



**HAL**  
open science

# Adaptation of discrete and continuous intracranial Brain-Computer Interfaces using neural correlates of task performance decoded continuously from the sensorimotor cortex of a tetraplegic.

Vincent Rouanne

► **To cite this version:**

Vincent Rouanne. Adaptation of discrete and continuous intracranial Brain-Computer Interfaces using neural correlates of task performance decoded continuously from the sensorimotor cortex of a tetraplegic.. Bioinformatics [q-bio.QM]. Université Grenoble Alpes [2020-..], 2022. English. NNT : 2022GRALS016 . tel-03917556

**HAL Id: tel-03917556**

**<https://theses.hal.science/tel-03917556>**

Submitted on 2 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## THÈSE

Pour obtenir le grade de

### DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

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

Arrêté ministériel : 25 mai 2016

Présentée par

**Vincent ROUANNE**

Thèse dirigée par **Tetiana AKSENOVA**

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

**Adaptation d'Interfaces Cerveau-Machines discrètes et continues grâce à des corrélats neuronaux de performance de tâche détectés continuellement dans le cortex sensorimoteur d'un tétraplégique**

**Adaptation of discrete and continuous intracranial Brain-Computer Interfaces using neural correlates of task performance decoded continuously from the sensorimotor cortex of a tetraplegic.**

Thèse soutenue publiquement le **29 juin 2022**,  
devant le jury composé de :

**Madame TETIANA AKSENOVA**

Ingénieur HDR, CEA CENTRE DE GRENOBLE, Directrice de thèse

**Monsieur FABIEN LOTTE**

Directeur de recherche, INRIA CENTRE BORDEAUX SUD-OUEST,  
Rapporteur

**Monsieur ANTOINE SOULOUMIAC**

Ingénieur HDR, CEA CENTRE DE PARIS-SACLAY, Rapporteur

**Monsieur STEPHAN CHABARDES**

Professeur des Univ. - Praticien hosp., UNIVERSITE GRENOBLE ALPES, Examineur

**Monsieur FRANÇOIS CABESTAING**

Professeur des Universités, UNIVERSITE DE LILLE, Président



# Abstract

Brain-computer interfaces (BCIs) transform neural signals into commands for effectors. They are mainly used as tools for functional compensation of impaired functions in disabled subjects. The Clinattec clinical trial “BCI and tetraplegia” aims at providing a proof of concept of long-term functional compensation of upper and lower limb motor deficits in tetraplegics using a BCI (motor BCI). The clinical trial showed promising results for compensation of motor functions. However, the BCI’s usability remains to be improved. The decoder used to estimate the user’s intention from the processed neural signal (control decoder) must be fitted before the BCI can be used. This is done during dedicated training sessions, during which the user is directed to perform specific motor imagery tasks. Training sessions are downtime for the end-goal of the BCI, as the user does not freely decide which actions to perform. Generally, training sessions have to be held regularly in order to update or retrain the control decoder due to degradation of performance over time. In the scope of improving usability of BCIs, this PhD research proposes to limit the negative impact of training sessions. In this work, the control decoder is trained and updated using inferred labels instead of labels acquired during a dedicated training session. The labels are inferred using the output of the control decoder and neural correlates of task performance (i.e., how well the effector’s actions match the user’s intentions). We call such a BCI an “auto-adaptive BCI”. Although the decoder responsible for the detection of these neural correlates of task performance is trained during a dedicated training session, it can still provide a net improvement in usability, depending on the complexity of the control decoder trained and the relative stability in time and across tasks of the task performance decoder. However, in order to be usable for state-of-the-art motor BCIs the adaptation process must be possible for control decoders with multiple discrete or continuous outputs. We argue that adaptation of a control decoder with multiple continuous outputs is best done using neural correlates of task performance that can be decoded continuously in time. Additionally, in order to be usable with current state-of-the-art motor BCI, these neural correlates should be detected in the sensorimotor cortex due to the position of the implanted neural acquisition system. Using multiple datasets from a tetraplegic enrolled in the Clinattec BCI clinical trial, we first show that it is possible to detect such continuous in time neural correlates of task performance from the sensorimotor cortex. We then show that control decoder labels can be inferred using these neural correlates. Finally, we perform an offline simulation of online use to demonstrate that the auto-adaptive BCI can be used to successfully train decoders for discrete or continuous control.

This new approach was tested on ECoG datasets from a tetraplegic enrolled in the Clinattec BCI clinical trial. The subject used motor imagery to control a binary avatar (Runner MI dataset), a virtual exoskeleton with four discrete motor states (Exo dataset) or a hand-shaped cursor on a two-dimensional screen (Cursor dataset). In five-fold cross-validations, the mean and standard deviation of the area under the receiver operating characteristic curve (AUC of the ROC) for the decoder of neural correlates of task performance were  $0.6225 \pm 0.0429$  in the Runner MI dataset,  $0.5677 \pm 0.0427$  in the Exo dataset and  $0.6570 \pm 0.0188$  in the Cursor dataset. In a pseudo three-fold cross-validation simulating online use, the accuracy of the estimated labels was 64.9% in the Runner dataset and 64.5% in the Exo dataset. In the Cursor dataset, 63.3% of the estimated labels were less than  $60^\circ$  away from the actual labels. The AUC of the ROC of the control decoder was  $0.6360 \pm 0.0958$  in the Runner MI dataset when trained auto-adaptively compared to  $0.8958 \pm 0.0153$  when trained in a classical supervised manner, and a chance level of  $0.5007 \pm 0.0691$ . The multiclass generalization of the AUC of the ROC of the control decoder was  $0.7595 \pm 0.0278$  in the Exo dataset when trained auto-adaptively compared to  $0.8177 \pm 0.0301$  when trained in a classical supervised manner, and a chance level of



## Abstract

0.5163±0.0580. In the Cursor dataset, the cosine similarity between the output of control decoder and the ideal trajectory was 0.1589±0.0668 when trained auto-adaptively compared to 0.2107±0.0664 when trained in a classical supervised manner, and a chance level of -0.0231±0.0327.

These results first show that continuous in time neural correlates of task performance are detectable in ECoG recordings of the sensorimotor cortex, and then that it is possible to use them to train auto-adaptively control decoders that have multiple discrete or continuous outputs, with performances significantly better than chance levels. Although these findings are promising for the future of adaptive complex motor BCIs, they should first be replicated outside of this case study and during online use of the BCI. Perspective future study also includes evaluation of the robustness in time and to tasks of the task performance decoder, as well as adaptation of even more complex motor BCIs.

# Table of contents

CHAPTER 1: INTRODUCTION TO BRAIN-COMPUTER INTERFACES (BCIS).....	1
<b>I. Neural signals for BCI systems</b> .....	<b>1</b>
<b>II. Components of BCI systems</b> .....	<b>3</b>
II.1. Acquisition systems .....	3
II.2. Effectors.....	6
II.3. Transducers .....	6
<b>III. BCIs for functional compensation</b> .....	<b>8</b>
CHAPTER 2: SCIENTIFIC CONTEXT AND OBJECTIVES.....	11
<b>I. Scientific context: the Clinattec BCI project</b> .....	<b>11</b>
I.1. Clinical trial goals .....	11
I.2. Recording system of the Clinattec clinical trial: the WIMAGINE ECoG implant.....	12
I.3. Effectors of the Clinattec BCI clinical trial .....	13
I.4. Software chain .....	14
<b>II. Clinical trial progress</b> .....	<b>15</b>
II.1. Inclusion criteria and subjects enrolled .....	15
II.2. Implantation .....	15
II.3. Progressive training of subjects enrolled in the clinical trial .....	16
<b>III. Objectives of the thesis</b> .....	<b>16</b>
CHAPTER 3: AUTO-ADAPTIVE BCIS AS SOLUTION TO REDUCE BCI DECODER TRAINING .....	19
<b>I. Auto-adaptive BCIs based on unsupervised adaptation of control decoders</b> .....	<b>19</b>
<b>II. Auto-adaptive BCIs based on weakly supervised adaptation of control decoders using neural correlates of task performance</b> .....	<b>21</b>
II.1. The Error-related Potential (ErrP).....	21
II.2. ErrP based auto-adaptation.....	22
II.3. Auto-adaptation of control decoders with multiple continuous outputs .....	23
II.4. Detection of neural correlates of TP in the sensorimotor cortex.....	25
<b>III. Proposed mechanism of auto-adaptation for state-of-the-art BCIs</b> .....	<b>26</b>
CHAPTER 4: EXPERIMENTAL DESIGN .....	27
<b>I. Data acquisition</b> .....	<b>27</b>
<b>II. Datasets</b> .....	<b>29</b>
II.1. Datasets using a binary discrete effector .....	30
II.2. Dataset using a multi-class discrete effector.....	33
II.3. Dataset using a bi-dimensional continuous effector .....	34
<b>III. Decoder training for data acquisition</b> .....	<b>34</b>
CHAPTER 5: DETECTION OF NEURAL CORRELATES OF TASK PERFORMANCE .....	37
<b>I. Methods</b> .....	<b>37</b>

## Table of contents

I.1. Neural feature extraction for single-trial detection of neural correlates of MTP .....	37
I.2. Data labeling .....	38
I.3. Data balance .....	41
I.4. Decoders for the detection of neural correlates of task performance.....	42
I.5. Performance measure for single trial detection of neural correlates of MTP .....	44
I.6. Population response to erroneous and correct events .....	45
<b>II. Results .....</b>	<b>46</b>
II.1. Visualization of population response to correct and erroneous events.....	46
II.2. Detection of event-locked neural correlates of motor task performance .....	50
II.3. Detection of continuous in time neural correlates of motor task performance .....	52
II.4. Impact of NPLS factor numbers on the performance of detection of neural correlates of cMTP.....	54
II.5. Relative feature importance .....	55
<b>III. Discussion .....</b>	<b>60</b>
III.1. Event-locked population response to correct and erroneous events .....	60
III.2. Relative detectability of event-locked versus continuous-in-time neural correlates of MTP .....	62
III.3. Neural correlates of task performance during BCI observation or BCI control with motor imagery .....	62
III.4. Relative feature importance .....	63
III.5. Temporal stability of neural correlates of cMTP in the sensorimotor cortex.....	64
<b>CHAPTER 6: AUTO-ADAPTIVE BCI DESIGN USING NEURAL CORRELATES OF TASK PERFORMANCE FOR REAL-TIME LABELING .....</b>	<b>65</b>
<b>I. Methods.....</b>	<b>65</b>
I.1. Pseudo-online simulation .....	65
I.2. Formation of datasets for cMTP decoders training .....	65
I.3. Formation of datasets for the training of control decoders in the auto-adaptive BCI framework .....	65
I.4. Formation of datasets for the training of control decoders for comparison with the aaBCI framework ..	68
I.5. Algorithm for the training of control decoders in the aaBCI framework.....	69
I.6. Performance evaluation .....	69
<b>II. Results .....</b>	<b>70</b>
II.1. Performance of the cMTP decoders on the full control decoder training dataset.....	70
II.2. Impact of the trade-off parameter $\alpha$ on the cMTP decoder performances .....	71
II.3. Accuracy of estimated labels for the control decoder.....	73
II.4. Performance of control decoders trained using the aaBCI.....	74
<b>III. Discussion .....</b>	<b>76</b>
III.1. aaBCI performances.....	76
III.2. Parametrization of the aaBCI.....	77
III.3. Label estimation for the control decoder .....	78
<b>CHAPTER 7: LIMITATIONS AND PERSPECTIVES.....</b>	<b>81</b>
<b>SUPPLEMENTARY DATA .....</b>	<b>85</b>
<b>I. Electrode set selection preliminary study.....</b>	<b>85</b>
<b>II. Motor imagery decoders trained on the same epochs as the MTP decoders.....</b>	<b>87</b>
<b>COMMUNICATIONS, PUBLICATIONS AND AWARDS.....</b>	<b>92</b>
<b>BIBLIOGRAPHY.....</b>	<b>93</b>

# Résumé en français

Adaptation d'Interfaces Cerveau-Machines discrètes et continues grâce à des corrélats neuronaux de performance de tâche détectés continuellement dans le cortex sensorimoteur d'un tétraplégique.

## Chapitre 1 : Introduction aux Interfaces Cerveau-Machines (ICMs)

Cette thèse porte sur les interfaces cerveau machines. Les interfaces cerveau machines (ICMs) sont des dispositifs qui utilisent des signaux cérébraux afin de décoder les intentions de leur utilisateur. Ces ICMs sont principalement développées pour des individus ayant des déficits moteur, tels que des lésions de la colonne vertébrale, la sclérose amyotrophique latérale ou des accidents vasculaires cérébraux (Lebedev and Nicolelis, 2017). Pour ces personnes, les ICMs peuvent remplacer partiellement les fonctions motrices perdues, telles que la communication, la locomotion, ou la préhension. En raison de leur complexité, les ICMs ne peuvent détecter qu'un nombre limité de commandes présélectionnées. Ces commandes sont générées en effectuant des tâches mentales qui génèrent des signaux cérébraux spécifiques. Certains signaux cérébraux sont régulièrement utilisés dans le domaine des ICMs :

- Les **rythmes sensorimoteurs** sont des variations dans l'amplitude des oscillations neuronales dans certaines bandes de fréquences, générées par des mouvements exécutés, imaginés ou tentés (par exemple si l'utilisateur est handicapé) et sont principalement utilisés dans les ICMs qui ont pour but de restaurer les fonctions motrices.
- Le **P300** est un potentiel électrique élicité environ 300ms après qu'un stimulus ait été détecté (Nicolas-Alonso and Gomez-Gil, 2012). Il est principalement utilisé dans les ICMs qui visent à restaurer la communication chez leur utilisateur.
- Les **potentiels évoqués stables** sont des modulations fréquentielles dans les signaux cérébraux élicitées par un stimulus présenté à une fréquence constante, qu'il soit visuel, auditif ou tactile.
- Les **potentiels liés aux erreurs** (ErrP) sont des potentiels qui apparaissent dans les signaux cérébraux après que l'utilisateur ait commis ou observé un agent commettre une erreur.

Chacun de ces signaux a des avantages et inconvénients, et le signal utilisé pour contrôler une ICM doit être choisi en prenant en compte l'application de l'ICM ainsi que les préférences de l'utilisateur.

Les interfaces cerveau-machine sont composées de trois blocs principaux : un système d'acquisition, un traducteur et un ou plusieurs effecteurs. Le système d'acquisition échantillonne, amplifie et digitalise l'activité cérébrale de l'utilisateur. Le traducteur transforme les signaux cérébraux acquis en une estimation de l'intention de l'utilisateur. Les effecteurs réalisent les actions estimées voulues par l'utilisateur. Les différents systèmes d'acquisition les plus utilisés dans le domaine des ICMs sont l'**électroencéphalographie** (EEG), l'**électrocorticographie** (ECoG), les grilles de **micro-électrodes** (MEAs), la **magnétoencéphalographie** (MEG), l'**imagerie spectroscopique proche infra-rouge fonctionnelle** (fNIRS) et l'**imagerie par résonance magnétique fonctionnelle** (fMRI). Les ICMs qui visent à restaurer la motricité (ICM motrices) chez les handicapés utilisent souvent des systèmes d'acquisition invasifs (MEA et ECoG) (Benabid et al., 2019; Willett et al., 2021; Wodlinger et al., 2014) en raison de leur meilleure qualité de signal. Les ICMs visant des individus sains reposent principalement sur l'EEG en raison de son faible coût et de sa portabilité. Une grande diversité d'effecteurs existe pour les ICMs. Pour les ICMs motrices, il est par exemple possible d'utiliser un bras robotique, un exosquelette (Benabid et al., 2019) ou même de la stimulation électrique fonctionnel

pour activer les membres du patient. Pour les ICMs de communication l'effecteur le plus courant est un ordinateur qui affiche les caractères choisis afin d'écrire. Le traducteur est au cœur de l'ICM de par sa fonction de traduction des signaux cérébraux en commande. Plusieurs étapes existent dans le bloc du traducteur : le **pré-traitement**, l'**extraction de caractéristiques**, le **décodage** et le **post-traitement**. Le pré-traitement améliore la qualité du signal fourni par le système d'acquisition. L'extraction de caractéristiques transforme le signal pré-traité en information considérée optimale pour le décodage de l'intention de l'utilisateur. Les caractéristiques extraites sont transformées en prédiction de l'intention de l'utilisateur par un décodeur (décodeur de contrôle). Le décodage est généralement effectué au moyen d'un décodeur (décodeur de contrôle) dont. Les paramètres du décodeur sont généralement estimés grâce à l'apprentissage machine supervisé. L'apprentissage machine supervisé requiert des données d'entraînement labélisées, c'est-à-dire des données cérébrales pour lesquels on connaît l'intention de l'utilisateur. Ces données labélisées sont obtenues lors de sessions d'entraînement, durant lesquelles l'utilisateur de l'ICM doit effectuer des tâches spécifiques. Le post-traitement est une étape optionnelle appliquée à la sortie du décodeur pour améliorer la qualité des prédictions.

Les ICMs pour la compensation fonctionnelle de déficits moteurs devraient idéalement avoir les caractéristiques suivantes (Wolpaw and Wolpaw, 2012):

- Sans danger pour l'utilisateur.
- Activable et utilisable à tout moment par l'utilisateur sans assistance de la part d'aide de soin, technicien ou scientifique.
- Utilisable en dehors des laboratoires.
- Offre une compensation fonctionnelle équivalente aux capacités des individus sains.
- Ne demande pas plus d'effort à utiliser que la capacité compensée demanderait à un individu sain.
- Peut être utilisé de manière chronique, ne demandant que peu de modifications au fil des ans.
- Bon marché et transportable facilement.
- Esthétiquement acceptable ou invisible.

Une majeure partie de ces propriétés reste non atteintes ou partiellement atteintes par les ICMs actuelles. En fonction de chaque ICM, le bon choix de système d'acquisition, traducteur et effecteur doit être fait pour maximiser la satisfaction de leur utilisateur et effectuer un compromis entre les différentes propriétés désirables décrites ci-dessus.

## Chapitre 2 : Contexte scientifique et objectifs

Cette thèse s'inscrit dans le contexte d'un essai clinique d'ICM chronique. L'essai clinique « BCI et tétraplégie » se déroule à Clinatex et a été accepté par l'Agence nationale de sécurité du médicament et des produits de santé (numéro 2015-A00650-49) et le Comité de Protection des personnes (15-CHUG-19). Le but de cet essai clinique est la réalisation d'une preuve de concept que des patients tétraplégiques peuvent contrôler des effecteurs moteurs très complexes, tel qu'un exosquelette 4 membres, afin de promouvoir une restauration fonctionnelle des capacités motrices des patients tétraplégiques de façon chronique. L'électrocorticographie (ECoG) est utilisé comme système d'acquisition malgré le fait qu'il soit invasif, en raison de la qualité de signal supérieure aux méthodes non invasives. Un implant ECoG a été développé spécifiquement pour cet essai clinique (Mestais et al., 2015). L'implant ECoG WIMAGINE possède 64 électrodes et enregistre avec une fréquence d'échantillonnage de 585Hz, fonctionne sans fil, et est adapté à une utilisation chronique dans le domaine médical. Cependant, l'implant est actuellement en mesure de transmettre des données de 34 électrodes à la fois seulement. Un exosquelette quatre membres a aussi été développé pour

permettre la restauration de la mobilité. D'autres effecteurs sont aussi utilisés dans le cadre de cet essai clinique, tel qu'un fauteuil roulant motorisé, ainsi que des effecteurs virtuels affichés sur écran d'ordinateur. Ces effecteurs virtuels ont l'avantage d'être utilisables facilement chez leurs utilisateurs ainsi que d'être moins complexes à créer, permettant d'avoir plus facilement un panel d'effecteurs de niveau de difficultés de contrôle variés. Les décodeurs de contrôle utilisés dans cet essai clinique sont entraînés grâce à l'algorithme REW-NPLS avec validation récursive des hyperparamètres (Eliseyev et al., 2017). Un modèle de Markov caché est aussi utilisé pour stabiliser les différents états moteurs activable de l'exosquelette (par exemple le mouvement du bras droit vs le mouvement du bras gauche) (Schaeffer and Aksenova, 2016).

Jusqu'à présent, trois patients ont été inclus dans l'essai clinique. Les implants du premier patient ont cessé de fonctionner rapidement après leur implantation. Les implants ont été explantés et le patient a été retiré de l'essai clinique. Le second patient a été implanté en Juin 2017, et a participé à l'essai clinique jusqu'à mi 2021. Un troisième patient a été implanté en Novembre 2019 et est actuellement toujours inclus dans l'essai clinique. Chaque patient a été implanté avec deux implants WIMAGINE, un au-dessus de chaque cortex sensorimoteur (droit et gauche). Le sujet dans cette thèse est le deuxième patient inclus dans l'essai clinique « BCI et tétraplégie ». C'est un homme de 28 ans qui est tétraplégique en raison d'une lésion de la colonne vertébrale au niveau C4-C5.

Les résultats publiés de l'essai clinique sont prometteurs, avec la possibilité pour l'un des patients de contrôler l'exosquelette en 8 dimensions continues (positions des deux mains en 3D et pronosupination des deux poignets) après un entraînement progressif. Bien que le patient ait pu contrôler l'exosquelette sur une période de plusieurs mois sans nécessiter de mise à jour du décodeur de contrôle, une légère baisse de performance a pu être observée avec le temps. L'entraînement de ce décodeur a nécessité six sessions d'entraînement, réparties sur deux semaines, pour un total de trois heures et demi de données dans le set d'entraînement.

L'objectif de cette thèse est d'améliorer les ICMs en général en limitant la nécessité des sessions d'entraînements pour l'entraînement du décodeur de contrôle. Les sessions d'entraînement sont un temps mort dans l'utilisation libre de l'ICM, et doivent de plus être effectuées régulièrement afin de garder à jour le décodeur de contrôle. Le but est ici d'obtenir des données labélisées en dehors des sessions d'entraînement. Pour ce faire, il est proposé dans cette thèse d'estimer les labels en se basant sur les données neuronales acquises pendant le contrôle de l'ICM. Plus précisément, les labels sont estimés grâce à des corrélats neuronaux de **performance de tâche motrice (MTP)**. Ces signaux neuronaux reflètent l'adéquation entre les actions effectuées par l'ICM et les intentions de l'utilisateur. **Le but de cette thèse est d'effectuer une preuve de concept montrant que des corrélats neuronaux de MTP peuvent être détectés dans le cortex sensorimoteur et utilisés afin de permettre l'adaptation en temps réel du décodeur de contrôle d'une ICM motrice complexe** (contrôlant au moins deux degrés de liberté de manière continue).

### Chapitre 3 : Les ICMs auto-adaptatives en tant que solution pour réduire les entraînements de décodeurs de contrôle

Utiliser des signaux cérébraux afin d'estimer les labels nécessaires à l'entraînement ou à la mise à jour du décodeur de contrôle a déjà été fait par le passé. La stratégie la plus répandue actuellement consiste à considérer que la sortie du décodeur de contrôle est correcte et à l'utiliser en tant que label. Ce type d'entraînement du décodeur de contrôle est usuellement appelé entraînement non-supervisé dans le domaine des ICMs. Parmi les études utilisant cette stratégie, on citera celles de Shenoy et al. (2006), Vidaurre et al. (2011b, 2011a), Gu et al. (2013), Kindermans et al. (2014) et Li et al. (2011). Chacune de ces études rapporte une amélioration des performances des décodeurs de contrôle mis à

jour de manière non-supervisée comparé aux décodeurs de contrôle non mis à jour. Cependant, nous estimons que la mise à jour non supervisée de décodeurs de contrôle n'est pas optimale pour des ICMs à usage chronique. En effet, un cercle vicieux peut s'enclencher si les performances du décodeur de contrôle venaient à baisser car les labels estimés sont basés sur la sortie du décodeur de contrôle. Les labels estimés seraient alors moins corrects, ce qui pourraient encore détériorer les performances du décodeur de contrôle après sa mise à jour suivante. Pour cette même raison, l'entraînement non-supervisé est difficilement utilisable avec des décodeurs de contrôle initialisés de manière aléatoire. Enfin, l'adaptation non-supervisée n'est probablement pas optimale pour suivre les variations de concept brusques dans les données d'entrée du décodeur de contrôle car ces variations pourraient être trop rapide pour qu'une adaptation puisse se faire avant que les performances du décodeur de contrôle n'aient baissé de manière critique. Pour ces raisons, nous suggérons que l'estimation des labels pour le décodeur de contrôle doit être indépendante des performances du décodeur de contrôle.

L'idée d'utiliser des corrélats neuronaux de performance de tâche est aussi présente dans la littérature (Buttfield et al., 2006). Jusqu'à présent, toutes les études utilisant des corrélats neuronaux de performances de tâche pour l'adaptation de décodeurs de contrôle ont soit utilisé des Potentiels Liés aux Erreurs (ErrP) soit des signaux simulés (Artusi, 2012; Artusi et al., 2011; Blumberg et al., 2007; Buttfield et al., 2006; Gürel and Mehring, 2012; Llera et al., 2012, 2011; Spüler et al., 2012; Zeyl and Chau, 2014). A l'exception de l'étude de Gürel et Mehring (2012), toutes ces études sont limitées à l'adaptation de décodeur de contrôle pour la classification. Cependant, les ICMs motrices complexes peuvent contrôler plusieurs degrés de libertés continus simultanément (Benabid et al., 2019; Willett et al., 2021; Wodlinger et al., 2014). Nous supposons que cette limitation vient de la façon dont sont générés les ErrPs. Ceux-ci sont induits par des événements erronés discrets dans le temps. Dans cette thèse, on appellera ce type de corrélats neuronaux des corrélats de **performance de tâche motrice liés à un évènement (eMTP)**. Le contrôle d'un effecteur à plusieurs degrés de liberté continue peut effectivement générer des événements erronés discrets. Cependant, lors de ce type de contrôle, certaines erreurs ne sont pas des événements discrets. C'est par exemple le cas lorsqu'une trajectoire 2D dévie graduellement de la trajectoire idéale. Il n'y a alors pas un événement erroné discret mais une multitude de petits déplacements, qui deviennent graduellement de plus en plus erronés. Un corrélat neuronal qui serait émis à tout instant et pas seulement lors d'événements discrets pourrait être plus approprié pour effectuer l'adaptation d'ICMs motrices complexes. Cette hypothèse est corroborée par l'étude de Gürel et Mehring (2012), qui est la seule à utiliser un tel signal (bien que simulé) et qui est aussi la seule à faire l'adaptation d'un décodeur de contrôle avec plusieurs sorties continues. De nombreuses études ont étudié la détection de corrélats d'erreurs lors de tâches de contrôle continu (Kreiling et al., 2009; Lopes Dias et al., 2018; Lopes-Dias et al., 2019; Milekovic et al., 2013, 2012; Omedes et al., 2015; Spüler and Niethammer, 2015; Wilson et al., 2019). Cependant, une partie des tâches est effectuée lors de contrôle continu unidimensionnel. Or une erreur dans ce cas reviendrait à un changement de direction, ce qui peut être considéré comme un événement discret. Les autres études s'intéressent à des tâches de contrôle continu multi-dimensionnel. Cependant, dans chacune d'entre elles les erreurs étudiées sont introduites artificiellement de manière discrète. Actuellement, il n'existe pas d'étude faisant état de la détectabilité d'erreurs non discrètes en utilisant des signaux neuronaux.

En plus de devoir détecter un nouveau type de corrélats de **performance de tâche motrice continus dans le temps (cMTP)** cette étude fait face à une seconde difficulté. Comme dans la plupart des essais clinique d'ICMs motrices, les implants utilisés dans cette étude enregistrent l'activité du cortex sensorimoteur. Or la majeure partie des études de détection de corrélats neuronaux de performance de tâche utilisent des données cérébrales acquises en dehors du cortex sensorimoteur (tel que les

ErrPs). Un nombre limité d'étude porte sur la détection de corrélats neuronaux de performance de tâche dans le cortex sensorimoteur (Koelewijn et al., 2008; Milekovic et al., 2013, 2012; Schie et al., 2004; Völker et al., 2018; Wilson et al., 2019). Ces études rapportent que des corrélats neuronaux d'eMTP sont détectables après des événements erronés discrets dans le cortex sensorimoteur. Les corrélats neuronaux décrits sont principalement dans le domaine fréquentiel (contrairement aux ErrPs).

Dans cette thèse, le processus d'auto-adaptation développé ajoute un module auto-adaptatif à la boucle fermée classique des ICMs (Figure 8). Des corrélats neuronaux de cMTP sont détectés grâce à un décodeur entraîné spécifiquement pour cette tâche (décodeur de cMTP). Les sorties des décodeurs de contrôle et de cMTP sont ensuite combinées pour produire une estimation des labels du décodeur de contrôle. Ces labels estimés sont finalement utilisés pour entraîner ou mettre à jour le décodeur de contrôle en temps réel, ce qui reste possible durant l'utilisation libre de l'ICM par l'utilisateur.

#### Chapitre 4 : Design expérimental

Plusieurs paradigmes de contrôle de BCI sont utilisés dans cette thèse, avec des effecteurs de complexité croissante. Dans le premier paradigme, le patient contrôlait un avatar humain affiché sur un écran d'ordinateur. Cet avatar humain pouvait soit être immobile debout soit marcher tout droit à vitesse constante. Le patient utilisait de l'imagerie motrice des jambes pour faire avancer l'avatar, et pas d'imagerie motrice pour l'arrêter. Un panneau d'instruction était aussi affiché pour indiquer au patient quel état l'avatar devrait avoir. Trois jeux de données ont été enregistrés avec différentes configuration d'électrodes utilisées pour enregistrer les données (32 électrodes parmi 64 peuvent transmettre des données par implant). La première configuration est celle qui maximise la détection de l'imagerie motrice et est celle utilisée lors des enregistrements effectués avant le début de cette thèse. Le jeu de données correspondant est appelé Runner MI central. Ensuite deux jeux de données ont été enregistrés pour maximiser la surface d'enregistrement afin de détecter des corrélats neuronaux de performance de tâche. Le jeu de données avec le plus grand nombre d'expériences est nommé Runner MI, l'autre étant appelé Runner MI inverted. Dans le quatrième et dernier jeu de données enregistré avec ce paradigme, le patient n'avait aucun contrôle sur l'effecteur. Les mouvements de l'avatar étaient contrôlés par l'ordinateur, et des périodes d'erreurs étaient introduites automatiquement. Ce jeu de données est appelé Runner no\_MI.

Dans le deuxième paradigme le patient contrôlait un exosquelette virtuel affiché sur un écran d'ordinateur. L'exosquelette pouvait soit être immobile, soit être dans l'un des quatre états moteurs suivant : mouvement du bras droit, mouvement du bras gauche, rotation du poignet droit ou rotation du poignet gauche. Le patient effectuait de l'imagerie motrice directe afin d'activer les différents états moteurs. Des instructions étaient affichées, et le patient avait pour consigne d'activer les états moteur demandés sans considération pour les mouvements effectués à l'intérieur de chaque état. Le jeu de données correspondant est appelé Exo.

Dans le troisième paradigme, le patient contrôlait un curseur en forme de main affiché sur écran d'ordinateur. Le curseur pouvait se déplacer librement et en deux dimensions continues dans un espace carré. A tout instant, une cible était affichée et le patient devait faire de l'imagerie motrice pour déplacer le curseur. Lorsqu'une cible était atteinte, la position du curseur était réinitialisée au centre de l'espace et une nouvelle cible était affichée. Le jeu de données correspondant est appelé Cursor.

#### Chapitre 5 : Détection de corrélats neuronaux de performance de tâche

Dans ce chapitre nous étudions la détectabilité de corrélats neuronaux d'eMTP et de cMTP dans nos différents jeux de données. La détection d'eMTP est faite pour les jeux de données Runner MI, Runner



no\_MI et Runner MI inverted. La détection de cMTP est faite pour les jeux de données Runner MI, Runner no\_MI, Exo et Cursor.

Dans les deux cas, l'extraction de caractéristiques d'entrée pour le décodeur de MTP est basée sur la littérature la plus proche, c'est-à-dire l'extraction d'informations temps-fréquence. Pour chaque epoch d'une seconde, les informations temps fréquences sont extraites de 10 à 150Hz, tous les 10Hz. Dans le domaine temporel, l'information est moyennée en dix points par epoch. Les epochs successifs ont un recouvrement temporel de 90%. Les labels sont créés différemment pour la détection d'eMTPs et de cMTPs. Dans les deux cas, les labels sont binaires : si un epoch est labélisé, il peut être correct ou erroné. De plus, un équilibrage des quantités de données dans les deux classes est effectué afin de minimiser les risques de confusion de détection de corrélats d'imagerie motrice au lieu de corrélats de MTP par le décodeur de MTP : un sur-échantillonnage des epochs de chaque classe motrice est effectué à l'intérieur de chaque classe MTP (erroné ou correct). Plusieurs algorithmes d'apprentissage machine ont été testés pour l'entraînement du décodeur d'eMTP discrets : SVM, régression logistique, NPLS, MLP et CNN. Les performances de chaque décodeur sont évaluées en utilisant l'aire sous la courbe (AUC) ROC moyenne dans un validation croisée à 5 blocs.

Les résultats de détection de corrélats neuronaux d'eMTP montrent des différences non significatives entre les performances des différents décodeurs testés sur les jeux de données Runner MI et Runner no\_MI (Runner MI AUC moyenne : NPLS 0.601, régression logistique 0.605, SVM 0.623, MLP 0.626, CNN 0.580. Test de Friedman, p-valeur 0.08 ; Runner no\_MI AUC moyenne : NPLS 0.653, régression logistique 0.662, SVM 0.645, MLP 0.680, CNN 0.630. Test de Friedman, p-valeur 0.13). Pour le reste de la thèse, la NPLS a été utilisée pour entraîner les décodeurs de MTP. On note que la détection de corrélats de MTP est supérieure dans le jeu de données Runner no\_MI que dans le jeu de données Runner MI. Ceci était attendu car les signaux d'imagerie motrice générés pour contrôler l'avatar dans le jeu de données Runner MI peuvent être considérés comme du bruit pour la détection de corrélats de MTP.

Les résultats de détection de corrélats neuronaux de cMTP donnent une AUC de 0.623 pour le jeu de données Runner MI, 0.678 pour le jeu de données Runner no\_MI, 0.568 pour le jeu de données Exo et 0.657 pour le jeu de données Cursor. Ces résultats montrent que la détection de corrélats de cMTP dans des signaux neuronaux acquis depuis le cortex sensorimoteur est possible, ce qui est un point clé pour l'implémentation de l'ICM auto-adaptative élaboré dans cette thèse. Une étude des performances des décodeurs de cMTP en fonction du nombre de facteurs utilisés dans la NPLS montre qu'un nombre assez faible de facteurs permet d'atteindre un plateau de performance (entre 4 et 20 en fonction des jeux de données). Il peut aussi être noté que les paramètres des décodeurs de cMTP varient fortement entre les différents jeux de données. Cela ne permet cependant pas de conclure quant à la stabilité potentielle d'un décodeur de cMTP utilisé pour différentes tâches motrices et des études plus approfondies doivent être effectuées à ce sujet. La stabilité temporelle des décodeurs de cMTP n'est pas non plus démontrée, mais la façon dont la validation croisée a été effectuée (chaque bloc contient des sessions d'enregistrement distinctes, parfois enregistrées avec plusieurs jours d'écart) semble indiquer qu'une certaine stabilité temporelle existe.

### Chapitre 6 : ICM auto-adaptative

Dans ce chapitre, l'interface cerveau-machine auto-adaptative est implémentée et testée dans une simulation d'utilisation en ligne. Dans cette simulation, chaque jeu de données est séparé en trois. Une partie est utilisée pour entraîner le décodeur de cMTP, une partie est utilisée pour entraîner le décodeur de contrôle de manière auto-adaptative et une partie est utilisée pour tester les performances dudit décodeur de contrôle.

Afin de minimiser le bruit dans les labels estimés, seule une partie des epochs est utilisée. Dans les données d'entraînement, la sortie du décodeur de cMTP pour chaque classe (erroné ou correct) est modélisée par une gaussienne. Les paramètres de ces gaussiennes (moyenne et déviation standard) sont utilisés pour définir deux seuils qui permettent de n'utiliser que les epochs pour lesquels la sortie du décodeur de cMTP est estimée avec assez de confiance. Pour ces epochs, un label (pour l'entraînement du décodeur de contrôle) est ensuite estimé à l'aide d'une fonction de *re-labélisation*. Lorsqu'un epoch est détecté correct, la sortie des fonctions de re-labélisation correspond à la prédiction effectuée par le décodeur de contrôle. Lorsqu'un epoch est détecté erroné, la sortie des fonctions de re-labélisation est soit la deuxième classe la plus probable estimée par le décodeur de contrôle (paradigmes de contrôle binaire et multi-classes, i.e. jeux de données Runner MI et Exo), soit les erreurs ne sont pas utilisées pour mettre à jour le décodeur de contrôle (paradigme de contrôle continu bidirectionnel, i.e. jeu de données Cursor). Dans cette simulation, le décodeur de contrôle est entraîné en utilisant la REW-NPLS comme les autres décodeurs de contrôle entraînés dans cet essai clinique. En plus du décodeur de contrôle entraîné de manière auto-adaptative, deux autres décodeurs de contrôle sont entraînés pour chaque jeu de données à des fins de comparaison. Le premier est entraîné de façon supervisée, en utilisant les vrais labels. Le second est entraîné de façon auto-adaptative, mais en mélangeant la sortie du décodeur de cMTP de façon aléatoire afin de représenter le niveau de chance. Les performances des décodeurs de contrôle sont évaluées différemment pour chaque paradigme. Pour le paradigme de contrôle binaire (jeux de données Runner MI), les performances sont évaluées en utilisant l'aire sous la courbe ROC. Pour le paradigme de contrôle multi-classes (jeu de données Exo), les performances sont évaluées en utilisant une généralisation multi-classe de l'aire sous la courbe ROC (Hand and Till, 2001). Pour le paradigme de contrôle bidimensionnel continu (jeu de données Cursor), les performances sont évaluées à l'aide de la similarité cosinus.

Les résultats principaux de cette étude concernent la précision des labels estimés à l'aide du processus auto-adaptatif ainsi que les performances des décodeurs de contrôle entraînés de façon auto-adaptative. La précision des labels estimés était de 64.9% pour le paradigme de contrôle binaire (jeu de données Runner MI) et 64.5% pour le paradigme de contrôle multi-classes (jeu de données Exo). Pour le paradigme de contrôle bidimensionnel continu (jeu de données Cursor), 63% des labels estimés étaient à moins de 60° d'écart des vrais labels, 11.4% entre 60° et 90° d'écart et 25.3% à plus de 90° d'écart. Dans le paradigme de contrôle binaire, l'AUC du décodeur de contrôle entraîné de façon auto-adaptative était de 0.636 comparé à 0.896 lorsqu'il était entraîné de façon supervisée et à un niveau de chance de 0.501. Dans le paradigme de contrôle multi-classes, l'AUC du décodeur de contrôle entraîné de façon auto-adaptative était de 0.760 comparé à 0.818 lorsqu'il était entraîné de façon supervisée et à un niveau de chance de 0.516. Dans le paradigme de contrôle bidimensionnel continu, la similarité cosinus du décodeur de contrôle entraîné de façon auto-adaptative était de 0.159 comparé à 0.211 lorsqu'il était entraîné de façon supervisée et à un niveau de chance de -0.023. Pour les trois jeux de données, il y avait un effet significatif de la méthode d'entraînement du décodeur de contrôle sur les performances (tests de Friedman, p-valeur paradigme binaire MI 0.0009, multi-classes 0.0009, bidimensionnel continu 0.002). **Ces résultats valident la preuve de concept d'ICM motrice complexe auto-adaptative.** Les performances des décodeurs de contrôle entraînés de manière auto-adaptative sont inférieures à celles des décodeurs entraînés de manière supervisée. Cela était attendu étant donné que les labels estimés sont moins précis que les labels obtenus de manière supervisée, et que la taille des jeux de données d'entraînement est plus petite pour l'entraînement auto-adaptatif que pour l'entraînement supervisé étant donné que tous les epochs ne sont pas labélisés. Cependant, dans le long terme cette différence de taille de jeu de données d'entraînement est vouée à s'inverser étant donné que les ICMs auto-adaptative peuvent labéliser des données durant l'utilisation libre de l'ICM par son utilisateur. De plus, les décodeurs entraînés de façon supervisée peuvent voir leur

performance diminuer avec le temps en raison de dérives de concept par exemple. On peut donc s'attendre à ce que lors d'utilisation long terme l'écart entre entraînement auto-adaptatif et supervisé soit réduit ou même inversé.

### Chapitre 7 : Limitations et perspectives

La principale limitation de cette étude est le fait qu'elle soit limitée à un seul sujet. Cependant, un total de cinq patients est prévu pour cet essai clinique, qui pourraient être ajoutés à cette étude. D'autres essais cliniques, comme celui récemment lancé conjointement par l'EPFL et Clinatex (STIMO-BSI), pourraient aussi être utilisés pour augmenter le nombre de participants à cette étude. Quoiqu'il en soit, il est critique de répliquer les résultats décrits dans cette thèse avec d'autres sujets afin de confirmer l'usabilité de l'ICM auto-adaptative proposée.

Une seconde limitation de cette étude est qu'elle n'a pas été effectuée en ligne, mais dans une simulation d'utilisation en ligne. Cependant, cette simulation a été créée de façon à être aussi proche que possible d'une utilisation en ligne, et peut être utilisée telle quelle pour effectuer des expériences en ligne. Ces expériences ont d'ailleurs été initialisées durant cette thèse (mais fortement retardées en raison de la situation sanitaire mondiale), et pourraient être poursuivies dans le futur.

La stabilité du décodeur de cMTP dans le temps et vis-à-vis des variations de tâche est désirable pour les ICMs auto-adaptatives. La stabilité temporelle signifie que le décodeur de cMTP a besoin d'être mis à jour moins fréquemment, et la stabilité aux variations de tâche permet d'utiliser un unique décodeur pour entraîner n'importe quel décodeur de contrôle. Étant donné que les décodeurs d'ErrPs (un autre type de corrélats de performance de tâche) sont relativement stables dans le temps et aux variations de tâche, il est possible que ce soit aussi le cas des corrélats de cMTP détectés dans cette thèse. Bien que ces deux thèmes aient été abordés dans cette thèse, des expériences supplémentaires devraient être effectuées pour permettre une conclusion appropriée.

Les ICMs utilisées dans cette thèse restent relativement simples comparées à l'état de l'art des ICMs motrices complexes. Bien que théoriquement utilisable pour des décodeurs avec n'importe quel nombre de sorties continues, cela reste à confirmer en pratique. De plus, ce paradigme d'ICM auto-adaptatif devra probablement être adapté pour fonctionner avec certaines ICMs plus complexes, telle qu'une ICM avec plusieurs états moteurs, chacun avec plusieurs degrés de liberté continus.

Le paradigme d'ICM auto-adaptative utilisé peut encore être amélioré de nombreuses manières. Le pré-traitement et l'extraction de caractéristiques utilisés ici pour la détection de corrélats de cMTP n'est qu'une première ébauche et son optimisation pourrait améliorer les performances. De même, de nombreux hyper-paramètres utilisés pourraient être optimisés, bien que cela requerrait une quantité plus importante de données. Lors de la labélisation des epochs, ceux-ci sont actuellement inclus ou écartés grâce à deux seuils. Il serait possible d'utiliser des poids pour tous les epochs plutôt que des seuils durs. Les fonctions de re-labélisation pourraient aussi être raffinées, par exemple en ne se limitant pas à un seul epoch mais en utilisant les sorties des décodeurs de contrôle et de cMTP sur plusieurs epoch pour prédire chaque label.

Finalement, la preuve de concept réalisée dans cette thèse est prometteuse pour le futur des ICMs auto-adaptative, particulièrement lorsqu'elles contrôlent plusieurs degrés de liberté continus. Cependant, de nombreuses limitations et améliorations potentielles ont été décrites dans ce chapitre, mettant en avant la nécessité de raffiner les ICMs auto-adaptatives avant de pouvoir les utiliser en situations réelles.

# Chapter 1: Introduction to Brain-Computer Interfaces (BCIs)

Brain-computer interfaces (BCIs) are systems that convert signals recorded from the brain of a user into commands for effectors. Various effectors are possible, allowing BCIs to be used for diverse purposes. Examples of applications include neurorehabilitation, motor deficit compensation, mental state monitoring or computer gaming (Lebedev and Nicolelis, 2017). Nevertheless, up to now BCIs were mainly designed for individuals with motor disabilities as a solution to restore missing or damaged neural pathways between their brain and a bodily effector. In such cases, effector can be communication devices, robotic devices or computers. Motor dysfunctions can be caused by various conditions, such as brain strokes, amyotrophic lateral sclerosis or cerebral palsy or spinal cord injuries. As a technology, BCIs are in a development stage. However, the combined ongoing progress in artificial intelligence, improvements in brain-implantable devices and increasing computational power available are promising for the growing field of BCIs.

## I. Neural signals for BCI systems

BCIs use recordings of brain activity in order to estimate their user's intentions. The brain signals recorded are mostly generated by the activity of neurons. In a simplistic explanation, neurons are made of three main parts: dendrites, which are links to other neurons in charge of receiving information; an axon, which is used to send information to other neurons; and a main body, which integrates incoming information and decides whether it should send some information through its axon (Figure 1) (Reece, 2011). Neurons communicate with each other by generating action potentials in the main body and sending it through its axon. This action potential is propagated through the axon to the cellular junctions with other neurons, called synapses. Upon reaching a synapse, an action potential triggers the release of neurotransmitters in the synapse, where they can be taken in by the dendrites of the adjacent neuron. Action potentials are brief all-or-nothing localized changes in the axon's transmembrane potential, which are propagated in one way from the neuron's main body to the axon's terminations. This neuronal activity, or its consequences, are the brain signals used as input to BCI systems.

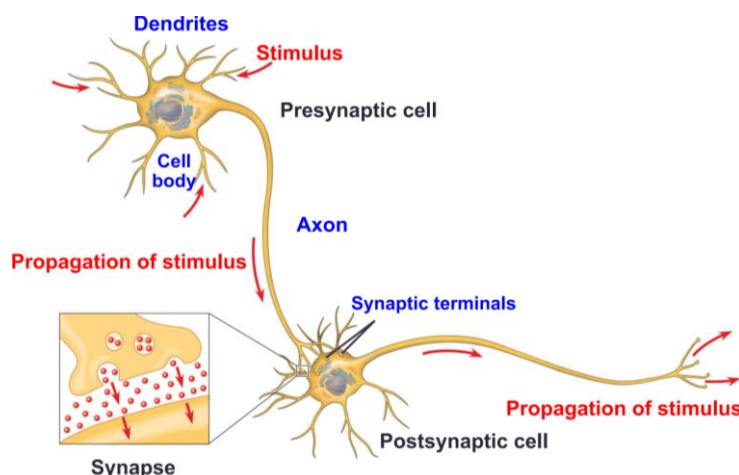


Figure 1 Schematic of inter-neuronal communication. Figure extracted from the Campbell Biology (Reece, 2011).

Due to the complexity of neural activity, BCIs may only detect a few preselected mental commands. These mental commands are carefully chosen in order to elicit neural signals as easy to distinguish from noise as possible. In order to be practical for BCI use, the neural signals generated by mental commands should be detectable reliably at the single-trial level. Due to these constraints, some neural signals are commonly used in the BCI field.

**Sensorimotor rhythms** are variations in the amplitude of neural oscillations in specific frequency bands that are generated by executed, attempted or imagined movements. These sensorimotor rhythms modulations can mainly be recorded from the sensorimotor cortex, mainly in the mu (8Hz – 13Hz), beta (14Hz – 30 to 35Hz) (Nicolas-Alonso and Gomez-Gil, 2012) and gamma (>60Hz) frequency bands (Degenhart et al., 2018). Several characteristics of sensorimotor rhythms are appealing for BCIs. They can be generated spontaneously by the user, which means BCI usage is not tied to an external stimulus. When used for restoration of movements of disabled individuals, the mental command is closer to the one used by healthy individuals than for other mental tasks. Additionally, it was shown that there is somatotopic mapping in the sensorimotor cortex during executed, attempted or imagined movement (Stippich et al., 2002). This means that executed, attempted or imagined movements of different body parts will lead to different spatial patterns in the sensorimotor cortex and can therefore be used as different mental commands for a BCI.

