–1254. https://doi.org/10.1038/nn1158

Mend, M., Kullmann, W.H., 2012. Human computer interface with online brute force feature selection. Biomedical Engineering / Biomedizinische Technik 57, 659–662. https://doi.org/10.1515/bmt-2012-4082

Meng, J., Zhang, S., Bekyo, A., Olsoe, J., Baxter, B., He, B., 2016. Noninvasive Electroencephalogram Based Control of a Robotic Arm for Reach and Grasp Tasks. Scientific Reports 6, 1–15. https://doi.org/10.1038/srep38565

Mestais, C.S., Charvet, G., Sauter-Starace, F., Foerster, M., Ratel, D., Benabid, A.L., 2015. WIMAGINE: Wireless 64-Channel ECoG Recording Implant for Long Term Clinical Applications. IEEE Transactions on Neural Systems and Rehabilitation Engineering 23, 10–21. https://doi.org/10.1109/TNSRE.2014.2333541

Milekovic, T., Ball, T., Schulze-Bonhage, A., Aertsen, A., Mehring, C., 2013. Detection of Error Related Neuronal Responses Recorded by Electrocorticography in Humans during Continuous Movements. PLOS ONE 8, e55235. https://doi.org/10.1371/journal.pone.0055235

Milekovic, T., Sarma, A.A., Bacher, D., Simeral, J.D., Saab, J., Pandarinath, C., Sorice, B.L., Blabe, C., Oakley, E.M., Tringale, K.R., Eskandar, E., Cash, S.S., Henderson, J.M., Shenoy, K.V., Donoghue, J.P., Hochberg, L.R., 2018. Stable long-term BCI-enabled communication in ALS and locked-in syndrome using LFP signals. Journal of Neurophysiology 120, 343–360. https://doi.org/10.1152/jn.00493.2017

Millan, Jdel.R., 2004. On the need for on-line learning in brain-computer interfaces, in: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541). Presented at the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), pp. 2877–2882 vol.4. https://doi.org/10.1109/IJCNN.2004.1381116

Millan, Jd.R., Renkens, F., Mourino, J., Gerstner, W., 2004. Noninvasive brain-actuated control of a mobile robot by human EEG. IEEE Transactions on Biomedical Engineering 51, 1026–1033. https://doi.org/10.1109/TBME.2004.827086

Miller, K.J., Zanos, S., Fetz, E.E., Nijs, M. den, Ojemann, J.G., 2009. Decoupling the Cortical Power Spectrum Reveals Real-Time Representation of Individual Finger Movements in Humans. J. Neurosci. 29, 3132–3137. https://doi.org/10.1523/JNEUROSCI.5506-08.2009

Mishra, P.K., Jagadish, B., Kiran, M.P.R.S., Rajalakshmi, P., Reddy, D.S., 2018. A Novel Classification for EEG Based Four Class Motor Imagery Using Kullback-Leibler Regularized Riemannian Manifold, in: 2018 IEEE 20th International Conference on E-Health Networking, Applications and Services (Healthcom). Presented at the 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 1–5. https://doi.org/10.1109/HealthCom.2018.8531086

Mladenović, J., Mattout, J., Lotte, F., 2017. A generic framework for adaptive EEG-based BCI training and operation. CRC Press: Taylor & Francis Group.

Mobaien, A., Boostani, R., 2016. ACSP: Adaptive CSP filter for BCI applications, in: 2016 24th Iranian Conference on Electrical Engineering (ICEE). Presented at the 2016 24th Iranian Conference on Electrical Engineering (ICEE), pp. 466–471. https://doi.org/10.1109/IranianCEE.2016.7585567

Mollazadeh, M., Aggarwal, V., Thakor, N.V., Schieber, M.H., 2014. Principal components of hand kinematics and neurophysiological signals in motor cortex during reach to grasp movements. Journal of Neurophysiology 112, 1857–1870. https://doi.org/10.1152/jn.00481.2013

Moly, A., Aksenov, A., Aksenova, T., 2020. Online adaptive group-wise sparse NPLS for ECoG neural signal decodin. arXiv.

Mongillo, G., Deneve, S., 2008. Online Learning with Hidden Markov Models. Neural Computation 20, 1706–1716. https://doi.org/10.1162/neco.2008.10-06-351

Morales-Flores, E., Schalk, G., Ramirez-Cortes, J.M., 2014. Non-supervised technique to adapt spatial filters for ECoG data analysis, in: 2014 IEEE Symposium on Computational Intelligence in Brain Computer Interfaces (CIBCI). Presented at the 2014 IEEE Symposium on Computational Intelligence in Brain Computer Interfaces (CIBCI), pp. 43–48. https://doi.org/10.1109/CIBCI.2014.7007791

Moran, D., 2010. Evolution of brain–computer interface: action potentials, local field potentials and electrocorticograms. Current Opinion in Neurobiology, Motor systems – Neurobiology of behaviourv 20, 741–745. https://doi.org/10.1016/j.conb.2010.09.010

Morinière, B., Verney, A., Abroug, N., Garrec, P., Perrot, Y., 2015. EMY: a dual arm exoskeleton dedicated to the evaluation of Brain Machine Interface in clinical trials, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Presented at the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5333–5338. https://doi.org/10.1109/IROS.2015.7354130

Moro, R., Berger, P., Bielikova, M., 2017. Towards adaptive brain-computer interfaces: Improving accuracy of detection of event-related potentials, in: 2017 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP). Presented at the 2017 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), pp. 34–39. https://doi.org/10.1109/SMAP.2017.8022664

Moses, D.A., Leonard, M.K., Chang, E.F., 2018. Real-time classification of auditory sentences using evoked cortical activity in humans. J. Neural Eng. 15, 036005. https://doi.org/10.1088/1741-2552/aaab6f

Motrenko, A., Strijov, V., 2018. Multi-way feature selection for ECoG-based Brain-Computer Interface. Expert Systems with Applications 114, 402–413. https://doi.org/10.1016/j.eswa.2018.06.054

Mudgal, S.K., Sharma, S.K., Chaturvedi, J., Sharma, A., 2020. Brain computer interface advancement in neurosciences: Applications and issues. Interdisciplinary Neurosurgery 20, 100694. https://doi.org/10.1016/j.inat.2020.100694

Müller, S.M.T., Diez, P.F., Bastos-Filho, T.F., Sarcinelli-Filho, M., Mut, V., Laciar, E., Avila, E., 2015. Robotic Wheelchair Commanded by People with Disabilities Using Low/High-Frequency SSVEP-based BCI, in: Jaffray, D.A. [Ed.], World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada, IFMBE Proceedings. Springer International Publishing, pp. 1177–1180.

Müller-Putz, G.R., 2020. Chapter 18 - Electroencephalography, in: Ramsey, N.F., Millán, J. del R. [Eds.], Handbook of Clinical Neurology, Brain-Computer Interfaces. Elsevier, pp. 249–262. https://doi.org/10.1016/B978-0-444-63934-9.00018-4

Müller-Putz, G.R., Scherer, R., Pfurtscheller, G., Rupp, R., 2006. Brain-computer interfaces for control of neuroprostheses: from synchronous to asynchronous mode of operation / Brain-Computer Interfaces zur Steuerung von Neuroprothesen: von der synchronen zur asynchronen Funktionsweise. Biomedizinische Technik 51, 57–63. https://doi.org/10.1515/BMT.2006.011

Muñoz-Romero, S., Arenas-García, J., Gómez-Verdejo, V., 2015. Sparse and kernel OPLS feature extraction based on eigenvalue problem solving. Pattern Recognition 48, 1797–1811. https://doi.org/10.1016/j.patcog.2014.12.002

Murguialday, A.R., Aggarwal, V., Chatterjee, A., Cho, Y., Rasmussen, R., O'Rourke, B., Acharya, S., Thakor, N.V., 2007. Brain-Computer Interface for a Prosthetic Hand Using Local Machine Control and Haptic Feedback, in: 2007 IEEE 10th International Conference on Rehabilitation Robotics. Presented at the 2007 IEEE 10th International Conference on Rehabilitation Robotics, pp. 609–613. https://doi.org/10.1109/ICORR.2007.4428487

Murphy, M.D., Guggenmos, D.J., Bundy, D.T., Nudo, R.J., 2016. Current Challenges Facing the Translation of Brain Computer Interfaces from Preclinical Trials to Use in Human Patients. Front. Cell. Neurosci. 9. https://doi.org/10.3389/fncel.2015.00497

Mutasim, A.K., Tipu, R.S., Bashar, M.R., Islam, Md.K., Amin, M.A., 2018. Computational Intelligence for Pattern Recognition in EEG Signals, in: Pedrycz, W., Chen, S.-M. [Eds.], Computational Intelligence for Pattern Recognition, Studies in Computational Intelligence. Springer International Publishing, Cham, pp. 291–320. https://doi.org/10.1007/978-3-319-89629-8_11

Nagel, S., Spüler, M., 2019. Asynchronous non-invasive high-speed BCI speller with robust non-control state detection. Scientific Reports 9, 1–9. https://doi.org/10.1038/s41598-019-44645-x

Naik, G.R., 2011. An overview of independent component analysis and its applications. Informatica (Ljubljana) 35, 63–81.

Nakanishi, Y., Yanagisawa, T., Shin, D., Fukuma, R., Chen, C., Kambara, H., Yoshimura, N., Hirata, M., Yoshimine, T., Koike, Y., 2013. Prediction of Three-Dimensional Arm Trajectories Based on ECoG Signals Recorded from Human Sensorimotor Cortex. PLOS ONE 8, e72085. https://doi.org/10.1371/journal.pone.0072085

Nakanishi, Y., Yanagisawa, T., Shin, D., Kambara, H., Yoshimura, N., Tanaka, M., Fukuma, R., Kishima, H., Hirata, M., Koike, Y., 2017. Mapping ECoG channel contributions to trajectory and muscle activity prediction in human sensorimotor cortex. Scientific Reports 7, 1–13. https://doi.org/10.1038/srep45486

Naseer, N., Qureshi, N.K., Noori, F.M., Hong, K.-S., 2016a. Analysis of Different Classification Techniques for Two-Class Functional Near-Infrared Spectroscopy-Based Brain-Computer Interface. Computational Intelligence and Neuroscience. https://doi.org/10.1155/2016/5480760

Naseer, N., Qureshi, N.K., Noori, F.M., Hong, K.-S., 2016b. Analysis of Different Classification Techniques for Two-Class Functional Near-Infrared Spectroscopy-Based Brain-Computer Interface [WWW Document]. Computational Intelligence and Neuroscience. https://doi.org/10.1155/2016/5480760

Neuron, 2020. . Wikipedia.

