Deep Brain Local Field Potential Neural Decoders for Real-Time Computer Interfacing

Thomas Laurent Christophe Marie Martineau

Department of Mechanical Engineering

Imperial College of Science, Technology and Medicine London, UK

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Supervisor: Prof Ravi Vaidyanathan, Department of Mechanical Engineering, Imperial College London

Co-supervisor: Prof Huiling Tan, MRC Brain Network Dynamics Unit, University of Oxford

Statement of Originality

I hereby certify that all material in this thesis is the result of my own work, and which is not has been acknowledged and appropriately accredited.

PUBLISHED BY THE IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

First Edition, April 2023

Acknowledgment

I would like to first thank the Tan Group from the MRC Brain Network Dynamics at the University of Oxford for their continued collaboration throughout my doctoral work, Prof Huiling Tan and Dr Shenghong He, for sharing their data with me and helping me with my own data gathering. I wish to recognise my colleague, Nathan Steadman, for providing me with his recording software and support. I would also like to thank my supervisors, Prof Ravi Vaidyanathan and Prof Huiling Tan, for their advice, feedback, and constant encouragement, especially in these last years, which have been collectively difficult.

Last and not least, I would like to thank my family for their unconditional support; my parents, who both inspired me to undertake a career in science and technology; my siblings, of whom I am immensely proud and look up to; and my partner who has always stood by me during all these years of doctoral studies.

I dedicate this work to my grandparents, Danièle, Martine, Jacques, et Jean, who have always championed my education and studies.

Abstract

The progressive adoption of deep brain stimulation (DBS) therapy to treat neurological disorders offers a unique opportunity to research a new class of invasive brain-computer interface (BCI). DBS electrodes can record electrophysiological activity or local-field potentials (LFP) within the subcortical structures, particularly the subthalamic nucleus (STN). Although STN-LFP biomarkers have been correlated to multiple motor processes, including motion, force generation, action sequencing, and gait, subcortical structures have not received nearly as much attention as other motor cortical areas. Instantaneous STN interfacing could inform the control of a wide range of medical devices and possibly neuroprosthetics.

This work provides evidence that STN-LFP decoders can predict volitional motor states and presents solutions tested on patients' recordings undergoing DBS surgery. Decoding needed to be further validated under real-time constraints. Part of this work was dedicated to implementing, testing, and comparing compatible algorithms, particularly for improved time-frequency feature extraction across multiple signal channels. Decoders were customised for different decoding tasks, and parameter tuning methods were also introduced. Subsequent experiment results showed that these strategies helped increase performance.

The first experiment decoded discrete motor states from STN-LFP to achieve action detection and left vs right-hand selection. Results showed that those motor state variables could be predicted under asynchronous and real-time constraints (median 0.7-0.3 action detection rate and 0.7 left vs right classification accuracy). However, performance was very variable across study participants. Evidence pointed to an unequal signal-to-noise ratio of STN-LFP biomarkers in different recordings, which correlated directly to individual performance ceilings.

The second set of BCI tasks focused on continuous state decoding for fine motor control. New force tracking paradigms revealed new association of biomarkers with motor states of different dynamical orders. Some decoding algorithms showed favourable results.

This work concludes that STN-LFP neural decoders could have promising applications, given some adaptations and further innovation.

Table of Contents

List of Tables

List of Figures

Nomenclature

Acronyms

Anatomical and Biological Terms

CNS	Central nervous system
BG	Basal ganglia
CN	Caudate Nucleus
CBGTC	Cortical basal ganglia thalamic cortical loop
GPe	Globus pallidus external
GPi	Globus pallidus internal
MVC	Maximum voluntary contraction
PD	Parkinson's disease
PNS	Peripheral nervous system
PT	Putamen
SNr	Substantia nigra
STN	Subthalamic nucleus

Electrophysiological Terms

BCI	Brain-computer interface
DBS	Deep brain stimulation
EEG	Electroencephalography
ECoG	Electrocorticographic grids
EMG	Electromyography
ERD	Event-related desynchronisation
ERS	Event-related synchronisation
LFP	Local field potentials
MEA	Micro-electrode array

Signal Processing and Feature Extraction Terms

ADC	Analogue-to-digital conversion
AM	Amplitude modulated signal
AR	Autoregressive
MA	Moving-average
CHS	Channel selection

CSP	Common spatial pattern
CWT	Continuous wavelet transform
DWT	Discrete wavelet transform
FFT	Fast Fourier transform
FIR	Finite impulse response
FM	Frequency modulated signal
FWHM	Full width at half-maximum of a wavelet function
HJ	Hjorth parameters
IIR	Infinite impulse response
QMF	Quadrature mirror filter
RLS	Recursive least squares
RWT	Recursive wavelet transform
SF	Spatial filter
SNR	Signal-to-noise ratio
STFT	Short-time Fourier transform
TF	Time frequency

Optimisation, State Estimation and Machine Learning

BO	Bayesian optimisation
ENR	Elastic-network regression
GBTR	Gradient-boosted trees regression
HMM	Hidden Markov model
KNN	k-nearest neighbours
KNR	k-nearest regression
LDA	Linear discriminant analysis
LR	Logistic regression
NB	Naïve Bayes
PCA	Principal component analysis
RBF	Radial-basis function kernel
ROC	Receiver-operator characteristic
RR	Ridge regression
SFA	Slow feature analysis
SQFE	Sequential feature selection
SVM	Support-vector machine
SVR	Support-vector regression

Other

CPU	Central Processing Unit

DC	Direct current
GPU	Graphical processing unit
IC	Integrated circuits
MCU	Micro-controller unit
ANOVA	Analysis of Variance
IQR	Inter-quartile range

Mathematical Symbols

$\delta, \theta, \alpha, \beta, \gamma$	Specific LFP frequency bands

Discrete Signal

k	Discrete time step
$x[k]$	LFP or LFP feature signal array
z	Memory state
$\bar{x}[k]$	Trial average
$X[k, l]$	Feature array
$\chi[k, \dots, l]$	Time-frequency array
$y[k]$	Motor state signal array
$\tilde{y}[k]$	Decoded motor state signal array
τ_y	Target-lead time
$\theta[k]$	ARMA coefficient array
$p[k]$	Probability signal array
$u[k]$	Tracker signal array
$e[k]$	Error signal array
f_s	Signal sampling frequency
N_k	Temporal length of array
N_{ch}	Total channel number
N_c	Number of selected channels or components
N_{trial}	Number of trials
N_{bd}	Number of bands
N_{AR}	Number of autoregressive coefficients
N_{MA}	Number of moving-average coefficients
N_l	Number of frequency bases, wavelets, or decomposition levels
N_w	Size of epoch length
N_f	Number of features

N_y	Number of states
N_{SV}	Number of support-vectors

Pipeline Description

f_i	Local fitting function
h_i	Stage transform
u_i	Update function
w_i	Local parameters (or weights)
g	Gradient descent algorithm
ρ	Hyperparameter vector
$\mathcal{S}_\rho$	Hyperparameter search space
μ, σ	Mean and standard deviation of a signal
$\mathrm{MED}, \mathrm{MAD}$	Median and mean absolute deviation operators
S	Task performance scoring function
s	Score scalar
r_o	Overfitting ratio

Signal Processing

a, b	Recursive and forward filter arrays
$\psi(t)$	Mother wavelet function
R	Downsampling ratio
L	Feature extraction length
H_1, H_2, H_3	Hjorth parameters
P	Power operator
τ_d	Decoder analysis window

BO-Tuned Parameters

β_K	Kaiser window parameter
Θ_1, Θ_2	QMF filter parameters
η	Learning rate for RLS algorithm and GBTR classifiers
n_o	Number of overlaps between successive extraction windows
λ	Box-Cox parameter
r_σ	Second order Kalman filter process to observation noise ratio
r_c	Feature array compression ratio
α_{L_2}	L_2-norm (Tikhonov) regularisation
r_{L_1}	Ratio of L_1 to L_2-norm regularisation
T	Threshold

r_T	Threshold deadband ratio
ΔT	Threshold deadband
C	SVM and LR regularisation constant
γ_C	LDA shrinkage and SVM kernel parameter
n_k	Number of k-neighbours
n_d	GBTR maximum depth
n_e	GBTR number of estimators
r_s	GBTR subsample ratio

Metrics

M	Metric function measuring the relationship strength between feature and motor state
MI	Mutual information
CS	Class separability
ISF	Inter-class separability F-score
MR	Multiple regressions
FPR	False-positive rate
TPR	True-positive rate
GM	Geometric mean between sensitivity and specificity
ACC	Accuracy
r^2	Coefficient of determination
MSE	Mean square error

Other

t	Time
ω	Normalised frequency
Ω	Normalised frequency domain
Ω_i	Sub-band frequency sub-domain
S_w, S_b	Scattered within-class matrix and scattered between-class matrix
C_1,C_2	Class covariance matrices
c	Correlation vector
S_X	Correlation Matrix
C_X	Covariance Matrix
τ_L	Ramp holding time
τ_c	Chirp ramping time
K_{SVM}	SVM kernel function
$\mathcal{D}$	Distance function

Chapter 1. Introduction

Most of us take for granted the natural ability to control our mind and body, choose our movements or experience the full extent of our senses. Regaining a fraction of this control for those who have lost it because of irreversible physical traumas or degenerative neuronal diseases, even if momentarily, can be a blessing. Devices such as powered exoskeletons, robotic prosthetic limbs, or neural stimulators (Figure 1.1a) are being actively researched and developed for these purposes: rehabilitate, restore or augment human natural motor functions. Informing the control of such devices to achieve any of these goals is a demanding problem. There are many ways these devices can interact with their users and the concurrent operating environment. Finding and exploiting the correct interaction modality is critical in ensuring that a device is cognisant of its users' intentions. If correctly interpreted, the device can physically assist, correct, become a remote extension to one's own body, or deliver therapeutic stimulation in a manner that is, above everything else, helpful, and not impeding.

A. User-Device Interaction: Promises of an Invasive BCI

Pons (2008) identifies two principal sub-classes of user-device interactions: physical and cognitive (Figure 1.1b).

Physical interactions can be measured through kinematic and kinetic signals acting at distal end-effector points or at the limb joints by compact sensors such as inertial measurement units (Huo et al., 2020), flexible goniometers or force sensors (Ding et al., 2014). These sensors can be installed in clothing, accessories such as smartwatches (Cagnan et al., 2019; Powers et al., 2021) or even directly embedded within an implanted therapeutic device, such as a neural stimulator unit (Rouse et al., 2011). Physical signals are contextual and do not provide direct information on motor intentions; those must be instead forecasted based on past observations and models, such as a gait stage predictor (Pons Jose, 2008). Although this is rapidly evolving, human users remain better at perceiving, planning, and executing motion than machines. Therefore, it may still be beneficial to directly measure the users' intended actions rather than guessing them and running the risk to interfere with them (Pons Jose, 2008).

Cognitive interaction involves communication with the peripheral (PNS) or central nervous (CNS) systems. Unlike physical interactions, cognitive state decoding does not suffer from mechanical latency, and it should, in theory, empower users with more intuitive control over the target device (Pons Jose, 2008). For neuro-rehabilitative or therapeutic applications, specific cognitive and pathological biomarkers should also be monitored (Daly and Wolpaw, 2008; Little et al., 2013; Pons Jose, 2008).

Progress in measuring peripheral activity has been significant. For instance, spinal neuronal activity can be estimated from various bio-signals, primarily electromyography (EMG) (Merletti et al., 2008) and translated into intended motor action. However, interfacing with the PNS is impossible when the spinal tracks are physically severed (Capogrosso et al., 2016) or compromised by a neuro

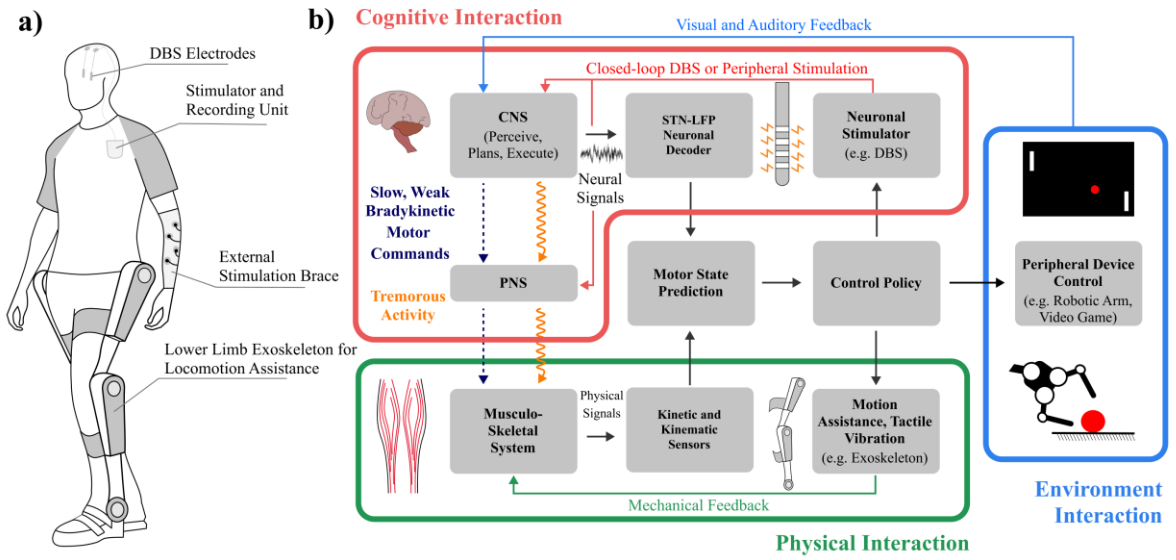


Figure 1.1 – Illustration presenting different examples of user-device interaction loops; **a)** Assistive, therapeutic and monitoring devices, including a DBS neural decoder communicating to an external stimulator and exoskeleton **b)** Different user-device interaction loops; Voluntary motor commands originate in the CNS, after perception and planning; They travel down the PNS to the musculoskeletal system; In PD, these commands are compromised by involuntary tremorous activity or by improper generation, leading to weak, slow and inconsistent bradykinetic motion; Neural signals are measured through the DBS probe and processed; Decoded motor states from the neural interface and other sensors are collected and used by a control policy to actuate different devices; BCIs can act in various interaction loops with their users; (**Red**) cognitive interactions through direct electrical stimulation of the CNS or PNS; (**Green**) physical interaction through power assist or vibration stimulation of the musculoskeletal system; (**Blue**) environment interaction with external devices stimulating high-level perception through audio or visual feedback.

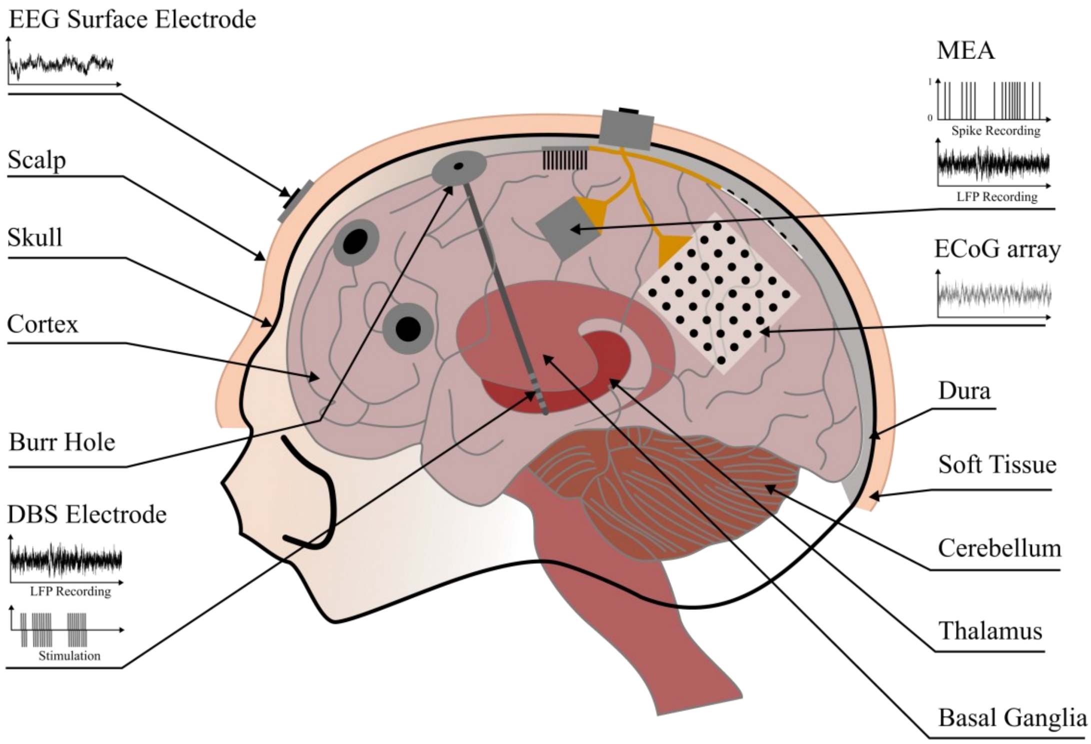


Figure 1.2 – Illustration summarising the positioning of different implantable devices (**MEA, ECoG, DBS**) for invasive and externally placed electrodes for non-invasive (**EEG**) electrophysiological recordings; Deep brain structures include the basal ganglia, the thalamus and the cerebellum; Different layers of tissues between the cortex and the scalp are represented; ECoG and MEA are implanted under the dura; MEA electrode spikes penetrate the cortical tissues; MEA have sufficient spatial resolution to measure individual action potential activity; ECoG are larger and measure the LFPs present at the surface of cortical tissues; EEG measures the same signals but filtered down by the dura, skull and scalp tissues; DBS electrodes are longer probes inserted through a burr hole to reach a target sub-cortical organ, such as the basal ganglia; The probe can measure LFPs in the neighbouring tissue volume; DBS contacts are larger and do not typically allow yet for action potential recording.

-degenerative disease such as amyotrophic lateral sclerosis (Lafleur et al., 2013; Rao, 2013; Wolpaw et al., 2002).

Interaction with the CNS is then required through what is commonly referred to as a brain-computer interface (BCI) (Nicolas-Alonso & Gomez-Gil, 2012; Rao, 2013; Vansteensel et al., 2016; Wolpaw et al., 2002). A BCI system can use various sensor arrays to measure brain activity from broad to localised spatial scales (Figure 1.2) (Daly & Wolpaw, 2008; Liu & Van der Spiegel, 2017; Nicolas-Alonso & Gomez-Gil, 2012; Schwartz et al., 2006).

Although brain imaging approaches, such as functional magnetic resonance imaging or magnetoencephalography, can be employed (Mridha et al., 2021; Nicolas-Alonso & Gomez-Gil, 2012; Wolpaw et al., 2002), the required hardware is bulky and expensive. It will most likely never permit any ambulatory applications (with the exception of facial near-infrared spectroscopy) and will always be constrained to a clinical or a laboratory environment (Daly & Wolpaw, 2008; Mridha et al., 2021; Nicolas-Alonso & Gomez-Gil, 2012).

Electroencephalographic (EEG) has been the most common non-invasive approach to BCI and continues to drive most of the research within the field (Wolpaw et al., 2002). EEG external headsets or caps are composed of a grid of electrodes that ideally sit directly on the scalp surface (Figure 1.2). These devices have been steadily improving in quality, practicality (for example, with the development of dry electrodes (Lopez-Gordo et al., 2014)), and price-point (such as the OpenBCI system (OpenBCI Documentation, n.d.)). EEG-driven BCIs have been employed to control target devices of increased complexity, such as 2D cursors (Wolpaw and McFarland, 2004), exoskeletons (He et al., 2018), robotic arm manipulators (Bradberry et al., 2010; Korik et al., 2018) and even drone quadcopters (Lafleur et al., 2013). However enticing those applications may be, these BCI systems will always be constrained by the low temporal and spatial resolutions of EEG signals (Liu & Van der Spiegel, 2017; Nicolas-Alonso & Gomez-Gil, 2012), sensibility to external noise sources, and to a lesser extent, the limited everyday wearability and conspicuity of a recording cap.

For long-term stability, practicality, and high-fidelity electrophysiological recordings, sensor arrays must be installed near or within the relevant brain tissues (Liu & Van der Spiegel, 2017). Many promises have been made about the limitless possibilities a robust implanted BCI would unlock. Some remarkable technical demonstrations have paced the progress carried within the discipline and helped anchor potential applications into reality. For example, the spiking action potential activity of multiple neuron units is now reliably recorded using microelectrode arrays (MEA) implanted directly within the deeper layers of the cortex (Figure 1.2). The interpretation of the resulting neuronal code has accurately provided tetraplegic individuals with enough control over a complex robotic arm to feed themselves (Collinger et al., 2013; Wodlinger et al., 2015). New MEA bi-directional interfaces, this time capable of stimulating neuronal cells, have enabled some to re-establish lost cutaneous sensory inputs (Flesher et al., 2016), enough to feel the touch of a loved one through a tactile prosthetic. Advances in subcranial electrocorticographic grids (ECoG) implant technology offer lesser invasive options to measure broader cortical activity patterns. ECoG installation consists in laying thin macro-electrode sheets between the dura and the relevant cortical region (Figure 1.2). These grids provide a high fidelity reading of the local EEG activity. Motor information extracted from these signals has permitted other tetraplegic patients to communicate directly with a full-body powered exoskeleton and regain momentarily upright mobility (Benabid et al., 2019).

B. Challenges of Invasive BCI Development

BCI technologies combined with state-of-the-art robotic devices could assist treatment, enhance quality of life and alleviate both physical and mental pain. However, taking any neural-interfacing concept out of the laboratory and translating it into a successful medical product that can contribute to the welfare of the many is a tortuous and complex joint endeavour crossing over the clinical, regulatory, financial, commercial and technological domains (Borton et al., 2020).

One problem is challenging to circumvent with technical breakthroughs alone: access to invasive human electrophysiological recorded data. Data is first required on a small scale for research, engineering development and proof of concept. Secondly, small patient cohorts (relative to that necessary for testing a new drug) are still needed for clinical trials to demonstrate safety, validate therapeutical claims, and, finally, for marketing clearance from state regulators (Borton et al., 2020). Thirdly, data would most likely have to be gathered and shared on larger scales for the most ambitious theoretical applications to train the necessary interfacing algorithms and monitor any product deployment (Borton et al., 2020). Invasively recorded data is challenging to access in practice, especially for research purposes. In contrast, EEG recordings can be aggregated in large dataset repositories. For example, Luciw et al. (2014) have recorded 3,936 slight variants of the same motor actions from 12 individual participants. EEG data collection thereby has permitted the training of advanced deep machine learning algorithms (Lawhern et al., 2018).

Ethical and safety considerations associated with installing cortical implants are substantial. The installation first involves a craniotomy, with significant associated transient risks and device failures (Wong et al., 2009). Surgeons cannot physically press high-density MEA into the tissue with the correct amount of force and precision required for a full insertion; instead, a powered insertion device is required (Nordhausen et al., 1996). In addition, projected risks may overwhelm benefits, given that the signal strength is known to decay with time and the uncertainty surrounding long-term implant integrity and safety is unknown (Schwartz et al., 2006). In the United States, MEA installations have required investigational device exemptions from the federal food and drugs administration (Wodlinger et al., 2015). So far, participants who had a minimal risk exposure to the installation procedure (most notably tetraplegic patients) have been the prime recipients of these devices. The potential patient pool from which participants can be drafted into basic science studies is thus restricted (Benabid et al., 2019).

Instead, animal models, in particular, non-human primates, have been and remain essential for the development of new hardware (Nordhausen et al., 1996; Stanslaski et al., 2018), algorithms (Schwartz et al., 2006) or medical procedures (Aziz et al., 1991). Despite strong electrophysiological similarities, these models are insufficient for complete clinical validation and are carried out within strict clinical protocols and constrained environments. Resources, special facilities, and time have to be allocated for training the animal subjects to carry out an experimental task and attend to their welfare. In addition, it is impossible to gain the informed feedback necessary for user-centred product development. These points are particularly problematic for devices that should be deployed in the real -world and positively affect users' daily lives.

Other paths for invasive BCI development, which rely on a significantly large and accessible human participant cohort for invasive data recording, are required. For instance, research has and still is greatly benefiting from the treatment of epilepsy (Gruenwald et al., 2019; Leuthardt et al., 2004; Torres Valderrama et al., 2010). ECoG is now commonly employed for short-term monitoring of seizure focal to inform surgical decisions. The implanted arrays can remain accessible for recordings, and patients can be recruited for various studies. At first, the implant recordings can deepen the fundamental understanding of both healthy and pathological electrophysiology, used to improve hardware, treatment delivery and even general-purpose BCI research. Tailoring the BCI to the disorder in question can provide a more helpful research synergy and serve more intermediate goals. For instance, methodology derived from BCI research is being employed for advanced seizure detection and disease monitoring (Toth et al., 2020; Zhu et al., 2020). These applications may pave a straighter translational path to a functional medical device than a general-purpose BCI system.

C. DBS: New Opportunities for Invasive Interfacing

Patient cohorts of common neurological disorders for which invasive recording probes are a medical necessity represent a more reliable data source.

An increasing number of such disorders are now treated with deep brain stimulation (DBS), the first of which is Parkinson's disease (PD). Since its development throughout the 1990s (Aziz et al., 1991; Benabid et al., 1991; Gardner, 2013; Limousin et al., 1998), DBS has become the standard of care for PD alongside levodopa medication (Lozano et al., 2019). According to NHS England's commissioning board, it was estimated that approximately up to 300 PD patients alone could be eligible for DBS treatment each year ("Clinical Commissioning Policy : Deep Brain Stimulation (DBS) in Movement Disorders," 2013). DBS involves the insertion of millimetre thin electrode leads (usually two, one per-hemisphere) through small burr holes in subcortical brain structures, delivering high-frequency current pulses (Figure 1.2). The procedure is minimally invasive compared to other methods, is supported by a wealth of clinical evidence and has well-known associated risks (Beric et al., 2002; Kenney et al., 2007; Pozzi & Pacchetti, 2017). DBS electrode contacts can be employed to record invasively electrophysiological activity emanating from local field potentials (LFP). The long term stability of these recordings is also well established (Abosch et al., 2012). The signal acquisition can also be completed with current and modified implanted stimulator hardware, familiar to clinicians and surgeons, benefitting from past clinical experience and already approved for market (Borton et al., 2020; Rouse et al., 2011).

Developing BCIs using LFP recordings from DBS macro-electrodes offers an opportunity akin to ECoG for seizure monitoring: fundamental electrophysiological research, an accelerated transition of scientific discovery towards new medical technologies and applications to increase the effectiveness of DBS therapy. DBS treatment can provide sufficiently large patient cohorts for data gathering, stable and safe recording implants for long-term deployment of a neural interface and a well-developed and funded technological platform for future application integration.

Nonetheless, it is important to reflect on the types of applications suitable for this new class of neural interface, specifically to elicit at first and later sustain interest in their development. Justifying the use of a DBS electrode for something completely outside of its original application scope can be a hard case to make. The same ethical, regulatory, or even commercial barriers faced by other invasive system would again apply. In addition, these applications should consider the more limited recording capabilities of current DBS hardware.

Application synergies with DBS therapy complementing its effects or addressing its shortcomings (for PD or that of a similar neuro-degenerative disease also treated with DBS, such as, essential tremor), may be a good initial course. Symptoms, such as, bradykinesia[1], rigidity[2] or uncontrollable tremors, strongly impair motor functions. Fortunately, they also leave distinguishable biomarkers in neural signals. Combined with other sensor modalities, a robust neural decoder could build an accurate observation space of its user's current condition, intensity of symptoms, and intended motor actions. Control policies informed by this space can be enforced across several invasive and peripheral actuation points to assist treatment automatically through different interaction loops (Borton et al., 2020) (Figure 1.1b).

One example is closed-loop or adaptive regulation of stimulation delivery (Little et al., 2013) (Figure 1.1b). Habituation to stimulation is known to take hold with time resulting in a gradual diminishment in effectiveness (Lozano et al., 2019). Minimising stimulation to instances where it is truly needed is thus of importance to maximise treatment quality. Another good incentive is also to reduce power consumption to extend battery life and minimise its replacement. Stimulation control can be achieved by defining an activation context based on decoded motor and pathological states from LFP decoding (He et al., 2021; Little et al., 2013). Depending on the pathology, activating or deactivating stimulation can be helpful, for instance, if motion is detected.

[1] Bradykinesia translates into a slow and poorly scaled development of movement.
[2] Rigidity, or, high muscle stiffness caused by an abnormally high muscular tone, also reduces motion.

Stimulation of peripheral nerve endings can also induce tremor reduction (Dosen et al., 2015). Again a neural decoder could inform external stimulation braces to complement DBS when it cannot sufficiently suppress tremors (Oertel et al., 2017).

Furthermore, decoded neural states could also be useful to inform physical interaction loops between wearable devices and its user through mechanical feedback.

A particular shortcoming of DBS is its inability to resolve (or even its inclination to exacerbate) PD symptoms associated with gait (Lozano et al., 2019), including postural and balancing problems, festination, or gait freezing. It has been shown that these symptoms can partially be alleviated by triggering external stimuli acting on mechano- and proprioceptive receptors, for instance, using small vibration actuators (Volpe et al., 2014). Meanwhile, it has also been shown that gait patterns, such as, cadence, are encoded in DBS neural recordings (Fischer et al., 2018; Tan et al., 2018). Vibration wearables could be triggered when these patterns are abruptly disturbed during a freezing episode.

Driving a lower-limb exoskeleton using DBS decoded states would also be plausible in more severe cases. Again informed by these states, the device could help users increase mobility, follow a correct step cadence, balance, and even prevent falls in freezing events (Ferrari et al., 2016). Exoskeleton usage has been proposed for treating other Parkinsonian symptoms, such as, tremor suppression (Rocon and Pons, 2008) and movement amplification for bradykinesia (Raciti et al., 2022). Their control could also be directly informed by neural state decoding.

Moreover, as in more classical BCI paradigms, decoded states can also be used to take over control of remote devices or applications.

DBS users could be trained to move cursors in simple tasks such as a centre-out paradigm (Wolpaw and McFarland, 2004) or to play simplified video games (He et al., 2019). Although PD patients do not per se suffer from the same kind of paralysis caused by some severing of the PNS, their mobility can be severely reduced to virtually none in some extreme manifestation of the disease (instead caused by a specific breakdown of internal CNS pathways, which will be further discussed in Chapter 2). A cursor interface would not serve as a communication interface but rather as a neuro-rehabilitative tool geared at restoring some functionality in the degraded pathways. Certain gamified paradigms have been proposed to help DBS users reduce certain pathological neuronal biomarkers (He et al., 2019). Feedback loops are then closed by video, and auditory cues produce by the rehabilitative application (Figure 1.1b).

Lastly, pure device control, such as control of robotic manipulanda, can also be imagined for more exploratory work, as it is done, for instance, with EEG (Bradberry et al., 2010; Korik et al., 2016). DBS-recorded neural signals have been shown to contain biomarkers associable with precise motor control, such as end-effector grip force (Tan et al., 2016). Although the recording ability of DBS probes is limited, they provide a unique insight into CNS regions heavily involved in regulating motor processes. Compared to cortical regions, the subcortical regions targeted by DBS, most notably, the basal ganglia (BG) and thalamus, have been relatively, and regrettably so, underexplored in the broader field of BCI research. Part of the key to unlocking fine-motor decoding for neuroprosthetics could be cortical and auxiliary deep-brain signal acquisition probes and integrated decoding of motor commands. A multisite BCI could become a reality with recent advances in high-density implantable recording probes (Musk, 2019).

Deep-BCI research can help further expand technologies associated with DBS in the short run and promises pertinent contributions to the broader field of BCI research in the long term. In the meantime, much is left to investigate and to research in order to support future successful technological translations.

D. Dissertation Objective

This dissertation studies the usage of LFP recorded signals via DBS electrodes implanted in the subthalamic nucleus (STN, a sub-organ of the BG) of human subjects treated for PD to predict simple volitional motor states. The interfacing problem of interest is the control of simple peripheral devices. In future applications, these devices could be a one-dimensional robot or a closed-loop stimulation system. The work tested the hypothesis that neural decoders could be designed and deployed under real-time constraints to translate STN-LFP signal features or biomarkers into elementary control signals that a peripheral target device could interpret. The end-goal is thus to achieve an intuitive control link for a user equipped with DBS leads towards the target device. Reaching this objective would support that STN-LFP decoding could be a valuable avenue for specific and general-purpose BCI applications. It would also further confirm that DBS data gathering can propel the development of neurotechnologies.

E. Scope of Work

This work addresses specific gaps in past and concurrent translative DBS and BCI research.

It first discusses the requirements of STN-LFP decoders, and DBS recorded datasets. It contributes to the extensive review and applications of algorithms to efficiently extract biomarkers from STN-LFP. It further extends ideas surrounding decoder parameterisation and decoder tuning for BCI task optimisation. Too often, the studies of constrained experimental tasks overlook the demands of robust real-world deployment. The investigation of different tuning strategies aims at elaborating solutions to this specific translational problem.

Secondly, this work demonstrates how elementary motion, particularly the bulk detection of simple iso-metric gripping actions, can be inferred asynchronously from STN-LFP. Asynchrony in this context refers to continuous decoding in time without any prior knowledge of the action onset. So far, only a handful of studies have explicitly addressed this problem for motor state decoding, with varying degrees of success. Demonstrating asynchronous decoding is essential as any control interfacing problem is unviable without it. Bulk movement decoding (or discrete state decoding) was also employed to evaluate the various decoder design implementation and tuning strategies.

Thirdly, this work further studies the coupling between the dynamics of STN-LFP biomarkers and isometric gripping actions. So far, only simple first-order transient dynamical models have been proposed and investigated. In particular, the relationship between STN-LFP biomarkers and the rate of application of gripping force (or yank) is unknown. From the decoder design perspective, this lack of understanding is problematic. Many successful BCI decoders function by assigning different biomarkers to specific motor state variables of a different order (such as end-effector position or velocity). The study of yank, grip force, and neural dynamics here permits exploring similar approaches for decoding STN-LFP and further comprehending through its electrophysiology the involvement of the BG in the generation of motor processes.

Lastly, in addressing those points, this work helps to set a realistic range of applications for STN-LFP decoding with potential paths for improvement and expectations relative to other types of BCI driven either by DBS recorded LFP or by other invasive acquisition means.

F. Contribution to Knowledge

This dissertation first reports investigative contributions of interest to the field of deep-brain interfacing research:

- A systematic review that surveyed state-of-the-art studies in the field.
- Evidence assessing the feasibility of asynchronous decoding of bulk motor-states from STN-LFP signals in both an action detection and a laterality selection problem (derived from isometric hand contraction motor tasks).
- An extensive comparison of decoding algorithms applied to the problems mentioned above. The algorithms were compared across multiple characteristic metrics (e.g., feature signal relevance to decoded state, decoding performance, model complexity).
- Evidence attesting to the relevance of decoder tuning for performance optimisation.
- Evidence describing the variable response STN-LFP biomarkers under different force delivery rate regimes in isometric hand motor tasks.
- Comparisons of algorithms for the asynchronous decoding of continuous isometric states, including force and yank, from STN-LFP.

In doing so, this dissertation also reports a series of relevant technical contributions:

- A unified methodology for the design of STN-LFP decoder pipelines under the form of an architectural framework:
  - The framework formalises concepts such as the decoding problem definition (mainly by discerning between discrete and continuous state decoding) and design specifications (e.g., the refreshing decoder rate, the overall analysis window, or feature space size).

    - The framework breaks down the pipeline into main sub-stages (the principals are pre-processing, feature extraction, and state estimator stages) and sub-modules. It insists on modularity to swap components and make comparisons between algorithms.
    - The framework describes how to consistently implement constraints for real-time and asynchronous decoding (mainly by considering memory states and dataset organisation), how to define different parameter levels, and consider strategies for parameter fitting or optimisation throughout pipeline stages.
    - The framework insists on defining a global objective function for a given decoding problem and lays out methods for decoder fitting and validation.
- A repository of real-time and multi-channel feature extraction, spatial filters, and feature selection modules for STN-LFP signals, including conventional techniques and advanced bio-signal decomposition methods such as a parameterised discrete wavelet transform. Repository items can be used to build tailored decoding algorithms within the architectural framework and could be shared as a standalone library.
- Two task and user-specific decoder tuning methods based on gradient-backpropagation and Bayesian optimisation, respectively.
- An experimental methodology to investigate motor tasks with different force and yank coupling.

G. Dissertation Structure

This dissertation has the following structure:

Chapter 2 sets the relevant background necessary to contextualise the research topic within the broader field of BCI, electrophysiology, neuroanatomy, and DBS. The characteristics of LFP, deep brain structures and their involvements in the generation of volitional movement are discussed. Parallels with cortical BCI systems are as well made. The chapter then reports on the systematic review completed on the state of BCI research in the deep brain. Gaps and potential areas of innovation in the literature are identified. Those are employed in the subsequent chapters to define more specific motivations and characterise the contributions made to the field.