The **P300** is a positive peak of the electrical potential that is elicited approximately 300ms after an infrequent stimulus is detected (Nicolas-Alonso and Gomez-Gil, 2012). Contrary to sensorimotor rhythms, P300 cannot be elicited voluntarily by the user, which makes BCI usage timings dependent on the interface. However, P300 can be used as a control signal without any user training. P300 is mainly used as a control signal in BCIs that restore patient communication, which are often called P300-spellers (Daucé et al., 2015; Donchin et al., 2000; Mattout et al., 2015). In P300 spellers, characters (or lines/rows) of characters flashes successively. Since characters are written one by one, during each flashing cycle only one character will correspond to the one the user wishes to write. Flashing of this target character is less frequent than flashing of any non-target character, and therefore elicits a P300 signal. The target character can then be determined since P300 appearance is time-locked to approximately 300ms after the infrequent stimulus.

**Steady-state evoked potentials** are frequency modulations in the brain signals that are elicited by a stimulus presented at a constant frequency (Rezeika et al., 2018). There is an increase of the amplitude of the neural signal in the same frequency band as the stimulus, with stronger increase for attended stimulus than non-attended stimulus. Although steady-state evoked potentials are generally elicited using visual stimulus (Zhu et al., 2010), it was shown that they can also be generated with other modalities, such as audio (Kim et al., 2011) or tactile (Ahn et al., 2015) stimuli. Similarly to P300, steady-state-evoked potentials can be detected without user training, but is not generated autonomously by the user. BCIs that use steady-state evoked potentials inherently depend on an external system that generates the stimuli. Furthermore, these BCIs are gaze-dependent when the stimulus used is visual, which can be problematic for some patients or can be tiring in the long run (Treder et al., 2011). Finally, the risks of epileptic seizure due to the rapid visual oscillations have not been quantified yet (Zhu et al., 2010).

Other neural signals are also used, although less frequently, to control BCIs. Among them, one could mention **neural correlates of covert attention** (Andersson et al., 2013) or **error-related potentials** (Chavarriaga et al., 2016).

None of these neural signals is strictly superior to the other for BCI operation. As mentioned, each of them has its respective advantages and drawbacks. For instance, P300 is well-suited for spellers, as

one control signal is enough to operate them. It is however, not ideal for the control of complex motor BCIs, as a binary control signal can hardly be used effectively to control multiple degrees of freedom continuously. Such BCIs are mainly controlled using motor imagery, since each imagined movement can be used as a different control signal as long as they can be distinguished from one another.

## II. Components of BCI systems

Brain computer interfaces are composed of three main components: an acquisition system, a transducer and one or more effectors (Figure 2). The acquisition system samples, amplify and digitize the brain activity of the user (Wolpaw and Wolpaw, 2012). The transducer translates the digitized brain signals into an estimation of the intention of the user. Finally, the intention estimation is converted into command for the BCI effector. The user receives feedback from the effector's actions, most often through visual feedback.

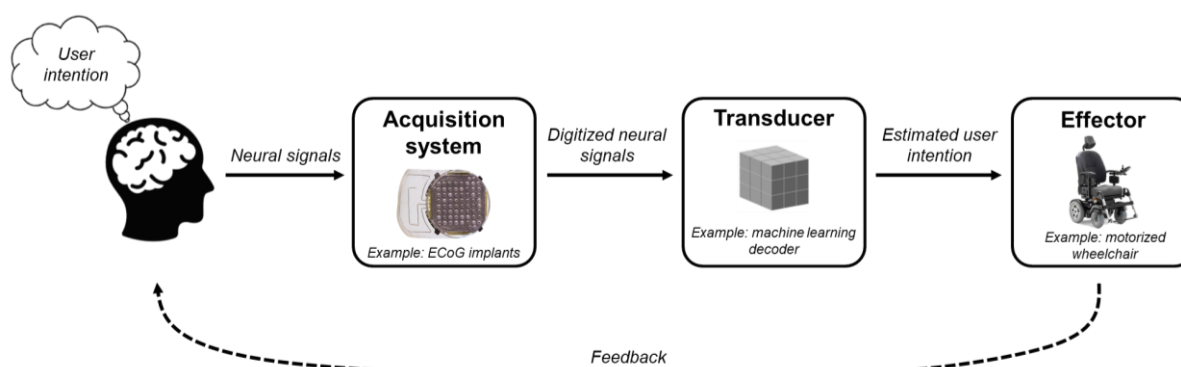


Figure 2 Architecture of a brain-computer interface. The main components are the acquisition system, that digitize brain signals; the effector, in charge of performing actions; and the transducer, that translates the digitized brain signals into commands for the effector.

### II.1. Acquisition systems

As mentioned by Wolpaw and Wolpaw (2012), some characteristics are especially desirable for acquisition systems used in brain-computer interfaces:

- Good spatial resolution
- The ability to record from a large area
- Temporal stability of recordings
- Low recording latency
- High temporal resolution
- Good frequential resolution
- Portability
- Affordability
- Minimal risk.

Numerous acquisition systems exist in order to record brain activity. They can directly record electrophysiological signals generated by neuronal activity, like electroencephalography (EEG), electrocorticography (ECoG), micro-electrode arrays (MEA) and magnetoencephalography (MEG) (Stieglitz et al., 2009) or rely on indirect recording of neuronal activity, like functional near-infrared spectroscopy (fNIRS) or functional magnetic resonance imaging (fMRI) (Cohen et al., 2014; Gunasekera et al., 2015; Hämäläinen et al., 1993; Naseer and Hong, 2015; Stieglitz et al., 2009). Each of these acquisition systems has associated advantages and disadvantages.

**Micro-electrode arrays (MEAs)** are intracortical acquisition systems that are implanted directly into the cortex. They can allow recording of single-unit activity, multi-unit activity or local field potentials (Wolpaw and Wolpaw, 2012). Single or multi-unit activity reflects the spiking activity of one or a few neurons located near the electrodes' tips. Local field potentials aggregate the activity of a larger population of neurons (tens of thousands) (Lebedev and Nicolelis, 2017). Due to the proximity of the electrodes to the neurons, MEA is the only acquisition system that is able to record spiking activity of neurons. Other acquisition systems record the activity of large neural populations. Individual neuronal patterns are lost due to the spatial averaging (Wolpaw and Wolpaw, 2012). MEAs have very high spatial and temporal resolution, but spatial information is limited to the close vicinity of the electrodes (Figure 3). Thanks to the high information content in MEA signals, it can be used for the control of complex BCIs with a large number of degrees of freedom. Disabled subjects were able to control a robotic arm (Hochberg et al., 2012), or an upper limb prosthesis with up to ten degrees of freedom (Collinger et al., 2013; Wodlinger et al., 2014). However, there is high variation in MEA signals in between and during days (Perge et al., 2013), and the decoders used in these studies had to be retrained frequently. Furthermore, MEAs have an additional drawback in the immune reaction induced by the intracortical implantation of electrodes. The immune reaction leads to the formation of glial scar tissue around the electrodes. Glial encapsulation of the electrodes may significantly degrade the quality of the recorded signal over time (Gunasekera et al., 2015). Another drawback of MEAs is its current limitation to wired acquisition systems due to the high data transfer rates. Wired systems are associated to increased medical risks, such as increased possibility of infection (Lee et al., 2000). Finally, MEAs have good spatial, temporal and frequential resolution, have low recording latency, and most of all can detect single neuron activity. However, MEAs carry risks and are not portable due to the wires, have very limited recording area on the cortex and have low temporal stability of recordings. Therefore, MEAs provide the most informative signals but its drawbacks currently make it unsuited for chronic BCI usage.

**Electrocorticographic arrays (ECoG)** are also invasive acquisition systems but they do not affect the brain's integrity. They are usually placed below the skull and either above or under the dura matter. Due to electrodes positioning and sizes, ECoG recordings are limited to population recordings of the superficial neurons of the cortex. The reduced invasiveness leads to more benign surgery procedures and to long-term stability of the neural signals acquired (Larzabal et al., 2021). To date, the ECoG grids used in most of the ECoG BCI studies were fitted for other purposes than BCI, such as localization of epileptic foci. Localization of electrodes arrays, as well as electrode spacing and number in the arrays was not optimized for BCI, and this should be taken into account when reviewing the performance of ECoG-based BCI studies. ECoG arrays provide good temporal and spatial resolution, although lower than MEAs (Figure 3). Subjects implanted with ECoG grids were able to control up to three degrees of freedom of a computer cursor (Degenhart et al., 2018), or more recently long-term control of up to eight degrees of freedom of an exoskeleton (Benabid et al., 2019). ECoG arrays are safer than MEAs but they remain an invasive procedure. Contrary to MEAs, some wireless ECoG implants were successfully developed and used in BCI human clinical trials in the past years (Benabid et al., 2019; Vansteensel et al., 2016). Finally, ECoG has good temporal and frequential resolution, low recording latency, can be portable (wireless versions) and recordings are stable in time. However, ECoG has medium spatial resolution, can record medium sized cortical areas, and is not completely risk free as it involves a craniotomy. For disabled users ECoG strikes a good balance between safety and control possibilities.

**Electroencephalography (EEG)** is a non-invasive acquisition system that is highly portable with a relatively low cost. Electrodes are placed on the surface of the scalp, allowing high spatial coverage. Like ECoG, EEG cannot record spiking activity of neurons. Additionally, it is limited to low-passed neuronal activity. This is due to the amplitude of electrophysiological signals depending on the inverse

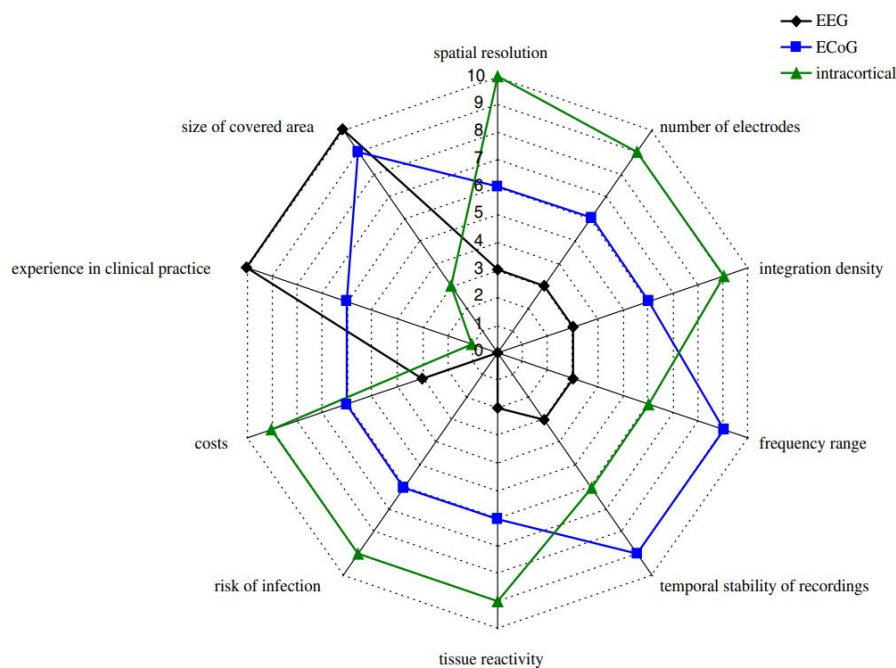


Figure 3 Comparison of the characteristics of EEG, ECoG and MEA acquisition systems. Figure extracted from Stieglitz et al. (2009). Scoring: 0 = low; 10 = high.

of the distance between the neural source and the electrodes (Buzsáki et al., 2012), and to the neural sources producing current amplitude inversely proportional to their frequency (Leuthardt et al., 2006). EEG has relatively good temporal resolution, but suffers from poor spatial resolution, which restricts the possibilities of separating activation of nearby cortical areas (Figure 3). Finally, EEG is strongly sensitive to artifacts, which are easily generated by ocular movements, muscular activity or movements of the electrodes (Nicolas-Alonso and Gomez-Gil, 2012). Nevertheless, EEG has been successfully used in a large number of BCI studies (Lebedev and Nicolelis, 2017; Lotte et al., 2018) due to its practicality, wide-spread availability and relatively low cost. Finally, EEG has good temporal resolution, can record from a wide cortical area, has a low recording latency, is highly portable and has neglectable risks. However, EEG has poor spatial and frequential resolution and medium temporal stability of recordings.

**Magnetoencephalography (MEG)** measures the magnetic field induced by neural activity. MEG has better spatial and frequential resolution than EEG and is not invasive either (Buzsáki et al., 2012; Hämäläinen et al., 1993). However, MEG recording systems are expensive, very bulky and require the subject to be immobile, which is impractical for most BCI applications.

**Functional near-infrared spectroscopy (fNIRS)** is based on the indirect recording of neural activity. It records the brain's blood oxygenation level, which is increased locally with neural activity (Wolpaw and Wolpaw, 2012). fNIRS detects cortical activity with better spatial resolution than EEG, but lower temporal resolution (~100ms). To the temporal resolution of the acquisition system must be added a physiological delay of (several seconds), which corresponds to the delay between neural activity and blood oxygenation changes. BCI experiments were successfully performed using fNIRS (Coyle et al., 2007; Khan et al., 2018; Naseer and Hong, 2015). fNIRS is a relatively recent acquisition system in the field of BCI (Coyle et al., 2004), but it is gaining in popularity because it is relatively inexpensive, portable and can even be combined with EEG in bi-modal acquisition (Sirpal et al., 2019), which could improve BCI performances.



**Functional magnetic resonance imaging (fMRI)** is also based on indirect recording of neural activity. Similarly to fNIRS, fMRI detects changes in the blood oxygenation levels (Wolpaw and Wolpaw, 2012). In addition to the hemodynamic delay, fMRI suffers from a low temporal resolution (one to two seconds). The main draw of fMRI is its high spatial resolution - usually around a few millimeters depending on the voxel size used - of the whole brain (Weiskopf, 2012). However, the temporal delay and low resolution makes fMRI more suited to BCIs for neurorehabilitation than for functional compensation (Lebedev and Nicolelis, 2017). Furthermore, fMRI recording systems are expensive and very bulky, which makes them even more impractical for many BCI applications.

Similarly to neural signals, each acquisition technique has its own advantages and drawbacks. Depending on the BCI application, some acquisition system can be preferred. Non-disabled users have no incentive to use invasive BCIs, even if the associated risks have been decreasing. Conversely, long-term users (such as locked-in patients) benefit from the increased signal quality and reduced setup time of invasive acquisition systems. Complex motor BCIs, which require near real-time control (Shanechi et al., 2017) and multiple control signals, are better achieved with ECoG or MEAs than fMRI or fNIRS. Proper acquisition system selection is crucial in order to optimize the potential of a BCI.

## II.2. Effectors

A wide variety of effectors are possible for brain computer interfaces. However, most BCIs are targeted toward disabled individuals which is reflected in the effector choices. Effectors in the BCI field are often designed to be used for functional compensation of communication or movement. For communication, effectors are usually spellers on computer screens, which can be a simple grid of characters or use more complex designs (Treder and Blankertz, 2010). For movement, a wide range of effectors can be used such as robotic arms, electric wheelchairs, prostheses or exoskeletons (Benabid et al., 2019; Hochberg et al., 2012; Tanaka et al., 2005; Wodlinger et al., 2014). Effectors that produce movements must however remain safe and harmless at all time, especially since disabled user may not react to protect themselves. Therefore, many studies replace real effectors by virtual equivalents on computer screens, providing safer and cheaper alternatives for BCI research (Cunningham et al., 2011; Leuthardt et al., 2006; Schalk et al., 2008).

An important secondary role of effectors is to provide feedback to the user. Feedback has been shown to have a strong impact on BCI performances (Cunningham et al., 2011; Jarosiewicz et al., 2013; Shanechi et al., 2017). It is one of the mechanisms responsible for user training (McFarland and Wolpaw, 2018; Perdakis and Millan, 2020), creating a double learning paradigm in BCIs (although decoders are trained only when labels are known, whereas user training can occur at any time). In most studies, feedback is only visual. Indeed, visual feedback is easy to create with both real and virtual effectors, and it can be used with tetraplegic patients. However, completely locked-in individuals may not have control over their eyes, which may render visual feedback unusable. Some studies investigated the use of other sensitive modalities (alone or combined with vision) to provide feedback, such as tactile stimulation (Tidoni et al., 2014) or audio cues (Tidoni et al., 2014).

## II.3. Transducers

The transducer in a BCI is in charge of transforming the digitized neural activity into a command for the effector. This is usually done in a serial process of several steps, including signal pre-processing, feature extraction, decoding and post-processing. The transducer can be considered to be at the core of the BCI system, due to its effective role as a bridge between neural signals and actions. The transducer directly impacts the decoding accuracy of the overall system, through its ability to reliably extract information from neural signals and interpret it. The design of each part of the transducer is highly dependent on the rest of the BCI system: the acquisition system and effectors used, the desired

application or the neural commands used to control the BCI. Generally, the design of the transducer is kept constant between users in a study, with parameters fitted to each user.

**Pre-processing** improves the signal quality provided by the acquisition system before performing any other action (Bashashati et al., 2007). Pre-processing can include a wide range of operations, such as artifact removal, re-sampling operations, spatial filtering, anti-aliasing filtering or other temporal filters more specific to the BCI system considered.

**Feature extraction** transforms the pre-processed neural signals into information that is thought to be optimal for the decoding of the user's intention. Compared to pre-processing, where the goal is to maximize the signal-to-noise ratio of brain signals in general, feature extraction aims at extracting from these neural signals the most relevant part for intention decoding. The feature extraction step is highly dependent on the neural command used to control the BCI. For instance, when using neural population recordings, temporal, frequential or a combination of both is often used (Wolpaw and Wolpaw, 2012). The decoding of imagined movements is mainly done with information located in specific bands in the frequential domain (Müller-Putz et al., 2016), whereas the decoding of errors is most often done in the temporal domain (Chavarriaga et al., 2014). When using neural data from MEAs, feature extraction can be done for the detection of individual neurons spike counts over short time bins.

The neural data available for the training of control decoder is often limited, while the dimension of the input space can be high. Thus, an important additional component of feature extraction is dimensionality reduction of the input space. Several dimensionality reduction methods exist, mainly divided between methods that project the feature space to a subspace of lower dimension (such as Principal Component Analysis or Independent Component Analysis) or methods that simply select a number of features to create a subspace (such as wrapper methods) (Schaeffer, 2018).

**Decoding** in BCIs is generally done with a decoder that is fitted to the problem at hand using machine learning algorithms. The decoder parameters are generally patient-specific and data-driven. Many decoder types exist, but a strong distinction in complexity can be made between decoders that have to predict discrete states of the user's intention (such as "move" vs "don't move" commands) and decoders that have to predict continuous variables (such as the direction of movements of a classic cursor on a computer screen). Regardless of their type, decoders have to be trained e.g., fitted to the problem in order to provide accurate estimation of the user's intention using the neural features previously extracted. Decoder training (also called "calibration" or "learning") is commonly done using supervised machine learning algorithms. Supervised training algorithms require a training data set which includes both the input data (neural data) and the output data (the user's intention, also called labels). The decoder parameters are then fitted by minimizing a loss function, which formalize the cost of erroneous predictions. The training data is often collected during specific sessions, which we call "training session" from here on. During these training sessions, the user is instructed to perform specific mental tasks. The instructions given to the user during these training sessions can be considered as the user's intention for the sake of labelling neural data. The labeled data collected during training sessions is then used to train the decoder before the user can control the BCI freely in subsequent sessions. Decoder training should be done in conditions as close as possible to free use of the BCI in order to maximize the performance of the system (Shenoy et al., 2006). In the literature, decoders are also referred to as "models". For consistency, we use "decoder" throughout this thesis, although the two terms are interchangeable.

Compared to other fields that use machine learning, the BCI field has to deal with a large number of constraints at the same time. The data in BCI has high dimension, training datasets are relatively small, the signal-to-noise ratio in the input data is low, labels are imperfect, and the problem that is fitted is

constantly changing due to user training effects, neural plasticity and other sources of non-stability of the signals.

**Post-processing** is sometimes applied to the output of the decoder in order to improve the quality of the predictions, such as smoothing predictions or preventing rapid back and forth switches between states, or to take into account a priori knowledge that is not available to the decoder, such as effectors limitations (i.e., a robotic arm may not be able to perform some movements or reach some destinations) (Schaeffer and Aksenova, 2018).

### III. BCIs for functional compensation

Ideally, BCIs aimed at disabled individuals should ideally have the following characteristics (Wolpaw and Wolpaw, 2012):

- Safe, BCIs should not pose any risk to their user.
- Activable and usable at will by the user at all times.
- Do not require the assistance of a caregiver, technician or scientist.
- Robust to out-of-the-lab environments.
- Provide functional compensation equivalent to the capacities of healthy individuals. Motor BCIs should restore real-time, multi-dimensional, dexterous control of one's own limbs or provide control of prosthetic limbs. Communication BCIs should provide communication rates as high as vocal communication.
- Require no more concentration than the same functions do for an able-bodied person.
- Chronically reliable, lasting several years between battery changes and hardware/software upgrades.
- Affordable & and easily transportable.
- Esthetically acceptable or invisible.

Most of these properties remain unattained or partially attained by today's state-of-the-art BCIs. Progress was made in the past years regarding the level of functional compensation offered by state-of-the-art communication and motor BCIs. A state-of-the-art communication BCI achieved a typing rate of 90 characters per minute with 94.1% accuracy (Willett et al., 2021). The MEA-based BCI allowed a tetraplegic to use attempted handwriting in order to write characters. The writing rate is impressively close to the average typing rate of healthy individuals of the same age on mobile devices (115 characters per minute). Regarding motor BCIs, an invasive MEA-based BCI (Wodlinger et al., 2014) enabled 10 degrees of freedom control of a robotic arm (3D translation, 3D orientation and four hand shapes) by a tetraplegic. Although current state-of-the-art motor and communication BCIs cannot yet compensate deficits to the level of healthy individuals, these results are promising and showcase the potential of BCIs for disabled users. However, long-term usability of MEA BCIs is still up to debate. More generally, wired connections are detrimental to the safety and social acceptability of the BCI. ECoG implants, which can be fully implanted and transmit data wirelessly, are safer, designed-for long term use and more esthetic, at the cost of some performance decrease. Similarly, non-invasive acquisition systems have even lower performances, and may suffer from repetitive set up time or non-transportability. As mentioned previously, control signals have different properties and not all control signals are optimal for all BCIs. BCIs for motor functional compensation are more likely to use internally generated signals in order to provide self-paced activation and control than BCIs for communication. Transducer design is dependent on the type of neural signals used and the type of control desired over the effector.

The selection of an acquisition system, control signal, transducer and effector are a series of trade-offs. The balance between risks, performance and usability should only be made with the target user and desired application in mind.



# Chapter 2: Scientific context and Objectives

## I. Scientific context: the Clnatec BCI project

### I.1. Clinical trial goals

As discussed in Chapter 1, the design of a BCI system is based on the target population and the desired application. The Clnatec BCI clinical trial “BCI and tetraplegia” was initiated with the goal of providing a proof of concept of long-term functional motor compensation for tetraplegic users. In order to perform efficient functional restoration of mobility, several properties are either desirable or required from the BCI system components:

- A suitable recording system. The recording system used must have a high enough temporal resolution to detect change in control commands at a high pace (Shanechi et al., 2017). It should have enough spatial coverage over the motor cortex to enable motor imagery detection from different limbs, and enough spatial resolution to enable detection of fine movements from each limb. The recording system used must also be suited for chronic use, since the end goal is to provide functional restoration of mobility to tetraplegic users. For this clinical trial, it was estimated necessary to use a recording system that would still be usable by the user for several years, without significant loss of signal quality. Finally, the recording system must be safe for the patient, even in the context of chronic use.
- Effectors designed for functional compensation of motor deficits. Motor functional compensation requires effectors capable of performing the motor actions desired by the user for him. For compensation of precise and advanced motor functions (such as upper limb motricity), effectors with a high number of degrees of freedom are needed. It also requires effectors that are practical for the gradual training of both the user and the BCI system. These effectors must be safe to use. For tetraplegic subjects, the inability of the user to feel potentially damaging actions performed by the effector must be taken into account.
- A decoder suitable for control of motor effectors with multiple degrees of freedom. As this thesis focuses on the decoding aspect of BCI, we expand on several specific points regarding the desired properties of the decoder used to control the BCI:
  - o The control decoder must be usable in real-time. Precise motor control requires low latency between the instruction sent by the user and the action being performed by the effector. This restricts the pre-processing and feature extraction to methods with low computational cost.
  - o The decoder must work in an asynchronous manner. Since the goal of the BCI is the functional restoration of movement, the BCI should be usable at any time. Synchronous BCI systems, which can only be used during specific time periods or conditions, are not well-suited to real-life use of BCIs for motor deficit compensation.
  - o The control decoder must be able to control effectors with a large number of degrees of freedom. For the functional compensation of basic lower limbs mobility, such as activating a walking switch, this may not be needed. However, functional compensation of upper limb mobility is more complex because upper limb actions are less prone to binary tasks. In order to control the position of the hands of the exoskeleton, the control decoder must accurately

decode six continuous degrees of freedom. More complex tasks may require grasping or rotation of specific joints, which could require even more complex decoding.

- The control decoder performance should be stable in time. It can be frustrating for the user if his control over the BCI changes noticeably from session to session or over time. If performance decrease too much, the control decoder will have to be updated or even retrained from scratch. Additionally, the complexity of the output of the control decoder is correlated to the amount of data required to train the model and therefore to the time required to acquire this training data. The stability in time of the performance of the decoder is thus even more important for complex control decoders, as these decoders take more time to train.
- Adaptive online training of the decoder is desirable. During supervised open-loop training the user does not have control over the BCI. However, performance drop-offs between training and online use have been reported with open-loop training. In addition, adaptive online training has several other advantages. It can be more engaging for users as they can directly see the effect of their mental commands on the effectors, and may visualize the progressive effect of both decoder and user-training. Adaptive online training can also help optimize the amount of training data required. In open-loop training, the decoder is usually trained offline after the training session. If its performances are not good enough, successive open-loop training sessions can be made. In adaptive online training however, it would be possible to check some aspects of the control decoder while it's being trained (such as convergence of the parameters or control performances) and take advantage of it, e.g. to know if enough data has been acquired.

### 1.2. Recording system of the Clinatec clinical trial: the WIMAGINE ECoG implant

The requirements for the recording system that were described above cannot be simultaneously all optimized. For all the existing recording techniques for brain signals, there exists a trade-off between several desired properties such as safety, signal quality or portability. Currently, stability of recording over several years is best achieved using ECoG recordings techniques. Non-invasive techniques are either not robust enough, have too much latency, are not portable or are too susceptible to noise and artifacts. Invasive techniques are less-safe and can also have some issues with stabilities due to biological reaction to the implant. Indeed, reactive tissue formation next to the implant can slowly decrease the quality of the signal acquired over time. Epidural ECoG recording strikes a good balance between safety, signal quality and chronic usability because it does not suffer from the issues of non-invasive techniques, it's relatively safe as it does not go under the dura matter, and it produces less reactive tissue formation than more invasive techniques (Chapter 1).

The WIMAGINE implant, for Wireless implantable Multi-channel Acquisition system for Generic Interface with Neurons (Mestais et al., 2015), was developed at CLINATEC for the purpose of safe and stable chronic brain signal recording for complex brain computer interfaces. The WIMAGINE implant was designed for epidural ECoG recording. The main body of the WIMAGINE implant hosts the recording device, including the recording electrodes and the electronics. Two flexible antennas integrated in a biocompatible silicone rubber are attached to the main body for wireless inductive powering and data communication. The main body of the implant is a 50mm-wide cylinder. Its base is flat and hosts 64 2.3mm-wide data recording electrodes, spaced by 4mm in one dimension and 4.5mm in the other (Figure 4). The top is pseudo-spherical to match as well as possible the average human skull curvature.

The WIMAGINE implants filter the data with a bandpass filter between 0.5Hz and 300Hz and transmit it wirelessly to a helmet worn by the patient. The helmet also hosts radio frequency (13.56MHz)

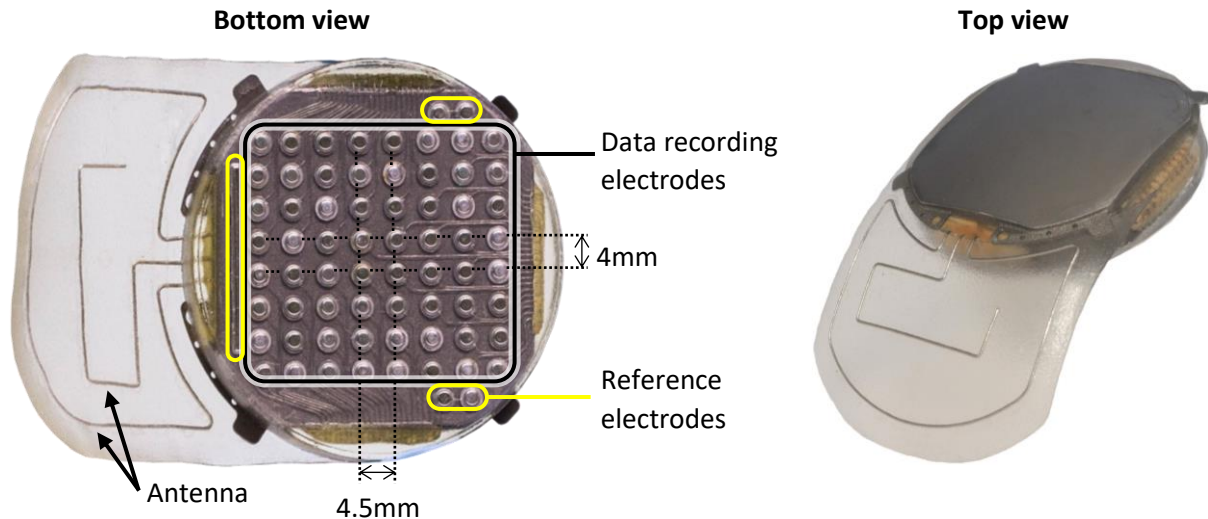


Figure 4 Bottom and top view of the WIMAGINE implant used in the clinical trial to record ECoG data from the enrolled subjects. Each implant has 5 reference electrodes and 64 electrodes used to record cortical brain activity. The implants are hermetically packaged in biocompatible materials in order to ensure the subject's safety and long-term reliability.

antennas to power the implants inductively. The helmet is connected to a base station in charge of generating the high frequency field used to power the implants and transmitting neural data to a computer in the MICS band (402-405MHz).

### 1.3. Effectors of the Clinatec BCI clinical trial

This clinical trial is performed with the objective of providing motor functional compensation to tetraplegic patients with the use of a several effectors. Thus, the main effector is a full body exoskeleton. However, subjects should first learn to use BCIs to control effectors with low complexity and then gradually work their way to more complex effectors. Therefore, a large set of effectors was used in this clinical trial to allow progressive BCI training, from binary control to high dimensional continuous control (Figure 5). The Enhancing MobilitY (EMY) exoskeleton was designed for this clinical trial, taking into account the additional requirements of being usable by the target population: the exoskeleton must be extremely safe because tetraplegics may not feel if they are damaging themselves with it, and it must be able to safely carry the weight of its users because they are unable to do so. In addition to the exoskeleton, other real effectors are used in the clinical trial. An electronic wheelchair was used as a BCI effector to demonstrate the capabilities of the BCI system when compensating for locomotion only. A commercial robotic arm (JACO from Kinova Robotics) was also used in the clinical trial for 2D and 3D hand movements.

In addition to these real effectors, virtual effectors were designed for this clinical trial. Virtual effectors are more convenient than real effectors in order to perform experiments from subjects' homes. Indeed, transporting or duplicating complex effectors such as a full body exoskeleton individually fitted to a tetraplegic is not a straightforward task. Additionally, virtual effectors are less complex to design, which makes it easier to have several virtual effectors with varying complexity level. However, virtual effectors do not provide actual motor capabilities. Actual motor function can be a major engagement factor for tetraplegics during experiments, which could make virtual effectors less engaging than real effectors. Nevertheless, virtual effectors can be made engaging through gamification of the tasks.

The real and virtual effectors designed for the clinical trial cover a wide range of motor complexity, and allow both in-lab and at-home training.



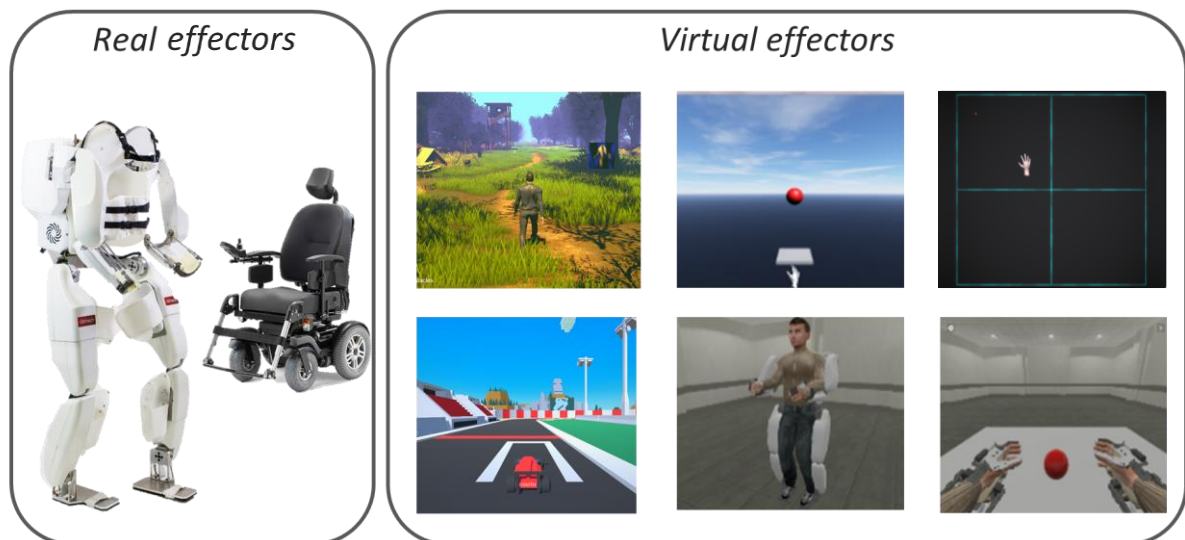


Figure 5 Example of real and virtual effectors used in the scope of the clinical trial. Real effectors included a full-body exoskeleton and an electronic wheelchair. Several virtual effectors were used, including the ones described here from left to right and top to bottom: a human avatar in a forest environment, which can either stand still or walk forward at a fixed speed (binary control); a platform that can be moved left or right in order to catch falling objects (one dimensional continuous control); a hand-shaped cursor that can be moved to reach targets in a center-out task (bi-dimensional continuous control); a virtual racing car on a track (bi-dimensional continuous control); and a replica of the full-body exoskeleton, seen here in third-person and first-person views (from binary control to potential full control of the exoskeleton, depending on the task designed and degrees of freedom made available for control to the user).

#### 1.4. Software chain

Multiple software pieces are used to transform the neural signals acquired by the recording system into movement commands for the effectors of the BCI (Figure 6). The neural signals are sent from the base station of the recording system to a computer. These digitized signals are received by the Wireless Implant Software Control Interface (WISCI) which makes them available for real-time batch processing by the next software. The ECoG signals are transmitted to the Adaptive Brain Signal Decoder (ABSD) software, which is in charge of performing signal processing for motor imagery decoding, training of decoders used to control the BCI and using them to estimate the patient’s movement intentions. The commands generated by ABSD are then treated differently depending on the effector. For the EMY exoskeleton, the commands generated are treated by the EMY Motion Manager (EMM) and EMY Motion Controller (EMC) in order to generate the correct motor commands for the exoskeleton. For virtual effectors, the commands generated are treated through a dedicated portal software in order to generate the appropriate movements of the virtual effectors.

In the scope of this thesis, we focus on the component of the software chain in charge of the training and usage of the control decoder. As mentioned earlier in this chapter, several properties are required from the control decoder in order to control motor effectors with multiple degrees of freedom. For

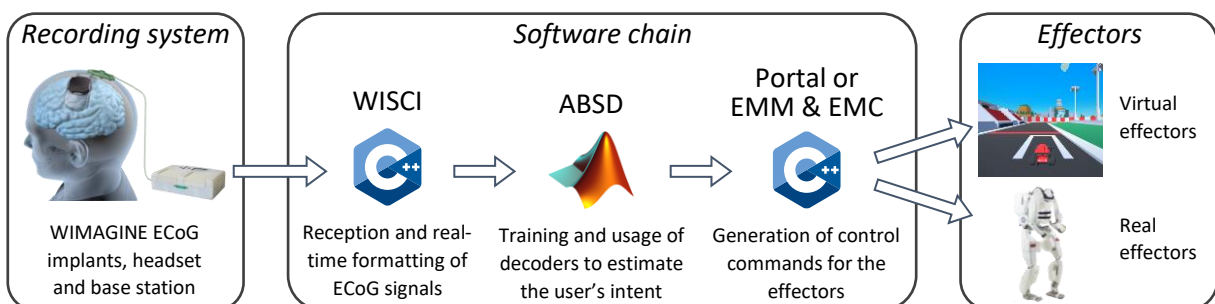


Figure 6 Diagram of the place of the software chain in the BCI. The software chain is made of several pieces of software, each with its dedicated purpose.

this clinical trial, a new algorithm was developed specifically to address these requirements. The Recursive Exponentially Weighted N-Way Partial Least Squares (REW-NPLS) with recursive validation of hyperparameters algorithm (Eliseyev et al., 2017) fits the requirement aforementioned. The REW-NPLS can be used to train the control decoder online in real-time. The algorithm is compatible with BCIs that have a large number of degrees of freedom, it is well-suited to high dimensional data and can take advantage of the tensor form of input data. Additionally, the REW-NPLS is quite resistant to noise (Eliseyev et al., 2017; Eliseyev and Aksenova, 2013).

In addition to the REW-NPLS, a Hidden Markov Model (HMM) is used in order to stabilize the different motor states activated in each effector (Moly et al., 2022). For instance, when controlling the EMY exoskeleton only one motor state (e.g., movement of the right arm, grasping of the left hand, walking, etc.) can be activated at any given time. The movements performed in each motor state (e.g., trajectory of the right hand, grasping state, walking state, etc.) are controlled each by a different decoder trained with the REW-NPLS. The transition between states is handled by the HMM. This separation of motor states removes parasitic movements related to motor states different from the one the user intended to activate. Indeed, if the user tried to move his right hand the output of the decoder in charge of the movement of the left hand will not be exactly zero. Without the HMM, this non-zero output would result in small parasitic movements of the left hand. This also enables a true idle state to be created when the user tries not to move (Schaeffer and Aksenova, 2016).

## II. Clinical trial progress

The “BCI and Tetraplegia” clinical trial (ClinicalTrials.gov identifier: NCT02550522) was approved by French authorities: Agence nationale de sécurité du médicament et des produits de santé (ANSM) with the registration Number: 2015-A00650-49 and the ethical committee (Comité de Protection des Personnes - CPP) with the Registration number: 15-CHUG-19.

### II.1. Inclusion criteria and subjects enrolled

In order to be eligible for inclusion in the clinical trial, participants must be between 18 and 45-years-old and fluent in French. Patients had to be in need for additional mobility and their neurological deficit had to have stabilized. They had to sign informed consent and be registered with the French social security. Exclusion criteria were contraindications to EEG, MEG or MRI, previous brain surgeries, current use of anticoagulant treatments or substance dependence, depression or neuropsychological sequelae.

To this date, three patients were included in the clinical trial. A first patient was implanted in May 2016 with WIMAGINE ECoG implants. Due to technological issues, the implants stopped working rapidly after the implantation and were explanted. Subsequently, the patient was excluded from the clinical trial. A second subject was implanted in June 2017 after the technological issues from the first patient were solved. The second patient participated in the clinical trial until mid-2021. A third subject was implanted in November 2019 and is currently actively taking part in the clinical trial.

### II.2. Implantation

Prior to surgery, patients included in the clinical trial performed motor imagery for the movement of each limb under fMRI and MEG in order to localize the relevant areas of the sensorimotor cortex. They then underwent bilateral implantation of two ECoG-recording wireless devices. Implantations were performed under general anesthesia with Image Guided Functional NeuroSurgery. The recording devices were implanted into the skull within a 25mm radius craniotomy. The contact between the devices and the dura mater was made at the level of the previously selected optimal area of the sensorimotor cortex.

### II.3. Progressive training of subjects enrolled in the clinical trial

Subjects enrolled in the clinical trial underwent training to control the various effectors available. At first, the effectors controlled were limited to low numbers of degrees of freedom. The number of degrees of freedom was increased when subjects achieved good control over the BCI. The decoders used were trained in an online adaptive manner, which means the actions performed by the effectors were determined by the output of the control decoder, even during decoder training sessions. Therefore, the subject had feedback on his control during training. This training method is better than classical open-loop training as it reduces the difference between the training dataset and the data acquired during actual use of the BCI (Shenoy et al., 2006).

The results that were published from the clinical trial are promising. The quality of the signal recorded with the WIMAGINE implants stayed stable over multiple years (Larzabal et al., 2021). Control of a complex motor BCI was achieved: the second subject implanted managed to control 8 continuous degrees of freedom of the EMY exoskeleton and its virtual counterpart (Benabid et al., 2019; Moly et al., 2022). These included position of the right hand (3 DoF), position of the left hand (3 DoF), prono-supination of the right hand (1 DoF) and prono-supination of the left hand (1 DoF). In addition to these motor states, the subject was also able to trigger at will the idle state. The decoding models used to control the BCI were usable for several months without needing daily updates. There was, however, a slow and gradual decrease of performance over time when using these fixed models. Additionally, a substantial amount of labeled training data was required in order to train the models. For instance, the decoder used to control the EMY exoskeleton was trained during 6 recording sessions over the span of 2 weeks and approximately three and a half hours of data were used for its calibration. The third subject's training was perturbed by the COVID-19 pandemic, and his training results are not yet published.

## III. Objectives of the thesis

In this thesis, we propose a potential improvement for BCIs. It is developed in the scope of the BCIs used in the Clinattec clinical trial, but without loss of generality for any other BCIs. This thesis stems from the realization that an important number of flaws of BCIs can be linked to the cost of acquiring labeled data during training sessions. Lack of labeled data can be a source of non-optimal decoder accuracy (for instance when overfitting the data). Lack of labeled data is also one of the reasons why concept drifts have a strong impact on BCI decoders' performances. Indeed, if labeled data can be acquired after concept drifts occur, decoders could be updated using this new data in order to minimize the impact of the drift. Therefore, labeled data acquisition is necessary in order for most BCIs to function. However, acquisition of labeled data is costly: it requires a controlled environment, most often supervision by experienced researchers and a substantial amount of time from both the user and the research team (as was the case in this clinical trial). Additionally, during labeled data acquisition the user is constrained to perform specific actions in order to be able to train the decoder of the BCI, and therefore cannot freely use the BCI. The time and material cost of labeled data acquisition also increases with the complexity of the BCI. For instance, a specific platform was built during the clinical trial in order to enable proper training. A training table was designed with LEDs at known positions. The lighting states of the LEDs could indicate which LED had to be reached, or if wrist rotation was required (Figure 7). The position of the LEDs relatively to the exoskeleton must be known with as much precision as possible in order to ensure that the labels used to train the decoder are accurate. This cost



Figure 7 Environment used for the training of the decoders used to move the hands of the exoskeleton in 3D. At any time, one LED was turned on and the subject had to try to reach the LED using motor imagery.

of label acquisition is also one of the barriers to long-term use of BCIs at home. Indeed, due to concept drifts, the decoders of the BCI must be updated regularly. However, the requirements for labeled data acquisition make it hard to perform updates at home, especially for complex motor BCIs. BCIs could therefore strongly benefit from methods that either reduce the cost of labeled data acquisition or do not use labeled data.

Several strategies could potentially be used in order to label data, with different associated costs:

- Labels can come from an agent external to the user or the BCI. This is the case in classical training paradigm, where the user is instructed to perform a task either by a researcher or by a cue in the training environment (Lotte et al., 2018). Depending on the type of BCI, labels can also be obtained using other external agents such as eye-trackers, cameras or other sensors (Allison et al., 2012; Yong et al., 2012). For instance, eye-trackers could be used in some cases to acquire labeled data for communication BCIs, such as P300 spellers, and cameras could be used in motor BCIs to identify objects and estimate which one a user may try to reach or grasp. However, subjects that are in need for communication BCIs may not be able to have precise control over their eye movements (Birbaumer, 2006; Hinterberger et al., 2003), which impairs the usability of eye-trackers to label data. Cameras can identify objects, but they would not allow data labeling when a user performs a movement that is not directed at an object.
- Alternatively, labels could be derived from internal agents, such as signals generated by the user. These signals could be generated either consciously or unconsciously by the user. For instance, the user could regularly say a predetermined word or sentence (e.g., 'correct BCI control') when the output of the BCI is consistently correct. This would mean that the output of the decoder of the BCI was close enough to the actual labels, and therefore these outputs of the control decoder could be used as labels. However, consciously generated labels (i.e., the user provides vocal feedback on the BCI's action) may impose a strong strain on the user and are not optimal for chronic BCI use. Alternatively, unconscious user-generated signals could be used to determine the correctness of actions performed by the BCI. Some signals such as heart rate, skin conductance response or pupillometry may contain such information (Wheeler, 2019). However, the time constant of variations in heart rate and skin conductance response is most likely too high to accurately reflect correctness of complex motor BCIs, which may control their effectors several times per second as is the case in the Clinatex clinical trial. Although pupillometry is known to change rapidly, it currently requires external equipment in order to be recorded accurately.

Finally, labels could also be derived from neural data (Lotte et al., 2018). We chose to derive estimated labels using neural data in this thesis. Additionally, deriving control labels using neural data acquired

from the sensorimotor cortex is the method with the lowest additional material cost, as patients enrolled in the clinical trial are implanted with ECoG-recording implants over the sensorimotor cortex. Therefore, no additional hardware was needed, and test could be performed using data recorded prior to this study. We use neural signals correlated to the perceived performance of the BCI by the user in order to estimate control labels. These neural signals reflect whether the actions performed by the BCI match the intention of the user. We call such signals neural correlates of **BCI task performance**. More specifically, in this thesis we use neural correlates of **motor task performance (MTP)**, in order to estimate labels for the control decoder. **The goal of the thesis is to perform a proof of concept that neural correlates of MTP detected from the sensorimotor cortex can be used to perform real-time adaptation of the control decoder of a complex motor BCI**, e.g. a BCI that controls more than 1 continuous degree of freedom. In order to achieve this, several building bricks are needed:

- A detection system. Neural correlates of MTP must be detected at the single trial level and in real-time. It is also beneficial to detect them from the sensorimotor cortex. In this clinical trial, as is typically case in motor BCIs clinical trial, implants are positioned over the sensorimotor cortex only. Detection of neural correlates of MTP from the sensorimotor cortex makes this adaptive process easier to use for motor BCI clinical trials. Additionally, these neural correlates of MTP should be detectable even when state of the art motor BCIs are used (i.e., BCIs that have multiple continuous outputs).
- A labeling system. A way to derive labels for the control decoder using the output of the MTP decoder is need.
- A control decoder that can be updated online in real-time. This last point does not need additional development, as the control decoder used in the Clnatec clinical trial can already be trained online in real-time.

# Chapter 3: Auto-adaptive BCIs as solution to reduce BCI decoder training

Retraining BCI control decoders regularly due to concept drifts comes at large costs in terms of usability. Training sessions most often require specific training environments and researcher supervisions, which cannot be made at the user's home. Additionally, during these training sessions the user must perform specific tasks instead of freely using the BCI for his personal use. One of the solutions developed in the BCI field is the use of adaptive BCIs. As their name suggest, adaptive BCIs are BCIs that use decoders that can be adapted during online use. We distinguish between two kinds of adaptive BCIs: adaptive BCIs that must adapt their control decoder during dedicated training sessions and adaptive BCIs that can adapt their control decoder during free use. The goal of the first kind is to provide control feedback to the user during training, enabling co-adaptation (Vidaurre et al., 2011b) and reducing the potential differences in neural signals between training and testing data (Shenoy et al., 2006). To distinguish between these and adaptive BCIs that can adapt their control decoder during free use, we refer to the latter as 'auto-adaptive BCIs' (or aaBCIs). Auto-adaptive BCIs do not rely on classical training sessions in order to obtain labeled data and update their control decoder. Instead, two main strategies are used to obtain labeled data: either the output of the control decoder is used as ground truth and is used as labels, or the labels are inferred using another source. The first strategy is usually referred to as 'unsupervised' adaptation (even though it often relies on supervised machine learning algorithms).