Nguyen, C.H., Karavas, G.K., Artemiadis, P., 2019. Adaptive multi-degree of freedom Brain Computer Interface using online feedback: Towards novel methods and metrics of mutual adaptation between humans and machines for BCI. PLOS ONE 14, e0212620. https://doi.org/10.1371/journal.pone.0212620

Nguyen, N.T., Phung, D.Q., Venkatesh, S., Bui, H., 2005. Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Presented at the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp. 955–960 vol. 2. https://doi.org/10.1109/CVPR.2005.203

Nicolas-Alonso, L.F., Corralejo, R., Gomez-Pilar, J., Álvarez, D., Hornero, R., 2015. Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces. Neurocomputing 159, 186–196. https://doi.org/10.1016/j.neucom.2015.02.005

Nicolas-Alonso, L.F., Gomez-Gil, J., 2012. Brain computer interfaces, a review. Sensors (Basel) 12, 1211–1279. https://doi.org/10.3390/s120201211

Nordhausen, K., Oja, H., 2018. Independent component analysis: A statistical perspective. WIREs Computational Statistics 10, e1440. https://doi.org/10.1002/wics.1440

Nurse, E.S., John, S.E., Freestone, D.R., Oxley, T.J., Ung, H., Berkovic, S.F., O'Brien, T.J., Cook, M.J., Grayden, D.B., 2018. Consistency of Long-Term Subdural Electrocorticography in Humans. IEEE Transactions on Biomedical Engineering 65, 344–352. https://doi.org/10.1109/TBME.2017.2768442

Obermaier, B., Guger, C., Neuper, C., Pfurtscheller, G., 2001. Hidden Markov models for online classification of single trial EEG data. Pattern Recognition Letters, Selected Papers from the 11th Portuguese Conference on Pattern Recognition - RECPAD2000 22, 1299–1309. https://doi.org/10.1016/S0167-8655(01)00075-7

Oganesyan, V.V., Agapov, S.N., Bulanov, V.A., Biryukova, E.V., 2018. Comparison of Results Obtained Using Brain–Computer Interface Classifiers in a Motor Imagery Recognition Task. Neurosci Behav Physi 48, 1164–1168. https://doi.org/10.1007/s11055-018-0681-6

Olcay, B.O., Karaçalı, B., 2019. Evaluation of synchronization measures for capturing the lagged synchronization between EEG channels: A cognitive task recognition approach. Computers in Biology and Medicine 114, 103441. https://doi.org/10.1016/j.compbiomed.2019.103441

Oliveira, S.C. de, Gribova, A., Donchin, O., Bergman, H., Vaadia, E., 2001. Neural interactions between motor cortical hemispheres during bimanual and unimanual arm movements. European Journal of Neuroscience 14, 1881–1896. https://doi.org/10.1046/j.0953-816x.2001.01801.x

Oliver, G., Sunehag, P., Gedeon, T., 2013. Online feature selection for Brain Computer Interfaces, in: 2013 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB). Presented at the 2013 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), pp. 122–129. https://doi.org/10.1109/CCMB.2013.6609175

Oliver, G., Sunehag, P., Gedeon, T., 2012. Asynchronous Brain Computer Interface using Hidden Semi-Markov Models, in: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Presented at the 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2728–2731. https://doi.org/10.1109/EMBC.2012.6346528

Omedes, J., Schwarz, A., Montesano, L., Müller-Putz, G., 2017. Hierarchical decoding of grasping commands from EEG, in: 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2085–2088. https://doi.org/10.1109/EMBC.2017.8037264

Onaran, I., Ince, N.F., Cetin, A.E., 2011a. Classification of multichannel ECoG related to individual finger movements with redundant spatial projections, in: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Presented at the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5424–5427. https://doi.org/10.1109/IEMBS.2011.6091341

Onaran, I., Ince, N.F., Cetin, A.E., Abosch, A., 2011b. A hybrid SVM/HMM based system for the state detection of individual finger movements from multichannel ECoG signals, in: 2011 5th International IEEE/EMBS Conference on Neural Engineering. Presented at the 2011 5th International IEEE/EMBS Conference on Neural Engineering, pp. 457–460. https://doi.org/10.1109/NER.2011.5910585

Onishi, A., Phan, A.H., Matsuoka, K., Cichocki, A., 2012. Tensor classification for P300-based brain computer interface, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Presented at the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 581–584. https://doi.org/10.1109/ICASSP.2012.6287946

Orsborn, A.L., Dangi, S., Moorman, H.G., Carmena, J.M., 2012. Closed-Loop Decoder Adaptation on Intermediate Time-Scales Facilitates Rapid BMI Performance Improvements Independent of Decoder Initialization Conditions. IEEE Transactions on Neural Systems and Rehabilitation Engineering 20, 468–477. https://doi.org/10.1109/TNSRE.2012.2185066

Orsborn, A.L., Dangi, S., Moorman, H.G., Carmena, J.M., 2011. Exploring time-scales of closed-loop decoder adaptation in brain-machine interfaces, in: 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Presented at the 2011 Annual International

Conference of the IEEE Engineering in Medicine and Biology Society, pp. 5436–5439. https://doi.org/10.1109/IEMBS.2011.6091387

Orsborn, A.L., Moorman, H.G., Overduin, S.A., Shanechi, M.M., Dimitrov, D.F., Carmena, J.M., 2014. Closed-Loop Decoder Adaptation Shapes Neural Plasticity for Skillful Neuroprosthetic Control. Neuron 82, 1380–1393. https://doi.org/10.1016/j.neuron.2014.04.048

Ortner, R., Allison, B.Z., Korisek, G., Gaggl, H., Pfurtscheller, G., 2011. An SSVEP BCI to Control a Hand Orthosis for Persons With Tetraplegia. IEEE Transactions on Neural Systems and Rehabilitation Engineering 19, 1–5. https://doi.org/10.1109/TNSRE.2010.2076364

Palmer J.A., Hirata M., 2018. Independent Component Analysis (ICA) Features for Electro-corticographic (ECoG) Brain-Machine Interfaces (BMIs). 臨床神経生理, Jpn J Clin Neurophysiol 46, 55–60. https://doi.org/10.11422/jscn.46.55

Pandarinath, C., O'Shea, D.J., Collins, J., Jozefowicz, R., Stavisky, S.D., Kao, J.C., Trautmann, E.M., Kaufman, M.T., Ryu, S.I., Hochberg, L.R., Henderson, J.M., Shenoy, K.V., Abbott, L.F., Sussillo, D., 2018. Inferring single-trial neural population dynamics using sequential auto-encoders. Nat Methods 15, 805–815. https://doi.org/10.1038/s41592-018-0109-9

Paraskevopoulou, S.E., Wu, D., Eftekhar, A., Constandinou, T.G., 2014. Hierarchical Adaptive Means (HAM) clustering for hardware-efficient, unsupervised and real-time spike sorting. Journal of Neuroscience Methods 235, 145–156. https://doi.org/10.1016/j.jneumeth.2014.07.004

Park, S.-M., Kim, J.-Y., Sim, K.-B., 2018. EEG electrode selection method based on BPSO with channel impact factor for acquisition of significant brain signal. Optik - International Journal for Light and Electron Optics 155, 89–96. https://doi.org/10.1016/j.ijleo.2017.10.085

Pels, E.G.M., Aarnoutse, E.J., Leinders, S., Freudenburg, Z.V., Branco, M.P., van der Vijgh, B.H., Snijders, T.J., Denison, T., Vansteensel, M.J., Ramsey, N.F., 2019. Stability of a chronic implanted brain-computer interface in late-stage amyotrophic lateral sclerosis. Clinical Neurophysiology 130, 1798–1803. https://doi.org/10.1016/j.clinph.2019.07.020

Pereira Da Silva, A., Comon, P., De Almeida, A.L.F., 2015. Rank-1 Tensor Approximation Methods and Application to Deflation [Research Report]. GIPSA-lab.

Perge, J.A., Homer, M.L., Malik, W.Q., Cash, S., Eskandar, E., Friehs, G., Donoghue, J.P., Hochberg, L.R., 2013. Intra-day signal instabilities affect decoding performance in an intracortical neural interface system. J. Neural Eng. 10, 036004. https://doi.org/10.1088/1741-2560/10/3/036004

Peterson, V., Wyser, D., Lambercy, O., Spies, R., Gassert, R., 2019. A penalized time-frequency band feature selection and classification procedure for improved motor intention decoding in multichannel EEG. J. Neural Eng. 16, 016019. https://doi.org/10.1088/1741-2552/aaf046

Pfeiffer, T., Knight, R.T., Rose, G., 2018. Hidden Markov model based continuous decoding of finger movements with prior knowledge incorporation using bi-gram models. Biomed. Phys. Eng. Express 4, 025007. https://doi.org/10.1088/2057-1976/aa99f3

Pfurtscheller, G., Guger, C., Müller, G., Krausz, G., Neuper, C., 2000. Brain oscillations control hand orthosis in a tetraplegic. Neuroscience Letters 292, 211–214. https://doi.org/10.1016/S0304-3940(00)01471-3

Pfurtscheller, G., Müller-Putz, G.R., Scherer, R., Neuper, C., 2008. Rehabilitation with Brain-Computer Interface Systems. Computer 41, 58–65. https://doi.org/10.1109/MC.2008.432

Pfurtscheller, G., Solis-Escalante, T., Ortner, R., Linortner, P., Muller-Putz, G.R., 2010. Self-Paced Operation of an SSVEP-Based Orthosis With and Without an Imagery-Based "Brain Switch:" A Feasibility Study Towards a Hybrid BCI. IEEE Transactions on Neural Systems and Rehabilitation Engineering 18, 409–414. https://doi.org/10.1109/TNSRE.2010.2040837

Pistohl, T., Ball, T., Schulze-Bonhage, A., Aertsen, A., Mehring, C., 2008. Prediction of arm movement trajectories from ECoG-recordings in humans. Journal of Neuroscience Methods, Brain-Computer Interfaces (BCIs) 167, 105–114. https://doi.org/10.1016/j.jneumeth.2007.10.001

Polikar, R., 1996. THE WAVELET TUTORIAL SECOND EDITION.

Postelnicu, C.-C., Girbacia, F., Voinea, G.-D., Boboc, R., 2019. Towards Hybrid Multimodal Brain Computer Interface for Robotic Arm Command, in: Schmorrow, D.D., Fidopiastis, C.M. [Eds.], Augmented Cognition, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 461–470. https://doi.org/10.1007/978-3-030-22419-6_33

Pourbakhtiar, A., Shamsi, M., Farrokhshad, F., 2013. Neuro-fuzzy classification of brain computer interface data using phase based feature, in: 2013 13th Iranian Conference on Fuzzy Systems (IFSC). Presented at the 2013 13th Iranian Conference on Fuzzy Systems (IFSC), pp. 1–4. https://doi.org/10.1109/IFSC.2013.6675683

Purves, D., Augustine, G.J., Fitzpatrick, D., Hall, W.C., LaMantia, A.-S., McNamara, J.O., Williams, S.M., 2004. Neuroscience, 3rd ed, Neuroscience, 3rd ed. Sinauer Associates, Sunderland, MA, US.