Chapter 3 presents the algorithmic foundations employed to design the proposed STN-LFP decoders. It introduces the overarching structure of the decoder, the different strategies for performance optimisation and model validation. The methods for signal pre-processing, artefact effect mitigation, channel management and feature extraction are also presented. The chapter details how methods employed in past BCI studies were adapted to suit the characteristic of STN-LFP signals, the implemented optimisation strategies and the BCI tasks studied in subsequent chapters. Hardware solutions for translation into real-world applications are lastly discussed.

Chapter 4 summarises work related to discrete state decoding. Two BCI tasks were studied: asynchronous detection of grip isometric actions and hand laterality selection. First, metrics measuring the relationship between discrete states and the features extracted from the signal are discussed, specifically their application to channel and feature selection. Second, the methodology for designing the discrete state estimator and task performance objectives are presented. The performances of the different prototype decoders are reported and analysed. The effects of optimisation on task performance on parameterisation are discussed.

Chapter 5 summarises work related to continuous state decoding. First, a new paradigm mixing variable gradient and force level conditions is described. It permitted the study of relationships between the STN-LFP biomarkers and dynamically generated isometric grip force. An offline analysis methodology was developed to test general hypotheses on the STN-LFP behaviour. The decoding strategy for continuous state estimation is then presented. Performance results, optimisation improvement and decoder parameterisation are again discussed.

Chapter 6 provides a general discussion on the overall dissertation outcome. The chapter touches on the limits of STN-LFP decoding and proposes possible applications and future work avenues.

Appendices provide details on datasets used in this dissertation, details on hyperparameter optimisation investigation and on signal processing for neural feature extraction.

Chapter 2. Literature Review

This chapter summarises different background items and the surveyed literature to help ground the research topic. These include a brief description of the neuroanatomic relations between cortical and deep brain regions and their inter-connectivity, mainly through the scope of the cortical-basal ganglia thalamic cortical (CBGTC) circuit. The positioning and function of the subthalamic nucleus (STN) within this circuit are emphasised. The involvement of the described structures in the generation, regulation of volitional motion and the symptoms associated with Parkinson's disease (PD) are also discussed. Secondly, a simplified overview of DBS surgery explains how LFP data is acquired. The nature, properties, and associated biomarkers present in those signals are then thoroughly presented. An explanation is also given to how those biomarkers are most employed for cortical BCI systems. Lastly, the chapter presents a survey of past studies which researched BCI applications using deep brain recorded LFP. The survey summarises the type of participants enrolled, the type of state decoded from the LFP, and the constraints and methodology for each study. The identified literature gaps and remaining challenges for DBS based decoding are then discussed.

A. The Cortical-Basal Ganglia-Thalamic-Cortical Loop

In the context of DBS, the deep brain principally refers to several subcortical structures, including but not limited to the BG and the thalamus. Their functionality is better understood within the CBGTC loop illustrated in Figure 2.1a. This circuit plays a central role in executing and regulating all motor and cognitive functions through a process referred to as reinforcement learning (K. Doya, 2000).

1. Basal Ganglia Structures and Pathways

The putamen (PT) and the caudate nucleus (CN) together form the input layer for the BG, otherwise known as the striatum. Cortical neurons project to the BG through the corticostriatal tracks originating across frontal, motor, and sensory regions. Because of its input connectivity alone, it can be argued that part of the purpose of the BG consists in compressing and binding various and somewhat disparate cortical maps (Brown, 2007; Rivlin-Etzion et al., 2006).

The inner workings of the BG follow then a unique dichotomy. Some fibres of the striatum are routed through the direct pathway reaching directly to the output of the BG, the globus pallidus internal (GPi). The other fibres constituting the indirect pathway are rerouted through the external capsule of the globus pallidus (GPe), deviated through the STN and merged back into the GPe. The GPe connects the BG to the thalamus, which integrates information from cerebellar drives and projects back to the cortex. Although cortico-cerebellar and BG-cerebellar connectivity are not of direct interest here, it is worth mentioning briefly that the assistance of the cerebellum in the generation (Pons Jose, 2008), and

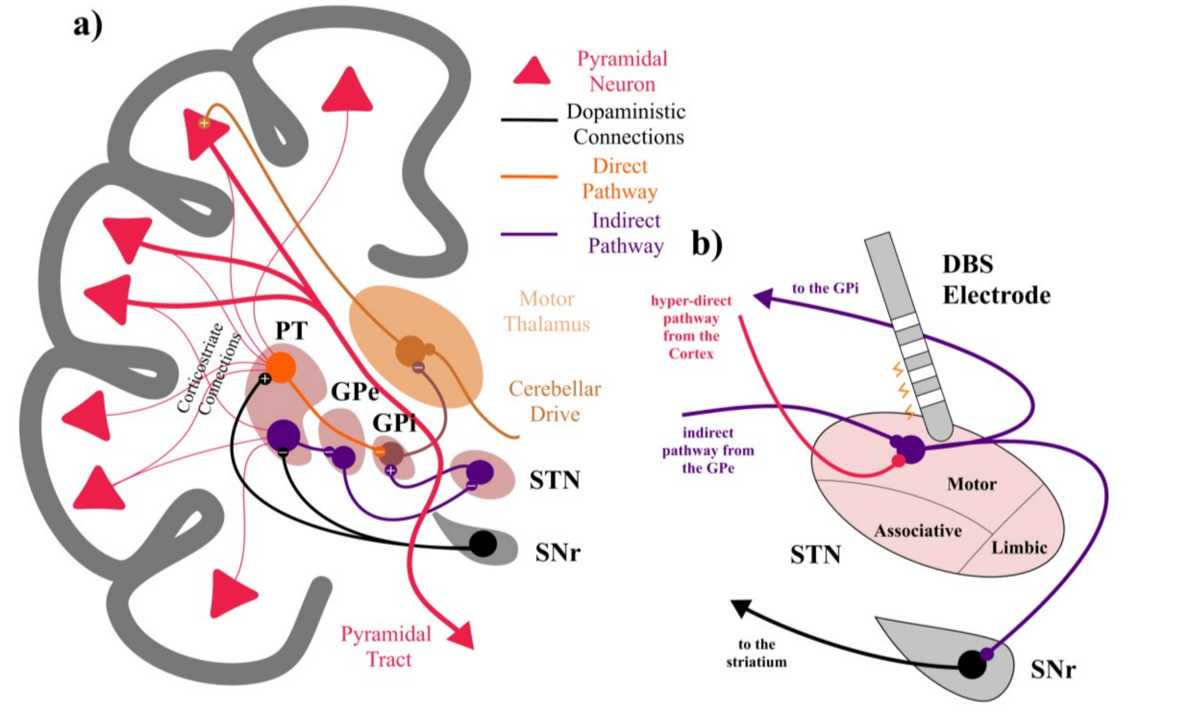


Figure 2.1 - Illustration representing major neural pathways within the deep brain and their connection to the cortex; **a)** Organisation of the CBGTC loop, including the relative positioning of the relevant organs within a coronal section and the different pathways; The **CBGTC** circuit is a feedback loop regulating activity in the cortex routed through the BG and the motor thalamus; The cortex is connected to the striatum (**PT**); Connections are split between the direct and indirect pathways; the direct pathway is routed directly routed to the BG output (the **GPi**); The indirect pathway is rerouted in the **GPe** through the **STN** back to the **GPi**; The direct pathway removes inhibition in the thalamus which drives voluntary motion; The indirect pathway reinforces this inhibition; Dopaminergic neurons in the **SNr** gate signals entering the **BG** through temporal difference learning; **b)** Local connectivity of neurons present within the **STN** and its topological partitioning; The **STN** is split into motor, associative and limbic areas; Neurons in the motor area reroute the indirect pathway to the **GPi**; The hyper-direct pathway enables the cortex to directly inhibit the thalamic drive.

regulation of fine motion is considerable. Unregulated neural activity emanating from this organ has been proposed as one possible source of pathologic PD tremor (Brown, 2007).

Several CBGTC loops exist parallel to one another and map back to different cortical regions (the primary motor cortex, the prefrontal cortex, the supplementary motor area…) (Brown, 2007; Kandel et al., 2012a). The topography of the BG and the thalamus is partitioned at the macro-level per functionality (sensory-motor, associative and limbic), classically described by the tripartite organisation (Emmi et al., 2020). It is thought (although this hypothesis is regularly challenged) that the pathways of each parallel loop dominantly interconnect areas of the same functionality through each organ, thus segregating them from one another (Brown, 2007; Emmi et al., 2020). According to Emmi et al. (2020), the motor areas of interest are nominally the primary, supplementary and pre-motor cortex correspond to, respectively, in the BG, to the dorsolateral part of the caudal PT, to the head of the CN, to the ventrolateral two-thirds of the GP and to part of the lateral substantia nigra (SNr); in the thalamus, to the ventrolateral, to the ventral anterior and to the centromedian regions.

Both direct and indirect pathways are regulated in the striatum by dopaminergic neurons projecting from an auxiliary nucleus of the BG known as the SNr. The dopamine delivery in the BG is thought to function according to principles well described by reinforcement learning, particularly temporal difference learning (B. K. Doya et al., 2001; K. Doya, 2000).

Furthermore, the STN is a small nucleus, estimated to have a volume between 100 to 240mm^3 and to contain 250-500 000 neurons, located cranially below the thalamus and above the SNr in the coronal plane (Emmi et al., 2020). It is layered between the red nucleus at is posterior and the cerebral peduncle at its anterior in the horizontal plane (Emmi et al., 2020). The posterior areas of the STN are topographically assigned to motor functions connecting the STN to other relevant motor centres (Emmi et al., 2020). Although, as indicated earlier, it is an important node in the indirect pathway of the BG, its connectivity is not limited to the GPi and GPe exclusively (Figure 2.1b). Projecting axons from the STN bifurcate in one direction towards the GPi and towards the SNr (Emmi et al., 2020). The hyper-direct pathways also project directly from the cortical regions to the STN (Emmi et al., 2020; Levy et al., 2002; Nambu et al., 1998). There is also evidence of other direct connections of the STN to the thalamus and cerebellum (Emmi et al., 2020).

Therefore, the STN is far more than just an intraneuronal relay within the indirect pathway. Instead, it is a fully integrated network node that interlinks various motor processing centres and helps regulate the activity circulating within the CBGTC loop during the generation of volitional movement.

2. Volitional Movement Generation

Understanding the origin of volitional movement has been key to the development of invasive BCI (Schwartz et al., 2006). A BCI is nothing more than a physical and digital tap within the neural pathway carrying the most relevant information to the interfacing problem (Pons Jose, 2008). In the case of an invasive implant, the tap is made incomparably close to the information source (Schwartz et al., 2006).

The origin of volitional movement is often traced to the primary motor cortex M1 (Kandel et al., 2012b). M1 is a principal penultimate network node where information from the frontal cortical (supplementary and pre-motor) areas and the deep brain are integrated into a single cohesive set of motor instructions (Schwartz, 1994). Large pyramidal neurons at the top of the pyramidal tracks broadcast those instruction sets directly to the motor neuron pools within the spine, arguably the ultimate network node before the activation of muscle groups. At the macro-level, the homunculus organisation of M1 implies that sub-areas are grossly dedicated to a particular end-effector (arm, hand, leg, feet…) (Kandel et al., 2012b). At the micro-level, preferred-direction encoding in the local pyramidal cell population informs the intended motion directionality (Georgopoulos et al., 1986). Preferred direction, or cosine tuning function, stems from the observation that each cell fires (action potentials) only when the desired motion is aligned with a given direction. The intensity of the firing rate of individual cells is a function of the motion alignment to its preferred direction. Motion is thereby encoded through a set of neuronal bases, and it can be reconstructed by superimposing the firing of those individual units and forming an estimated population vector (Georgopoulos et al., 1986).

Moreover, it is not easy to precisely summarise what the BG does within the CBGTC loop and how it modulates activity in M1. As the precise answer to this question is constantly evolving and beyond the scope of this work.

According to the 'Go and No-Go' paradigm (Dunovan et al., 2015), the BG acts as a passive break regulating motion initiation in the cortex. The activation in the direct pathways releases that break by removing the thalamic inhibition, whereas activation in the indirect pathway reinforces it. The hyperkinetic pathway can also fast track cancelling commands from the cortex to inhibit a developing action (Dunovan et al., 2015).

The BG has also been conceived as a ramp generator (Kornhuber, 1971). Under normal circumstances, the BG initiates, amplifies and smooths the firing patterns of cortical neurons. Imaging studies seemed to have confirmed this hypothesis, notably by demonstrating that the BG helps select different force levels for isometric grip actions (Spraker et al., 2007; Vaillancourt et al., 2007) and scale motion amplitude (Turner et al., 2003). In particular, Vaillancourt et al. (2004) have shown that the BG played an essential role in regulating the rate of change of force, or yank, during a pinching task. More specifically, it has been demonstrated that yank regulation translates into increased metabolic demand of the STN during prolonged pinching actions. Vaillancourt et al. (2004) further showed that smaller, quicker and more automatic actions do not involve the same metabolic demand. Those findings have further reinforce the idea that the STN is a central node in the feedback mechanism, which holds volitional control over the temporal pacing of complex motor actions (Spraker et al., 2007; Vaillancourt et al., 2007, 2004). Given the close parallels with temporal difference learning, the BG has been the subject of many neuro-dynamical modelling attempts (Chakravarthy et al., 2010). Notably, an interesting approach by Gupta et al. (2013) have illustrated how the BG could help regulate sensory-motor control loops to generate precision grip force. Lastly, it has been shown that preferred-direction encoding is also present within the GPi, GPe and the STN (Georgopoulos et al., 1983; Turner & Anderson, 1997), implying that the BG could also be involved in movement directionality control. In this sense, many neural encoding properties found in the cortex could be inherited in the BG.

The BG is an essential selector, gating and feedback organ that channels a substantial amount of neural activity during volitional movement. This channelling capacity is supported by the dichotomy of the CBGTC pathways, the macro-level tripartite organised mappings, micro-level cellular directional organisation, and the STN interconnectivity. Tapping within the information stream flowing through the BG could be thus extremely useful from a computer interfacing perspective.

B. Deep Brain Stimulation Surgery for Parkinson’s Disease

Much of the understanding of the BG functionality also comes from studying neurological disorders such as PD (Brown & Marsden, 1998). It is easier to appreciate the range of motor functions regulated by this structure as it ceases to operate normally. The manifestation of PD is associated with a failure in the dopaminergic circuitry of the BG (Brown, 2007). Dopamine deficiency translates into a broad spectrum of motor and cognitive symptoms, including bradykinesias, which involves slow development of motion, rigidity, causing muscular inflexibility, greatly amplified uncontrollable tremors, and difficulty in action sequencing, particularly during gait (Baltadjieva et al., 2006). Levodopa medication act as a dopamine substitute and helps alleviate some symptoms. When this medication becomes insufficient in the later stages of the disease, DBS tends to be prescribed. Whereas medication helps rebalance the BG circuits neuro-chemically, DBS acts electrophysiologically by delivering short stimulation pulses. DBS electrodes are implanted in either the GPi or the STN (in particular its motor regions) in the case of PD treatment (Aziz et al., 2002; Limousin et al., 1998).

In summary, there are two main stages to installing the electrodes. In a first surgery, the DBS probes are implanted via a small drilled burr hole (Benabid et al., 1991) whilst the probe leads are partially laid down and secured under the scalp skin. Medical imaging is a priori employed to calibrate a stereotactic frame guiding the implantation of the probe to the desired target coordinates. LFP measurement, partial stimulation and a posteriori medical imaging can be employed to confirm a successful DBS installation (Aziz et al., 2002). The stimulator device is installed at the upper-chest level in a second surgery. The DBS probe is connected to the device by redirecting the leads subcutaneously along the neckline. An external connection to the probes is then impossible. At the discretion of the neurosurgeon, the DBS can be left accessible, hanging from the scalp and secured by a full head bandage. In terms of experimental opportunity, this is the only window where it is possible to record STN-LFP from DBS patients. Lead externalisation is more often than not unnecessary and is usually avoided by scheduling the two surgeries back-to-back. Those decisions ultimately are taken for logistical reasons, concerns about patient comfort, and assertion of increased postoperative risk. It is worth emphasising that this last point may not be warranted and that increased postoperative risk is not systematically measured in practice (Rosa et al., 2017).

The DBS electrode fundamentally can act as a two-way interface. Instead of using the adapter to connect the leads to a stimulator, the adapter can be plugged into a compatible EEG recording amplifier, and LFP can be directly recorded. The logistics, the hospital settings, the availability and the state of the patient, the window between the two surgeries and time constraints stand as the main limiting factors when collecting LFP data.

C. Local Field Potentials

Any electrophysiological probe inserted into neural tissue can record a relatively strong electrical potential field propagating through the extracellular medium, commonly referred to as LFP. The precise nature of LFP is complex (Bedard and Destexhe, 2012). It remains an active area of research and can be subject to many misconceptions (Herreras, 2016). Although the precise answer to the origins of LFP is outside the scope of this work, an electrophysiological basis is still required to relate LFP to other neuronal modalities employed for BCI and to understand some elementary signal characteristics, especially those related to identifiable biomarkers.

1. Electrophysiological Characteristic and Source

LFP typically possess a strength of 100 to 1000uV (Liu & Van der Spiegel, 2017). In comparison, ECoG activity possesses a field strength of 10 to 100uV and EEG activity of 1 to 10uV. Both modalities are nothing more than a toned-down measure of LFP captured over a broader scale and filtered down by several additional tissue layers. All emanate from the same physiological phenomenon. Moreover, in contrast to patch-clamping, an LFP measurement is made within the extracellular medium; it does not directly tap the membrane of a neuron and thus does not measure single-unit activity. That is not to say that the action potential of neighbouring neurons cannot be estimated. Given an adequate density, size of electrode contacts and sufficient acquisition temporal resolution, those can be extracted from the of the LFP through various techniques (Liu & Van der Spiegel, 2017). The LFP signals are thought instead to occupy the lower frequencies of the electrophysiological spectra (Liu & Van der Spiegel, 2017). The cumulation of small inter-synaptic current residuals leaking from neuronal cell aggregates is likely the principal source of the LFP (Destexhe and Bedard, 2013) and not so much that of firing activity. The nature of LFP is thus a far more complex epiphenomenon that results from the collective activity of ensembles of neurons and is reflective of the overall state of the network. In this sense, LFP could be qualified as a medium-fidelity modality where single spike action-potential recording ranks the highest. An average ensemble of individual action-potential units can then, in theory, be employed to recover some similar network properties (Rouse et al., 2011). There are many other caveats when it comes to the interpretation and even the semantics of LFP, to name a few, the actual physical locality of the fields, the reach of stronger sources over weaker ones or the distorting effects of the extracellular medium geometry (Herreras, 2016).

This is where the precise electrophysiological nature of LFP is of no further interest, and what matters instead is the capacity of those fields to carry precise biomarkers associable with distinct motor, pathological or cognitive states.

2. Synchronisation and Spectral Properties

The intrinsic dynamics of neural cells facilitate various firing regimes (Izhikevich, 2007). Some of those regimes are based on periodic or pseudo-periodic patterns, with each cell acting as an oscillator. Synchronisation refers to the physical phenomenon where two interacting oscillators are progressively forced to adopt a shared frequency and lock phase with one another (Destexhe and Bedard, 2013; Pikovsky and Rosenblum, 2007). It is believed that the CNS can enforce population-wide synchronisation regimes across cells of the same functional network. These synchronisation events are themselves a mechanism thought to bolster communication between different macro-level network nodes and leave discernible changes in the spectral behaviour of the surrounding LFP.

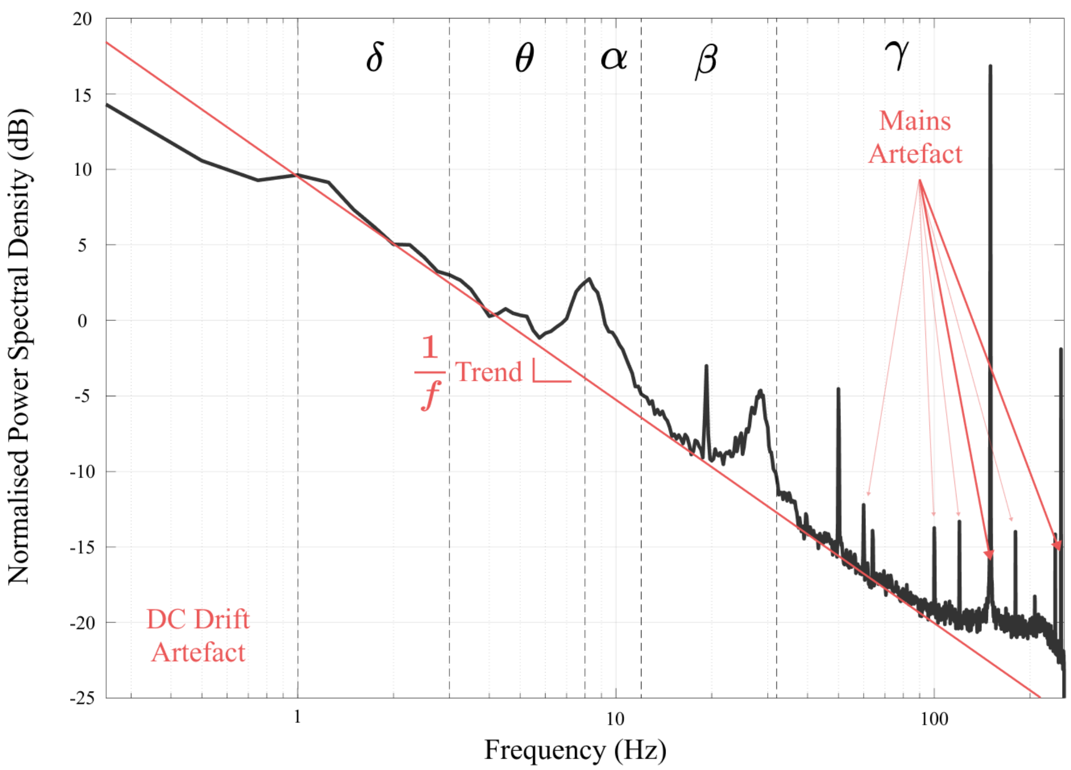


Figure 2.2 - Example of an STN-LFP signal spectral density with the key spectral sub-domains and the location of artefacts; Recording from Sub01_Level_Right (Appendix A) was used; Welch's method was used to estimate the spectral density using a resolution of 0.25Hz or 8s-long windows with 50% overlap between windows; LFP signal was sampled at 512Hz; The signal was normalised by its standard deviation making spectral estimation relative to total power; Bipolar construction was used; Contact pair with the maximum β activity was selected.

The baseline LFP spectra, i.e. when a resting state is adopted with, for example, a total absence of movement, is characterised by an inversely proportional relationship between the amplitude content and frequency (or 1/f). Several hypotheses have been researched to determine the precise cause of this spectral profile (Baranauskas et al., 2012; Destexhe and Bedard, 2013; Szendro et al., 2001). In terms of signal processing, the main implication is that the noise baseline is not white (i.e. not normally distributed Gaussian noise) but rather pink. Figure 2.2 presents an example of a spectral power density estimation of STN-LFP signal with its characteristic 1/f trend ranging from 1-100Hz.

Any non-stationary deviations or modulation in time from this baseline state can be characterised as neural activity. Those modulations are far from being purely stochastic but coincide with changes from baseline motor or cognitive states to new synchronisation regimes of the local neural network. Increases and decreases in the power of a given frequency band from baseline activity are otherwise known as event-related synchronisation (ERS) or desynchronisation (ERD), respectively (Pfurtscheller & Lopes da Silva, 1999). Both ERS and ERD are the principal biomarkers found in LFP. Gruenwald et al. (2017) proposed the following simplified model to help further generalise the behaviour of ERS and ERD: if the power-spectra $P(t, j\omega)$ of an arbitrary LFP signal could be perfectly estimated at a time t and at an angular frequency $j\omega$, then P is modulated from its resting baseline state

$P_0(j\omega)$ by a set of slower and temporally non-stationary signals $\{x_i\}_{0\leq i<N_s}$. Each signal x_i can be attributed to a specific ERS and ERD such that:

$$P(j\omega,t) = P_0(j\omega) + \sum_{i=0}^{N_s-1} W_i(j\omega)x_i(t) \qquad (2\text{-}1)$$

where W_i is the unknown frequency weight distribution of the i^{th} component and N_s the total number of slow signal components. Each ERS and ERD signal x_i enacts the amplitude modulation (AM) of a carrier signal $W_i(j\omega)$. Most approaches consider that ERS or ERD are bound to frequencies defined within the limits of a sub-band range: for $\omega \in \Omega = [0,\pi]$ (assuming that ta Nyquist frequency bounds the spectral domain), the sub-band domain of W_i is hence defined as $\Omega_i = [\omega_i^l, \omega_i^u]$ where $\omega_i^l, \omega_i^u \in \Omega$ define the lower and upper-frequency bounds of Ω_i (Pfurtscheller & Lopes da Silva, 1999). The sub-bands are ideally assumed to be non-overlapping such that $\Omega_o \cap \Omega_1 \ldots \cap \Omega_{N_s} = \{\emptyset\}$. The sub-band domains Ω_i are at times interchangeably referred to as neural rhythms. The nature of carrier frequency $W_i(jw)$ vary depending on its sub-band domains Ω_i (Table 2.1). The lower frequency domain can, for instance, be difficult to distinguish in the spectra. Because of the 1/f characteristics present in $P_0(j\omega)$, LFP recordings can be subject to low-frequency direct current (DC) drift artefacts. Motion artefacts can also contaminate the lower signal spectra. Typically, low frequencies are defined within 1-8Hz and can be split into two sub-domains delta (δ, 1-3Hz) and theta (θ, 3-8Hz). Two important rhythms are ubiquitously present within the LFP spectra under the form of narrowband peaks: the alpha (α, 8-12Hz) and beta (β, 12-32Hz). α and β rhythms can be referred to as 'idling', meaning that they are continuously present with the rest spectra and suppressed through an ERD during a change in brain state. A famous example of α suppression is, for instance, measured in the occipital region associated with eye movement (Adrian and Matthews, 1934). It can also be relabelled as the mu (μ) rhythm when associated with sensory-motor process. β rhythm is more exclusively associated with sensory-motor processes. It is thought that this rhythm is present when a current resting state is passively or actively maintained (Engel and Fries, 2010; Kristeva et al., 2007; Omlor et al., 2011). Departing from this state would require the suppression of this idling rhythm. Neural rhythms above 32Hz, significantly gain in bandwidth. Those are mostly contained within the gamma (γ) range traditionally defined as 32-90Hz. The γ spectrum does not typically exhibit any features at rest. γ ERS translates however in a pronounced gain in signal energy relative to the baseline (Gruenwald et al., 2017) and occurs typically during motion. The γ spectrum can be broadband or partitioned into lower (32-60Hz) and upper (60-90Hz) sub-domains (Jenkinson et al., 2013). Depending on the function of interest and the recording regions, those sub-bands can have different interpretations. For instance, the lower cortical γ coincide with the Piper band, which emanate from a direct synchronisation drive between cortical neurons and spinal motor neurons (Brown, 2000). The upper γ can also coincide with the so-called fine-tuned γ , which acts as a binding mechanism between different actions (Jenkinson et al., 2013). Broadband γ is lastly associated with movement ERS notably at the cortical level. One of the benefits of LFP recording with ECoG, DBS and MEA over EEG, is an increased temporal resolution and thus increased acquisition bandwidth (Liu & Van der Spiegel, 2017). Frequencies activity above 90Hz, or high frequencies, are being actively researched and have become of interest in particular for BCI development (Gruenwald et al., 2017, 2019; Hirschmann et al., 2017; Yao et al., 2020; P. Zhang et al., 2019).

Table 2.1 - Summary of the different Ω_i sub-domains of the LFP spectra and their properties as biomarkers for decoding motor and pathological states.

Domain Ω_i	Bounds (Hz)	General Description and Association	Motor State Association	Pathological State Association in PD or Other	Recorded and Decoded from
δ	1-3	Very low frequencies rhythm.	Associated with cognitive states (e.g. sleep).		EEG, ECoG, DBS, MEA.
θ	3-8	Low frequencies.	Associated with motor states and with cognitive state (e.g. sleep).	Has been associated to neuropathic pain.	EEG, ECoG, DBS MEA.
α (or μ)	8-12	Narrowband idling rhythm associated with sensory-motor processes or visual processes.	α is associated with cognitive states (e.g. sleep, attention). μ ERD is commonly associated with movement or motor action, exploited in EEG-BCI.	α stereotyped and underregulated bursts have been associated with PD symptoms; Has been associated with neuropathic pain.	EEG, ECoG, DBS, MEA.
β	12-32	Narrowband idling rhythm associated with sensory-motor processes.	β ERD is commonly associated with movement or motor action.	β stereotyped and underregulated bursts have been associated with PD symptoms.	EEG, ECoG, DBS, MEA.
γ	32-90	Broadband rhythm associated with sensory-motor processes.	γ ERS is commonly associated with movement or motor processes such as cortico-spinal synchronisation drive and action binding.	γ activity is suppressed in PD but restored with dopamine medication intake.	EEG limited to low γ (<60Hz), DBS, ECoG, MEA.
High Frequencies	90-1000Hz	Broadband activity only measurable in LFP.	High frequencies are primarily associated with motor processes but remain an active area of research.	High frequencies may also correlate to pathological effects.	ECoG, DBS, MEA.
Single Unit Spikes	-	Single spike units extracted from LFP by dense electrode arrays; High-spatial and temporal resolution is required for source separation.	Preferred direction encoding permits to reconstruct motion.	Need many units required to find firing patterns associable with pathological dysfunction of the local network.	MEA.

D. Decoding Motor States using LFP Biomarkers

The LFP spectral structure contains a plethora of biomarkers, predominantly under the form of neural-rhythm synchronisation and desynchronisation, which neural decoders can interpret to predict motor actions.

1. Cortical LFP Decoding

In the case of EEG, α and β ERD rhythms are the principal biomarkers employed for remote device control. Wolpaw and McFarland (2004), for instance, have trained users to displace a 2D cursor towards a given target by inducing broad-scale modulated suppression of those sub-bands at the level of the cortical sensorimotor regions. Users learned to balance activity between their right and left hemispheres to control each degree of freedom of the cursor. Low γ can also be utilised (Korik et al., 2018) as well as low δ frequencies, which are thought to be relatable to velocity (Bradberry et al., 2010). Hence, with training, despite the low spatial resolution, large scale EEG oscillatory activity change can therefore be recruited spatially on the cortex macro-organised mappings to control objects.

ECoG also functions according to the same principle but with increased effectiveness. Smaller and denser implanted electrode arrays can be placed directly above the relevant cortical areas and capture the activity variation of these mappings with a far increased spatial resolution (Schwartz et al., 2006). For instance, visual stimuli decoding can be accurately achieved from an array implanted in occipital and frontal regions (Kapeller et al., 2018) or motion selection from those implanted in the sensory-motor cortex (Gruenwald et al., 2019; Merk et al., 2021). In addition, the higher temporal resolution of ECoG permits tapping into the high γ range of the LFP spectra, which again have been shown to be reliable biomarkers for motor action decoding (Gruenwald et al., 2017, 2019; Merk et al., 2021). One of the principal benefits of accurately capturing relevant biomarkers through EGoG is a significant reduction in training time for the user to attain optimal performance in a given interfacing task (Leuthardt et al., 2004).

Moreover, MEA electrodes, because of their reduced size and high electrode contact density, only focus on very precisely localised areas to capture a unique cortical functionality pin-pointed by pre-trial imaging experiments (Collinger et al., 2013; Wodlinger et al., 2015). In addition, to single-unit recording MEA can also measure LFP activity, which has repeatedly been employed to drive BCI systems (Ahmadi et al., 2019; Flint et al., 2012; Hsieh and Shanechi, 2016; Zhang et al., 2019). What had been shown prior to those studies by Heldman et al. (2006) was that ERS and ERD in cortical LFP, in particular in the γ range followed similar cosine-tuning functions as single spike units. MEA recordings can also, therefore, exploit the micro-organisational layout of the cortex to achieve LFP decoding without the need for single-unit activity isolation (Flint et al., 2012).

There is no lack of evidence that information can be successively extracted and decoded at every scale and across different neural rhythms from LFP measured with invasive and even non-invasive recording arrays. Comparable results could be potentially achieved with STN-LFP.

2. Evidence of LFP-Biomarkers for Deep Brain and STN Decoding

LFP recordings in the BG have helped elucidate its inner workings in both pathologic and normal regimes.

Synchronisation regulation is essential for the correct operation of the CBGTC loop. First, the γ rhythm has been proposed to be a critical synchronisation frequency (Brown and Marsden, 1998; Jenkinson et al., 2013) to help bind together different actions stages. Under PD conditions, however, it is thought that the segregation between CBGTC functional loops is severely degraded with phase-locking of the β synchronisation across parallel loops (Brown, 2007). This phase-locking induces an overdrive of β activity in the BG marked by stereotypical bursts and is conjointly acted with a suppression of γ activity (Brown & Marsden, 1998). γ rhythms are thought to be pro-kinetic (Brown, 2000), and β oscillation has previously been mentioned as anti-kinetic, thus inhibiting motion, especially at the cortical level. The suppression of γ and de-regulation of β are therefore biomarkers associable with rigidity and bradykinesia symptoms (Brown, 2003, 2007; Little et al., 2012). Cortical-STN coherence has been shown to contribute to the β overdrive (Brown, 2007; Silberstein et al., 2005), as STN neurons are synchronised to the cortex through the hyper-direct pathway (Levy et al., 2002). STN DBS, in part, helps break this resonance coupling (Brown, 2007), again further emphasising the involvement of the STN in the regulation and amplification of motor processes originating from the cortex.

Furthermore, synchronisation patterns also travel through the cortical-STN tracks as part of normal motor regimes (Cassidy et al., 2002; Levy et al., 2002; Nambu et al., 1998). Movement has been associated with simultaneous β ERD and γ ERS in the STN, GPi and the cortex, suggesting that synchronisation coordinates temporally those different remote network nodes (Cassidy et al., 2002). β ERD is present both during self and externally paced movements and is correlated to motor performance (Kühn et al., 2004), further confirming the ability of the STN to prepare and regulate movement dynamically. The generation of neural activity during self-paced movement also suggests that the neural rhythm modulations are driven endogenously in BG and are instead not caused by a response to an exogenous stimulus (Kühn et al., 2009).

STN-LFP are also modulated during isometric grip contraction (Spraker et al., 2007). Anzak et al. (2012) have first revealed a strong ERS coupling between slow rhythms θ, α and higher frequency γ occurring during the application of maximum voluntary contractions (MVC). Tan et al. (2013) have shown that γ rhythms scales with the application of high regime grip force (making for more pronounced ERS), and the β rhythms scales inversely with weaker (making for more pronounced ERD) and moderate grip force regimes. Low and α frequencies also tends to precede the start of any action.

Additionally Tan et al. (2015) have demonstrated that the scaling of β ERD is dependent on the amount of effort employed to deliver the demanded force level, specifically in relation to MVC, rather the absolute level of force itself, fitting the idea put forward by Kornhuber (1971) of the BG behaving as a ramp generator. The BG may control both kinematic and kinetic states through a common effort variable.

There are thus important and complementary neural-dynamic coupling of STN-LFP biomarkers that could be employed to quantify the effort required for complex motor actions. Since new kinds of couplings have been uncovered, particularly relating to the temporal dynamics of neural rhythms across hemispheres. A common practice in DBS surgery is to implant both left and right STNs irrespectively on the side most affected by the disease. Bilateral implantation makes for a straightforward study of bilateral motor tasks involving left and right hands or gait patterns. It was first assumed by Anzak et al. (2012), Tan et al. (2013) and then further confirmed by Fischer et al. (2017) that, as with cortical activity, β ERD and γ ERS are principally measured in the contral-lateral STN in respect to the side of the executed motor action, with activity in the ispi-lateral STN remaining stationary. Fischer et al. (2018) have further showed that those β ERD patterns are actively alternated during gait cycles from one contra-lateral hemisphere to another through recording PD patients standing or sitting while repeating stationary stepping motion.

Lastly, β ERD and γ ERS are also observable when actions are imagined and not executed, scaled again by the intensity of demanded effort (Fischer et al., 2017). These cognitive state changes further suggest that the CBGTC loop, particularly the STN, regulates motor-imagery processes that may prepare and rehearse future motor plans. It can be thus reasonably argued that motor imagery could be exploited to train BCI users.

In summary, neural activity changes within the CBGTC loop, which are necessary for the preparation, development, and regulation of motor processes, heavily rely upon synchronising various neural rhythms. This observation has been principally funded by severe synchronisation deregulation observed in the CBGTC loop caused by neurodegenerative conditions such as PD. Similar to the pathway dichotomy in the BG, synchronisation follows a similar duality with anti-kinetic β frequencies and pro-kinetic γ frequencies. The complementary nature of those biomarkers and the described connectivity and functional relevance of the STN is what make this deep brain nucleus a promising interfacing target for developing neural decoders.