## I. Auto-adaptive BCIs based on unsupervised adaptation of control decoders

The most straightforward way to perform adaptation of control decoders without access to real labels is to use the output of the control decoder as labels. This strategy is based on the assumption that the output of the control decoder is mostly correct.

In a study by Shenoy et al. (2006), subjects performed a binary motor imagery task that was detected by an EEG-based BCI. After a preliminary training session, a Linear Discriminant Analysis (LDA) decoder was trained to differentiate between two different motor imagery tasks (Bishop, 2006). Additionally, they combine the LDA decoder with a Common Spatial Pattern feature extraction step (Guger et al., 2000). They compare three different unsupervised adaptation strategies. For each of them, the output of the LDA is considered as correct and used as labels for adaptation. In the first strategy, the decoder is retrained completely. In the second strategy, they only perform adaptation of the bias of the decoder, and in the third strategy they recompute the Common Spatial Patterns only. All three adaptive methods performed better than the original fixed classifier that was used to acquire the dataset. In a similar vein, Vidaurre et al. (2011b, 2011a) performed unsupervised adaptation of binary LDA classifiers for motor-imagery-based BCIs using EEG. Offline and online results, on simulated neural data or real EEG data, confirm the performance gains obtained from adaptation that were described by Shenoy et al.

In an online EEG study, Gu et al. (2013) performed a short (1min) training of least-squares support vector machine for a P300 BCI speller. Although control performances were initially low, the decoder was adapted online in order to reach above 90% spelling accuracy for each of the eight subjects. They



provide an upgraded algorithm for online training that reduces computational cost. However, this improvement does not completely remove the dependency of the computational cost on the number of labeled datapoints. Finally, their approach can be suited to shorten training of BCIs by performing part of the training during self-directed use. It cannot be used for lifelong learning, or to follow concept drifts as the computational costs will explode at some point.

In a similar vein, Kindermans et al. (2014) performed an online study on unsupervised learning in an auditory event-related potential BCI paradigm. They use a LDA decoder for the binary distinction between target and non-target audio stimuli, enabling writing letters sequentially. Their decoder is initialized randomly, and updated during usage to distinguish between the two classes. Similarly to other adaptation based on LDA decoders, the mean and bias of the classes distributions are updated using the Expectation Maximization algorithm (Dempster et al., 1977). Although this unsupervised adaptation technique can separate the two classes, it does not provide knowledge on which class is the target class and which class is the non-target class. They rely on the binary aspect of the decoding task in order to solve this issue. Finally, their unsupervised decoder can perform comparably to a classic supervised decoder on this task, but is unlikely generalizable to more complex BCIs.

In an online study with a monkey implanted with intracortical electrodes, Li et al. (2011) used a modified Bayesian linear regression model (Bishop, 2006) and smoothed its output with an unscented Kalman filter (Li et al., 2009) in order to control a cursor in two dimensions during various tasks. They performed unsupervised adaptation of the Bayesian model in batch updates using the output of the Kalman filter as labels. Their results show that their adaptation provided control over 29 days with a small decrease of performance in time, compared to the performance decrease of a fixed decoder. However, it should be noted that during control of the BCI, the monkey had to have his hand on a joystick that could move freely, and that he had used in previous experiments in order to control the same cursor instead of using the BCI. Therefore, it is highly likely that the control decoder uses neural correlates of motor execution in order to predict cursor movements. Using neural correlates of attempted movement or imagined movement could decrease the performance of the control decoder in a comparable study. Lower control decoder performances would lead to more uncertainty in the labels used for unsupervised adaptation, which could compromise the reported performances gain of this unsupervised adaptation strategy.

The results of all these studies on unsupervised adaptation of the control decoders of BCIs are promising. Indeed, each of them report improvement over static decoders. However, unsupervised adaptation has yet to be performed during human control of a BCI that control multiple continuous degrees of freedom, or any other BCI comparable to state-of-the-art motor BCIs. In some of the studies discussed above, unsupervised adaptation was performed successfully with binary decoders that had initially random performance or were initialized randomly. However, this was only possible because of specific properties of the BCI paradigms used, and would not be possible for most BCIs. Using outputs of a random control decoder for adaptation seems less likely if said decoder had multiple continuous outputs. Furthermore, we suggest that long-term unsupervised adaptation of control decoders with multiple continuous outputs is not optimal, even when their performances are initially far from random. Similarly to the study performed by Li et al., control decoders updated unsupervisedly may still have a slow decrease of performances with time. Decreasing performances of the control decoder will lead to less accuracy in the labels used for unsupervised adaptation, which can further decrease the performance of the updated control decoder. In addition, unsupervised adaptation is most likely not well suited to strong concept drifts because a sharp decrease in the performance of the control decoder may occur too fast for proper adaptation to take place. For these reasons, we suggest that

adaptation is optimal if labels can be estimated independently of the performance of the control decoder.

## II. Auto-adaptive BCIs based on weakly supervised adaptation of control decoders using neural correlates of task performance

In unsupervised adaptation, the output of the control decoder during self-directed control of the BCI is used as label for adaptation of the control decoder. Alternatively, labels can be inferred without relying solely on the output of the control decoder. We distinguish between labels that are estimated using neural data only and labels that are estimated using information external to the BCI system. Deriving labels from external sources during free-use of the BCI is possible, as was done by Yong et al. using an eye-tracker (2012). However, using an external source to estimate labels comes with additional constraints, that may restrict the use of the BCI. For instance, using an eye-tracker requires the user to have proper eye-movement control, and restricts BCI usage to environments where an accurately calibrated eye-tracker is available. To our knowledge, the study by Yong et al. is the only one deriving labels from an external source in order to perform adaptation of a BCI that would also be usable during self-directed use of the BCI. On the other hand, adaptation of control decoders using labels estimated using neural data (excluding unsupervised adaptation) is a hot topic in the BCI field. More specifically, the idea of using neural correlates of task performance for adaptation of control decoders was already discussed more than a decade ago (Buttfield et al., 2006). We use ‘task performance’ to refer to the difference between the actions performed by the BCI and the actions the user intended to perform. Neural correlates of task performance are neural patterns that contain information on the user’s conscious or unconscious estimation of how well the BCI performed the actions he desired. To date, every implementation of an auto-adaptive BCI using labels inferred thanks to correlates of task performance used Error Related Potentials or a simulated task performance signal (Artusi, 2012; Artusi et al., 2011; Blumberg et al., 2007; Buttfield et al., 2006; Gürel and Mehring, 2012; Llera et al., 2012, 2011; Spüler et al., 2012; Zeyl and Chau, 2014).

### II.1. The Error-related Potential (ErrP)

The Error-related Potential, or ErrP, was first described by Falkenstein et al. (1991). It is a waveform that is elicited in the EEG recordings of a subject after they perform an erroneous discrete action. The waveform is composed of several components (Falkenstein et al., 2000), the main ones being a negative deflection of the potential acquired over the fronto-central scalp area, roughly 50 to 100ms after the event that induced it, followed by a centro-parietal positive deflection of the potential (Chavarriaga et al., 2014). ErrPs can be detected at the single trial level (Parra et al., 2003), even when the erroneous actions is performed by a third party that the user observes or controls, such as a BCI effector (Ferrez, 2007; Ferrez and del R. Millan, 2008; Schalk et al., 2000).

Additionally, several studies suggest that ErrPs are relatively stable in time (Ferrez and del R. Millan, 2008; Olvet and Hajcak, 2009), with decoders being able to detect ErrPs at the single trial in EEG data more than 600 days after the training of the decoder (Chavarriaga and Millan, 2010). A study by Iwane et al. (2016) showed that ErrPs can be detected reliably not only across sessions, but also across different task conditions. They trained a decoder to detect ErrPs occurring when a user monitored a computer-controlled cursor making erroneous movements in a discrete 2D space. Their decoder was able to detect ErrPs at the single-trial level across different movement speed of the cursor. Finally, there is mostly evidence in favor of ErrPs being stable to variations in the tasks performed (Omedes et



al., 2013; Riesel et al., 2013). However, a study by Volker et al. (2018) reported poor generalization across tasks and good generalization across subjects.

Stability in time and across tasks is a valuable property of the ErrP for auto-adaptive BCI. Robustness of ErrP decoders across tasks means a single ErrP decoder can be used for auto-adaptive training of various control decoders of a BCI. Robustness of the ErrP decoder to time may be even more valuable. The goal of the aaBCI is to limit the drawbacks of supervised training session. If the decoder for neural correlates of task performances is not robust to time, it will have to be updated in a dedicated training session, mitigating part of the advantages of the aaBCI concept. The stability of the ErrP in time and across tasks makes it a well-suited correlate of neural performance for the purpose of aaBCIs.

Finally, the fact that ErrPs can be detected using EEG with good single trial accuracy explains why it is the neural correlate of task performance most used in the BCI field, not only in auto-adaptive BCIs, but also as a mental commands for BCIs (Chavarriaga et al., 2016) or for error correction during BCI operation (Even-Chen et al., 2018; Parra et al., 2003).

## II.2. ErrP based auto-adaptation

The first implementation of auto-adaptive BCIs were performed with simulated detection of neural correlates of task performance. Blumberg et al. (2007) performed an offline study of auto-adaptation of a binary motor imagery decoder on EEG data from the BCI competition III. They simulated an ErrP decoder with either perfect or 80% accuracy. They show that auto-adaptation of their ALDA decoder makes it outperform the fixed decoder, and reaches performances close to the one obtained if the decoder had been updated supervisedly. Similarly, Llera et al. (2012, 2011) used a simulated ErrP decoder and showed that auto-adaptation of control decoder are possible in both a MEG binary covert attention-based BCI and an EEG binary motor imagery-based BCI.

Artusi et al. (2011) also performed a simulation of an EEG-based aaBCI using ErrPs. In their experiment, subjects performed motor imagery (MI) of fast or slow right arm flexion and received fake feedback of the MI decoder output. They used the accuracies of ErrP detection of this binary BCI in a simulation of an aaBCI that has three discrete classes. They use only 'correct' data, i.e. data for which their ErrP decoder would not detect an ErrP and add it to a training dataset for their control decoder. They show that their simulated aaBCI can follow drift in their input feature space relatively well. In follow up studies, Artusi (2012) implemented a double threshold strategy on the output of the ErrP decoder when labeling trials as correct or erroneous, with the goal of reducing the number of mistakes made by the ErrP decoder. Although they use this double thresholding strategy in the context of a BCI that corrects mistakes instead of an aaBCI, this thresholding strategy could also be used for aaBCIs.

Zeyl and Chau (2014) performed adaptation of linear binary decoders using the output of a simulated ErrP decoder with accuracy similar to the one reported in the literature. They used EEG data from the BCI competition III, in which participants performed right-hand or left-hand motor imagery in order to control the horizontal position of a ball falling vertically on a computer screen. Interestingly, they compared two different learning strategy. With the first strategy, only data detected as erroneous was used to update the control decoder. Samples that are estimated erroneous are labeled with the other motor class and used for supervised update of the decoder. The second strategy is to use both correct and erroneous data to update the control decoder. The first strategy was based on the assumption that mistakes are made by the control decoder when there is concept drift and that these samples are most important to follow concept drifts. This strategy is compared to the more intuitive strategy of using every sample for adaptation. Their results indicate that it is generally better to use both erroneous and correct samples for adaptation, especially when the label estimator (the simulated ErrP decoder in their case) is imperfect.

Although less frequent, some aaBCIs were implemented using actual ErrP decoding and real neural data. For instance, Spüler et al. (2012) performed online adaptation of a decoder using ErrPs. In their work, the user controlled an EEG-based communication BCI using code-modulated visual evoked potentials. They performed time-lock single-trial detection of ErrPs after selection of each target. Only when no ErrP was detected, the data was used for adaptation of the one-class SVM decoder (Schölkopf et al., 2001).

Although the idea of auto-adaptation of control decoders using neural correlates of task performance has been in the BCI field for more than a decade, for now, neural correlates of task performance are mostly used in order to correct mistakes online (Cruz et al., 2018; Even-Chen et al., 2017; Kreiling et al., 2009; Parra et al., 2003; Yousefi et al., 2019) rather than perform decoder adaptation.

### II.3. Auto-adaptation of control decoders with multiple continuous outputs

All the auto-adaptive studies using ErrPs discussed above are limited to control decoders for classification, and mostly binary classification. State-of-the-art motor BCIs can control several continuous degrees of freedom (Benabid et al., 2019; Willett et al., 2021; Wodlinger et al., 2014). The main issue with ErrP based auto-adaptation is that ErrPs are known to be elicited by erroneous discrete events. Although continuous control can elicit discrete erroneous events, errors during multi-dimensional continuous control are not limited to discrete events. State-of-the-art motor BCIs control effectors with multiple continuous degrees of freedom with a high control rate (Shanechi et al., 2017), therefore trajectories can be modified slightly every few milliseconds. Errors during trajectory control in two or three dimensions are most likely to occur as gradual deviations from the desired trajectory rather than jumping from a perfect trajectory to a strictly erroneous one.

#### *Neural correlates of TP during control of continuous degrees of freedom*

Nevertheless, several studies investigated detection of neural correlates of task performance, including ErrPs, during continuous tasks. Although there are some studies that investigate ErrPs in one dimensional continuous task (Kreiling et al., 2009; Milekovic et al., 2013, 2012; Wilson et al., 2019), we do not detail them here. Errors in a continuous one-dimensional task are basically discrete. Either the controlled effector is going toward the target or away from it. We focus instead on studies that investigated the existence of neural correlates of task performance during task with more than one continuous degree of freedom.

Lopes-Dias et al. (2018) performed detection of ErrPs during 2D control of a cursor using a joystick. They even perform asynchronous detection of erroneous trials. However, their error trials were created artificially by adding sudden changes in the trajectory of the cursor by 90 degrees and removing control of the cursor from the user. Additionally, the users were instructed to stop trying to use the joystick to direct the cursor to the target as soon as they noticed such an erroneous event occurred. This strategy has two flaws. The erroneous events created are 'discrete' erroneous events in the sense that a very large error is suddenly introduced in an otherwise easy to control paradigm. As mentioned previously, during continuous control of multiple degrees of freedom errors are not necessarily sudden, but can also occur gradually. Although the detection of sudden errors during continuous 2D control performed in Lopes-Dias et al. (2018) is interesting, it is not sufficient for proper detection of errors of all types in multidimensional control. The second flaw is due to the user stopping its intent of performing the task when he realizes he has lost control over the cursor. There is no way to ensure that the decoder detects actual error onsets and not disengagement of the user in the task, or even simply the end of the motor action responsible for cursor control (joystick displacement).

In a following study, Lopes-Dias et al. (2019) managed to perform online asynchronous detection of erroneous events during 2D control of a robotic arm using hand movements. However, the errors were again introduced artificially as loss of control over the robotic arm. Therefore, the main contribution from this study resides in the ability to detect online and asynchronously ErrPs and not in the detection of neural correlates of task performance during continuous multi-dimensional control of the robotic arm, since it is limited to discrete erroneous events.

Spüler and Niethammer (2015) designed a study on a variation of a protocol used by Milekovic et al. (2013, 2012). In their experiments, subjects controlled a cursor in 2D using a joystick and had to avoid blocks falling from the top of the screen. However, execution errors were introduced artificially as sudden deviations of the cursor by 45°, 90° or 180°. They were able to perform single-trial, asynchronous detection of ErrPs after execution errors with EEG. Although execution errors occurred during continuous control of a cursor in 2D, these errors were again sudden errors and not gradual deviations.

A study by Omedes et al. (2015) investigated detection of errors in EEG signals during monitoring of a reaching task in a 2D plane. They designed two types of errors: one similar to previously discussed studies, in which at one point the cursor suddenly and sharply deviated from the correct trajectory, and another one in which the trajectory gradually deviated from the correct one. They were able to detect asynchronously errors of the first kind. However, the decoder they trained to detect the second kind of errors was not able to perform better than chance level. Interestingly, they report that a decoder trained to detect the first kind of errors performed slightly better than chance level for the detection of errors of the second kind. This ability to detect gradually occurring errors is what is needed for error detection in state-of-the-art motor BCIs. However, it should be noted that gradual errors were made by deviating the trajectory toward one of the non-selected targets. Therefore, the trajectory of the cursor was 2D but only 3 different trajectories were possible, and subjects knew at which point the trajectories started being different. In state-of-the-art motor BCIs, unlimited number of trajectories are possible and trajectory deviations are possible at any given point.

To date, the only neural correlates of task performance that were reported during continuous control of multi-dimensional effectors are linked to discrete errors in these continuous control task. Such neural correlates could be sufficient to label some data as erroneous. However, it does not ensure that data for which such neural correlates are not detected is correct. Indeed, slow deviations could occur without eliciting such discrete neural correlates of task performance. For the purpose of the aaBCI, it would mean that it is not possible to label data as correct, and that the data labeled as erroneous represents only a portion of all the erroneous data. **We thus consider that ErrPs are not optimal neural correlates of task performance for the purpose of enabling auto-adaptation in state-of-the-art motor BCIs.** Similarly, any neural correlates of task performance that is event-locked to a discrete correct or erroneous event would not be suitable for aaBCIs. In contrast to these neural correlates of Event-locked Motor Task Performance, or eMTP, we consider the existence of neural correlates of *continuous in time* Motor Task Performance, or cMTP. By continuous in time, we mean neural correlates that do not require a discrete event to be elicited, but are correlated to the instantaneous task performance of the BCI system perceived by the user.

Interestingly, Gürel and Mehring (2012) performed a simulation study of an auto-adaptive BCI that would use such a cMTP signal. They are able to perform auto-adaptive update of a control decoder used to control a cursor in two dimensions in a reaching task. In this simulation, they simulate both the user control (i.e., the user is simulated) and a 'binary neuronal error-signal'. Compared to the previous studies with simulated neural decoder of task performance, they did not mimic a neural decoder of ErrPs as their task performance neural decoder provides an estimation of the performance

of the control decoder at each time step and does not require discrete events to do so. More specifically, they simulate a task performance decoder that detects deviation in the ideal trajectory that are superior to 20 degrees. They also studied the impact of the reliability of the cMTP signal (from 0 to 80%), and show that adaptation is possible as soon as the reliability of the task performance decoder is above 50%.

Although the existence of such neural correlates of cMTP has not been reported yet, in this thesis we investigate their existence and usability in the context of aaBCIs.

#### II.4. Detection of neural correlates of TP in the sensorimotor cortex

In addition to the issue described above, ErrPs are also ill-suited for auto-adaptation of state-of-the-art motor BCIs due to the brain structure they can be recorded from. State-of-the-art motor BCIs use motor imagery or attempted motor execution as neural signals to control the BCI. They also often rely on invasive recording techniques, with micro-electrode arrays or ECoG grids located over the sensorimotor cortex (Benabid et al., 2019; Wodlinger et al., 2014). As ErrPs are generally not recorded from the sensorimotor cortex, they are ill-suited for the auto-adaptation of current state-of-the-art motor BCIs. We realize that this issue could be resolved in future clinical trial for motor BCIs by changing the location of the recording system, or by implanting additional acquisition systems. However, for current state-of-the-art motor BCIs, including the one used in the Clnatec BCI clinical trial, invasive neural signal acquisition can only be done from the sensorimotor cortex. Therefore, single trial detection of neural correlates of task performance from the sensorimotor cortex is required. We thus investigated the existing literature on neural correlates of task performance in the sensorimotor cortex.

The brain network in charge of error processing has been shown to not be limited to the regions ErrPs originate from (Gueguen et al., 2021; Jung et al., 2010; Miltner et al., 1997; Wilson et al., 2019). More specifically, several studies report the existence of neural correlates of task performance in the sensorimotor cortex.

Van Schie et al. (2004) demonstrated the existence of EEG error correlates in the motor cortex by showcasing the variability of the lateralized readiness potential between correct and erroneous response in an Eriksen flanker task. In a MEG study, Koelewijn et al. (2008) reported a stronger beta (15-35 Hz) rebound after an outcome error than after a correct task outcome, both when observing or performing a motor task.

In two successive studies, Milekovic et al. (2013, 2012) performed detection of neural correlates of TP from the sensorimotor cortex. The subjects in these studies were implanted with large ECoG grids covering part of the primary and pre-motor cortex for the sake of pre-neurosurgical epilepsy diagnosis. They used a joystick to control the horizontal movements of a space ship in a video game, in order to avoid blocks falling from the top of the screen. In this setup, two kinds of errors occurred: outcome error, when the spaceship collided with one of the blocks, and execution errors, when the movement direction of the spaceship was artificially changed to the opposite of the one imputed by the joystick. They managed to detect asynchronously both outcome and execution errors at the single-trial level, using a combination of temporal and time-frequency features. The averaged normalized spectrograms of neural responses to outcome and execution errors show a strong response in the gamma band (at > 90Hz).

A study by Völker et al. (2018) investigated error-processing dynamics over the scalp using non-invasive or ECoG during an Eriksen flanker task. They found that error correlates were mainly located in low frequency bands (<30Hz) and in the high-gamma frequency band (60-90 Hz) for ECoG and EEG.

Additionally, they report that such neural correlates could be recorded using both modality from the sensorimotor cortex (among other areas). However, they did not perform single-trial detection of these error correlates. Similarly, Wilson et al. (2019) studied the cortical repartition of high frequency components (70-100 Hz) of neural correlates of task performance during control of an ECoG and motor imagery-based BCI with one continuous output. The neural correlates they report are also detectable from the sensorimotor cortex.

Interestingly, the MTP correlates reported in all these studies were largely in the time-frequency domain rather than in the temporal domain like ErrPs. More importantly, all these studies reported neural correlates of eMTP. To our knowledge, to date no study reported the existence of cMTP correlates in the sensorimotor cortex.

### III. Proposed mechanism of auto-adaptation for state-of-the-art BCIs

As described in this chapter, most of the auto-adaptive BCIs implemented or simulated use either unsupervised adaptation, or adaptation using labels inferred thanks to a signal correlated to task performance, such as the ErrP neural correlate. We suggest that the latter is better suited for long-term adaptation. **We also suggested that auto-adaptive BCIs should use neural correlates of continuous in time Motor Task Performance (cMTP) in order to allow adaptation of decoders with multiple continuous outputs.**

In the auto-adaptive BCI framework developed in this thesis, an auto-adaptive module is added to the BCI online training loop (Figure 8). Neural correlates of cMTP are detected using a decoder trained for this task. The output of the cMTP decoder is combined with the output of the control decoder in order to produce estimated labels for the control decoder. These estimated labels are then used to train or update the control decoder of the BCI in an online and adaptive manner. The aaBCI framework aims at enabling auto-adaptation of any BCI during free control by the user.

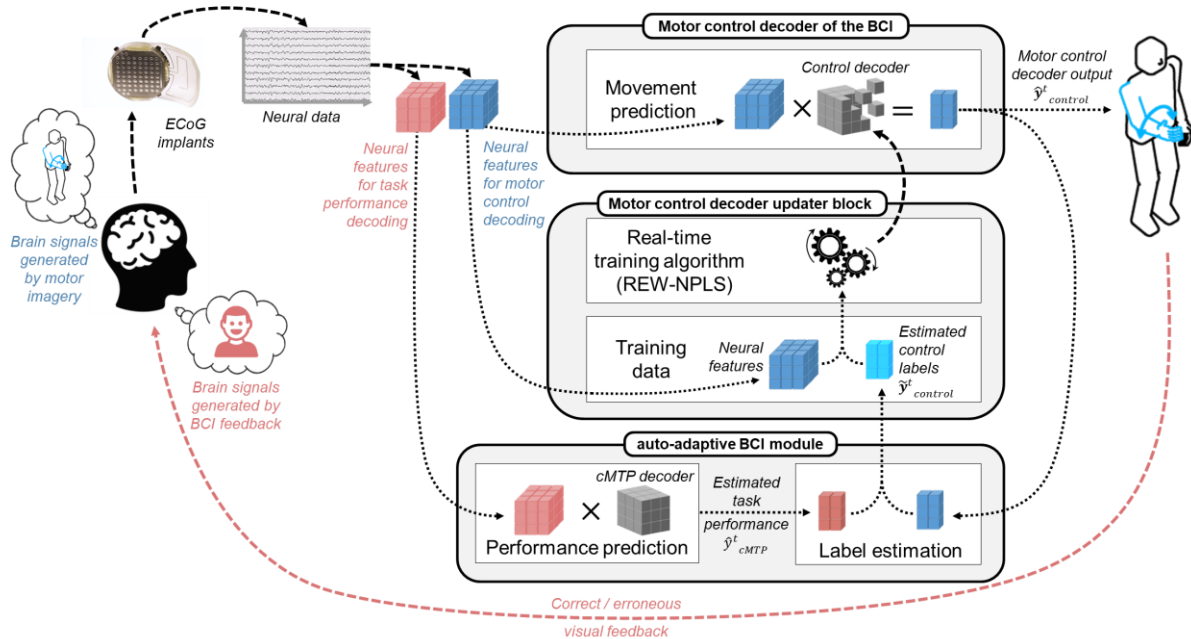


Figure 8 Diagram of the auto-adaptive BCI framework. An auto-adaptive BCI module is added to the classic BCI loop. The aaBCI module is responsible for the detection of neural correlates of continuous in time task performance (cMTP), as well as the estimation of labels for the control decoder ( $\hat{y}^t_{control}$ ) based on the output of the cMTP decoder ( $\hat{y}^t_{cMTP}$ ) and the output of the control decoder ( $\hat{y}^t_{control}$ ).

# Chapter 4: Experimental design

## I. Data acquisition

The subject in this thesis was the second patient included in the clinical trial. He is 28-year-old male who had tetraplegia following a C4-C5 spinal cord injury (Figure 9). The subject had some residual motor control over his upper limbs: biceps could be contracted (American Spinal Injury Association Impairment [ASIA] scores: 4 right, 5 left) as well as extensors of the wrist (0 right, 3 left). All other muscles below were scored 0 on the ASIA scale (Benabid et al., 2019). The subject was implanted with two WIMAGINE (Mestais et al., 2015) ECoG implants in June 2017 as a participant in the clinical trial “BCI and Tetraplegia”. The goal of this clinical trial is the control of various motor effectors, which is commonly performed using motor imagery (Lebedev and Nicolelis, 2017) and using neural data from sensorimotor cortices. Therefore, implants were positioned over the left and right sensorimotor cortex (Figure 10). Each WIMAGINE implant transmitted neural data at a sampling rate of 586Hz from 32 out of its 64 electrodes because of limited data transfer rates. The data treatment software used in the clinical trial transmitted data by blocks of 59 points, which corresponds to approximately 0.101s (we consider it to be 0.1s for the remainder of the thesis). Time steps  $t$  in this thesis refer to each time a new block of 59 points is received.

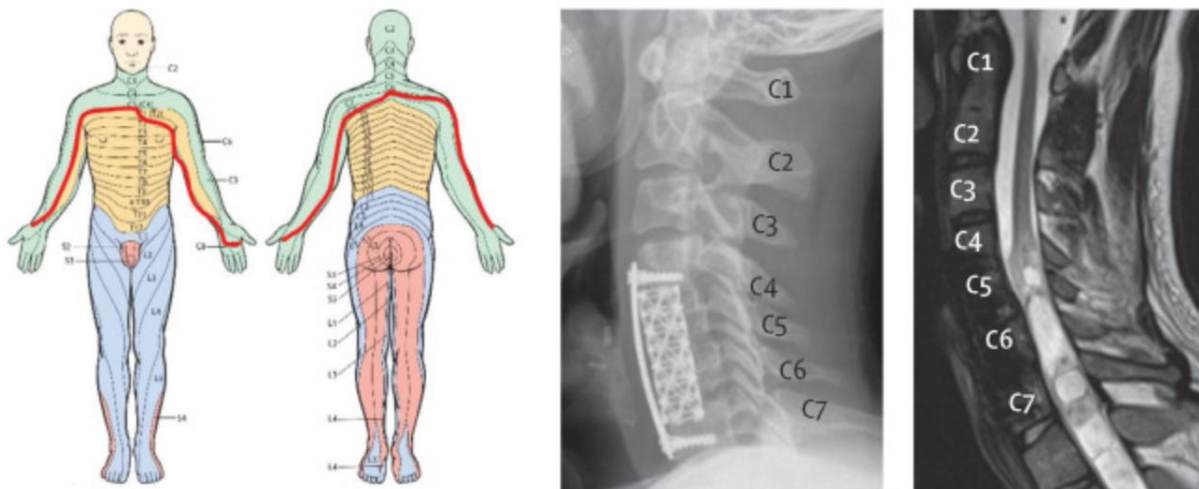


Figure 9 (A) Representation of the patient's sensorimotor state. The patient had no sensitivity nor motor control below the metameric level represented by the red line. (B) MRI and radiography of the patient's spinal cord at the level of the lesion. Figure taken from Benabid et al. (2019)



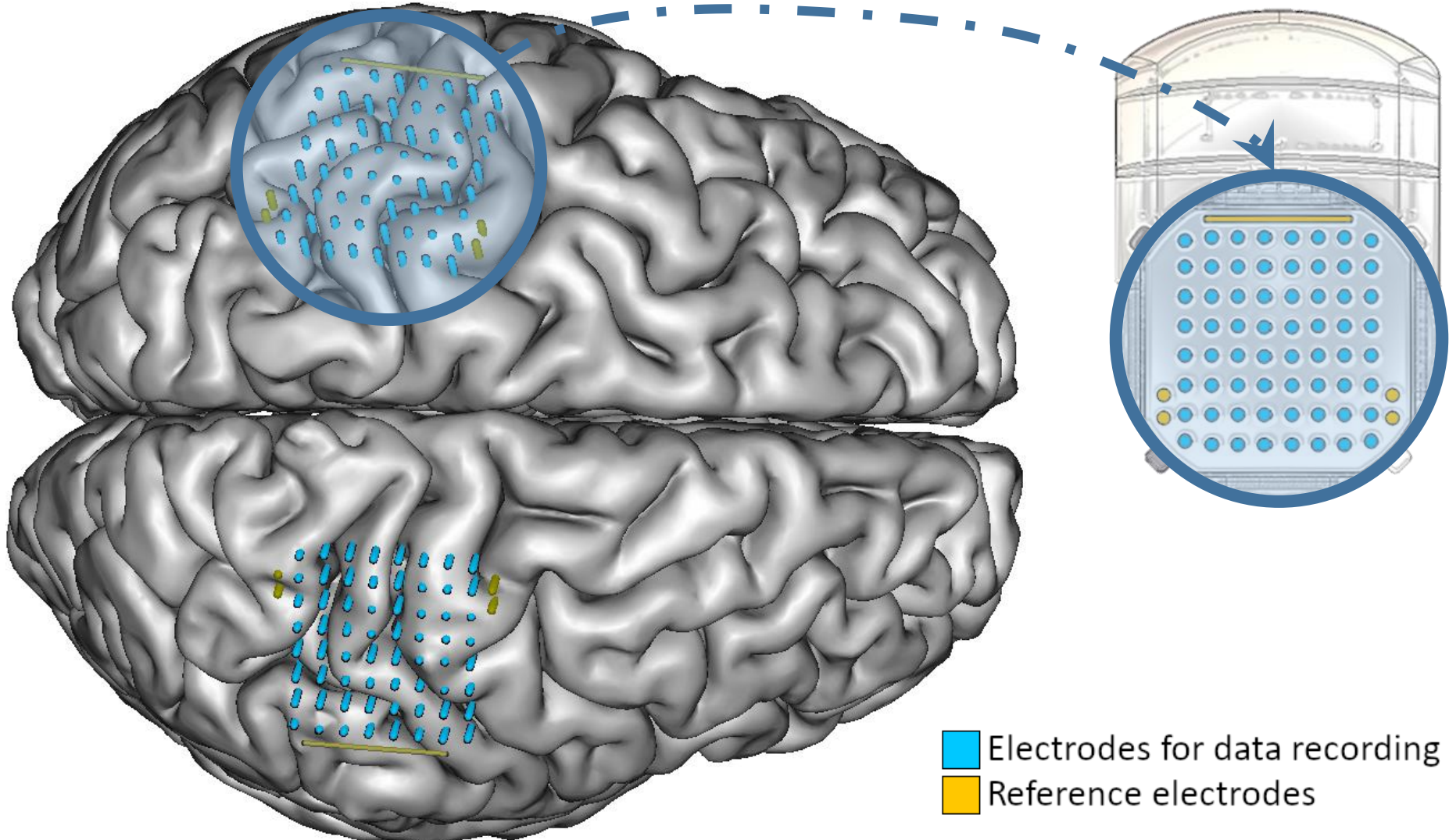



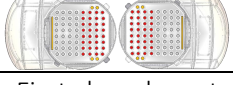
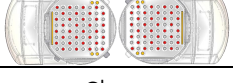
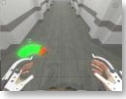
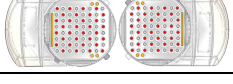




Figure 10 Reconstruction of the subject's brain from MRI data, with the estimated position of the electrodes of the two ECoG-recording implants. The yellow electrodes are the reference electrodes of the implants and the blue electrodes are the electrodes that can be used for data recording. Each implant features 64 recording electrodes (8x8 grid), but data can only be acquired from 32 electrodes at a time per implant due to currently limited data transfer rates. Data was acquired at 586Hz

## II. Datasets

The goal of the aaBCI is to be able to train or update control decoders during free use of the BCI. However, the first step in the development of such aaBCI was to test its feasibility and capabilities offline. The datasets used to perform these offline tests made use of control decoder that were not trained using the aaBCI framework. This means these control decoders were trained supervisedly while the subject was instructed to perform specific tasks. The datasets used were either previously recorded during the clinical trial or in newly designed experiments. These datasets contained data from BCI paradigms with various levels of complexity. Four datasets were acquired using a binary discrete effector. One dataset was acquired using a multi-class discrete effector. One dataset was acquired using a bi-dimensional continuous effector. In this section, we describe the experimental tasks performed in each of these datasets. Every dataset and their characteristics are listed in Table 1.

Table 1 List of datasets used

Experimental paradigm	Dataset name	Acquisition dates	Number of sessions	Average session length	Electrode set
	Runner no_MI	06.06.2019 - 28.02.2020 } 268 days	19	~11 minutes	Chess 
	Runner no_MI inverted	07.06.2019 - 26.06.2019 } 20 days	3	~12 minutes	Second chess layout 
	Runner MI central	22.06.2017 - 04.09.2017 } 90 days	18	~8.5 minutes	Central layout 
	Runner MI	06.09.2019 - 24.01.2020 } 141 days	13	~11 minutes	First chess layout 
	Exo	31.10.2019 - 24.01.2020 } 86 days	10	~20 minutes	Chess 
	Cursor	21.08.2017 - 06.10.2017 } 47 days	19	~16 minutes	Chess 



## II.1. Datasets using a binary discrete effector

In the first group of datasets used, the subject controlled a binary effector using a BCI. This BCI paradigm is from here on referred to as the Runner. The effector was a human avatar on a computer screen, represented from a third person perspective (Figure 11). This human avatar had two possible states: a walking state and an idle state. In the idle state, the avatar was standing still. In the walking state, the avatar walked forward at a fixed speed in an empty, infinite environment. In addition to the avatar, an instruction panel was also displayed on the computer screen. This instruction panel also had two possible states. In one state, the instruction panel was a blue circle with a white arrow pointing forward in it. In the other state, the instruction was a red hexagon with STOP written on it. To better explore the detectability of MTP in the sensorimotor cortex, two conditions were designed. The first one, called no\_MI, was designed to detect neural correlates of task performance without the neural noise induced by motor imagery. This ensured that the BCI-task-performance correlates found in the sensorimotor cortex were not motor imagery confounds. The second one, called the MI condition, was designed to be as close as possible as real case uses, and ensure that neural correlates of task performance can be detected from the sensorimotor cortex even when the user is controlling the BCI using motor imagery (which is noise for the purpose of task performance detection independently of the command sent).

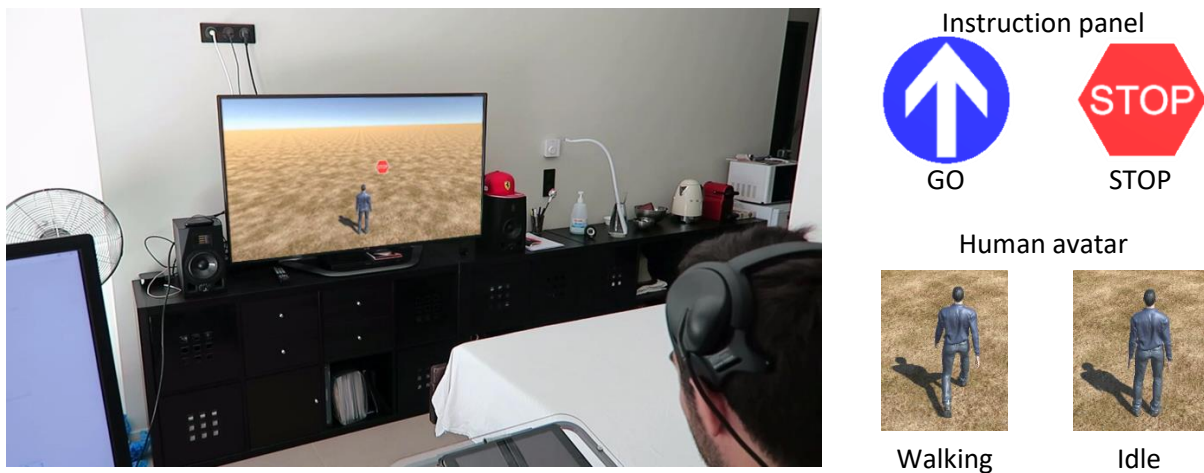


Figure 11 Example of the experimental setup for the binary discrete effector (Runner). A human avatar is displayed from a third person perspective in an empty environment on a computer screen. The avatar can take two exclusive states: either walking forward at a fixed speed or idle standing still. In addition to the human avatar, an instruction panel is also displayed. The instruction panel also has two possible states: either a blue circle with a white arrow within (GO instruction) or a red hexagon with STOP written within (STOP instruction).

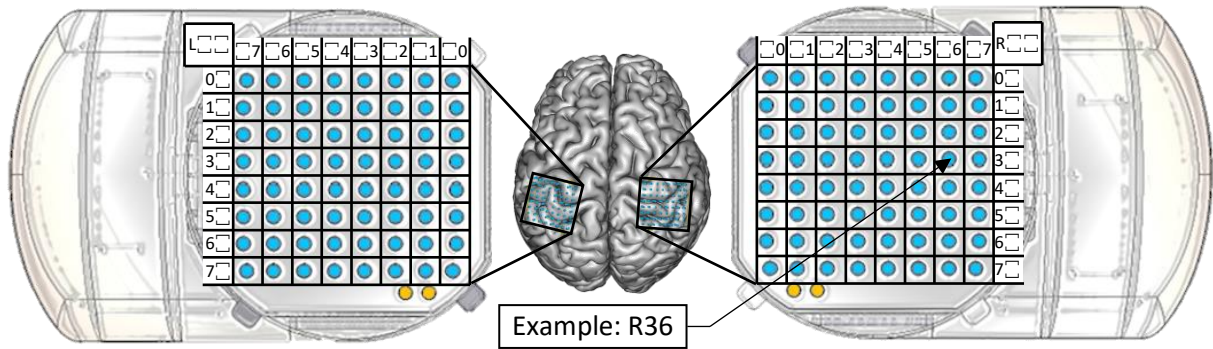


Figure 12 Electrode naming convention used in the thesis. The letter indicates which implant the electrode belong to, the first number indicates the vertical position of the electrode on the grid and the second number indicates the horizontal position of the electrode on the grid.

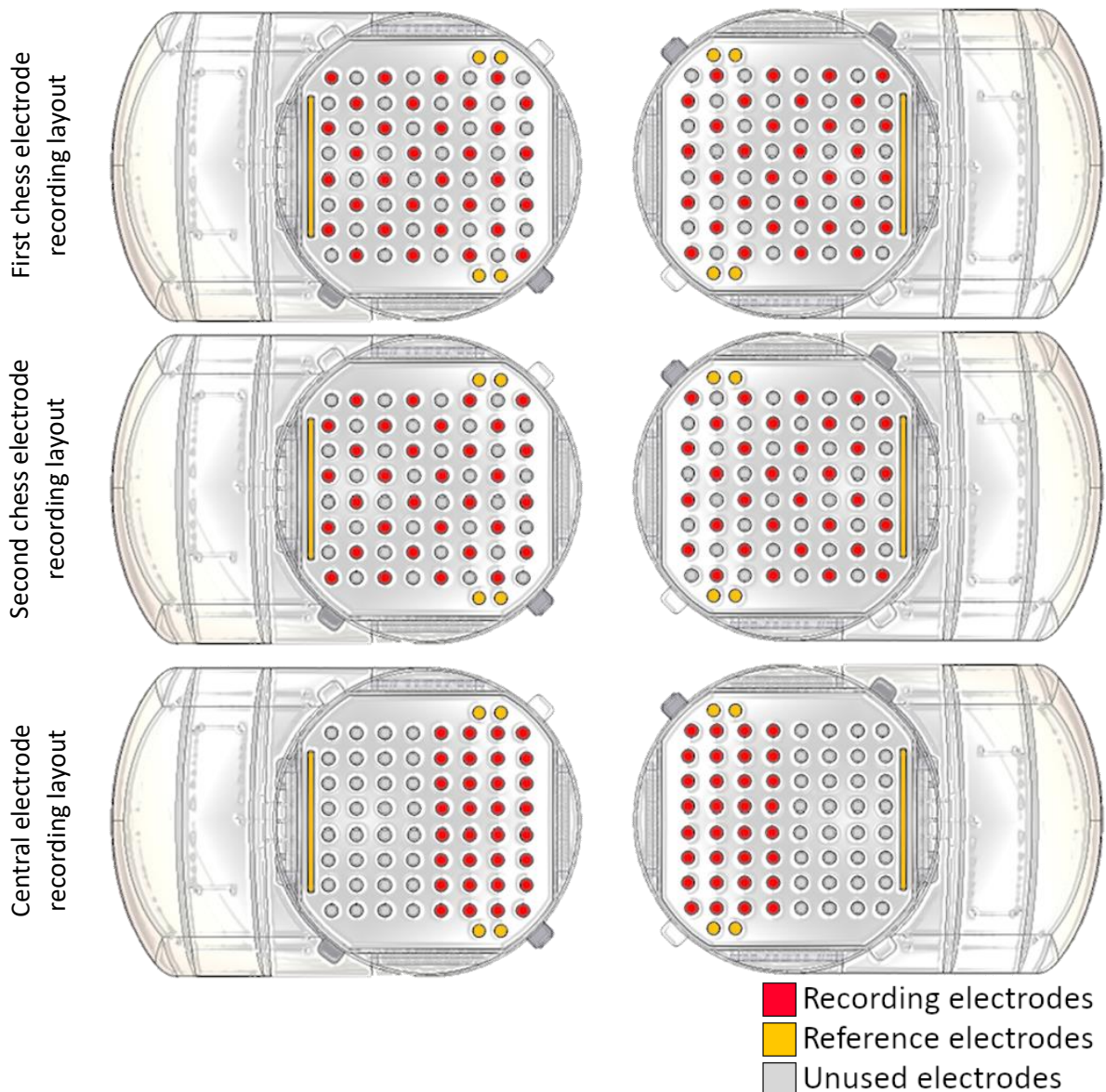


Figure 13 Schematic of the WIMAGINE implants with the different electrode groups used for data acquisition. The acquisition system is currently limited to receiving data from only 32 electrodes per implant (out of 64). The central electrodes recording layout was used in recordings performed to maximize the control of an effector controlled using motor imagery (Runner MI central). Electrode were grouped in chess patterns in order to maximize the recording surfaces when looking for correlates of task performance. The first chess electrode group was used in most experiments (Runner no\_MI dataset, Runner dataset, Exo dataset, Cursor dataset), whereas the second chess electrode group was only used for an electrode selection study (Runner no\_MI inverted dataset)

### *No\_MI condition*

In the no\_MI condition, the subject had no control on the movement of the avatar. The movements of the avatar were controlled by the computer. The subject was instructed to expect the avatar to behave as if he was controlling it while trying to follow the instructions displayed on the instruction panel. The subject was also instructed not to perform MI. The instruction panel state changed every five to fifteen seconds. The avatar state followed the instruction panel with a random delay between 0.2s and 0.5s to mimic human reaction time plus the latency of a BCI controlled binary effector. The avatar was also programmed to perform mistakes during its actuation. Error periods were introduced at random with the following constraints:

- Error periods had to occur in between instruction panel state changes.
- Error periods lasted a random time between two and three seconds.
- The error period rate was approximately two and a half error period per minute.

At first, two different electrode sets were used to record data in this condition. The two electrode sets were in chess patterns over all the possible recording electrodes (Figure 12 Figure 13). After a preliminary electrode selection study, data was acquired with only one of these electrode sets (see “*Supplementary data: 1. Electrode*”). With the first electrode setup, nineteen sessions of approximate duration of eleven minutes were performed over 268 days. This dataset is called the ‘Runner no\_MI’ dataset. With the second electrode setup, three recording sessions of approximate duration of twelve minutes were performed over 20 days. This dataset is called the ‘Runner no\_MI inverted’.

### *MI condition*

In the MI condition the subject controlled the avatar using leg motor imagery. The user was instructed to perform leg motor imagery when the instruction panel displayed GO and not to perform any motor imagery when the instruction panel displayed STOP. Two different datasets were recorded in the MI condition. The first one was recorded before the beginning of the thesis. These recordings were performed with motor control in mind, the subject using leg motor imagery to activate the walking state of the avatar. The goal was to maximize the control of the effector by the subject. The leg region of the sensorimotor cortex is more in the central area of the motor cortex. Considering the position of the implants, the electrodes used in these experiments were the central half of each electrode grid (Figure 13 and Figure 14). This dataset consisted of 18 recording sessions acquired over two and a half months. This dataset is referred to as the ‘Runner MI central’ dataset.

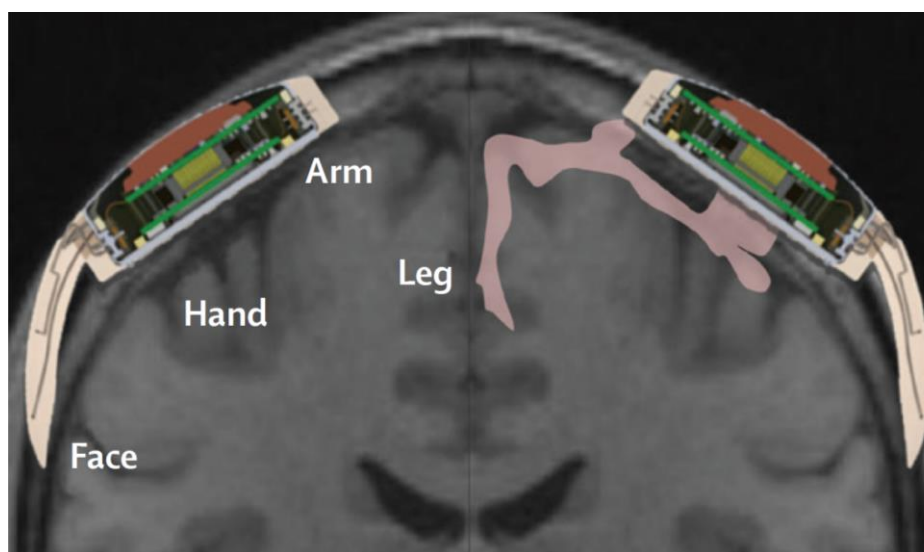


Figure 14. Position of the WIMAGINE implants relatively to motor areas. Figure taken from Benabid et al. (2019)



The second dataset was acquired roughly at the same time as the no\_MI dataset, with the recording electrodes arranged in a chess pattern to maximize the chance of detection of neural correlates of motor task performance. This second dataset is referred to as the Runner MI dataset, as it is the main dataset for this effector. At the time of collection of the second dataset, the subject was able to control complex BCIs (up to 10 degrees of freedom) and was proficient in the control of simple BCIs such as the Runner. In order to study the detection of task performance correlates, it is necessary to have both periods of good performance and periods of poor performance. In order to ensure that both performances levels occurred, the control decoder used to acquire data in this second Runner MI dataset was purposely trained using a very low amount of data. Apart from this purposely not optimal control decoder, there were no external influence on the control of the BCI by the subject. Thirteen sessions of approximate duration of eleven minutes were performed over 141 days.