Qi, Y., Klein-Seetharaman, J., Bar-Joseph, Z., 2007. A mixture of feature experts approach for protein-protein interaction prediction. BMC Bioinformatics 8, S6. https://doi.org/10.1186/1471-2105-8-S10-S6

Qibin Zhao, Liqing Zhang, Cichocki, A., Jie Li, 2008. Incremental Common Spatial Pattern algorithm for BCI, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). Presented at the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 2656–2659. https://doi.org/10.1109/IJCNN.2008.4634170

Qin, S.J., 1998. Recursive PLS algorithms for adaptive data modeling. Computers & Chemical Engineering 22, 503–514.

Qin, S.J., 1993. Partial least squares regression for recursive system identification, in: Proceedings of 32nd IEEE Conference on Decision and Control. Presented at the Proceedings of 32nd IEEE Conference on Decision and Control, pp. 2617–2622 vol.3. https://doi.org/10.1109/CDC.1993.325671

Qin, Z., Xu, Y., Shu, X., Hua, L., Sheng, X., Zhu, X., 2019. eConHand: A Wearable Brain-Computer Interface System for Stroke Rehabilitation, in: 2019 9th International IEEE/EMBS Conference on Neural Engineering (NER). Presented at the 2019 9th International IEEE/EMBS Conference on Neural Engineering (NER), pp. 734–737. https://doi.org/10.1109/NER.2019.8716940

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257–286.

Rak, R.J., Kołodziej, M., Majkowski, A., 2012. Brain-computer interface as measurement and control system The review paper. Metrology and Measurement Systems Vol. 19, 427–444.

Ramadan, R.A., Refat, S., Elshahed, M.A., Ali, R.A., 2015. Basics of Brain Computer Interface, in: Brain-Computer Interfaces, Intelligent Systems Reference Library. Springer, Cham, pp. 31–50. https://doi.org/10.1007/978-3-319-10978-7_2

Rani, M.S., S, S., 2017. Perspectives of the performance metrics in lexicon and hybrid based approaches: a review. International Journal of Engineering & Technology 6, 108–115. https://doi.org/10.14419/ijet.v6i4.8295

Rashid, U., Niazi, I.K., Signal, N., Taylor, D., 2018. An EEG Experimental Study Evaluating the Performance of Texas Instruments ADS1299. Sensors 18, 3721. https://doi.org/10.3390/s18113721

Rembado, I., Castagnola, E., Turella, L., Ius, T., Budai, R., Ansaldo, A., Angotzi, G.N., Debertoldi, F., Ricci, D., Skrap, M., Fadiga, L., 2016. Independent Component Decomposition of Human Somatosensory Evoked Potentials Recorded by Micro-Electrocorticography. Int. J. Neur. Syst. 27, 1650052. https://doi.org/10.1142/S0129065716500520

Remeseiro, B., Bolon-Canedo, V., 2019. A review of feature selection methods in medical applications. Computers in Biology and Medicine 112, 103375. https://doi.org/10.1016/j.compbiomed.2019.103375

Rivet, B., Souloumiac, A., Attina, V., Gibert, G., 2009. xDAWN Algorithm to Enhance Evoked Potentials: Application to Brain–Computer Interface. IEEE Transactions on Biomedical Engineering 56, 2035–2043. https://doi.org/10.1109/TBME.2009.2012869

Roberts, T.T., Leonard, G.R., Cepela, D.J., 2017. Classifications In Brief: American Spinal Injury Association (ASIA) Impairment Scale. Clin Orthop Relat Res 475, 1499–1504. https://doi.org/10.1007/s11999-016-5133-4

Robinet, S., Audebert, P., Regis, G., Zongo, B., Beche, J.-F., Condemine, C., Filipe, S., Charvet, G., 2011. A Low-Power 0.7 $\mu \rm V_rms$ 32-Channel Mixed-Signal Circuit for ECoG Recordings. IEEE Journal on Emerging and Selected Topics in Circuits and Systems 1, 451–460. https://doi.org/10.1109/JETCAS.2011.2180835

Robinson, N., Guan, C., Vinod, A.P., Ang, K.K., Tee, K.P., 2013. Multi-class EEG classification of voluntary hand movement directions. J. Neural Eng. 10, 056018. https://doi.org/10.1088/1741-2560/10/5/056018

Roijendijk, L., Gielen, S., Farquhar, J., 2016. Classifying Regularized Sensor Covariance Matrices: An Alternative to CSP. IEEE Transactions on Neural Systems and Rehabilitation Engineering 24, 893–900. https://doi.org/10.1109/TNSRE.2015.2477687

Rong, H., Li, C., Bao, R., Chen, B., 2018a. Incremental Adaptive EEG Classification of Motor Imagery-based BCI, in: 2018 International Joint Conference on Neural Networks (IJCNN). Presented at the 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. https://doi.org/10.1109/IJCNN.2018.8489283

Rong, H., Li, C., Bao, R., Chen, B., 2018b. Incremental Adaptive EEG Classification of Motor Imagery-based BCI, in: 2018 International Joint Conference on Neural Networks (IJCNN). Presented at the 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. https://doi.org/10.1109/IJCNN.2018.8489283

Rosas-Cholula, G., Ramírez-Cortes, J.M., Alarcón-Aquino, V., Martinez-Carballido, J., Gomez-Gil, P., 2010. On Signal P-300 Detection for BCI Applications Based on Wavelet Analysis and ICA Preprocessing, in: 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference. Presented at the 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference, pp. 360–365. https://doi.org/10.1109/CERMA.2010.48

Rouhi, A., Nezamabadi-Pour, H., 2020. Feature Selection in High-Dimensional Data, in: Amini, M.H. [Ed.], Optimization, Learning, and Control for Interdependent Complex Networks, Advances in Intelligent Systems and Computing. Springer International Publishing, Cham, pp. 85–128. https://doi.org/10.1007/978-3-030-34094-0_5

Rousche, P.J., Normann, R.A., 1999. Chronic intracortical microstimulation (ICMS) of cat sensory cortex using the Utah intracortical electrode array. IEEE Transactions on Rehabilitation Engineering 7, 56–68. https://doi.org/10.1109/86.750552

Rouse, A.G., Williams, J.J., Wheeler, J.J., Moran, D.W., 2016. Spatial co-adaptation of cortical control columns in a micro-ECoG brain–computer interface. J. Neural Eng. 13, 056018. https://doi.org/10.1088/1741-2560/13/5/056018

Roy, R.N., Moly, A., Dehais, F., Scannella, S., 2018. Chapter 98 - EEG and FNIRS Connectivity Features for Mental Workload Assessment: A Preliminary Study, in: Ayaz, H., Dehais, F. [Eds.], Neuroergonomics. Academic Press, pp. 327–328. https://doi.org/10.1016/B978-0-12-811926-6.00098-1

Royer, A.S., Doud, A.J., Rose, M.L., He, B., 2010. EEG Control of a Virtual Helicopter in 3-Dimensional Space Using Intelligent Control Strategies. IEEE Transactions on Neural Systems and Rehabilitation Engineering 18, 581–589. https://doi.org/10.1109/TNSRE.2010.2077654

Ruan, J., Wu, X., Zhou, B., Guo, X., 2017. Automatic electroencephalogram channel selection of independent component analysis based motor imagery brain-computer interface, in: 2017 10th International

Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). Presented at the 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–6. https://doi.org/10.1109/CISP-BMEI.2017.8302213

Saa, J.F.D., Çetin, M., 2013. Discriminative Methods for Classification of Asynchronous Imaginary Motor Tasks From EEG Data. IEEE Transactions on Neural Systems and Rehabilitation Engineering 21, 716–724. https://doi.org/10.1109/TNSRE.2013.2268194

Saa, J.F.D., Çetin, M., 2012. A latent discriminative model-based approach for classification of imaginary motor tasks from EEG data. J. Neural Eng. 9, 026020. https://doi.org/10.1088/1741-2560/9/2/026020

Saa, J.F.D., Pesters, A. de, Cetin, M., 2016. Asynchronous decoding of finger movements from ECoG signals using long-range dependencies conditional random fields. J. Neural Eng. 13, 036017. https://doi.org/10.1088/1741-2560/13/3/036017

Sakhavi, S., Guan, C., Yan, S., 2018. Learning Temporal Information for Brain-Computer Interface Using Convolutional Neural Networks. IEEE Transactions on Neural Networks and Learning Systems 29, 5619–5629. https://doi.org/10.1109/TNNLS.2018.2789927

Salari, E., Freudenburg, Z.V., Branco, M.P., Aarnoutse, E.J., Vansteensel, M.J., Ramsey, N.F., 2019. Classification of Articulator Movements and Movement Direction from Sensorimotor Cortex Activity. Scientific Reports 9, 1–12. https://doi.org/10.1038/s41598-019-50834-5

Salazar-Ramirez, A., Martin, J.I., Martinez, R., Arruti, A., Muguerza, J., Sierra, B., 2019. A hierarchical architecture for recognising intentionality in mental tasks on a brain-computer interface. PLOS ONE 14, e0218181. https://doi.org/10.1371/journal.pone.0218181

Salinas, E., Abbott, L.F., 1994. Vector reconstruction from firing rates. J Comput Neurosci 1, 89–107. https://doi.org/10.1007/BF00962720

Sannelli, C., Vidaurre, C., Müller, K.-R., Blankertz, B., 2016. Ensembles of adaptive spatial filters increase BCI performance: an online evaluation. J. Neural Eng. 13, 046003. https://doi.org/10.1088/1741-2560/13/4/046003

Sauter-Starace, F., Ratel, D., Cretallaz, C., Foerster, M., Lambert, A., Gaude, C., Costecalde, T., Bonnet, S., Charvet, G., Aksenova, T., Mestais, C., Benabid, A.-L., Torres-Martinez, N., 2019. Long-Term Sheep Implantation of WIMAGINE®, a Wireless 64-Channel Electrocorticogram Recorder. Front. Neurosci. 13. https://doi.org/10.3389/fnins.2019.00847

Sburlea, A.I., Montesano, L., Minguez, J., 2017. Advantages of EEG phase patterns for the detection of gait intention in healthy and stroke subjects. J. Neural Eng. 14, 036004. https://doi.org/10.1088/1741-2552/aa5f2f

Schaeffer, M.-C., 2017. ECoG signal processing for Brain Computer Interface with multiple degrees of freedom for clinical application. Université Grenoble Alpes, Grenoble.