E. DBS recorded LFP Decoding

The study of deep brain LFP biomarkers is a relatively emerging field, and new relationships with motor, cognitive and pathologic states are constantly being discovered. This survey catalogues studies demonstrating that those biomarkers can be successfully employed in a given BCI application. The elements of interest were: the type of states which have been already decoded, whether those were motor, pathologic or cognitive; the method employed for decoding; the number of participants enrolled and their condition; the degree of success achieved, both in terms of performance but also in terms of deployment capability. Answers to those contextual questions were necessary to help guide the development of this body of work and help frame its contribution to the field.

1. Survey Methodology

A systematic review approach was adopted for the survey. A scientific search engine (Clarivate Web of Science) was first employed to regroup relevant publications made in journals or conference proceedings. The engine query was designed to be initially broad: were desired all publications whose topic was first of all related to LFP recorded within the deep brain or the CBGTC loop (i.e., within the targets used for DBS to treat pathology related to motor symptoms, STN, GPi or GPe, thalamus and also the pedunculopontine nucleus (Lozano et al., 2019)) and which touch to a BCI application of sort (brain-computer, brain-machine or neural interface, systems performing detection, decoding, machine learning or which were adaptive or closed-loop).

The query returned 310 publications. 40 of them were first excluded as they were not relevant to the topic of interest but still stretched the interpretation of the query. 13 review articles were excluded. 20 publications focusing on pure simulation methods (primarily based on neuro-computational models) and 41 publications focusing principally on hardware design features (mainly linked to LFP recording) or related issues for DBS were further removed. 111 publications only studying biomarkers from deep brain LFP were excluded. More often, adaptive or closed-DBS and, more rarely, BCI were cited as potential applications. Those works, however, lacked the technical propositions or demonstrations for said applications. 33 publications only relied on animal models and were thereby disregarded. 4 studies employed ECoG arrays for signal decoding instead of deep brain LFP. 1 study also relied on EMG. 11 publications were judged to be closely related to similar publications made by the same authors. 1 publication was not in English, and 1 last publication was excluded because it only contained a single participant and was a case study. 34 studies remained to which 3 other known relevant studies were added (1 was repeatedly cited and 1 is undergoing review).

2. Results

The 37 studies included are summarised in Table 2.2. All (except for one) studies have been produced in the last decade, which is concomitant with the adoption trend of DBS surgery. On average past studies were based on small cohorts and have only enrolled on average 9±4 (mean±standard deviation) participants. As expected, an absolute majority of participants were treated for PD and the STN was targeted. Only in two instances the GPi (Islam et al., 2017; Mamun et al., 2015), and in one instance the pedunculopontine nucleus (Mace et al., 2014) were targeted in addition to the STN. Other pathologies were also studied. 2 instances focused on participants treated for neuropathic pain using DBS targets in the thalamus and the periaqueductal grey. Luo et al. (2018) focused on pain state detection in time and

Zhang et al. (2013) pain rate scaling from LFP. 2 instances employed participants treated for essential tremor using DBS targets in the thalamus to test action and tremor detection (He et al., 2021; Tan et al., 2019). 1 study was completed on participants treated for seizures (Toth et al., 2020). Zhu et al. (2020) also decoded seizures but from ECoG arrays and relied on STN signals to decode tremor. 1 study included participants treated for dystonia (Islam et al., 2017).

The type of decoded state variable in each study is also included in Table 2.2, as well as the associated performance reported by the authors. In most cases, those consist of classification metrics, such as classification accuracy or receiver-operator characteristic (ROC) area under the curve. Extreme care should be taken when comparing performances given the variability in cohort size, method, decoding task and collected LFP.

In 20 studies, healthy states were decoded from the recorded LFP. Two notable bodies of work produced from relatable authors and datasets stood-out. The work of Islam et al. (2017), Jameel & Mamun (2017), Mamun et al. (2015), Mace et al. (2012) focused extensively on discriminating resting states from actions and action laterality based on dual hand button-pressing tasks. Golshan et al. (2020, 2018, 2017, 2016) and Zaker et al. (2015) contributed extensively to behavioural task classification given segmented LFP recordings. Both bodies have implemented increasingly advanced and tailored machine learning techniques to increase decoding performances progressively. Very high performance (accuracy > 0.99) was reported for rest/action discrimination and laterality selection (accuracy > 0.85). The latest work of Golshan et al. (2020) showed that deep learning helped increase the overall population performance average (accuracy > 0.85). Loukas & Brown (2004), Mace et al. 2014, Niketeghad et al. (2018) and Zaker et al. (2016) contributions focused on continuous action detection and provided more mixt results, and direct comparison was more difficult to establish. Tan et al. (2016) was the first work addressing continuous state decoding by creating offline dynamical models mapping γ ERS and β ERD to isometric grip force contractions. Shah et al. (2018) then proposed a Wiener model capable of operating in real-time. Merk et al. (2021) recently explored alternative estimators for continuous decoding for STN and ECoG recorded LFP. Shah et al. (2018) has demonstrated strong performance results, although Merk et al. (2021) has claimed that STN-LFP decoding is far inferior to ECoG. Another theoretical modelling included grip force stage (Shah et al., 2018), gait state discrimination (Canessa et al., 2020; Tan et al., 2018) and action sequencing (Ruiz et al., 2014). Lastly, Chen et al. (2019) was the only work that decoded non-motor states and focused on continuous arousal monitoring.

16 further studies were dedicated to the decoding of pathological states. The landmark publication of Little et al. (2013) was the first to report a functioning β triggered closed-loop DBS system to treat PD bradykinesia and rigidity. He et al. (2019) studied how β bursting could be diminished through neurofeedback. Moreover, many studies were dedicated to detecting tremor from STN-LFP (Camara, Isasi, et al., 2015; Hirschmann et al., 2017; Zhu et al., 2020). Unlike rigidity or bradykinesia, no evident biomarkers have been so far associated with tremor. Instead, multi-feature

classifier approaches have been progressively adopted, such as those in movement decoding. Tremor detection has become more reliable, with performances reaching accuracy > 0.8. He et al. (2021) and Tan et al. (2019) were the only works decoding motion and tremor simultaneously. The action detection rate in the thalamus was shown to be high, with performances averaging a specificity > 0.85 (He et al., 2021). Lastly, Wang et al. (2018) focused on decoding motor impairment, Toth et al. (2020), seizure detection, Khawaldeh et al. (2020), the effect of α and β bursts on motion prediction and Sand et al. (2021) on monitoring of levodopa medication state.

There was a striking variability in adopted methodologies, with different groups having different preferences and argumentative inclinations on decoding pipeline design. Overall small-scale machine learning was preferred, with two notable exceptions Golshan et al. (2020) and Islam et al. (2017) which have opted for deep learning. Hand-crafted feature sets have been chiefly employed with a preference for time-frequency based features. There was a prevalence of the continuous wavelet transform (CWT) approach, which is a standard tool in electrophysiological analysis (Cohen, 2018) (Chapter 3.E.3). More complex and customised feature space were also proposed, including time features extracted on signal epoch windows, such as higher statistical moments, entropy (Bakstein et al., 2012; Toth et al., 2020; Wang et al., 2018), Granger causality (Mamun et al., 2015), local non-linear regression (Niketeghad et al., 2018), and cross-correlation (Bakstein et al., 2012) (Table 2.2). Methodology for selecting decoding contacts was very variable, and some studies avoided the topic, although both machine learning and manual selection have been employed (Table 2.2). In the case of adaptive DBS, the choice of recording channel was made conjointly with stimulating contact (He et al., 2021; Little et al., 2013; Yao et al., 2018). Moreover, the approach to real-time was inconsistent across studies. Only 31% of the studies surveyed proposed decoders deployed or that could be deployable in real-time. 36% did not specifically treat the subject in their methodology, but real-time could still be hypothetically achieved with some adjustments. The reminder 33% were more explicitly offline analysis or decoding. Two principal obstacles to real-time implementation were noted: the usage of pre-segmented data and computationally intensive feature extraction methods such as the CWT. As offline processing is not an option for a closed-loop DBS system, studies investigating that topic have proposed systems more likely to operate in real-time. Many of them are more straightforward and rely on threshold triggering. However, machine learning-driven decoders have been adapted for real-time purposes, and Zhu et al. (2020) went to a remarkable extreme to optimise the computational cost of the decoder algorithm for hardware embedding. More than half of the real-time studies have been published after 2017 (Figure 2.1a).

Lastly, decoding performance optimisation has rarely been addressed. Some works have optimised individual aspects of the decoder, such as feature selection or classifier parameters (Table 2.2). Only in the case of Mace et al. (2014) and Zhu et al. (2020) systematic performance optimisation was implemented.

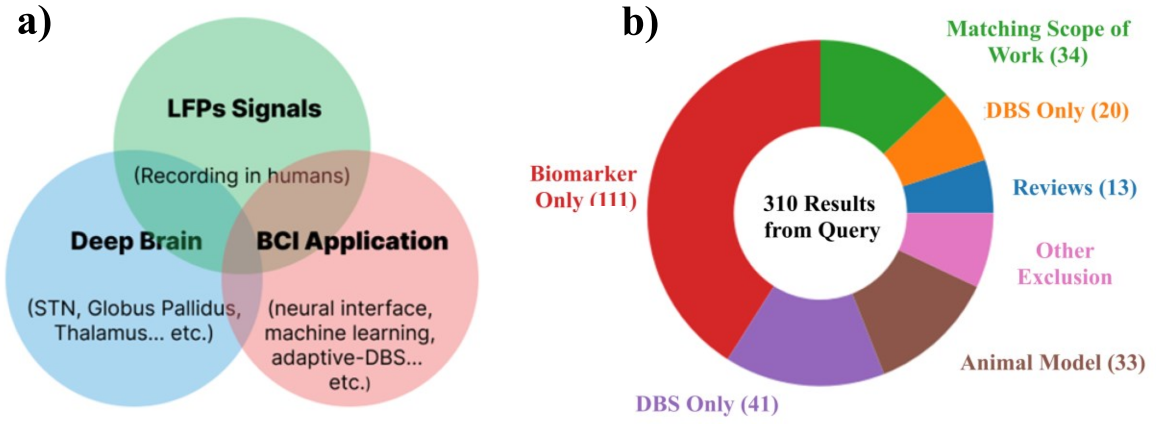


Figure 2.3 - Survey query and results; **a)** Diagram representing the query design to select relevant studies, i.e., an intersection of three different domains; 1) studies looking at invasive LFP recordings; 2) made in the deep brain; 3) looking at BCI like-applications; **b)** Breakdown of the 310 query results; only 34 studies matched the exact scope of work; others were excluded.

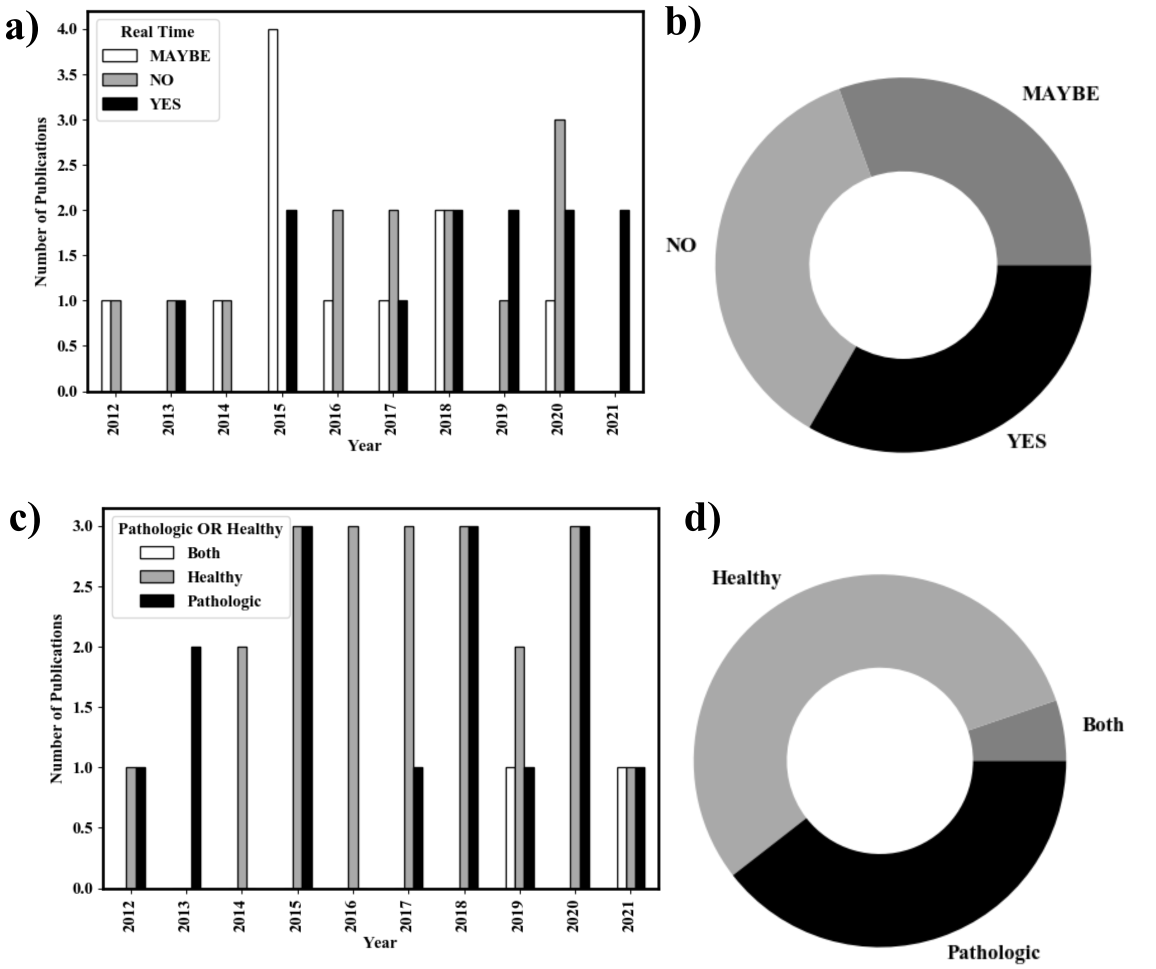


Figure 2.4 - Summary of selected studies according to real-time capabilities and type of state decoded; **a)** Previous studies achieving real-time conditions; **b)** Distribution of studies according to real-time constraints; **c)** Distribution according to the type of decoded states. **d)** Type of states decoded in previous studies.

Table 2.2 -Literature survey summary; Additional acronyms: neuropathic pain (NP), dystonia (DY), essential tremor (ET), seizure (SZ), anterior nucleus of the thalamus (ANT), ventral lateral posterior thalamus (VLP), ventral intermediate thalamus (VIM), periaqueductal grey (PAG), periventricular grey (PVG), pedunculopontine nucleus (PNN), healthy (H), pathologic (P), both (B), yes (Y), no (N), area under the curve (AUC); Studies were broken down in terms of included participants, The condition for which those participants were treated, and the motor state decoded in the study; Methodologies were broken down in terms of channel management, features extracted from the LFP signals, the type of machine learning strategy which was employed to relate the state to be decoded to those features and reported performance; Real-time capability of the of method and optimisation techniques were noted.

Reference	**Participant**	**Condition**	**Electrode Type**	**Electrode Location**	**Decoded State**	**Pathologic or Healthy**	**Overall Reported Performance**	**Channel Management**	**Feature Processing**	**Machine Learning Strategy**	**Real-Time**		**Optimisation**	
(Loukas & Brown, 2004)	3	PD	DBS	STN	Movement with joystick.	H	0.95 sensitivity; 0.97 specificity.	Selection on highest power.	FFT (12-27Hz); CWT (Meyer); temporal statistics.	Learning Vector Quantisation (LVQ).	Y	With FFT or temporal statistics.	N	
(Mamun, Mace, et al., 2012)	5	PD	DBS	STN	Button press; Action detection; Laterality selection.	H	0.99 detection; 0.83 laterality accuracy.		DWT (Meyer)-packet; Hilbert transform for power band estimation; Granger causality.	Sequential weighted feature selection; SVM or Bayesian classifiers.	M	Pre-segmented training data, lack of validation; Burden of Granger causality; Latency introduced by Meyer's wavelet.	Y	Optimisation of feature subset; Classifier parameters.
(Bakstein et al., 2012)	8	PD	DBS	STN	Tremor detection.	P	0.80 accuracy (with noted variance).	All channel employed.	Epoch average, magnitude, variance, entropy; FFT and CWT tremor power; Auto-correlation features; Magnitude of DWT detail coefficient.	Neural network classifier, both subject-specific and population-wide models.	N	Burden of feature extraction.	Y	Feature selection based on mutual information, but neural network parameter tuning avoided.
(Zhang et al., 2013)	6	NP	DBS	VIM PAG	Pain intensity.	P	0.2-1.0 correlation.		CWT (sym10)	Fuzzy logic classifier.	N	CWT burden.	N	

(Little et al., 2013)	8	PD	DBS	STN	Bradykinesia and rigidity (beta burst).	P	Improved symptom scores.	Configure for stimulation and recording.	Rectified bandpass beta band power.	Manually set threshold.	Y	System validated in real-time.	N	
(Mace et al., 2014)	3	PD	DBS	STN PPN	Body orientation.	H	0.85 sensitivity; 0.69 specificity	Re-optimised on every channels.	Short-term energy in DWT (Meyer) coefficients.	Adaptive threshold.	M	Latency introduce by Meyer's wavelet.	Y	Multi-objective genetic algorithm to tune threshold, filter length and down sampling ratio.
(Ruiz et al., 2014)	12	PD	DBS	STN	Action sequencing.	H	Above chance prediction.	All channel employed.	CWT (Morlet) power band.	SVM	N	CWT burden.	N	
(Mamun et al., 2015)	12	PD	DBS	STN GPi	See (Mamun, Mace, et al., 2012)	H	0.998 detection; 0.82 laterality accuracy.		See (Mamun, Mace, et al., 2012).	See (Mamun, Mace, et al., 2012).	M	See (Mamun, Mace, et al., 2012).	Y	See (Mamun, Mace, et al., 2012).
(Mohammed et al., 2015)	9	PD	DBS	STN	Bradykinesia and rigidity.	P	0.70% accuracy.	Test monopolar against bipolar settings.	FFT band power estimation.	SVM with different kernel types.	M	Lack of validation.	Y	Kernel selection.
(Camara, et al., 2015)	7	PD	DBS	STN	Tremor detection; Clustering tremor types.	P	0.832 accuracy with cluster labels.		DWT coefficient energy, average, variance, first derivative, entropy; Selection based on correlation.	Clustering with K-means to identify tremor types and label data; Neural network for detection.	M	Provided adequate epoching.	N	

(Zaker et al., 2015)	12	PD	DBS	STN	Behavioural task (motor, speech, motor, and speech).	H			Matching pursuit decomposition; K-mean clustering or GP mixtures.	Supervised HMM-SVM with top level SVM.	M	Provided adequate epoching and reasonable computation efficiency.	Y	Parameter grid search; Feature clustering.
(Zaker et al., 2016)	7	PD	DBS	STN	Action onset.	H	0.84 accuracy.		See (Zaker et al., 2015).	See (Zaker et al., 2015).	M	See (Zaker et al., 2015).	Y	See (Zaker et al., 2015).
(Golshan et al., 2016)	5	PD	DBS	STN	Behavioural task (motor, speech, motor, and speech, reach).	H	0.70 accuracy.	PCA compression.	CWT (Morlet); PCA compression.	See (Mohammed et al., 2015).	N	Pre-segmented data; CWT burden.	Y	See (Mohammed et al., 2015).
(Tan et al., 2016)	9	PD	DBS	STN	Grip force transient response.	H		Manual selection on beta ERD.	CWT (Morlet).	Dynamic model with different ERS/ERDs combination.	N	CWT burden.	N	
(Golshan et al., 2017)	5	PD	DBS	STN	See (Golshan et al., 2016).	H	0.60 Accuracy.	Selection based on FFT synchronisation.	CWT (Morlet) beta band power estimation.	See (Mohammed et al., 2015).	N	See (Golshan et al., 2016).		
(Hirschmann et al., 2017)	12	PD	DBS	STN	Tremor detection.	P	0.82 ROC-AUC		Multi-taper FFT, band power estimation.	HMM	Y	Provided efficient computation of taper.,	N	
(Jameel & Mamun, 2017)	4	PD	DBS	STN	See (Mamun, Mace, et al., 2012).	H	0.78 accuracy.	Bipolar setting selection on beta ERD and location.	See (Mamun, Mace, et al., 2012); Additional use of discrete cosine transform.	See (Mamun, Mace, et al., 2012)	M	See (Mamun, Mace, et al., 2012).	N	

(Islam et al., 2017)	12	PD; DY	DBS	STN GPi	See (Mamun, Mace, et al., 2012).	H	0.89 DP 0.82 DY accuracy.		See (Mamun, Mace, et al., 2012).	Ensemble classifier, multi-layer perceptron, radial-basis neural and probabilistic network.	M	See (Mamun, Mace, et al., 2012).	N
(Niketeghad et al., 2018)	11	PD	DBS	STN	Finger movement detection.	H	0.7 ROC AUC.	Selection based on ROC-AUC of feature-space PCA components.	Non-linear regression on epochs.	Template matching.	Y		N
(Tan et al., 2018)	16	PD	DBS	STN	Step sequence.	H		Bipolar setting, selection based on beta ERD.	CWT (Morlet) power band estimation.	Multi-variate AR HMM.	N	CWT burden.	N
(Wang et al., 2018)	12	PD	DBS	STN	Motor impairment.	P	0.70 accuracy.	All channel employed.	Different band power; Phase coupling; Beta complexity; Tremor power; Peak power; Wavelet entropy; Sample entropy; Hjorth parameters; Coherence; ANOVA feature selection.	K-mean generated; LDA; Extreme gradient boosted trees, Multi-layer perceptron.	M	Lack of validation.	N
(Golshan et al., 2018a)	9	PD	DBS	STN	See (Golshan et al., 2016).	H	0.75 accuracy.	Selection based on FFT-synchronisation	See (Golshan et al., 2016).	Hierarchy of SVM-kernel classifiers.	N	See (Golshan et al., 2016).	N

(Shah et al., 2016)	14	PD	DBS	STN	Isometric grip force stages.	H	0.75-0.85 ROC-AUC	Contra and ipsi-lateral comparisons.	CWT (morlet) band power estimation; Hjorth parameters.	Logistic regression.	N	Pre-segmented training; CWT burden.	N	
(Luo et al., 2018)	6	NP	DBS	VPL PAG PVG	Pain reception scale.	P	0.80> specify; sensitivity.		DWT coefficients (test different type of mother wavelets).	Adaptive threshold.	Y	Efficient extraction with DWT; Simple state estimation with thresholds.	Y	DWT optimised based on tree-node entropy; Grid search for threshold parameter.
(Shah et al., 2018)	8	PD	DBS	STN	Isometric grip force.	H	Up to 0.79 correlation.	Selection based on Gamma ERS; All channels comparison.	DWT, AR, and STFT comparisons for power band estimation; Epoch average; Higher order moments.	Comparison of Wiener, Kalman filters and neural dynamic network.	Y		Y	Search for best output decoding rate and Wiener window length.
(Chen et al., 2019)	13	PD	DBS	STN	Sleep stage classification.	H	0.90 accuracy.		STFT band-power estimation with Welch averaging.	SVM and decision trees.	Y		Y	Parameter grid search.
(He et al., 2019)	3	PD	DBS	STN	Beta power.	P		Manual selection.	Beta bandpass power.	Manually set threshold.	Y		N	
(Tan et al., 2019)	16	ET	DBS	VIM VLP	Voluntary movement.	B	0.8 sensitivity; 0.2 specificity.	Bipolar settings classifier retrain for every channel.	CWT (Morlet) band power estimation.	Logistic regression.	N	CWT burden.	N	

(Yao et al., 2020)	12	PD	DBS	STN	Tremor.	P	0.89 F1-score.	Edge contact to avoid DBS artefact.	Band power estimation; Hjorth parameters; Kalman filtering and ANOVA feature selection.	Comparison multi-layer perceptron LR, SVM-kernel, LDA, KNN, extreme gradient boosting and random trees.	M	Assuming fast power estimation.	N	
(Khawaldeh et al., 2020)	18	PD	DBS	STN	Upper and lower limb movement.	H	0.80 accuracy.	Manual selection.	CWT (Morlet), alpha and beta burst detection.	Gaussian naïve bayes classifier.	N	Offline analysis, CWT burden.	M	
(Golshan et al., 2020)	12	PD	DBS	STN	See (Golshan et al., 2016).	H	0.88 accuracy.	Employed all channel within the neural network.	CWT (Morlet)	Convolution neural network.	N	Pre-segmented training data; Burden of the CWT; Burden of the network.	Y	End-to-end optimisation.
(Canessa et al., 2020)	8	PD	DBS	STN	Walk and rest states.	H			CWT (Morlet) power spectrum; Gaussian fitting.	Logistic regression.	N	CWT burden; Spectrum fitting; Offline analysis.	N	
(Toth et al., 2020)	10	SZ	Depth Elec.	ANT	Seizure onset.	P	0.973 accuracy	Deepest Contact.	DWT coefficient power; Multi-scale entropy.	Line length triggering; Comparison of random forest and random kitchen sink classifiers.	Y		N	
(Zhu et al., 2020)	12	PD; SZ	DBS; ECoG	STN; Cortex	Tremor; Epilepsy; Finger movement.	P	0.8 tremor; 0.8 epilepsy; 0.6 finger movement accuracy.	Employed all channels.	FIR bandpass power estimation; Hjorth parameters.	Oblique gradient boosted classifiers.	Y		Y	Model computational burden optimisation.

(Merk et al., 2021)	10	PD	DBS; ECoG	STN Cortex	Isometric grip force.	H	< 0.1 coefficient of determination	Comparison between all channel and best channel selected by model.	Bandpass power estimation.	Elastic-net regression, Wiener; Gradient boosted trees, neural-network.	Y		Y	Hyper-parameter tuning with Bayesian optimisation.
(He et al., 2021)	8	ET	DBS	VIM Zi	Tremor and voluntary movement.	B	0.85 sensitivity for movement.	3 recording channels, 1 stimulation channel.	STFT band power estimation, Hjorth parameters.	Comparison of LDA, SVM, NB, LR, hierarchic extreme learning decision trees.	Y		Y	
(Sand et al., 2021)	4	PD	DBS	STN	ON/OFF medication state.	P	0.80 accuracy.	Selection based on location.	Welch method beta and gamma power band estimation; bursting features with thresholding.	SVM	Y		Y	Channel selection based on HMM, SVM parameter grid search.

F. Discussion

More and more contextual and direct evidence has demonstrated the importance of the deep brain in assisting cortical motor areas to plan, initiate, and control volitional movement. Coincidently research based on DBS recordings has been also pointing towards the role of population-wide synchronisation to support a circular and continuous communication between the cortex and the deep brain through highly specialised pathways. Synchronisation is an important biophysical basis that facilitates information transfer through these pathways and correlates with measurable neuronal rhythm modulations in the LFP signals.

Therefore, the development of neural decoders for deep-BCI hinged on several questions: are these signal modulation biomarkers sufficiently strong and persistent in STN-LFP to serve as a reliable modality for interfacing? Can the information vehiculated through the STN be translated into intended motor actions? Can STN-LFP decoding be achieved to support real-time control of and, thus, seamless interfacing with peripheral devices? To which extent is it deployable, and for which relevant applications?

1. Comparison between Deep and Cortical Brain Interfacing

State-of-the-art MEA-driven BCIs have enabled individuals to gain remote control over complex devices such as anthropomorphic robotic manipulanda (Collinger et al., 2013; Wodlinger et al., 2015). This control is nearly instantaneous, continuous in time, and continuous across multiple degrees of freedom. ECoG-driven BCIs have demonstrated impressive capabilities, including complex gesture recognition (Obermaier et al., 2001), cursor control (Leuthardt et al., 2004), or even control of an exoskeleton applications (Benabid et al., 2019). Lastly, EEG has demonstrated good capabilities in similar applications, albeit with typically lower performances, in simpler decoding problems for lesser-complex target devices and longer user training times.

There is undoubtedly some continuity between cortical and deep brain anatomy, organisational patterns, functionality, and, most importantly, circulation of common synchronisation patterns. Research has notably showed the importance of anti-kinetic β and pro-kinetic γ neural rhythms, the principal modality extracted from non-invasive EEG, ECoG, and MEA recorded LFP signals and conventionally decoded for motor-oriented BCI systems. It was thus reasonable to expect DBS-recorded signals to be suitable for applications covered at the very least by EEG, potentially ECoG, and less likely MEA. The latter would have been the most difficult to compare, given that DBS leads were still macro-electrodes at the time of writing and could not record single-unit spiking nor spatial high-density LFP activity, at least without significant amounts of advance processing and specialised datasets.

Moreover, the survey has shown that pattern recognition to decode discrete state changes from DBS-recorded LFP signals has been the main focus of BCI research in the deep brain, with variable degrees of success. The low dimensionality of the decoded state spaces was noted with very few attempts to handle complex multiclass problems. Binary discrete state decoding (e.g., rest vs. movement, left vs. right, no-symptom vs. symptom) has been the object of most studies, with some exceptions of multi-state discrimination approaches. Typically, those studies have attempted to solve classification problems based on STN-LFP biomarkers that serve as a feature representation space of the signals. Work on continuous state spaces has also been limited to only three studies that carried decoding across a single degree of freedom. The development path of DBS- driven BCIs has not been able to catch up with their cortical-LFP counterparts. No examples of a 2D cursor BCI have been proposed. There has also been no attempt to train users to control an interface and improve performance.

Part of the explanation could be that decoding pathological states, which could permit an effective closed-loop DBS strategy, has been driving research. Alternatively, it could be that underlying technical challenges are hindering progress, or even the nature of DBS-recorded LFP signals cannot ultimately permit similar applications to those possible with the cortex. There were thus motivations to research further, on the one hand, the technical aspects of DBS recorded LFP decoding and, on the other, the fundamental electrophysiological behaviour of the STN. These knowledge frontiers were to be pushed if the benefits, described in Chapter 1, of DBS-driven BCI were to be unlocked.

2. Diversity of Decoding Strategies and Cohesive Architectural Framework

The survey results showed that a reasonably wide range of approaches had been adopted for deep-brain state decoding. Every study, and understandably so, did not weigh the different aspects of the decoder pipeline in the same way, partly to suit different types of applications. Many of the best-performing decoders have employed complex techniques, to name a few, advanced feature selection (Mamun et al., 2015), advanced signal decomposition (Zaker et al., 2016), extensive feature space construction (Bakstein et al., 2012), genetic algorithms optimisation (Mace et al., 2014), and even classifier structure optimisation (Zhu et al., 2020). It can be conceded that even the simpler decoding pipeline requires a stack of diverse and non-trivial algorithms. However, the complexity adopted in the surveyed decoding solutions was not necessarily present because of the complexity of the decoding task itself (e.g., multiple degrees of freedom, continuous state decoding, or online adaption for training). Performant EEG (Wolpaw and McFarland, 2004) or ECoG (Gruenwald et al., 2019) systems employ far simpler feature extraction and state estimation methods in comparatively more complex tasks.

Two main arguments have been used to justify this complexity. First, LFP biomarkers are particularly subtle within the signals and thus require advanced processing to be adequately isolated. For instance, Mamun et al. (2015) and Niketeghad et al. (2014) have resorted to windowed non-linear regression and granger-causality models to measure activity differentials between hemispheres. Second, given the various biomarkers present in the STN-LFP signals and their relatively complex intrinsic dynamic, advanced methods, such as, iterative feature selection (Mamun et al., 2015) or full spectrogram multi-channel decomposition (Golshan et al., 2018), were the only significant means to increase performance. Similar arguments have been made to justify the additional complexity overhead required for extensive decoder parameter tuning. Decoding performance can be optimised, and leaner deployment solutions can also be produced (Zhu et al., 2020). Although the proposed methods addressed at least one of these points, none covered them simultaneously. Consistent gaps were thus identified in individual solutions, particularly regarding lean feature extraction supporting real-time realisation, consistent channel management, or decoding performance optimisation for deployment.

Furthermore, the broadness of the resulting method catalogue, the variety of problems tackled, and the complexity of the individual solutions motivated the development of a single and cohesive architectural framework to design and implement neural decoders (Chapter 3). This framework provided the structural backbone required to support the intrinsic complexity of any elementary decoder pipelines. The enjoyed formalism first helped devise the decoder architecture consistently across different identified problem classes and apply the proper constraints to these problems. Different solutions could also be grafted onto the framework, enabling consistent comparisons. These could range from what had already been proposed in the literature, including other conventional BCI techniques, and permitted the addition of innovative and customised propositions. Given the interesting contributions of Mace et al. (2014), decoder optimisation was an important point of interest. At the time of writing, it was a topic less addressed in the broader BCI literature yet increasingly relevant to deployment in real-world applications. Such an approach, however, required a framework to help define the different parameter groups and orchestrate their respective optimisation according to a common objective function.

The decision to create a specialised framework also came after reviewing other pre-existing BCI programming frameworks, libraries, and environments (such as OpenVibe (Renard et al., 2010), BCILab (Kothe and Makeig, 2013), and Fieldtrip (Oostenveld et al., 2011)). Most of these frameworks are geared specifically towards EEG. It was difficult to predict where the similarities between EEG and LFP would end and the limitations that could then arise. Thus, it made sense to tailor development to DBS-recorded LFP, the norm followed by the surveyed literature so far. Even though some compelling features and functionalities were proposed, none matched sufficiently other aforementioned needs and did not exhibit sufficient flexibility. As seen in more recent studies (Merk et al., 2022; Zhu et al., 2020), a shift occurred with researchers moving more and more from traditional engineering software environments (such as Matlab) supporting these pre-existing frameworks, towards the Python language. Most new research developments in machine learning were, at the time of writing, distributed through Python packages such as scikit-learn (Pedregosa et al., 2011) or PyTorch (Paszke et al., 2017). Python is entirely open-source, permitting a more in-depth third-party code inspection. The ecosystem for BCI research in Python was still nascent; thus, space remained for more innovation and the development of specialised toolkits and frameworks.

3. Shortcomings of Current Paradigms: Real-Time and Asynchronous Decoding

Moreover, there has been a convergence towards employing more real-time compatible methodologies, specifically pushed by studies addressing closed-loop DBS problems. Major STN-LFP works have been somewhat lagging in terms of real-time deployment. A possible criticism of Islam et al. (2014), Jameel & Mamun (2017), Mamun et al. (2015, 2012), Golshan et al. (2020, 2018, 2017, 2016) and Zaker et al. (2015) was the lack of measures to improve and validate the real-time deployment of the propose decoding pipeline. Offline approaches can be problematic for gauging real-world deployment accurately, as performance can be overestimated. Feature extraction was identified as a principal limiting factor in many studies. Many relied on offline feature extraction technics such as Morlet's CWT (Canessa et al., 2020; Golshan et al., 2017, 2016; Khawaldeh et al., 2020; Ruiz et al., 2014; Shah et al., 2016; Tan et al., 2019, 2018, 2016). These technics can either be too computationally demanding or introduce important processing lags.

A second problem was associated with BCI asynchrony[3] (Nicolas-Alonso and Gomez-Gil, 2012; Niketeghad et al., 2018). Whereas a synchronous BCI system is generally constrained by predefined external timings and analysis windows (Nicolas-Alonso and Gomez-Gil, 2012), an asynchronous interface should continuously stream in neural signals and broadcast the decoded output without prior knowledge of the temporal onset a state change. The survey noted that many past STN-LFP studies had implemented strategies circumventing the asynchronous problem. First, certain studies constrained themselves to asynchronous frameworks Golshan et al. (2020, 2018, 2017, 2016), Zaker et al. (2015) and only dealt with pre-segmented training examples. Although Islam et al. (2014), Jameel & Mamun (2017), and Mamun et al. (2015, 2012) tackled an action detection problem by classifying resting and action states, only pre-segmented training examples were employed in model training with no final validation and demonstration on complete timeseries. Other studies that accepted the premise of asynchrony as a design specification showed comparatively more limited results (Bakstein et al., 2012; Mace et al., 2014; Niketeghad et al., 2018, 2014; Yao et al., 2020; Zaker et al., 2016). In particular, Niketeghad et al. (2018) concluded that STN-LFP had a limited signal-to-noise ratio (SNR), making detection difficult. Mace et al. (2014) specifically assumed that significant movement, such as body rotation, was required to generate detectable and reliable activity changes in the STN-LFP. Bakstein et al. (2012) and Niketeghad et al. (2015, 2014) also noted disparate performance across participants. The approach chosen by Zaker et al. (2016) provided better results but was judged to be more challenging to reproduce. Another limit of Islam et al. (2014), Jameel & Mamun (2017), Mamun et al. (2015, 2012) was that the decoded state was only short button pressed actions. Decoding sustained, long and complex

[3] Unfortunately, this terminology overlaps semantically with the electrophysiological LFP synchronisation phenomenon (discussed in Chapter 2.C) and refers to how the BCI operates instead. Another possible categorisation could be temporally continuous decoding, which was not retained as it overlapped with the terminology used to differentiate between discrete against continuous state decoding.

actions has been comparatively more difficult so far (Niketeghad et al., 2015, 2014; Tan et al., 2019; Zaker et al., 2016) in part because the different components of the action need to be simultaneously detected, such as action onset, action holding, and action offset (Martineau, 2017; Shah et al., 2016).