## II.2. Dataset using a multi-class discrete effector

In one of the datasets used, the subject controlled a virtual exoskeleton using motor imagery. The exoskeleton had four possible movement states in addition to its idle state. The movement states were: movement of the right hand, movement of the left hand, rotation of the right wrist and rotation of the left wrist. The subject performed direct motor imagery to activate each of the motor states. This means the subject tried to move his right hand in order to move the right hand of the exoskeleton, to rotate his right wrist in order to rotate the right wrist of the exoskeleton and so on for the left hand and wrist. When one of the states was activated, the exoskeleton would perform actions

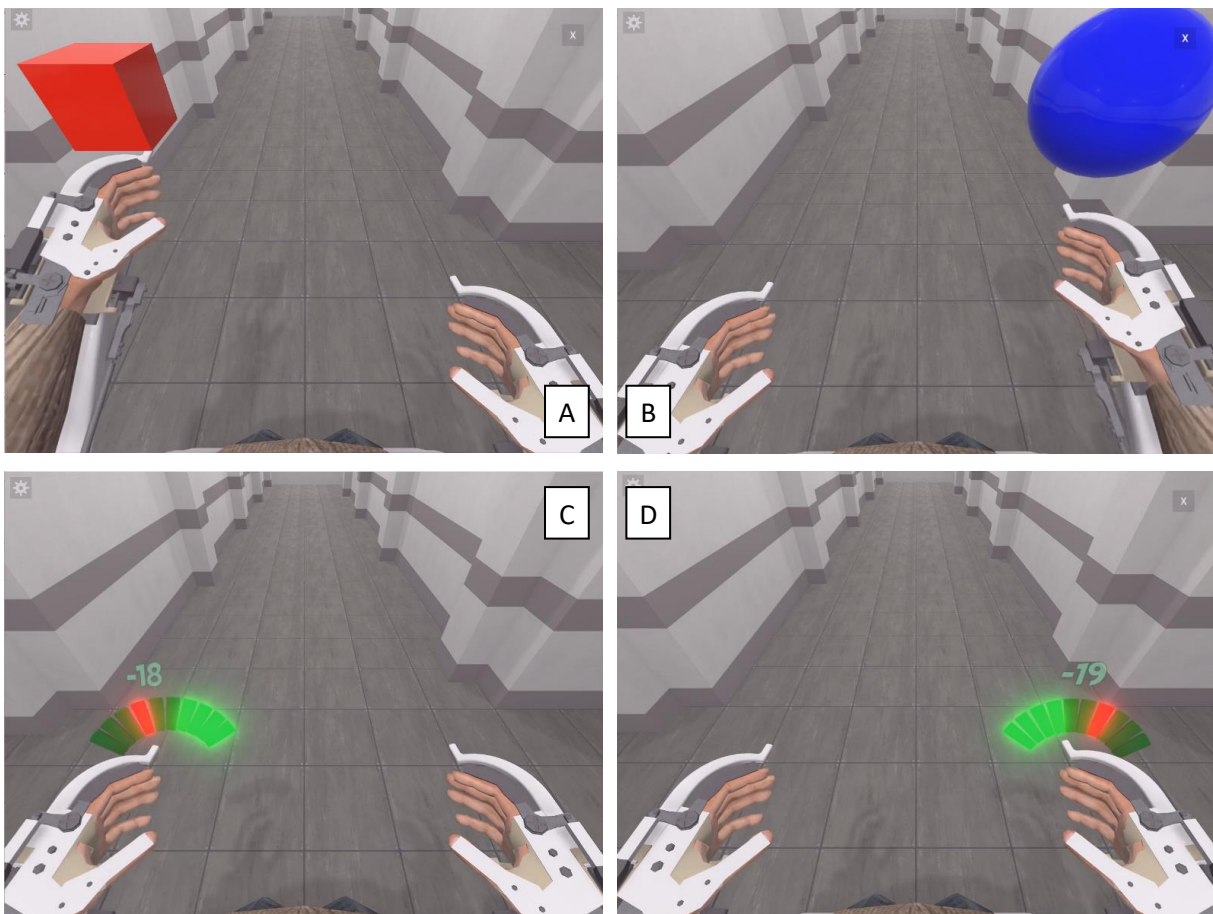


Figure 15 Images of the environment displayed on the computer screen during the recording of the Exo dataset. The subject controls an exoskeleton with four mutually exclusive movement states: movement of the left hand (A), movement of the right hand (B), rotation of the left wrist (C) and rotation of the right wrist (D). Instructions were displayed in the form of targets close to the hand to move (A&B) or angular deviations close to the wrist to move (C&D).

corresponding to the activated state, i.e. up or down movement of the hand and pronation or supination of the wrist. The subject was asked to focus on activating and maintaining each motor state without regards for the actions performed by the exoskeleton inside the active motor state. The virtual exoskeleton was displayed on a computer screen, with a human avatar being placed in it. The view was displayed from a first-person perspective of the human avatar. Instructions were displayed on the computer screen. Targets appeared in the virtual environment close to one arm when the instruction was to move the corresponding hand (Figure 15 A&B). Similarly, angles were displayed close to wrist that had to be activated (Figure 15 C&D). Ten sessions of approximate duration of twenty minutes were performed over 86 days. The data was acquired using the first chess electrode recording layout (Figure 13). We refer to this dataset as the Exo dataset.

### II.3. Dataset using a bi-dimensional continuous effector

In one of the datasets used, the subject controlled a hand-shaped cursor on a computer screen (Figure 16). The cursor could move in a square environment ( $25\text{cm} \times 25\text{cm}$ ) displayed at the center of the screen. The subject used motor imagery in order to control the direction and the speed of the cursor in two continuous dimensions. The goal given to the subject was to perform center-out tasks. This means that recording sessions were made of successive trials, and that at the beginning of each trial the position of the hand-shaped cursor was reset to the center of the environment and a new target was chosen. There were eight possible targets for this center-out task. Although the user controlled the speed and direction of the cursor, the speed was controlled up to a ceiling value of  $2\text{cm} \cdot \text{s}^{-1}$ . Nineteen sessions of average duration of 16 minutes were performed over 47 days. The data was acquired using the first chess electrode recording layout (Figure 13). We refer to this dataset as the Cursor dataset.

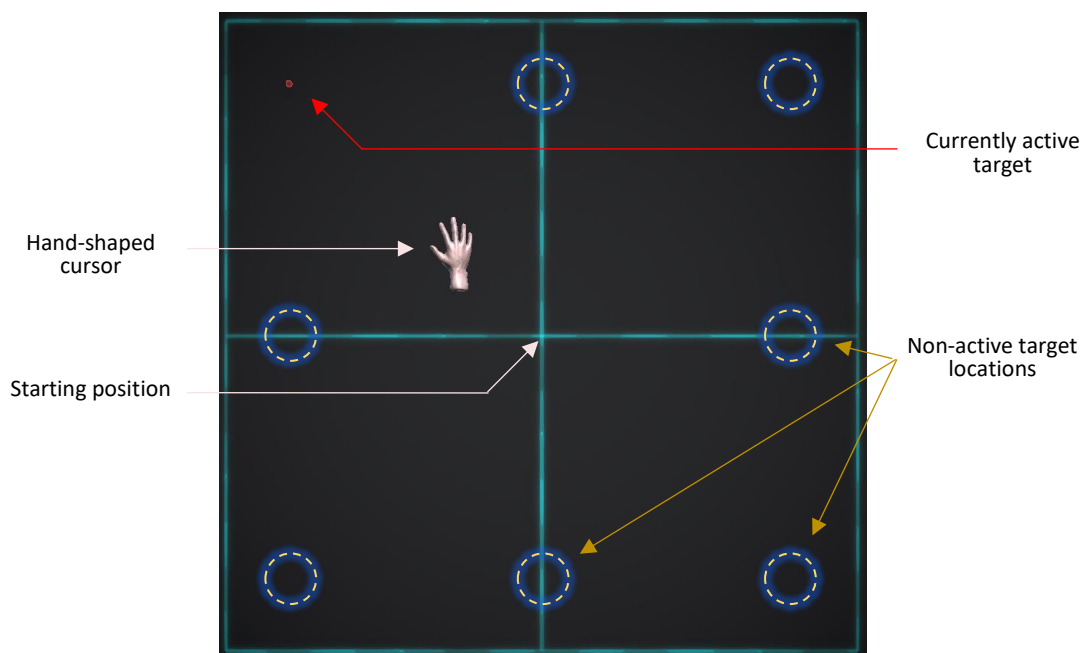


Figure 16 Environment used to record the Cursor dataset. The subject performed center-out tasks in a  $25\text{cm} \times 25\text{cm}$  square using a hand-shaped cursor controlled through motor imagery. There were eight possible target positions, with a new target being selected at the beginning of each trial.

## III. Decoder training for data acquisition

For each of the dataset where the effector was controlled by the subject (all datasets except the Runner no\_MI and Runner no\_MI inverted datasets), the control decoder was trained online using the Recursive Exponentially Weighted N-way Partial Least Squares algorithm (REW-NPLS) (Eliseyev et al.,

2017), similarly to other control decoders trained in the scope of this clinical trial (Benabid et al., 2019). These decoders were trained in a supervised manner using labeled data. Labels were available, as the subject had to follow the instructions that were given to him. As the effectors were controlled using motor imagery, the input features to the decoders were based on time-frequency decompositions of the ECoG signals. Time-frequency information was extracted using the continuous complex wavelet transform with fifteen Morlet wavelets with central frequencies ten hertz apart, from 10Hz to 150Hz. Data was acquired at 585Hz and at each time step  $t$ , the processing software received data by chunks of 59 points (approximately every 0.1s). The CCWT was applied to the last 177 points acquired (three time steps), and only the middle 59 points were kept to minimize edge effects. Epochs of one second were considered at each time step  $t$ , with the epoch acquired at time step  $t$  being referred to as epoch  $t$ . Successive epochs had a 90% overlap. For each epoch, electrode and frequency band, the modulus of the extracted time-frequency information was averaged in ten temporal points over windows of approximately 0.1s. The corresponding label for each epoch  $t$  is noted  $\mathbf{y}_{control}^t$ . For the datasets where the effector controlled had discrete outputs,  $\mathbf{y}_{control}^t$  was a vector containing a dummy encoding of the desired state of the BCI at the end time of the epoch. In the Cursor dataset, the output label  $\mathbf{y}_{control}^t$  for each epoch  $t$  contained the  $x$  and  $y$  directed Cartesian distance from the cursor to the target. The matrix of all real control decoder input labels is noted  $\mathbf{Y}_{control} = (\mathbf{y}_{control}^1, \dots, \mathbf{y}_{control}^N)^T$  with  $\mathbf{y}_{control}^t \in \mathbb{R}^m$  and  $m$  the number of possible classes (Runner and Exo) or the number of degrees of freedom (Cursor).



# Chapter 5: Detection of neural correlates of task performance

Erroneous events and correct events are known to produce different neural responses in specific areas of the cortex. The most studied of these neural responses is the Error Related Potential (ErrP) with its frontocentral and centroparietal components. Other event-locked neural correlates of task performance were studied to a lesser extent. Such neural correlates were recorded in various regions of the brain, including the sensorimotor cortex as described by Milekovic et al. (2013, 2012), Völker et al. (2018) or Wilson et al. (2019). All these neural responses to correct or erroneous events can be visualized through averages in the temporal or frequential domain, and were sometimes detected at the single trial level.

In this chapter, we investigate different methods to detect neural correlates of task performance. More specifically, to detect neural correlates of task performance in ECoG recordings of the sensorimotor cortex (due to the position of the implants). We aim to detect neural correlates of task performance regarding the actions performed by motor BCIs (as described in Chapter 4: Experimental design), which are most often controlled using motor imagery. We detect correlates of task performances in two manners. In an event-locked manner as is classically done in the literature for the Runner MI, Runner no\_MI and Runner no\_MI inverted datasets. And in a novel, continuous in time manner for the Runner MI, Runner no\_MI, Exo and Cursor datasets.

## I. Methods

### I.1. Neural feature extraction for single-trial detection of neural correlates of MTP

Time-frequency information is known to be suited to motor imagery tasks with most of the information being in the 0\*Hz-200Hz frequency band in the motor cortex (Waldert et al., 2009). Temporal information is known to be well suited for the detection of ErrPs over the frontocentral to centroparietal cortex (Chavarriaga et al., 2014). On the other hand, there is little known regarding neural feature extraction for single-trial detection of task performance correlates in the sensorimotor cortex due to comparatively low number of studies conducted. In the existing studies, time-frequency information was used to detect event-locked neural correlates of errors, with relatively high frequencies being the most relevant (Milekovic et al., 2013, 2012; Völker et al., 2018; Wilson et al., 2019). Although there is no widely accepted consensus yet, the best features for the single-trial detection of event-locked neural correlates of MTP from the sensorimotor cortex thus seem to be time-frequency decompositions. We therefore used time-frequency features as input to the eMTP decoders. Similarly, to our knowledge there is no state-of-the-art regarding the detection of continuous in time neural correlates of task performance. Since time-frequency features were successfully used to detect neural correlates of task performance in ECoG recordings of the sensorimotor cortex (although these were event-locked neural correlates), we also used time-frequency information for the detection of cMTP neural correlates.

Similarly to existing studies, we used continuous complex wavelet transform with Morlet wavelet to extract time-frequency information (Benabid et al., 2019). We used fifteen Morlet wavelets with



central frequencies ten hertz apart, from 10Hz to 150Hz. The feature extraction was made compatible with real-time predictions for the task performance decoders. At each time step  $t$ , the processing software received 59 new datapoints. The CCWT was applied to the last 177 points acquired (three time steps), and only the middle 59 points were kept to minimize edge effects. The logarithm of one plus the modulus of the CCWT was taken. Epochs of one second were considered at each time step  $t$ , with the epoch acquired at time step  $t$  being referred to as epoch  $t$ . Successive epochs had a 90% overlap. For each epoch, electrode and frequency band, the processed time-frequency information was averaged in ten temporal points over windows of approximately 0.1s. We note the resulting neural feature tensors as  $\underline{X}^t \in \mathbb{R}^{\tau \times f \times s}$ , with  $\tau = 10$ ,  $f = 15$  the number of analyzed frequencies and  $s = 64$  the number of recording electrodes. The data tensor of all samples is noted  $\underline{X} \in \mathbb{R}^{N \times \tau \times f \times s}$ , with  $N$  the total number of epochs for a given series of experiments.

## 1.2. Data labeling

The classical way to label data for the detection of neural correlates of task performance is to define events and label them as correct or erroneous. This ‘*event-locked*’ labeling is only possible if the actions performed by the BCI can easily be defined as events. For BCIs with discrete outputs, events occur each time the output of the BCI changes. As the possible outputs are discrete, a change in the output creates a distinguishable change and can therefore be called an event. For BCIs with continuous outputs, a change in the output of the BCI may not always create a variation strong enough to be called an event. As an example, let’s consider the case of a BCI with a one-dimensional output, such as a cursor on a line that is controlled in speed. Changes in the output of this BCI can be separated in two groups. Changes that reverse the direction of the cursor’s movement and changes that modify the speed of the cursor without reversing its direction. If the change in the output of the BCI reverses the direction of movement of the cursor, it can be considered as an event since the reversing of the direction of the cursor is easily distinguishable. If the change in the output of the BCI does not change the direction of the cursor but only changes its speed slightly, it is not possible to call this an event. This difficulty of labeling neural data for the detection of neural correlates of task performance is increased for BCIs with more than one continuous output. Let us consider the previous BCI, modified to control a cursor in two dimensions instead of one, with control over its speed and direction (in 2D). Neither change in the cursor’s speed or direction are guaranteed to produce an event if the variations are small. Therefore, event-locked labeling is not suitable for BCIs multi-dimensional continuous outputs. In the literature, there has been studies on the detection of neural correlates of task performance during continuous control of effectors (through BCIs or not). However, most of these studies had one-dimensional continuous effectors (Krelinger et al., 2012; Milekovic et al., 2013, 2012; Wilson et al., 2019). As explained, errors with one-dimensional effectors can be seen as discrete events as the controlled effector is either going directly toward the desired outcome or directly away. In the few studies where effectors with multiple degrees of freedom were controlled, either these degrees of freedom were not controlled continuously, or artificial strategies were set up in order to create erroneous events (Lopes Dias et al., 2018; Lopes-Dias et al., 2019; Omedes et al., 2015; Spüler and Niethammer, 2015). Therefore, all these studies can be assimilated to event-locked labeling.

In this section we first describe how we performed event-locked labeling similarly to existing literature. This labeling is performed for the detection of event-locked motor task performance, or eMTP. Then we describe how we labeled data in a novel, ‘*continuous in time*’ manner. The continuous in time labeling refers to how we label each data point instead of labeling events only. Since it does not require events for labeling, it is compatible with BCIs that have continuous outputs. This labeling is performed for the detection of continuous in time motor task performance, or cMTP.

*Event-locked labeling*

We performed event-locked labeling for the Runner MI, Runner no\_MI and Runner no\_MI inverted datasets. For each epoch  $t$ , the output variable noted  $y^t_{eMTP}$  could be equal to 0, to 1 or be undefined. It was equal to 1 when epochs were labeled as correct, to 0 when epochs were labeled as erroneous and undefined when epochs were unlabeled or rejected.

In our event-locked labeling, the first six full epochs after an event were labeled according to the event type. Epochs were one second long, therefore the first epoch labeled after an event was acquired one second after the event, and contained temporal data from the event onset to one second after the event. The last such epoch was acquired one and a half second after the event and contained temporal data from half a second after the event to one and half second after the event (Figure 17). Six epochs were labeled per events in order to mitigate two issues. The first one is the synchronization of events and their neural correlates. Indeed, the timing of brain responses to erroneous events is known to vary depending on various conditions such as workload, attention level or tiredness of the subject (Kumar et al., 2019). The second issue is due to effectors having inertia in their movement state. For instance, in the Runner the dataset, the avatar had some jitter in its reaction time to changes in the commands received from the BCI. Indeed, the avatar could instantly start moving when transitioning from the idle state to the walking state, but it had to finish its current stepping motion before being able to stop when transitioning from the walking state to the idle state. This jitter was estimated to be up to 0.3s. Adding six epoch for each event increased the probability of having the desired neural correlates in one of them, at the cost of some label uncertainty.

Additionally, we rejected some epochs that were too close to more than one event. Epochs acquired from an event onset to 0.9s after the event, as well as from 1.6s to 2.2s after the event were excluded from the analysis. This exclusion criterion was used to prevent epochs from containing neural correlates from more than one event. The number of events and the resulting number of epochs obtained with this labeling strategy in the Runner MI and Runner no\_MI datasets are presented in Tables 2,3 and 4.

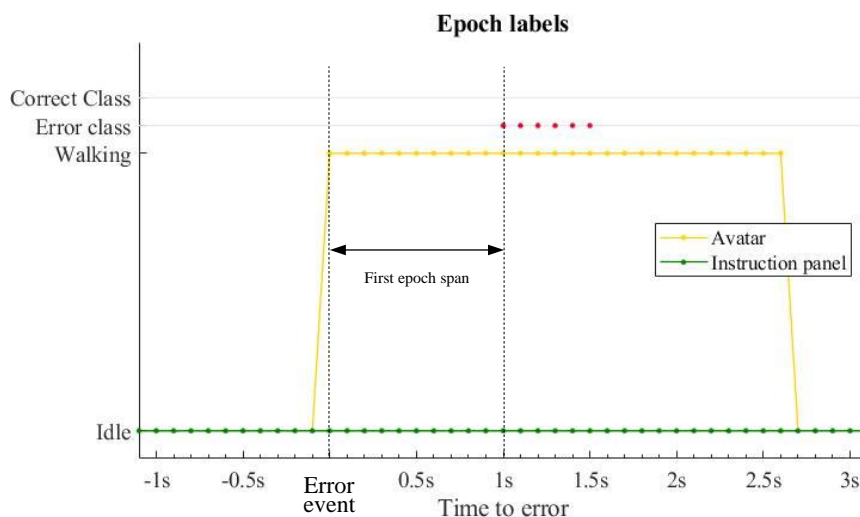


Figure 17 Example of event-locked epoch labeling in the Runner no\_MI dataset following an erroneous event where the avatar started walking when the instruction panel displayed the STOP sign.

Table 2 Repartition of error and correct events in the Runner MI and Runner no\_MI dataset.

Number of epochs	Epoch type	Runner no_MI	Runner MI
	Correct	7539	4412
Error	2307	3580	

Table 3 Repartition of error and correct events in each fold for the Runner no\_MI dataset.

Event type	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train error	283	308	316	317	332
Test correct	253	267	274	278	187
Test error	106	81	73	72	57

Table 2 Repartition of error and correct events in each fold for the Runner MI dataset.

Event type	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train error	519	457	464	469	483
Test correct	103	182	178	146	127
Test error	79	141	134	129	115

Continuous in time labeling

We performed continuous in time labeling for the Runner MI, Runner no\_MI, Exo and Cursor datasets. For each epoch  $t$ , the output variable noted  $y^{t}_{cMTP}$  could be equal to 0, to 1 or be undefined. It was equal to 1 when epochs were labeled as correct, to 0 when epochs were labeled as erroneous and undefined when epochs were unlabeled or rejected.

The detection of continuous in time neural correlates of task performance is not a common study. There is no prior knowledge on labeling for this purpose. We decided to label and use only epochs for which their correctness or erroneousness was certain. The labeling of the data was thus performed in a way that maximize the accuracy of the labels.

For datasets where the effector performed discrete tasks (Runner MI, Runner no\_MI and Exo), we took advantage of events in order to label data. Events were defined by changes in the state of the controlled effector. After a correct event, the effector of the BCI should be in the state required until the next event or next change of instruction. Similarly, after an erroneous event the effector of the BCI would stay in an incorrect state until the next event or change in instruction. In order to prevent neural data from one cMTP class to leak into the other, the fifteen first epochs following an event were discarded from the analysis. Epochs are one second long and are spaced by 0.1s, therefore the first ten

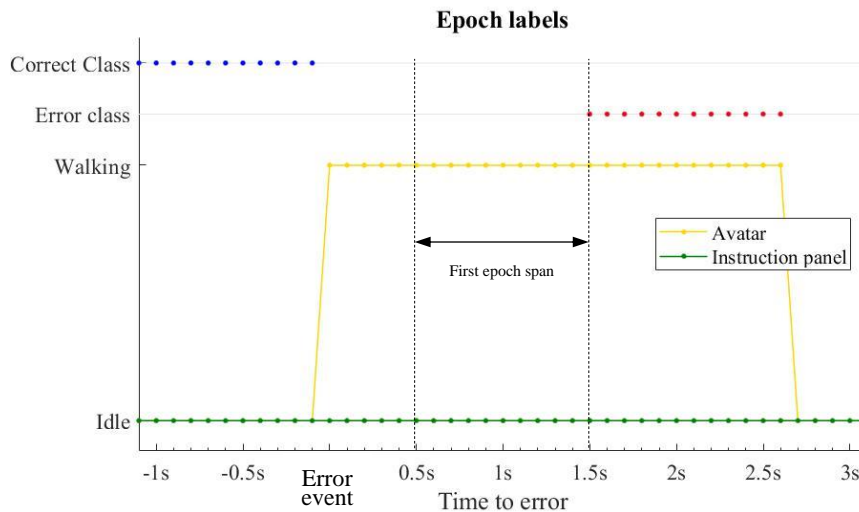


Figure 18 Example of continuous in time epoch labeling in the Runner no\_MI dataset following an erroneous event where the avatar started walking when the instruction panel displayed the STOP sign.

epochs after an event contain neural data from before the event. Additionally, there is both a reaction time needed for the user's brain to process the event, and some jitter in the timing of the actions of the effectors due to the necessity to finish its current action before being able to change state. We thus excluded epochs that contained neural data from up to 0.5s after the event. This led to the first fifteen epochs after each event not being labeled (Figure 18).

For datasets where the effector performs non-discrete tasks (Cursor), a different labeling strategy must be used since there are no events. Correctness must be estimated every time a new command was sent to the effector, i.e. for each epoch. We did it in the following way for the Cursor dataset, but without loss of generality regarding the number of continuous degrees of freedom of the effector. If we note as  $d^t \in \mathbb{R}$  the Euclidean distance between the hand-shaped cursor and the target for epoch  $t$  and  $d_{sat}$  as the saturated maximum displacement of the cursor at each time step, we define the  $correctness^t$  of the command performed at time  $t$  as:

$$correctness^t = \begin{cases} \frac{d^{t-1} - d^t}{d_{sat}}, & \text{if } d^{t-1} > d_{sat} \\ \text{sign}(d^{t-1} - d^t) \max\left(1, \frac{\text{abs}(d^{t-1} - d^t)}{d^{t-1}}\right), & \text{if } d^{t-1} < d_{sat} \end{cases}$$

$correctness^t$  is equal to 1 when the movement performed at time  $t$  was the best possible, i.e. directly toward the target at max speed when the target is further away than  $d_{sat}$  or on the target when the target is closer than  $d_{sat}$ .  $correctness^t$  is equal to 0 when the distance between the target and the cursor stays constant, and  $correctness^t$  goes down to  $-1$  as the movement performed brings the cursor away from the target. However, this evaluation of correctness is instantaneous and may therefore not reflect what the user perceives as correct or incorrect. We expect the neural data in a given epoch to reflect how good the control over the BCI was over a past window of time. We therefore use a smoothed-in-time correctness in order to label epochs. This smoothed-in-time correctness,  $\overline{correctness}^t$ , was obtained using a sliding window averaging with a uniform window of five epochs (500ms).

Finally, the label  $y^t_{cMTP}$  of epoch  $t$  for the cMTP decoder is derived as:

$$y^t_{cMTP} = \begin{cases} 1, & \text{if } \overline{correctness}^t > \theta_{corr} \\ 0, & \text{if } \overline{correctness}^t < \theta_{err} \end{cases}$$

Concretely, this means labeling was performed by thresholding the average correctness over the latest five epochs. We used  $\theta_{corr} = 0.5$  and  $\theta_{err} = -0.1$ . The relation between the deviation angle (the angle between the cursor trajectory and the direction toward the target) and the correctness value used here depends on the distance to the target and the size of the movement. However, a correctness of 0.5 or more can only be obtained if the angular deviation is less than  $60^\circ$ . Inversely, a deviation angle above  $90^\circ$  cannot provide a correctness value above 0. Unless the target is very close to the cursor, a correctness value of  $-0.1$  can only be obtained if the angular deviation is more than  $90^\circ$ . These angles ( $60^\circ$  and  $90^\circ$ ) matched the ones mentioned when discussing with the user what kind of cursor trajectory he would consciously evaluate as erroneous or correct. Finally, the first fifteen epochs after the end of a trial where not labeled as the target position had just been reset.

### 1.3. Data balance

In BCIs, confounds are a preeminent concern due to the very low signal to noise ratio. In the case of the auto-adaptive BCI, this signal to noise ratio may be even lowered. Indeed, the BCI is controlled using mental tasks that elicit specific neural correlates in the recording area. However, these neural correlates of the mental task performed are noise for the task of detecting neural correlates of task

performance. In our recording setup, the implants are positioned over the sensorimotor cortex. The sensorimotor cortex is known to produce strong neural correlates for the mental task used to control the BCI (motor imagery). As discussed extensively previously (Chapter 3:II.4 Detection of neural correlates of TP in the sensorimotor cortex), there are neural correlates of task performance in the sensorimotor cortex, but it is not widely regarded as the optimal area. Therefore, the noise created by motor imagery mental tasks may be strong compared to the neural correlates of task performance. Additionally, this noise can be correlated to the actual label of the MTP. If one of the mental tasks is much more often erroneous than correct (or the opposite), this can easily lead to a MTP decoder detecting neural correlates of the mental task instead of neural correlates of task performance.

One solution to solve this issue would have been to modify the output of the control decoder in real-time during the acquisition of the data used to train the MTP decoders in order to have as many erroneous and correct epochs for each mental task (or for each desired direction in the Cursor dataset). However, in most of the datasets we used, the user controlled the BCI without artificial interference on the output of the control decoder. This choice was done to have data as close as possible to out-of-the-lab conditions. Instead, this issue was solved post data acquisition for datasets with discrete outputs of the control decoder. It was done by artificially creating a balance between error and correct epochs for each mental task. The largest sub-class was downsampled to have the same number of epochs as the second largest sub-class, and the other sub-classes were oversampled to have the same number of epochs as the second largest sub-class. Oversampling was performed by repetition of the epochs. Oversampling and downsampling were only performed on the training data. For datasets with continuous outputs of the control decoder, there is no similar solution as there are no well-divided motor class to oversample or downsample. However, the distributions of the direction of desired movements in the correct and error class were quite similar, as can be seen in Figure 19. Therefore, no additional step was performed to minimize confounds.

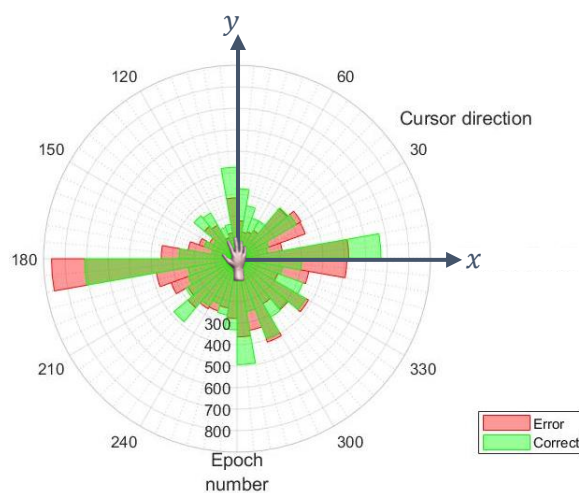


Figure 19 Distribution of direction of cursor displacement for the correct and error classes in the training dataset (for one fold).

#### 1.4. Decoders for the detection of neural correlates of task performance

The neural feature space has a high dimensionality. Indeed, the time-frequency features have ten temporal dimensions, fifteen frequential dimensions and sixty-four spatial dimensions. The input feature space has a total of 9600 features. Compared to other fields where machine learning is extensively used, BCI problems usually have a relatively low amount of labeled data and a low signal to noise ratio. This is also the case for our datasets. Due to the combination of high dimensionality, low signal to noise ratio and small size of the datasets, we had to use decoders with relatively low

complexity. We tried using a large panel of state-of-the-art decoders (Lotte et al., 2018). The decoders used were support vector machine (SVM), logistic regression, N-way partial least squares (NPLS), multilayer perceptron (MLP) and convolutional neural networks (CNN).

#### *Support Vector Machine & Logistic Regression*

Support Vector Machines (SVM) and logistic regression are considered as state-of-the-art methods for binary classification. Both methods are often used in combination with kernels, which can act as nonlinear projections of the input data into high dimensional spaces with the benefit of not having to specify the transformed input data. We tested commonly used kernels (Gaussian and polynomial) in preliminary test, but they tended to strongly overfit the training datasets, even with strong regularization parameters and low Gaussian kernel scale ( $<10^{-5}$ ) or low polynomial kernel order (order of 2 or 3). We therefore did not use kernels with SVM or logistic regression in the results presented in this study.

Since we had a high number of input features compared to the number of samples in our training datasets (more features than sample points), regularization was used for both SVM and logistic regression. For both methods, ridge regularization was applied. After preliminary tests, lambda was set to one.

#### *Multi-layer perceptron*

The multi-layer perceptron (MLP) is a fully connected feedforward artificial neural network. It can be interpreted as a logistic regression model that is preceded by a nonlinear transformation which increases predictive power of the model. The MLP model used in this study consisted of one hidden layer with 100 neurons (with learnable weights) followed by a ReLU activation. As all neurons are connected to each input component and produce linear combination of input features, there is an important number of parameters to optimize. Considering the size of the dataset and number of parameters, we regularized the model by applying batch normalization, dropout with probability of a neuron being zeroed 0.5 and L2 regularization on the model's weights with lambda equal 0.1 and early stopping on the validation set.

#### *Convolutional neural network*

Convolutional Neural Networks (CNNs) can take advantage of data structure. Due to their design, they are able to capture invariant patterns that occur in different parts of the signal. They have less trainable parameters than an equivalent MLP because of filters weight sharing, i.e. the same set of small filters is applied all over the data. We decided to use a CNN as there is a shift in error correlates synchronization inside epochs (Chapter 5:1.2 Data labeling, Event-locked labeling). By sliding convolutional filter over the signal in the time domain we expected the network to recognize error correlates (which we expect to be relatively time invariant) occurring at different moments in different epochs with the same filter. This resulted in a lower number of parameters and an expected higher performance in detecting time invariant patterns. The proposed CNN used 128 filters of shape  $5 \times 15 \times 64$  respectively in the time, frequency and spatial dimensions. Each filter was slid only over the time dimension with stride equal to one. We applied the same regularization methods and parameters as for the MLP.

#### *N-way partial least squares*

NPLS is a lesser-used algorithm in the BCI field although it is well suited to high dimensional, tensor-shaped data (Bro, 1996). This algorithm projects the tensor decomposition of the input and output variables into a latent space where their covariance is maximized. The NPLS has the advantage of being robust to irrelevant or correlated input variables. It also has relatively low computational cost. Finally, the final model produced can be used to interpret the relative importance of each input feature for



the prediction of the output variables. The NPLS algorithm has a hyperparameter in the dimension of the latent space on which the input and output variables are projected. Test studies showed that twenty factors provided good performances or was even sometimes more than necessary for optimal performances. We did not perform optimization of this hyperparameter, but instead provide performance results for factor numbers between one and twenty.

All these decoders were tested when the labeling was performed in an event-locked manner. Since there was no significant difference between the decoders (see below, Chapter 5:II.2.Detection of event-locked neural correlates of motor task performance) for the detection of eMTP, we only used the NPLS for the detection of cMTP. The NPLS was chosen due to its high robustness to noise in the labels, which is a strong possibility when using the cMTP decoder for the aBCI (see Chapter 6). The NPLS algorithm models a multi-linear relationship between the tensor input and the tensor output. In our case, we note as  $\underline{\beta} \in \mathbb{R}^{\tau \times f \times s \times m}$  the tensor of parameters of the decoder,  $\beta_0 \in \mathbb{R}^m$  the matrix of biases of the decoder,  $\underline{X}^t \in \mathbb{R}^{\tau \times f \times s}$  the input tensor for epoch  $t$  and  $\mathbf{y}^t \in \mathbb{R}^m$  the output vector for epoch  $t$ . We have  $\mathbf{y}^t = \beta_{(4)}^t \mathbf{x}_{(4)}^t + \beta_0$ , with  $\beta_{(4)}^t \in \mathbb{R}^{m \times \tau f s}$  the notation for the tensor  $\underline{\beta}$  unfolded along the fourth dimension, and  $\mathbf{x}_{(4)}^t \in \mathbb{R}^{\tau f s}$  the notation for the tensor  $\underline{X}^t$  unfolded along the fourth dimension. The parameters of the decoder can be used to study the relative importance of the input features. In order to do so, we use normalized versions of the decoders' parameters. To simplify visualization, we study the decoders' parameters for each modality, by averaging the absolute value of the parameter tensor over the other modalities:

$$\beta_{temporal} = \left( \sum_{j=1..f} \sum_{k=1..s} \frac{|\beta_{i,j,k}|}{f s} \right)_{i=1..m}$$

$$\beta_{frequentia} = \left( \sum_{i=1..m} \sum_{k=1..s} \frac{|\beta_{i,j,k}|}{\tau s} \right)_{j=1..f}$$

$$\beta_{spatial} = \left( \sum_{i=1..m} \sum_{j=1..f} \frac{|\beta_{i,j,k}|}{\tau f} \right)_{k=1..s}$$

### 1.5. Performance measure for single trial detection of neural correlates of MTP

When labeling was performed in an event-locked manner, the goal was the detection of correct and erroneous events. As described in the data labeling section (Chapter 5:1.2 Data labeling), up to six epochs were considered for each event. An event was considered erroneous as soon as one of its associated epochs was classified as an error. When the labeling was performed in a continuous in time manner, the goal was the detection of correct and erroneous epochs directly.

Performances of the MTP decoders were evaluated in five-fold cross-validations. Two types of cross-validation were used: a 'full-session' five-fold cross-validation and a 'split-session' five-fold cross validation. In full-session cross-validations, each split of the data contained the same number of full sessions (up to a one-session difference). In split-session cross-validations, each session was separated in five consecutive in time chunks. For a given session, each of these chunks contained approximately the same number of labeled epochs. The data from the first chunk of each session was put together to form one split, and the same was done for the other four chunks of each session. The full-session cross-validation scheme provides information on how well MTP decoders perform when they are used on

different recording sessions than the one trained on. On the other hand, the split-session cross-validation provides information on how well MTP decoders perform when trained in optimal conditions, that is to say trained and used in the same session.

For both cross-validation schemes, each split was used as a test split once, with the other four split being used to train the MTP decoder. This provided five MTP decoders for each cross-validation performed. The performance criterion we used to measure the performances of MTP decoders was the area under the curve (AUC) of the receiver operating characteristic curve (ROC curve). The ROC curve plots the true positive rate versus the false positive rate for a binary classification based on a continuous variable (e.g., the output of the decoder). Each point of the curve corresponds to one possible threshold of the decision variable, and has an associated true positive rate and false positive rate. The AUC of the ROC curve is commonly used as an indicator of performance for binary classification (Hastie et al., 2009), even in the field of BCIs. We note the AUC of the ROC curve as AUC for simplification purpose, as no other area under curves are considered in our work. The final performance of the MTP decoders was given by the mean and standard deviation of the AUC over the test folds.

### 1.6. Population response to erroneous and correct events

We tried to visualize the neural response to both type of events, as well as their differences. These visualizations were performed in the temporal and time-frequency domains, using the Runner MI central dataset.

We defined events as changes in the state of the avatar. Either these events were correct (after the event the avatar state was the one demanded) or erroneous (after the event the avatar state was not the one demanded). Changes in the instruction panel state were not considered as events as their labeling and timing would be uncertain. Indeed, the user may not consider an instruction change as erroneous or correct per se. Instruction changes which result in the avatar and instruction panel states being similar could potentially be labeled as correct events, event though that would be debatable. Instruction changes that result in different avatar and instruction panel states would not be immediately considered as erroneous or correct by the user. The user would consider there is an error only if he did not manage to change the avatar state to the requested one after a certain period. The duration of this period is unknown and likely to vary between events. Thus, changes in the state of the instruction panel were not considered as events. Epochs of 2.6 seconds were considered in this visualization study. Epochs started 600ms before an event and ended 1500ms after. Only epochs that had no other event or instruction between 1.4s before and 1s after their associated event were used for this visualization study. There were 282 correct and 146 error epochs in this preliminary visualization study. For the visualization in the temporal domain, the data from each recording session was filtered before epoch extraction with a 4<sup>th</sup> order IIR bandpass filter with cut-off frequencies of 0.2Hz and 7Hz. Each epoch was baselined by removing the average filtered signal over the 400ms preceding its event. For the visualization in the time-frequency domain, time-frequency information was extracted from the raw ECoG epochs with the short-time Fourier transform, using 100ms Hanning windows with 80% overlap. The Fourier coefficient were computed for frequencies from 1Hz to 200Hz with a step size of 0.5Hz. The log of these coefficients was taken. They were then normalized per frequency using their median and median absolute deviation over the 400ms preceding the event.



## II. Results

### II.1. Visualization of population response to correct and erroneous events

The visualization of the median waveform after correct or erroneous events can be seen in Figure 21. The visualization in the time-frequency domain can be seen in Figure 22 for the difference between erroneous and correct events, and in Figure 23 for the patterns after erroneous events.

In the temporal domain, the strongest differences between correct and error epochs were seen in the fronto-central part of the left implant (L01, L10, L11, L00). On the right implant, differences were smaller than on the left implant, but less constrained to a single area. In the time-frequency domain, differences were mainly in the frequencies lower than 50Hz. The spatial pattern of the differences in the time-frequency domain was close to the one in the temporal domain. Although there were some differences between the correct and error epochs in the temporal domain and the time-frequency domain, these differences were relatively small when compared to the corresponding deviation measures (Figure 20).

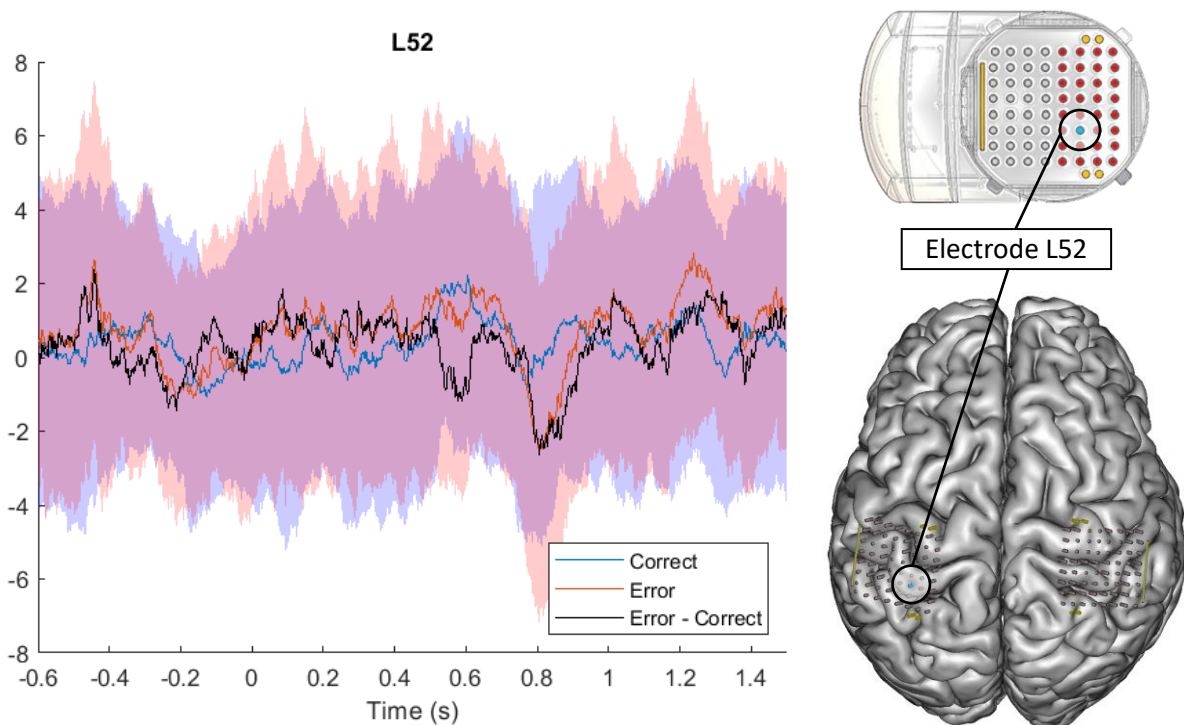
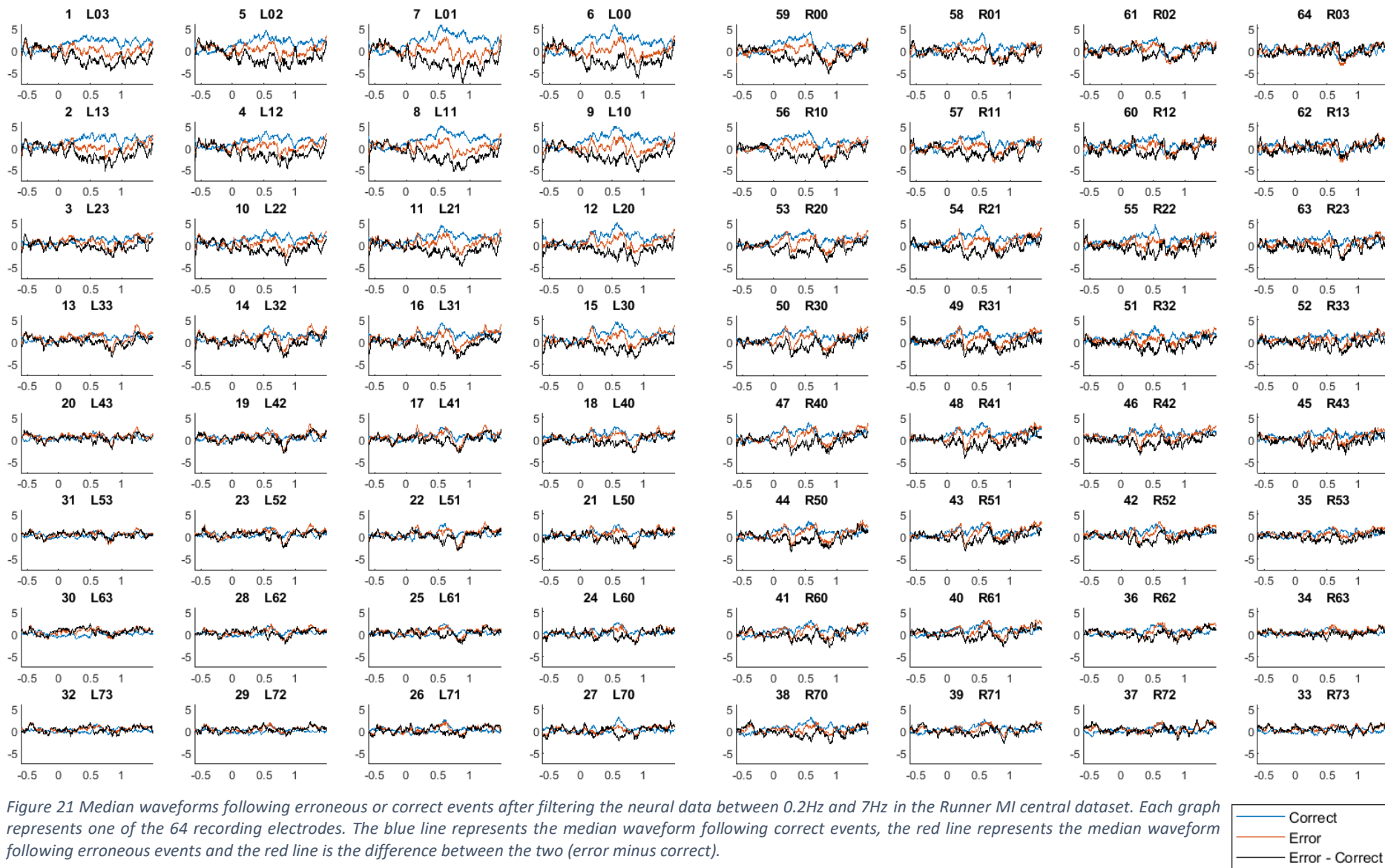


Figure 20 Filtered median waveform after correct (blue line) or erroneous (red line) events in a motor BCI with control over two discrete states recorded from ECoG electrode L52 over the sensorimotor cortex. The black line represents the difference between the correct and erroneous waveforms, and the blue and red patches represent the median absolute deviations of the correct and erroneous waveforms respectively. Electrode 52 is one of the electrodes with the most disjointed blue and red patches.