Schaeffer, M.-C., Aksenova, T., 2018. Data-Driven Transducer Design and Identification for Internally-Paced Motor Brain Computer Interfaces: A Review. Frontiers in Neuroscience 12, 540. https://doi.org/10.3389/fnins.2018.00540

Schaeffer, M.-C., Aksenova, T., 2016a. Switching Markov decoders for asynchronous trajectory reconstruction from ECoG signals in monkeys for BCI applications. Journal of Physiology-Paris, SI: GDR Multielectrode 110, 348–360. https://doi.org/10.1016/j.jphysparis.2017.03.002

Schaeffer, M.-C., Aksenova, T., 2016b. Switching Markov decoders for asynchronous trajectory reconstruction from ECoG signals in monkeys for BCI applications. Journal of Physiology-Paris, SI: GDR Multielectrode 110, 348–360. https://doi.org/10.1016/j.jphysparis.2017.03.002

Schaffelhofer, S., Agudelo-Toro, A., Scherberger, H., 2015. Decoding a Wide Range of Hand Configurations from Macaque Motor, Premotor, and Parietal Cortices. J. Neurosci. 35, 1068–1081. https://doi.org/10.1523/JNEUROSCI.3594-14.2015

Schalk, G., Kubánek, J., Miller, K.J., Anderson, N.R., Leuthardt, E.C., Ojemann, J.G., Limbrick, D., Moran, D., Gerhardt, L.A., Wolpaw, J.R., 2007. Decoding two-dimensional movement trajectories using electrocorticographic signals in humans. J. Neural Eng. 4, 264–275. https://doi.org/10.1088/1741-2560/4/3/012

Schalk, G., Leuthardt, E.C., 2011. Brain-Computer Interfaces Using Electrocorticographic Signals. IEEE Reviews in Biomedical Engineering 4, 140–154. https://doi.org/10.1109/RBME.2011.2172408

Schalk, G., Miller, K.J., Anderson, N.R., Wilson, J.A., Smyth, M.D., Ojemann, J.G., Moran, D.W., Wolpaw, J.R., Leuthardt, E.C., 2008. Two-dimensional movement control using electrocorticographic signals in humans. J. Neural Eng. 5, 75. https://doi.org/10.1088/1741-2560/5/1/008

Schenker, A., Last, M., Bunke, H., Kandel, A., 2003. Comparison of Distance Measures for Graph-Based Clustering of Documents, in: Hancock, E., Vento, M. [Eds.], Graph Based Representations in Pattern Recognition, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 202–213. https://doi.org/10.1007/3-540-45028-9_18

Scherer, R., Faller, J., Friedrich, E.V.C., Opisso, E., Costa, U., Kübler, A., Müller-Putz, G.R., 2015. Individually Adapted Imagery Improves Brain-Computer Interface Performance in End-Users with Disability. PLOS ONE 10, e0123727. https://doi.org/10.1371/journal.pone.0123727

Schlögl, A., Lee, F., Bischof, H., Pfurtscheller, G., 2005. Characterization of four-class motor imagery EEG data for the BCI-competition 2005. J. Neural Eng. 2, L14–L22. https://doi.org/10.1088/1741-2560/2/4/L02

Schlögl, A., Vidaurre, C., Müller, K.-R., 2010. Adaptive Methods in BCI Research - An Introductory Tutorial, in: Graimann, B., Pfurtscheller, G., Allison, B. [Eds.], Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction, The Frontiers Collection. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 331–355. https://doi.org/10.1007/978-3-642-02091-9_18

Schreuder, M., Höhne, J., Blankertz, B., Haufe, S., Dickhaus, T., Tangermann, M., 2013. Optimizing event-related potential based brain–computer interfaces: a systematic evaluation of dynamic stopping methods. J. Neural Eng. 10, 036025. https://doi.org/10.1088/1741-2560/10/3/036025

Schroder, M., Bogdan, M., Hinterberger, T., Birbaumer, N., 2003. Automated EEG feature selection for brain computer interfaces, in: First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings. Presented at the First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings., pp. 626–629. https://doi.org/10.1109/CNE.2003.1196906

Schwartz, A.B., Taylor, D.M., Tillery, S.I.H., 2001. Extraction algorithms for cortical control of arm prosthetics. Current Opinion in Neurobiology 11, 701–708. https://doi.org/10.1016/S0959-4388(01)00272-0

Schwarz, A., Höller, M.K., Pereira, J., Ofner, P., Müller-Putz, G.R., 2020. Decoding hand movements from human EEG to control a robotic arm in a simulation environment. J. Neural Eng. https://doi.org/10.1088/1741-2552/ab882e

Schwemmer, M.A., Skomrock, N.D., Sederberg, P.B., Ting, J.E., Sharma, G., Bockbrader, M.A., Friedenberg, D.A., 2018. Meeting brain–computer interface user performance expectations using a deep neural network decoding framework. Nature Medicine 1. https://doi.org/10.1038/s41591-018-0171-y

Seifzadeh, S., Rezaei, M., Faez, K., Amiri, M., 2017. Fast and Efficient Four-class Motor Imagery Electroencephalography Signal Analysis Using Common Spatial Pattern-Ridge Regression Algorithm for the Purpose of Brain-Computer Interface. J Med Signals Sens 7, 80–85.

Senhadji, L., Kachenoura, A., Albera, L., Comon, P., 2009. Sur l'exploitation des approches d'analyse en composantes indépendantes dans les interfaces cerveau machine. IRBM, NUMÉRO SPÉCIAL TECHNOLOGIES POUR L'AUTONOMIE 30, 211–217. https://doi.org/10.1016/j.irbm.2009.10.005

Serby, H., Yom-Tov, E., Inbar, G.F., 2005. An improved P300-based brain-computer interface. IEEE Transactions on Neural Systems and Rehabilitation Engineering 13, 89–98. https://doi.org/10.1109/TNSRE.2004.841878

Shalamov, V., Efimova, V., Muravyov, S., Filchenkov, A., 2018. Reinforcement-based Method for Simultaneous Clustering Algorithm Selection and its Hyperparameters Optimization. Procedia Computer Science, 7th International Young Scientists Conference on Computational Science, YSC2018, 02-06 July2018, Heraklion, Greece 136, 144–153. https://doi.org/10.1016/j.procs.2018.08.247

Shanechi, M.M., Orsborn, A., Moorman, H., Gowda, S., Carmena, J.M., 2014. High-performance brain-machine interface enabled by an adaptive optimal feedback-controlled point process decoder. IEEE, pp. 6493–6496. https://doi.org/10.1109/EMBC.2014.6945115

Shanechi, M.M., Orsborn, A.L., Carmena, J.M., 2016. Robust Brain-Machine Interface Design Using Optimal Feedback Control Modeling and Adaptive Point Process Filtering. PLOS Computational Biology 12, e1004730. https://doi.org/10.1371/journal.pcbi.1004730

Shanechi, M.M., Orsborn, A.L., Moorman, H.G., Gowda, S., Dangi, S., Carmena, J.M., 2017. Rapid control and feedback rates enhance neuroprosthetic control. Nature Communications 8, 13825. https://doi.org/10.1038/ncomms13825

Shanechi, M.M., Williams, Z.M., Wornell, G.W., Hu, R.C., Powers, M., Brown, E.N., 2013. A Real-Time Brain-Machine Interface Combining Motor Target and Trajectory Intent Using an Optimal Feedback Control Design. PLOS ONE 8, e59049. https://doi.org/10.1371/journal.pone.0059049

Sharghian, V., Rezaii, T.Y., Farzamnia, A., Tinati, M.A., 2019. Online Dictionary Learning for Sparse Representation-Based Classification of Motor Imagery EEG, in: 2019 27th Iranian Conference on Electrical Engineering (ICEE). Presented at the 2019 27th Iranian Conference on Electrical Engineering (ICEE), pp. 1793–1797. https://doi.org/10.1109/IranianCEE.2019.8786703

Sheikhattar, A., Fritz, J.B., Shamma, S.A., Babadi, B., 2015. Adaptive sparse logistic regression with application to neuronal plasticity analysis, in: 2015 49th Asilomar Conference on Signals, Systems and Computers. Presented at the 2015 49th Asilomar Conference on Signals, Systems and Computers, pp. 1551–1555. https://doi.org/10.1109/ACSSC.2015.7421406

Shenoy, K.V., Carmena, J.M., 2014. Combining Decoder Design and Neural Adaptation in Brain-Machine Interfaces. Neuron 84, 665–680. https://doi.org/10.1016/j.neuron.2014.08.038

Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P.N., Müller, K.-R., 2006. Towards adaptive classification for BCI. Journal of neural engineering 3, NaN-NaN. https://doi.org/10.1088/1741-2560/3/1/R02

Shimoda, K., Nagasaka, Y., Chao, Z.C., Fujii, N., 2012. Decoding continuous three-dimensional hand trajectories from epidural electrocorticographic signals in Japanese macaques. J. Neural Eng. 9, 036015. https://doi.org/10.1088/1741-2560/9/3/036015

Shin, Y., Lee, S., Ahn, M., Cho, H., Jun, S.C., Lee, H.-N., 2015. Noise robustness analysis of sparse representation based classification method for non-stationary EEG signal classification. Biomedical Signal Processing and Control 21, 8–18. https://doi.org/10.1016/j.bspc.2015.05.007

Shojaedini, S.V., Morabbi, S., Keyvanpour, M., 2018. A New Method for Detecting P300 Signals by Using Deep Learning: Hyperparameter Tuning in High-Dimensional Space by Minimizing Nonconvex Error Function. Journal of Medical Signals and Sensors 8, 205–214.

Shpigelman, L., Lalazar, H., Vaadia, E., 2009. Kernel-ARMA for Hand Tracking and Brain-Machine interfacing During 3D Motor Control, in: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. [Eds.], Advances in Neural Information Processing Systems 21. Curran Associates, Inc., pp. 1489–1496.

Silva, A.P. da, Comon, P., Almeida, A.L.F. de, 2015. An iterative deflation algorithm for exact CP tensor decomposition, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing

(ICASSP). Presented at the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3961–3965. https://doi.org/10.1109/ICASSP.2015.7178714

Simeral, J.D., Kim, S.-P., Black, M.J., Donoghue, J.P., Hochberg, L.R., 2011. Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array. J Neural Eng 8, 025027. https://doi.org/10.1088/1741-2560/8/2/025027

Sirpal, P., Kassab, A., Pouliot, P., Nguyen, D.K., Lesage, F., 2019. fNIRS improves seizure detection in multimodal EEG-fNIRS recordings. JBO 24, 051408. https://doi.org/10.1117/1.JBO.24.5.051408

Slivkins, A., 2019. Introduction to Multi-Armed Bandits. MAL 12, 1–286. https://doi.org/10.1561/2200000068

Solaimanpour, S., Doshi, P., 2017. A layered HMM for predicting motion of a leader in multi-robot settings, in: 2017 IEEE International Conference on Robotics and Automation (ICRA). Presented at the 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 788–793. https://doi.org/10.1109/ICRA.2017.7989097

Song, X., Yoon, S.-C., 2015. Improving brain–computer interface classification using adaptive common spatial patterns. Computers in Biology and Medicine 61, 150–160. https://doi.org/10.1016/j.compbiomed.2015.03.023

Sparks, E.R., Talwalkar, A., Haas, D., Franklin, M.J., Jordan, M.I., 2015. Automating model search for large scale machine learning, in: Proceedings of the Sixth ACM Symposium on Cloud Computing. Presented at the Proceedings of the Sixth ACM Symposium on Cloud Computing.