The capability for STN-LFP to support robust asynchronous and real-time decoding remained an important open question, and validation was still required, especially concerning the relationship between STN-LFP feature SNR and decoder performance. Again the end objective was to be capable of performing bulk motion decoding to control instantaneously simple target devices (Martineau, 2017). Such an objective is relatively trivial for other invasive modalities (and, to a lesser extent, with EEG) and prompted the work on bulk and sustained motor action detection (presented in Chapter 4).

Many of the solutions subsequently proposed in this work aimed at overcoming the technical barriers associated with asynchronous and real-time decoding. Following the work of Shah et al. (2018), much emphasis was first placed on feature extraction by implementing past and new real-time enabled methods (which cumulated into a significant repository, Chapter 3.E, Appendix D). Given the reported low SNR of DBS recordings, there was also a vested interest in improving feature quality. Feature optimisation, in particular, through decoder parameter tuning, was judged to be a right step in this direction and further comforted the idea of a specialised architectural framework for decoder design. The constraints associated with real-time and asynchronous decoding were consolidated directly within the architecture (Chapter 3). The asynchronous constraint also impacted how the decoding problems were defined and the objective functions used for optimisation (Chapter 4, Chapter 5).

4. Improving Continuous Decoding: Associating Biomarkers to Correct Motor State

Continuous state decoding is important for all applications requiring precise motor control but is, as one would expect, more challenging in practice. Merk et al. (2021), Shah et al. (2018), and Tan et al. (2016) stood out as the only studies considering continuous problems, i.e., finding a mapping between continuous variation in delivered force from variations of biomarkers. These studies attempted exclusively to solve a regression problem between the STN-LFP biomarkers and the continuous grip timeseries, albeit with varying success. It was unclear how to proceed from hereon. For instance, moving to a 2D cursor control task would have been incredibly challenging, given the limited pre-existing results and experimental paradigm setup.

It was observed in the broader BCI literature that neural biomarkers have been associated with motor states of different kinematic and kinetic order. Motion decoding with MEAs has been primarily achieved by associating population spike firing rates with velocity (Collinger et al., 2013; Hsieh et al., 2017; Wodlinger et al., 2015). Isometric force has also been decoded similarly (Sergio, 2002; Sergio and Kalaska, 1998). Todorov (2000) extended these traditional models by positing that certain neural population biomarkers also encode both accelerations and positions (or at least a relation between all kinematic variables with the addition of force). Since the combined position and velocity models have been proposed in Kalman filter-based designs (Li et al., 2009), not only for spike unit decoding but also for LFP activity patterns (Hsieh and Shanechi, 2016). Relating different motor states to the correct biomarkers thus matters. It could be an important path to improve decoding and, more fundamentally, could help delineate previously unknown couplings between neuro and motor control dynamics.

These two premises thus provided more reasonable research goals centred around the following idea: improved model representation by augmenting the motor-state space with a first-order derivative for isometric grip force, otherwise known as a yank. Previous findings supported that the STN may be involved in yank regulation, and that specific LFP biomarkers could be sensitive to yank and force. In imaging studies, Spraker et al. (2007) and Vaillancourt et al (2007, 2004) have first uncovered a tight correlation between increased yank delivery and reduced STN activity. In LFP, Anzak et al. (2012) have found that mean positive power changes in the θ, α and γ domains were correlated to mean force level and to the peak of the yank impulse during force onset. However, unlike in Vaillancourt et al. (2004) the type of hand-grip force trajectory employed in LFP research has so far only consisted of step-responses to a demanded force level, a steady-state stabilisation at and holding of the demanded level, followed by a fast return to rest. Although this type of motor action has proven to be reliable to explore the relationship between grip force and STN-LFP response (Anzak et al., 2012; Tan et al., 2013), it is more limiting when assessing the relationship with yank. The yank impulse tends to be relatively short relative to the onset and holding of the grip force. If STN-LFP signals are more sensitive to yank than force level, it could help explain part of the difficulties encountered in decoding continuous grip force. In the current paradigm, yank only contributes transiently to the motor action. Another question actively discussed was the symmetry of the biomarker response. In previous studies, the response of the LFP signals was primarily studied at the onset of the grip force and never at release.

These observations further motivated a third set of research contributions to this body of work presented in Chapter 5. In practice, however, decoupling the yank and the force regimes can be difficult. No natural isometric trajectories are equivalent to simple movement profile descriptors such as bell-shaped velocity curves (Flash and Hogan, 1985) with well-defined derivatives. Further experimental innovation and alternative paradigms similar to that proposed by Vaillancourt et al. (2004) were required to study the STN-LFP temporal dynamics in complex motor tasks. Thus, different trajectories were investigated to guide force regulation and tested on participants. As aforementioned, the purpose of the new paradigm was to directly associate a given LFP biomarker to either yank or force and explore the local temporal variations and symmetries of the biomarker signals in regards to tightening and release. The resulting dataset also provided an opportunity to extend continuous decoding research in more dynamic tasks. If successful , yank and force models could be then integrated to enhance decoding performance.

Decoded force and its first-order derivative could be a useful control space in applications requiring accurate voluntary motor-action prediction, such as guiding a motion-assisting exoskeleton. Moreover, aside from kinetic and kinematic states, mechanical impedance is another important variable used to regulate motor activity. Modulating mechanical impedance (in particular stiffness) through muscular co-contraction is important in unstable dynamical environments and is a problem primarily relevant in human-like control systems. It has received accrued attention in the field of BCI, with several studies trying to integrate impedance concepts within their decoding models (Todorov, 2000). However, these approaches have not usually been integrated into state-of-the-art systems. Given its relation to Parkinsonian rigidity, it was posited that biomarkers highly informative of impedance could be found in STN-LFP signals. This observation motivated a last body of work (briefly reported in Chapter 6) which focused on constructing a new experimental paradigm with an active mechanical perturbation interface.

G. Chapter Summary

The generation and regulation of motion through the CBGTC loop are discussed in this chapter. The BG acts as a passive break and a motion amplifier to regulate the firing of neurons in the cortex. Diseases such as Parkinson's disturb the normal functioning of the CBGTC loop. DBS treatment involves disturbing hyper-synchrony of β rhythms within the loop at the level of the STN to help restore part of its functionality. Research has shown that LFP biomarkers, such as neuronal synchronisation, are measurable from DBS probes. So far, the patterns in synchronisation events carry some continuity to what has already been observed with LFP recordings made in the cortex and should allow for similar BCI applications. It can be reasonably argued that the same anti-kinetic β ERD and pro-kinetic γ ERS propagate through the CBGTC loop and are associable at the STN level with the regulation of effort, bulk movement, grip force and even motor-imagery processes, all of which relevant to motor state decoding and thus peripheral device interfacing.

A systematic review of past studies researching BCI applications for DBS-recorded LFP signals was completed. Progress has been substantial in the past decade, with 37 studies included in the survey. There has been balanced attention to both pathological and healthy state decoding using a wide variety of methods. No study, however, simultaneously addressed some of the principal technical problems associated with STN-LFP decoding. This first observation prompted the development of a coherent BCI architecture presented in Chapter 3 that supports full decoding solutions for different problem classes and parameter tuning capabilities. Secondly, a need for more validation for real-time and asynchronous decoder deployment was also identified, especially to establish that LFP biomarkers are sufficiently persistent for robust decoding. The architecture developed in Chapter 3 was first applied to a series of discrete-state decoding problems in Chapter 4. The problem investigated the detection of bulk motor state changes asynchronously in time. The decoder architecture allowed to test and compare decoding solutions and employed parameter tuning for optimal performance. Lastly, further investigation of the dynamical relationship between motor state and LFP biomarkers was identified as an avenue to improve continuous-state decoding. Investigating more complex dynamics required the development of new experimental paradigms and the collection of a new dataset, presented in Chapter 5. These paradigms mainly achieved force and yank level decoupling. Decoding models were then applied to these motor-state variables.

Chapter 3. Design and Architecture of STN-LFP based BCI for Online Decoding

A. Introduction

This chapter sets the specifications and lays the methodology for the design of STN-LFP decoders. Otherwise referred to as a pipeline, a decoder is a sequence of algorithms used to fully process the raw STN-LFP into control signals for a peripheral device. The proposed end-to-end structure of the decoder and the implemented algorithms utilised for pre-processing, extracting, and processing features, managing channels, state-estimator fitting, and decoder tuning, are presented. Cost for real-time deployment are also considered and discussed.

1. Challenges of an STN-LFP decoder

The nature of the DBS recorded STN-LFP datasets imposed certain constraints on the decoder design. Currently, participant cohorts of past studies have been relatively small (10 in average, Chapter 2.E.2). In practice, only limited renditions, or trials, of the experimental task are realistically collectable. Valid training points for the decoder algorithms are thereby sparse. In addition to its small size, DBS recorded LFP datasets are largely heterogeneous (Golshan et al., 2020; Tan et al., 2016). First, the electrode type is rarely the same for every patient, meaning that the numbers of contacts differ from one recording to another. Standard electrode leads have four contacts, whilst extended leads have 8. In comparison, the electrode layouts in EEG caps are standardised. Although there may be some morphologic variance, it can be safely assumed that EEG readings are spatially comparable and consistent from one subject to the other. To further complicate things, whereas MEA cortical coordinates are chosen to record a specific function (Wodlinger et al., 2015), the precise positioning of DBS electrodes is chosen to dimmish symptoms through stimulation (Mamun et al., 2015). There is no guarantee that the electrode will be inserted directly within the motor STN and not possibly in its periphery (Mamun et al., 2015). It should also be considered that the participants undergoing DBS surgery possess a complex neurodegenerative disease, and each of them may have different symptom manifestations of disease progression. Again, it has been shown that symptoms and medication have inherent effects on deep brain electrophysiology (Brown, 2003). The electrophysiological signals are complex processes, and the reliable isolation of ERS/ERD from the stationary-background noise stands as a difficult task. Even with the exact repetition of a particular motor task, it can be hard to reproduce comparable measurements of the same biomarker, making single-trial decoding and overall performance validation difficult. Overall, the DBS-LFP datasets were expected to suffer from substantial inter and intra-subject variability, which the decoder structure should attempt to accommodate.

2. Decoder Requirements

Some requirements were first laid to help draw the architectural framework first it described in Chapter 2.F, describe the overall decoder design and ultimately help address the specific challenges of STN-LFP decoding.

Task and user-specific: A series of motor tasks with their associated datasets (Appendix A) were divided for designing and testing STN-LFP decoders. Those were defined as ‘action detection,' ‘lateral end-effector selection’ (Chapter 4), and ‘continuous force decoding’ (Chapter 5). Specific performance metrics were defined for every task. The motivation was to compare the performance of different decoder architectures for a given BCI task and individual users. Given the inconsistencies across participants’ recordings, it was decided not to develop cohort-wide decoders (a single decoder for multiple users) (Lawhern et al., 2018). Data-augmentation and transfer learning (reusing pre-trained models) from other sources (such as large EEG or ECoG recording datasets from either human or animal models (Luciw et al., 2014)) did not seem straightforward to implement given the described specific characteristics of the available DBS recorded datasets (again mainly in terms of task design, electrophysiology, electrode-type and recorded brain region). Instead, emphasis was placed on decoder parameterisation and automatic parameter tuning to optimise performance of a single user on a given task.

Automatic parameter tuning: BCI parameters are commonly set through trial and error or best practices for a single dataset (Bashashati et al., 2016a, 2016b). The automatic parameter tuning problem has only been addressed by a few past studies (Mace et al., 2014; Zhu et al., 2020) and deserved further attention. It seemed to be the correct approach here. Given the specificity and the relative novelty of the STN-LFP datasets, those best practices were not yet fully known. An accelerated and robust approach was also preferable because of the limited time with patients, thus avoiding fiddling with parameter settings during experiment run-time. Automated parameter tuning supported the task and user-specific paradigm as it gave more flexibility to tailor the pipeline design. It could be helpful to integrate increasingly complex algorithms in the long run and remove many manual decisions. Automated decisions are trackable and can give an insight into the relationship between the different parameters and the task performance, ultimately providing feedback on decoder design choices. Lastly BCI user training is slow yet dynamic (Wolpaw and McFarland, 2004). Past studies have shown the importance of accompanying user learning by regularly and iteratively updating decoder parameters (Grado et al., 2018; Shanechi, 2017; Shanechi & Carmena, 2013). Slow tracking can be achieved if the method is iterative and regularly updated.

Real-time deployable: it was essential to ensure that the proposed BCI system could operate in real-time, especially for asynchronous decoding. It was noted that the translation from offline analysis to real-time processing was technically not trivial. Real-time automatically implies single-trial decoding under the constraints of streaming input data (where a new data point is made available at every refreshing cycle of the decoder). Pipeline operation needs to be causal, i.e. only depend on current or buffered data points. As summarised in the review, many different algorithms have been proposed, especially for feature extraction. Several methods proposed in past works had to be adapted to ensure that causality and filtering latency was considered. This approach permitted fairer comparisons between methodologies. Although the following work has been carried out offline from experimental recordings, the pipeline structure was devised to maintain causality. The maximum lag was kept within reasonable margins otherwise specified in the literature. The computational burdens of the implemented algorithms were considered and discussed as they can ultimately impose some hardware deployment constraints.

Modularity, Prototyping and Comparisons: past research has put forward custom decoding solutions, but rarely are those compared or even tested against one another. Various feature extraction methods, estimators, feature-selection, or dimensionality reduction techniques have been separately investigated. Pipeline modularity was essential to evaluate and test those different decoding methods. Each module unit, or stage, was interchangeable, used to assemble different prototype permutations and iterate designs. Modules were a useful abstraction: they permitted to store local parameters and fitting and transformation functions, define relationships between one another, and compartmentalise different aspects of the decoding problem. The Python machine learning ecosystem (Head et al., 2021; Lemaître et al., 2017; Paszke et al., n.d.; Pedregosa et al., 2011) inspired this approach. Shared templates throughout this ecosystem have significantly eased cross-library sharing. It was thus judged essential to make them part of the module design (Borton et al., 2020). It was hoped that those modules could coalesce into a toolbox that could be expanded and adapted to other works and applications. As the literature survey suggested, many different applications exist for DBS-driven BCIs, whether for decoding motor, pathologic or cognitive states or closed-loop stimulation. Some of the implemented methods may better serve some functionality over others. With the strategies implemented for real-time and automatic parameter tuning, those solutions could serve as a foundation for general BCI development.

B. Decoder Architecture, Fitting and Parameter Tuning

Given the requirements, the end-to-end structure was first planned according to the input it should receive, and the output it should produce. Principal sub-components of this structure were defined. Inputs and outputs signal characteristics (sampling rates, latencies…) were defined and constrained by different practical design specifications and considerations. Processing flows are then described for the overall architecture, fitting, and parameter tuning.

1. End-to-End Structure and System Specifications

The proposed end-to-end structure of the decoder is presented in Figure 3.1.

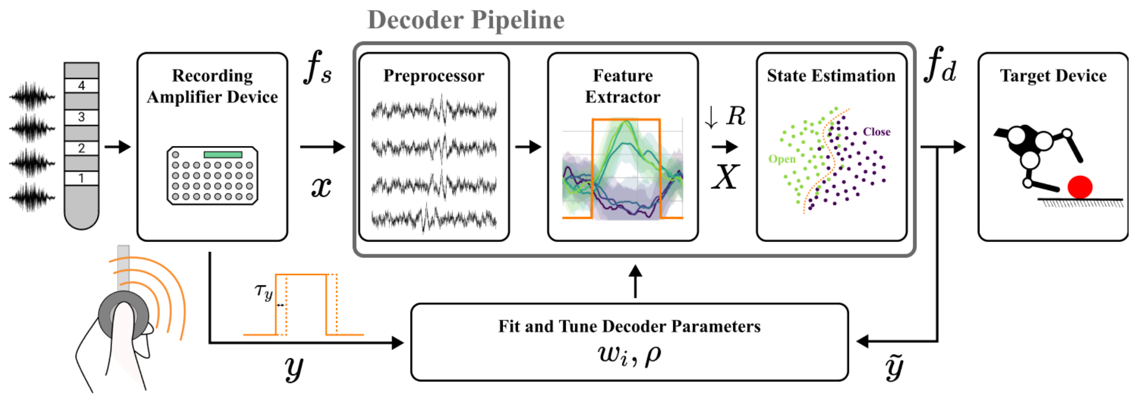


Figure 3.1 - End-to-end structure diagram of the decoder, from signal recording to target device; LFP signals x are acquired through a recording system at a rate f_s, here, a DBS lead connected directly to a TMSi-Porti (TMSi Ltd., the Netherlands); Labelling motor signals y are simultaneously acquired (here using a pinch dynamometer); x is first pre-processed to remove acquisition artefacts; Features are extracted and stored in array X; A state estimator is fitted to map X to y, then predicts the decoded state $\tilde{y}$; $\tilde{y}$ is refreshed at rate f_d by processing a signal block x of length R; Parameter groups w_i and hyper-parameters ρ are optimised during decoder fitting and tuning; τ_y shifts the label so that decoder is trained to estimate y a certain number of decoding steps in advance.

The pipeline was divided into the following standard stages:

Acquisition System: recorded the signals from the DBS electrodes. It consisted of a TMSi Porti (TMSi Netherland) and software drivers. It provided the decoder with a input signal x sampled at a frequency f_s and N_{ch} different channels. During experimental recordings, the acquisition system also recorded labelling signals y measuring a given motor state and used to fit and tune the decoder.

Pre-processor: treated the raw STN-LFP signals x by removing artefacts introduced during the recording, centring and scaling in preparation for feature extraction (Chapter 3.C).

Feature Extractor: isolated relevant features in the pre-processed signal x, later associated with the motor state to be predicted. This stage included a strategy to manage the channel space (Chapter 3.D), a time-frequency based feature extractor (Chapter 3.E), and some feature postprocessing transformations (Chapter 3.F). The resulting output are vectors of feature (N_f in total) stored into X.

State Estimator: included the principal machine learning stage, either a classifier (for discrete state estimation, Chapter 4) or a regressor (for continuous state estimation, Chapter 5). The estimator used a training dataset (composed of a pair X, y) to build a predictive relationship between the signal features and the motor state. The state estimator yield the predicted output motor state $\tilde{y}$ which was sampled at a lower output frequency f_d.

Parameter Tuner: during decoder fitting and tuning the label y and estimated output $\tilde{y}$ were used to assess the decoder performance. Fitting functions and optimisers were used to fit the various parameter groups w_i and hyper-parameters ρ stored in the decoder.

Table 3.1 – Specification for the decoder and selected constraints; These include input signal bandwidth, output refreshing rate, analysis window, total number of features and prediction horizon; The target hardware/software was a Python environment operating on Windows with an Intel(R) Core(TM) i7-7700K 4.20GHz CPU; (*) base 2 frequencies were chosen to easily support down sampling and were consistent with the sampling frequencies supported by TMSi-Porti (TMSi Ltd., the Netherlands); (**) N_{ch} the total number of channels varies from a participant's recording to another; All channels could be selected for decoding if necessary.

Specification Variable	Description	Constraint Value(s)
f_s	Input frequency of the decoder corresponding to the sampling frequency of the neural signals.	512Hz
f_d	Output frequency of the decoder.	4, 8 or 16Hz*
τ_d	Total decoder analysis window (3-39).	≤ 3s
N_c	Number of components or selected channels used for decoding (3-15) and (3-16).	$\leq N_{ch}$**
N_f	Total number of features used for decoding (3-40).	Indirectly constraint by N_c and τ_d.
τ_y	Decoder prediction horizon (3-4).	≤ 250ms
	Time to complete a computational cycle to update decoder output state.	$< 1/f_d$

As various decoding solutions were to be explored, the initial target for deployment remained a computer while maintaining capabilities for the algorithms to run online instead of directly targeting an embedded system. This choice placed less emphasis on selecting specific system hardware and low-level programming details, considering heavy constraints on memory allocations or limited computational capabilities. Embedded capabilities are constantly evolving, and a decoding algorithm can be deployed on various specialised layers of an application-driven architecture. Regardless, some broad processing specifications (Table 3.1) had to be drawn to ensure the architecture would be operational in a Python environment operating on a standard Windows machine (Windows 10, Microsoft Corporation, Intel(R) Core(TM) i7-7700K 4.20GHz CPU) alongside the digital acquisition system used for LFP recording (TMSi Ltd., the Netherlands). Algorithm training and fine-tuning time also needed to be carefully considered to be practically manageable.

A first specification layout for the decoder architecture was associated with the sampling frequency of the STN-LFP signals, f_s and the output frequency of the decoded state f_d. Both these specifications can alter how a decoder operates. A step-down R was always carried in the feature extraction stage such that:

$$R = \frac{f_d}{f_s} \tag{3-1}$$

f_s controlled the analysable bandwidth of the signal spectrum directly. A larger bandwidth implied that more information could be extracted from the signal at the expense of a faster computational rate, longer signal epochs to process, and more memory storage. Defining the correct bandwidth for the application was thereby critical. As previously mentioned in Chapter 2, LFP signals contain relevant biomarkers not only in lower frequencies domains from 1 to 60Hz but also in the fine-tuned γ bandwidth from 60 to 100Hz (Jenkinson et al., 2013). Although valid information content was also found in the higher-frequency domain from 100-375 Hz and even beyond (Gruenwald et al., 2017; Tan et al., 2013; Yao et al., 2020; Zhang et al., 2019), it was found in preliminary data analysis that this information, if not completely absent, was extremely weak in the available recordings (Appendix A). Consequently, a bandwidth 256Hz was sufficient to capture most of the information in the neural signal without unnecessarily burdening training and fitting processes. It also conveniently corresponded to the lowest sampling frequency (512Hz) which the recording amplifier supported.

Moreover, setting f_d defined how fast information can be transferred from the BCI towards a control device. Depending on the application, especially those geared towards communication, a high transfer rate can be desirable. Shanechi et al. (2017) has shown that a high output refresh rate of the decoding state could improve control accuracy. However, supporting fast decoding rates was unrealistic given the documented slow and noisy response of the STN-LFP signals (Tan et al., 2013). A high refreshing rate implied that many time points would have been treated throughout the decoder, dramatically slowing fitting operations and increasing overall calibration time. f_d was thus limited to 16Hz, which was in line with other studies (He et al., 2021; Shah et al., 2018), including those investigating spike decoding (Wodlinger et al., 2015). It also implied that a minimum of 62.5ms was present between decoding steps. This time interval was judged sufficiently long to complete a full decoding step. Most operating systems difficultly support smaller resolution timings (at least without specialised software tools).

Furthermore, given the real-time requirements, other specification constraints had to be implemented to ensure that the decoder would only use temporally and causally relevant time points to the current decoding time step. The decoder analysis window length τ_d (3-39) was defined as the overall past-time horizon the decoder used to estimate a state (Benabid et al., 2019). A reasonable constraint on τ_d was that it should only capture activity associated with an entire or a fraction of a single motor action. After considering the average cohort isometric holding times (Appendix A), it was decided that τ_d should not exceed 3s.

Limiting the number of past samples used to make a prediction also helped regulate the total number of features (associated with different biomarker signals), N_f (3-40), used to make a state prediction. N_f was also dependent on the number of channels used for decoding. Carrying feature extraction across multiple channels had important implications on processing times and deployment. Selecting channels or producing sparser channel components was therefore necessary. However, no initial limit was placed precisely on either the number of channel, sparse components or feature selected for decoding. It was unclear how these elements would influence decoder accuracy and were all points to investigate.

2. Processing Flow, Sequential Stage Fit and Backpropagation Approaches

Before designing each individual stage, general guiding principles were drawn to help define how the data should flow through the pipeline.

Firstly, any recording data STN-LFP was always stored in a multi-dimensional array $x \in \mathbb{R}^{N_k \times N_{trial} \times N_{ch}}$ to ease processing, where N_k is the temporal size, N_{trial} the number of trials and $N_{ch}$the number of channels. The first dimension of size N_k was reserved for time. The second dimension of size N_{trial} was reserved to concatenate different timeseries examples or trials alongside one another. A trial was defined as the fundamental repeatable unit of an experimental task. After every trial, a rest time was allowed to permit the STN-LFP activity to return momentarily to its baseline regime. Trial segmentation helped align and time-lock the STN-LFP response to a common reference (most often a cue presentation). For a signal x (and subsequently extracted feature signals), the trial average $\bar{x}$ was an essential visual aide to capture the non-stationary profile of the signal:

$$\bar{x}[k] = \frac{1}{N_{trial}} \sum_{l=0}^{N_{trial}-1} x[k,l] \tag{3-2}$$

Trial averages were considered the best measure of a signal ground truth. Segmented trials could be used instead of the full-timeseries to fit the decoder. Only relevant time segments were isolated for model training which helped constrain some fitting processes while reducing computational load. The drawback was then that out-of-trial information was disregarded. Depending on the task, this could be undesirable. Unless stated otherwise, the trial dimension is never explicitly mentioned when describing any process operating on x.

Moreover, the pipeline was conceived as a sequence of operations (Figure 3.2.a). For each stage, all N_k points along the temporal dimension of x were processed before passing the operation output to the next pipeline stage. In other words, it was easier to have a sequential approach to computation as opposed to a nested one.

At each i^{th} stage, the signal x_i array was transformed by a function h_i:

$$\begin{gathered} x_{i+1} = h_i(x_i, z_i, w_i) \\ z_i = u_i(x_{i+1}, x_i, z_i, w_i) \end{gathered} \tag{3-3}$$

where x_i and x_{i+1} are respectively the inputs and output data arrays of the stage, z_i the memory of the operator, and w_i the operation parameters (or weights). z_i ensured continuity between two refreshing steps and was updated in its own function u_i. The method also enabled the implementation of multirate processing when operating online. Incoming streaming data could be delivered in chunks of size $N_k = R$ and processed in block by operations carried at a rate f_s. After step-down, N_k is set to 1, which then corresponds to a single step process at a rate f_d. When operating offline, x_i was transformed from start to endpoint. z_i only mattered to initiate correctly the transform functions but was otherwise redundant. z_i and u_i then became inherent internal processes to the transformation functions.

Decoder stages could contain some fitting parameters (or weights) w_i. The pipeline was fully fitted once each parameter value was found from training examples. Each training example array x was associated with a target state array y (with $y \in \mathbb{N}^{\frac{N_k}{R} \times N_{trial}}$ or $y \in \mathbb{R}^{\frac{N_k}{R} \times N_{trial}}$ depending if the decoded state was discrete or continuous) and served as a data label, defined at a refreshing rate f_d. A small lead time τ_y was introduced in y for forecasting (Benabid et al., 2019; Yao et al., 2020). Relative to x, the label y was shifted such that:

$$y_s[k] = y\left[k - \lfloor \tau_y f_d \rfloor\right] \tag{3-4}$$

where y_s is the shiftted target. Staying a few computational times ahead was necessary to compensate for cumulated latencies in the system (possibly due to data transfer and processing). The lead time could also compensate for certain non-causalities (specifically in the feature extraction window). Given that none of the applications reviewed in Chapter 1 was very sensitive to timing, a forecast horizon of 250ms was judged to be amply sufficient for latency compensation.

Prior to model fitting, the data was split into two pairs consisting of a training set $(x_i, y_i)_{train}$ and of a testing set $(x_i, y_i)_{test}$. The first method consisted of fitting every pipeline stage sequentially, following the flow diagram presented in Figure 3.2b. The fitting function f_i yielded the parameters w_i for the i^{th} stage such that:

$$w_i = f_i(x_i, y_i) \tag{3-5}$$

The fitting function operated using an objective function optimisation or statistical estimations (such as, mean and standard deviation for timeseries scaling) locally defined to the stage. Once the parameters were found, the data was transformed to provide the next stage with the modified training set. During offline fitting, the model was trained on a single chunk of data was available such that z_i could be ignored.

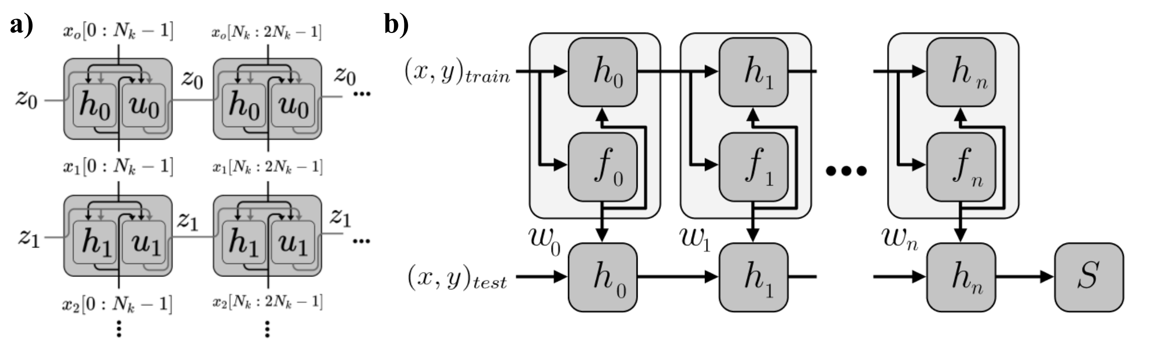


Figure 3.2 - Data flow during decoder realisation, fitting and tuning; **a)** Sequential realisation for real-time processing; Signals blocks (preferably of size R) are processed sequentially through the decoder; Memory arrays z_i for each transform are stored between realisation; **b)** Offline sequential fitting, transformation and scoring; each fit function f_i yields a parameter group w_i; The transform function is then applied to x_i and passed on to the next stage for fitting; Test branch occur subsequentially using the resulting parameter group w_i; Resulting performance is given by the criterion function S.

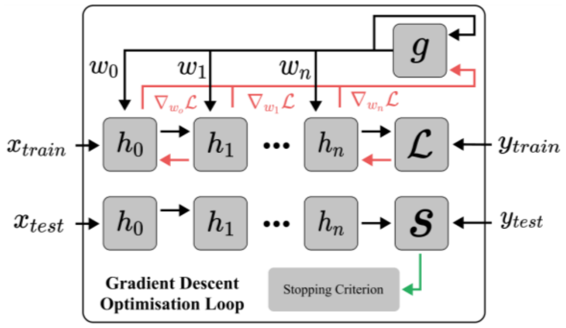


Figure 3.3 - Gradient-descent optimisation procedure; (**Black**) the training data x_{train} is processed sequentially; (**Red**) gradient from every parameter group is estimated by backpropagation from the end cost function $\mathcal{L}$; A gradient descent optimiser g is used to updated every parameter group w_i; (**Green**) a parallel x_{test} is used to monitor the descent through the decoder performance criterion S.

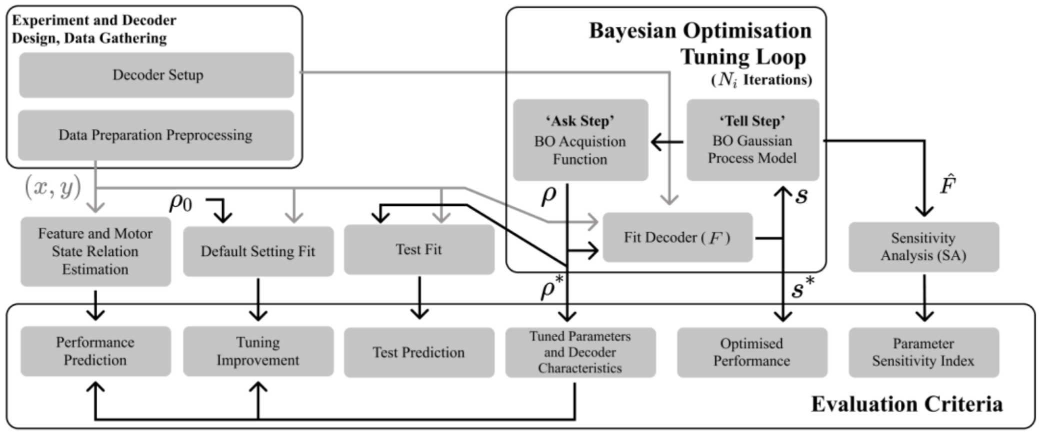


Figure 3.4 - BO for parameter tuning and decoder evaluation; The decoder is first setup (e.g., select extractor or estimator); LFP data x and labels y are prepared and passed on to the optimiser; The BO algorithm iteratively calls the decoder fitting function F; In the 'tell step' the optimiser maps ρ to s using $\hat{F}$, a gaussian process regression; In the 'ask' step an acquisition function uses the mapping to find the next ρ which maximises certain expected gains in performance (Appendix B); Various metrics are gathered at the end of the procedure for analysis.

It was also convenient to process y in parallel of h_i, for instance, to apply temporal shifting, downsampling, or feature space resampling. The n^{th} stage yielded the motor state prediction $\tilde{y}$:

$$\tilde{y} = h_n(x_n, y_n) \tag{3-6}$$

The overall pipeline fitting was finally evaluated on the transformed test set using the task performance metric function S. For a pair $(\tilde{y}, y)_{test}$ the scoring function returned a single scalar value s:

$$s = S\,(\tilde{y}, y) \tag{3-7}$$

The main limitation of this approach was that the stage parameters were found irrespectively of S. Each fitting function f_i solved a specific local problem unrelated to a global fitting objective. In particular, the feature extraction fitting was wholly decoupled from this objective.

Deep learning has been the object of ever-increasing attention in BCI research (Ahmadi et al., 2019; Golshan et al., 2020) and has bolster the development of many gradient based optimisation tools. Deep artificial neural network model fitting performs end-to-end optimisation orchestrated from a single objective function $\mathcal{L}$ which is minimised using gradient descent. The same approach was here employed to fit the decoder (Martineau et al., 2020). It was again defined as a sequence of the operations h_i, as described in Figure 3.3. h_n again yielded the state-estimation $\tilde{y}$ which was inputed into $\mathcal{L}$. Following the gradient descent procedure, the weights were recursively updated until a convergence criterion was met or a maximum number of iterations was reached:

$$(w_i)_{n+1} = g\left((w_i)_n, \left(\nabla_{w_i}\mathcal{L}\,\right)_n\right) \tag{3-8}$$

where g is the optimiser update step function and n is the current parameter update step. The decoder objective gradient $\nabla_{w_i}\mathcal{L}$ could have been derived manually, for example, as it is typically done for adaptive filters (Leuthardt et al., 2004). However, this went against the principle of modularity: in the case of any minor stage alteration, the entire gradient expression would have to be rederived. Instead, the gradient $\nabla_{w_i}\mathcal{L}$ was estimated through automated gradient-backpropagation numerical approximation (Paszke et al., n.d.). The choice of h_i were, therefore, limited to differentiable functions with known gradients $\nabla_{x_i}h_i$ and $\nabla_{w_i}h_i$. Gradient descent required repeating the forward computation pass of the pipeline to find $\mathcal{L}$ and a backward computation pass to find each $\nabla_{w_i}\mathcal{L}$. A foreseen advantage of the gradient descent approach was that the optimisation could be intermittently resumed to support online BCI training. Figure 3.3 further describes the training and testing routing. Again, the function S was always applied to the test output, evaluate the task performance and monitor the descent process. Unlike $\mathcal{L}$, S was not differentiable and could not be used as a starting point for the backpropagation.

In summary, the back-propagation approach supported end-to-end optimisation and, by extension, automated parameter tuning. Various layers of parameters could be composed to try and optimise the decoder performance. It was, however, more unconventional, complex, and much effort had to be invested in taming the gradient descent procedure. The decoder pipeline could also only be composed of differentiable functions. In contrast, the sequential-fitting approach was the most straightforward, conventional and robust Again, the drawback of this approach was some lack of malleability to different BCI users' recordings and to different tasks and the lack of performance objective optimisation.

3. Hyperparameter Optimisation

Hyperparameter optimisation permitted to work around this problem. Hyperparameters refers to parameters that cannot be directly set in the fitting stage because they are directly related to the evaluation of S. Those hyperparameters were identified everywhere in the pipeline, whether it was in a fitting function f_i (e.g., regularisation constant), a transformation function h_i (e.g., detection thresholds) or in the case of backpropagation, in the optimiser function g (e.g., learning rate, momentum).