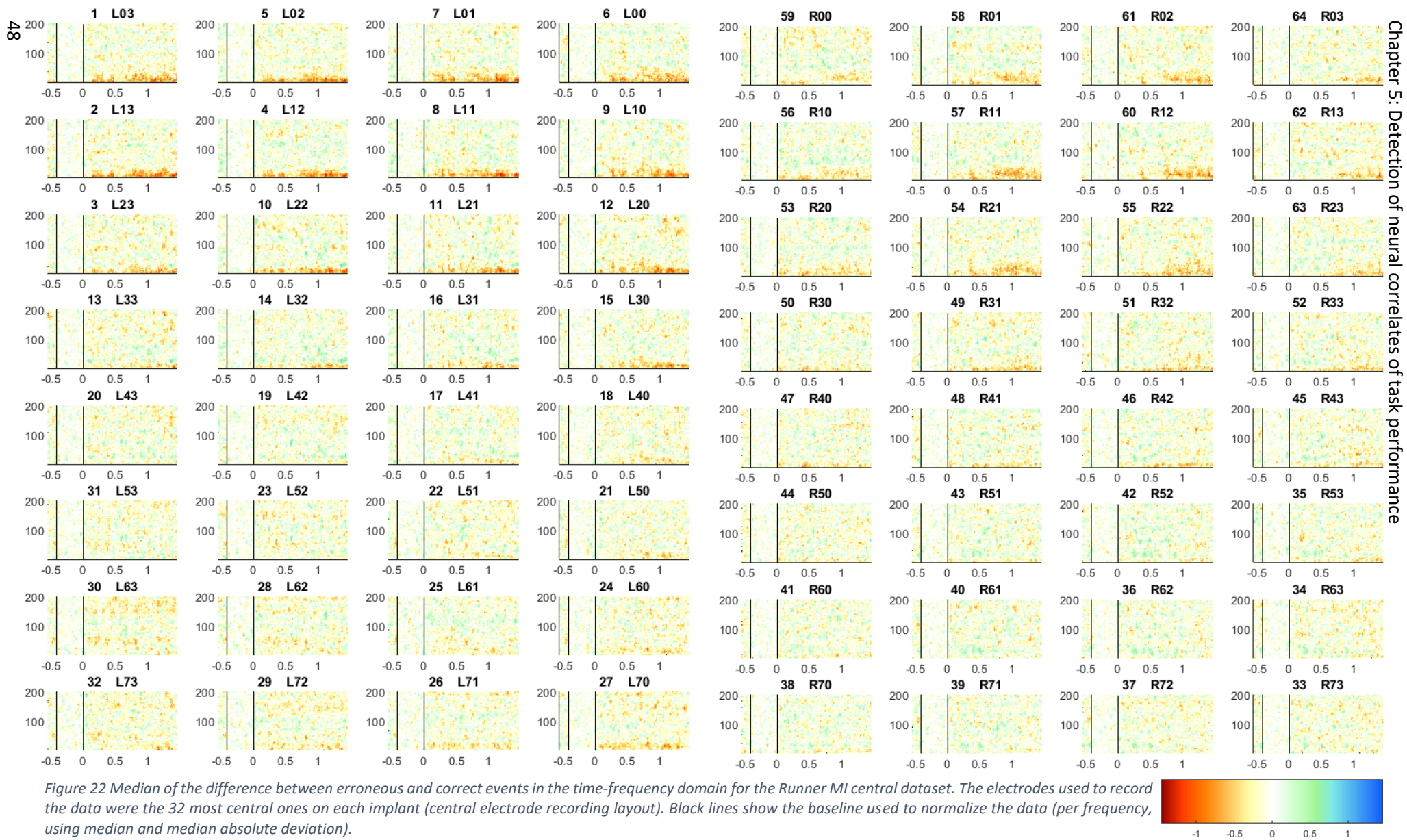


Figure 22 Median of the difference between erroneous and correct events in the time-frequency domain for the Runner MI central dataset. The electrodes used to record the data were the 32 most central ones on each implant (central electrode recording layout). Black lines show the baseline used to normalize the data (per frequency, using median and median absolute deviation).



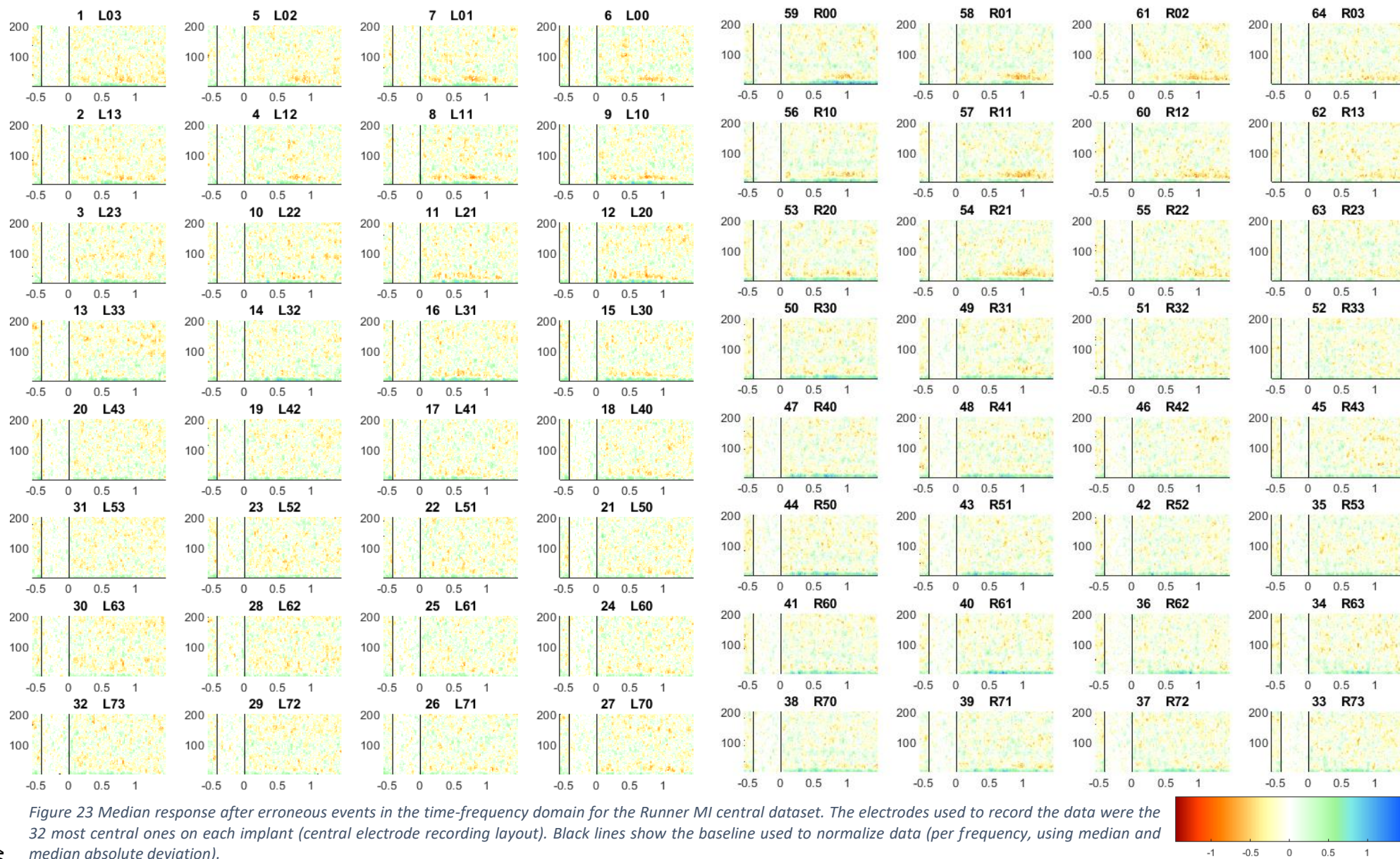


Figure 23 Median response after erroneous events in the time-frequency domain for the Runner MI central dataset. The electrodes used to record the data were the 32 most central ones on each implant (central electrode recording layout). Black lines show the baseline used to normalize data (per frequency, using median and median absolute deviation).

## II.2. Detection of event-locked neural correlates of motor task performance

We performed detection of event-locked neural correlates of MTP on the Runner MI and Runner no\_MI datasets using full-session cross-validation. The results for each decoder (SVM, logistic regression, MLP, CNN and NPLS) are summarized in Table 3 and Figure 24. A Friedman test was performed to compare model performances within each dataset group. No significant differences were found between models in the Runner no\_MI dataset (Friedman test,  $p$ -value = 0.13) or between models in the Runner MI dataset (Friedman test,  $p$ -value = 0.08).

Table 3 Mean and standard deviation over five test folds of the area under the curve of the receiver operating characteristic curve for the classification of error vs correct events.

Runner no_MI	NPLS	Logistic	SVM	MLP	CNN
AUC mean	0.653	0.662	0.645	<b>0.680</b>	0.630
AUC std	0.096	0.106	0.119	0.131	0.124
Runner MI	NPLS	Logistic	SVM	MLP	CNN
AUC mean	0.601	0.605	0.623	<b>0.626</b>	0.580
AUC std	0.037	0.040	0.027	0.014	0.022

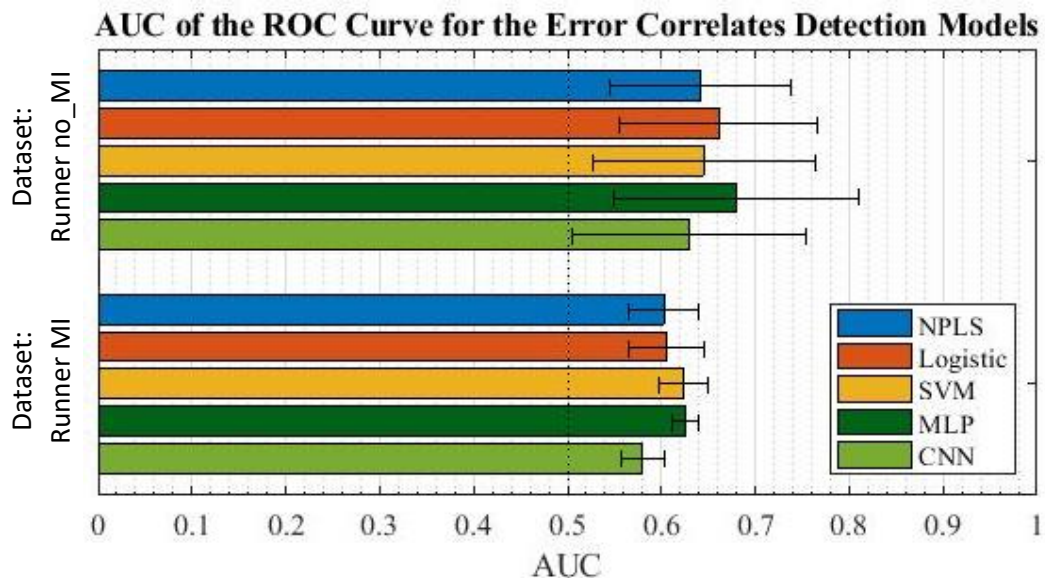


Figure 24 Mean area under the curve of the receiver operating characteristic curve on the cross-validation test folds, of each decoder tested, for the detection of event-locked neural correlates of MTP. Error bars on the left and right of the mean each represent one time the standard deviation on the test folds.

The results of this study show that the MLP decoder achieved the best performance in both the Runner no\_MI and MI datasets. CNN and MLP were the two models that allowed for the most complex representations, such as nonlinear relationships. Taking into account that regularization would limit the drawbacks associated to their high number of parameters, we expected these models to perform the best out of all the models tested. CNN had less parameters than MLP and was also more adapted to the task of re-synchronizing error correlates. However, the performances of CNN models were unexpectedly the worst across all decoders. A possible explanation for this is that both neural network architectures (and neural networks in general) had a high number of hyperparameter spaces (e.g., learning rate, number of filters, regularization weight).

In both the Runner MI and Runner no\_MI datasets, the three other decoders performed similarly, with small variabilities demonstrating better performances for logistic regression in the first dataset and for SVM in the second dataset. NPLS always performed slightly worse than SVM and logistic regression and up to 3.7% lower than MLP. However, NPLS has the technical advantage of being the method used in the context of the clinical trial. It is also trainable online. Although the MTP decoder does not have to be created in an online manner, it enables to have a model ready directly after the data was recorded instead of training more complex decoders in an offline manner. For these reasons, we used NPLS for the training of MTP decoders in the remainder of this thesis.

Performance across different folds was represented by the standard deviation of the AUC (ROC curves per fold can be seen in Figure 25). Since cross-validation was performed with the full-session scheme, this standard deviation can be used to predict the generalization capabilities of each model over different datasets. In the Runner no\_MI dataset, the standard deviations of the AUC for each decoder were close to one another, with NPLS having the lowest. In the MI dataset MLP had the lowest standard deviation, close to twice lower than the standard deviation of other decoders.

On average, the AUC of the decoders were 6.8% lower in the Runner MI dataset than in the Runner no\_MI dataset. This was expected since the motor imagery signals used to control the BCI in the MI dataset can be regarded as noise for the classification of error and correct events. However, the standard deviation of the AUC was up to ten times larger in the no\_MI dataset than in the MI dataset. We suggest that the higher variability in the no\_MI dataset was due to a higher variability in the attention level of the subject than in the MI dataset. Indeed, in the Runner MI dataset the subject was more engaged in the task since he had active control over the avatar's actions. In the Runner no\_MI dataset, the subject was more vulnerable to distractions due to the lack of interaction required by the experimental task. We hypothesize that the attention level modulated the strength of the error correlates in the motor cortex, similarly to how it modulates classical ErrPs (Yeung, Holroyd, & Cohen, 2005).

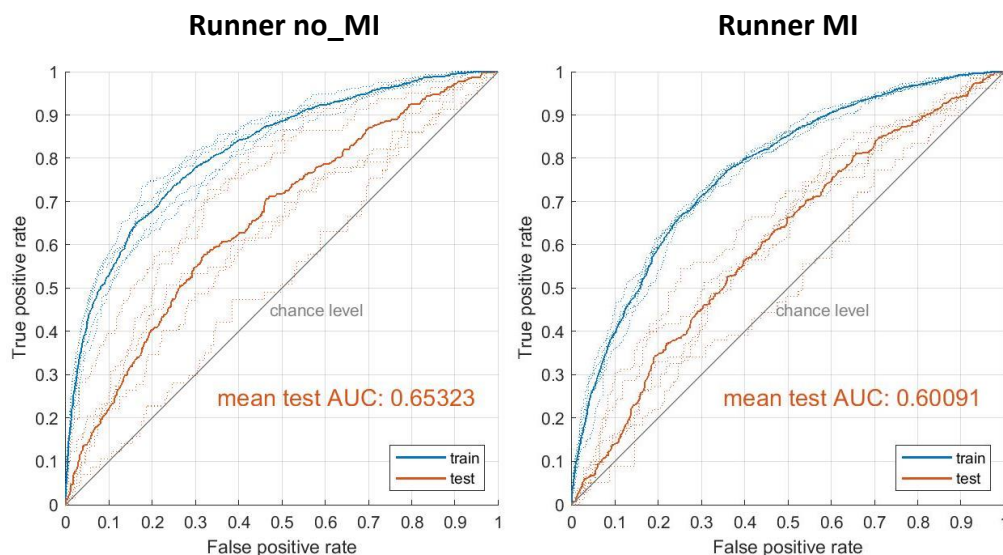


Figure 25 Receiver operating characteristic (ROC) curve for the single trial detection of neural correlates of correct or erroneous events (eMTP). Each curve represents the ROC on one of the test folds of the five-fold full-session cross validation.



### II.3. Detection of continuous in time neural correlates of motor task performance

**Runner Dataset.** In the full-session cross-validation, the mean and standard deviation of the AUC of the ROC over the test splits were  $0.6780 \pm 0.1165$  in the no\_MI dataset and  $0.6225 \pm 0.0429$  in the MI dataset. In the split-session cross-validation, the mean and standard deviation of the AUC of the ROC were  $0.7198 \pm 0.0287$  in the no\_MI dataset and  $0.6554 \pm 0.0228$  in the MI dataset.

**Exo dataset.** In the full-session cross-validation, the mean and standard deviation of the AUC of the ROC over the test splits were  $0.5677 \pm 0.0427$ . In the split-session cross-validation, the mean and standard deviation of the AUC of the ROC were  $0.5782 \pm 0.0252$ .

**Cursor dataset.** In the full-session cross-validation, the mean and standard deviation of the AUC of the ROC over the test splits were  $0.6570 \pm 0.0188$ . In the split-session cross-validation, the mean and standard deviation of the AUC of the ROC were  $0.6838 \pm 0.0133$  Figure 26.

These results are summarized in Table 4 for readability. The average ROC curves as well as the ROC curves for each fold are shown in Figure 26.

*Table 4 Performances of the cMTP decoder on the Runner no\_MI, Runner MI, Exo and Cursor datasets, depending on the cross-validation type. Performance is reported as the mean AUC of the ROC curve over the test folds. Standard deviations are reported in brackets.*

	Runner no_MI	Runner MI	Exo	Cursor
Full-session	$0.6780 (\pm 0.1165)$	$0.6225 (\pm 0.0429)$	$0.5677 (\pm 0.0427)$	$0.6570 (\pm 0.0188)$
Split-session	$0.7198 (\pm 0.0287)$	$0.6554 (\pm 0.0228)$	$0.5782 (\pm 0.0252)$	$0.6838 (\pm 0.0133)$

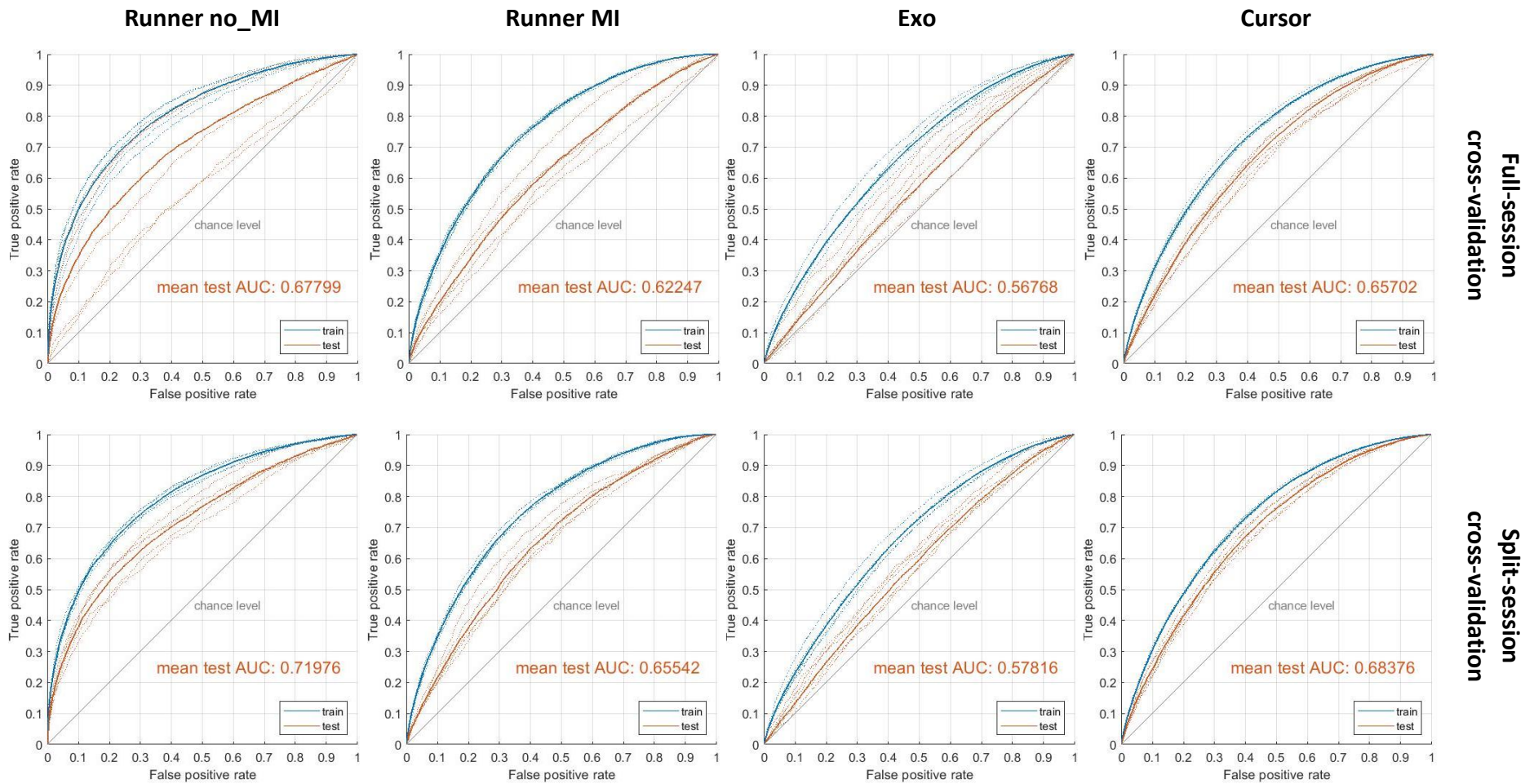


Figure 26 Receiver operating characteristic curves for the detection of epochs labeled as 'errors' for the cMTP decoder for each experimental paradigm and both types of cross-validation. Performances on the training sets are displayed in blue, while performances on the testing set are displayed in red. Each dotted line represents one of the five cross-validation folds, while the full lines represent the interpolated mean ROC over the five folds.



## II.4. Impact of NPLS factor numbers on the performance of detection of neural correlates of cMTP

The NPLS algorithm has a hyperparameter in the number of factors used. As mentioned previously, we did not optimize this hyperparameter through nested cross-validation due to dataset sizes. However, during training the NPLS algorithm computes the decoders for each factor number under the final one. We used 20 factors for the NPLS computation and therefore have access to the decoders with 1 to 20 factors. We show in Figure 27 (Runner MI & Runner no\_MI), Figure 28 (Exo) and Figure 29 (Cursor) the evolution of the AUC of the MTP decoders for each experimental paradigm and both cross-validation schemes. For each decoder, the performances on the test data are at their lowest for low numbers of factors (1-7) after which they increase and then reach a plateau at higher numbers of factors (10-20). The number of factors needed to reach the plateau varies between datasets, with the Cursor dataset needing the least (~4 factors), the Exo dataset needing the most, and the two Runner datasets in between.

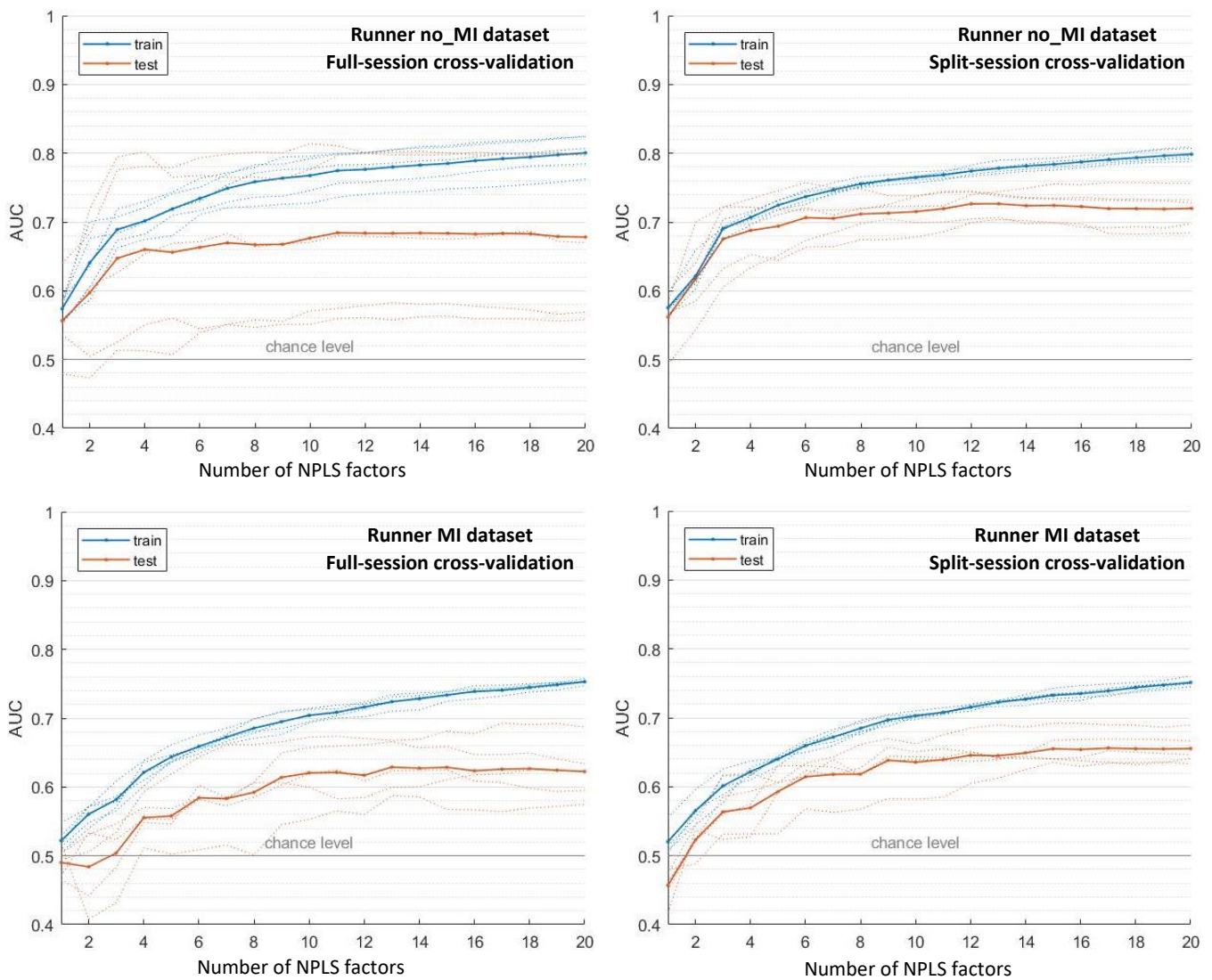


Figure 27 Performance measure (mean AUC of the ROC) for the detection of cMTP neural correlates in the Runner task (no\_MI and MI datasets), depending on the number of factors kept in the NPLS algorithm and on the type of cross-validation used. Performances on the training sets are displayed in blue, while performances on the testing set are displayed in red. Each dotted line represents one of the five cross-validation folds, while the full lines represent the mean AUC over the five folds.

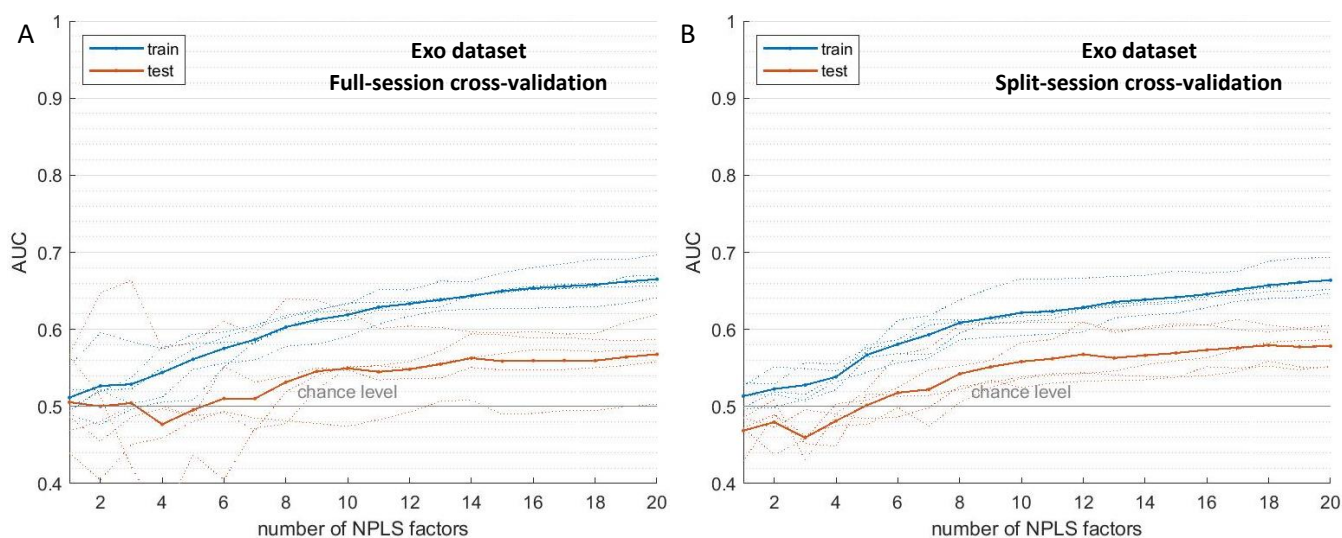


Figure 28 Performance measure (mean AUC of the ROC) for the detection of cMTP neural correlates in the multi-class discrete task, depending on the number of factors kept in the NPLS algorithm and on the type of cross-validation used. Performances on the training sets are displayed in blue, while performances on the testing set are displayed in red. Each dotted line represents one of the five cross-validation folds, while the full lines represent the mean AUC over the five folds. (A) Full-session cross-validation. (B) Split-session cross-validation.

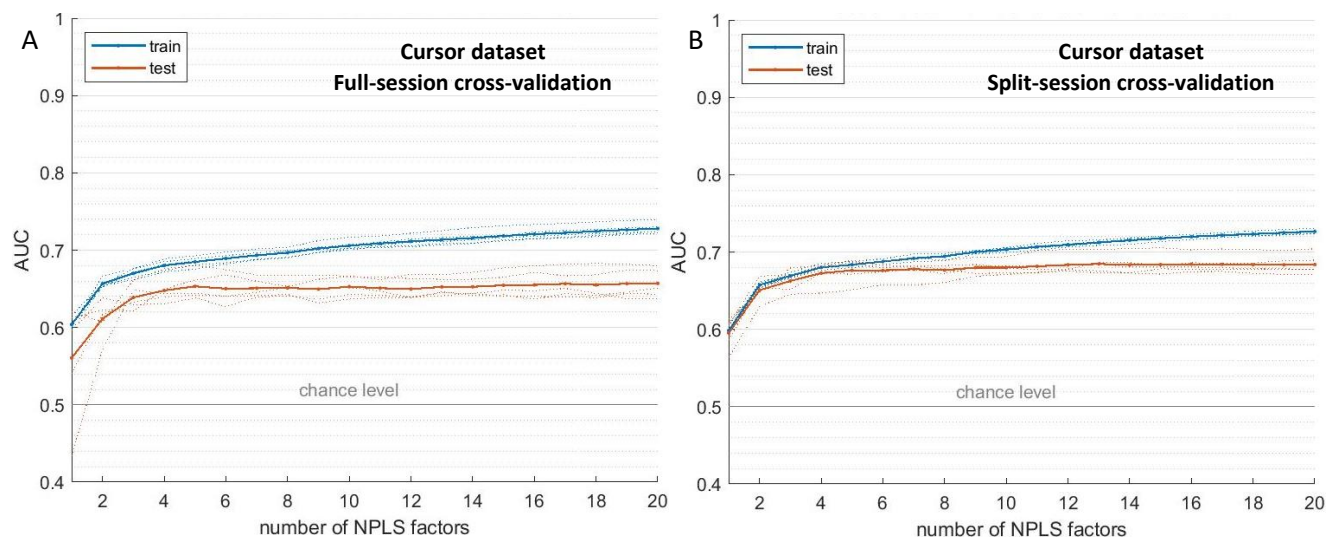


Figure 29 Performance measure (mean AUC of the ROC) for the detection of cMTP neural correlates in the bi-dimensional continuous task, depending on the number of factors kept in the NPLS algorithm and on the type of cross-validation used. Performances on the training sets are displayed in blue, while performances on the testing set are displayed in red. Each dotted line represents one of the five cross-validation folds, while the full lines represent the mean AUC over the five folds. (A) Full-session cross-validation. (B) Split-session cross-validation.

## II.5. Relative feature importance

Decoders obtained using the NPLS algorithm can be used to evaluate the relative importance of input features. This provides insight into which features are the most relevant for the decoding of cMTP neural correlates. We investigated the eMTP and cMTP decoding models per feature axis (temporal, frequential or spatial) for each experimental paradigm (Figure 31, Figure 30, Figure 32 and Figure 33).

For the eMTP decoders, in the Runner no\_MI dataset the temporal parameters at the end of the temporal window were more relevant than the ones at the beginning of the window. In the runner MI dataset, the very first temporal parameters were the most relevant along with the second half of the temporal window. In the Runner no\_MI dataset, the 10Hz to 30Hz frequency band was the most

relevant, followed by the 80Hz to 150Hz band. On the other hand, in the Runner MI dataset the 50Hz to 90Hz frequency band was the most relevant, followed by a peak in the 20Hz band.

For the cMTP decoders, the  $\beta_{temporal}$  have the same pattern for the Runner MI, Runner no\_MI and Exo datasets. For these three datasets, the temporal parameters on the edge of the temporal windows were lower than the temporal parameters closer to the center of the temporal window. For the Cursor dataset however, the temporal parameters at the beginning of the temporal window were the lowest and increased to reach a maximum at the end of the temporal window. The  $\beta_{frequency}$  and  $\beta_{spatial}$  were non-constant between the different datasets.

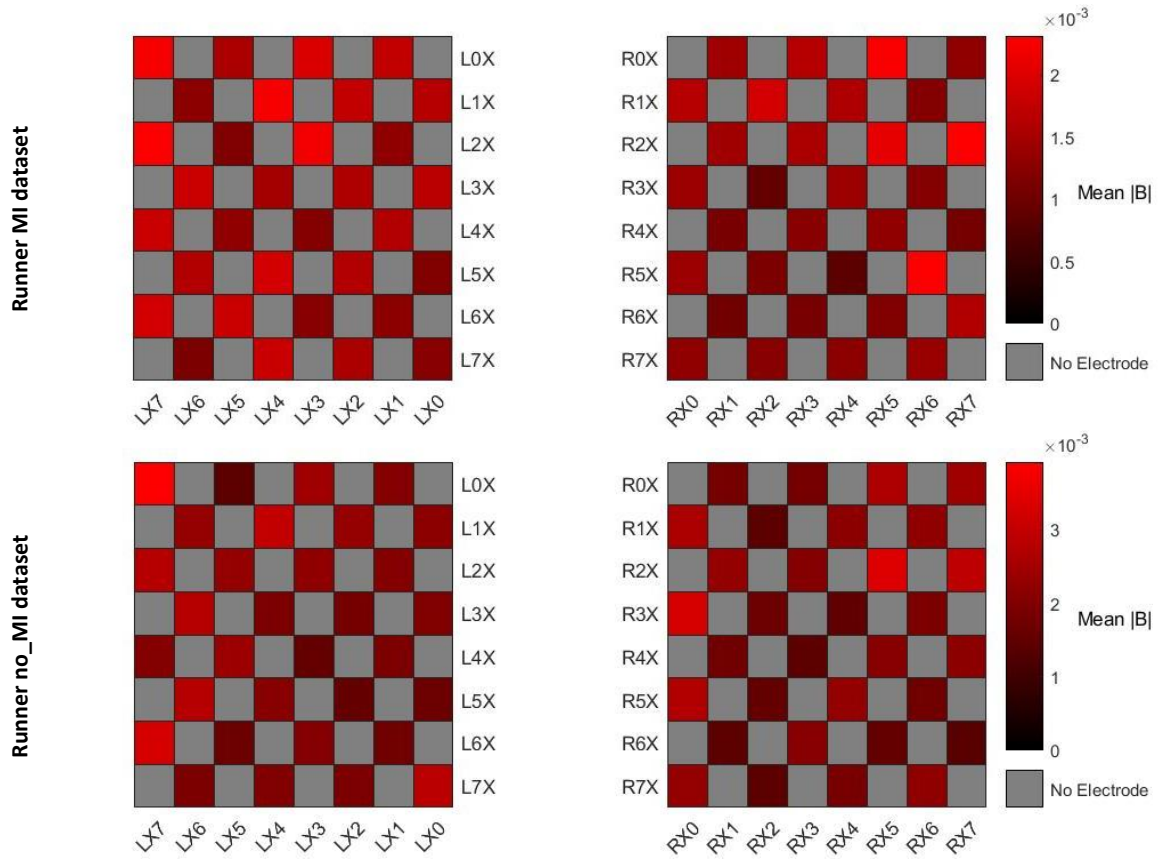


Figure 31 Relative importance of the spatial parameters of the eMTP decoders. These figures were obtained by averaging over the two non-visualized modalities.

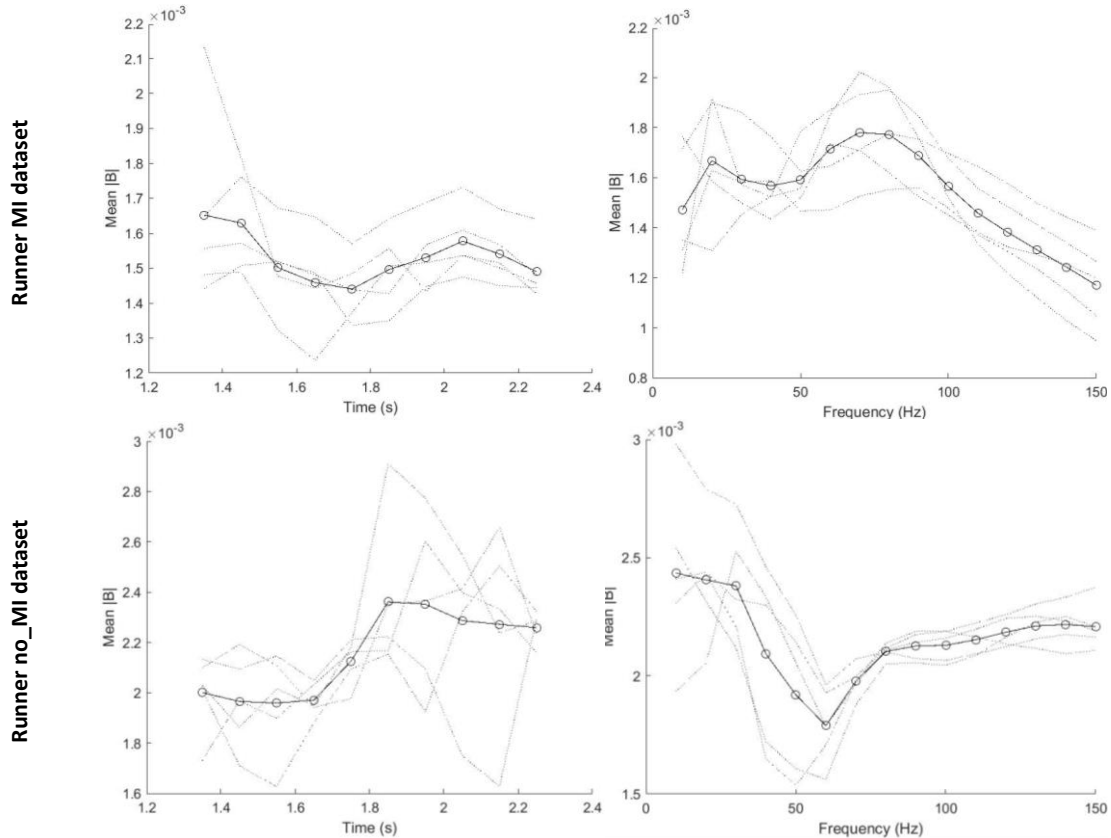


Figure 30 Parameters of the eMTP decoders along the temporal or frequential modalities. These figures were obtained by averaging over the two non-visualized modalities. Dotted lines are betas obtained for individual test folds, while plain lines are the average betas across each test fold.

## Chapter 5: Detection of neural correlates of task performance

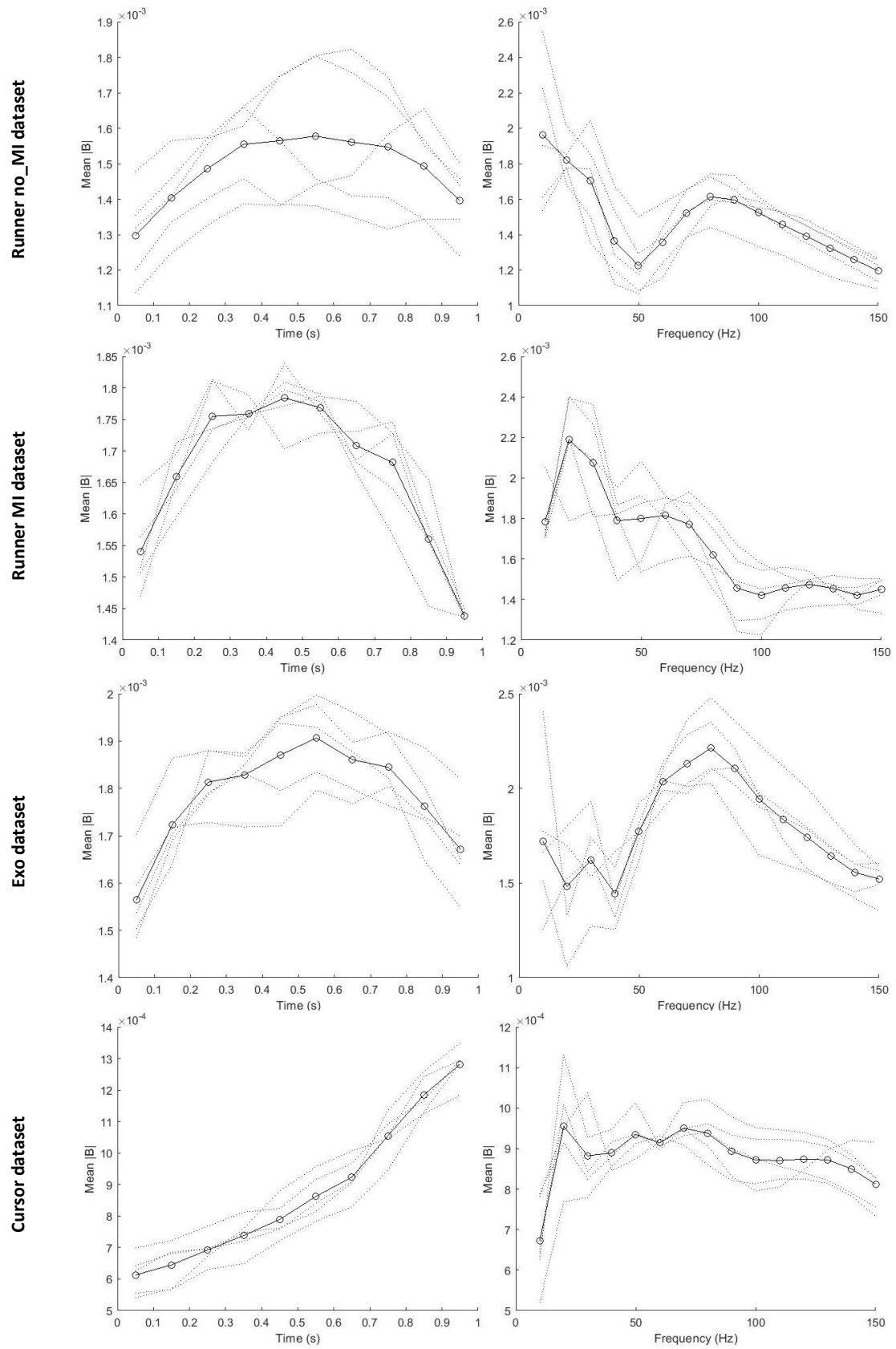


Figure 32 Parameters of the cMTP decoders along the temporal or frequential modalities. These figures were obtained by averaging over the two non-visualized modalities. Dotted lines are betas obtained for individual test folds, while plain lines are the average betas across each test fold.



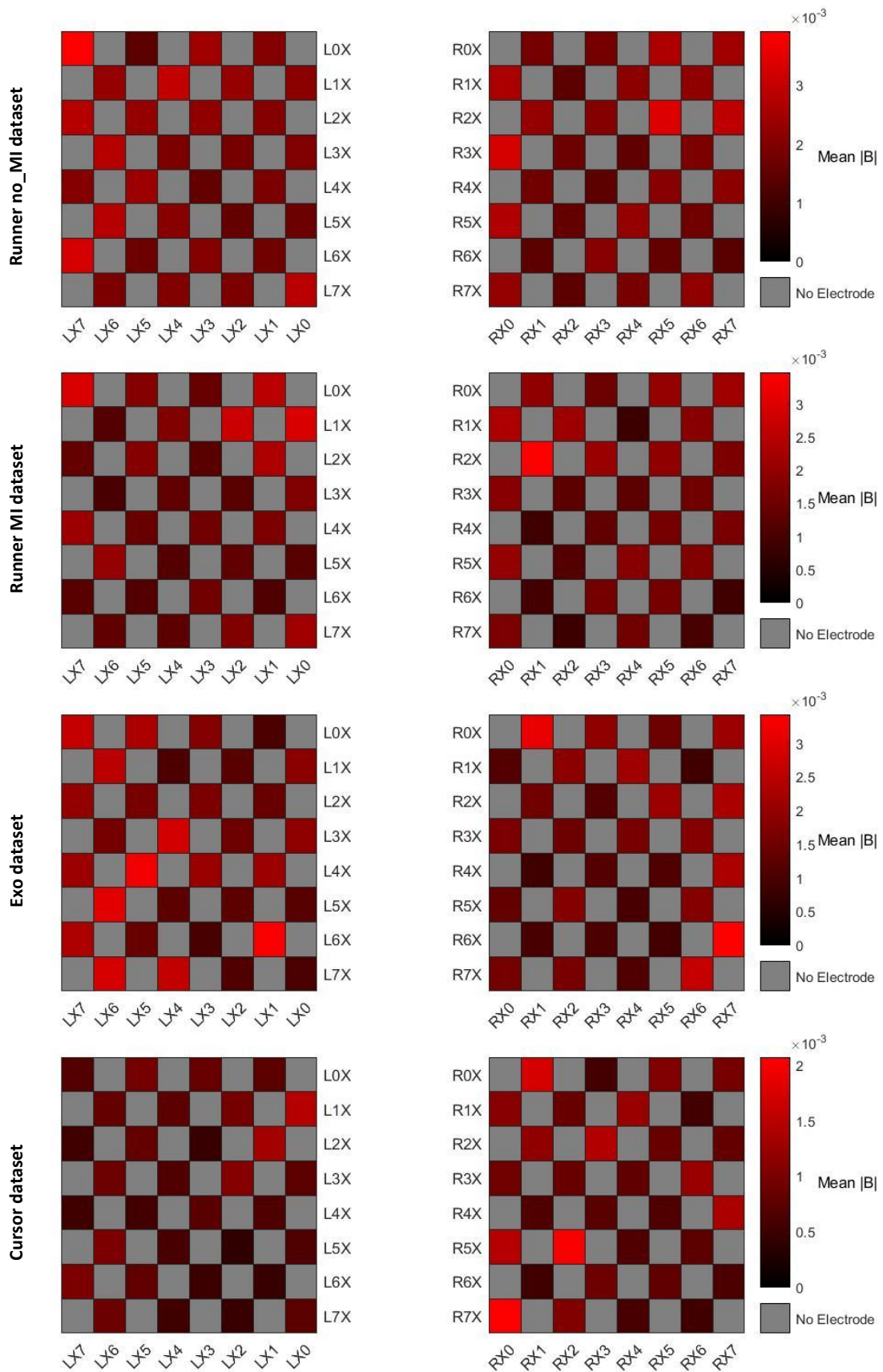


Figure 33 Relative importance of the spatial parameters of the cMTP decoders for each dataset. These figures were obtained by averaging over the two non-visualized modalities.

### III. Discussion

In this chapter, we investigated different methods to detect neural correlates of MTP from the sensorimotor cortex, during control or observation of a binary BCI or control of a multi-class BCI and a bi-dimensional BCI with continuous outputs. We first showed that we could detect neural correlates of event-locked MTP at the single trial level in the binary BCI. We also proposed a novel strategy for the detection of MTP, which is suitable for BCI that have multiple continuous outputs. We were able to detect these novel cMTP neural correlates at the single-trial level for different all 3 types of BCI types. Notably, detection of execution errors during control of a BCI with multiple continuous outputs was never achieved prior to this study. Less importantly, this study is also the first one to report single-trial detection of neural correlates of task performance in the sensorimotor cortex of a tetraplegic subject. Finally, the detection of MTP at the single trial-level from the sensorimotor cortex, during control of the BCI with motor imagery and with better accuracy for cMTP detection than chance level was a mandatory first step for the proof of concept of the aaBCI for complex motor BCIs. Nevertheless, several points are up to discussion regarding our results.