Spinnato, J., Roubaud, M.-C., Burle, B., Torrésani, B., 2015. Detecting single-trial EEG evoked potential using a wavelet domain linear mixed model: application to error potentials classification. J. Neural Eng. 12, 036013. https://doi.org/10.1088/1741-2560/12/3/036013

Spüler, M., Bensch, M., Kleih, S., Rosenstiel, W., Bogdan, M., Kübler, A., 2012a. Online use of error-related potentials in healthy users and people with severe motor impairment increases performance of a P300-BCI. Clinical Neurophysiology 123, 1328–1337. https://doi.org/10.1016/j.clinph.2011.11.082

Spüler, M., Rosenstiel, W., Bogdan, M., 2012b. Adaptive SVM-Based Classification Increases Performance of a MEG-Based Brain-Computer Interface (BCI), in: Villa, A.E.P., Duch, W., Érdi, P., Masulli, F., Palm, G. [Eds.], Artificial Neural Networks and Machine Learning – ICANN 2012, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 669–676.

Sreeja, S.R., Himanshu, Samanta, D., Sarma, M., 2019. Weighted sparse representation for classification of motor imagery EEG signals, in: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6180–6183. https://doi.org/10.1109/EMBC.2019.8857496

Sreenath, R., Ramana, R., 2017. Classification of denoising techniques for EEG signals: A review. International Journal of Pure and Applied Mathematics 117, 967–972.

Steinberg, O., Donchin, O., Gribova, A., Oliveira, S.C.D., Bergman, H., Vaadia, E., 2002. Neuronal populations in primary motor cortex encode bimanual arm movements. European Journal of Neuroscience 15, 1371–1380. https://doi.org/10.1046/j.1460-9568.2002.01968.x

Stieglitz, T., Rubehn, B., Henle, C., Kisban, S., Herwik, S., Ruther, P., Schuettler, M., 2009. Brain–computer interfaces: an overview of the hardware to record neural signals from the cortex, in: Verhaagen, J., Hol, E.M., Huitenga, I., Wijnholds, J., Bergen, A.B., Boer, G.J., Swaab, D.F. [Eds.], Progress in Brain Research, Neurotherapy: Progress in Restorative Neuroscience and Neurology. Elsevier, pp. 297–315. https://doi.org/10.1016/S0079-6123(09)17521-0

Stiller, J.C., Radons, G., 1999. Online estimation of hidden Markov models. IEEE Signal Processing Letters 6, 213–215.

Stone, J.V., 2002. Independent component analysis: an introduction. Trends in Cognitive Sciences 6, 59–64. https://doi.org/10.1016/S1364-6613(00)01813-1

Sugiura, T., Goto, N., Hayashi, A., 2007. A Discriminative Model Corresponding to Hierarchical HMMs, in: Yin, H., Tino, P., Corchado, E., Byrne, W., Yao, X. [Eds.], Intelligent Data Engineering and Automated Learning - IDEAL 2007, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 375–384.

Suk, H., Lee, S., 2010. Two-Layer Hidden Markov Models for Multi-class Motor Imagery Classification, in: 2010 First Workshop on Brain Decoding: Pattern Recognition Challenges in Neuroimaging. Presented at the 2010 First Workshop on Brain Decoding: Pattern Recognition Challenges in Neuroimaging, pp. 5–8. https://doi.org/10.1109/WBD.2010.16

Suminski, A.J., Fagg, A.H., Willett, F.R., Bodenhamer, M., Hatsopoulos, N.G., 2013. Online adaptive decoding of intended movements with a hybrid kinetic and kinematic brain machine interface, in: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 1583–1586. https://doi.org/10.1109/EMBC.2013.6609817

Suminski, A.J., Tkach, D.C., Fagg, A.H., Hatsopoulos, N.G., 2010. Incorporating Feedback from Multiple Sensory Modalities Enhances Brain–Machine Interface Control. J. Neurosci. 30, 16777–16787. https://doi.org/10.1523/JNEUROSCI.3967-10.2010

Sun, Y., Zeng, H., Song, A., Xu, B., Li, H., Liu, J., Wen, P., 2017. Investigation of the phase feature of low-frequency electroencephalography signals for decoding hand movement parameters, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Presented at the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2312–2316. https://doi.org/10.1109/SMC.2017.8122966

Sussillo, D., Stavisky, S.D., Kao, J.C., Ryu, S.I., Shenoy, K.V., 2016. Making brain–machine interfaces robust to future neural variability. Nature Communications 7, 13749. https://doi.org/10.1038/ncomms13749

Sutton, C., McCallum, A., 2012. An Introduction to Conditional Random Fields. MAL 4, 267–373. https://doi.org/10.1561/2200000013

Sutton, C., McCallum, A., 2010. An Introduction to Conditional Random Fields.

Sutton, R.S., Barto, A.G., 2017. Reinforcement Learning: An Introduction, Second edition. ed. Bradford Book, The MIT Press, Cambridge, MA.

Suway, S.B., Tien, R.N., Jeffries, S.M., Zohny, Z., Clanton, S.T., McMorland, A.J.C., Velliste, M., 2013. Resting state detection for gating movement of a neural prosthesis, in: 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER). Presented at the 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER), pp. 665–668. https://doi.org/10.1109/NER.2013.6696022

Syan, C.S., Harnarinesingh, R.E.S., 2010. Comparison of Pre-Processing and Classification Techniques for Single-Trial and Multi-Trial P300-Based Brain Computer Interfaces. American Journal of Applied Sciences 7, 1219–1225. https://doi.org/10.3844/ajassp.2010.1219.1225

Tan, P., Wang, X., Wang, Y., 2020. Dimensionality reduction in evolutionary algorithms-based feature selection for motor imagery brain-computer interface. Swarm and Evolutionary Computation 52, 100597. https://doi.org/10.1016/j.swevo.2019.100597

Taylor, D.M., Tillery, S.I.H., Schwartz, A.B., 2002. Direct Cortical Control of 3D Neuroprosthetic Devices. Science 296, 1829–1832. https://doi.org/10.1126/science.1070291

Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K., 2013. Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13. ACM, New York, NY, USA, pp. 847–855. https://doi.org/10.1145/2487575.2487629

Thulasidas, M., Cuntai Guan, Jiankang Wu, 2006. Robust classification of EEG signal for brain-computer interface. IEEE Transactions on Neural Systems and Rehabilitation Engineering 14, 24–29. https://doi.org/10.1109/TNSRE.2005.862695

Toda, A., Imamizu, H., Kawato, M., Sato, M., 2011. Reconstruction of two-dimensional movement trajectories from selected magnetoencephalography cortical currents by combined sparse Bayesian methods. NeuroImage 54, 892–905. https://doi.org/10.1016/j.neuroimage.2010.09.057

Tomasi, G., 2006. Practical and Computational Aspects in Chemometric Data Analysis. The Royal Veterinary and Agricultural University, Frederiksberg, Denmark.

Tomasi, G., Bro, R., 2006. A comparison of algorithms for fitting the PARAFAC model. Computational Statistics & Data Analysis 50, 1700–1734. https://doi.org/10.1016/j.csda.2004.11.013

Townsend, G., Graimann, B., Pfurtscheller, G., 2004. Continuous EEG classification during motor imagery-simulation of an asynchronous BCI. IEEE Transactions on Neural Systems and Rehabilitation Engineering 12, 258–265. https://doi.org/10.1109/TNSRE.2004.827220

Trejo, L.J., Rosipal, R., Matthews, B., 2006. Brain-computer interfaces for 1-D and 2-D cursor control: designs using volitional control of the EEG spectrum or steady-state visual evoked potentials. IEEE Transactions on Neural Systems and Rehabilitation Engineering 14, 225–229. https://doi.org/10.1109/TNSRE.2006.875578

Tseng, P.-H., Urpi, N.A., Lebedev, M., Nicolelis, M., 2019. Decoding Movements from Cortical Ensemble Activity Using a Long Short-Term Memory Recurrent Network. Neural Computation 31, 1085–1113. https://doi.org/10.1162/neco_a_01189

Úbeda, A., Azorín, J.M., Chavarriaga, R., R. Millán, J. del, 2017. Classification of upper limb center-out reaching tasks by means of EEG-based continuous decoding techniques. Journal of NeuroEngineering and Rehabilitation 14, 9. https://doi.org/10.1186/s12984-017-0219-0

Úbeda, A., Azorín, J.M., Farina, D., Sartori, M., 2018. Estimation of Neuromuscular Primitives from EEG Slow Cortical Potentials in Incomplete Spinal Cord Injury Individuals for a New Class of Brain-Machine Interfaces. Front. Comput. Neurosci. 12. https://doi.org/10.3389/fncom.2018.00003

Umakanth, N., Santhi, S., 2020. Classification and ranking of trending topics in twitter using tweets text. Journal of Critical Reviews 7, 895–899. https://doi.org/10.31838/jcr.07.04.171

Uschmajew, A., 2015. A new convergence proof for the higher-order power method and generalizations. arXiv:1407.4586 [math].

Utsumi, K., Takano, K., Okahara, Y., Komori, T., Onodera, O., Kansaku, K., 2018. Operation of a P300-based brain-computer interface in patients with Duchenne muscular dystrophy. Scientific Reports 8, 1753. https://doi.org/10.1038/s41598-018-20125-6

van Dokkum, L.E.H., Ward, T., Laffont, I., 2015. Brain computer interfaces for neurorehabilitation – its current status as a rehabilitation strategy post-stroke. Annals of Physical and Rehabilitation Medicine 58, 3–8. https://doi.org/10.1016/j.rehab.2014.09.016

van Gerven, M., Hesse, C., Jensen, O., Heskes, T., 2009. Interpreting single trial data using groupwise regularisation. NeuroImage 46, 665–676. https://doi.org/10.1016/j.neuroimage.2009.02.041

Vansteensel, M.J., Pels, E.G.M., Bleichner, M.G., Branco, M.P., Denison, T., Freudenburg, Z.V., Gosselaar, P., Leinders, S., Ottens, T.H., Van Den Boom, M.A., Van Rijen, P.C., Aarnoutse, E.J., Ramsey, N.F., 2016. Fully Implanted Brain–Computer Interface in a Locked-In Patient with ALS. New England Journal of Medicine 375, 2060–2066. https://doi.org/10.1056/NEJMoa1608085

Vaskov, A.K., Irwin, Z.T., Nason, S.R., Vu, P.P., Nu, C.S., Bullard, A.J., Hill, M., North, N., Patil, P.G., Chestek, C.A., 2018. Cortical Decoding of Individual Finger Group Motions Using ReFIT Kalman Filter. Frontiers in Neuroscience 12, 751. https://doi.org/10.3389/fnins.2018.00751