For an end-to-end pipeline, the fitting function F yielding the performance score s, was defined to intake the complete dataset (x, y) and a hyper-parameter vector ρ such that:

$$s = F(x, y, \rho) \tag{3-9}$$

The optimal hyperparameter vector ρ^* then solved the following optimisation problem:

$$\rho^* = \underset{\rho \in \mathcal{S}_\rho}{\operatorname{argmax}} F(x, y, \rho) \tag{3-10}$$

where $\mathcal{S}_\rho$ is the whole hyperparameter search space with each dimension bounded under the form of a finite search interval. $\mathcal{S}_\rho$ could be constituted of both discrete (e.g., the number of selected channels or the length of a filter) and continuous dimensions (e.g., a regularisation constant). F was thought to be a black box function with unknown underlying structures (e.g. linearity, concavity or derivatives (Frazier, 2018; Shahriari et al., 2016)). ρ^* could only be found by sampling $\mathcal{S}_\rho$ using F. Practically, the quality of tuning depended on how many solutions could be investigated in a fix amount of iterations. In individual stage fitting, this was dependent on the computation required for each fitting function f_i and to some extend for each transformation function h_i. For gradient descent the average computation time of each complete gradient descent cycle ultimately indicated how many times F could be realistically computed. Different strategies were available to sample $\mathcal{S}_\rho$ such as parameter grid search or random search (Mace et al., 2014) (Appendix B).

Bayesian optimisation (BO) has increasingly been employed to address those types of problems (Head et al., 2021), notably in BCI research. At the time of writing, BO was traditionally reserved for tuning the hyperparameters of the end stage of the pipeline (Ahmadi et al., 2019). Bashashati et al. (2016a, 2016b) have successively employed BO to tune elements in other stages of a BCI pipeline, such as the settings of feature extractor or of spatial filter. An opportunity was seen to unify the tuning process of the end-to-end structure using BO.

Moreover, the core principle of BO is to create a surrogate probabilistic model $\hat{F}$ of F over $\mathcal{S}_p$ to guide a sequential search of parameter vectors. In other words, the method functions by accumulating prior knowledge of already sampled parameter points. BO consequently added minimal overheads as the surrogate model $\hat{F}$ fitting was orders of magnitude faster than F. The search space was kept reasonably, $\dim(\mathcal{S}_\rho) \leq 20$ (Frazier, 2018) as larger spaces require more iterations for convergence (Appendix B).

In the decoder tuning implementation, a default parameter vector ρ_0 was first evaluated. ρ_o was constructed from experience and best practices found in the literature. Several random instances of ρ were then drawn to initiate the fitting of $\hat{F}$. In a 'ask' step, the optimiser, following the policy of a given acquisition function, suggested a new ρ. This new point was sampled by evaluating F and $\hat{F}$ was updated in a 'tell' step. Figure 3.4 further describes the flow control for the algorithm, and further details on the BO procedure and comparisons with the random search method are provided in Appendix B. Specific criteria were also associated with the BO process to evaluate the decoding performance, the impact of tuning, and the decoder parameterisation (Figure 3.4). The difference between to score obtained with the tuned parameters that obtained with the default parameters measured to the total improvement I delivered:

$$I = F(\rho^*, x, y) - F(\rho_0, x, y) \tag{3-11}$$

Sensitivity analysis was employed to determine the impact of each dimension S_ρ on the output of $\hat{F}$ (Herman & Usher, 2017).

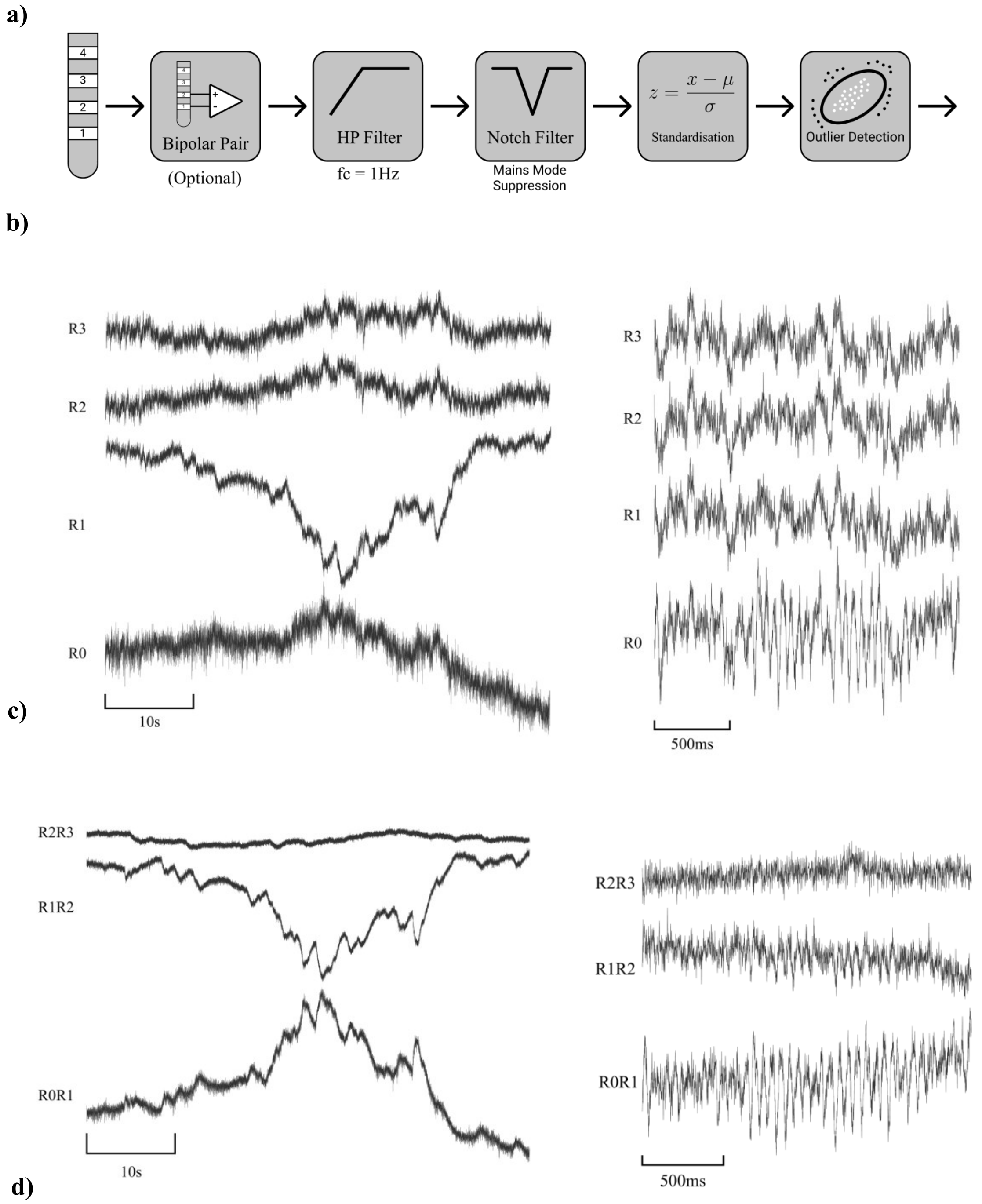
C. LFP Pre-processing

The acquisition and pre-processing of STN-LFP were handled in the first stage of the decoder. The pre-processor pipeline presented in Figure 3.5 was designed to remove the recording artefacts, i.e. dc-drift, mains harmonics, and movement artefacts while preserving the spectral structure of the signal. The pre-processor design was specific to experimental externalised recordings. The recording artefacts encountered in invasively embedded systems were likely to be different and mitigated principally through hardware design (Liu & Van der Spiegel, 2017). No tuning optimisation was applied to the pre-processor stage as its purpose was not related to the decoding performance. Pre-processing was conducted in Matlab (The Math Works, Inc. MATLAB. Version 2017a).

LFP could be recorded using the monopolar channels, as customarily done in EEG acquisition or bipolar differential channel pairs. The monopolar configuration measured the potential at each contact in reference to a mean potential, while the bipolar differential configuration measured the potential difference between two adjacent contacts (Figure 3.5). Bipolar timeseries could always be easily digitally reconstructed from the monopolar recording of two adjacent contact monopolar timeseries (Figure 3.5b). If x_1 and x_2 are two monopolar recordings, the bipolar timeseries x_{12} was given by the difference:

$$x_{12}[k] = x_1[k] - x_2[k] \tag{3-12}$$

at every sampling time index k.



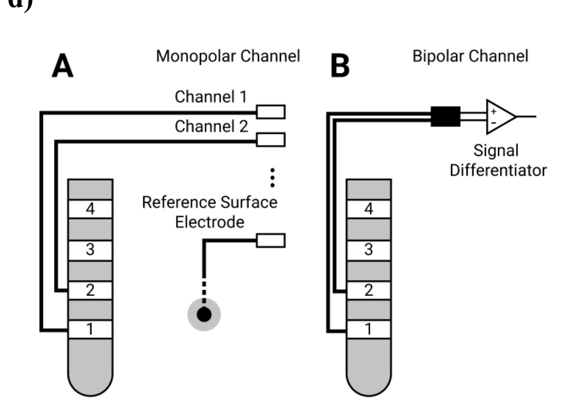


Figure 3.5 – Raw signals and pre-processing pipeline; **a)** Pre-processing and artefact mitigation stages; The first stage applies an optional digital bipolar reconstruction from two neighbouring contacts; High-pass filtering is applied to remove low frequency drift; Notch filters mitigate mains-artefact at 50, 100, 150, 200 and 250Hz; Signal is centred by mean μ and scaled by standard deviation σ; An outlier algorithm can be optionally applied to detect large artefact (order of magnitude larger than σ); **b)** Example of monopolar recordings and **c)** bipolar reconstruction from Sub01_Level_Right (Appendix A) prior to pre-processing; **d)** Diagram representing monopolar (A) and bipolar (B) contact configurations setup when a DBS lead is connected to a recording amplifier.

The recorded LFP were subject to very pronounced DC drift artefacts (Figure 3.5b), which needed to be removed utilising a digital filter high-pass (5^{th} order infinite response (IIR) Butterworth) with a cut-off frequency set to 1Hz. A 10^{th} order IIR multiple-notch filter with notch frequencies set at 50, 100, 150, 200 and 250Hz was employed to remove mains artefacts. The coefficients for the filter were found using the modified all-pass methodology proposed by Pel & Tseng (1997). The rejection bandwidth for each notch taper was set to 0.05π. The signals were then all individually normalised such that:

$$x_s[k] = \frac{x[k] - \mu(x)}{\sigma(x)} \tag{3-13}$$

where μ and σ are the mean and the standard deviation operators acting on the temporal dimension of x. The μ and σ were then respectively replaced with the median (MED) and mean-absolute-deviations (MAD) operators to increase the robustness of the fitting to outliers caused by pronounced motion artefacts:

$$\begin{gathered} x_s[k] = \frac{x[k] - \mathrm{MED}(x)}{\mathrm{MAD}(x)} \\ \mathrm{MAD} = \mathrm{MED}(|x - \mathrm{MED}(x)|) \end{gathered} \tag{3-14}$$

Outlier detection algorithms were also studied for the isolations of motion artefacts. An example result is presented in Appendix C.

D. Channel Space Management

The channel space is an intrinsic part of the feature extraction process. Despite its relevance to the decoding problem, few studies detailed a precise strategy to manage the different signal channels. Tan et al. (2016) has shown that ERS/ERD are not present in every channel and Shah et al. (2018) posited that the total inclusion may lead to decoder overfitting. Some studies relied upon manual selection, others based on ranking measured from one single feature (He et al., 2020; Khawaldeh et al., 2020; Loukas & Brown, 2004; Merk et al., 2021; Shah et al., 2018; Tan et al., 2016) and only few proposed automated algorithmic approaches (Golshan et al., 2020, 2017, 2016). Two principal paradigms were distinguished in the broader BCI research field. The first preferred to extract a maximum number of features from signals, and only a minimum number of selected channels were then be used for decoding (Flint et al., 2012; Mace et al., 2014; Shah et al., 2016, 2018). It stands to reason that this approach has somewhat been favoured for STN-LFP decoding, given the limited number of channels available on DBS probes. Conversely, the second paradigm preferred to extract a minimum number of features and incorporate a maximum number of channels in the decoding algorithm (Benabid et al., 2019; Golshan et al., 2018; Gruenwald et al., 2019). After that, a spatial filter (SF), feature selection or dimensionality reduction method may be employed to filter down the number of features.

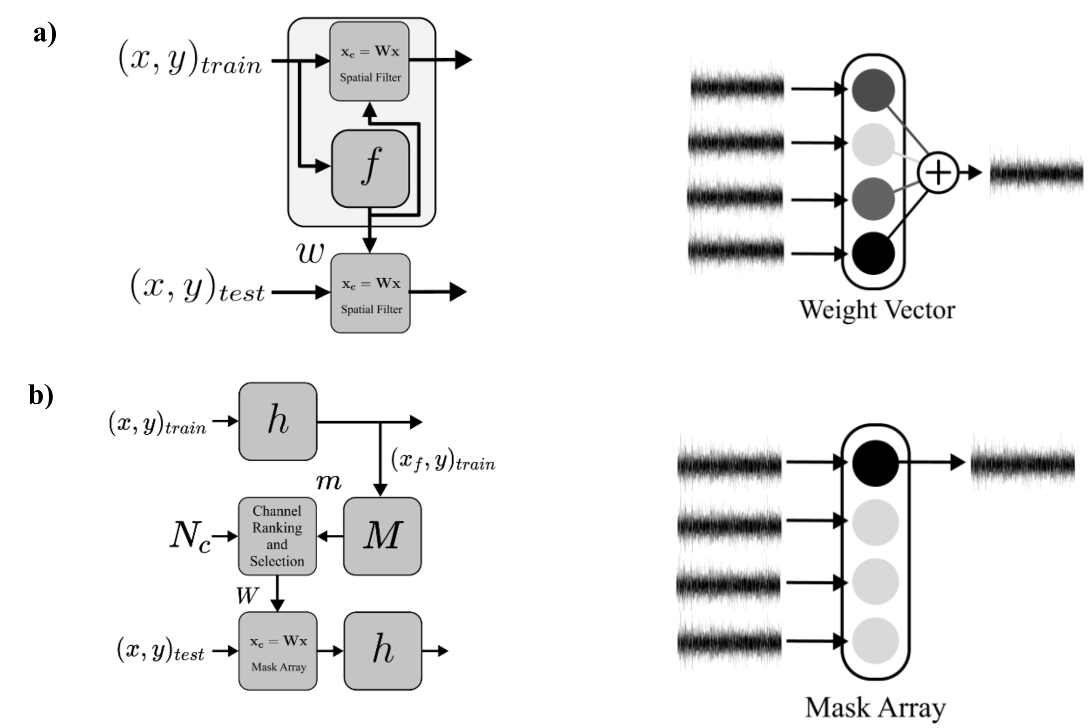


Figure 3.6– Channel management strategies; **a)** Spatial filters use a mixing matrix W to combine different channel signals into signal components x_c; There are as many components as there are rows in W; The mixing matrix is found during decoder fitting from training data x and labels y; **b)** Channel selection uses a selection vector W to remove undesired channels; In the proposed scheme channels are ranked according to a metric function M; N_c channels are then selected based on the output of M; Channel ranking is done after data transformation h into the feature space.

This approach was found to be more common when dealing with larger EEG or ECoG recording arrays (Kapeller et al., 2014). Hence, spatial filter (SF) and channel selection (CHS) technics were investigated to best use every available recording channel and test those paradigms.

Even with a limited number of channels, SFs could, in theory, produce stronger component signals by mixing several channels together (Figure 3.6a). Given a signal $x \in \mathbb{R}^{N_k \times N_{ch}}$ the component signal x_c was obtained at every timestep k as follows:

$$x_c[k, n] = \sum_{l=0}^{N_{ch}-1} w[n, l] x[k, l] \tag{3-15}$$

where l is the channel index, $n \in [\![0, N_c - 1]\!]$, the component index and w, the mixing weights. To achieve dimensionality reduction, the number of components N_c must be inferior or equal to N_{ch}. Common spatial patterns (CSP) algorithms (Chapter 4.C) and back-propagation (Chapter 4.E) were investigated for the fitting of w.

Moreover, channels needed to be selected based on the overall quality of the features they could provide. An automated approach was possible by introducing a metric function M. M measured the strength of the relationship between the features and the state to be decoded y (Chapter 4, Chapter 5). The workflow, described in Figure 3.6b, for CHS fitting differed from that of SFs. All features must be extracted from all the channels to complete the selection. The contribution of each channel measured was given by:

$$m = M(x_f, y) \tag{3-16}$$

where $x_f \in \mathbb{R}^{\frac{N_k}{R} \times N_{ch} \times N_f}$ is the complete extracted feature array, where N_f is the number of features and m is the channel metric vector such that $m \in \mathbb{R}^{N_{ch}}$. M evaluated the contribution of the features associated with each channel as a whole and translated it into a single scalar assigned to m. The indexes of the channels were then to be ranked according to m and incrementally added using the parameter N_c.

E. Feature Extraction

Feature extraction remained a recurring problem for STN-LFP decoding and was undoubtedly the most challenging stage to implement in the pipeline. Time-frequency (TF) features were favoured as they have been shown in past experimental and decoding work to be the most effective predictors of volitional motor state changes (Chapter 2.D). A series of different extractors stages were designed to test both CHS and SF paradigms, each centred around a different TF representation method. Each method has proven to function in real-time for electrophysiological signal analysis. A standard workflow was established for all the different feature extractors. The signals were decomposed into a set of subjacent TF components and segmented into epochs. From those epochs, the instantaneous power of each TF component was estimated. Some postprocessing was applied to help clean and standardise the final feature signals. The method presented were based on infinite impulse filters (IIR), the short time Fourier transform (STFT), the recursive wavelet transform (RWT), the discrete wavelet transform (DWT), parametric models (ARMA) and time-based Hjorth parameters (HJ). Detailed signal processing can be found in Appendix C. Example of extracted feature signals are given in Appendix D.

1. Sub-band Approach with IIR Filters

The first feature extraction method investigated aimed to extract the instantaneous power content of pre-defined sub-bands using digital filters. The time-variant modulation ERS and ERD were assumed to be constrained to a specific sub-band Ω_i (Figure 3.7a, the precise band definitions are given in Chapter 4.D and Chapter 5.C). Using the standard definition of a linear-time invariant filter, $x_{bd} \in \mathbb{R}^{N_k \times N_{ch} \times N_{bd}}$ was obtained from x:

$$x_{bd}[k, \dots, i] = \sum_{m=0}^{N_b - 1} b[m, i] x[k - m] - \sum_{n=1}^{N_a - 1} a[n, i] x_{bd}[k - n] \tag{3-17}$$

where N_{bd} is the total number of bands, $a \in \mathbb{R}^{N_a \times N_{bd}}$ (or $a \in \mathbb{C}^{N_a \times N_{bd}}$, with always $a[0] = 1$) and $b \in \mathbb{R}^{N_b \times N_{bd}}$ (or $b \in \mathbb{C}^{N_b \times N_{bd}}$) are respectively the recursive and forward coefficient banks for the filter, each of size of respective length N_a and N_b. Gruenwald et al. (2017, 2019) have proposed to use IIR filters for their computational efficiency and limited lag (with both recursive and forward coefficients). The Butterworth filter type was favoured due to its simple parameterisation and maximal flatness in the passband. Two limitations were noted. Firstly, the selection sharpness of the filters was limited, and some overlap was present between the bands. Other filter types, such as the elliptical or Chebyshev type-I, could have been employed to improve frequency selectability, but this came at the expense of adjusting more filter parameters and ripples in the passbands. Chebyshev type-II filters were investigated as a potential solution, but the filter passband was harder to define (Martineau et al., 2020). Secondly, the filter order was limited because of narrower bands such as α (8-12Hz). A stable filter solution only existed below the 5th order. The SF were applied following the IIR bank. The fitting of each SF and mixing of channels was repeated in each sub-band domain (Gruenwald et al., 2019). The sub-bands were then epoched for power extraction. Following the extractor pipeline proposed by Gruenwald et al. (2017), the epoch variance operator was implemented. The raw power array $x_p \in \mathbb{R}^{\frac{N_k}{R} \times N_c \times N_s}$ was given by:

$$x_p = \mathrm{Var_e}(x, R, n_o) \tag{3-18}$$

where $\mathrm{Var_e}$ is the epoch operator defined in Appendix C, R the downsampling ratio or epoch strides (3-1), and n_o the number of overlaps between epoch windows.

2. Narrow Filter Basis Approach with STFT

The STFT algorithm has been extensively used in previous decoder architectures (He et al., 2021; Shah et al., 2018; Stanslaski et al., 2018). For an LFP signal x, the complex STFT transform $\chi \in \mathbb{C}^{\frac{N_k}{R} \times N_c \times N_l}$ was given by Oppenheim & Schafer (2010):

$$\chi[kR, \dots, l] = \sum_{m=0}^{L(R,n_o)-1} x[kR - m]\, w[m] \exp\left(-j \frac{2\pi l m}{N_l}\right) \tag{3-19}$$

where w is a window function and N_l, the total number of bases such that $N_l = \frac{L(n_o)}{2} + 1$. Again, the size of L was made as a function of n_o, congruent to R and its exact definition is given in Appendix C. The windowed Fourier basis set $\mathcal{S}_F = \left\{w[m] \exp\left(-\frac{j2\pi l m}{N}\right)\right\}_{0 \le l < N_l}$ formed a bank of narrow-band filters spanning linearly Ω. Short extraction window have lower spectral resolution from which not all frequency band can be estimated.

Moreover, the filter bank interpretation of the STFT helped understand how L and w limited the frequency selectivity of each basis in $\mathcal{S}_F$ (Figure 3.7b) (Oppenheim & Schafer, 2010). The Kaiser function allowed to vary the window shape with the help of a single shape parameter β_K. The combination of the overlap number n_o and β_w enabled some control over both the temporal and spectral resolution of the transform. χ could then be partitioned into power bands. The power array x_p was obtained as follows:

$$x_p[k, \dots, i] = \frac{1}{|S_i|} \sum_{l \in s_i} \chi[k, \dots, l] \chi[k, \dots, l]^* \qquad (3\text{-}20)$$

$$S_i = \left\{ l \,\middle|\, \frac{l\,\pi}{N_l} \in \Omega_i \right\}$$

Moreover, the positioning of the power operator before the frequency selection in (3-20) made it impossible to place an SF like in the IIR extractor. Only a channel selector could be employed at the beginning of the extractor. The band averaging step was hence disregarded entirely and replaced by a single SF. For a modified TF array $\chi^{\frac{N_k}{R} \times N_{ch} N_l}$, the power component array x_p was given by:

$$x_p[k, n] = \sum_{l=0}^{N_{ch}N_l - 1} (w[n, l]\chi[k, \dots, l])(w[n, l]\chi[k, \dots, l])^* \qquad (3\text{-}21)$$

with $x_p \in \mathbb{R}^{\frac{N_k}{R} \times N_c}$, $n \in [\![0, N_l N_c - 1]\!]$ and where w is a complex SF mixing matrix as defined in (3-15). The SF was also employed as a pattern recogniser to compress χ across channels into a sparser sub-space of signal components (Golshan et al., 2016). The resulting extractor took full advantage of the STFT representation of the signal. Lastly, an optimised multi-taper approach was also investigated, where several STFT transformations were super-imposed for power estimation (Martineau et al., 2020) (Chapter 4.E).

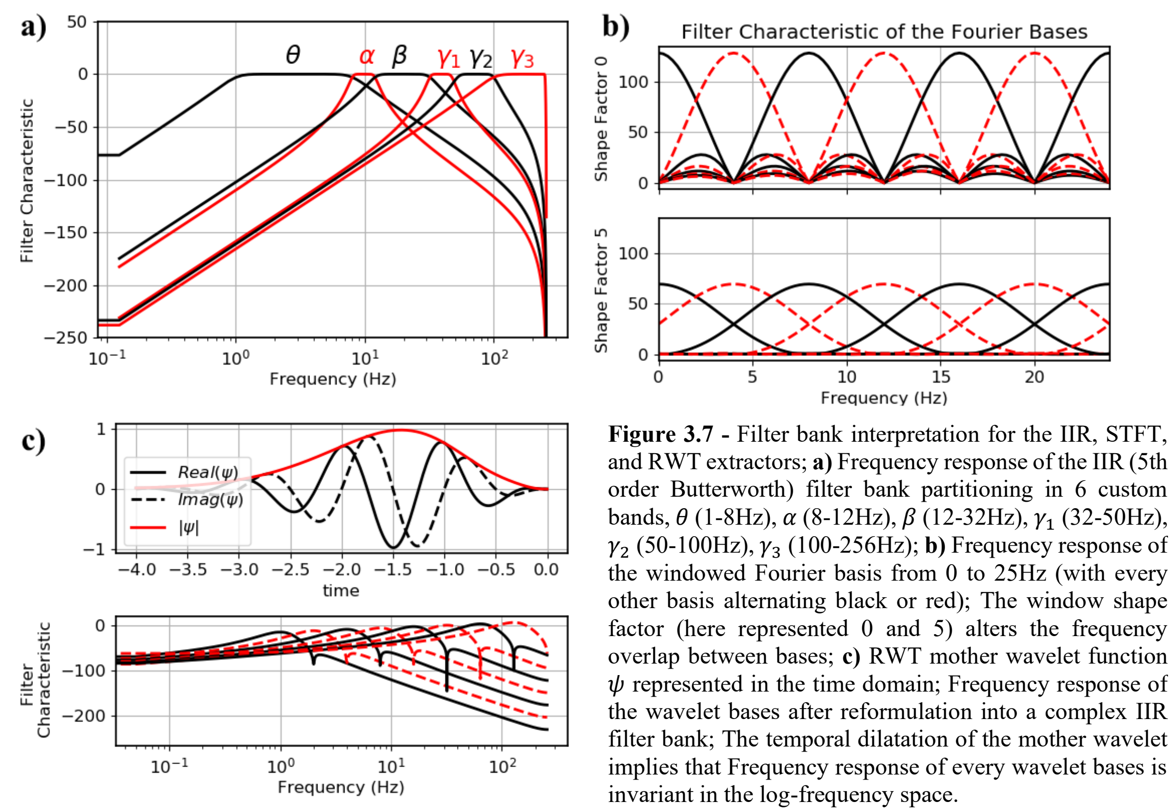


Figure 3.7 - Filter bank interpretation for the IIR, STFT, and RWT extractors; **a)** Frequency response of the IIR (5th order Butterworth) filter bank partitioning in 6 custom bands, θ (1-8Hz), α (8-12Hz), β (12-32Hz), γ_1 (32-50Hz), γ_2 (50-100Hz), γ_3 (100-256Hz); **b)** Frequency response of the windowed Fourier basis from 0 to 25Hz (with every other basis alternating black or red); The window shape factor (here represented 0 and 5) alters the frequency overlap between bases; **c)** RWT mother wavelet function ψ represented in the time domain; Frequency response of the wavelet bases after reformulation into a complex IIR filter bank; The temporal dilatation of the mother wavelet implies that Frequency response of every wavelet bases is invariant in the log-frequency space.

3. Continuous Wavelet Transform

Unlike the STFT, the continuous wavelet transform (CWT) is thought to better suit the decomposition of non-stationary signals and is a standard tool in bio-signal analysis (M. X. Cohen, 2018). In offline analysis, for a complex-valued CWT, the TF representation was constructed by convolving x with the wavelet kernels to obtain the wavelet coefficients $\chi \in \mathbb{C}^{\;N_k \times N_{ch} \times N_l}$, where N_l is again the total number of bases. Every kernel is derived from a mother wavelet $\psi(t)$. For a discrete signal, ψ needed to be sampled such that:

$$\chi[k, \dots, l] = \frac{\sqrt{f_l}}{f_s} \sum_{m=-\infty}^{+\infty} x[k]\psi\left(\frac{f_l}{f_s}(k-m)\right) \tag{3-22}$$

where f_l is the frequency at which ψ is scaled (Ren et al., 2011). The square of elements of χ returned the instantaneous TF power mapping (3-20), (3-21). Moreover, the shape of the frequency response of the sampled wavelet filters logarithmically stretched with frequency (Figure 3.7c). It was thus preferable to discretise the frequency dimension in octaves to maintain uniform overlap between the wavelet filters (Russell & Han, 2016):

$$f_l = \frac{f_s}{2^{\frac{l}{N_{wo}}+1}} \tag{3-23}$$

where N_{wo} is the number of wavelets per octave (e.g., for $f_s = 512$ and $N_{wo} = 1$ then $f_l = 256, 128, 64, 32 \dots$ for $l = 0,1,2,3 \dots$). Wavelets basis are centred on increasingly small frequencies (the minimal frequency usually being defined). Morlet's wavelet was chosen for its prevalence in LFP signal processing for offline analysis given its prevalence in the field:

$$\psi(t) = \exp(2\pi j t) \exp\left(-\left(\frac{t}{2\sigma}\right)^2\right) \tag{3-24}$$

where σ is the width of its gaussian envelope. σ served to adjust the temporal and spectral trade-off of the transform. Although the number of cycles of decay is most commonly employed to parameterise σ, the full-width at half-maximum (FWHM) of the wavelet was instead employed (M. X. Cohen, 2018). The FWHM was maintained superior to the period length of the wavelet and thereby constraining σ to be inversely proportional to f_l (M. X. Cohen, 2018).

4. Recursive Wavelet Transform Approach

Even though the usage of Morlet-CWT has been suggested for BCI application (Bashashati et al., 2016b; Benabid et al., 2019; Golshan et al., 2020; Tan et al., 2019; Wright et al., 2016), it was practically unsuited for real-time application. A relatively large number of filter kernels approximated by long finite response (FIR) filters would have been required to span the frequency space fully. In offline analysis, techniques such as the fast Fourier transform (FFT)-based convolution or graphical processor units (GPU) could have helped reduce the computation time. More problematically, due to the size of the filters, the signals must be delayed significantly. Ren et al. (2011) proposed a new class of mother wavelet functions approximated by compact IIR filter kernels. Here the gaussian enveloped was traded for a time based fifth-order skewed polynomial defined on $t \in]-\infty; 0]$ (Ren and Kezunovic, 2010):

$$\psi_{RWT}(t) = \left(-\frac{8}{9\sqrt{3}}(\pi t)^3 - \frac{8}{27}(\pi t)^4 - \frac{32}{135\sqrt{3}}(\pi t)^5\right) e^{2\pi\left(\frac{1}{\sqrt{3}}+j\right)t} \qquad (3\text{-}25)$$

Rent et al. (2010) proved that this wavelet functions could be reformulated recursively to match the format digital filters, as expressed in (3-15). A pair of complex coefficients (a, b) could therefore be obtained for each wavelet filter and implemented in an IIR filter bank yielding a TF array $\chi \in \mathbb{C}^{N_k \times N_{ch} \times N_l}$. The resulting RWT extractor resembled a cross-over between the IIR and STFT extractors. The power x_p was extracted either using band averaging (3-22) or with a SF (3-23) (Figure 3.7). Epoch averaging was employed to smooth further the signals (Appendix C).

5. Discrete Wavelet Transform Approach

Setting aside the RWT as a particular case, the more conventional implementation of wavelet-based feature extraction in real-time BCI remained the discrete wavelet transform (DWT) (Luo et al., 2018; Mace et al., 2014; Mamun, Huda, et al., 2012; Robinson et al., 2013). The fundamental unit of the transform was characterised using a quadrature mirror filter (QMF), which defined the mother wavelet (Vaidyanathan, 1993). The QMF filter constituted a pair of half-bands, low-pass and mirroring high-pass, filters that split every incoming $x \in \mathbb{R}^{N_k \times N_{ch}}$ signal into a low and mirrored high component x_g, $x_h \in \mathbb{R}^{\frac{N_k}{2} \times N_{ch}}$:

$$[x_g, x_h] = QMF(x, \Theta) \qquad (3\text{-}26)$$

where Θ is a parameter vector defining the QMF filter (Appendix C). Every other point of x_g and x_h were decimate. Matching the correct wavelet to the desired application was a problem raised in past LFP studies (Luo et al., 2018; Mamun et al., 2015). Therefore, the QMF-lattice parameterisation paradigm was adopted as it had already been shown to increase performance in other BCI systems (Farina et al., 2007). The computation graph and parameterisation of the QMF-lattice are presented in further details in Appendix C. Moreover, the DWT was obtained by iteratively applying the QMF to x_g resulting in a multi-scale decomposition of the signal and resembling the octave progression described in equation (3.25) (Figure 3.8). The wavelet coefficients were stored in a set $\mathcal{S}_C = \{x_l\}_{0 \leq l < N_l}$ where the number of levels, N_l, is given by:

$$N_l = \log_2(R) \qquad (3\text{-}27)$$

where R is again the downsampling ratio. The temporal size of each x_l was a division of the original time-length N_k by a 2^{l+1} factor. To reconstruct a single power array x_p from $\mathcal{S}_C$, each x_l were matched with a corresponding epoch set with the correct down-sampling rate, $R_l = \frac{R}{2^{l+1}}$. The usage of n_o also allowed to define every epoch length according to every R_l. As in the IIR extractor, a spatial filter could be defined and fitted on every level (Robinson et al., 2013). The power features were estimated from the time domain using the epoch sum of squares $\mathrm{SS_e}$ operator (Appendix C). The $\mathrm{SS_e}$ the operator was prefered over $\mathrm{Var_e}$ as it was defined in the case of unitary epoch length. Since the application did not demand a high output refreshing rate, there was no need to use a synthesis QMF bank and re-assemble specific signal sub-bands to the original input frequency (Robinson et al., 2013).

Furthermore, the DWT corresponded to one arrangement of the QMF filters. A complete tree decomposition (referred to as DWT-full[4]) was also obtained by filtering both x_g and x_h iteratively (Figure 3.8). Since the resulting coefficient have matching sampling frequencies, they were assembled directly within a single array $x_c \in \mathbb{R}^{N_k \times N_{ch}}$. Instead of spanning the frequency domain logarithmically, x_c partitions Ω in sub-bands of equal size $\frac{\pi}{R}$. A single SF was again employed as a pattern-recogniser to extract features throughout the entire TF representation of the signal.

Lastly, an adaptive decomposition of the wavelet (referred to as DWT-packet) tree was also investigated. Sparser packet transforms have been employed in other BCI architectures (Brechet et al., 2007). The DWT-packet relied on a metric function M to prune un-informative tree branches. The precise decomposition algorithm is also presented in Appendix C.

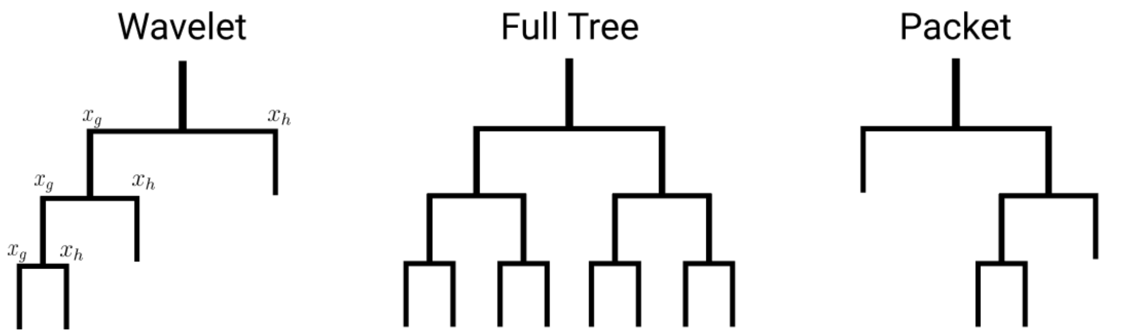


Figure 3.8 - Example of each DWT tree decomposition strategy; **(Wavelet)** each coefficient x_g is repeatedly decomposed on every level; Every x_h are collected from the tree leaves into the coefficient ensemble S_c; **(Full Tree)** both component x_g and x_h are decomposed at each tree node; **(Packet)** x_g and x_h are selected at every node to prune lesser informative tree branch; A metric function M is used for pruning; The decomposition can individually carried across channels.