#### III.1. Event-locked population response to correct and erroneous events

The average time-frequency response to errors can be compared to the ones described by Milekovic et al. (2012) (Figure 34) or Wilson et al. (2019) (Figure 35). In the study by Milekovic et al., sharp increases in the amplitude of the time-frequency responses were observed for each subject. These changes took place in the 80Hz-100Hz frequency band (apart for one type of error for one subject), around 500ms after the erroneous event. However, these neural correlates of eMTP were not

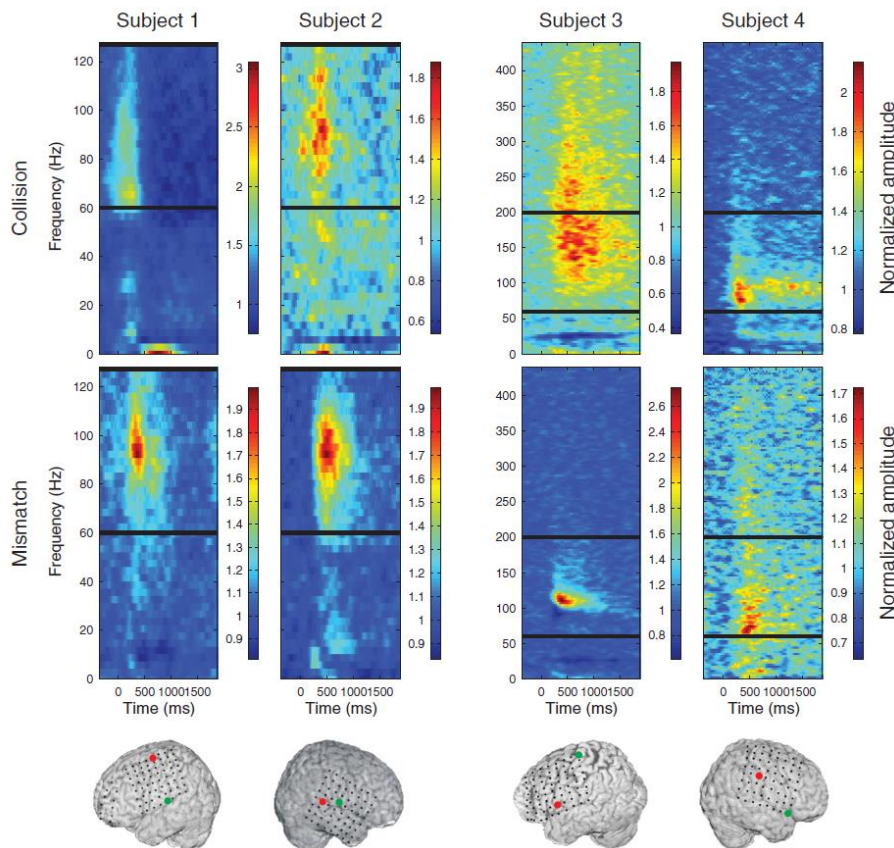


Figure 34 Average normalized spectrogram after two types of errors in the study by Milekovic et al. (2012). Red dots in the lower pictures show the location of the electrodes used to display the response to erroneous task outcome (Collision). Green dots show the location of electrodes used to display the response to errors during the task (erroneous change of direction of a one dimensional continuously controlled cursor). Figure taken from Milekovic et al. (2012).



consistently localized in the sensorimotor cortex. In the study by Wilson et al., they report significant differences in the high-gamma band (70Hz-100Hz) between neural responses to correct and erroneous events in several brain regions, including the sensorimotor cortex. Comparatively, in our study the strongest changes in the amplitude of the time-frequency responses after erroneous events were seen in frequencies below 50Hz, as well as the strongest differences between responses to correct and erroneous events. However, in both the studies by Milekovic et al. and Wilson et al., the localization where the high frequency modulations were observed varied strongly between subjects. In our study, the recording area covers a much narrower surface of the cortex, although with a higher spatial resolution. Since only one subject was included in our study and our recording area over the cortex was limited, the results obtained in the two studies discussed are not incompatible with ours. Indeed, in both studies, some subject did not have high-frequency modulations in the sensorimotor cortex after erroneous events. Nonetheless, the result of future visualization studies for the next subjects should be compared to the results discussed here.

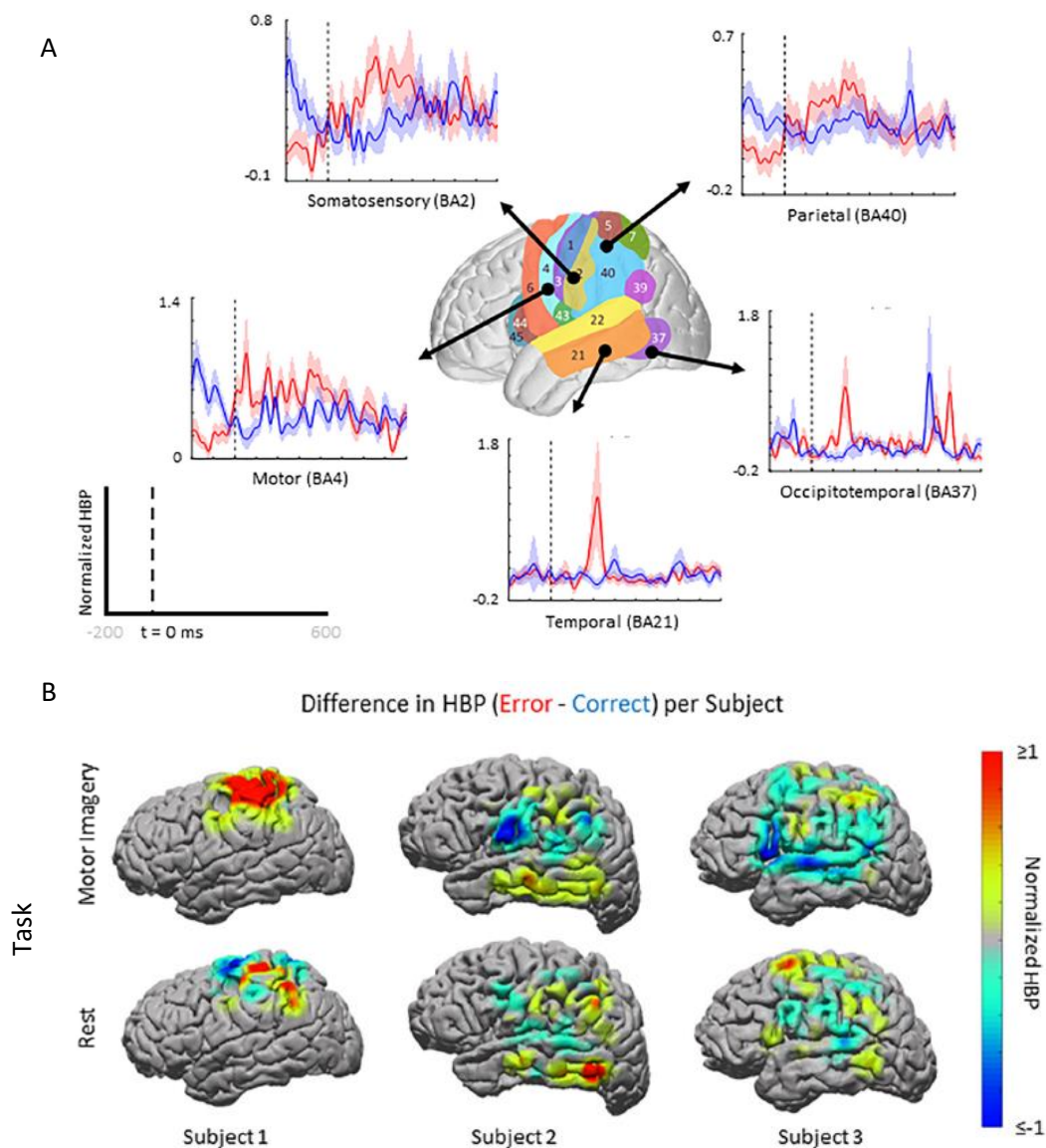


Figure 35 (A) Averaged normalized response to erroneous events (red curves) and correct events (blue curves) in the study by Wilson et al. (2019). (B) Difference in high gamma band power (HBP, 70Hz-100Hz) between correct and erroneous events in the same study. Figure adapted from Wilson et al. (2019).

### III.2. Relative detectability of event-locked versus continuous-in-time neural correlates of MTP

As discussed previously, event-locked neural correlates of task performance are ill-suited for the auto-adaptation of complex motor BCIs. However, if a large performance gap existed between the detection of neural correlates of cMTP and eMTP it may have been better to work with the limitations of eMTP instead of using cMTP. We tested the performance of both the eMTP and cMTP decoders for the Runner MI and Runner no\_MI datasets. In both datasets, the performances of the cMTP decoder were higher than the performances of the eMTP decoders, with an average increase of the AUC of 3.7%. Therefore, the single trial detection of correlates of motor task performances is both more desirable and more efficient with cMTP than eMTP. As mentioned above however, it may be the case that future subjects will have stronger responses in the sensorimotor cortex to erroneous events than the subject included in this study. Therefore, the difference in decoding capabilities of neural correlates of cMTP and eMTP should be investigated for future subjects, in order to confirm that cMTP detection is always more efficient, or at least as efficient, as eMTP detection.

We hypothesize that one of the reasons for the relatively low detectability of neural correlates of eMTP may be that the eMTP correlates are modulated by the length of correct or error periods prior to an event. Although the duration of these periods, or latency before each event, was partly controlled in the Runner no\_MI dataset, there was no inclusion or exclusion criterion based on it in the Runner MI dataset. We suggest that this latency may influence the brain response to events. For example, correct events after a long erroneous period may elicit a stronger brain response than after a short erroneous period. Due to the relatively small dataset acquired in this experiment, separating the events based on latency was not possible, but larger studies should take it into account when possible.

### III.3. Neural correlates of task performance during BCI observation or BCI control with motor imagery

As mentioned earlier, we record the cMTP neural correlates and the motor imagery neural correlates from the same brain regions. This situation could lead to confounds in the signals detected by the MTP decoders. We took care to minimize the possibility of confounds with motor imagery signals (Chapter 5:1.3.Data balance). Nevertheless, we also designed an experimental paradigm in which the subject did not perform motor imagery and only expected the effector to move according to the instructions. Being able to detect neural correlates of MTP even when no motor imagery is performed provides additional proof that the neural patterns detected by the MTP decoders are not by-products of the motor imagery tasks. Additionally, it also provides some insight on how the presence of motor imagery related neural patterns in the brain signals can influence the detection of neural correlates of MTP.

Our results show that the single-trial detection of neural correlates of eMTP or cMTP is more efficient when no motor imagery is performed. This is consistent with expectations since motor imagery signals can be considered as noise for the detection of MTP. An unforeseen result was however that the cMTP and eMTP decoders performed more consistently when motor imagery was performed. Indeed, higher standard deviations of the AUC over test folds were reported in the Runner no\_MI than in the Runner MI dataset, for both the eMTP and cMTP decoders. We suggest that this variation is due to a difference in the level of engagement. The existing literature on ErrPs (which are a specific eMTP) have them influenced by the engagement level of the user (Yeung et al., 2005), which may also be the case for other MTPs such as cMTP neural correlates recorded from the sensorimotor cortex. The level of engagement varied between the Runner MI and the Runner no\_MI datasets. In the Runner MI dataset, the subject may have been more engaged since he had active control over the avatar's actions. Comparatively, in the Runner no\_MI dataset the subject simply observed the avatar move.

Additionally, in the Runner no\_MI dataset we did not have any way of controlling whether the subject was properly doing the task (watch the avatar and expect it to move as if he was controlling it and followed the instructions). Therefore, the engagement level may have been further reduced by the subject not always properly doing the task requested.

### III.4. Relative feature importance

#### *Comparison of eMTP decoder parameters between the Runner no\_MI and Runner MI datasets*

As to continue from the previous point, there are differences in the eMTP decoder parameters' relative importance between the Runner MI and Runner no\_MI datasets. Firstly, the main frequential parameters in the Runner no\_MI dataset are the sub 50Hz frequencies, whereas the influence of these parameters in the Runner MI dataset is comparatively lower. At the same time, a control decoder trained on the same epochs has its main frequential parameters in the 10Hz-50Hz frequency band (see "Supplementary data: Motor imagery decoders trained on the same epochs as the MTP decoders"). We suggest that the discriminatory information found in the 10Hz-50Hz band for the control decoder could have a negative influence on the ability of the eMTP decoder when neural correlates of motor imagery are found in the data. However, this theory does not explain the difference in temporal parameters of the eMTP decoders between the Runner MI and the Runner no\_MI datasets. The MI based control decoder has relatively uniform importance of the temporal features, with a slight emphasis on the early parameters. Therefore, the stronger discrimination power of late temporal parameters in the Runner no\_MI dataset than in the Runner MI dataset was not expected. Although unlikely, we suggest that the difference in user engagement between the two datasets (MI vs no\_MI) could impact the user's latency of neural correlates of eMTP after events. This difference in latency of neural correlates would in turn explain the difference in relative importance of temporal parameters between the Runner MI and Runner no\_MI datasets. The spatial parameters differ between the two datasets, but the same general spatial patterns can be observed. The top right-hand quadrant of the right implant (i.e., electrode R25) as well as the top left-hand quadrant of the left implant (i.e., electrode L07) both remain among the most discriminative between the two datasets. We suggest the variations between the two groups of spatial parameters remain in the scope of what can be expected for decoders that are not trained using the same data, independently of the presence of neural correlates of motor imagery.

#### *Comparison of decoder parameters between cMTP decoders and control decoders*

One of the reasons for the creation of the Runner no\_MI dataset was to ensure that neural correlates of cMTP do exist and we're not mistakenly detecting neural correlates of motor imagery instead of neural correlates of cMTP. In order to provide an additional safeguard to this issue, we compared the decoder parameters between the cMTP decoders and control decoders trained offline on the same epochs as the cMTP decoders see "Supplementary data: Motor imagery decoders trained on the same epochs as the MTP decoders").

**Temporal parameters.** In the Runner MI dataset and Exo dataset, the temporal parameters were highly similar between the cMTP decoders and the control decoders. In the Cursor dataset, the temporal parameters are more distinct than for other datasets, but still close between the cMTP and control decoders. However, we did not expect significantly large differences between the temporal parameters of the cMTP and control decoders. Indeed, in the Runner MI and Exo datasets both the detection of cMTP and motor imagery neural correlates have the goal of detecting a state (error/correct for cMTP, avatar walking/idle for control decoder in the Runner MI dataset, right hand/left hand/right wrist/left wrist in the Exo dataset) that lasts for relatively long periods of time. Indeed, the labeling for the cMTP decoder only labeled data which had a correct or incorrect state for more than 1500ms, and the control decoders were trained on the same epochs as the cMTP decoder

which means the motor state were stable as well. In the Cursor dataset, there is some temporal filtering for the cMTP decoder labels since the cMTP labels are based on the averaged correctness over 500ms. However, in each epoch the neural data at the beginning of the epoch can be of a different correct/erroneous state than data at the end of the epoch. Additionally, due to the nature of the task, error and correct periods could be expected to last shorter than for classification tasks. Therefore, it was expected that the temporal parameters applied to the end of the epoch would be more discriminative than temporal parameters applied to the beginning of the epoch for the detection of neural correlates of cMTP. For the control decoder in the Cursor dataset, the position of the hand-shaped cursor changed continuously as the user controlled it. The labels for the control decoder changed at a fast pace in the data and the user could update his neural command at an equally fast pace to ensure the hand-shaped cursor is moving toward the target. Therefore, it was also expected to have temporal parameters applied to the end of the epoch be more discriminative than temporal parameters applied to the beginning of the epoch for the detection of neural correlates of motor imagery.

**Frequential parameters.** In the Runner MI dataset and Exo dataset, the frequential parameters were fairly similar between the control decoders and cMTP decoders. In the Cursor dataset however, the frequential parameters were more distinct between the cMTP decoder and the control decoder. Generally, it would have been desirable to have more differences in the frequential parameters between cMTP and control decoders. This would mean that discriminative information for both decoders was in different frequency bands and thus be less likely to disrupt one another.

**Spatial parameters.** For each dataset, the spatial parameters were largely distinct between the control decoders and cMTP decoders. This difference in spatial parameters strongly comforts the idea that the cMTP decoder is not detecting neural correlates of motor imagery instead of neural correlates of cMTP.

### *Stability of cMTP decoder parameters between datasets*

The cMTP decoder parameters varied strongly between the different datasets, mainly in the frequential and spatial modalities. This suggests that the neural correlates of cMTP used in this study are not stable across task. Interestingly, ErrPs have been shown to have some stability across tasks (Iwane et al., 2016; Riesel et al., 2013). Although ErrPs are different MTPs than the neural correlates of cMTP detected in this study, it may be possible these cMTPs are partly robust to task changes. However, to date no study investigated specifically the robustness to the cMTP neural correlates to task change. Future studies should focus on this point as robustness to task is a desirable property of cMTP decoders in the scope of the aaBCI framework.

### III.5. Temporal stability of neural correlates of cMTP in the sensorimotor cortex

It should be noted that the different recording sessions for each dataset were recorded over the course of several weeks or months. In the full-session cross validation scheme, the models were partly trained on data recorded far away temporally from the data they were tested on. This leads us to suggest that the error correlates we report in the motor cortex may exhibit temporal stability, similarly to ErrPs (Olvet and Hajcak, 2009). Temporal robustness of cMTP decoders is desired in the aaBCI framework, as the cMTP decoder is still trained in a supervised manner. Therefore, temporal robustness ensures that the cMTP decoders don't have to be retrained frequently, or ideally ever. Although that was not in the scope of the proof of concept performed in this study, future experiments should compare the difference in temporal stability between the cMTP decoder described here and control decoders. In order to make full use of the aaBCI framework, the temporal stability of the cMTP decoder needs to be non-marginally higher than the temporal stability of the control decoder.

# Chapter 6: Auto-adaptive BCI design using neural correlates of task performance for real-time labeling

In the previous chapter, we successfully detected neural correlates of cMTP. In this chapter, we use these cMTP neural correlates in an auto-adaptive BCI framework. cMTP neural correlates are used to estimate labels for the control decoder neural data. These estimated labels are used to train the control decoder in real-time during free use of the BCI.

## I. Methods

### I.1. Pseudo-online simulation

We implemented a pseudo-online simulation in order to test the aaBCI framework. This implementation provided conditions as close as possible to online use of the aaBCI. In order to do so, the datasets were split into three. One part was used to train the cMTP decoder as it would be required prior to use the aaBCI in online conditions. One part was used to train the control decoder using the aaBCI framework, i.e. without knowledge of the real labels for the control decoder. In online use, the neural data used in this part would be collected during free use of the BCI. In our case, the online simulation study was performed on already recorded datasets. Therefore, the subject was still instructed to perform specific actions during this period. The control decoder was trained in a pseudo-online manner. Neural data was fed to the algorithm through a loop mimicking online data acquisition. Label estimation and training of the control decoder were performed using the aaBCI every time fifteen seconds of labeled data were acquired. However, the new control decoder had no influence on the actions of the BCI, as the datasets were pre-recorded. Finally, one part of each dataset was used to test the performance of the newly trained control decoder.

### I.2. Formation of datasets for cMTP decoders training

We trained cMTP decoders in a similar way as in the previous chapter. Data labeling, data balance, neural feature extraction and training algorithm were similar to what is described in “Chapter 5: Detection of neural correlates of task performance”. For each dataset, the label for the cMTP decoder is noted  $y^t_{cMTP} \in \{0, 1\}$  for each epoch  $t$ . The output of the cMTP decoder for any epoch  $t$  is noted  $\hat{y}^t_{cMTP} \in \mathbb{R}$ .

### I.3. Formation of datasets for the training of control decoders in the auto-adaptive BCI framework

In this pseudo-online simulation of online use of the aaBCI, new control decoders are trained using labels estimated thanks to the aaBCI framework. These labels are partly derived from the output of the cMTP decoder, which cannot be expected to have a perfect accuracy. Therefore, these derived control labels cannot be expected to be perfect either. We designate these labels as ‘*estimated labels*’ for the control decoder. For a given epoch  $t$ , these estimated labels are noted  $\tilde{y}^t_{control} \in \mathbb{R}^m$ , with  $m$



the dimension of the output of the control decoder ( $m = 1$  in the Runner MI dataset,  $m = 4$  in the Exo dataset and  $m = 2$  in the Cursor dataset). For classification tasks (Runner and Exo datasets), we have  $\hat{\mathbf{y}}^t_{control} \in \{0, 1\}^m$  a dummy encoding of the estimated class, with  $m$  the number of classes controlled. For the regression task (Cursor dataset), we have  $\hat{\mathbf{y}}^t_{control} \in \mathbb{R}^m$  with  $m = 2$  the number of continuous dimensions controlled. The generation of these estimated labels can be separated in two distinct steps. In the first step, we select epochs that can reliably be estimated as correct or erroneous. Other samples are discarded. In a second step, we estimate the control decoder labels for these epochs.

#### *Inclusion of data in datasets for the training of control decoder in the aaBCI framework*

As the cMTP decoder is imperfect, there will be some noise in the estimated labels  $\hat{\mathbf{y}}^t_{control}$  compared to the real labels  $\mathbf{y}^t_{control}$ . To limit the noise in the estimated labels, we discard epochs for which there is uncertainty on the task performance estimation obtained from the cMTP decoder. The output of the cMTP decoder is a continuous variable. We decided to use two thresholds for classification of epochs as correct or erroneous. One threshold is used for the classification of epochs as correct ( $th_{corr}$ ) and the other is used to classify epochs as erroneous ( $th_{err}$ ). Epochs for which  $\hat{y}^t_{cMTP} > th_{corr}$  were considered correct. Epochs for which  $\hat{y}^t_{cMTP} < th_{err}$  were considered erroneous. Samples for which  $th_{err} < \hat{y}^t_{cMTP} < th_{corr}$  were not included into the new training sets. The output of the cMTP decoder on the training set can be used as an estimation of the probability densities for the error and correct class. We model the output of the cMTP decoder  $\hat{\mathbf{y}}_{cMTP}$  on the training set as a mixture of two Gaussians. Each Gaussian is used to model one of the two classes of the cMTP decoder. The correct class Gaussian is noted  $\mathcal{N}(\mu_{corr}, \sigma_{corr}^2)$  and the error class Gaussian is noted  $\mathcal{N}(\mu_{err}, \sigma_{err}^2)$ . The parameters of the two Gaussians were estimated on the training data as the mean and standard deviation over the correct or erroneous samples:

$$\begin{aligned}\mu_{err} &= \frac{1}{n_{err}} \sum_{\{t: y^t_{cMTP}=0\}} \hat{y}^t_{cMTP} \\ \mu_{corr} &= \frac{1}{n_{corr}} \sum_{\{t: y^t_{cMTP}=1\}} \hat{y}^t_{cMTP} \\ \sigma_{err}^2 &= \frac{1}{n_{err} - 1} \sum_{\{t: y^t_{cMTP}=0\}} (\mu_{err} - \hat{y}^t_{cMTP})^2 \\ \sigma_{corr}^2 &= \frac{1}{n_{corr} - 1} \sum_{\{t: y^t_{cMTP}=1\}} (\mu_{corr} - \hat{y}^t_{cMTP})^2\end{aligned}$$

With  $n_{err}$  and  $n_{corr}$  the number of erroneous and correct samples respectively in the training set. If normalized for the number of epochs in each class, these Gaussians would be equivalent to the estimated probability densities of having a given cMTP decoder output when an epoch is from the correct or error class ( $P(\hat{y}^t_{cMTP}|corr)$  and  $P(\hat{y}^t_{cMTP}|err)$ ). The  $th_{corr}$  and  $th_{err}$  disjointed thresholds leave some epochs unclassified, which corresponds to epochs for which the estimated probability of being correct or erroneous is too close to one another. This solution effectively decreases the noise in the estimated labels at the cost of a decrease in the size of the training set for the control decoder.

The thresholds were then defined as  $th_{corr} = \mu_{corr} + a \sigma_{corr}$  and  $th_{err} = \mu_{err} - a \sigma_{err}$  with  $a$  a constant  $a$  that balances the trade-off between the number of samples added to the training set and

the confidence of their labeling by the cMTP decoder (Figure 36). This strategy has the advantage of not being influenced by the balance of data. We set  $a = 1$  in this study. Optimization of this hyperparameter could be performed with nested cross-validation, at the cost of setting data aside for this purpose. Although we did not optimize  $a$ , we evaluated the impact of  $a$  on the trade-off between amount of data unlabeled and performance of the cMTP decoder on the data used to train the control decoder. We report the accuracy of the cMTP decoder, the true positive rate and the false positive rate for the detection of errors depending on the proportion of data discarded and  $a$ . In the analysis stage, each sample has a corresponding  $\hat{y}_{cMTP}^t$  value. Each sample thus has exactly one corresponding  $a$  value for which  $\hat{y}_{cMTP}^t = \mu_{corr} + a\sigma_{corr}$  and one  $a$  value for which  $\hat{y}_{cMTP}^t = \mu_{err} - a\sigma_{err}$ . For each of these values of  $a$  over all the samples in the test dataset there is a corresponding percentage of data discarded. For each of the  $a$  values, we also compute the accuracy of the cMTP decoder, the true positive rate and the false positive rate for the detection of errors on the data not discarded.

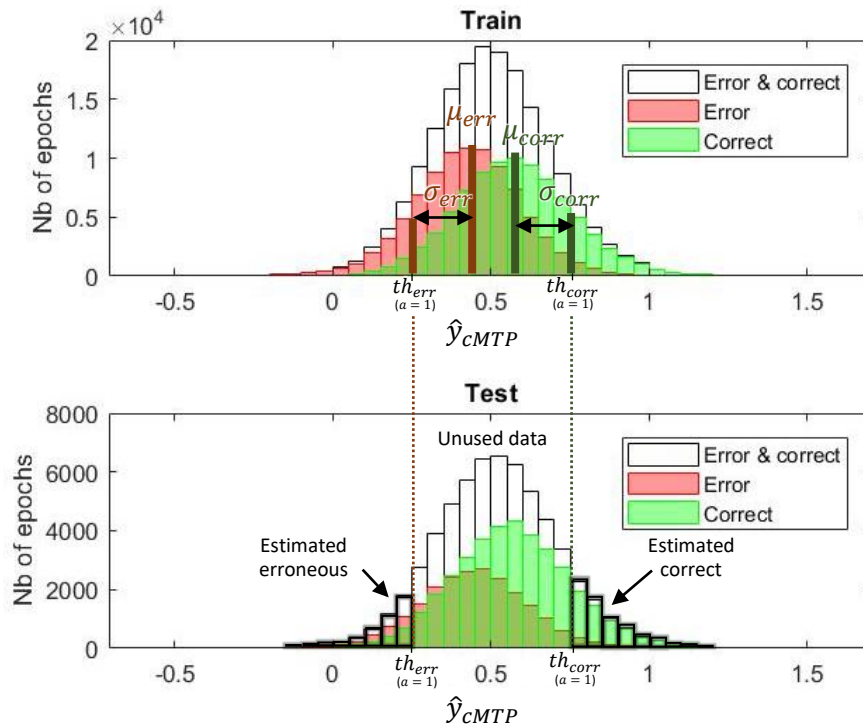


Figure 36 Example of histogram of the output of the cMTP decoder on one training fold and its associated test set. The thresholds for the inclusion of epochs in the training set of the control decoder are based on a trade-off parameter  $a$  and the means and standard deviations of the Gaussians fitted to the error and correct class on the training set.

#### Labeling of data in the control decoder training dataset

Updates of the control decoder were performed every time there was fifteen seconds of labeled data available. For an update  $u$ , we note the estimated control labels as  $\tilde{\mathbf{Y}}^u_{control} = (\tilde{\mathbf{y}}^{t_1}_{control}, \dots, \tilde{\mathbf{y}}^{t_n}_{control})^T$ , with  $(t_1, \dots, t_n)$  the time points of the epochs that were labeled between update  $u - 1$  and update  $u$  ( $n = 150$ ). Similarly, we note  $\tilde{\mathbf{X}}^u_{control} \in \mathbb{R}^{n \times \tau \times f \times s}$  the neural features for the control decoder that were acquired at times  $(t_1, \dots, t_n)$ . We define *re-labeling functions* as functions that uses the output of the cMTP decoder and the output of the control decoder to derive estimated labels for the control decoder. The automatic creation of a new training set for the auto-adaptive BCI is defined as:

$$\tilde{\mathbf{y}}^t_{control} = \varphi(\hat{y}_{cMTP}^t, \hat{\mathbf{y}}^t_{control})$$

$$\tilde{\mathbf{X}}^u_{control} = \mathbf{X}^u_{control}$$

We used two re-labeling functions  $\varphi_1$  and  $\varphi_2$  depending on the dataset used.



**Runner MI and Exo datasets.** For the Runner MI and Exo datasets, we used  $\varphi_1: ]-\infty; th_{err}] \cup [th_{corr}; \infty[ \times \mathbb{R}^m \mapsto \mathbb{R}^m$ , with  $m$  the number of classes:

$$\varphi_1(\hat{\mathbf{y}}^t_{cMTP}, \hat{\mathbf{y}}^t_{control}) = \begin{cases} \mathbf{e}_{c_1^t}, & \text{if } \hat{\mathbf{y}}^t_{cMTP} > th_{corr} \\ \mathbf{e}_{c_2^t}, & \text{if } \hat{\mathbf{y}}^t_{cMTP} < th_{err} \end{cases}.$$

With  $(\mathbf{e}_i)_{i=1,\dots,m}$  the notation for the canonical base of  $\mathbb{R}^m$  and  $c_k^t$  the  $k^{th}$  most probable class estimated by the control decoder for sample  $t$ , defined as:

$$c_1^t = \operatorname{argmax}_{i \in \{1,\dots,m\}} \hat{y}_{i\ control}^t,$$

$$k \in \{2, \dots, m\}, \quad c_k^t = \operatorname{argmax}_{i \in \{1,\dots,m\} \setminus \{c_1^t, \dots, c_{k-1}^t\}} \hat{y}_{i\ control}^t.$$

The output of the relabeling function  $\varphi_1$  corresponds to the dummy encoding of the estimated class label. If a sample was estimated correct ( $\hat{\mathbf{y}}^t_{cMTP} > th_{corr}$ ), the estimated label is the most probable class at time  $t$ , i.e. the one corresponding to the action performed by the BCI at time  $t$ . If a sample was estimated erroneous ( $\hat{\mathbf{y}}^t_{cMTP} < th_{err}$ ), the estimated label was the second most probable class at time  $t$ .

**Cursor dataset.** For the Cursor dataset, we used  $\varphi_2: [th_{corr}; \infty[ \times \mathbb{R}^m \mapsto \mathbb{R}^m$ ,  $m$  being the number of continuous outputs of the control decoder:

$$\varphi_2(\hat{\mathbf{y}}^t_{cMTP}, \hat{\mathbf{y}}^t_{control}) = \hat{\mathbf{y}}^t_{control}.$$

The relabeling function  $\varphi_2$  only keeps samples that were estimated corrects, with the label being the one estimated by the control decoder.

#### 1.4. Formation of datasets for the training of control decoders for comparison with the aaBCI framework

The control decoder trained using the aaBCI framework was compared to control decoders trained in two other ways. The first one was control decoders trained supervisedly with the labels of each epoch known. In the Runner and Exo datasets, the output label  $\mathbf{y}^t_{control}$  for each epoch  $t$  was a vector containing a dummy encoding of the real class label, which was the desired state of the BCI at the end time of the epoch. In the Cursor dataset, the output label  $\mathbf{y}^t_{control}$  for each epoch  $t$  contained the  $x^t$  and  $y^t$  directed Cartesian distance from the cursor to the target (Figure 37). The matrix of all real control decoder input labels is noted  $\mathbf{Y}_{control} = (\mathbf{y}^1_{control}, \dots, \mathbf{y}^N_{control})^T$  with  $\mathbf{y}^t_{control} \in \mathbb{R}^m$  and  $m$  the number of possible classes (Runner and Exo) or the number of degrees of freedom (Cursor).

Additional control decoders were trained for the purpose of comparison with the aaBCI. These control decoders were trained following the aaBCI framework, but with random output of the cMTP decoder.

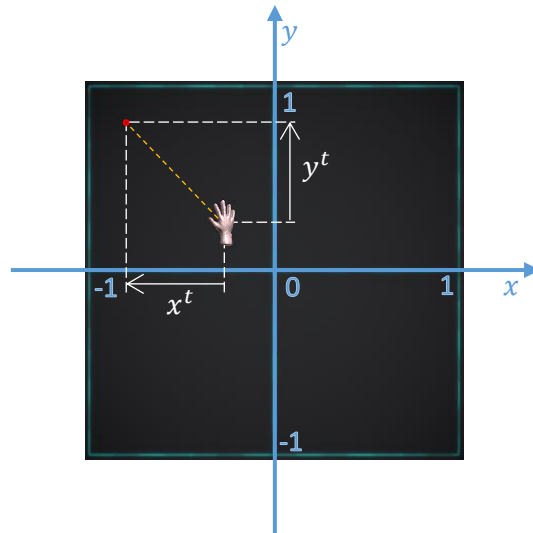


Figure 37 Illustration of how labels are defined for the supervised training of the control decoder in the Cursor dataset

### 1.5. Algorithm for the training of control decoders in the aaBCI framework

The main goal of the aaBCI is to enable online training and update of its control decoder during self-directed use. Therefore, the algorithm used to train the control decoder is required to function online. The rate of labeled data acquisition is not necessarily as high as in online-supervised learning, since not every epoch is labeled. This provides some more flexibility in the online capacity of the algorithm used, depending on the trade-off parameter  $\alpha$  between the proportion of data labeled and the accuracy of the labels. We used the REW-NPLS (Benabid et al., 2019; Eliseyev et al., 2017; Eliseyev and Aksenova, 2013) as it can be updated online at high rates, is not limited by growing amounts of training data, is suited to high-dimensional, tensor-shaped data and is relatively robust to noise. The robustness to noise is especially valuable for the aaBCI-training of the control decoder. Indeed, the process of label estimation is bound to create some label noise. Compared to supervised training, aaBCI-training inherently increases the label noise due to the estimated labels being less accurate than labels acquired in dedicated training sessions.

### 1.6. Performance evaluation

To estimate the performances of the auto-adaptive BCI, each dataset was separated in three non-overlapping splits, each split containing the same number of recording sessions (up to one session difference). We performed a pseudo-cross-validation: one split was used to train the cMTP decoder, one split was used to train the control decoder and one split was used to test the performances of the control decoder. Permuting the roles of each split led to six performance measures for each effector.

The performance of the control decoder was assessed on test splits. Datasets that featured different effectors used different performance criteria. For the Runner MI dataset (binary control decoder), performance was evaluated using the AUC of the ROC curve of the control decoder. For the Exo dataset (multi-class classification), performance was evaluated using a generalized version of the AUC of the ROC curve for multi-class classification (Hand and Till, 2001). The generalized version of the AUC to a multiclass problem with  $m$  classes is defined as follows:

$$AUC = \frac{2}{m(m-1)} \sum_{i=1..m} \sum_{\substack{j=1..m \\ i \neq j}} \hat{A}(i|j)$$

With  $\hat{A}(i|j)$  the estimated probability that a randomly drawn member of class  $j$  will have a lower estimated probability of belonging to class  $i$  than a randomly drawn member of class  $i$ .  $\hat{A}(i|j)$  is obtained by computing the binary test AUC between class  $i$  and  $j$  using the scores from class  $i$ , while  $\hat{A}(j|i)$  is obtained by computing the binary test AUC between class  $i$  and  $j$  using the scores from class  $j$ .

For the Cursor dataset (bi-dimensional continuous output of the control decoder), performance was evaluated using the cosine similarity between the predicted trajectory and the optimal trajectory to reach the current target. If we note as  $\mathcal{T}$  an ensemble of epochs with  $T$  elements, the cosine similarity on  $\mathcal{T}$  is computed as follows:

$$\text{CosSim}(\mathcal{T}) = \frac{1}{T} \sum_{t \in \mathcal{T}} \frac{\hat{\mathbf{y}}^t_{\text{control}} \cdot \mathbf{y}^t_{\text{control}}}{\|\hat{\mathbf{y}}^t_{\text{control}}\| \|\mathbf{y}^t_{\text{control}}\|}$$

The performance of the auto-adaptive BCI for each effector was assessed with the mean of its associated performance criterion over each test split. Additionally, we also assessed the performance of the cMTP decoder on the split used to train the control decoder.

## II. Results

### II.1. Performance of the cMTP decoders on the full control decoder training dataset

Although we previously evaluated the performances of the cMTP decoders on each dataset in Chapter 5, it was necessary to perform this evaluation again for the aaBCI study. Indeed, the previous evaluation of the cMTP decoders may not properly reflect their performance in the pseudo-online aaBCI simulation performed here. Previously, the cMTP decoders were trained using 80% of the data in their respective datasets. In pseudo-online aaBCI simulations, part of the data must be set aside to train the control decoder and part of the data must be set aside to test the performances of the control decoder (in real use-cases of the aaBCI, the cMTP decoder labels would not be known on either of these data parts, and therefore cannot be used to train the cMTP decoders). Therefore, in the pseudo-online aaBCI simulation the cMTP decoders were trained using significantly less data, i.e. 33.3% of the data in their respective datasets. As the size of the training datasets could impact the cMTP decoder performances, we report here the new performances of the cMTP decoders on the parts of the datasets set aside to train the control decoders.

In the Runner MI dataset, the mean AUC of the ROC curve of the cMTP decoder was  $0.5754 \pm 0.0321$ . In the Exo dataset, the mean AUC of the ROC was  $0.5757 \pm 0.0590$ . In the Cursor dataset, the mean AUC of the ROC was  $0.6211 \pm 0.0120$  (Figure 38).

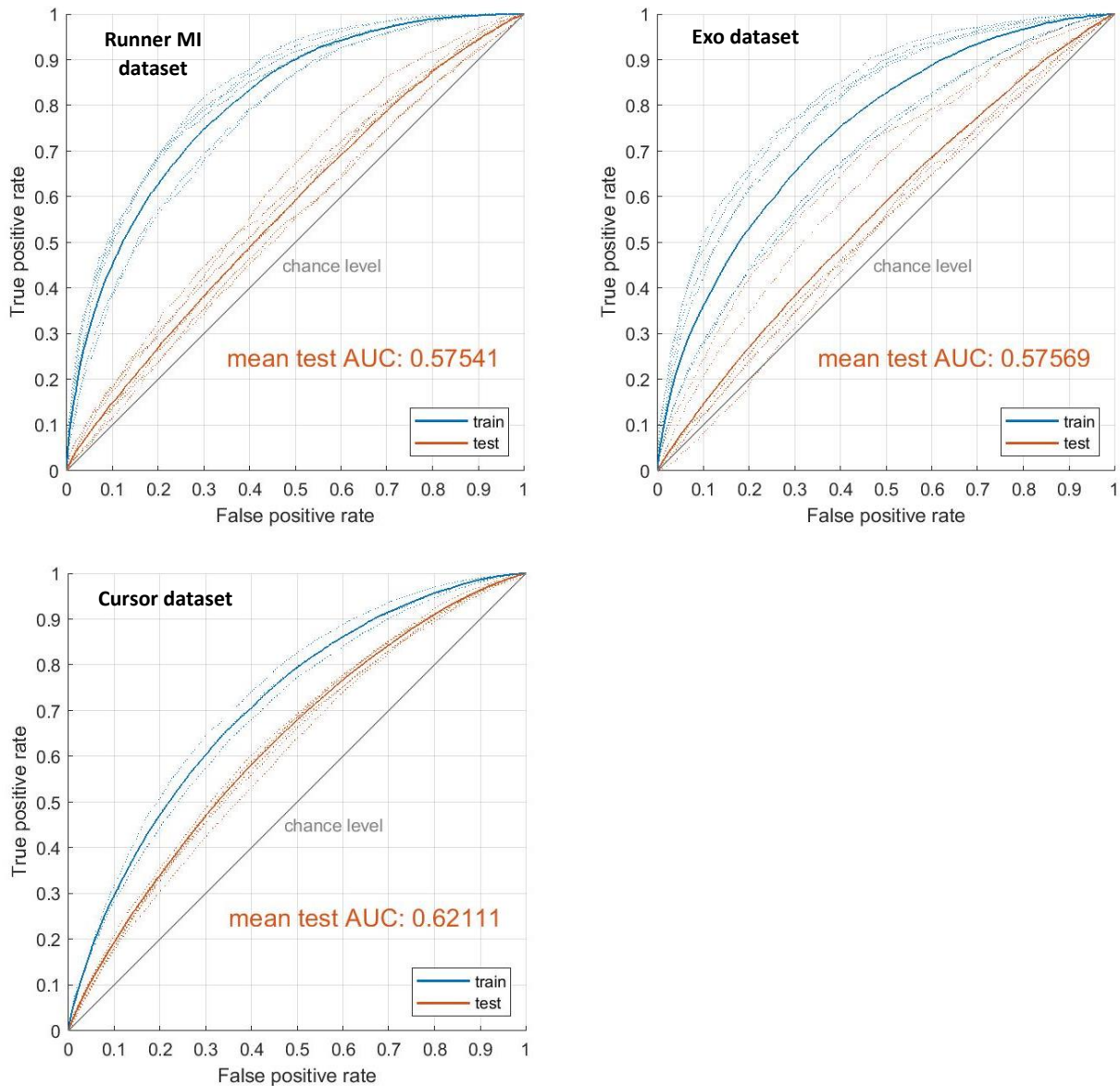


Figure 38 Performances of the cMTP decoders used in the aaBCI online simulation. Shown are the receiver operating characteristic curves of the cMTP decoders for the detection of epochs labeled as ‘errors’. Performances on the training sets are displayed in blue, while performances on the testing set are displayed in red. In the aaBCI online simulation, the test set of the cMTP decoders and the train set of the control decoders are on the same data split. Each dotted line represents one of the six cross-validation folds, while the full lines represent the interpolated mean ROC over the six folds.

## II.2. Impact of the trade-off parameter $\alpha$ on the cMTP decoder performances

The parameter  $\alpha$  balanced the trade-off between the amount of data labeled for the training of the control decoder and the performance of the cMTP decoder on this data. With  $\alpha = 1$ , in the Runner MI dataset 15.5% of the data was labeled while the accuracy of the cMTP decoder was 65.7%, the true positive rate was 0.57 and the false positive rate was 0.33 (Figure 39). In the Exo dataset, 17.5% of the data was labeled while the accuracy of the cMTP decoder was 70%, the true positive rate was 0.44 and the false positive rate was 0.24 (Figure 39). Finally, in the Cursor dataset 25.7% of the data was labeled while the accuracy of the cMTP decoder was 65.6%, the true positive rate was 0.63 and the false positive rate was 0.43 (Figure 39).

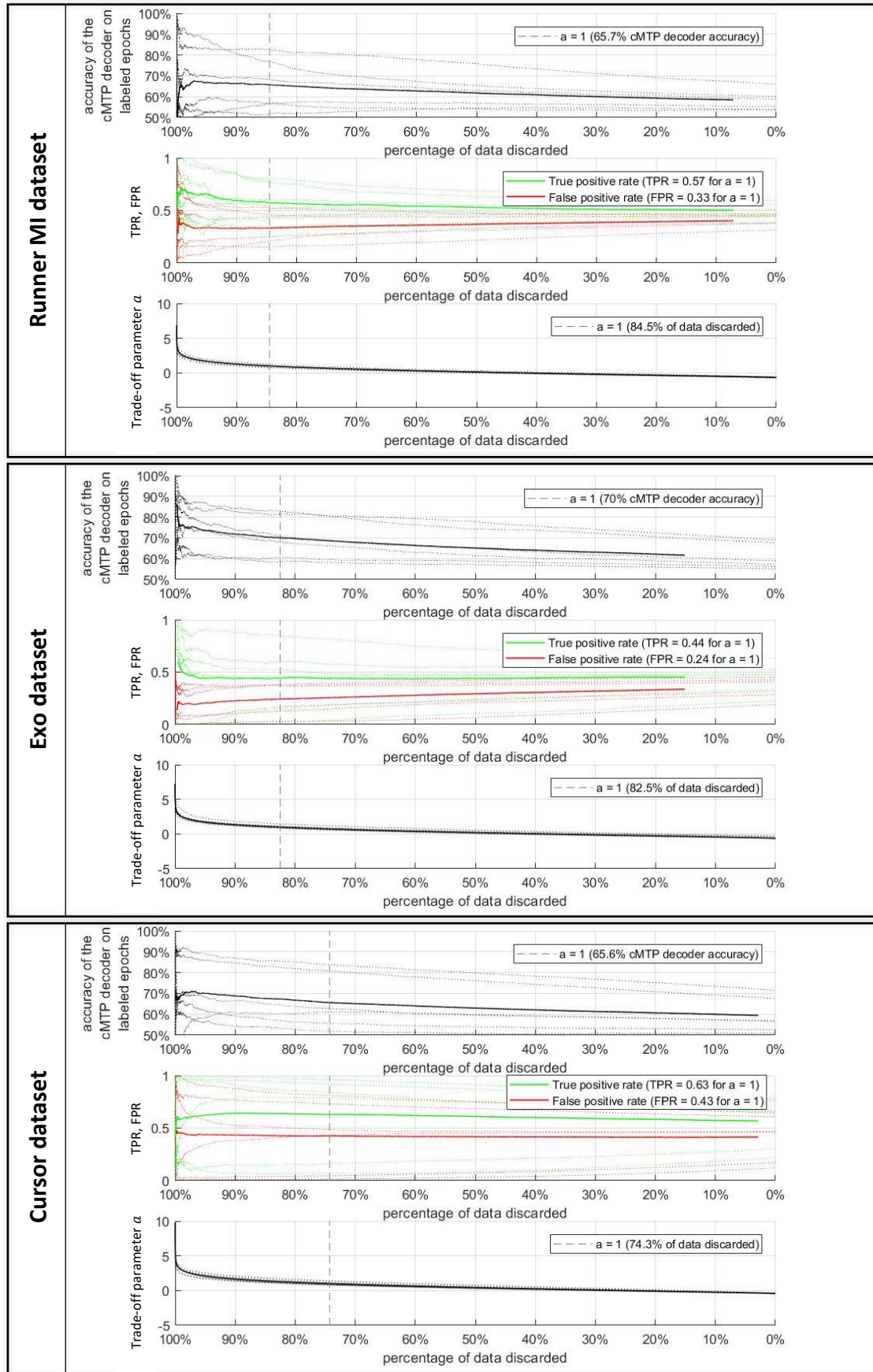


Figure 39 Impact of the trade-off parameter  $\alpha$  on the accuracy of the labels provided by the cMTP decoders, the true positive and false positive rates, as well as the amount of data not included in the training set of the control decoder. Each dotted line represents one of the six cross-validation folds, while the full lines represent the interpolated mean ROC over the six folds.



Additionally, it can be noted that as more data gets discarded, the TPR, FPR and accuracy becomes more variable. As more data is discarded, these values are computed with less samples and are thus less reliable and more prone to variations. Inversely, when less data is discarded, these values are more stable, but they also reflect poorer performances due to more samples being mislabeled.

### II.3. Accuracy of estimated labels for the control decoder

The aaBCI uses estimated labels for the training of the control decoder. The estimated labels have noise compared to the real labels. In the Runner dataset, the accuracy of the estimated labels was 64.9% (Figure 40). In the Exo dataset, the general accuracy of the estimated labels was 64.5%. More specifically for each motor class, the accuracy of the estimated label was 54.1% for the right-hand movement, 71.1% for the left-hand movement, 64.5% for the right wrist rotation and 68.4% for the left wrist rotation. In the Cursor dataset, 63.3% of the estimated labels were less than 60° away from the actual labels, 11.4% were in between 60° and 90° and 25.3% were more than 90° away from the actual labels.