Velliste, M., Kennedy, S.D., Schwartz, A.B., Whitford, A.S., Sohn, J.-W., McMorland, A.J.C., 2014. Motor Cortical Correlates of Arm Resting in the Context of a Reaching Task and Implications for Prosthetic Control. J. Neurosci. 34, 6011–6022. https://doi.org/10.1523/JNEUROSCI.3520-13.2014

Velliste, M., Perel, S., Spalding, M.C., Whitford, A.S., Schwartz, A.B., 2008. Cortical control of a prosthetic arm for self-feeding. Nature 453, 1098–1101. https://doi.org/10.1038/nature06996

Vidal, J.J., 1973. Toward Direct Brain-Computer Communication. Annual Review of Biophysics and Bioengineering 2, 157–180. https://doi.org/10.1146/annurev.bb.02.060173.001105

Vidaurre, C., Kawanabe, M., Bünau, P. von, Blankertz, B., Müller, K.R., 2011. Toward Unsupervised Adaptation of LDA for Brain–Computer Interfaces. IEEE Transactions on Biomedical Engineering 58, 587–597. https://doi.org/10.1109/TBME.2010.2093133

Vidaurre, C., Klauer, C., Schauer, T., Ramos-Murguialday, A., Müller, K.-R., 2016. EEG-based BCI for the linear control of an upper-limb neuroprosthesis. Medical Engineering & Physics 38, 1195–1204. https://doi.org/10.1016/j.medengphy.2016.06.010

Vidaurre, C., Krämer, N., Blankertz, B., Schlögl, A., 2009. Time Domain Parameters as a feature for EEG-based Brain–Computer Interfaces. Neural Networks, Brain-Machine Interface 22, 1313–1319. https://doi.org/10.1016/j.neunet.2009.07.020

Vidaurre, C., Scherer, R., Cabeza, R., Schlögl, A., Pfurtscheller, G., 2006a. Study of discriminant analysis applied to motor imagery bipolar data. Med Bio Eng Comput 45, 61. https://doi.org/10.1007/s11517-006-0122-5

Vidaurre, C., Schlogl, A., Cabeza, R., Scherer, R., Pfurtscheller, G., 2007. Study of On-Line Adaptive Discriminant Analysis for EEG-Based Brain Computer Interfaces. IEEE Transactions on Biomedical Engineering 54, 550–556. https://doi.org/10.1109/TBME.2006.888836

Vidaurre, C., Schlogl, A., Cabeza, R., Scherer, R., Pfurtscheller, G., 2006b. A fully on-line adaptive BCI. IEEE Transactions on Biomedical Engineering 53, 1214–1219. https://doi.org/10.1109/TBME.2006.873542

Vilela, M., Hochberg, L.R., 2020. Chapter 8 - Applications of brain-computer interfaces to the control of robotic and prosthetic arms, in: Ramsey, N.F., Millán, J. del R. [Eds.], Handbook of Clinical Neurology, Brain-Computer Interfaces. Elsevier, pp. 87–99. https://doi.org/10.1016/B978-0-444-63934-9.00008-1

Vinjamuri, R., Weber, D.J., Mao, Z.-H., Collinger, J.L., Degenhart, A.D., Kelly, J.W., Boninger, M.L., Tyler-Kabara, E.C., Wang, W., 2011. Toward Synergy-Based Brain-Machine Interfaces. IEEE Transactions on Information Technology in Biomedicine 15, 726–736. https://doi.org/10.1109/TITB.2011.2160272

Vo, K., Pham, T., Nguyen, D.N., Kha, H.H., Dutkiewicz, E., 2018. Subject-Independent ERP-Based Brain–Computer Interfaces. IEEE Transactions on Neural Systems and Rehabilitation Engineering 26, 719–728. https://doi.org/10.1109/TNSRE.2018.2810332

Volkova, K., Lebedev, M.A., Kaplan, A., Ossadtchi, A., 2019. Decoding Movement From Electrocorticographic Activity: A Review. Front. Neuroinform. 13. https://doi.org/10.3389/fninf.2019.00074

Wahnoun, R., He, J., Helms Tillery, S.I., 2006. Selection and parameterization of cortical neurons for neuroprosthetic control. Journal of Neural Engineering 3, 162–171. https://doi.org/10.1088/1741-2560/3/2/010

Waldert, S., Pistohl, T., Braun, C., Ball, T., Aertsen, A., Mehring, C., 2009. A review on directional information in neural signals for brain-machine interfaces. Journal of Physiology-Paris, Neurorobotics 103, 244–254. https://doi.org/10.1016/j.jphysparis.2009.08.007

Waldert, S., Preissl, H., Demandt, E., Braun, C., Birbaumer, N., Aertsen, A., Mehring, C., 2008. Hand Movement Direction Decoded from MEG and EEG. J Neurosci 28, 1000–1008. https://doi.org/10.1523/JNEUROSCI.5171-07.2008

Wang, D., Miao, D., Blohm, G., 2012. Multi-Class Motor Imagery EEG Decoding for Brain-Computer Interfaces. Front. Neurosci. 6. https://doi.org/10.3389/fnins.2012.00151

Wang, K.-Y., Ho, Y.-L., Huang, Y.-D., Fang, W.-C., 2018. A 16 Channel Real-Time EEG Processing Based on ORICA Algorithm using 28nm CMOS Technology, in: 2018 IEEE International Workshop on Signal Processing Systems (SiPS). Presented at the 2018 IEEE International Workshop on Signal Processing Systems (SiPS), pp. 1–6. https://doi.org/10.1109/SiPS.2018.8598440

Wang, Liqi., Chu, M.T., 2014. On the Global Convergence of the Alternating Least Squares Method for Rank-One Approximation to Generic Tensors. SIAM J. Matrix Anal. Appl. 35, 1058–1072. https://doi.org/10.1137/130938207

Wang, P.T., King, C.E., McCrimmon, C.M., Lin, J.J., Sazgar, M., Hsu, F.P.K., Shaw, S.J., Millet, D.E., Chui, L.A., Liu, C.Y., Do, A.H., Nenadic, Z., 2016. Comparison of decoding resolution of standard and high-density electrocorticogram electrodes. Journal of Neural Engineering 13, 026016. https://doi.org/10.1088/1741-2560/13/2/026016

Wang, P.T., McCrimmon, C.M., King, C.E., Shaw, S.J., Millett, D.E., Gong, H., Chui, L.A., Liu, C.Y., Nenadic, Z., Do, A.H., 2017. Characterization of electrocorticogram high-gamma signal in response to varying upper extremity movement velocity. Brain Struct Funct 222, 3705–3748. https://doi.org/10.1007/s00429-017-1429-8

Wang, P.T., Puttock, E.J., King, C.E., Schombs, A., Lin, J.J., Sazgar, M., Hsu, F.P.K., Shaw, S.J., Millett, D.E., Liu, C.Y., Chui, L.A., Do, A.H., Nenadic, Z., 2013. State and trajectory decoding of upper extremity movements from electrocorticogram, in: 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER). Presented at the 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER), pp. 969–972. https://doi.org/10.1109/NER.2013.6696097

Wang, W., Chan, S.S., Heldman, D.A., Moran, D.W., 2007. Motor Cortical Representation of Position and Velocity During Reaching. Journal of Neurophysiology 97, 4258–4270. https://doi.org/10.1152/jn.01180.2006

Wang, W., Collinger, J.L., Degenhart, A.D., Tyler-Kabara, E.C., Schwartz, A.B., Moran, D.W., Weber, D.J., Wodlinger, B., Vinjamuri, R.K., Ashmore, R.C., Kelly, J.W., Boninger, M.L., 2013. An Electrocorticographic Brain Interface in an Individual with Tetraplegia. PLOS ONE 8, e55344. https://doi.org/10.1371/journal.pone.0055344

Wang, W., Degenhart, A.D., Collinger, J.L., Vinjamuri, R., Sudre, G.P., Adelson, P.D., Holder, D.L., Leuthardt, E.C., Moran, D.W., Boninger, M.L., Schwartz, A.B., Crammond, D.J., Tyler-Kabara, E.C., Weber, D.J., 2009. Human Motor Cortical Activity Recorded with Micro-ECoG Electrodes During Individual Finger Movements. Conf Proc IEEE Eng Med Biol Soc 2009, 586–589. https://doi.org/10.1109/IEMBS.2009.5333704

Wang, Y., Hong, B., Gao, X., Gao, S., 2006. Phase Synchrony Measurement in Motor Cortex for Classifying Single-trial EEG during Motor Imagery, in: 2006 International Conference of the IEEE Engineering in Medicine and Biology Society. Presented at the 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 75–78. https://doi.org/10.1109/IEMBS.2006.259673

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., Feitas, N. de, 2016. Bayesian Optimization in a Billion Dimensions via Random Embeddings. 1 55, 361–387. https://doi.org/10.1613/jair.4806

Wang, Z., Ji, Q., Miller, K.J., Schalk, G., 2011. Prior Knowledge Improves Decoding of Finger Flexion from Electrocorticographic Signals. Front. Neurosci. 5. https://doi.org/10.3389/fnins.2011.00127

Wang, Z., Ji, Q., Miller, K.J., Schalk, G., 2010. Decoding Finger Flexion from Electrocorticographic Signals Using a Sparse Gaussian Process, in: 2010 20th International Conference on Pattern Recognition. Presented at the 2010 20th International Conference on Pattern Recognition (ICPR), IEEE, Istanbul, Turkey, pp. 3756–3759. https://doi.org/10.1109/ICPR.2010.915

Ward, M.P., Rajdev, P., Ellison, C., Irazoqui, P.P., 2009. Toward a comparison of microelectrodes for acute and chronic recordings. Brain research 1282, 183--200.