6. Parametric Extraction Approach

All the methods implemented have involved real or complex-valued filter banks either spanning linearly, logarithmically, or adaptively the frequency space. Parametric-based power estimation has also been successfully applied to STN-LFP signals (Foffani et al., 2004, 2005). In comparison, the parametric approach functions through a model-based signal approximation, and the parameters of said model, in turn, serve to estimate the power spectrum. The approach proposed by Tarvainen et al. (2004),

[4] Even though, this method is usually known to as the DWT-packet decomposition, it was here referred to as the DWT-full decomposition. The term DWT-packet was instead reserved for the adaptive arrangement of the tree where branches are pre-emptively pruned.

consisting of an auto-regressive (AR) and moving average (MA) model, was adopted. x was here approximated from its past values (the AR terms), its fit residual e and past value of this residuals (the AM terms):

$$x[k] = e[k] + \sum_{m=1}^{N_{MA}} b[k,m]e[k-m] - \sum_{n=1}^{N_{AR}} a[k,n]x[k-n] \tag{3-28}$$

where a and b are respectively the AR and MA coefficient (with $a[0] = 1$ and $b[0] = 1$), and where N_{AR} and N_{MA} are respectively the number of AR and MA coefficients. The residual $e[k]$ was considered to be random and drawn from a normal distribution with variance σ_e (Khan & Dutt, 2007; Tarvainen et al., 2004). Equation (3-28) could be further vectorised to provide a compact observation equation:

$$x[k] = e[k] + \varphi_k^T \theta[k] \tag{3-29}$$

where the signal states were stored (by means of buffers) at every time step k in the vector $\varphi_k = \left[x[k-1], \dots, x[k-n], e[k-m], \dots., e[k-m]\right]^T$ and the model parameters in the array $\theta[k] = \left[-a[k,1], -a[k,2], \dots - a[k, N_{AR}], b[k,1], b[k,2] \dots, b[k, N_{MA}]\right]^T$. To model the dynamical behaviour of θ, the recursive least squares (RLS) adaptive filter method was employed (Khan & Dutt, 2007; Tarvainen et al., 2004). The RLS algorithm required only a single decay parameter η such that $\eta \in]0,1[$. The state space model under the RLS decay of θ (Khan & Dutt, 2007):

$$\theta[k] = \eta^{-1/2}\theta[k-1] \tag{3-30}$$

The generalised filter update is given in Appendix (C). Once the θ series was extracted from x, it was further epoched and averaged to gain a more reliable estimate of the parameters $\bar{\theta}$ at the timestep kR and the standard residual error σ_e^2 was as well computed using the $\mathrm{Var_e}$ operator. The power operator P was then given by:

$$P(\bar{\theta}, \sigma_e) = \frac{\sigma_e^2}{f_s} \frac{\left|1 + \sum_{n=N_{AR}}^{N_{AR}+N_{MA}-1} \bar{\theta}[n] \exp\left(-\frac{j2\pi lf}{f_s}\right)\right|^2}{\left|1 - \sum_{n=0}^{N_{AR}-1} \bar{\theta}[n] \exp\left(-\frac{j2\pi lf}{f_s}\right)\right|^2} \tag{3-31}$$

where the f is the continuous frequency defined within the operational bandwidth of the BCI such that $f \in [0, \frac{f_s}{2}[$ (Tarvainen et al., 2004). The power array x_p was then found via band averaging (3-22). Alternatively, $\bar{\theta}$ was also used directly as the feature array (Obermaier et al., 2001; Pfurtscheller & Neuper, 2001). Lastly, as in the STFT and RWT cases, the SF could not be placed in an intermediary stage before the power operator. Channel selection technics were thus preferred when using the ARMA extractor.

7. Temporal Domain Approach

Finally, time-domain features were extracted using the Hjorth parameters (HJ):

$$
\begin{aligned}
H_1(x) &= \mathrm{Var_e}(\mathrm{x}) \\
H_2(x) &= \sqrt{\frac{H_1(x[k]-x[k-1])}{H_1(x)}} \\
H_3(x) &= \frac{H_2(x[k]-x[k-1])}{H_2(x)}
\end{aligned}
\qquad (3\text{-}32)
$$

where H_1, H_2, H_3 are respectively the activity, mobility and complexity of the signal. Those were concatenated in a single feature array $x_p \in \mathbb{R}^{N_k \times N_c \times 3}$ with N_c selected channels. x_p measured indirectly through finite time differentials, key characteristics of the spectrum of x. Compared to other extractors, Hjorth parameters are inexpensive to compute and thus have served as complementary features (Shah et al., 2018; Zhu et al., 2020). The method chosen here was rather to keep investigating the usage of multiple channels with a single set of features as previously done in Martineau (2017). Previous results showed that the Hjorth parameters were sufficient for complete decoding, making them an attractive solution for embedded systems.

F. Feature Postprocessing and Space Assembly

Regardless of the type of power estimator or feature extractor employed, it was conjectured that raw power signals tended to adopt a strictly positive and naturally left-skewed distribution. A point-wise logarithmic transformation was applied to re-adjust the distribution (Gruenwald et al., 2017):

$$x_t[k] = \log\left(x_p[k]\right) \qquad (3\text{-}33)$$

where x_t is the transformed power array. The log-transform can be described as a particular case of the Box-Cox transform:

$$x_t[k] = \begin{cases} \log\left(x_p[k]\right) \text{ if } \lambda = 0 \\ \dfrac{x_p[k]^{\lambda} \ -1}{\lambda} \ \text{ if } \lambda > 0 \end{cases} \qquad (3\text{-}34)$$

where $\lambda \in [0,2]$ is the Box-Cox parameter. Studies on the application of Box-Cox transform to EEG data have suggested that the log transform should be sufficient (Smulders et al., 2018). Consequently, unless specified otherwise, the log operator was always employed. The feature signals were then again normalised using either (3-14) or (3-4) to centre the feature distributions as it was shown to help discern ERS/ERD from baseline activity (Wang et al., 2008).

Another common practice found in post-extraction processing is to use state-space modelling of the feature temporal dynamics to construct a Kalman-like filter (Wang et al., 2014). The single feature-size state-space proposed by Gruenwald et al. (2017) was first utilised to avoid having to fit an oversized state space. For a signal x and the hidden state x_s, the state and observation equations for the model filter were given by:

$$\begin{gathered} x_s[k] = x_s[k-1] + \; w[n] \\ x[k] = x_s[k] + u[n] \end{gathered} \tag{3-35}$$

where w and u are respectively the process and observation noise of the model drawn from two normal distributions such that $w \sim N(0, \sigma_w)$ and $u \sim N(0, \sigma_u)$. The model assumed that the hidden feature x_s followed a non-stationary random walk and a noise-corrupted observation. Trial averaging was employed to estimate those parameters. If $x_s \approx \bar{x}$ as defined in (3-2), then the process and noise variance was estimated from the following residuals:

$$\begin{gathered} \sigma_w^2 = \mathrm{Var}(\bar{x}[k] - \bar{x}[k-1]) \\ \sigma_u^2 = \mathrm{Var}(x - \bar{x}) \end{gathered} \tag{3-36}$$

This method was proven to be simultaneously efficient and robust. This Kalman filter was employed for discrete state decoders in Chapter 4.

A second order Kalman filter employed by Yao et al. (2020) was later introduced to provide even more smoothing for the continuous decoders in Chapter 5. This Kalman filter functioned according to the following state equations:

$$\begin{gathered} x_s[k] = x_s[k-1] + \frac{1}{f_d} v[k-1] + \; w_1[n] \\ v[k] = v[k-1] + \; w_2[n] \end{gathered} \tag{3-37}$$

where v is a latent velocity-like state. Instead of estimating σ_w and σ_u those were set using the ratio $r_\sigma = \sigma_w / \sigma_u$ during tuning (Yao et al., 2020).

The simplified update equations for both Kalman filters are presented in Appendix C.

Moreover, the feature space array $X \in \mathbb{R}^{\frac{N_k}{R} \times N_c N_s}$ was then assembled by merging any remaining parallel selected channel or component dimensions of size N_c with any new dimensions created in the feature extractor pipeline of size N_s ($N_s = N_{bd}$ where the bands were employed, $N_s = \log_2 R$, for the DWT-wavelet extractor, $N_s = 1$ for DWT-full, STFT and RWT SF based methods and DWT-packet extractors, $N_s = N_{AR} + N_{MA}$ for the ARMA coefficient extractor, and $N_s = 3$ for the HJ extractor). Current and past features were concatenated into a second epoch window to create a time-dependent estimator (Flint et al., 2012; Shah et al., 2018). The epoched feature space array X_e was given by:

$$X_e[k] = [X[k], X[k-1] \ldots, X[k - N_w + 1]] \tag{3-38}$$

where N_w is the length of the epoch window. If $N_w = 0$, then no feature epoch was required and $X_e = X$. Knowing n_o and N_w, the overall analysis window of the decoder τ_d was then given by:

$$\tau_d = \frac{1}{f_d}(N_w + n_o + 1) \tag{3-39}$$

N_w was defined as an additional hyperparameter to set during decoder tuning. The total number of features N_f finally amounted to:

$$N_f = N_c N_s (N_w + 1) \tag{3-40}$$

N_f was therefore equivalent to a cubic volume expanding depending on the number of channels, features and the size of the final epoch window.

G. Estimator Parameterisation and Fitting

Classifiers were used to estimate discrete states in Chapter 4.C, and regressors for continuous states in Chapter 5.C. Both types of estimators shared common models and algorithmic conceptions to map back feature signals to a given state (Table 3.2). The cost of deploying individual estimators was dependent on how this mapping is formulated (i.e., computational insensitivity of the realisation process from $X[k]$ to $\tilde{y}[k]$) and on how it is parameterised (i.e., storage of estimator parameters w).

For a single dimension discrete state $y[k] \in \{0,1\}$ or a continuous state $y[k] \in \mathbb{R}$ with $k = 0, 1, \dots \frac{N_k}{R} - 1$, the simplest state estimators studied assumed that a linear relation existed between y and the feature map $X \in \mathbb{R}^{\frac{N_k}{R} \times N_f}$, such that:

$$d[k] = b + \sum_{l=0}^{N_f - 1} w[l] X[k, l] \tag{3-41}$$

where $w \in \mathbb{R}^{N_f}$ is the weight vector, $b \in \mathbb{R}$ a bias constant and $d \in \mathbb{R}^{\frac{N_k}{R}}$ the model output. Classifiers, such as Linear Discriminant Analysis (LDA) or Logistic regression (LR), or regressors, such as the Ridge Regression (RR) or Elastic Net Regression (ENR), follow different methods and assumptions to fit w and b from training data. Each estimator possesses different fitting strategies. For instance, LDA uses a statistical model to define its parameterisation. Fitting itself required tuning hyperparameters such as regularisation constants.

In a regressor, d is directly the estimated state $\tilde{y}$ whilst in a classifier, d is commonly referred to as a decision function. Decision functions are associated with a linear hyper-plane separating classes in the feature space. A sigmoid activation function was used to normalise the decision output into the state probability $p[k] \in \{0,1\}$, such that:

$$p[k] = \frac{1}{1 + e^{-d[k]}} \tag{3-42}$$

Linear models had the advantage of being simple, interpretable and invertible. Multiple linear models could also be stacked together if multiple states needed be decoded in parallel (Chapter 4.E). The softmax function was then used to find class probabilities. Backpropagation eased fitting models composed of such linear transformation and activation functions.

Moreover, the Gaussian Naïve Bayes (NB) model studied. It uses a quadratic model also parameterised through statistical interpretation of a training dataset:

$$d[k] = b + a \sum_{l=0}^{N_f-1} (X[k,l] - w[l])^2 \tag{3-43}$$

where $a \in \mathbb{R}$ is an additional scaling factor. p was again found using a sigmoid activation function (Zhang, 2004).

Support vector machines differed from the previously presented models as they use feature vector instances drawn from the training dataset set to make predictions. For a feature vector $X[k]$, the decision function or regression output d is given by:

$$d[k] = b + \sum_{l=0}^{N_{sv}-1} w[l] K_{SVM}(X\,[k], X_{SV}[l]) \tag{3-44}$$

where $X_{SV} \in \mathbb{R}^{N_{SV} \times N_f}$ is the collection of feature vectors, or support vectors, N_{SV} the total number of support vectors employed for estimation, $w \in \mathbb{R}^{N_{SV}}$ a weight assigned to every support vector and K_{SVM} the kernel function (Platt, 1999). For classification, the probability signal p could be estimated by fitting an additional linear model and sigmoid normalisation on top of the decision function d (Platt, 1999). Which and the number of support vectors selected occurred during estimator fitting and depended on a tuning parameter (the regularisation constant C, Chapter 4.C). Kernel transformations make SVMs versatile and adaptable to non-linear problems. Radial-Basis Function (RBF) kernel SVM classifiers were studied for discrete state estimation (Chapter 4.C). Both linear and RBF SVMs regressors were studied for continuous state estimation Chapter 5.C).

Whereas SVM use only some training feature vector instances, nearest neighbour algorithms used all training examples made available during training. The nearest neighbours algorithm tabulates the feature space which acts as a reference to issue new predictions. K-Nearest-Neighbour Regression (KNR) was studied for continuous state estimation (Chapter 5.C). A feature vector $X[k]$ was looked-up against the training data X_{train}. The k-nearest algorithm relied on a distance function $\mathcal{D}(X[k], X_{train})$ to locate $X[k]$ within the training table. n_k neighbours then determined the prediction output $\tilde{y}[k]$, such that:

$$\tilde{y}[k] = \frac{1}{n_k} \sum_{l \in S_k} y_{train}[l] \tag{3-45}$$

where S_k is the set of neighbour indexes mapping back to the reference state table y_{train}. Finding the neighbours and storing the reference table was thus directly dependent on the dimensions of X_{train}.

Ensemble estimators coalesce multiple smaller estimators (a total of n_e) to form more robust state predictions. Methods exist to orchestrate fitting, such as gradient boosting (Zhu et al., 2020). Gradient-

Boosted Tree Regression (GBTR) was studied for continuous state estimation (Chapter 5.C). GBTR used decision-tree regressors as base unit estimators, decomposing the feature space sequentially. To estimate $y[k]$, the tree root started with one feature l from the vector $X[k,l]$ and threshold the feature to guide the decomposition down its left or right child branch, very much like in the DWT decomposition. This process was repeated at every tree node until a tree leaf was reached. Each leaf was matched with a corresponding state estimate $\tilde{y}[k]$. Given some constraints (e.g., the maximum tree depth), the fitting process determined the feature to the threshold at every node, the threshold value, and a corresponding estimated state. Although decision-trees could be on their own accord computationally efficient structures, ensemble estimators needed to rely on many of them.

Given that STN-LFP datasets were highly heterogeneous and the limited number of trials per recording, cross-validation of the decoder fitting was an absolute necessity. The risk of model overfitting was non-neglectable on such small, irregular and noisy datasets. Finding adequate subsets for performance testing representative of the remainder of the dataset was as well challenging. High inter-trial variance meant that test trials easily under or over-represented the true performance of the decoder. The k-fold approach (Shah et al., 2016) was employed to splits the dataset into several complimentary training and testing datasets or folds. The number of folds, N_{fold} (8 in this case) was chosen to be congruent with the number of CPU cores available to allow for faster parallel processing. The fold splitting was done alongside the trial dimension of data or by segmenting full-timeseries depending on the decoding task.

Cross-validation required certain stages or the entirety of the decoding pipeline to be fitted and evaluated for every fold. The final performance metric became the average $\bar{s}$ of each score s_i evaluated on the i^{th} fold:

$$\bar{s} = \frac{1}{N_{fold}} \sum_{i=0}^{N_{fold}-1} s_i \tag{3-46}$$

Given the small size and heterogeneity of the dataset again, it was preferred to trade lower variance over higher bias with a relatively large fold number. Moreover, the sequential and compartmentalised pipeline representation was relevant here. If none of the feature extractor sub-stages required cross-validated fitting, then the k-fold splitting occurred between the feature extractor and the estimator stages, as depicted in (Figure 3.9). Conversely, if the feature extractor contained sensitive parameters and hyperparameters, such as those associated with the SF filters, the k-fold splitting took place at the intake of the pipeline (Robinson et al., 2013).

Lastly, the following ratio measured overfitting r_o between cross-validation training and testing scores:

$$r_o = \frac{\bar{s}_{train} - \bar{s}_{test}}{\bar{s}_{train}} \tag{3-47}$$

Overfitting was useful to check if the trained estimator build a good representation of the data. If overfitting is very large, it is likely that the model is too complex for the dataset. Prediction are then made by shaping noise rather than by meaningful underlying relationships.

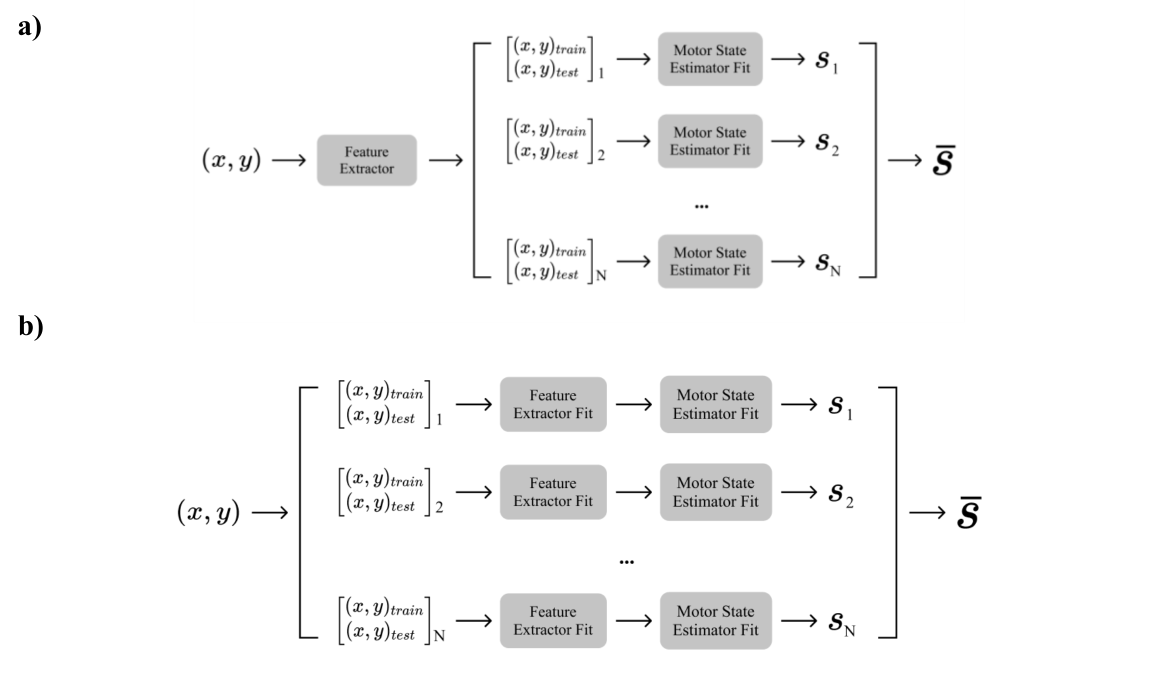


Figure 3.9 - Cross-validation flow; Pipeline inputs a signal, label pair (x, y) and outputs $\bar{s}$ the fold score average **a)** Single stage fitting; Only the estimator stage contains fitting parameters which requires cross-validation; **b)** Multiple stages fitting; Both feature extractor and estimator stages contain parameters which require cross-validation (in particular SF present in the feature extractor).

H. Assessment of Computational Cost for Decoder Update

Table 3.2 and Table 3.3 summarise different computational cost and memory allocation considerations in a full update cycle of the decoder. The proposed computational cost estimates were done at a high abstraction level, only taking into account the most direct formulation of every algorithm. No specific implementation optimisation (whether associated with the hardware or software) strategies were initially considered, as the aim was only to understand how the deployment costs would naturally scale given simple design parameters.

Table 3.2 – Summary of the different types of state estimators (including classifier and regressors) studies in sequentially fitted and BO tuned decoders, including algorithm type, description and computational considerations; Computational cost of the estimator Q_e was approximated to scale as a function of the total number of features N_f and other specific parameters (number of support vectors N_{sv}, number of estimators n_e, number of neighbours n_k, or number of temporal points used for training N_k); Memory allocation is estimated in terms of parameters stored and required to map the feature vector X to a decoded output $\tilde{y}$.

Properties	Linear Models	Quadratic Models	Support Vector Machine	Nearest Neighbours	Gradient-Boosted Tree
Classifier studied.	LDA, LR	NB	RBF-SVM		
Regressor studied.	RR, ENR		SVR, RBF-SVR	KNR	GBTR
Linearity of output/Decision function.	Linear	Quadratic	Non-Linear	Non-Linear	Non-Linear
Maximum memory requirement scaling.	$N_f + 1$	$N_f + 1$	$N_{SV}(N_f + 1) + 1$	$(N_f + 1)\frac{N_k}{R}$	n_e, N_f
Variables scaling computational demand for a single realisation (Q_e).	N_y,N_f	N_y, N_f	N_y, N_f, N_{SV}	$N_f, \frac{N_k}{R}, n_k$	N_y, N_f, n_e
Additional computational considerations.			Kernel computation K_{SVM}	Distance function $\mathcal{D}$.	Implementation of the ensemble; Type of unit estimator.

Table 3.3 - Summaries of the different characteristics for each type of feature extractor; Including method description; Support for complex-valued time-frequency representation for phase estimation; Tuneable parameters used in Bayesian Optimisation; Frequency space representation by filter bank; The summaries also include an estimation of computational cost for the extractor sub-elements (temporal filters Q_{TF}, spatial filter Q_{SF}, and other costs Q_o); Computation cost were estimated in terms of number of multiplications (M) and of addition (A) during a complete realisation; The costs were estimated as function of several design and hyperparameters (filter order N_a, extraction epoch length L, number of channels N_{ch}, channel or component selected N_c, number of bands N_{bd}); Depending on the extractor temporal and spatial filters need to be updated n_u (a function of the dowsampling ratio R) times to provide a decoded state at a rate f_d; The size of the memory arrays z_i were used for memory allocation estimation between computational updates; A three tap wavelet was considered for the DWT; $N_a = N_b$ for the IIR; Smoothing and power transform costs were not considered as they are not extractor specific and would depend on the total number of feature N_f.

Feature Extractor	**Description**	**Operation Frequency**	**Complex-Valued**	**Tuneable Parameters**	**Frequency Space**	**Computational Cost of Temporal Filters (Q_{TF})**	**Number of Temporal Filters (N_{TF})**	**Computational Cost of Spatial Filters (Q_{SF})**	**Number of Spatial Filter (N_{SF})**	**Number of Filter Update per Cycle (n_u)**	**Other Operation Cost (Q_o)**	**Filter Memory Allocation (size required in z_i)**
IIR-CHS	Direct Sub-band extraction	f_s	No	n_o	Custom Partition	$M = 2N_a + 1$ $A = 2N_a$	$N_{bd}N_c$			R	Cost of Variance Epoch: $M = N_{bd}N_c(L+3)$ $A = N_{bd}N_c(2L-1)$	IIR Filters: $N_{TF}(N_a - 1)$ Variance Epoch: $N_c N_{bd} R n_o$
IIR-SF							$N_{bd}N_{ch}$	$M = N_c N_{ch}$ $A = N_c(N_{ch} - 1)$	N_{bd}			
STFT-band-CHS	Complex FIR narrow band extraction	f_d	Yes	n_o, β_K	Linear Spanning	$M = 2L$ $A = 2(L-1)$	$\left(\frac{L}{2}+1\right)N_c$			1	Cost of Complex Norm and Band Average: $M = N_c(2\left(\frac{L}{2}+1\right) + N_{bd})$ $A = N_C\left(2\left(\frac{L}{2}+1\right) - N_{bd}\right)$	STFT Filters: $N_{TF} R n_o$
STFT-component-SF							$\left(\frac{L}{2}+1\right)N_{ch}$	$M = 4N_c\left(\frac{L}{2}+1\right)N_{ch}$ $A = N_c\left(5\left(\frac{L}{2}+1\right)N_{ch} - 1\right)$	1		Cost of Complex Norm: $M = 2N_c$ $A = N_c$	
RWT-band-CHS	Complex IIR narrow band extraction	f_s	Yes	n_o	Logarithmic Spanning	$M = 22$ $A = 20$	$N_l N_c$			R	Cost of Complex Norm and Band Average: $M = RN_c(2N_l + N_{bd})$ $A = RN_c(2N_l - N_{bd})$ Cost of Epoch Average: $M = N_c N_{bd}$ $A = N_c N_{bd}(L-1)$	IIR Filters: $10N_{TF}$ Epoch Average: $N_c N_{bd} R n_o$
RWT-component-SF							$N_l N_{ch}$	$M = 4N_c N_l N_{ch}$ $A = N_c(5N_l N_{ch} - 1)$	1		Cost of Complex Norm: $M = 2RN_c$ $A = RN_c$ Cost of Epoch Average: $M = N_c$ $A = N_c(L-1)$	IIR Filters: $10N_{TF}$ Epoch Average: $N_c R n_o$

Feature Extractor	Description	Operation Frequency	Complex-Valued	Tuneable Parameters	Frequency Space	Computational Cost of Temporal Filters (Q_{TF})	Number of Temporal Filters (N_{TF})	Computational Cost of Spatial Filters (Q_{SF})	Number of Spatial Filter (N_{SF})	Number of Filter Update per Cycle (n_u)	Other Operation Cost (Q_o)	Filter Memory Allocation
DWT-wavelet-CHS	QMF FIR sub-band extraction	f_d	No	$\Theta_0, \Theta_1, n_o, \tau$	Logarithmic Spanning	$M = 6$ $A = 6$	N_c			R-1	Cost of Sum of Squares of Epochs: $M = N_c L$ $A = N_c(L - \log_2 R + 1)$	QMF Filters: $2N_c \log_2 R$ Sum of Squares of Epochs: $N_c R n_o$
DWT-wavelet-SF							N_{ch}	$M = N_c N_{ch}$ $A = N_c(N_{ch} - 1)$	$\log_2 R + 1$			QMF Filters: $2N_{ch} \log_2 R$ Sum of Squares of Epochs: $N_c R n_o$
DWT-full-SF					Linear Spanning		N_{ch}	$M = N_c R N_{ch}$ $A = N_c(R N_{ch} - 1)$	1	$\frac{1}{2} R \log_2 R$	Cost of Sum of Squares of Epochs: $M = \frac{N_c L}{R}$ $A = N_c\left(\frac{L}{R} - 1\right)$	QMF Filters: $2N_{ch}(R-1)$ Sum of Squares of Epochs: $N_c n_o$
DWT-packet					Adaptive							
ARMA-band-CHS	Parametric modelling	f_s	Yes	η, N_{AR}, N_{MA}, n_o	Modelling of the Continuous Spectral Function	$N_\theta = N_{AR} + N_{MA}$: $M = 3N^2_\theta + 3N_\theta + 2$ $A = 2N_\theta{}^2 + 2N_\theta$	N_c			R	Cost of Epoch Average of ARMA Parameters and Epoch Variance of Error: $M = N_c(N_\theta + L + 3)$ $A = N_c(N_\theta(L-1) + 2L - 1)$ Cost of Power Operator: $M = N_c(2N_\theta + 2)$ $A = N_c(2N_\theta + 1)$ Cost of Band Average: $M = N_c N_{bd}$ $A = N_c N_{bd}(L-1)$	ARMA Filter: $N_c(N_\theta + 1)^2$ Epoch Average: $N_c N_{bd} R n_o$
ARMA-coeff-CHS											Cost of Epoch Average of ARMA parameter: $M = N_c(N_\theta)$ $A = N_c(N_\theta(L-1))$	ARMA Filter: $N_c(N_\theta + 1)^2$ Epoch Average: $N_c N_\theta R n_o$
HJ	Time derivative weighting	f_d	No	n_o	Linear weight	$M = 1$ $A = 3$	N_c			R	Cost of Epoch Variance: $M = 3N_c(L+3)$ $A = 3N_c(2L+3)$	Epoch Variance: $N_c(3Rn_o + 2)$

When deployed in real-time, it was considered that for each refreshing step k, the decoder received a block of raw data $x \in \mathbb{R}^{R \times N_c}$ from the acquisition system with R signal timepoints and N_c selected channels (or, if needed, all available N_{ch} channels) (Figure 3.10). Every proposed feature extractor required N_{TF} temporal filters and N_{SF} spatial filters to decompose x into a feature vector $X[k] \in \mathbb{R}^{N_f}$. Depending on the extractor, these filters needed to be updated n_u times to process the entire data block. Temporal filters required a set of memory arrays z_i to be stored, red and overwritten once every n_u refreshing cycle. All filters also required coefficients arrays w_i to be accessible for processing. Again, additional processes were also required to assemble $X[k]$, such as epoching, normalisation, and smoothing, which are only carried out once every timestep k, all also requiring their memory states and parameter arrays. Lastly, the feature vector $X[k]$ was passed on to the state estimator, which required access to its parameter arrays to determine the estimated state $\tilde{y}$.

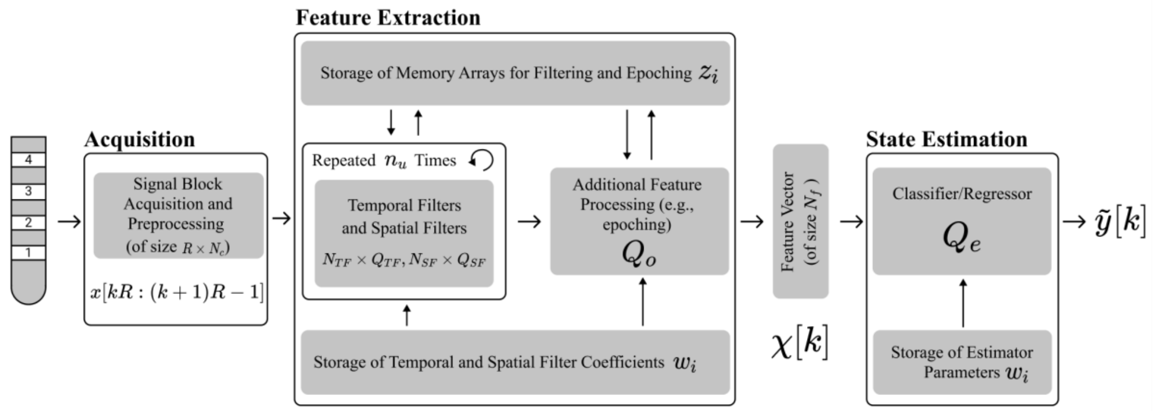


Figure 3.10 – Diagram presenting a summary of the computational process for a decoder update when operating online; The decoder receives a signal block x of length R from N_c activate channels at a rate f_d; In the feature extraction process temporal and spatial filters are updated n_u times to process the entirety of signal block across R time points; Memory states z_i are updated and a feature vector X is issued to the state estimator; The state estimator yields the decoded state $\tilde{y}$; Parameter groups w_i need to be accessed at each update step.

Overall, the total computational, Q_{tot}, cost was expressed as an estimate of the required floating point operations per second:

$$Q_{tot} = f_d(n_u(Q_{TF}N_{TF} + Q_{SF}N_{SF}) + Q_o + Q_{est}) \tag{3-48}$$

where, Q_{TF} is the cost of a time filter, Q_{SF} the cost of a spatial filter, Q_o other processes cost associated with feature extraction and Q_{est} the cost of a single state estimation from a feature vector $X[k]$ into an estimated state $\tilde{y}[k]$. Definitions for Q_{TF}, Q_{SF}, Q_o (in terms of the number of additions and multiplications), N_{TF} and N_{SF} can be found in Table 3.3.

Figure 3.11a presents estimates of computational costs and memory allocation requirements for every proposed extraction method given standard settings. Overall, initial computational cost varied widely from approximately a 1Gflops for the STFT-component-SF to 10kflops for the DWT-wavelet-CHS. The DWT-wavelet-CHS was, interestingly, less expensive than the Hjorth parameters. Recursive approaches were relatively demanding, with IIR-CHS and ARMA methods requiring 100kflops and the

RWT near a 1Mflop. The STFT-CHS was the most expensive requiring approximatively 10Mflops. Spatial filters represented a significant computational overhead, typically increasing cost by one to two orders of magnitude over equivalent extractors.

Figure 3.11b presents memory storage requirements for temporal filter memory. Except for the STFT, all extractors required approximately about or less than 20kbit of memory to operate.

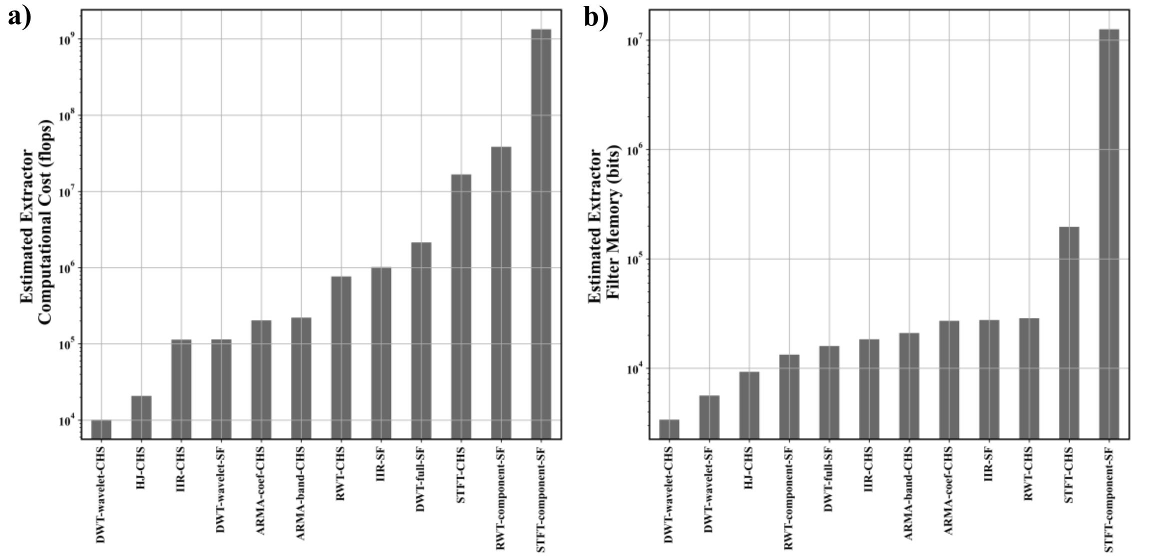


Figure 3.11 – Theoretical estimates of **a)** computational cost and **b)** memory allocation requirement for each feature extraction method; For this reference example, the decoder frequency (f_d) was set at to 16Hz; 8 signal channels (N_{ch}), 3 overlaps between extraction windows (n_o), and 6 frequency bands (N_{bd}) were used; IIR filters were of 5th order and DWT used 3 taps; It is assumed that operations and storage are using 32bit floating point numbers.

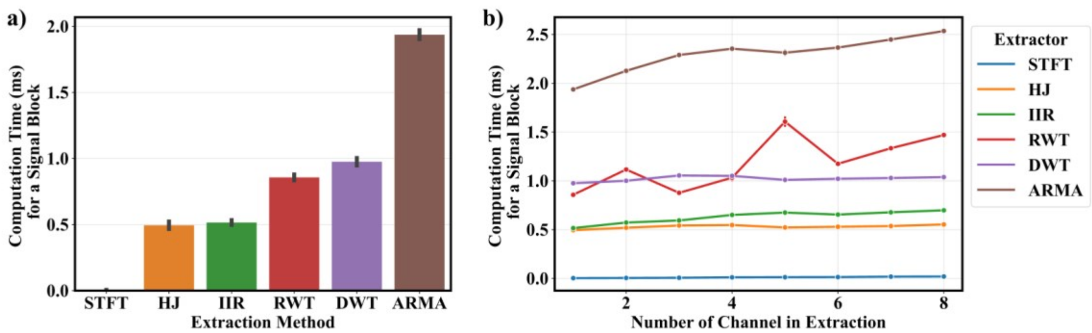


Figure 3.12 – © 2023 Frontiers, Reprinted, from Martineau et al . (2023); Computational time benchmark for CHS based feature extraction method (Intel(R) Core(TM) i7-7700K 4.20GHz); Each method was evaluated using a 100s long, 512Hz (f_s), gaussian random noise signal; **a)** The extraction was repeated 30 times using default setting parameters and a single channel signal; The average time taken to complete a single computational step (process a block of R = 32 signal points) at 16Hz (f_d) was estimated; **b)** The benchmark was reproduced by incrementally adding channels to a total of 8; Each method benchmark is significantly inferior to the refreshing time (62.5ms) between decoder output updates; All methods were custom implemented using the NumPy library, except the STFT which used a specialised function from the SciPy library; Error-bars indicate the standard-deviation taken across the 30 extractions.

Potential storage computational and storage cost for state estimators are comparable (at least in for the simpler linear models). For example if 8 channels (N_{ch}), 15 past feature vectors (N_w), or a second worth of data for a refreshing decoder frequency f_d set to 16Hz, and 6 frequency bands are used, 720 features would be produced for state estimation. Assuming 32bit floats are used, approximately 23kbit of memory and 23kflops would be required to store and compute a simple linear model.

The number of flops a hardware system can support scales with its clock frequency. A modern CPU core (such as the one used for this work) has a clock system of several GHz and Gbits of random-access-memory. Thus, computational resources were sufficient to service all extraction methods (assuming that the STFT is optimised with the FFT algorithm) and most likely all state estimations under real-time (or pseudo-real-time) constraints. To confirm theorical estimates, offline benchmarking was carried using a random signal processed by the CHS feature extractors (Figure 3.12) and by estimating the time taken by software to carry one update step. All method were updated under $1/f_d$s as specified. The STFT was the fastest because it relied upon well-developed and low-level FFT implementation from the SciPy library. All other methods were implemented in NumPy. A significant overhead occurred because of slow array manipulation. This overhead is quantifiable when adding decoding channels to the benchmark (Figure 3.12b). Computational time for efficient methods such as the DWT and the HJ did not scale with the number of channels whereas that of more complex and computationally intensive methods such as the ARMA and RWT experienced a steady increase with every added channels.