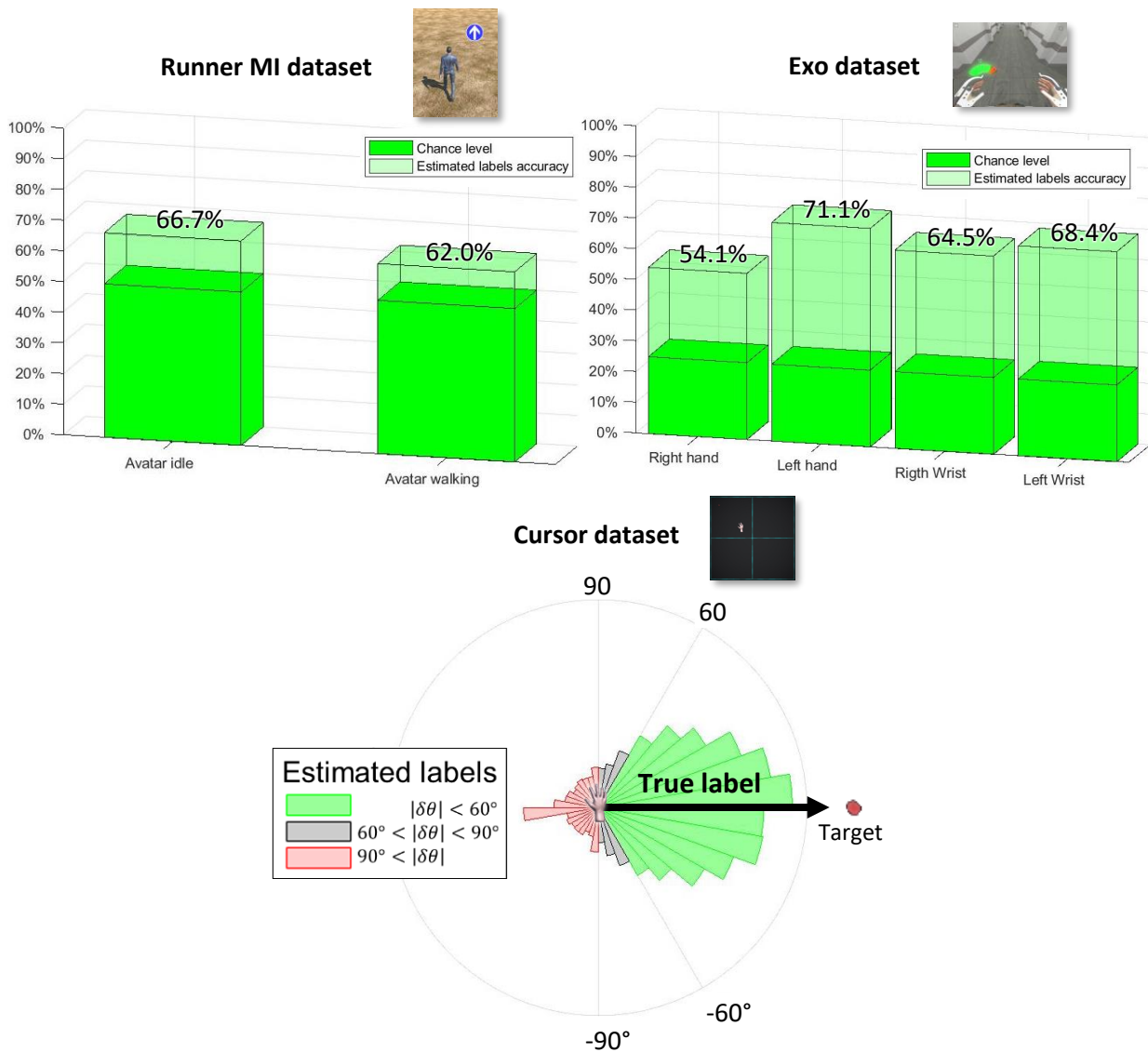


Figure 40 Performance of the aaBCI framework for the labeling of data.

#### II.4. Performance of control decoders trained using the aaBCI

Finally, the performance of the aaBCI is best measured by the performance of the control decoder compared to control decoders trained supervisedly or with random output of the cMTP decoder. In the Runner dataset, the mean AUC of the ROC of the control decoder was  $0.6360 \pm 0.0958$  when trained auto-adaptively compared to  $0.8958 \pm 0.0153$  when trained supervisedly and  $0.5007 \pm 0.0691$  when trained with random cMTP decoder outputs. There was a significant effect of training type on the performance of the control decoder (Friedman test, p-value = 0.0009). There was a significant difference between each pair of training methods: between auto-adaptive training with randomized cMTP output and auto-adaptive training (two-sided Wilcoxon-Mann-Whitney test,  $p=0.0411$ ), between the auto-adaptive training with randomized cMTP output and supervised training (two-sided Wilcoxon-Mann-Whitney test,  $p=0.0022$ ) and between supervised training and auto-adaptive training (two-sided Wilcoxon-Mann-Whitney test,  $p=0.0022$ ).

In the Exo dataset, the mean AUC of the generalized ROC of the control decoder was  $0.7595 \pm 0.0278$  when trained auto-adaptively compared to  $0.8177 \pm 0.0301$  when trained supervisedly and  $0.5163 \pm 0.0580$  trained with random cMTP decoder outputs. There was a significant effect of training type on the performance of the control decoder (Friedman test, p-value = 0.0009). There was a significant difference between each pair of training methods: between auto-adaptive training with randomized cMTP output and auto-adaptive training (two-sided Wilcoxon-Mann-Whitney test,  $p=0.0022$ ), between the auto-adaptive training with randomized cMTP output and supervised training (two-sided Wilcoxon-Mann-Whitney test,  $p=0.0022$ ) and between supervised training and auto-adaptive training (two-sided Wilcoxon-Mann-Whitney test,  $p=0.0087$ ).

In the Cursor dataset, the mean cosine similarity of the control decoder was  $0.1589 \pm 0.0668$  when trained auto-adaptively compared to  $0.2107 \pm 0.0664$  when trained supervisedly and  $-0.0231 \pm 0.0327$  when trained with random cMTP decoder outputs. There was a significant effect of training type on the performance of the control decoder (Friedman test, p-value = 0.0020). There was a significant difference between auto-adaptive training with randomized cMTP output and auto-adaptive training (two-sided Wilcoxon-Mann-Whitney test,  $p=0.0022$ ) and between the auto-adaptive training with randomized cMTP output and supervised training (two-sided Wilcoxon-Mann-Whitney test,  $p=0.0022$ ). However, there was no significant difference between supervised training and auto-adaptive training (two-sided Wilcoxon-Mann-Whitney test,  $p=0.3095$ ). The performances of the control decoders depending on the training method are summarized in Table 5 and Figure 41.

Table 5 Performances of the control decoders trained using the aaBCI compared to control decoders trained using supervised learning or using shuffled outputs of the cMTP decoders. Means and standard deviations over the test folds are reported.

		Training method of the control decoder		
		Auto-adaptive training with random output of the cMTP decoder	Auto-adaptive training	Supervised training
Dataset	Runner MI (AUC of ROC)	$0.5007 \pm 0.0691$	$0.6360 \pm 0.0958$	$0.8958 \pm 0.0153$
	Exo (multi-class AUC of ROC)	$0.5163 \pm 0.0580$	$0.7595 \pm 0.0278$	$0.8177 \pm 0.0301$
	Cursor (cosine similarity)	$-0.0231 \pm 0.0327$	$0.1589 \pm 0.0668$	$0.2107 \pm 0.0664$



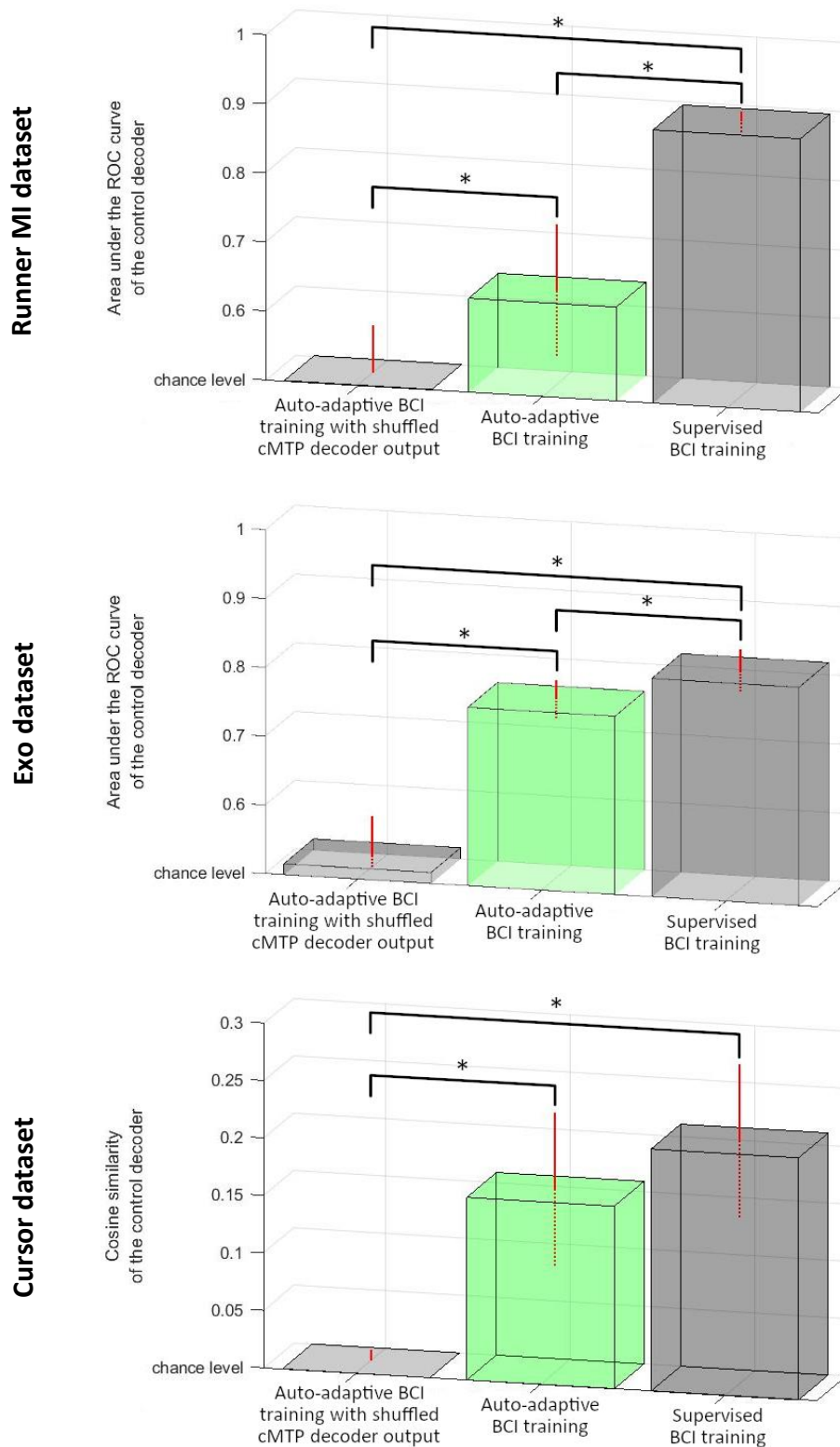


Figure 41 Performances of the control decoders trained using the aaBCI (in green) compared to control decoders trained using supervised learning or using shuffled outputs of the cMTP decoders. Stars denote significant differences between training methods (two-sided Wilcoxon-Mann-Whitney test,  $p$ -value $<0.05$ )

### III. Discussion

In this chapter, we performed a proof of concept of the aaBCI for a binary BCI, a multi-class BCI and a BCI with two continuous outputs. This proof of concept was performed using the neural correlates of cMTP described and detected in Chapter 5, in offline simulations of online use of the BCIs. Using the aaBCI framework, we were able to train control decoders from scratch that performed significantly better than chance level for all 3 types of BCIs. Nevertheless, several points are up to discussion regarding our results.

#### III.1. aaBCI performances

##### *Comparative performances of control decoders trained with the aaBCI*

As mentioned, the control decoders trained using the aaBCI always performed significantly better than chance level. This result is significant, as it validates the proof of concept that the aaBCI framework developed here can be used to train control decoders, even for complex motor BCIs. Expectedly, the control decoders trained using the aaBCI performed worse than the control decoders trained in a supervised manner. Indeed, the noise in the labels used to train the control decoder can be supposed to be much lower in supervised training than in the aaBCI training. Additionally, there is no discarding of data in the supervised training, which means the datasets used to train the control decoders in a supervised manner in our experiments were much larger than the ones used to train the control decoders with the aaBCI. Therefore, there is no surprise with the result obtained where aaBCI-trained control decoders performed worse than supervisedly-trained control decoders.

On a side note, we remind that the noise in the labels used in supervised training cannot be expected to be inexistent either. Indeed, in BCI experiments it is hard to be sure that the subject always performed the desired task, especially in complex motor task. Additionally, we performed our experiments using virtual effectors, which are less prone to noise. With non-virtual effectors, such as an upper-limb exoskeleton, the position of the hand measured by the system is even more prone to noise than with virtual effectors as there are more potential sources of error in the measured position (e.g., deformation of the materials used in the exoskeleton due to the subject's weight).

##### *Impact of the complexity of the motor control task on the accuracy of the estimated labels*

The accuracy of the estimated labels was always above chance level. It can be noted that the accuracy of the estimated labels was similar in the Runner MI and Exo datasets (64.9% and 64.5% respectively), whereas the chance level was much higher in the Runner MI dataset than in the Exo dataset (50% versus 25%). Similarly, even in the complex scenario of estimating labels for a decoder with two continuous outputs (in the Cursor dataset), the accuracy of the estimated labels was still reasonably high (63.3% of the estimated labels were less than 60° away from the actual labels). In our study, the accuracy of the estimated labels was roughly similar for all the datasets. The ability of the cMTP decoder to provide labels for the control decoder seem independent from the complexity of the output of the control decoder.

##### *Long term use of the aaBCI*

The results presented in this study compare aaBCI-trained control decoders and supervisedly-trained control decoder with the premise that the same total amount of data was available in both cases. However, one of the main characteristics of the aaBCI is that it is able to update its control decoder during free use of the BCI by its user, which is not possible with classical supervised training. The aaBCI is thus able to accumulate labeled data indefinitely, albeit with some noise in the labels. Therefore, in the long run the amount of data available for the training of control decoder using the aaBCI is bound to surpass the amount of data that would have been available for supervised training. The performance reported for the control decoders trained with the aaBCI in our study may therefore be

underestimated compared to real use-cases. Additionally, in the long run the performances of supervisedly-trained control decoders are expected to decrease due to concept drifts and non-stationarity in the acquired brain signals (Clerc et al., 2016). The aaBCI framework is theoretically able to overcome this weakness as the control decoders are updated continuously. Therefore, although in our experiments supervisedly-trained control decoders performed better than aaBCI-trained control decoders, we can expect this difference to be lower in actual use-cases of the aaBCI, with the aaBCI possibly outperforming supervisedly-trained control decoders in the long run. Long-term experiments on aaBCIs should be performed in order to get a definitive conclusion on this topic.

### III.2. Parametrization of the aaBCI

#### *Impact of the size of the training dataset on the cMTP decoders' performances*

The amount of data used to train the cMTP decoders in this pseudo-online test of the aaBCI was largely inferior to the amount of data used in the study on detectability of neural correlates of cMTP in the sensorimotor cortex (one third vs four fifth). The pseudo-cross validation scheme used in the aaBCI test corresponds to the full-session cross validation scheme used previously. When compared, there is a decrease in performance (AUC of the cMTP decoder) that averages at -3.3% over the Runner MI, Exo and Cursor datasets. This means that the single trial-detection of neural correlates of cMTP in the sensorimotor cortex requires a substantial amount of data for the training of the decoders. However, it also means that the performance of the aaBCI reported in this chapter could potentially be improved by increasing the amount of data used to train the cMTP decoders.

#### *Alternative threshold selection strategies for data inclusion in the training dataset of the control decoder*

Several strategies for the selection of the  $th_{err}$  and  $th_{corr}$  thresholds are possible. For instance, one could fix arbitrary thresholds on the output of the cMTP decoder. However, this method would not make use of knowledge gained on the training dataset. Another solution we tested was to set  $th_{corr}$  and  $th_{err}$  in order to have given true positive and false positive rates on the training dataset. This method effectively uses the knowledge acquired on the training dataset. However, these thresholds are influenced by the balance between the number of epochs in correct and error class in the training set. Although we perform post-processing in order to have balanced training datasets, we felt that it would be better to have a threshold selection method that can be independent of the data balance. The method used based on the Gaussian modelling of the error and correct class is robust to data imbalance and makes use of the knowledge acquired on the training set. It could however be improved by having an automatic selection of the trade-off parameter  $\alpha$ , for instance using nested cross-validation.

#### *Impact of the trade-off parameter $\alpha$ on the aaBCI*

As the distribution of outputs of the cMTP decoders are not well separated between the error and correct classes (Figure 36, page 67), the amount of data discarded will be substantial even if  $\alpha = 0$ . As expected, the accuracy of the cMTP decoder output on the labeled data goes up as  $\alpha$  increases and the amount of labeled data goes down. This trade-off between amount of data discarded and accuracy of the cMTP decoder on the labeled data directly generates an equivalent trade-off on the trained control decoder. If less data is discarded, the control decoder will be updated at a faster rate (as the control decoder is updated every time 15s of labeled data is acquired), but with more noise in the control decoder labels. Therefore, more general considerations have to be taken into account in order to properly balance the trade-off parameter  $\alpha$ . The rate of update may have a lower bound in order to be able to properly follow concept drifts. Depending on the motor tasks performed with the aaBCI, the control decoder's performance may also have a lower bound. Finally, this trade-off may be subject-dependent and subjective as subjects may prefer faster updates to higher accuracy and vice-versa.

It can be noted that when the general performance of the cMTP decoder (i.e., on the full data split used to train the control decoder, not only on the data included in the control decoder training dataset) was low, the amount of data discarded for a given value of  $\alpha$  was large (Runner dataset: cMTP AUC = 0.5754, 84.5% data discarded). When the general performance of the cMTP decoder was higher, the amount of data discarded was lower (Cursor dataset: cMTP AUC = 0.6211, 74.3% data discarded). Increasing the performance of the cMTP decoder should most likely be the focus if one were to try to improve the performances of the aaBCI.

### III.3. Label estimation for the control decoder

#### *Generality of re-labeling functions*

In our auto-adaptive BCI framework, we defined our relabeling functions as  $\hat{\mathbf{y}}^t_{control} = \varphi(\hat{\mathbf{y}}^t_{cMTP}, \hat{\mathbf{y}}^t_{control})$ . This relationship is defined at the epoch level, which means the estimated control label for epoch  $t$   $\hat{\mathbf{y}}^t_{control}$  is only dependent on the output of the cMTP decoder for epoch  $t$  ( $\hat{\mathbf{y}}^t_{cMTP}$ ) and the output of the control decoder at epoch  $t$  ( $\hat{\mathbf{y}}^t_{control}$ ). However, a more general version of the re-labeling function may be defined as  $\hat{\mathbf{Y}}^u_{control} = \Phi(\hat{\mathbf{Y}}_{mental\ state}, \hat{\mathbf{Y}}_{control})$ . In this relabeling function, the estimated label for the control decoder for epoch  $t$  does not necessarily only depend on the output of the cMTP decoder and control decoder at epoch  $t$ , but on the outputs of these decoders for any epoch available at the time of the update. A straightforward use of this more general relationship would be to smooth the outputs of the cMTP and control decoders and use the same re-labeling functions as previously. But other strategies can also be implemented, such as deciding that the estimated control label for error epochs should be put to the temporally closest control decoder output that was estimated correct. This more general framework may improve the robustness of the estimated labels for the control decoder.

#### *Labeling of epochs when detected erroneous by the cMTP decoder*

However, the main limitation of the re-labeling functions used in this thesis is the handling of epochs labeled as errors by the cMTP decoder. For the binary control decoder (Runner MI dataset), if an output of the control decoder is estimated as erroneous it can only be estimated as the other class of the control decoder. However, that is not the case for the multi-class control decoder (Exo dataset) and the bi-dimensional continuous control decoder (Cursor dataset). When an output of the control decoder is estimated erroneous, multiple (Exo dataset) or an infinite (Cursor dataset) number of other labels are possible for the control decoder. For the Exo dataset, we used the second most probable label as estimated by the output of the control decoder. However, if the control decoder is not good enough, the second most probable output of the control decoder can also be wrong. This strategy could potentially be improved by taking into account the prior probabilities of each class. For the Cursor dataset, we did not use the epochs labeled as erroneous. A possible solution would be to use a stricter definition for erroneous epochs (e.g.,  $\theta_{corr} = 0.5$  and  $\theta_{err} = -0.5$  instead of  $\theta_{corr} = 0.5$  and  $\theta_{err} = -0.1$ ) and then use the opposite of the output of the control decoder when an epoch is estimated erroneous.

In real use-cases of the aaBCI, it would be highly beneficial to always have a solution to use epochs estimated erroneous. Indeed, if the performance of the control decoder were to decrease too sharply, the amount of data added to the training dataset of the control decoder would also decrease, as only epochs estimated as correct are used. This would lead to a diminution of the update rate of the control decoder, as the rate of acquisition of labeled data would be lowered. The worsened update rate could diminish the ability of the aaBCI to follow concept drifts, thus decreasing the performance of the control decoder over time. This vicious circle can be prevented if the aaBCI is able to estimate labels for the control decoder using epochs estimated as erroneous by the cMTP decoder.

*Alternative cost function*

Finally, alternatives to the use of re-labeling functions may be considered. The concept of re-labeling function is based on the idea that it is possible to estimate the correct control decoder label. As discussed, the estimation of the correct label is not straightforward. We considered implementing a cost function that not only punishes predictions far from labels estimated correct, but also punishes predictions too close to labels estimated erroneous (instead of estimating the correct labels for epochs detected erroneous). Although it was not tested, we provide here an example of such a cost function for clarification purposes:

$$cost = \sqrt{\sum_{t \in \{t, \hat{y}_{cMTP}^t > th_{corr}\}} (\hat{y}_{control}^t - \hat{y}_{control}^t)^2} + \sum_{t \in \{t, \hat{y}_{cMTP}^t < th_{err}\}} e^{\frac{-\alpha(\hat{y}_{control}^t - \hat{y}_{control}^t)^2}{r}}$$

with  $\alpha$  a constant that balances the relative impact of the part of the cost function associated to samples estimated correct and the part of the cost function associated to samples estimated erroneous;  $r$  a constant that modulates how fast the cost decrease as predictions get further away from points estimated erroneous. However, such cost function is not convex anymore and is not designed to be optimized with the REW-NPLS or other simple algorithm. This would most likely be the case for any cost function that punishes predictions close to erroneous control decoder outputs. Therefore, these cost functions likely require some computational power in order to be optimized and may not be suited for real-time adaptation. Finally, it should be noted that the knowledge gained by learning not to predict known an erroneous label (e.g., the right part of the cost function proposed here) is higher for control decoder that have few dimensions of output than for control decoders that have a high dimensional output. Making such cost functions work with the aabCI framework would require extensive work and was not done in this thesis.





## Chapter 7: Limitations and perspectives

The concept of auto-adaptive brain-computer interface (aaBCI) was present in the literature long before this thesis. More specifically, aaBCIs were already implemented in online use-cases, by using neural correlates of task performance in order to drive adaptation of the decoders used to control BCIs (control decoders). However, such auto-adaptive BCIs were limited to non-complex communication BCIs such as BCIs with discrete outputs. In this thesis, we argued how the existing aaBCIs are hardly usable for adaptation of complex motor BCIs, which have multiple continuous outputs. Indeed, the event-locked neural correlates of task performance used in these aaBCIs are ill-suited for the labeling of data for decoders with multi-dimensional continuous outputs. Instead, we suggest that continuous-in-time neural correlates of **Motor Task Performance** (cMTP) are preferred for auto-adaptation of complex motor BCIs. The first main result of this thesis was that such neural correlates of cMTP are detectable at the single trial level from the sensorimotor cortex, even when the subject was using motor imagery to control the BCI. The second main result of this thesis was that the output of the cMTP decoder can be used to estimate labels for the control decoder, and that these estimated labels can be used to train control decoders in real-time. Finally, in pseudo-online simulations we managed to use this aaBCI framework to train control decoders with binary outputs, multiple discrete outputs and even bi-dimensional continuous outputs. The proof of concept performed in this thesis lays the foundations of auto-adaptation for complex motor BCIs. However, this proof-of-concept study has several limitations. In this chapter, we discuss such limitations as well as future prospective studies that should be performed to continue this work.

The first limitation is that this study was performed with only one subject. However, this clinical trial is expected to have a total of 5 subjects, who could later be added to this study. Additionally, another clinical trial is being conjointly carried out by EPFL and CEA, for the restoration of walking for people with spinal cord injuries using BCI (STIMO-BSI, ClinicalTrials.gov identifier: NCT04632290). This new clinical trial has a target of 3 subjects, whose data could also be used to validate the findings of this thesis. As the concept of neural correlates of cMTP is not discussed in the literature, it is of prime importance to confirm that such neural correlates can be found reliably across subjects.

A second strong limitation of this study is that the result presented were obtained in a simulation of online use rather than online use itself. Offline simulation studies allow greater parameter exploration and may not capture the full variability of what can occur in online experiments. Our online simulation study was designed to be as close as possible to online use. The cMTP decoders were trained on data from different sessions than the aaBCI-trained control decoders, since in future real case uses the cMTP decoders will require their own supervised training sessions. The control decoder used in the aaBCI tests are compatible with online training. The whole aaBCI test were performed in a way to mimic online conditions. Data was fed chunk by chunk to the training algorithm of the control decoder. None of the steps in the aaBCI framework when performing any update  $u$  required data collected after update  $u$ . The preprocessing and prediction for both the cMTP decoders and control decoders are simple enough to be performed in real time on consumer-grade computers. Strong care was given not to over-optimize parameters on the data acquired, as can easily occur in offline studies, in order to not artificially boost the results presented. However, convincing experts in the scientific community will most likely, and righteously so, require an online proof of concept of the aaBCI using complex motor effectors. Online tests were initiated during this thesis. However, due to world-wide sanitary reasons,

data collection was strongly impeded. Unfortunately, the amount of data collected to this date was not sufficient to conclude the study. The thesis was started late compared to the inclusion date of the subject used. Future subject enrolled in the clinical trial should be able to test out the online version of the aaBCI.

Robustness of the cMTP decoders is a very important feature for the aaBCI. Robustness to time is necessary as else the cMTP decoder would have to be retrained regularly. In this thesis, no specific tests were performed to test the robustness of the cMTP decoder with time. However, as mentioned in “Chapter 5:III.5 Temporal stability of neural correlates of cMTP in the sensorimotor cortex” our result may suggest that the cMTP decoders trained in this thesis were robust in time. Additionally, even if the cMTP decoder is not perfectly robust to time, the aaBCI can still provide some benefits as it only requires the training of a binary decoder (the cMTP decoder) instead of whatever complex control decoder is used by the aaBCI. However, the robustness and AUC of the cMTP decoder must be high enough or else the cMTP decoder may have to be retrained before a significant amount of data has been labeled for the control decoder. In this case, the amount of data needed to retrain the cMTP decoder may be higher than the amount of data added to the training dataset for the control decoder. This point has not been investigated in this thesis, but should be taken into account when designing and testing aaBCIs in future study.

Robustness of the cMTP decoder to change in the BCI paradigm is also desired in the aaBCI framework. Indeed, if cMTP decoders are not robust to task changes, then an additional cMTP decoder would need to be trained for each different control decoder that a BCI uses (as was done in this thesis). This would not prevent the aaBCI framework from being used, but training only one cMTP decoder and being able to use it to train or update any control decoder afterwards would be a strong advantage of the aaBCI. In this thesis we did not perform any specific study to test the robustness of the cMTP decoders to different tasks, like training a cMTP decoder on one of our datasets and testing it on another one, or training a global decoder on all the datasets combined. The strong difference in decoder parameters between the cMTP decoders trained on different datasets suggest that decoders trained on one dataset may not be used as is for cMTP decoding on other datasets. However, the inter-paradigm stability of the cMTP decoders should be explored in detail before conclusions can be drawn.

The proof of concept performed in this thesis was done with relatively simple BCIs. The aaBCI framework described here was shown to work for control decoders with discrete or bi-dimensional continuous outputs. Theoretically, it could also be used as is with control decoders with any number of continuous outputs. Still, experiments are required in order to confirm that the aaBCI framework can be used with control decoders that have three continuous outputs or more, as is the case with the most complex motor state controlled in the current clinical trial (3D movements of the hand).

Finally, the aaBCI framework may have to be adapted for more complex motor BCIs paradigms. For instance, the case of motor BCIs with several motor states where each motor state requires the control of a multi-dimensional effector is not yet possible. The effector controlled in the Exo dataset is close to such a case. It would be if the subject tried to control the trajectory of the hands of the exoskeleton when the ‘right hand’ or ‘left hand’ motor state was activated. In such a scenario, if the output of the cMTP decoder estimates a given epoch as erroneous, it is impossible to know if the error was due to the wrong motor state being activated or the hand trajectory being wrong. One saving point is that if the motor state activated is incorrect, the trajectory is automatically incorrect. Therefore, the control decoders in charge of in-states control can be updated using the aaBCI framework as is. However, it may be hard to estimate if the correct motor state was activated or not. In the future, the aaBCI framework will have to be modified to follow the increasing complexity of state-of-the-art motor BCIs.

It would be beneficial if cMTP neural correlates could be detected from the sensorimotor cortex reliably across users. It would improve usability with current state-of-the-art motor BCI (which mainly have implants in or over the sensorimotor cortex) by not requiring additional recording systems. There is a rationale that can support the hypothesis of a reliable presence of MTP neural correlates in the sensorimotor cortex. Modern computational models of the basal ganglia treat it as a part of a reinforcement-learning network for motor actions: mismatches between predicted movements and the reality are encoded in the spiking rate of midbrain dopaminergic neurons (Bergman et al., 2015). Furthermore, the basal ganglia projects into the primary motor cortex through the thalamus. The role of the basal ganglia in motor adaptation and its projections into the motor cortex could justify the presence of MTP neural correlates in the motor cortex.

Nevertheless, the aaBCI framework developed here could use neural correlates of cMTP regardless of the brain region they were acquired from. It could prove highly beneficial to perform a study to investigate if neural correlates of cMTP can be decoded from other brain regions with more reliability than what was achieved in this thesis. We suggest to try to detect neural correlates of cMTP from the frontocentral and centroparietal cortex, as these brain regions are suitable for the detection of the most studied neural correlate of eMTP, the ErrP. It could be possible that neural correlates of cMTP are also detectable at the single-trial level from these brain regions, and usable for our aaBCI. The fastest way to test this would be to use recording techniques that recorded brain signals from wide spread points over the cortex, such as EEG or MEG. The drawback from these techniques is that they are not well suited to complex motor decoding. As a first approach, it is still possible to try to detect neural correlates of cMTP from other brain regions with these recording techniques. However, aaBCI testing will most likely require data collected using more powerful recording techniques. Unfortunately, ECoG grids that are not implanted specifically for BCI are typically not usable for chronic data recording, and subjects may not be able to attain control of complex motor BCIs in the time available. Therefore, data collected outside of the scope of state-of-the-art motor BCI clinical trial will most likely only be usable for detection of cMTP. Proper validation of the aaBCI framework using neural correlates of cMTP acquired from brain regions other than the sensorimotor cortex will likely require specific clinical trials.

The aaBCI framework has room for a lot of improvement since this thesis was designed as a proof-of-concept study. An avenue for improvement lies in the preprocessing and feature extraction steps of the aaBCI. The preprocessing and feature extraction applied to the input data of the cMTP decoder was based on the existing literature. However, the existing literature was optimized for a different task, which is the detection of neural correlates of eMTP, and either not in the sensorimotor cortex, or in the sensorimotor cortex but not at the single trial level during concurrent use of motor imagery. Therefore, the preprocessing and feature extraction used in this thesis can most likely be refined to be more optimized to the task. It should however be noted that in this study we used the same preprocessing and feature extraction for the cMTP decoder and control decoder, which saves some computation power on the computer running the aaBCI software. Care should be taken to ensure that data processing can still be done in real time when pursuing better preprocessing or feature extraction.

Similarly, the hyper-parameters used were not optimized in this proof of concept, mainly because proper hyper-parameter optimization would require nested cross-validation. Nested cross-validation would both diminish the amount of data used for training and take significantly more time. For instance, the aaBCI results described here may be improved by optimizing the number of factors used in the NPLS when training the cMTP decoder (fixed to 20) or optimizing the data inclusion trade-off parameter  $\alpha$  (fixed to 1).

An improvement can also be made to the way data is included in the training dataset for the control decoder. The re-labeling function makes use of the dual thresholds in order to exclude data which is too uncertain to be estimated as correct or erroneous. This means a lot of data can go unused. Additionally, even the data included in the control decoder training dataset does not have an estimated probability of erroneous or correctness of 1. We could use this estimated probability in order to weight the data included in the control decoder training dataset. Data for which the estimated probabilities of correctness or erroneousness are close to one would weight more than other data, with the weight decreasing all the way to 0 when at the data inclusion thresholds. The uncertainty could also be used in other probabilistic models instead of simple weighting of samples.

On another note, it would be desirable to have a better understanding of neural correlates of cMTPs from a neuroscientific point of view. As cMTP neural correlates were never described before (to our knowledge), there is an inherent interest in understanding their neural basis. Interesting questions include and are not limited to: how are they linked to motor learning? Are they modified when the user is not conscious that he performed a mistake? Are they related to other metacognitive processes? Understanding neural correlates of cMTP may also shine a new light on other brain mechanisms. In addition to the pure neuroscientific interest that neural correlates of cMTPs should elicit, their understanding could also be beneficial for the field of BCIs. For instance, knowing where in the brain these signals are generated and why they are generated would make them easier to use in the aaBCI framework. This is especially true for the optimal location for recording neural correlates of cMTP, as it could be taken into account in future BCI clinical trials when deciding the optimal locations for implants.

Finally, the aaBCI framework was designed for adaptation of the control decoder. It helps preventing future errors from happening. However, neural correlates of task performance can also be used for error correction or instantaneous error prevention (Artusi et al., 2011; Even-Chen et al., 2018; Kreiling et al., 2009) in BCIs based on events. We expect both of these applications to be hard to implement for decoders with multiple continuous outputs, as strong artificial changes in trajectory controls could disturb the user more than it would help him.

The proof of concept realized in this thesis is promising for the future of aaBCI, particularly for complex state-of-the-art motor BCIs. However, in this chapter we outlined several limitations and improvements that could be made to the aaBCI framework. It highlighted that more development will be required before the aaBCI can be used in real life scenarios.

# Supplementary data

## I. Electrode set selection preliminary study

After the visualization study, we tried to spread the recording surface as much as possible over the sensorimotor cortex. We decided to perform preliminary recordings using the two chess-like electrode groups (Figure 13), before selecting the one that provided the best results for single trial detection of eMTPs and cMTPs. This electrode group selection was performed using the first three sessions of the Runner no\_MI dataset and the Runner no\_MI inverted dataset (three sessions as well). We also used the split-session cross validation scheme.

When using event-locked labeling, the mean and standard deviation of the AUC were  $0.771 \pm 0.055$  for the first chess electrode recording layout and  $0.764 \pm 0.070$  for the second chess electrode recording layout. When using continuous in time labeling, the mean and standard deviation of the AUC were  $0.868 \pm 0.056$  for the first chess electrode recording layout and  $0.813 \pm 0.061$  for the second chess electrode recording layout (Figure 42 and Figure 43). There were no statistically significant differences between the two electrode groups regarding the detectability of cMTP neural correlates (two-sided Wilcoxon-Mann-Whitney test,  $p=0.1508$ ) or eMTP neural correlates (two-sided Wilcoxon-Mann-Whitney test,  $p=0.8413$ ). Therefore, we arbitrarily decided to use the first electrode layout throughout the remainder of the thesis.

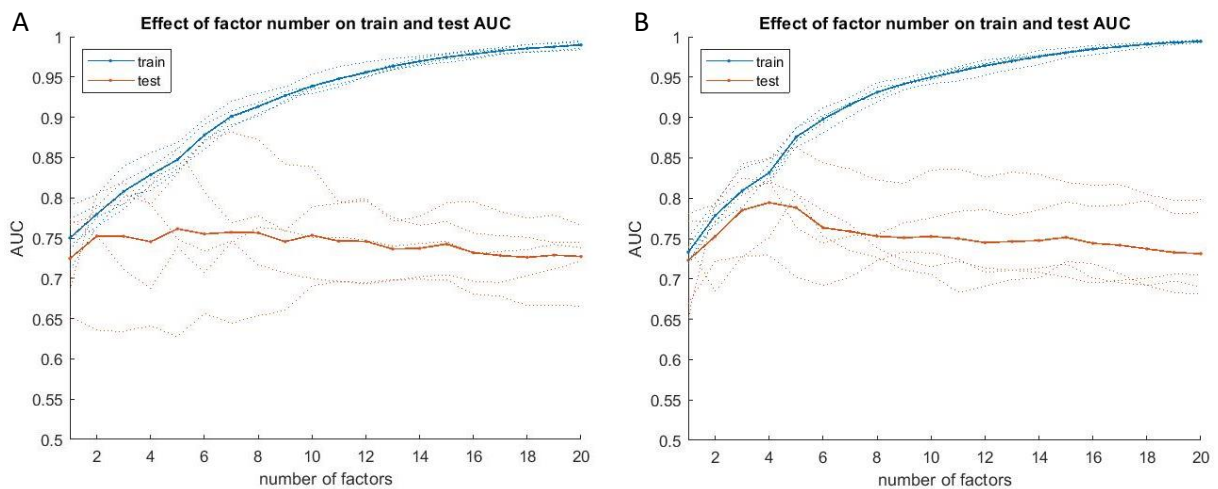


Figure 42. Performance of the eMTP decoder in a preliminary study for electrode selection ((A) electrode group one, (B) electrode group two), performed with the Runner experiment without motor imagery. Dotted lines show the AUC for each test (red lines) or train fold (blue lines), while full lines represent their mean.

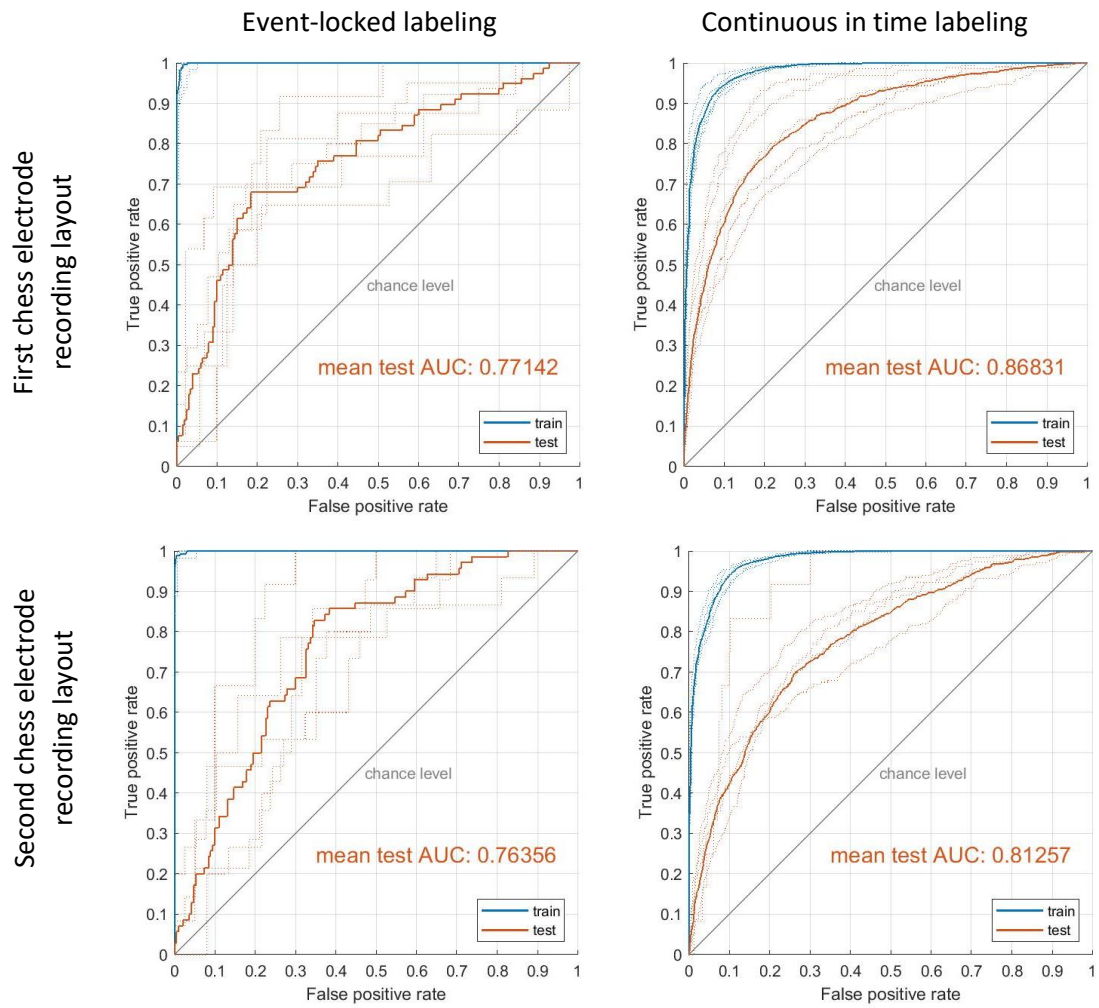


Figure 43 Performances of the eMTP and cMTP decoders for the two different electrode groups in the electrode selection study, and for the two labeling strategies. Performances are displayed using the receiver operating characteristic curves for the detection of erroneous task performance. Performances on the training sets are displayed in blue, while performances on the testing set are displayed in red. Each dotted line represents one of the five cross-validation folds, while the full lines represent the interpolated mean ROC over the five folds.



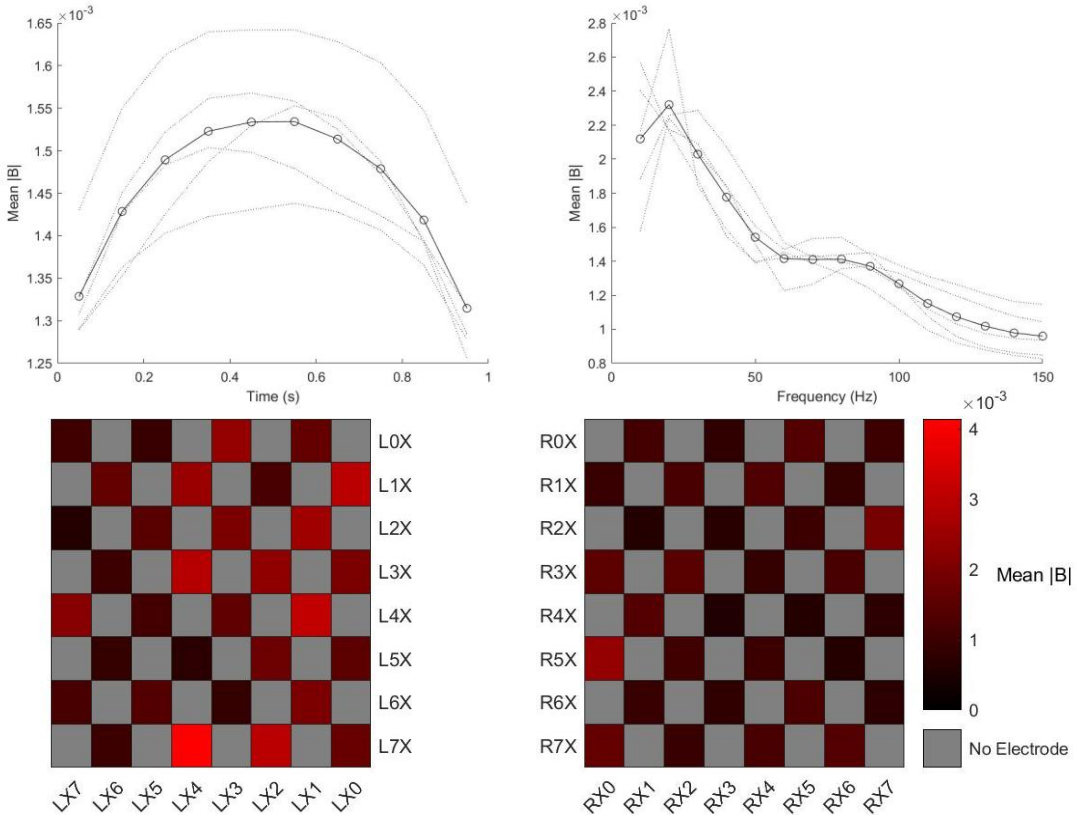
## II. Motor imagery decoders trained on the same epochs as the MTP decoders

In a supplementary study, we investigated the relative feature importance of control decoders. We used the same split-session cross-validation scheme as described in *“Chapter 5:1.5.Performance measure for single trial detection of neural correlates of MTP”*. Control decoders were trained using labels acquired in a supervised manner. However, only epochs that fitted the inclusion criteria of the eMTP or cMTP decoders were used (but with label for the training of the control decoder). This first allowed to study the relative feature importance for the control decoders, but also enabled comparing the feature maps of the control decoder and the cMTP decoder without any bias due to difference in epochs used for the training sets. This supplementary study was performed for the Runner MI, Exo and Cursor dataset. The Runner no\_MI dataset was not included since no motor imagery was performed during the recording of the dataset. For the Runner MI dataset, we present feature maps of control decoders trained using the epochs included in the training sets of either eMTP decoders (event-locked labeling) or cMTP decoders (continuous-in-time labeling). For the Exo and Cursor datasets, the epochs used were the one included in the training set of the cMTP decoder (continuous in time labeling). The decoder parameters were averaged over each modality (temporal, frequential or spatial) as described in *“Chapter 5:1.4.Decoders for the detection of neural correlates of task performance - N-way partial least squares”*.



### Runner MI dataset

Continuous in time labeling



Event-locked labeling

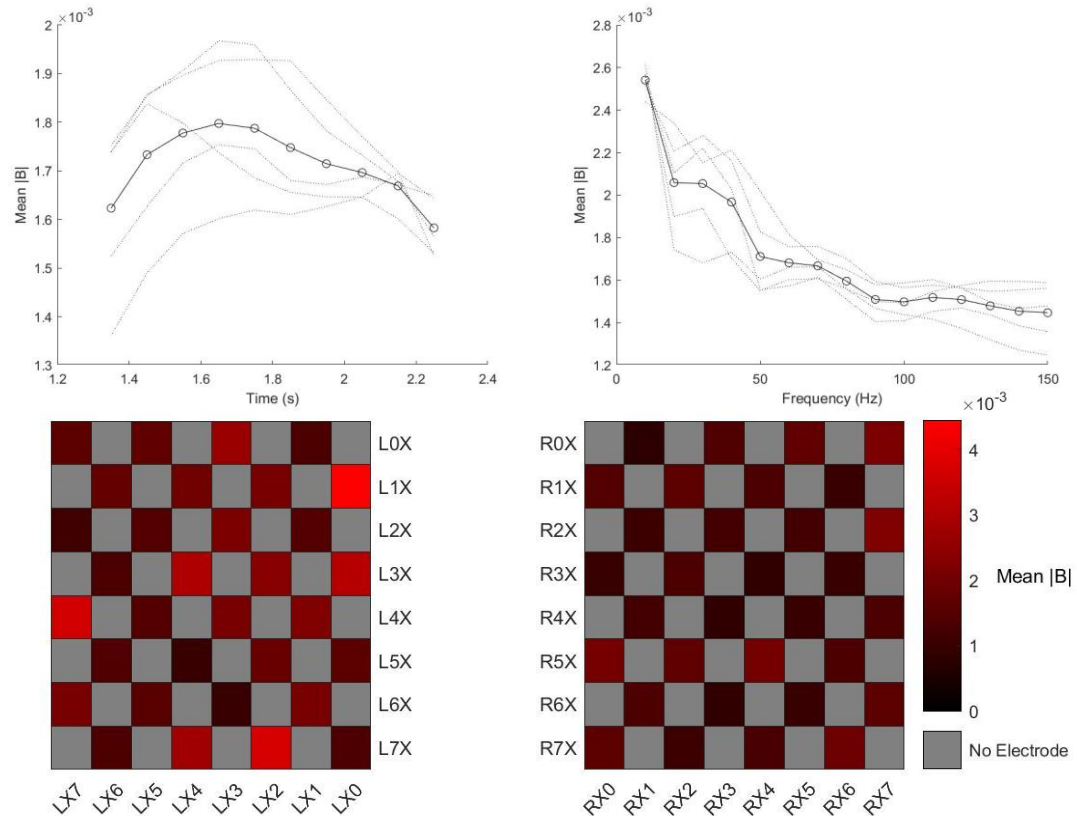
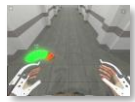


Figure 44 Relative importance of decoder parameters for control decoders trained supervisedly using only epochs that could be included in the training sets of the eMTP decoder (event-locked labeling) or cMTP decoder (continuous-in-time labeling) for the Runner MI dataset.