Washizawa, Y., Higashi, H., Rutkowski, T., Tanaka, T., Cichocki, A., 2010. Tensor Based Simultaneous Feature Extraction and Sample Weighting for EEG Classification, in: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. [Eds.], Neural Information Processing. Models and Applications, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 26–33. https://doi.org/10.1007/978-3-642-17534-3_4

Waterhouse, S.R., Robinson, A.J., 1994. Classification using hierarchical mixtures of experts, in: Proceedings of IEEE Workshop on Neural Networks for Signal Processing. Presented at the Proceedings of IEEE Workshop on Neural Networks for Signal Processing, pp. 177–186. https://doi.org/10.1109/NNSP.1994.366050

Webb, J., Xiao, Z.G., Aschenbrenner, K.P., Herrnstadt, G., Menon, C., 2012. Towards a portable assistive arm exoskeleton for stroke patient rehabilitation controlled through a brain computer interface, in: 2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). Presented at the 2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), pp. 1299–1304. https://doi.org/10.1109/BioRob.2012.6290674

Wei, H., He, J., Tan, J., 2011. Layered hidden Markov models for real-time daily activity monitoring using body sensor networks. Knowl Inf Syst 29, 479–494. https://doi.org/10.1007/s10115-011-0423-3

Wei, Q., Wang, Y., Gao, X., Gao, S., 2007. Amplitude and phase coupling measures for feature extraction in an EEG-based brain–computer interface. J. Neural Eng. 4, 120–129. https://doi.org/10.1088/1741-2560/4/2/012

Wen, D., Jia, P., Lian, Q., Zhou, Y., Lu, C., 2016. Review of Sparse Representation-Based Classification Methods on EEG Signal Processing for Epilepsy Detection, Brain-Computer Interface and Cognitive Impairment. Front. Aging Neurosci. 8. https://doi.org/10.3389/fnagi.2016.00172

Wen, Y., Huang, Z., 2017. Online motor imagery BCI based on adaptive and incremental linear discriminant analysis algorithm, in: 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN). Presented at the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), pp. 962–966. https://doi.org/10.1109/ICCSN.2017.8230254

Whitley, D., 1994. A genetic algorithm tutorial. Stat Comput 4, 65–85. https://doi.org/10.1007/BF00175354

Willett, F.R., Murphy, B.A., Young, D., Memberg, W.D., Blabe, C.H., Pandarinath, C., Franco, B., Saab, J., Walter, B.L., Sweet, J.A., Miller, J.P., Henderson, J.M., Shenoy, K.V., Simeral, J.D., Jarosiewicz, B., Hochberg, L.R., Kirsch, R.F., Ajiboye, A.B., 2018. A Comparison of Intention Estimation Methods for Decoder Calibration in Intracortical Brain–Computer Interfaces. IEEE Transactions on Biomedical Engineering 65, 2066–2078. https://doi.org/10.1109/TBME.2017.2783358

Willett, F.R., Suminski, A.J., Fagg, A.H., Hatsopoulos, N.G., 2013. Improving brain–machine interface performance by decoding intended future movements. J. Neural Eng. 10, 026011. https://doi.org/10.1088/1741-2560/10/2/026011

Williams, J.J., Rouse, A.G., Thongpang, S., Williams, J.C., Moran, D.W., 2013. Differentiating closed-loop cortical intention from rest: building an asynchronous electrocorticographic BCI. J. Neural Eng. 10, 046001. https://doi.org/10.1088/1741-2560/10/4/046001

Williams, N.J., Daly, I., Nasuto, S.J., 2018. Markov Model-Based Method to Analyse Time-Varying Networks in EEG Task-Related Data. Front. Comput. Neurosci. 12. https://doi.org/10.3389/fncom.2018.00076

Wissel, T., Pfeiffer, T., Frysch, R., Knight, R.T., Chang, E.F., Hinrichs, H., Rieger, J.W., Rose, G., 2013. Hidden Markov Model and Support Vector Machine based decoding of finger movements using Electrocorticography. J Neural Eng 10, 056020. https://doi.org/10.1088/1741-2560/10/5/056020

Wodlinger, B., Downey, J.E., Tyler-Kabara, E.C., Schwartz, A.B., Boninger, M.L., Collinger, J.L., 2015. Ten-dimensional anthropomorphic arm control in a human brain−machine interface: difficulties, solutions, and limitations. J. Neural Eng. 12, 016011. https://doi.org/10.1088/1741-2560/12/1/016011

Woehrle, H., Krell, M.M., Straube, S., Kim, S.K., Kirchner, E.A., Kirchner, F., 2015. An Adaptive Spatial Filter for User-Independent Single Trial Detection of Event-Related Potentials. IEEE Transactions on Biomedical Engineering 62, 1696–1705. https://doi.org/10.1109/TBME.2015.2402252

Wold, S., Ruhe, A., Wold, H., Dunn, W.J., III, 1984. The Collinearity Problem in Linear Regression. The Partial Least Squares (PLS) Approach to Generalized Inverses. SIAM J. Sci. Stat. Comput. 5, 735–743. https://doi.org/10.1137/0905052

Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M., 2002. Brain–computer interfaces for communication and control. Clinical Neurophysiology 113, 767–791. https://doi.org/10.1016/S1388-2457(02)00057-3

Wolpaw, J.R., McFarland, D.J., 2004. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. Proc Natl Acad Sci U S A 101, 17849–17854. https://doi.org/10.1073/pnas.0403504101

Wu, Q., Zhang, Y., Liu, J., Sun, J., Cichocki, A., Gao, F., 2019. Regularized Group Sparse Discriminant Analysis for P300-Based Brain–Computer Interface. Int. J. Neur. Syst. 29, 1950002. https://doi.org/10.1142/S0129065719500023

Wu, S.-C., Swindlehurst, A.L., 2018. Direct feature extraction from multi-electrode recordings for spike sorting. Digital Signal Processing 75, 222–231. https://doi.org/10.1016/j.dsp.2018.01.016

Wu, W., Black, M.J., Mumford, D., Gao, Y., Bienenstock, E., Donoghue, J.P., 2004. Modeling and Decoding Motor Cortical Activity Using a Switching Kalman Filter. IEEE Transactions on Biomedical Engineering 51, 933–942. https://doi.org/10.1109/TBME.2004.826666

Wu, X., Zhou, B., Lv, Z., Zhang, C., 2020. To Explore the Potentials of Independent Component Analysis in Brain-Computer Interface of Motor Imagery. IEEE Journal of Biomedical and Health Informatics 24, 775–787. https://doi.org/10.1109/JBHI.2019.2922976

Xie, Z., Schwartz, O., Prasad, A., 2018. Decoding of finger trajectory from ECoG using deep learning. J. Neural Eng. 15, 036009. https://doi.org/10.1088/1741-2552/aa9dbe

Xu, W., Wu, J., Huang, Z., Guan, C., 2005. Kernel based hidden Markov model with applications to eeg signal classification. Presented at the Proceedings of the 3rd IASTED International Conference on Biomedical Engineering 2005, pp. 401–404.

Xu, Y., Wei, Q., Zhang, H., Hu, R., Liu, J., Hua, J., Guo, F., 2019. Transfer Learning Based on Regularized Common Spatial Patterns Using Cosine Similarities of Spatial Filters for Motor-Imagery BCI. Journal of Circuits, Systems and Computers. https://doi.org/10.1142/S0218126619501238

Yanagisawa, T., Hirata, M., Saitoh, Y., Kishima, H., Matsushita, K., Goto, T., Fukuma, R., Yokoi, H., Kamitani, Y., Yoshimine, T., 2012. Electrocorticographic control of a prosthetic arm in paralyzed patients. Annals of Neurology 71, 353–361. https://doi.org/10.1002/ana.22613

Yang, B., Fan, C., Jia, J., Chen, S., Wang, J., 2017. Adaptive KF-SVM Classification for Single Trial EEG in BCI, in: Advanced Computational Methods in Life System Modeling and Simulation, Communications in Computer and Information Science. Springer, Singapore, pp. 35–45. https://doi.org/10.1007/978-981-10-6370-1_4

Yang, C., Qiao, J., Ahmad, Z., Nie, K., Wang, L., 2019. Online sequential echo state network with sparse RLS algorithm for time series prediction. Neural Networks 118, 32–42. https://doi.org/10.1016/j.neunet.2019.05.006

Yang, H., Sakhavi, S., Ang, K.K., Guan, C., 2015. On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).

Presented at the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2620–2623. https://doi.org/10.1109/EMBC.2015.7318929

Yeh, W.-L., Huang, Y.-C., Chiou, J.-H., Duann, J.-R., Chiou, J.-C., 2013. A self produced mother wavelet feature extraction method for motor imagery brain-computer interface, in: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Presented at the 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 4302–4305. https://doi.org/10.1109/EMBC.2013.6610497

Yelisyeyev, A., 2011. Brain-Computer Interface with cortical electrical activity recording [phdthesis]. Université Grenoble Alpes.

Yger, F., Berar, M., Lotte, F., 2017. Riemannian Approaches in Brain-Computer Interfaces: A Review. IEEE Transactions on Neural Systems and Rehabilitation Engineering 25, 1753–1762. https://doi.org/10.1109/TNSRE.2016.2627016

Yoon, J.W., Roberts, S.J., Dyson, M., Gan, J.Q., 2008. Adaptive Classification by Hybrid EKF with Truncated Filtering: Brain Computer Interfacing, in: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. [Eds.], Intelligent Data Engineering and Automated Learning – IDEAL 2008, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 370–377. https://doi.org/10.1007/978-3-540-88906-9_47

Young, D., Willett, F., Memberg, W.D., Murphy, B., Rezaii, P., Walter, B., Sweet, J., Miller, J., Shenoy, K.V., Hochberg, L.R., Kirsch, R.F., Ajiboye, A.B., 2019. Closed-loop cortical control of virtual reach and posture using Cartesian and joint velocity commands. J Neural Eng 16, 026011. https://doi.org/10.1088/1741-2552/aaf606

Yousefi, R., Sereshkeh, A.R., Chau, T., 2019. Development of a robust asynchronous brain-switch using ErrP-based error correction. J. Neural Eng. 16, 066042. https://doi.org/10.1088/1741-2552/ab4943

Yu, B.M., Kemere, C., Santhanam, G., Afshar, A., Ryu, S.I., Meng, T.H., Sahani, M., Shenoy, K.V., 2007. Mixture of Trajectory Models for Neural Decoding of Goal-Directed Movements. Journal of Neurophysiology 97, 3763–3780. https://doi.org/10.1152/jn.00482.2006

Yu, J.-M., Cho, S.-B., 2016. Prediction of Bank Telemarketing with Co-training of Mixture-of-Experts and MLP, in: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. [Eds.], Neural Information Processing, Lecture Notes in Computer Science. Springer International Publishing, pp. 52–59.