I. Discussion

The proposed pipeline architecture permitted to handle streaming and chunked data for real-time and offline operations. It was also designed for tuning and modularity. All the methods presented for pipeline optimisation and feature extraction have inherent advantages and general limitations. Alternatives and possible improvement have been noted, notably to improve the range and the quality for features extracted from the signals in future developments. There implementation on hardware is also discussed.

1. Alternatives and Improvements to the Tuning Methods

BO was not the only technic available for hyperparameter tuning. Other possible choice included, for instance, genetic algorithms, standard or adaptive (such as successive-halving) grid-search (Jamieson & Talwalkar, 2016; Mace et al., 2014). Even though BO was judged to be the most accessible method (Head et al., 2021), further comparisons between other methods would undoubtedly be insightful. It is worth noting that an unmatched advantage of BO remained the generation of the surrogate model.

Improvements to the method are possible and likely needed. First, convergence and stop criteria are needed to help reduce running time and better manage computational resources. Available criterions were judged too unreliable (Head et al., 2021), and more investigation was overall needed. In addition, an equal number of iterations were maintained to assure that adequate comparison could be made across all tuning outcomes and measured improvement. The acquisition function could also be upgraded. Technics such as no-regret (Grado et al., 2018; Srinivas et al., 2012) have been shown to provide the search algorithm with a better exploration/exploitation trade-off. Better exploration would imply a better sampling the of $\mathcal{S}_\rho$, localisation of potential local minima and a more representative surrogate model $\hat{F}$. The search algorithm should as well be re-started intermittently to help regularise the fitting of $\hat{F}$ and avoid local-minima entrapment (Head et al., 2021). Furthermore, a multi-objective variant of the algorithm such as the one proposed by Feliot et al. (2017) would undeniably increase the versatility of the tuning algorithm. Real-world applications are likely to require a particular trade-off between specific characteristics of the decoder, such as performance or complexity. As further explained and studied in Chapter 4, it can even be challenging to measure performance using a single objective function. Secondary objectives can help counterbalanced practical adverse optimisation effects, such as an accurate yet noisy or latent decoder output (Mace et al., 2014). A multi-objective function could avoid compounding several objectives into a unitary function.

The current limitation of the backpropagation approach is access to data, further discussed in Chapter 4. The recorded dataset was possibly too small, making the optimisation less efficient and the design of small decoder neural networks challenging (Martineau et al., 2020). An envisioned solution to this problem was concatenating the recording across participants to train population-wide decoders. Although employed in other past studies (Lawhern et al., 2018), this solution was judged yet experimental and its outcome too unpredictable (Golshan et al., 2020). Decoding outcomes were found very variable across sampled participants. The better individual performances could help bootstrap the population decoder to improve the individual worst performances. However, the converse could also be true, with the worst performances weighing more negatively on the overall outcome. Lastly, model compression for embedding should be addressed in future decoder optimisation work (Zhu et al., 2020).

2. Improvements to the Pre-processing Method

More could have been done to improve the offline pre-processing of the LFP signals. In particular, adaptive filtering has shown to be a good technic for sinusoidal-harmonic isolation (Riviere et al., 1997) in other applications. This technic could have made the artefact isolation more precise and enabled a lesser destructive cancellation. Digital spectral shaping could also have been employed to whiten the LFP signals (remove its 1/f characteristic) and increase frequency sensitivity (Gruenwald et al., 2017, 2019). This technic was attempted but did not deliver the expected performance improvement. The recorded γ frequencies were weak to start with, and the impact of the shaping was minimal (Merk et al., 2021). The outlier detection scheme for motion artefacts tagging could still be improved upon, but as discussed in Appendix C, the principal hurdle would remain the characterisation of the artefact. Although those measures would improve the quality of experimental recordings made from externalised DBS leads, many artefacts would not be recorded with specialised cutting-edge invasive hardware. DC battery power should resolve the problem of mains artefact, and removal of lead to amplifier wires mitigates the risk of significant motion artefacts. Analogue whitening and AC coupled recordings are possible (Liu & Van der Spiegel, 2017).

3. Comparison of, Improvements and Alternatives to the Feature Extraction Methods

Three distinctions helped better characterise each algorithm from one another (Table 3.3). The first was the operation frequency of the filter. First, the IIR, RWT and ARMA extractors operated at the input f_s frequency of the decoder. Further epoching followed by downsampling helped average further the feature signal. Those extractors had a recursive characteristic, whereas algorithms such as the STFT, HJ or DWT all operated at the output frequency f_d and were non-recursive. Shanechi et al. (2017) has shown that a decoder with a faster output frequency (~200Hz), closer to f_s, could help increase the user's task performance. However, when using LFP, Hsieh & Shanechi (2016) preferred to use slower extraction methods and employed a Kalman scheme to interpolate the decoded trajectory between available feature points. Fast extraction might only be of limited value here, and there might be more benefits in slowing-down the signal, until the representation of the state signal y becomes inadequate (Golshan et al., 2016). Secondly, a key difference between all those algorithms was how those partitioned the spectral space, either adaptively, linearly, logarithmically or by another mean. It was not obvious which would be the most effective. For instance, a case could be made that logarithm spanning should best represent an LFP signal given its 1/f characteristic (Liu & Van der Spiegel, 2017). Some representations could be helpful for their simplicity, such as with the STFT or the HJ. Others for the adaptivity (ARMA) or sparsity of the representation, such as the DWT-packet. Thirdly, the algorithm differed principally in their complexity and how it impacted computational cost.

Furthermore, improvements could also be carried on the individual feature extraction methods to address some of their shortcomings. Asymmetric or exponential windows could be investigated to mitigate the latency associated with the STFT (Rozman & Kodek, 2007). In particular, the exponential window would permit a recursive reformulation of the algorithm, making it computationally very efficient even compared to a standard FFT implementation (Grado et al., 2017). A full Kalman realisation of the ARMA extractor could increase the quality of the output features. The RLS recursion would be instead traded for a complete state-space modelling of the coefficient dynamics optimised using technics such as expectation maximisation without the need to set the η parameter (Khan & Dutt, 2007). There may be an incentive to investigate the parameterisation of the RWT further. ψ_{RWT} unlike Morlet's wavelet lacks any adjustable settings of the temporal or spectral trade-off. More taps could be as well added to the DWT wavelet at the expense of more latency. Another decomposition approach using QMF filters was theorised. It analysed at each level the rectified value of the incoming signal such that:

$$[x_g, x_h] = \mathrm{QMF}(x) \text{ and } [x_g^*, x_h^*] = \mathrm{QMF}(|x|) \tag{3-49}$$

Essentially, this approach would systematically perform envelope and carrier decomposition at every level and ensure maximal smoothness of the coefficients. The size of the coefficient set would, however explode as a full decomposition tree would be needed for every rectified and non-rectified node permutation. At present, there is no clear way to handle this set and eliminate redundant coefficients just yet.

Additional feature postprocessing technics could have been applied. Match-filtering may have helped to further isolate the AM signals from noise at a minimal computational cost (Liu & Van der Spiegel, 2017; Niketeghad et al., 2015). Kalman backwards smoothing could improve both the ARMA and the feature-wise Kalman filters (Gruenwald et al., 2017; Tarvainen et al., 2004). Simple exponential smoothing was considered, but difficulty arose in setting a forgetting constant. Over-smoothing could significantly distort the signal. This difficulty was also faced with the realisation of the ARMA extractor. Online optimal Kalman parameter estimation could also be implemented (Brittain & Halliday, 2011). One improvement that could benefit the conditioning of the features would be online corrections for feature baseline drifts. (2-1) did not capture any non-stationarity, which in practice was not always representative of the feature signal behaviours, which never tended to perfectly re-equalise after ERS/ERD.

The principal assumption on encoding information in LFP signals was that ERS/ERDs occurred on narrow and broadband carrier frequencies. Those amplitude modulated (AM) signals essentially formed a decodable feature space. There was some evidence that in addition to undergoing AM, LFP carrier signals also undergo frequency modulation (FM) within their bandwidth space (Foffani et al., 2005). Instead of the model proposed in (2-1), LFP signals could be better represented by a linear combination of modal signals of the form $x_i[k] = A_i[k]\sin[\phi_i[k]]$ where A_i and ϕ_i are the modulating amplitude and phase function, respectively.

Empirical mode decomposition has been considered for combined AM-FM signal representation of LFPs (Sun et al., 2018). The method, however, suffered from practical implementation difficulties. First, it is entirely unsuited for real-time feature extraction. The empirical mode algorithm needed to be recursively applied on segments of a signal of interest and could not function on a streaming signal. A new extractor would have to be designed to reproduce the output of the empirical mode in real-time. Secondly, the AM-FM decomposition would have been then purely heuristic. Although this meant that the full signal TF representation could be purely data-driven, a problem arose comparing decomposition across trials. Different trials would result in different decomposed modes, making constructing a feature space difficult. Recombination into bands would be possible but added to the algorithm complexity (Zaker et al., 2016).

Two other available applicable solutions to the AM-FM problem, capable of operating in real-time, were the iterated Hilbert transformation (Gianfelici et al., 2007) or the band-limited Fourier linear combiner (Y. Wang et al., 2013). The iterated Hilbert decomposition had the advantage of being fully theorised and of being carried on subsequent epoch windows for an optimal signal representation. The method, however, came at the cost of a very specialised implementation. Tests were also completed on a fast-IIR Hilbert transform to extract FM content using the IIR filter bank, but no significant FM modulation in the current frequency bands of interest was found. The band-limited Fourier linear combiner is an adaptive filter bank of Fourier basis spanning the frequency space Ω. Each basis phase and amplitude parameters could have been used to reconstruct the original signal using a streaming gradient descent technic, such as the least-mean-square or RLS. Finding the optimal learning rate for the algorithm could have been resolved using BO. FM content could have also been directly extracted from the ARMA coefficient model (Foffani et al., 2005).

A modification of the ARMA-RLS was implemented to study Granger causality (GC) to investigate features for laterality selection (Mamun et al., 2015). However, several problems were encountered, including low feature strength and additional constraints on channel selection. GC required the pairing of channels, which implied many permutations to be probed. Additionally reproducing different GC extraction across different frequency bands judged too computationally intensive (Mamun et al., 2015).

Other signal features could have as been studied, most notably, higher signal statistical moments (skewness, kurtosis), temporal features, entropy, relative band measurements, coherence (Camara, Isasi, et al., 2015; Shah et al., 2018; Zhu et al., 2020). A final argument against adding more types of features was the cubic nature of the feature space, scaling both the number of selected channels N_c and the length of the feature window N_w.

Lastly, random-correlation ensembles have proven to be effective substitutes to convolutional network features on small datasets (Dempster et al., 2020) and merit in depth investigation.

4. Real-World Deployment

Multiple design parameters scaled the cost of the feature extraction. First, the filter order was an important base unit of the cost. In methods based on the DWT, Hjorth parameters, IIR, and to some extent, the RWT, the filter order was relatively small, greatly reducing computation costs. For ARMA methods, adding AR or MA parameters could increase filter computation costs quadratically. Narrowband filtering approaches such as the STFT and the RWT were expected to be more expensive than other methods. The STFT computation was found to be most problematic as the filter order is directly proportional to the epoch length L. Long extraction windows were typically needed for sufficient frequency resolution and good temporal smoothing. Given that $\frac{L}{2}+1$ filters were then required, the computation cost was scaled quadratically with L. Optimising the computation STFT in particular using the FFT algorithm, is absolutely necessary for extractor deployment. The conducted benchmark indicated this solution was viable.

Epoching overlaps were also found to add important additional computation and memory costs to most processes. For instance, Hjorth parameters were more expensive to extract than DWT coefficients on long epoch windows. Even more, updating recursive filters also proved to be computationally intensive. For instance, for an incoming data block, an IIR or ARMA filter had to be computed R times over N_{bd} bands or $N_{AR}+N_{MA}$ parameters respectively. The DWT algorithm was incredibly efficient in both regards. The cost of taking an epoch and the required number of temporal (and spatial) filter updates was halved at every decomposition tree level, compounding in smaller overall costs for similar output features, even for full tree decomposition. Given these different decomposition approaches, the DWT is thus a good computationally efficient substitute for most other extraction algorithms.

Other computational overheads were identified. Generally, the number of channels or component N_c directly scaled most of the computational costs and the number of total features N_f extracted for decoding. The spatial filter introduced a noticeable overhead as they required that every N_{TF} temporal filters be computed across every N_{ch} channels. Depending on N_{ch}, adding a row for every N_c components could lead to an excessively large mixing matrix. A way around this problem could have been to combine channel selection and spatial filtering. Although the cost for post-processing features (normalising and smoothing) scaled with N_f they were considered to be relatively small in comparison to other extraction processes.

Furthermore, regardless of the estimator class chosen, computational cost also scaled with the number of features N_f employed for state estimation. For linear or quadratic models, the number of floating point operations required between model parameters and feature entries was directly proportional to N_f. For SVM estimators, the number of operations required for N_f features was dependent on the kernel function K_{SVM} which should typically require optimisation. In addition, the kernel operations have to be repeated for every N_{SV} support vector. Nearest-neighbour algorithm required the computation of distance functions $\mathcal{D}$ between the input vector X_f and tabulated training data. The larger the reference, both in terms of stored examples and N_f, more intensive this process becomes. Nonetheless, one undisputed advantage of the nearest method is that multiple states N_y can be mapped back to a vector X_f, whereas other methods require duplicated or parallelized representation for every single state. In ensemble methods, the cost of updating a single unit estimator, such as a decision tree, also depended on N_f and compounded across the total number n_e of estimators.

For memory storage, the same observation mostly holds. Linear and quadratic models required approximately N_f coefficient to be stored. SVM required N_{SV} support vector of size N_f which was a kind to storing multiple linear model. The same can be said of ensemble methods, which also required storing multiple models. Every model is, in sorts, a compressed representation of the training space. In this regard, nearest-neighbour algorithms are an edge case and storing the entire training representation and made them the most demanding in terms of memory allocation.

In sum, to minimise computational costs and memory, if the resulting performance is comparable, simple linear and quadratic models should be used over SVM, SVM over ensembles and only in specific circumstances should nearest algorithms be considered. More advanced algorithms would require multiples of these base reference specifications.

Consequently, complex representations, particularly neural networks or estimator-ensembles, require resource-aware optimisation for deployment, which was not originally considered but could be implemented in the future. For instance, the number of branches explored and leaves considered in each decision tree can slow the estimation time, especially when multiple tree outputs are combined. Zhu et al. (2020) proposed the usage of soft-oblique tree ensembles, which optimised the estimation routing through a large multi-level decision tree. A single estimator computation is equivalent to a succession of linear models and sigmoid activation (one per tree level). Zhu et al. (2020) also used a weight-sharing scheme between models to reduce model weight storage. In general, strategies that favour sparsity within models are desirable. These touch on weight regularisation, pruning, quantisation, and sharing to minimize basic linear transformation computation and storage cost (Zhu et al., 2020). Sparse mixing matrices would be particularly helpful for spatial filters, and selecting channels through sparse filter fitting would greatly reduce deployment costs (Lotte and Guan, 2011; Wang et al., 2020).

Hypothetical transition to real-world applications implies that hardware targets need to be identified to deploy the various algorithms discussed in this section. A BCI system is a precise algorithm pipeline for decoding and require a delicate stack of several hardware layers submitted to stringent specification and ranging from implanted, wearable, or remote devices. Power consumption, heat dissipation, physical hardware size, telemetric bottlenecks, memory, and computational caps are the most important constraints against which the hardware specifications need to be drafted (Avestruz et al., 2008; Liu & Van der Spiegel, 2017; Rouse et al., 2011; Stanslaski et al., 2018). Maintaining the flexibility and the performance targets of the software while abiding by the specification the hardware is a delicate balancing act.

Several possible levels of hardware integration were identified (Figure 3.13).

First was the front-end analogue conditioning circuitry. It was here assumed that as BCI hardware technologies matures, much of the conditioning circuitry and, like other lower hardware layers, will be hosted on an implanted chip at a minimal distance of the sensing electrodes. This transition has been so far supported by the introduction of low power-consumption electronics (Liu & Van der Spiegel, 2017).

The analogue conditioning should prepare the signal for analogue-to-digital conversion (ADC) by maximising its SNR and its dynamic range for a given maximum number of quantisation bit (more bits imply more buffer on-chip memory and more information to transmit out). Pre-processing steps included whitening, amplification, and band-limiting filtering should be implemented analogously (Liu & Van der Spiegel, 2017) as well as some specific feature extractors. The IIR filter bank was designed from a digitalised analogue bandpass filter template. The template could be re-optimised for hardware based on STN-LFP experimental recordings and findings (Liu & Van der Spiegel, 2017). Spectral partitioning could be flexible if analogue programmability is introduced (at the cost of more circuitry specialisation and complexity) (Liu & Van der Spiegel, 2017). Meanwhile, other specialised circuitries

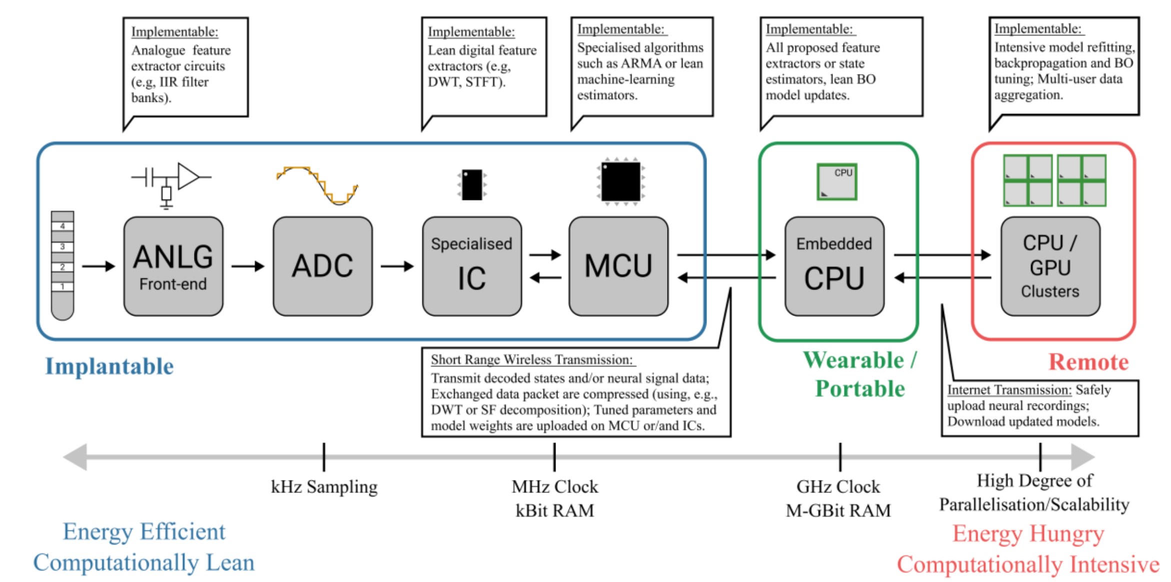


Figure 3.13 – Integration layers for neural interface system; In implant, signals are acquired by an analogue (ANLG) front-end and an ADC; Feature extractors are implemented on specialised ICs or MCU; MCU can also perform decoding using lean algorithms and transmit the data out of the implant; A local CPU (e.g., smartphone or another wearable device) can interface with the implant, collect data, send high level-commands, and host more intensive algorithms for decoding or re-adjust embedded parameters; The CPU serves as a relay to send decoded neural states to target devices; Recorded neural data can also be sent over to remote servers for more intensive processes such as model fitting and tunning; Processes on the (left) side need to be computational lean (to minimise heat dissipation in and power consumption of the implantable) and need to operate in real-time at a frequency relative to that of the neural signal; Processes on the (right) can afford to be computationally intensive and operate more slowly, for instance, to deliver parameter updates.

have been proposed and successfully applied for direct analogue estimation of band power, such as narrow-band orthogonal chopper pairs (Avestruz et al., 2008) or Gilbert transformer (Liu & Van der Spiegel, 2017). Time-division multiplexing would then permit sharing the different analogue and ADC stages across both the channel and spectral partitioning dimensions, limiting the need for duplications.

Features can only be extracted through once the ADC has been carried out. Standard operations could be efficiently carried on specialised integrated circuits (IC) modules. The fast-Fourier transform (FFT) (Stanslaski et al., 2018) for STFT computation and DWT (Gagnon-Turcotte et al., 2019; Yang et al., 2014) modules have been employed in BCI hardware. Those components are supported by widespread use in many other applications, but programmability, notably accurate quantisation of filter coefficients, is always of concern. IC modules have been be directly integrated into a single pipeline (Liu & Van der Spiegel, 2017) or queried by a master microcontroller unit (MCU) (Gagnon-Turcotte et al., 2019; Rouse et al., 2011; Stanslaski et al., 2018).

Alternatively, feature extraction could be directly performed on a MCU. MCUs could offer more programming flexibility to implement complex algorithms with adjustable settings such as the ARMA extractor or other compact machine learning algorithms (Zhu et al., 2020). MCU typically support MHz clock cycles and Mbits of random-memory access, which should be sufficient to host most of the extraction methods proposed (Figure 3.11). Being non-specialised ICs, MCUs would come at the cost of more power consumption and more chip footprint. There is no lack of incentives to develop new types of MCU meeting implantable specifications.

The digitalised signal must be then transmitted from the implantable acquisition device to an external relay unit for real-world application. The transmission can represent a vital bottleneck within the BCI system, and minimising the bit count is here essential (Liu & Van der Spiegel, 2017; Rouse et al., 2011). Transmitting compressed decoder features could be interesting, especially those forming a sparser representation of the signal, such as DWT-packet and ARMA coefficients, or components extracted through SF of the DWT or STFT. The relay unit itself is most likely to be connected to an external wearable/portable device, such as a smartphone. In any case, this device should be powered by a powerful MCU or onboard CPU with an operating system capable of supporting the more computationally intensive stages of the decoding pipeline, such as larger machine learning algorithms. BO parameter tuning, a non-time-sensitive task, could be continuously carried out as a background process and regularly queried for parameter firmware updates. The device should also relay through its local network control information integrating the decoded neural state to other actuating devices in a closed-loop system (Figure 1.1). Subsequentially, more intensive processes, primarily based on gradient-descent optimisation, may be done remotely by sending specific data packages to a remote server destination. Offline computationally intensive units such as graphical processor units (GPU) or large CPU clusters could train and tune the more demanding pipeline stages. Data from multiple BCI users could easily be merged to enhance machine learning training towards deep learning.

J. Chapter Summary

This work introduced unified solutions for decoder design, parameterisation, and tuning. Conventions were proposed to effectively manage data flow through the decoder pipeline for real-time processing and fitting of the various composing algorithms. The specificity of STN-LFP datasets and the BCI task to be decoded were carefully considered. Although the propositions made here were tailored to these datasets recorded from DBS implants, most of them should be generalisable to other neural signal modalities. Emphasis was then further placed on the feature extraction stage of the pipeline and implementing different TF based methods, which have proven to be effective for online feature extraction. A single unified framework was developed for implementation based on the principal design paradigms observed in the literature. Efforts were also put into integrating channel selector and spatial filter stages within the feature extractor to help filter or decompose parallel streaming channels. Recommendations have been made to improve further the methodology and how to deploy the processing and extraction stage on BCI hardware. The tuning methods were tested and the feature extraction methods in the subsequent chapters. Dimensionality reduction, classification and regression algorithms for the state estimator stage are also then further presented.

Chapter 4. Discrete State Decoding

A. Introduction

This chapter presents the decoder implementation and the performance results for the discrete state decoding tasks. The structure of the discrete state estimator and the associated algorithms for classification, state transition, feature selection, dimensionality reduction, spatial filtering, performance and state-feature relationship evaluation are presented. Performance obtained from tuned Bayesian optimised decoders for action detection and laterality selection are reported and discussed. An example of a backpropagation optimised decoder is presented and has previously been reported in Martineau et al. (2020).

1. Motivations and Assumptions

The approaches adopted in past studies to discrete state decoding from STN-LFP motivated the formulation of the BCI tasks, what they should demonstrate, and the assumptions under which the results obtained for the decoding task hold true.

A crucial milestone for STN-LFP decoders was to demonstrate that bulk motor regime, or discrete motors state, changes could be accurately predicted from STN-LFP continuously, or asynchronously in time. If successively passed, this milestone would validate that STN-LFP could serve as a reliable neural modality for a series of elementary control applications, including adaptive DBS triggering (He et al., 2021; Little et al., 2013). Aiming for asynchronous decoding promised the user a more natural experience for control and opened a broader range of applications. Solving decoder asynchrony was far more akin to a signal detection problem, such as the one found in the literature regarding tremor or seizure monitoring from neural signals (Mace et al., 2014; Yao et al., 2020). The imposed constraints were, for instance, that the sliding analysis window of the decoder should be relatively sized to the length of the state change and in doing so, only capture the neural-dynamic patterns temporally adjoint to the action. A simple action detection task was devised to asynchronously predict the application of isometric grip force against rest.

In addition, to determine when a state change occurred, it was also important for the decoder to recognise the end-effector recruited by the user to realise the action. Being able to focus attention from one end-effector to the other would open multiple parallel degrees of freedom. End-effectors could describe, for instance, different limbs (e.g. hand or feet) or the laterality of the body (left or right hand). The latest was straightforward to test as DBS electrodes are commonly implanted in pairs, one in each hemisphere. Laterality selection was hence studied. Unlike in previous studies (Mamun et al., 2015; Mamun, Huda, et al., 2012), the asynchrony constraint was also maintained to track the hand selection temporally throughout an action.

Moreover, Fischer et al. (2017) pointed out the discernability between the STN-LFP associated with imagined and acted movements may still be too subtle to be realistic exploitable. For the former, the user would remain steady imagining movements while the decoder acts the movement response through prediction. Such a motor-imagery paradigm could be loosely qualified as 'mind control'. With non-imagined movement, the decoder would aim to reproduce, or shadow, the user movement. This type of application is relevant for adaptive DBS or, for instance, to drive an exoskeleton. The decoder would still act as a motor-state observer to control the stimulation or robotic assistance delivery. Therefore, it was decided to abide by this second paradigm as it was realistically achievable, imposed easier procedures for data gathering, and retained meaningful applications.

Lastly, it was assumed that the changes in STN-LFP activity were entirely endogenous. Unlike exogenous triggers, endogenous biomarkers should not rely upon stimulating sensory events to trigger a predictable change in neural activity, as in the case of the P300 response, a precise type of event-related potential caused by visual stimuli (Nicolas-Alonso & Gomez-Gil, 2012). Instead, the changes in neural activity should be completely internal and only result from a volitional desire of the user to carry out a motor action (Nicolas-Alonso & Gomez-Gil, 2012). Most of the experiments measuring STN activity did incorporate some form of visual cues or even auditory cues to provide different task instructions and help pace the experiments. It was assumed that the sensory-motor feedback induced by this cue during the task execution did not act as a confounding factor. Enough evidence provided by Spraker et al. (2007) supported that BG activity changes are principally associable to voluntary changes in motor states and not external stimuli. More importantly, Kühn et al. (2004) showed that both externally cued and self-paced movement triggered β ERD. The endogenous assumption thus seemed reasonable to hold.

2. Technical Investigation

Additional technical points of interest were as well investigated. Understanding the complexity of a particular design solution was necessary. Indicators such as the length of the analysis window (3-39), the number of features selected (3-40), model overfitting (3-47) helped to measure complexity tangibly. Complexity had inherent implications on, for instance, hardware deployment (Zhu et al., 2020), system latency, time allocated for model fitting and tuning. The effectiveness of channel, component and feature selection were also seen as intimately linked to model complexity. Reduced model complexity in that sense was posited by Shah et al. (2018) to potentially act as a form of regularisation and consequently lead to an increase of performance. Different dimensionality reduction technics were implemented and compared to evaluate this claim.

More insight on the impact of the decoder design and its parameterisation was necessary to correctly validate key design features (such as the choice of feature extractor method, classifier type or parameterisation). It was essential to understand which parameter could be adequality leverage to deliver the optimal task performance. This objective was primarily achieved by comparing the performance of different decoder architectures, tuning improvement, and sensitivity of the performance to a given parameter. Understanding the decisions of the different optimisation algorithms was essential to constrain future designs better. Eventually, the goal was to assert the scope of decisions that should be left to the tuning algorithm and hence the possible degree of automatisation. Conjointly, it was hoped that task performance would also be increased by improving the feature representation extracted from the signals. It was important to assess if extraction methods relying on the parameterised DWT approach, on ARMA modelling and SF component extraction could yield better and individualised representations.

Once enough decoding solutions, the effect of parameterisation, the gains made by tuning and model complexity were investigated, the performance bounds of the decoder were made more precise. Those criteria helped perform model selection for an asynchronous STN-LFP BCI system. Testing and tuning all these decoders always remained costly (either computationally or timewise). Tan et al. (2016) prior to model fitting, measured β ERD and γ ERS modulation ratios during isometric grip applications. These ratios were then successfully related to the final model performance and were shown to be insightful predictors. Hence metric functions capable of rapidly measuring the relationship strength between the entirety of the recorded STN-LFP features and the experimental performance task were investigated. From past studies, it was known that performance across participants was unequal (Niketeghad et al., 2018; Tan et al., 2016). Those metrics could then serve as a predictor of the final BCI performance and help screen for best candidates.

B. Decoding Task and Participants

First, the recording data were assigned and correctly labelled for each task. The decoder needed to estimate a discrete target label $y[k] \in \{0,1,2 \dots N_y - 1\}$ with $k = 0,1,2 \dots \frac{N_k}{R} - 1$ where N_y is the number of possible states. Unless specified otherwise, the discrete state decoding tasks were simplified into binary problems such that $y[k] \in \{0,1\}$.

The action detection consisted in identifying wherever the BCI user decided to start, hold and stop a voluntary motor action. Participants in the ‘Single Hand Grip’ dataset (Appendix A.A) were provided with a hand-held dynamometer (Fischer et al., 2017; Tan et al., 2016), as depicted Figure 4.1a. The participant was then asked to exercise as much grip force as possible following a visual or/and auditive cue for a given amount of time and then relax. The output of the dynamometer served as visual feedback under the form of a graphical gauge when different force levels were demanded. The participants were asked then to adjust their grip force to match the gauge. Given the dynamometer trace, the action label was defined at first using a threshold. The label was then manual corrected to remove dynamometer artefacts, short involuntary movements, and transients at the edges of the force pulse truncated because of the thresholding. $y[k] = 0$ here encoded a rest state while $y[k] = 1$ an active state where isometric force is applied (Figure 4.1a).

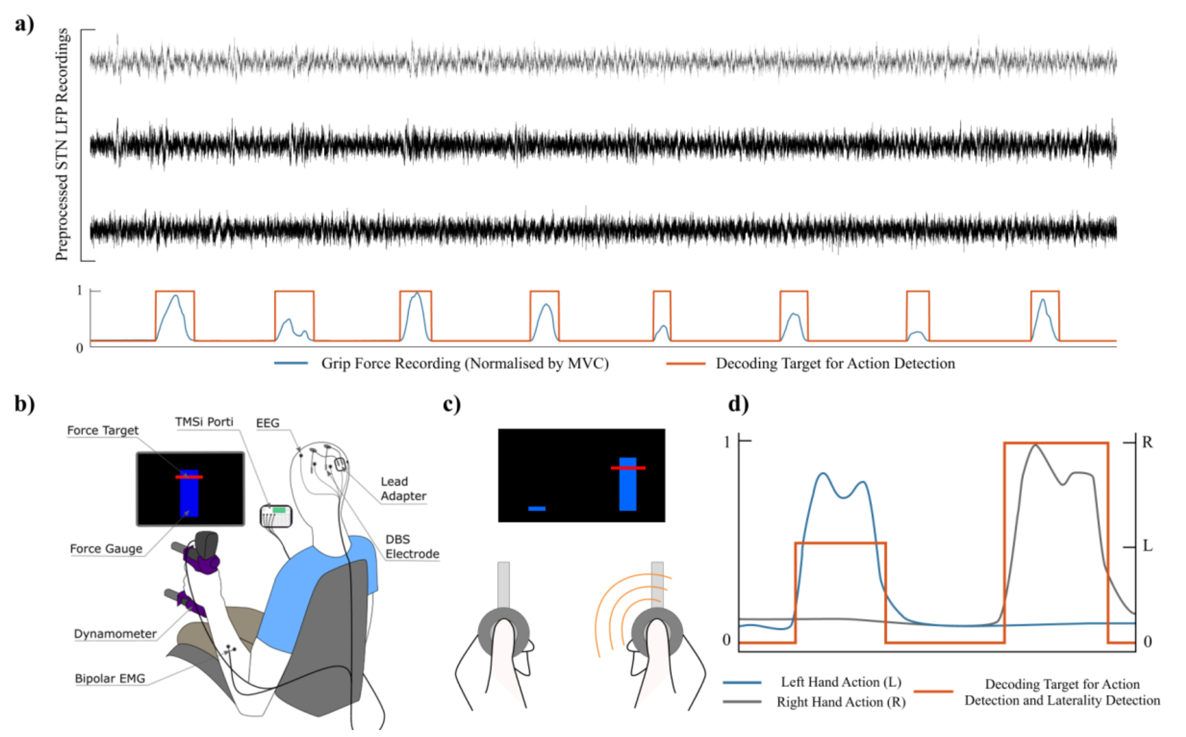


Figure 4.1 - Illustration of the experimental setting and data collected; **a)** Example of pre-processed STN-LFP of signals from Sub01_Level_Right (Appendix A.A) and corresponding action labels recorded with a dynamometer; The decoded target for action detection (**Red**) is obtained by thresholding the force signal (**Blue**); **b)** Recording setup with DBS participant performing the 'Single Hand Grip' experimental protocol (Appendix A.A); DBS leads are directly connected through a specialised adapter to an EEG recording amplifier; Participant use a hand-held dynamometer to perform the experiment task (the force is normalised by their respective MVC); Visual cue and force feedback are provided through a graphical interface; EEG and EMG signals can also be simultaneously recorded for other investigations; **c)** Setup for Dual Pinch' paradigm (Appendix A.B); Participants use two smaller pinched-dynamometer, one held in each hand; Visual cue and feedback indicate which hand should carry the action; **d)** Diagram explaining how dual-hand data is labelled; Two normalised force signals are recorded; The decoding target is composed as such; Left actions (**L**) is assigned to 1; Right actions (**R**) to 2; Rest is assigned to 0 if action detection is carried simultaneously (Martineau et al. 2020).

The dataset 'Dual Hand Pinch' (Appendix A.B) was recorded according to the methodology established by Fischer et al. (2017) and served to train decoders for the laterality discrimination task. Two pinch dynamometers were handed to the participants Figure 4.1b. The visual feedback provided by a graphical gauged enabled them to match their grip to the desired level while using the correct hand (Figure 4.1b and c). Again $y[k] \in \{0,1\}$ at each time step. Every timestep where the dynamometer trace surpassed the threshold, $y[k]$ is labelled 0 for a left pinch action and 1 for a right pinch action.

For the most part, action detection and laterality detection were kept separated into two distinct problems. In practice, an action detector would run continuously and trigger the laterality selector. Since the selector could be triggered at any point under asynchronous conditions, it had to be also fitted continuously in time. When both action detection and laterality selection were combined into a single supervised problem, then $y[k] \in \{0,1,2\}$ where 0 is the resting state where no action occurs, 1 is a left pinch action and 2 is a right pinch action (Figure 4.1d) (Martineau et al., 2020).

C. Discrete State Estimation with Sequential Fit and BO Tuning Approach

The approach to discrete state estimation was the following. Feature space and state relationship metrics were investigated. Then the state estimator pipeline was designed. The usage of different classifiers and dimensionality reduction strategies were investigated for this purpose. Performance objective functions and offline experiments for BO-tuned decoders are finally defined.

1. Measuring Feature and State Relation Strength

Prior to use X (or X_e) to train an estimator or tune a complete pipeline, different metric functions were used to estimate the strength of the predictive relationship between a feature or the entire feature set and a motor state. The interclass separability based on the F-score (ISF) measured how a single feature mean behaviour was displaced when switching from one discrete motor state to the other (Mamun, Mace, et al., 2012). For a complete feature set and a two-class problem (X, y), the intra-class separability ratio was measured as:

$$\mathrm{ISF} = \frac{|\mu_o^2 - \mu_1^2|}{\sigma_0^2 + \sigma_1^2} \tag{4-1}$$

where μ_i and σ_i are respectively the mean and standard deviation of a given feature in X when $y[k] = i$. Figure 4.3 provides examples of the different features distributed between the different states and the perceptible mean shifts when a regime change occurs. Considering the temporality of x, the ISF was analogous to a signal to noise ratio. The concept of separability was further extended to the feature set by generalising (4-1) to the idea of class separability (CS). Different ways to quantify this metric were available (Fukunaga, 1990), but the following definition was retained:

$$\mathrm{CS} = \mathrm{trace}(S_w^{-1} S_b) \tag{4-2}$$

where S_b and S_w are respectively the between-class and within-class scatter matrices defined as:

$$S_b = \sum_{i \in \{0,1 \dots N_y - 1\}} \frac{1}{N_i} (\mu_i - \mu)(\mu_i - \mu)^T \tag{4-3}$$

$$S_w = \sum_{i \in \{0,1 \dots N_y - 1\}} \frac{1}{N_i} \sum_{k=0}^{N_i - 1} (X_i[k] - \mu_i)(X_i[k] - \mu_i)^T \tag{4-4}$$

where μ is the mean vector of X, N_i the number of samples in class i, X_i the feature sample assigned to class i (where $y[k] = i$) and μ_i the mean vector of X_i. The criterion was as well reformulated as (Fukunaga, 1990):

$$\begin{gathered} CS = s_b^T S_w^{-1} s_b \\ s_b = \mathrm{sign}(S_b[0, :]) \odot \sqrt{\mathrm{diag}(S_b)} \end{gathered} \tag{4-5}$$

where $\odot$ is the pointwise operator.