## Exo dataset

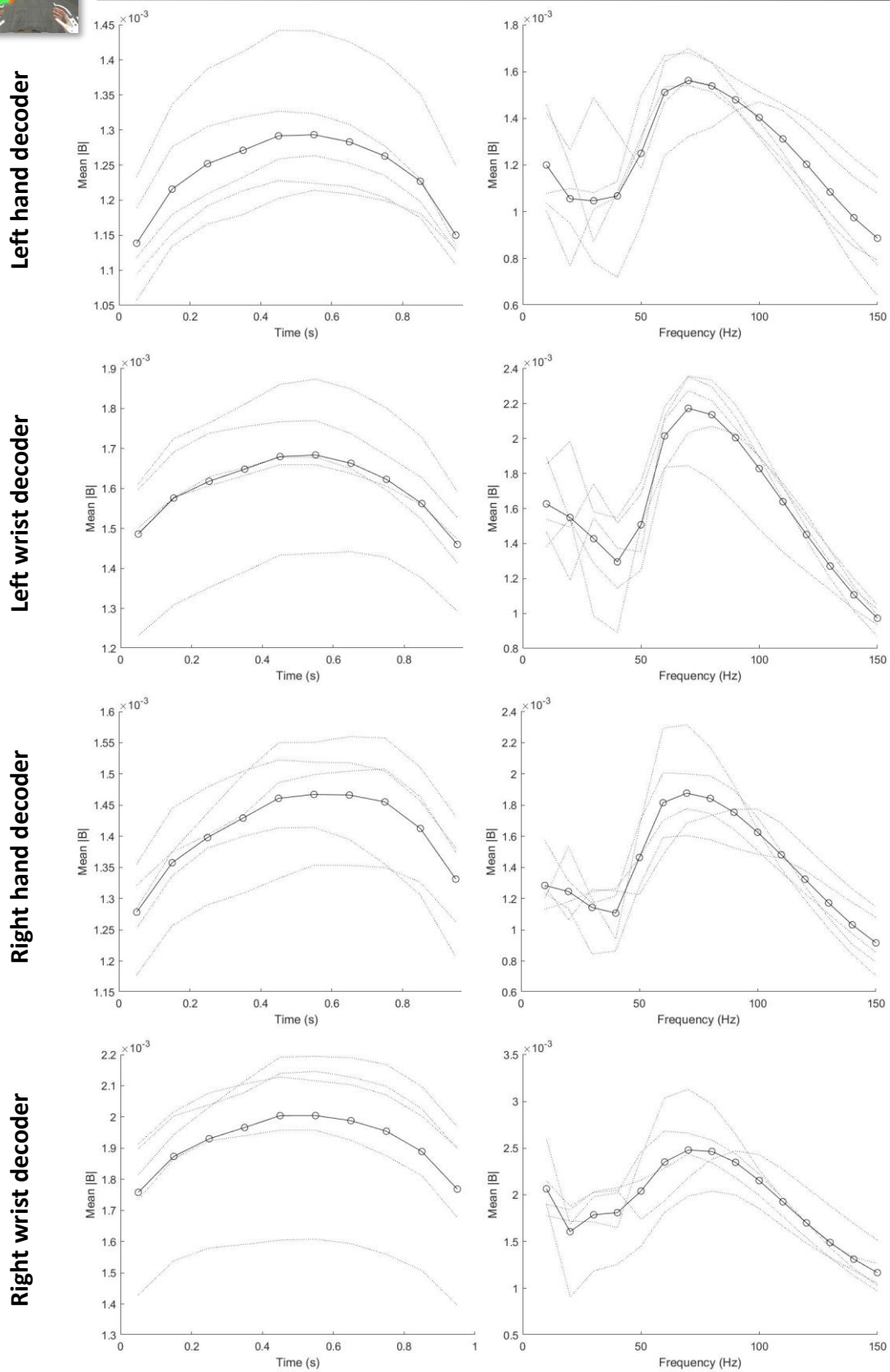


Figure 45 Relative importance of the temporal and frequency decoder parameters for control decoders trained supervisedly using only epochs that could be included in the training sets of the cMTP decoder (continuous-in-time labeling) for the Exo dataset. The decoder parameters are shown for each of the motor classes in the Exo dataset.

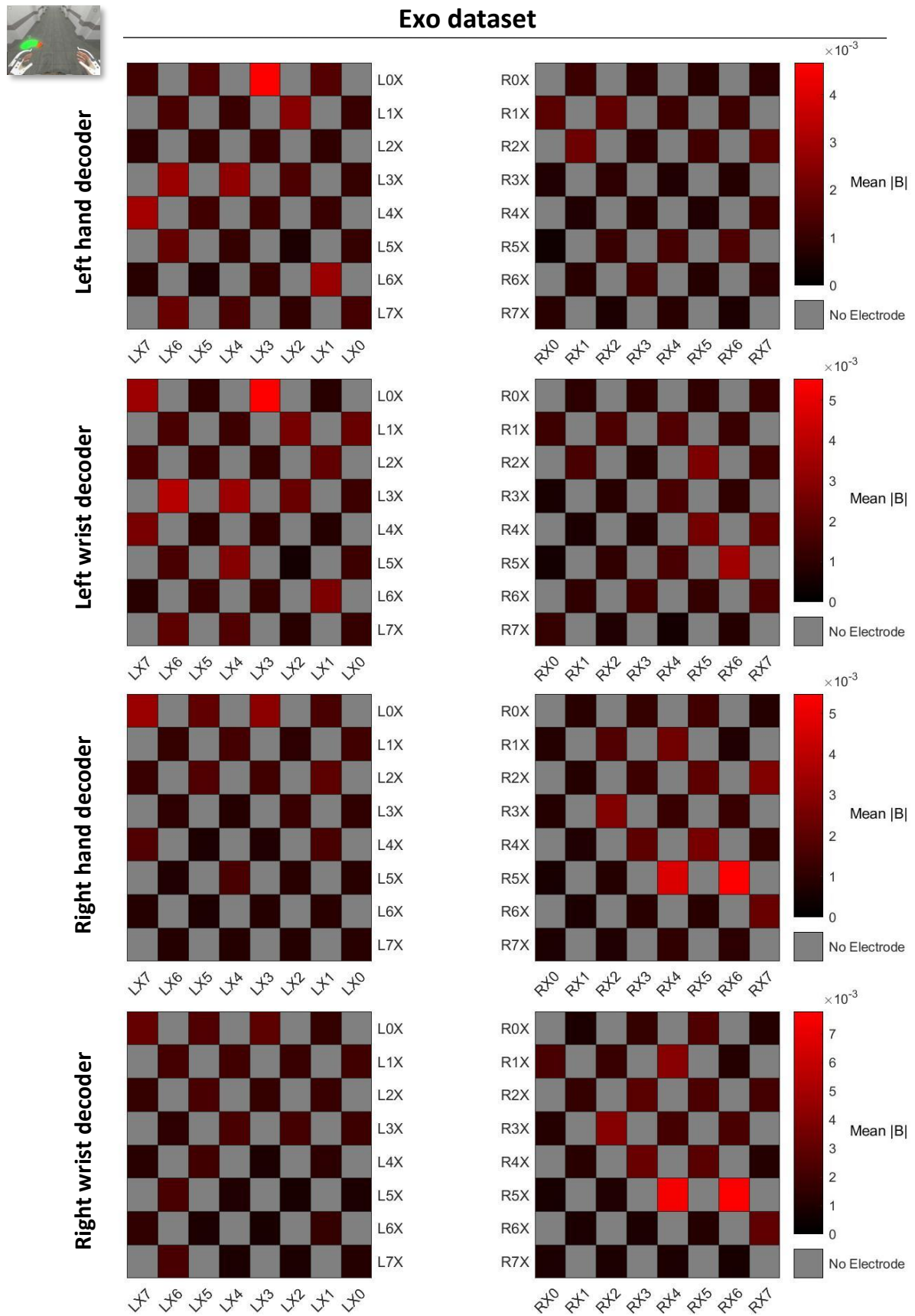


Figure 46 Relative importance of the spatial decoder parameters for control decoders trained supervisedly using only epochs that could be included in the training sets of the cMTP decoder (continuous-in-time labeling) for the Exo dataset. The decoder parameters are shown for each of the motor classes in the Exo dataset.

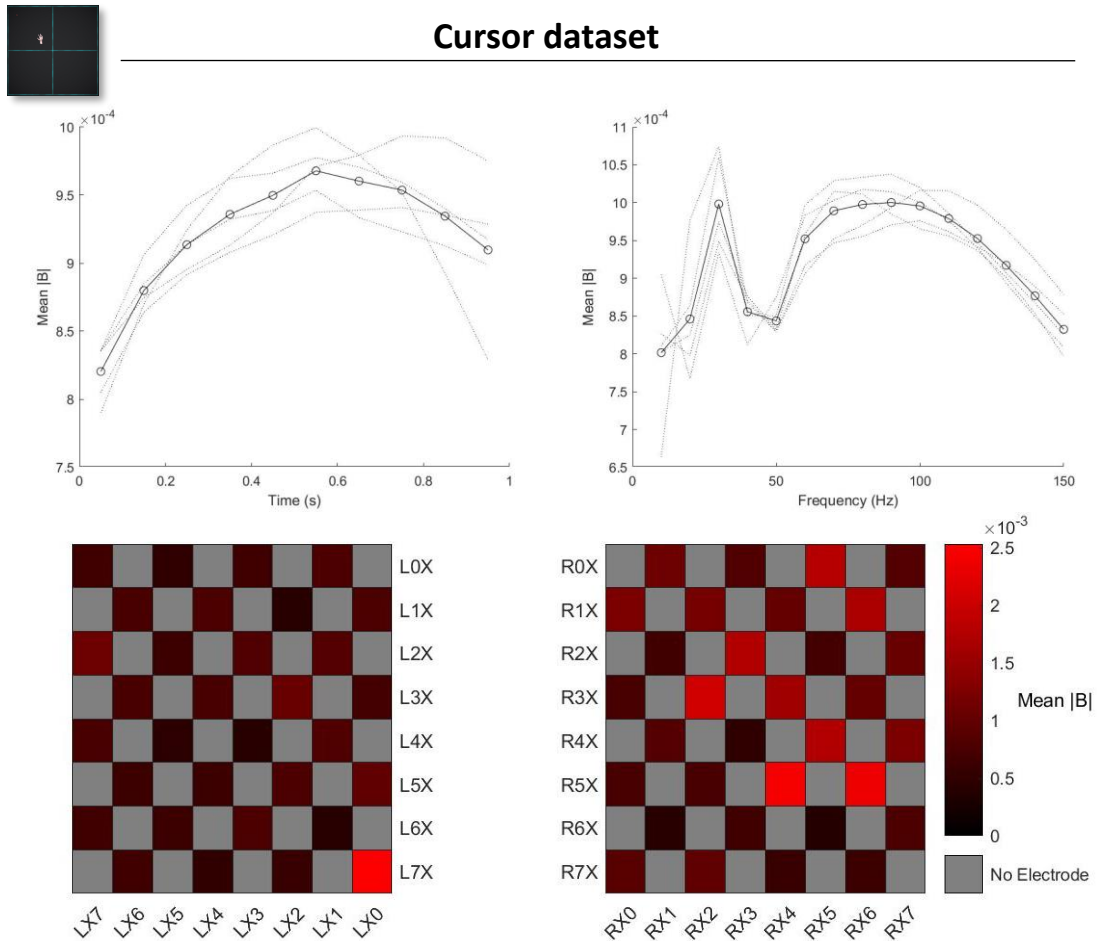


Figure 47 Relative importance of the decoder parameters for control decoders trained supervisedly using only epochs that could be included in the training sets of the cMTP decoder (continuous-in-time labeling) for the Cursor dataset. The decoder parameters shown are the average of the parameters obtained for the two continuous outputs of the control decoder in the Cursor dataset.

# Publications, communications and awards

This thesis led to several scientific contributions, which are listed below. Due to a combination of the communication strategy of the host company (priority on patents) and the worldwide sanitary situation (COVID19), the decision was made to publish one large article at the end of the thesis rather than multiple small ones continuously.

## Patents

Rouanne, V., Aksenova, T., 2022. Méthode d'apprentissage auto-adaptatif d'une interface neuronale directe utilisant une détection physique d'état mental. FR3118413A1.

## Awards

Rouanne, V., Costecalde, T., Martel, F., Karakas, S., Benabid, A.L., Aksenova, T., 2021. Auto-adaptive ECoG-based Brain Machine Interface: training and adaptation during self-directed use. BCI awards 2021 (nominee)

## Publications

Rouanne, V., Costecalde, T., Benabid, A.L., Aksenova, T., 2022. Unsupervised adaptation of an ECoG based brain-computer interface using neural correlates of task performance. *Sci Rep. Under review*

Wei, X., Faisal, A.A., Grosse-Wentrup, M., Gramfort, A., Chevallier, S., Jayaram, V., Jeunet, C., Bakas, S., Ludwig, S., Barmpas, K., Bahri, M., Panagakis, Y., Laskaris, N., Adamos, D.A., Zafeiriou, S., Duong, W.C., Gordon, S.M., Lawhern, V.J., Śliwowski, M., Rouanne, V., Tempczyk, P., 2022. 2021 BEETL Competition: Advancing Transfer Learning for Subject Independence & Heterogenous EEG Data Sets, in: *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*. Presented at the NeurIPS 2021 Competitions and Demonstrations Track, PMLR, pp. 205–219.

Rouanne, V., Śliwowski, M., Costecalde, T., Benabid, A., Aksenova, T., 2021. Detection of Error Correlates in the Motor Cortex in a Long Term Clinical Trial of ECoG based Brain Computer Interface;, in: *Proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies*. Presented at the 14th International Conference on Bio-inspired Systems and Signal Processing, SCITEPRESS - Science and Technology Publications, Online Streaming, pp. 26–34. <https://doi.org/10.5220/0010227800260034>

## Oral presentations and posters

Rouanne, V., Costecalde, T., Benabid, A.L., Aksenova, T., 2021. Auto-adaptive BCI using labels inferred from an ECoG-based continuous cognitive state signal. 9<sup>th</sup> International BCI Meeting.

Rouanne, V., Aksenova, T., 2022. Adaptation of a bi-dimensional continuous BCI using neural correlates of task performance in ECoG recordings from the sensorimotor cortex. *Cortico 2022*



# Bibliography

- Ahn, S., Kim, K., Jun, S.C., 2015. Steady-State Somatosensory Evoked Potential for Brain-Computer Interface-Present and Future. *Front Hum Neurosci* 9, 716. <https://doi.org/10.3389/fnhum.2015.00716>
- Allison, B.Z., Leeb, R., Brunner, C., Müller-Putz, G.R., Bauernfeind, G., Kelly, J.W., Neuper, C., 2012. Toward smarter BCIs: extending BCIs through hybridization and intelligent control. *J Neural Eng* 9, 013001. <https://doi.org/10.1088/1741-2560/9/1/013001>
- Andersson, P., Pluim, J.P.W., Viergever, M.A., Ramsey, N.F., 2013. Navigation of a Telepresence Robot via Covert Visuospatial Attention and Real-Time fMRI. *Brain Topogr* 26, 177–185. <https://doi.org/10.1007/s10548-012-0252-z>
- Artusi, X., 2012. Interface Cerveau Machine avec adaptation automatique à l'utilisateur.
- Artusi, X., Niazi, I.K., Lucas, M., Farina, D., 2011. Performance of a Simulated Adaptive BCI Based on Experimental Classification of Movement-Related and Error Potentials. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 1, 480–488. <https://doi.org/10.1109/JETCAS.2011.2177920>
- Bashashati, A., Fatourech, M., Ward, R.K., Birch, G.E., 2007. 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>
- 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*. [https://doi.org/10.1016/S1474-4422\(19\)30321-7](https://doi.org/10.1016/S1474-4422(19)30321-7)
- Bergman, H., Katabi, S., Slovik, M., Deffains, M., Arkadir, D., Israel, Z., Eitan, R., 2015. Motor Pathways, Basal Ganglia Physiology, and Pathophysiology, in: *Brain Stimulation*. John Wiley & Sons, Ltd, pp. 29–44. <https://doi.org/10.1002/9781118568323.ch3>
- Birbaumer, N., 2006. Breaking the silence: Brain-computer interfaces (BCI) for communication and motor control. *Psychophysiology* 43, 517–532. <https://doi.org/10.1111/j.1469-8986.2006.00456.x>
- Bishop, C., 2006. *Pattern Recognition and Machine Learning*.
- Blumberg, J., Rickert, J., Waldert, S., Schulze-Bonhage, A., Aertsen, A., Mehring, C., 2007. Adaptive Classification for Brain Computer Interfaces, in: *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Presented at the 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 2536–2539. <https://doi.org/10.1109/IEMBS.2007.4352845>
- Bro, R., 1996. Multiway calibration. Multilinear PLS. *Journal of Chemometrics* 10, 47–61. [https://doi.org/10.1002/\(SICI\)1099-128X\(199601\)10:1<47::AID-CEM400>3.0.CO;2-C](https://doi.org/10.1002/(SICI)1099-128X(199601)10:1<47::AID-CEM400>3.0.CO;2-C)
- Buttfield, A., Ferrez, P.W., Del R. Millan, J., 2006. Towards a Robust BCI: Error Potentials and Online Learning. *IEEE Trans. Neural Syst. Rehabil. Eng.* 14, 164–168. <https://doi.org/10.1109/TNSRE.2006.875555>
- 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>
- Chavarriaga, R., Iturrate, I., Millan, J. del R., 2016. Robust, accurate spelling based on error-related potentials.

## Bibliography

- Chavarriaga, R., Millan, J. d R., 2010. Learning From EEG Error-Related Potentials in Noninvasive Brain-Computer Interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 18, 381–388. <https://doi.org/10.1109/TNSRE.2010.2053387>
- Chavarriaga, R., Sobolewski, A., Millan, J. d 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>
- Clerc, M., Daucé, E., Mattout, J., 2016. Adaptive Methods in Machine Learning, in: *Brain–Computer Interfaces 1*. John Wiley & Sons, Ltd, pp. 207–232. <https://doi.org/10.1002/9781119144977.ch10>
- Cohen, O., Koppel, M., Malach, R., Friedman, D., 2014. Controlling an avatar by thought using real-time fMRI. *J Neural Eng* 11, 035006. <https://doi.org/10.1088/1741-2560/11/3/035006>
- Collinger, J.L., Wodlinger, B., Downey, J.E., Wang, W., Tyler-Kabara, E.C., Weber, D.J., McMorland, A.J.C., Velliste, M., Boninger, M.L., Schwartz, A.B., 2013. High-performance neuroprosthetic control by an individual with tetraplegia. *Lancet* 381, 557–564. [https://doi.org/10.1016/S0140-6736\(12\)61816-9](https://doi.org/10.1016/S0140-6736(12)61816-9)
- Coyle, S., Ward, T., Markham, C., McDarby, G., 2004. On the suitability of near-infrared (NIR) systems for next-generation brain–computer interfaces. *Physiol. Meas.* 25, 815–822. <https://doi.org/10.1088/0967-3334/25/4/003>
- Coyle, S.M., Ward, T.E., Markham, C.M., 2007. Brain–computer interface using a simplified functional near-infrared spectroscopy system. *J. Neural Eng.* 4, 219–226. <https://doi.org/10.1088/1741-2560/4/3/007>
- Cruz, A., Pires, G., Nunes, U.J., 2018. Double ErrP Detection for Automatic Error Correction in an ERP-Based BCI Speller. *IEEE Trans Neural Syst Rehabil Eng* 26, 26–36. <https://doi.org/10.1109/TNSRE.2017.2755018>
- Cunningham, J.P., Nuyujukian, P., Gilja, V., Chestek, C.A., Ryu, S.I., Shenoy, K.V., 2011. A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. *J Neurophysiol* 105, 1932–1949. <https://doi.org/10.1152/jn.00503.2010>
- Daucé, E., Proix, T., Ralaivola, L., 2015. Reward-based online learning in non-stationary environments: Adapting a P300-speller with a “backspace” key, in: *2015 International Joint Conference on Neural Networks (IJCNN)*. Presented at the 2015 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. <https://doi.org/10.1109/IJCNN.2015.7280686>
- Degenhart, A.D., Hiremath, S.V., Yang, Y., Foldes, S., Collinger, J.L., Boninger, M., 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>
- Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1–22. <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>
- Donchin, E., Spencer, K.M., Wijesinghe, R., 2000. The mental prosthesis: assessing the speed of a P300-based brain-computer interface. *IEEE Trans Rehabil Eng* 8, 174–179.
- Eliseyev, A., Aksenova, T., 2013. Recursive N-Way Partial Least Squares for Brain-Computer Interface, in: *PloS One*. <https://doi.org/10.1371/journal.pone.0069962>
- 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. *Sci Rep* 7, 1–15. <https://doi.org/10.1038/s41598-017-16579-9>
- 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>
- Falkenstein, M., Hohnsbein, J., Hoormann, J., Blanke, L., 1991. Effects of crossmodal divided attention on late ERP components. II. Error processing in choice reaction tasks. *Electroencephalography and Clinical Neurophysiology* 78, 447–455. [https://doi.org/10.1016/0013-4694\(91\)90062-9](https://doi.org/10.1016/0013-4694(91)90062-9)
- Falkenstein, M., Hoormann, J., Christ, S., Hohnsbein, J., 2000. ERP components on reaction errors and their functional significance: a tutorial. *Biol Psychol* 51, 87–107.
- Ferrez, P., 2007. ERROR-RELATED EEG POTENTIALS IN BRAIN-COMPUTER INTERFACES.
- Ferrez, P.W., del R. Millan, J., 2008. Error-Related EEG Potentials Generated During Simulated Brain–Computer Interaction. *IEEE Trans. Biomed. Eng.* 55, 923–929. <https://doi.org/10.1109/TBME.2007.908083>
- Gu, Z., Yu, Z., Shen, Z., Li, Y., 2013. An Online Semi-supervised Brain–Computer Interface. *IEEE Transactions on Biomedical Engineering* 60, 2614–2623. <https://doi.org/10.1109/TBME.2013.2261994>
- Gueguen, M.C.M., Lopez-Persem, A., Billeke, P., Lachaux, J.-P., Rheims, S., Kahane, P., Minotti, L., David, O., Pessiglione, M., Bastin, J., 2021. Anatomical dissociation of intracerebral signals for reward and punishment prediction errors in humans. *Nature Communications* 12, 1–12. <https://doi.org/10.1038/s41467-021-23704-w>
- Guger, C., Ramoser, H., Pfurtscheller, G., 2000. Real-time EEG analysis with subject-specific spatial patterns for a brain-computer interface (BCI). *IEEE Transactions on Rehabilitation Engineering* 8, 447–456. <https://doi.org/10.1109/86.895947>
- 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>
- Gürel, T., Mehring, C., 2012. Unsupervised Adaptation of Brain-Machine Interface Decoders. *Front Neurosci* 6. <https://doi.org/10.3389/fnins.2012.00164>
- 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>
- Hand, D.J., Till, R.J., 2001. A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning* 45, 171–186. <https://doi.org/10.1023/A:1010920819831>
- Hastie, T., Tibshirani, R., Friedman, J., 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics. Springer, New York, NY.
- Hinterberger, T., Kübler, A., Kaiser, J., Neumann, N., Birbaumer, N., 2003. A brain–computer interface (BCI) for the locked-in: comparison of different EEG classifications for the thought translation device. *Clinical Neurophysiology* 114, 416–425. [https://doi.org/10.1016/S1388-2457\(02\)00411-X](https://doi.org/10.1016/S1388-2457(02)00411-X)
- Hochberg, L.R., Bacher, D., Jarosiewicz, B., Masse, N.Y., Simeral, J.D., Vogel, J., Haddadin, S., Liu, J., Cash, S.S., van der Smagt, P., 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>
- Iwane, F., Chavarriaga, R., Iturrate, I., Millán, J. del R., 2016. Spatial filters yield stable features for error-related potentials across conditions, in: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Presented at the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 000661–000666. <https://doi.org/10.1109/SMC.2016.7844316>
- 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>

## Bibliography

- Jung, J., Jerbi, K., Ossandon, T., Rylvlin, P., Isnard, J., Bertrand, O., Guénot, M., Mauguière, F., Lachaux, J.-P., 2010. Brain responses to success and failure: Direct recordings from human cerebral cortex. *Hum. Brain Mapp.* NA-NA. <https://doi.org/10.1002/hbm.20930>
- Khan, R.A., Naseer, N., Qureshi, N.K., Noori, F.M., Nazeer, H., Khan, M.U., 2018. fNIRS-based Neurobotic Interface for gait rehabilitation. *Journal of NeuroEngineering and Rehabilitation* 15, 7. <https://doi.org/10.1186/s12984-018-0346-2>
- Kim, D.-W., Cho, J.-H., Hwang, H.-J., Lim, J.-H., Im, C.-H., 2011. A vision-free brain-computer interface (BCI) paradigm based on auditory selective attention, 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. 3684–3687. <https://doi.org/10.1109/IEMBS.2011.6090623>
- 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>
- Koelewijn, T., van Schie, H.T., Bekkering, H., Oostenveld, R., Jensen, O., 2008. Motor-cortical beta oscillations are modulated by correctness of observed action. *NeuroImage* 40, 767–775. <https://doi.org/10.1016/j.neuroimage.2007.12.018>
- Kreilinger, A., Neuper, C., Müller-Putz, G.R., 2012. Error potential detection during continuous movement of an artificial arm controlled by brain–computer interface. *Med Biol Eng Comput* 50, 223–230. <https://doi.org/10.1007/s11517-011-0858-4>
- Kreilinger, A., Neuper, C., Pfurtscheller, G., Müller-Putz, G.R., 2009. Implementation of Error Detection into the Graz-Brain-Computer Interface, the Interaction Error Potential. *Assistive technology research* 6.
- Kumar, A., Gao, L., Pirogova, E., Fang, Q., 2019. A Review of Error-Related Potential-Based Brain–Computer Interfaces for Motor Impaired People. *IEEE Access* 7, 142451–142466. <https://doi.org/10.1109/ACCESS.2019.2944067>
- Larzabal, C., Bonnet, S., Costecalde, T., Auboiroux, V., Charvet, G., Chabardes, S., Aksenova, T., Sauter-Starace, F., 2021. Long-term stability of the chronic epidural wireless recorder WIMAGINE in tetraplegic patients. *J. Neural Eng.* 18, 056026. <https://doi.org/10.1088/1741-2552/ac2003>
- 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>
- Lee, W.S., Lee, J.K., Lee, S.A., Kang, J.K., Ko, T.S., 2000. Complications and results of subdural grid electrode implantation in epilepsy surgery. *Surg Neurol* 54, 346–351. [https://doi.org/10.1016/s0090-3019\(00\)00324-4](https://doi.org/10.1016/s0090-3019(00)00324-4)
- Leuthardt, E.C., Miller, K.J., Schalk, G., Rao, R.P.N., Ojemann, J.G., 2006. Electrooculography-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>
- Li, Z., O’Doherty, J.E., Hanson, T.L., Lebedev, M.A., Henriquez, C.S., Nicolelis, M.A.L., 2009. Unscented Kalman Filter for Brain-Machine Interfaces. *PLOS ONE* 4, e6243. <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 Comput* 23, 3162–3204. [https://doi.org/10.1162/NECO\\_a\\_00207](https://doi.org/10.1162/NECO_a_00207)
- Llera, A., Gómez, V., Kappen, H.J., 2012. Adaptive Classification on Brain-Computer Interfaces Using Reinforcement Signals. *Neural Computation* 24, 2900–2923. [https://doi.org/10.1162/NECO\\_a\\_00348](https://doi.org/10.1162/NECO_a_00348)
- Llera, A., van Gerven, M.A.J., Gómez, V., Jensen, O., Kappen, H.J., 2011. On the use of interaction error potentials for adaptive brain computer interfaces. *Neural Networks* 24, 1120–1127. <https://doi.org/10.1016/j.neunet.2011.05.006>

- Lopes Dias, C., Sburlea, A.I., Müller-Putz, G.R., 2018. Masked and unmasked error-related potentials during continuous control and feedback. *Journal of Neural Engineering* 15. <https://doi.org/10.1088/1741-2552/aab806>
- Lopes-Dias, C., Sburlea, A.I., Müller-Putz, G.R., 2019. Online asynchronous decoding of error-related potentials during the continuous control of a robot. *Scientific Reports* 9, 1–9. <https://doi.org/10.1038/s41598-019-54109-x>
- Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., Yger, F., 2018. A review of classification algorithms for EEG-based brain-computer interfaces: A 10 year update. *Journal of Neural Engineering* 15. <https://doi.org/10.1088/1741-2552/aab2f2>
- Mattout, J., Perrin, M., Bertrand, O., Maby, E., 2015. Improving BCI performance through co-adaptation: Applications to the P300-speller. *Annals of Physical and Rehabilitation Medicine* 58, 23–28. <https://doi.org/10.1016/j.rehab.2014.10.006>
- McFarland, D.J., Wolpaw, J.R., 2018. Brain–computer interface use is a skill that user and system acquire together. *PLOS Biology* 16, e2006719. <https://doi.org/10.1371/journal.pbio.2006719>
- 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., Ball, T., Schulze-Bonhage, A., Aertsen, A., Mehring, C., 2012. Error-related electrocorticographic activity in humans during continuous movements. *J. Neural Eng.* 9, 026007. <https://doi.org/10.1088/1741-2560/9/2/026007>
- Miltner, W.H.R., Braun, C.H., Coles, M.G.H., 1997. Event-Related Brain Potentials Following Incorrect Feedback in a Time-Estimation Task: Evidence for a “Generic” Neural System for Error Detection. *Journal of Cognitive Neuroscience* 9, 788–798. <https://doi.org/10.1162/jocn.1997.9.6.788>
- Moly, A., Costecalde, T., Martel, F., Martin, M., Larzabal, C., Karakas, S., Verney, A., Charvet, G., Chabardes, S., Benabid, A.L., Aksenova, T., 2022. An adaptive closed-loop ECoG decoder for long-term and stable bimanual control of an exoskeleton by a tetraplegic. *J. Neural Eng.* 19, 026021. <https://doi.org/10.1088/1741-2552/ac59a0>
- Müller-Putz, G.R., Schwarz, A., Pereira, J., Ofner, P., 2016. From classic motor imagery to complex movement intention decoding: The noninvasive Graz-BCI approach. *Prog. Brain Res.* 228, 39–70. <https://doi.org/10.1016/bs.pbr.2016.04.017>
- Naseer, N., Hong, K.-S., 2015. fNIRS-based brain-computer interfaces: a review. *Front Hum Neurosci* 9, 3. <https://doi.org/10.3389/fnhum.2015.00003>
- Nicolas-Alonso, L.F., Gomez-Gil, J., 2012. Brain computer interfaces, a review. *Sensors (Basel)* 12, 1211–1279. <https://doi.org/10.3390/s120201211>
- Olvet, D.M., Hajcak, G., 2009. Reliability of error-related brain activity. *Brain Research* 1284, 89–99. <https://doi.org/10.1016/j.brainres.2009.05.079>
- Omedes, J., Iturrate, I., Minguez, J., Montesano, L., 2015. Analysis and asynchronous detection of gradually unfolding errors during monitoring tasks. *Journal of neural engineering* 12, 056001. <https://doi.org/10.1088/1741-2560/12/5/056001>
- Omedes, J., Iturrate, I., Montesano, L., Minguez, J., 2013. Using frequency-domain features for the generalization of EEG error-related potentials among different tasks, 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. 5263–5266. <https://doi.org/10.1109/EMBC.2013.6610736>
- Parra, L.C., Spence, C.D., Gerson, A.D., Sajda, P., 2003. Response error correction—a demonstration of improved human-machine performance using real-time EEG monitoring. *IEEE Transactions on*

## Bibliography

- Neural Systems and Rehabilitation Engineering 11, 173–177. <https://doi.org/10.1109/TNSRE.2003.814446>
- Perdikis, S., Millan, J. del R., 2020. Brain-Machine Interfaces: A Tale of Two Learners. *IEEE Systems, Man, and Cybernetics Magazine* 6, 12–19. <https://doi.org/10.1109/MSMC.2019.2958200>
- 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>
- Reece, J.B., 2011. *Campbell Biology*. Benjamin Cummings / Pearson.
- Rezeika, A., Benda, M., Stawicki, P., Gembler, F., Saboor, A., Volosyak, I., 2018. Brain-Computer Interface Spellers: A Review. *Brain Sci* 8. <https://doi.org/10.3390/brainsci8040057>
- Riesel, A., Weinberg, A., Endrass, T., Meyer, A., Hajcak, G., 2013. The ERN is the ERN is the ERN? Convergent validity of error-related brain activity across different tasks. *Biological Psychology* 93, 377–385. <https://doi.org/10.1016/j.biopsycho.2013.04.007>
- Schaeffer, M.-C., 2018. ECoG signal processing for Brain Computer Interface with multiple degrees of freedom for clinical application.
- Schaeffer, M.-C., Aksenova, T., 2018. Data-Driven Transducer Design and Identification for Internally-Paced Motor Brain Computer Interfaces: A Review. *Front. Neurosci.* 12. <https://doi.org/10.3389/fnins.2018.00540>
- Schaeffer, M.-C., Aksenova, T., 2016. Switching Markov decoders for asynchronous trajectory reconstruction from ECoG signals in monkeys for BCI applications. *Journal of Physiology-Paris* 110, 348–360. <https://doi.org/10.1016/j.jphysparis.2017.03.002>
- 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, 75–84. <https://doi.org/10.1088/1741-2560/5/1/008>
- Schalk, G., Wolpaw, J.R., McFarland, D.J., Pfurtscheller, G., 2000. EEG-based communication: presence of an error potential. *Clin Neurophysiol* 111, 2138–2144. [https://doi.org/10.1016/s1388-2457\(00\)00457-0](https://doi.org/10.1016/s1388-2457(00)00457-0)
- Schie, H.T. van, Mars, R.B., Coles, M.G.H., Bekkering, H., 2004. Modulation of activity in medial frontal and motor cortices during error observation. *Nat Neurosci* 7, 549–554. <https://doi.org/10.1038/nn1239>
- Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C., 2001. Estimating the Support of a High-Dimensional Distribution. *Neural Computation* 13, 1443–1471. <https://doi.org/10.1162/089976601750264965>
- 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, 1–10. <https://doi.org/10.1038/ncomms13825>
- Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P.N., Müller, K.-R., 2006. Towards adaptive classification for BCI. *J. Neural Eng.* 3, R13–R23. <https://doi.org/10.1088/1741-2560/3/1/R02>
- Sirpal, P., Kassab, A., Pouliot, P., Nguyen, D.K., Lesage, F., 2019. fNIRS improves seizure detection in multimodal EEG-fNIRS recordings. *J Biomed Opt* 24, 1–9. <https://doi.org/10.1117/1.JBO.24.5.051408>
- Spüler, M., Niethammer, C., 2015. Error-related potentials during continuous feedback: using EEG to detect errors of different type and severity. *Front. Hum. Neurosci.* 9. <https://doi.org/10.3389/fnhum.2015.00155>
- Spüler, M., Rosenstiel, W., Bogdan, M., 2012. Online Adaptation of a c-VEP Brain-Computer Interface(BCI) Based on Error-Related Potentials and Unsupervised Learning. *PLoS ONE* 7, e51077. <https://doi.org/10.1371/journal.pone.0051077>
- 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](https://doi.org/10.1016/S0079-6123(09)17521-0)



- Stippich, C., Ochmann, H., Sartor, K., 2002. Somatotopic mapping of the human primary sensorimotor cortex during motor imagery and motor execution by functional magnetic resonance imaging. *Neuroscience Letters* 331, 50–54. [https://doi.org/10.1016/S0304-3940\(02\)00826-1](https://doi.org/10.1016/S0304-3940(02)00826-1)
- Tanaka, K., Matsunaga, K., Wang, H.O., 2005. Electroencephalogram-based control of an electric wheelchair. *IEEE Transactions on Robotics* 21, 762–766. <https://doi.org/10.1109/TRO.2004.842350>
- Tidoni, E., Gergondet, P., Kheddar, A., Aglioti, S.M., 2014. Audio-visual feedback improves the BCI performance in the navigational control of a humanoid robot. *Frontiers in Neurorobotics* 8.
- Treder, M.S., Blankertz, B., 2010. (C)overt attention and visual speller design in an ERP-based brain-computer interface. *Behav Brain Funct* 6, 28. <https://doi.org/10.1186/1744-9081-6-28>
- Treder, M.S., Schmidt, N.M., Blankertz, B., 2011. Gaze-independent brain-computer interfaces based on covert attention and feature attention. *J Neural Eng* 8, 066003. <https://doi.org/10.1088/1741-2560/8/6/066003>
- 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>
- Vidaurre, C., Kawanabe, M., Büna, P. von, Blankertz, B., Müller, K.R., 2011a. 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., Sannelli, C., Müller, K.-R., Blankertz, B., 2011b. Machine-learning-based coadaptive calibration for brain-computer interfaces. *Neural Comput* 23, 791–816. [https://doi.org/10.1162/NECO\\_a\\_00089](https://doi.org/10.1162/NECO_a_00089)
- Völker, M., Fiederer, L.D.J., Berberich, S., Hammer, J., Behncke, J., Kršek, P., Tomášek, M., Marusič, P., Reinacher, P.C., Coenen, V.A., Helias, M., Schulze-Bonhage, A., Burgard, W., Ball, T., 2018. The dynamics of error processing in the human brain as reflected by high-gamma activity in noninvasive and intracranial EEG. *NeuroImage* 173, 564–579. <https://doi.org/10.1016/j.neuroimage.2018.01.059>
- Volker, M., Schirrmester, R.T., Fiederer, L.D.J., Burgard, W., Ball, T., 2018. Deep transfer learning for error decoding from non-invasive EEG, in: 2018 6th International Conference on Brain-Computer Interface (BCI). Presented at the 2018 6th International Conference on Brain-Computer Interface (BCI), IEEE, Gangwon, pp. 1–6. <https://doi.org/10.1109/IWW-BCI.2018.8311491>
- 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>
- Weiskopf, N., 2012. Real-time fMRI and its application to neurofeedback. *NeuroImage, 20 YEARS OF fMRI* 62, 682–692. <https://doi.org/10.1016/j.neuroimage.2011.10.009>
- Wheeler, J.J., 2019. Co-adaptation for learning and control of devices. US20190370650A1.
- Willett, F.R., Avansino, D.T., Hochberg, L.R., Henderson, J.M., Shenoy, K.V., 2021. High-performance brain-to-text communication via handwriting. *Nature* 593, 249–254. <https://doi.org/10.1038/s41586-021-03506-2>
- Wilson, N.R., Sarma, D., Wander, J.D., Weaver, K.E., Ojemann, J.G., Rao, R.P.N., 2019. Cortical Topography of Error-Related High-Frequency Potentials During Erroneous Control in a Continuous Control Brain–Computer Interface. *Front. Neurosci.* 13, 502. <https://doi.org/10.3389/fnins.2019.00502>
- Wodlinger, B., Downey, J.E., Tyler-Kabara, E.C., Schwartz, A.B., Boninger, M.L., Collinger, J.L., 2014. 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>
- Wolpaw, J., Wolpaw, E.W., 2012. *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, USA.

## Bibliography

- Yeung, N., Holroyd, C., Cohen, J., 2005. ERP correlates of feedback and reward processing in the presence and absence of response choice [WWW Document]. *Cerebral cortex* (New York, N.Y. : 1991). <https://doi.org/10.1093/cercor/bhh153>
- Yong, X., Fatourechi, M., Ward, R.K., Birch, G.E., 2012. Adaptive classification in a self-paced hybrid brain-computer interface system, 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. 3274–3279. <https://doi.org/10.1109/EMBC.2012.6346664>
- Yousefi, R., Sereshkeh, A.R., Chau, T., 2019. Online detection of error-related potentials in multi-class cognitive task-based BCIs. *Brain-Computer Interfaces* 0, 1–12. <https://doi.org/10.1080/2326263X.2019.1614770>
- Zeyl, T.J., Chau, T., 2014. A case study of linear classifiers adapted using imperfect labels derived from human event-related potentials. *Pattern Recognition Letters, Partially Supervised Learning for Pattern Recognition* 37, 54–62. <https://doi.org/10.1016/j.patrec.2013.05.020>
- Zhu, D., Bieger, J., Garcia Molina, G., Aarts, R.M., 2010. A survey of stimulation methods used in SSVEP-based BCIs. *Comput Intell Neurosci* 702357. <https://doi.org/10.1155/2010/702357>

## **Adaptation of discrete and continuous intracranial Brain-Computer Interfaces using neural correlates of task performance decoded continuously from the sensorimotor cortex of a tetraplegic.**

Brain-computer interfaces (BCIs) transform neural signals into commands for effectors. They are mainly used as tools for functional compensation of impaired functions in disabled subjects. The Clinattec clinical trial “BCI and tetraplegia” aims at providing a proof of concept of long-term functional compensation of upper and lower limbs motor deficits in tetraplegics using a BCI. The clinical trial showed promising results for compensation of motor functions. However, the BCI’s usability remains to be improved. The decoder used to estimate the user’s intention from the processed neural signal (control decoder) must be fitted before the BCI can be used. This is done during dedicated training sessions, during which the user is directed to perform specific motor imagery tasks. Training sessions have to be held regularly in order to update the control decoder due to degradation of performances with time. This thesis proposes to limit the negative impact of training sessions. In this work, the control decoder is trained and updated using inferred labels instead of labels acquired during a dedicated training session. The labels are inferred using the output of the control decoder and neural correlates of task performance. In order to be usable in state-of-the-art motor BCIs, the adaptation process must be possible for control decoders with multiple discrete or continuous outputs. We argue that adaptation of a control decoder with multiple continuous outputs is best done using neural correlates of task performance that can be decoded continuously in time. Additionally, these neural correlates should be detected in the sensorimotor cortex due to the position of the implanted neural acquisition system. Using multiple ECoG datasets from a tetraplegic enrolled in the Clinattec BCI clinical trial, we show in this thesis that it is possible to detect such continuous in time neural correlates of task performance from the sensorimotor cortex, that control decoder labels can be inferred using these neural correlates, and finally that these labels can be used to successfully train decoders for discrete or continuous control. The subject used motor imagery to control a binary avatar (Runner MI dataset), a virtual exoskeleton with four discrete motor states (Exo dataset) or a hand-shaped cursor on a two-dimensional screen (Cursor dataset). In five-fold cross-validations, the mean and standard deviation of the area under the receiver operating characteristic curve (AUC of the ROC) for the decoder of neural correlates of task performance were  $0.6225 \pm 0.0429$  in the Runner MI dataset,  $0.5677 \pm 0.0427$  in the Exo dataset and  $0.6570 \pm 0.0188$  in the Cursor dataset. In a pseudo cross-validation simulating online use, the accuracy of the estimated labels was 64.9% in the Runner dataset and 64.5% in the Exo dataset. In the Cursor dataset, 63.3% of the estimated labels were less than  $60^\circ$  away from the actual labels. The AUC of the ROC of the control decoder was  $0.6360 \pm 0.0958$  in the Runner MI dataset when trained auto-adaptively compared to  $0.8958 \pm 0.0153$  when trained supervisedly, and to a chance level of  $0.5007 \pm 0.0691$ . The multiclass generalization of the AUC of the ROC of the control decoder was  $0.7595 \pm 0.0278$  in the Exo dataset when trained auto-adaptively compared to  $0.8177 \pm 0.0301$  when trained supervisedly, and to a chance level of  $0.5163 \pm 0.0580$ . In the Cursor dataset, the cosine similarity was  $0.1589 \pm 0.0668$  when trained auto-adaptively compared to  $0.2107 \pm 0.0664$  when trained in a classical supervised manner, and a chance level of  $-0.0231 \pm 0.0327$ . These results are promising for the future of auto-adaptive complex motor BCIs.

**Keywords:** Brain Computer interfaces, BCI, ECoG, adaptation, Clinical trial, neural correlates of task performance, self-directed use, tetraplegia, machine learning, Brain signal processing.

## **Adaptation d'Interfaces Cerveau-Machines discrètes et continues grâce à des corrélats neuronaux de performance de tâche détectés continuellement dans le cortex sensorimoteur d'un tétraplégique.**

Les Interfaces cerveau machine (ICMs) transforment les signaux neuronaux en commande pour des effecteurs. Elles sont principalement utilisées pour compenser les déficits d'handicapés moteurs. L'essai clinique « BCI et tétraplégie » de Clinatec a pour but de fournir une preuve de concept de compensation fonctionnelle chronique des déficits moteurs des quatre membres de tétraplégiques au moyen d'ICMs. Bien que les résultats obtenus soient prometteurs, l'usabilité des ICMs utilisées reste améliorable. Le décodeur utilisé pour transformer les signaux neuronaux en commandes (décodeur de contrôle) doit par exemple être entraîné avant de pouvoir utiliser l'ICM. Ceci est fait durant des séances d'entraînement spécifiques, durant lesquelles le patient doit effectuer des tâches d'imagerie motrice imposées. Ces séances d'entraînement constituent un temps mort et doivent être effectuées régulièrement afin de compenser les pertes de performances qui surviennent avec le temps. Cette thèse vise à limiter les effets négatifs des séances d'entraînement. Pour cela, le décodeur de contrôle est entraîné et mis à jour avec des labels estimés au lieu de labels obtenus lors de séances d'entraînement. Les labels sont estimés grâce à la sortie du décodeur de contrôle ainsi qu'un décodeur de corrélats neuronaux de performance de tâche. Afin d'être utilisable avec les ICMs motrices les plus performantes, l'adaptation doit être faisable pour les décodeurs de contrôle qui ont plusieurs sorties continues. Nous suggérons qu'une telle adaptation requiert des corrélats neuronaux de performances de tâche acquis de manière continue dans le temps. Il est aussi nécessaire de détecter ces corrélats dans le cortex sensorimoteur à cause de la position des implants. Dans cette thèse, nous montrons sur plusieurs jeux de données ECoG d'un tétraplégique qu'il est possible de détecter continuellement des corrélats neuronaux de performance de tâche depuis le cortex sensorimoteur, que les labels des décodeurs de contrôle peuvent être estimés grâce à ces corrélats et enfin que ces labels peuvent être utilisés pour l'entraînement de décodeurs avec des sorties discrètes ou continues. Le patient a utilisé l'imagerie motrice pour contrôler un avatar binaire (jeu de données Runner MI), un exosquelette virtuel avec quatre états moteur discrets (jeu de données Exo) ou un curseur en deux dimensions continues sur un écran d'ordinateur (jeu de données Cursor). Lors de validations croisées à 5 blocs, la moyenne et la déviation standard de l'aire sous la courbe (AUC) ROC du décodeur de corrélats neuronaux de performance de tâche était de  $0.6225 \pm 0.0429$ ,  $0.5677 \pm 0.0427$  et  $0.6570 \pm 0.0188$  pour les jeux de données Runner MI, Exo et Cursor respectivement. Dans une simulation d'utilisation en ligne de cette méthode d'adaptation, la précision des labels estimés était de 64.9% et 64.5% pour les jeux de données Runner MI et Exo. Pour le jeu de données Cursor, 63% des labels estimés étaient à moins de  $60^\circ$  d'écart des vrais labels. Dans cette simulation avec le jeu de données Runner MI, l'AUC de la ROC du décodeur de contrôle était de  $0.6360 \pm 0.0958$  quand entraîné de manière auto-adaptative, comparé à  $0.8958 \pm 0.0153$  pour un entraînement supervisé et à un niveau de chance de  $0.5007 \pm 0.0691$ . Pour le jeu de données Exo, la généralisation multi classe de l'AUC de la ROC était  $0.7595 \pm 0.0278$  pour un entraînement auto-adaptative, comparé à  $0.8177 \pm 0.0301$  pour un entraînement supervisé et à un niveau de chance de  $0.5163 \pm 0.0580$ . Pour le jeu de données Cursor, la similarité cosinus était  $0.1589 \pm 0.0668$  pour un entraînement auto-adaptative, comparé à  $0.2107 \pm 0.0664$  pour un entraînement supervisé et à un niveau de chance de  $-0.0231 \pm 0.0327$ . Ces résultats sont prometteurs pour le développement des ICMs motrices complexes auto-adaptatives.

**Mots-clés :** Interface cerveau machine, ICM, ECoG, adaptation, corrélats neuronaux de performance de tâche, utilisation libre, tétraplégie, apprentissage machine, traitement des signaux cérébraux.