Yuksel, S.E., Wilson, J.N., Gader, P.D., 2012. Twenty Years of Mixture of Experts. IEEE Transactions on Neural Networks and Learning Systems 23, 1177–1193. https://doi.org/10.1109/TNNLS.2012.2200299

Yun Gao, Black, M.J., Bienenstock, E., Wei Wu, Donoghue, J.P., 2003. A quantitative comparison of linear and non-linear models of motor cortical activity for the encoding and decoding of arm motions, in: First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings. Presented at the First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings., pp. 189–192. https://doi.org/10.1109/CNE.2003.1196789

Zeng, H., Sun, Y., Xu, G., Wu, C., Song, A., Xu, B., Li, H., Hu, C., 2019. The Advantage of Low-Delta Electroencephalogram Phase Feature for Reconstructing the Center-Out Reaching Hand Movements. Front. Neurosci. 13. https://doi.org/10.3389/fnins.2019.00480

Zhang, A., Yang, B., Huang, L., 2008. Feature Extraction of EEG Signals Using Power Spectral Entropy, in: 2008 International Conference on BioMedical Engineering and Informatics. Presented at the 2008 International Conference on BioMedical Engineering and Informatics, pp. 435–439. https://doi.org/10.1109/BMEI.2008.254

Zhang, X., Wang, H., Tian, Y., Peyrodie, L., Wang, X., 2018. Model-free based neural network control with time-delay estimation for lower extremity exoskeleton. Neurocomputing 272, 178–188. https://doi.org/10.1016/j.neucom.2017.06.055

Zhang, Y., Zhou, G., Jin, J., Wang, M., Wang, X., Cichocki, A., 2013. L1-Regularized Multiway Canonical Correlation Analysis for SSVEP-Based BCI. IEEE Transactions on Neural Systems and Rehabilitation Engineering 21, 887–896. https://doi.org/10.1109/TNSRE.2013.2279680

Zhang, Y., Zhou, G., Jin, J., Wang, X., Cichocki, A., 2015. Optimizing spatial patterns with sparse filter bands for motor-imagery based brain–computer interface. Journal of Neuroscience Methods 255, 85–91. https://doi.org/10.1016/j.jneumeth.2015.08.004

Zhang, Yu, Zhou, G., Jin, J., Wang, X., Cichocki, A., 2013. Frequency recognition in ssvep-based bci using multiset canonical correlation analysis. Int. J. Neur. Syst. 24, 1450013. https://doi.org/10.1142/S0129065714500130

Zhang, Yu, Zhou, G., Jin, J., Zhang, Yangsong, Wang, X., Cichocki, A., 2017. Sparse Bayesian multiway canonical correlation analysis for EEG pattern recognition. Neurocomputing 225, 103–110. https://doi.org/10.1016/j.neucom.2016.11.008

Zhang, Z., Li, X., Deng, Z., 2010. A CWT-based SSVEP classification method for brain-computer interface system, in: 2010 International Conference on Intelligent Control and Information Processing. Presented at the 2010 International Conference on Intelligent Control and Information Processing, pp. 43–48. https://doi.org/10.1109/ICICIP.2010.5564336

Zhao, L., Wang, X., 2015. A multi-modal BCI system based on motor imagery, in: 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI). Presented at the 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI), pp. 137–141. https://doi.org/10.1109/BMEI.2015.7401488

Zhao, Q., Caiafa, C.F., Mandic, D.P., Chao, Z.C., Nagasaka, Y., Fujii, N., Zhang, L., Cichocki, A., 2013a. Higher Order Partial Least Squares (HOPLS): A Generalized Multilinear Regression Method. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1660–1673. https://doi.org/10.1109/TPAMI.2012.254

Zhao, Q., Liqing Zhang, Cichocki, A., Jie Li, 2008. Incremental Common Spatial Pattern algorithm for BCI, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). Presented at the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 2656–2659. https://doi.org/10.1109/IJCNN.2008.4634170

Zhao, Q., Zhou, G., Adali, T., Zhang, L., Cichocki, A., 2013b. Kernelization of Tensor-Based Models for Multiway Data Analysis: Processing of Multidimensional Structured Data. IEEE Signal Processing Magazine 30, 137–148. https://doi.org/10.1109/MSP.2013.2255334

Zhu, C., Sheng, W., 2009. Online hand gesture recognition using neural network based segmentation, in: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. Presented at the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2415–2420. https://doi.org/10.1109/IROS.2009.5354657

# Innovative decoding algorithms for Chronic ECoG-based Brain Computer Interface (BCI) for motor disabled subjects in laboratory and at home

Brain-computer interfaces (BCIs) are systems that allow the control of external devices from the brain neural signal recordings without neuromuscular activation. Among the various applications, functional compensation and rehabilitation of individuals suffering from severe motor disabilities (motor BCIs) has always been a focus for BCI research. Relying on the "BCI and Tetraplegia" clinical trial of CEA/LETI/CLINATEC, the present doctoral thesis aim to address the challenges related to motor BCIs control of multi-limb effectors namely asynchronous multi-limb BCI and online closed-loop decoder adaptation. The algorithm Recursive exponentially weighted Markov switching multi-linear model (REW-MSLM) integrating an online incremental closed-loop decoder adaptation procedure was designed to handle high dimensional multi-limb effector control. REW-MSLM is based on a Mixture of Experts (ME) architecture composed of several continuous "expert" models decoding continuous movements from the neural signals and a dynamic "gating" model activating or inhibiting the expert continuous outputs. The continuous expert models were evaluated using the Recursive Exponentially Weighted N-way Partial Least Squares (REW-NPLS) algorithm whereas the gating model is a Hidden Markov Model (HMM). REW-MSLM allows a tetraplegic patient, who underwent bilateral epidural electrocorticographic arrays (ECoG) implantation of chronic wireless implants (WIMAGINE), to perform the 8D control of a whole body exoskeleton over 6 months without model recalibrations. During this period, the patient was able to perform alternative 3D left and right hand translations and 1D left and right wrist rotations. For the experiments carried out from 0 to 37 days after the last model calibration experiments, the decoding performance highlighted a hit score of $71 \pm 12\%$ and $99 \pm 2\%$ for the 3D hand translation and 1D wrist rotation tasks whereas the dynamic classifier showed a five-state classification F-score performance of $77 \pm 14\%$. For the experiments performed from 0 to 167 days, the decoding performance highlighted hit scores of $67 \pm 21\%$ and $93 \pm 12\%$ for the hand translation and wrist rotation tasks whereas the dynamic classifier demonstrated a five-state classification F-score performance of $75 \pm 12\%$. Additionally, other algorithms were proposed and tested offline in order to be potentially integrated into the REW-MSLM, namely the (Automatic) Penalized REW-NPLS algorithms (PREW-NPLS and APREW-NPLS) and a version of Hierarchical Hidden Markov Model (H2M2). PREW-NPLS is an incremental adaptive regularized NPLS algorithm promoting group-wise sparse solutions using $L_p$ (p=0,0.5,1) norm/pseudo-norm penalization. APREW-NPLS is an automatic version of PREW-NPLS. It compares, based on reinforcement learning strategies, several penalized models with different penalization hyperparameters during the closed-loop experiments in order to optimize the best penalization hyperparameter in real-time. Both PREW-NPLS and APREW-NPLS algorithms were designed to reduce the feature space dimension and to improve the decoding performance. H2H2 is a dynamic classifier with a hierarchical architecture designed to promote decoding responsiveness and low latency state transitions. The proposed algorithms allowed performing asynchronous online direct neural decoding from epidural ECoG population recording system. Such results may open new application perspectives.

**Keywords:** Brain Computer interface, BCI, ECoG, Clinical trial, Asynchrone, Adaptive, Closed-loop, Online, Penalization, Hidden Markov Models, Machine learning, Brain signal processing, Tetraplegia.

# Une interface cerveau-machine basée sur des algorithmes de décodage innovants pour le contrôle d'effecteurs complexes en vue d'un usage au quotidien par des patients en situation de handicap moteur

Les interfaces cerveau-Machine (ICM) sont des systèmes permettant de traduire l'activité cérébrale d'un individu via un ordinateur afin d'effectuer des tâches nécessitant normalement une action des nerfs périphériques et/ou musculaires. En se basant sur l'essai clinique de CEA/LETI/CLINATEC, nommé « BCI et Tétraplégie », les recherches de thèse présentées dans ce manuscrit répondent aux challenges de l'intégration ICM dans la vie quotidienne, à savoir le contrôle complexe asynchrone d'effecteurs multi-membres et l'adaptation en temps réel des algorithmes de décodage durant des expériences en closed-loop. Pour répondre aux challenges de l'essai clinique, l'algorithme incrémental adapatif en temps réel Recursive exponentially weighted Markov switching multi-linear model (REW-MSLM) a été créé. L'algorithme REW-MSLM repose sur une architecture du type Mixture d'Experts (ME). Les MEs combinent plusieurs décodeurs continus dit «experts » dont les prédictions sont pondérées par un modèle discret nommé «gate » . Les modèles des experts sont estimés via l'algorithme REW-NPLS tandis que le modèle de gate est un modèle de markov caché (HMM). L'algorithme REW-MSLM a permis à un patient tétraplégique de contrôler un exosquelette 4 membres avec 8 degrés de liberté via le décodage de signaux électrocorticographiques (ECoG) enregistrés avec deux implants épiduraux sans fil nommé WIMAGINE. Dans une même expérience, avec un même modèle, le patient a réalisé des tâches de contrôle alternatif des mouvements du bras gauche et du bras droit dans l'espace 3D et de rotation 1D des poignets gauche et droit. Le contrôle du patient durant ces tâches est resté très stable, même durant des expériences 6 mois après la fin de la mise à jour du modèle. Par exemple, les performances de contrôle réalisées 0 à 37 jours après la dernière mise à jour du modèle ont montré un score de réussite de $71 \pm 12\%$ et $99 \pm 2\%$ pour les tâches de contrôle 3D des mouvements des mains et les tâches de contrôle 1D de rotation des poignets. Les performances de contrôle réalisées de 0 à 167 jours après la dernière mise à jour du modèle ont montré un score de réussite de $67 \pm 21\%$ et $93 \pm 12\%$ pour les tâches de contrôle 3D des mouvements des mains et les tâches de contrôle 1D de rotation des poignets. Dans le but d'être intégré dans REW-MSLM, de nouveaux algorithmes, à savoir le (Automatic) Penalized Recursive Exponentially Weighted N-way Partial Least Squares (PREW-NPLS et APREW-NPLS) et une version du Modèle de Markov Caché Hiérarchique (H2M2), ont été imaginés et testés en offline. $L_p$-PREW-NPLS est un algorithme incrémental adaptatif permettant une pénalisation par groupes du modèle de décodage suivant la norme/pseudo norme $L_p$=0, 0.5 ou 1. Pour une utilisation en temps réel, l'algorithme PREW-NPLS nécessite de déterminer avec une étude offline préliminaire l'hyperparamètre de pénalisation optimal. APREW-NPLS permet de comparer en temps réel plusieurs modèles avec des hyperparamètres de pénalisation différents afin de l'optimiser en temps réel durant l'expérience. Les algorithmes PREW-NPLS et APREW-NPLS ont été créés pour réduire la dimension de l'espace des caractéristiques et améliorer les performances de décodage. H2M2, quant à lui est un classifieur dynamique similaire aux modèles de type HMM mais avec une structure hiérarchique. La structure hiérarchique est répartie en couches avec les états des couches inférieures dépendants des états des couches supérieures. L'algorithme H2M2 a été conçu dans le but d'améliorer la réactivité du modèle de classification de REW-MSLM (gating). Les résultats de décodage neural direct des signaux épiduraux ECoG obtenus nous poussent à diversifier l'utilisation de ces algorithmes à d'autres domaines.

**Mots-clés :** Interface Cerveau-Machine, ICM, ECoG, Essai clinique, Asynchrone, Adaptatif, Closed-loop, Temps réel, Pénalisation, Modèles de Markov Cachés, Apprentissage automatique, Traitements des signaux cérébraux, Tétraplégie.

Université
Grenoble Alpes