Moreover, entropy and information-based approaches were commonly found in DWT-packet decomposition strategirs (Brechet et al., 2007; Rutledge, 1998) and have been shown to efficiently capture he longitudinal content of timeseries (Han & Liu, 2013). The usage of mutual information (MI) was thereby investigated. Sums or averages (to compare feature spaces of different sizes) of MI could be easily computed to assess the informativeness of a complete or sub-set of features. MI was computed using the implementation scikit-learn of the k-nearest MI approximation algorithm proposed by Pedregosa et al. (2011) and Ross (2014). In addition to capturing non-linear relationships between labels and features, the computation of MI has proven to be more reliable than that of CS. S_w is a symmetric matrix of rank two and is guaranteed, in theory, to be invertible. In practice, however, S_w could be ill-conditioned, resulting in failed inversions. Being more robust, MI was hence applied to the DWT-packet (Chapter 3.E) and for channel selection (Chapter 3.D). While MI was described as an 'ideal statistic' by Ross (2014), it could be less intuitive to interpret than the more straightforward ISF and CS.

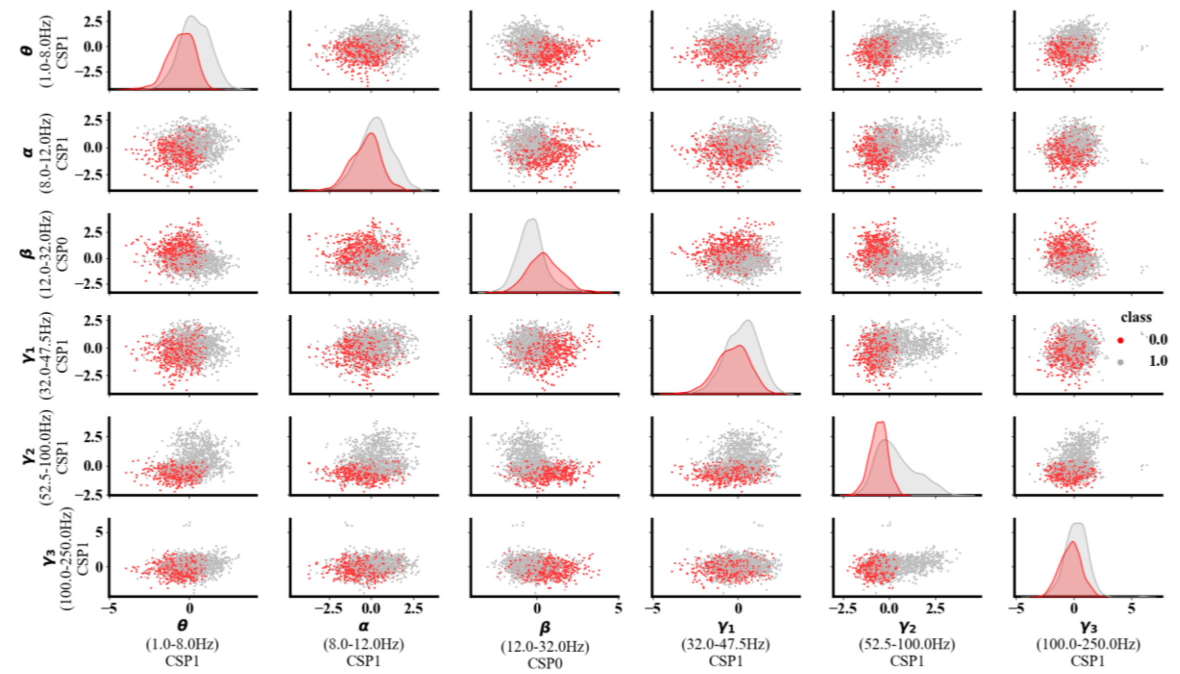


Figure 4.2 – Example of feature extracted for Sub01_Level_Right (Appendix A.A) in an action detection task using paired feature-to-feature scatter plots (off-diagonals) and probability density estimation from histogram for each feature (on-diagonal); Datapoints are separated in rest (**Grey** or 0), and action (**Red** or 1) classes; Features were extracted using the IIR-CSP extractor for six frequency bands; The first rest and action CSP components were extracted; In each band, only component with highest ISF were plotted; Rest CSP was only selected for the β band.

2. Robust State Estimators for STN-LFP Decoding

The clusters layout in Figure 4.2 represents the type of data encountered for STN-LFP. Specifications were drawn from such layouts and retained for the design of the state estimator:

- **Robustness to noise**: the feature space SNR characteristic was likely to be excessively poor, meaning many sample points overlap between feature clusters. Finding the correct feature extractor and state-estimator pair was judged extremely important.

- **Data Temporality**: the sequence order in which the data is presented to the state estimator was important. The label and features were likely misaligned temporally (Farina et al., 2007; Shah et al., 2018). These aspects can be captured by epoching the feature array using (3-38) and using a state transition function.
- **Feature Space Size**: the type of estimator employed needed to accommodate a large feature space given the cubic nature of (3-40).
- **Imbalance dataset**: the data collected for action detection was imbalanced, containing more resting states (all the $y[k] \in \{0\}$) than active states (all the $y[k] \in \{1\}$). Imbalances could introduce biases in the estimator fitting process and in the case of gradient descent fitting run the optimiser into local minima. Dataset imbalances are reported in Appendix A.
- **Non-linearity**: The possibility of non-linear boundaries between feature clusters could not be ruled out in advance, which meant that estimators capable of non-linear representation should be tested.

Figure 4.3 present the state estimator pipeline. The structure was centred around a classifier capable of predictive probabilities p of the current state given the STN-LFP features (Chapter 3.G). Through evaluation, the state transition function selected the next stage passed on p. The pair $(\tilde{y}, y)$ was then available to calculate the overall pipeline performance via the score function S. The sampler transformation h_s acted on both the feature space and the label vector (X, y) to produce a resampled pair (X_s, y_s). If $X \in \mathbb{R}^{\frac{N_k}{R} \times N_f}$ and $y \in \mathbb{N}^{\frac{N_k}{R}}$ then $X_s \in \mathbb{R}^{N_s \times N_f}$ and $y_s \in \mathbb{N}^{N_s}$ such that $N_s < \frac{N_k}{R}$. The library imblearn (Lemaître et al., 2017) offered a collection of algorithms for this precise purpose. After review, the repeated-nearest-neighbours sampler was chosen (Lemaître et al., 2017). Estimator fitting required a training pair $(X, y)_{train}$ and the evaluation of a testing pair $(X, y)_{test}$. This structure was duplicated N_{fold} during cross-validation (Chapter 3.G).

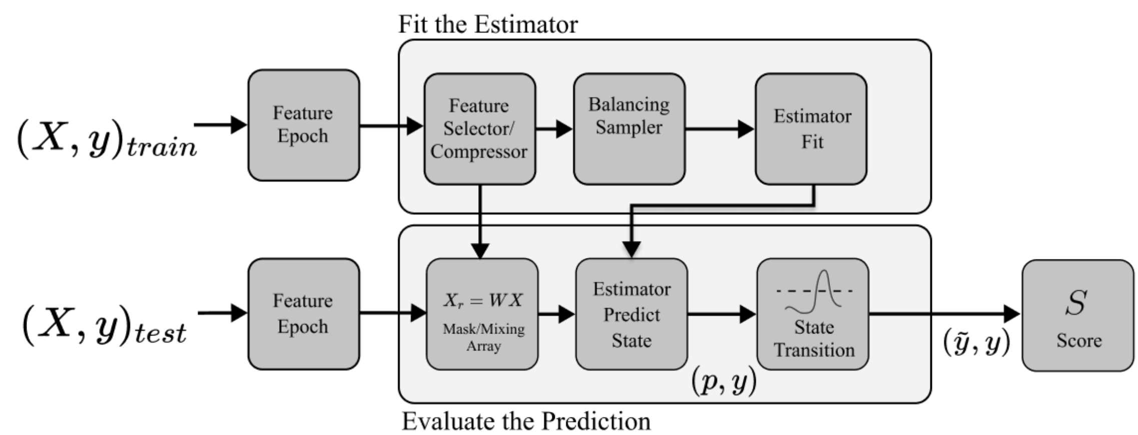


Figure 4.3 - State estimator pipeline with training and validation branches; Optional feature compressor (performing dimensionality reduction) is first applied; A sampler is used to further balance the training dataset; An estimator (of several classifiers, NB, LR, LDA or SVM) predicts the state probability p which is then used for state transition; Pipeline is duplicated during cross-validation.

3. Spatial Filter Solutions and Feature Selection for Discrete State Estimation

Depending on the type of estimator, a large feature space X could easily overwhelm its fitting process either by causing model overfitting or by inducing excessively long model fitting times. As defined in Chapter 3.D, spatial filters reduced the dimensionality in the channel space at the feature extractor stage of the pipeline. Although common spatial patterns (CSP) is a standard supervised-decomposition algorithm employed in BCIs which yields better discriminative components for pairwise classification problems (Lotte & Guan, 2011), it had yet to be apply to a STN-LFP specific problem. Given a binary signal label y, the training example signal x was separated into two subarrays, x_0 and x_1, each only containing the datapoints of a same state. The two correlation matrices $C_0 = x_0^T x_0$ and $C_1 = x_1^T x_1$, were assembled. Extension to complex signal arrays for the STFT and RWT was possible (Falzon et al., 2010) by defining $C_0 = x_0^H x_0$ and $C_1 = x_1^H x_1$ using the Hermitian operator. Given C_0 and C_1, the CSP problem aimed at fitting a set of SFs, w, which maximised either of the following Rayleigh quotients:

$$\begin{gathered} w = \underset{||w||=1}{\operatorname{argmax}} \frac{w^T C_0 w}{w^T\left(C_1 + \alpha_{L_2} I\right)w} \\ w = \underset{||w||=1}{\operatorname{argmax}} \frac{w^T C_1 w}{w^T\left(C_0 + \alpha_{L_2} I\right)w} \end{gathered} \tag{4-6}$$

where I is the identity matrix and α_{L_2} a Tikhonov regularisation constant (Lotte & Guan, 2011). CSP filters ultimately maximised the signal variance for the first state while also minimising it in the second state. The filters w were found by solving the following eigenvector problem:

$$\begin{gathered} C_0 w = \lambda(C_1 + \alpha I)w \\ C_1 w = \lambda(C_0 + \alpha I)w \end{gathered} \tag{4-7}$$

W was then assembled by ranking the eigenvector according to their respective eigenvalues and stacking them together. Each row was as well whiten by $\sqrt{\lambda^{-1}}$ (Nicolas-Alonso & Gomez-Gil, 2012). Two separate hyperparameters, $N_{c_0}, N_{c_1} \in [1, N_{ch}]$, were defined so to select the number of components provided by each decomposition. By default, the CSP filters were configured so that $N_{c_0} = 1$ and $N_{c_1} = 1$; i.e. to have two filters, one optimised for the rest class ($y[k] = 0$) covariance against the action class ($y[k] = 1$) covariance and vis-versa. Regularisation was introduced because the CSP algorithm is known to be prone to overfitting (Lotte & Guan, 2011; Robinson et al., 2013). Popular approaches were based on ‘subject-to-subject’ transfer consisting in correcting the covariance matrices by shrinking them towards more generic structures built from the general population data (Lotte & Guan, 2011). CSP could also be spatially regulated by using the physical geometry of the channel arrangement to ensure that each electrode shared similar weights with its neighbours (Lotte & Guan, 2011). Both methods were not possible given that individual subject recordings were associated with different electrodes that possessed different numbers of contacts, geometry and were implanted at variable locations.

Consequently, the Tikhonov method was chosen since it was agnostic in this regard while remaining effective (Lotte & Guan, 2011).

Although CSPs are popular in the BCI literature, they have been shown to perform more poorly than other feature selection or dimensionality reduction technics, which are directly applied on the feature array X (Gruenwald et al., 2019). Feature selection or feature space dimensionality reduction created a more compact space X_r such that $X_r \in \mathbb{R}^{\frac{N_k}{R} \times N_r}$ and $N_r, \leq N_f$. A compression ratio, r_c was defined and set as a hyperparameter such that $r_c \in [0,1]$ and that:

$$r_c = \frac{N_r}{N_f} \tag{4-8}$$

where N_r is the number of selected features or feature components.

The following methods were investigated and implemented: **PCA 1)** Principal Component Analysis (PCA) was the simplest and most common dimensionality-reduction. PCA has been shown to be effective for STN-LFP decoding (Golshan et al., 2018, 2017, 2016; Zaker et al., 2016), EEG (Lotte & Guan, 2011) and ECoG classification (Gruenwald et al., 2019). PCA aimed at solving a single Rayleigh quotient:

$$w = \underset{||w||=1}{\operatorname{argmax}} \frac{w^T C_X w}{w^T w} \tag{4-9}$$

where C_X is the convariance matrix of X. The sklearn (Pedregosa et al., 2011) singular-value decomposition was employed for its robustness. The number of filters w were selected according to r_c to assemble the final mixing matrix. **SFA 2)** linear slow feature analysis (SFA) could be described as an extension of the PCA algorithm aiming to rank feature components according to their first-order time derivative (or velocity). The finite difference and decorrelated velocity signals X_v were given by:

$$X_v[k] = W^T(X[k] + X[k-1]) \tag{4-10}$$

where is the W^T is the mixing matrix obtained from the PCA algorithm. The SFA problem aimed at finding the slowest components through the following PCA problem:

$$v = \underset{v=||1||}{\operatorname{argmin}} \left(\frac{v^T C_v v}{v^T v} \right) \tag{4-11}$$

where C_v is the correlation matrix of the finite velocities, and v is the component filter vector. The i^{th} linear SFA component was then given such that:

$$X_r[k,i] = v_i^T W^T X[k] \tag{4-12}$$

where v_i is the eigenvector found in the velocity PCA. Preliminary evidence suggested that SFA provided more natural feature components than PCA for a classification problem set in the time domain (Wiskott & Sejnowski, 2002). The implementation of the algorithm proposed by Wiskott-lab (2021) was employed. **SQFE 2)** sequential feature elimination consisted in iteratively eliminating features from X to form X_r. A direct approach to SQFE would have consisted in using cross-validation estimator fitting to eliminate features based on which feature sub-set yielded the best performance (Mamun et al., 2015). However, the process needed is iteratively repeated until the desired number of features is found. Unfortunately, using cross-validation in such a way was highly computationally intensive as it required numerous estimator fitting. Fukunaga (1990) instead proposed a recursive formulation of the class separability criterion CS for a faster evaluation of decreasingly smaller feature subsets. First CS_{N_f} was computed for the complete feature set. A first feature was then eliminated by evaluating the subset's criterion $CS_{k(N_f-1)}$ with $k = 0,1,2 \ldots N_F - 1$ such that:

$$CS_{k(N_f-1)} = CS_{N_f} - \frac{(S_w^{-1}[k,:]^T s_b)^2}{S_w^{-1}[k,k]} \tag{4-13}$$

where S_w and s_b are the scatter matrix and vector defined in (4-5). (4-13) avoided performing the inversion of S_w at every criterion evaluation and considerably accelerated the process provided the total number of features remained reasonable. The feature to eliminate was then indexed by $k^* = \underset{k}{\operatorname{argmax}}\, CS_{k(N_f-1)}$. Provided the appropriate entry of the S_w^{-1} were then removed, this process could be repeated at each elimination level l to find $CS_{k(N_f-l)}$ with $k = 0,1,2 \ldots N_f - l$ until the desired number of features were eliminated. **Estimator Regularisation 3)** Certain classifiers (such as logistic regression) were implemented with L_1-norm regularisation, which enforced a sparser fitting of the estimator internal weights which acted as a form of feature selection. The sparsity was controlled by the strength of the regularisation set during model tuning. Sparse estimator regularisation removed the need for a separate stage for feature dimensionality reduction or selection altogether (Shah et al., 2016, 2018).

4. Classifiers for Discrete State Estimation

All classifiers possessed a local fitting function with an individually defined objective and were selected from the scikit-learn library (Lemaître et al., 2017) based on the works of He et al. (2021): **Gaussian Naïve Bayes 1)** the gaussian Naïve Bayes (NB) classifier was chosen as a first default classifier. The principal benefits of the classifiers were foremost its simplicity, fast-fitting time and lack of additional hyper-parameters. Fitting the estimator could also be done iteratively as more data is available, thus being interesting for online learning. NB had been already employed in several LFP driven BCI studies (He et al., 2021; Mamun et al., 2015). The limitation of the algorithm was its over-simplistic assumption of conditional independence between the features (H. Zhang, 2004). In the variant of the algorithm chosen, univariate Gaussian distributions are employed to model the probability density function of each feature. **Linear Discriminant Analysis 2)** the linear discriminant analysis (LDA) classifier was previously implemented in previous STN-LFP studies and has proven to be an effective estimator for high-performance applications (Flint et al., 2012; Gruenwald et al., 2019; He et al., 2021). The primary benefits of the LDA were that their fitting possessed a closed-form solution implying a lesser computational burden, and could be naturally extended to multi-class problems. The LDA assumed that the features were distributed according to a Gaussian multivariate probability function. It measures the centroid shifts but does not rotate the covariance matrices between classes (Ledoit & Wolf, 2003). This invariance was the principal limitation of the LDA. The shrinkage parameter $\gamma_c \in [0,1]$ was employed to increase the robustness of covariance estimation (Ledoit & Wolf, 2003). **Logistic regression 4)** the logistic legression (LR) classifier was regularly used in preliminary STN-LFP classification studies (He et al., 2021; Martineau, 2017; Shah et al., 2016; Tan et al., 2019). The LR classifier consists in fitting a simple linear model according to the cross-entropy loss criterion and uses a sigmoid activation function to generate a the discrimination boundary. LR fitting required a numerical solver, such as a gradient descent algorithm. The principal benefits of LR were model simplicity and the fact that different regularisation strategies were implementable. Here, L_1 norm-regularisation was preferred as it enabled some form of feature selection (Shah et al., 2016) and made the LR estimator comparatively more relevant. The regularisation was controlled by a unique parameter C. **Support Vector Machine 5)** Support vector machines (SVM) served as a high-performance standard for many classification problems and have been repeatedly employed in the BCI literature (Golshan et al., 2018, 2016; He et al., 2021; Mamun et al., 2015) of which the radial-basis function (RBF) kernel implementation was the most prevalent variant. The RBF kernel turned the SVM into a very versatile classifier capable of handling non-linear classification problems. Two parameters required tuning: a regularisation constant C and the kernel parameter γ_C (Nandy et al., 2019). The principal limitations of the SVM algorithm were the limited scalability of the fitting process with the number of samples in the feature array, on the one hand, and the fact that the SVM estimator did not yield prediction probability on the other. The later could still be indirectly estimated in a secondary fitting process at the cost of the additional computational burden (Platt, 1999). For these reasons, the PCA dimensionality reduction transform was always employed in conjunction with the SVM (Golshan et al., 2018, 2016). A cap of 64 features was also placed on the total of features such that $N_r = \min(64, \lfloor r_c N_F \rfloor)$ to avoid overly long fitting times.

5. State Transition and Tuning Objective Functions

For action detection, once the state probability p was found, a state transition function was employed to select the next predictable state $\tilde{y}[k]$ based on the previous step $\tilde{y}[k-1]$. A double-threshold system was a robust yet straightforward transition method that could help denoise the classifier output:

$$\tilde{y}[k] = \begin{cases} 1 & \text{if} \quad p_1[k] \geq T \quad \text{and} \quad \tilde{y}[k-1] = 0 \\ 0 & \text{if} \quad p_1[k] \leq (1 - r_T)T \quad \text{and} \quad \tilde{y}[k-1] = 1 \end{cases} \tag{4-14}$$

where p_1 is the action prediction probability $p[k, 1]$, $T \in [0,1]$ is the upper threshold and $r_T \in [0,1]$ is the position of the lower threshold relative to T. In this respect, r_T controlled the size of the deadband ΔT such that $\Delta T = r_T T$. Both (r_T, T) were defined as hyper-parameters. Typically, a decision threshold would be set at the optimal point of the classifier ROC curve, balancing its sensitivity and specificity. Unfortunately, this curve cannot be plottable for a dual-threshold system. Instead, the geometric mean between the sensitivity and specificity was investigated as a performance metric. Given the true-positive rate (TPR) and false-positive rate (FPR) of the pair $(y, \hat{y})$, the geometric mean GM was given by (Cai et al., 2010):

$$GM(\tilde{y}, y) = \sqrt{TPR(\tilde{y}, y) \times (1 - FPR(\tilde{y}, y))} \tag{4-15}$$

Both TPR and FPR objectives were compounded into a single scoring function S which could be handled by the BO tuning algorithm and avoided using a multi-objective optimisation method (Mace et al., 2014). An ideal detector converges to 0 FPR and 1 TPR, resulting in a maximum GM of 1.

Lastly, a general objective function for decoding performance is classification accuracy (ACC). ACC measured the number of points $\tilde{y}[k]$ correctly attributed to each state $y[k]$ against the total number of points. The most straightforward transition was, therefore, implemented as:

$$\tilde{y}[k] = \underset{n \in \{0,1,\ldots N_y - 1\}}{\operatorname{argmax}} (p[k, n]) \tag{4-16}$$

which was extendable to the selection N_y state. ACC was thereby employed as the scoring function for the laterality discrimination task.

D. Offline Decoding Experiments with BO Tuned Models

Multiple offline decoding experiments were carried out. First, decoders were tuned for the action detection task using the GM as a cost function S for BO (using 90 fitting iterations, an gaussian process edge acquisition function and Gaussian process surrogate model, Appendix B). Multiple feature extraction methods were tested, including CHS combined extractors (ARMA-band-CHS, ARMA-coeff-CHS, DWT-wavelet-CHS, HJ-CHS, IIR-CHS, RWT-CHS and STFT-CHS), CSP combined extractors (DWT-wavelet-CSP, DWT-full-CSP, IIR-CSP, RWT-component-CSP and STFT-component-CSP) and the DWT-packet extractor (Appendix D). Those were combined with several classifiers (LDA, LR, NB and SVM-PCA). Dimensionality reduction technics (PCA, SFA and SQFE) were further tested with the LDA classifier and CHS extractors. Performance metrics (including GM, TPR, FPR), optimum decoder parameters, and sensitivity index were collected. The full length of the recording was employed to fit the decoder classifier using 8-fold cross-validation (although CSP filters were fitted using segmented data to estimate the action covariance matrix correctly).

A pre-analysis was included to measure the strength of the relationship of extracted features under default settings (Table 4.1) using the average MI and CS criteria. Monopolar and bipolar settings were then tested. The recordings of 10 participants constituting the 'Single Hand Grip' dataset (Appendix A.A) were included for the action detection task. The same approach was followed for the laterality detection task using the ACC metric as a S function for BO. The recordings of 12 participants constituting the 'Dual Hand Pinch' dataset (Appendix A.B) were employed. The recordings were segmented around each action for decoder fitting with an additional 1.5s edges covering the pre-onset

Table 4.1 - Pipeline parameters contained in ρ and tuned with BO for discrete state decoders including: parameter description and stage assignment, default values stored in ρ_o, search interval for every dimensioned searched from which ρ is drawn; Dimension searched through a log mapping (Y) or through a linear mapping (N).

Stage	Parameter Symbol	Description	Default Value	Search Interval	Log Scale
Channel Selector & Spatial Filter	N_c	Number of selected Channel	2	$[1, N_{ch}]$	N
	N_{c_0}	Number of states 0 CSP components	1	$[1, N_{ch}]$	N
	N_{c_1}	Number of states 1 CSP components	1	$[1, N_{ch}]$	N
	α_{L_2}	CSP regularisation	0	[0, 100]	N
Feature Extractor	n_o	Number of overlaps	3	[0,3]	N
	β_K	Kaiser window parameter	5	[0,15]	N
	θ_1, θ_2	QMF lattice parameter	28.64°, -6.04°	$\left]-\frac{\pi}{2}, \frac{\pi}{2}\right[$	N
	τ	DWT-packet pruning threshold	0	[0,0.05]	N
	η	ARMA decay rate	0.99	[0.5, 0.9999]	Y
	N_{AR}	Number of auto-regressive term	6	[1,12]	Y
	N_{AM}	Number of moving average term	2	[0,6]	Y
	N_w	Length of the feature epoch window	0	[0,17]	Y
Discrete State Estimator	T	Upper detection threshold	0.5	[0,1]	Y
	r_T	Lower detection relative position	1	[0,1]	Y
	C	Regularisation constant for LR	1	[0.001, 10.0]	Y
	C	Regularisation constant for SVM	1	[0.001, 10.0]	Y
	γ_C	Shrinkage LDA	0	[0,1]	Y
	γ_C	SVM-RBF kernel parameter	0.1	[0.0001,5]	Y
	r_c	Feature compression	0.1	[0,1]	N

and post-offset rest period. Parameter estimation for normalisation of the features was completed on those edges. This approach was more robust to capture the displacement of activity from the resting baseline. The SVM constant C was set to 1.0 has was found significantly slow-down the estimator fitting time (Golshan et al., 2016).

Realistic settings were chosen to model real-time constraints. The input frequency of the decoder f_s was set to 512Hz and its output f_d to 16Hz. A target lead τ_y of 125ms was set. It allowed to centre a maximum feature extraction window of 250ms (or 3 overlaps per windows). A maximum feature window of 1s was lastly permitted. Figure 4.4 summarises the relation between the different windows. The frequency band were defined as θ (1-8Hz), α (8-12Hz), β (12-32Hz), γ_1 (32-50Hz), γ_2 (50-100Hz), γ_3 (100-256Hz). One single large-bandwidth low frequency was employed to construct stable IIR filters and γ was partitioned to avoid power line artefacts.

All decoder models were implemented in Python using primarily the following libraries: numpy (Harris et al., 2020) for efficient array computations, scipy (Virtanen et al., 2019) to support signal processing, scikit-learn (Pedregosa et al., 2011) for machine learning and scikit-optimise (Head et al., 2021) for BO. Statistical analysis was completed using again the scipy (Virtanen et al., 2019) and the statsmodels (Seabold & Perktold, 2010) libraries. Sensitivity analysis was carried with the SALib library (Herman & Usher, 2017).

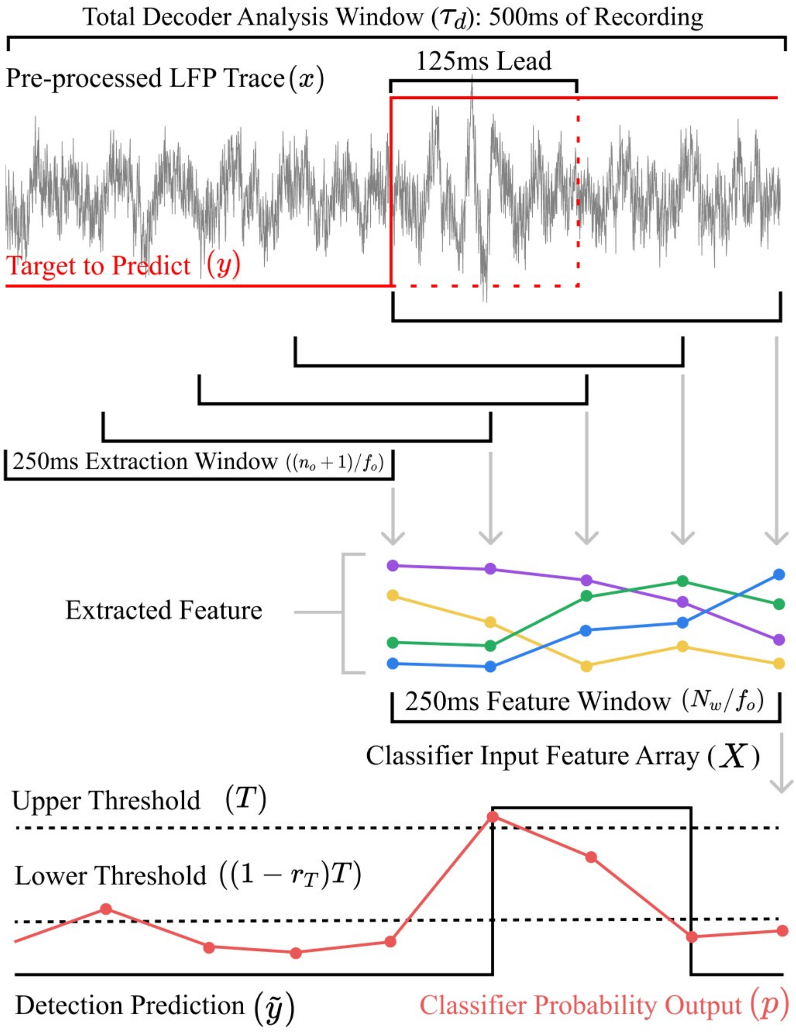


Figure 4.4 - © 2023 Frontiers, Reprinted, from Martineau et al . (2023); Schematic of the data flow within the action detection decoder; Features are extracted in a first epoch window controlled in length by the parameter n_o; Those are accumulated in a second epoch window controlled in length by N_w; The classifier employs the features vector X to a make a prediction p on the state estimate; p triggers the final decoder output.

E. Case study: Integrated Pipeline Tuned with Gradient Backpropagation and Descent

An experimental decoder tuneable through gradient-descent was designed with the pytorch (Paszke et al., n.d.) library and reported in Martineau et al. (2020). Figure 4.5 presents the structure of the pipeline architecture. It included an IIR filter decomposition bank (3-18), a single spatial filter (3-15), a parameterised multi-taper power estimation stage, a feature epoch, linear models (or Wiener filter) and a softmax classifier. The softmax classifier was a natural multi-class extension of the LR classifier and most often forms the final layer of an artificial neural network as it is differentiable. The generated probabilities were employed in the objective function from which the backpropagation optimisation stems. The objective function to train the decoder was the cross-entropy criterion (NLLLoss) (Paszke et al., n.d.; Torch Contributors, 2019) with L_2 norm regularisation. Two elements were tested: first was the optimisation of the feature space through the spatial filter and the taper shape; second, the usage of the softmax classifier to predict either right or left action detection simultaneously. A comparison of pipeline performance was completed with and without those stage optimisation, and the performance of single-action detection alone was also compared to dual-action detection. The advantage of backpropagation optimisation was two-fold: it permitted end-to-end optimisation and unified the tuning problem. It also permitted to unify the decoding problem and avoid using multiple classifiers and objective functions. Some parameters (learning rate and regularisation constant related to the RMSprop optimiser) were tuned using BO. Decoders were tuned for 5 participant recordings (Sub01, Sub05, Sub08, Sub09, Sub11) of the 'Dual Hand Pinch' (Appendix A.B). The recordings were downgraded at an input frequency of 256Hz and segmented (usually in pairs of two actions) into batches to ease gradient descent optimisation.

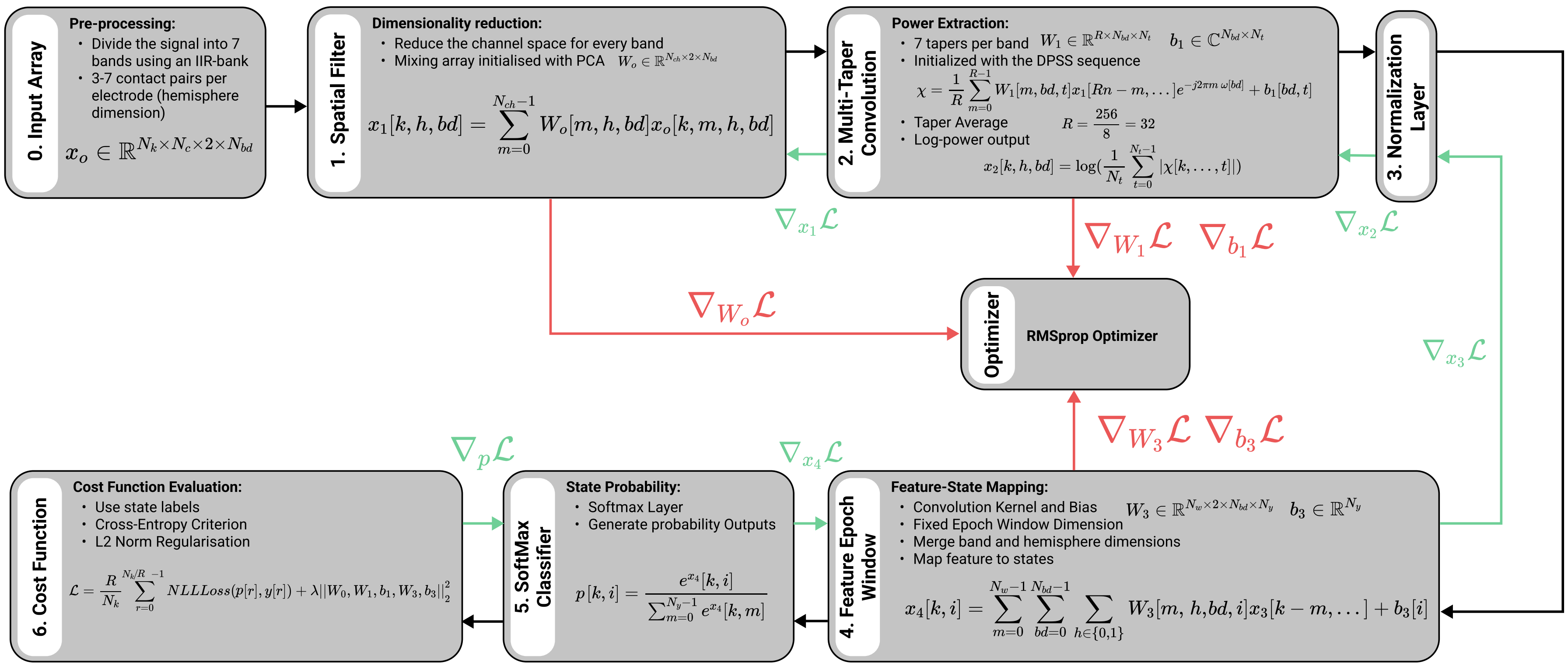


Figure 4.5 - Proposed decoder pipeline optimised with gradient descent approximated through backpropagation; Spatial filter and taper parameters are both optimised; An pre-processed input array is passed to a spatial filter; The filters issue two components per band, one mixing channels of the right hemisphere, the other from the left; multiple tapers are employed to measure the power in each band; The power features are normalised and send to a linear model; The model includes a kernel per hemisphere and per class, Softmax classification issues the state prediction probabilities; Cross-entropy loss criterion is used for final evaluation.

F. Results

This section reports the results for the offline BO tuned decoder and summarises the finding of backpropagation study. First, the outcomes for the feature pre-screening investigation are reported for the action detection and laterality selection tasks. Examples of extracted features were provided for both tasks. Tuned decoder results are then presented for the action detection, decoder equipped with feature reduction, and action laterality results. The case study of backpropagation tuning is presented last. In summary, decoding performance was consistently heterogeneous across participants in the action detection and laterality tasks, and only a subset of participant recordings rendered good decoding performance. For action detection, a median rate 0.75-0.25 (TPR, FPR) performance threshold was reached and for laterality selection, a median rate of 0.7 accuracy. In this last case, choosing between different extractor-classifier pairs did help in finding better individual solutions and tuning did lead to improvement in decoding performances. Results pointed to the un-equal quality of the features extracted from one participant recording to the other as an important factor for decoding. It would be reasonable to posit that this variance in feature quality was likely the source of the heterogeneity of the performance results.

1. Feature Screening

Examples of feature trial averages for action detection are presented in Figure 4.6 for the IIR-CSP for three different participants. The ISF and MI were here computed for each individual feature (Figure 4.7). The CSP algorithm permitted sorting both ERSs and ERDs across the channel space and was practical for rapid visual inspections. Typically, the CSP sensitive to the rest state isolated the β ERD, whereas the CSP sensitive to the action class captured the γ ERS. In particular, mid-γ_2 and high-γ_3 deviated more prominently from the resting baseline during action application and was the most informative for decoding. β ERD and α ERS were informative while the γ_1 and θ were typically the lesser informative bands. Except for the lesser contribution of θ (Mamun et al., 2015), those preliminary results were consistent with the findings of Tan et al. (2013).

The CSP filters also gave some insights on how the ERS/ERDs occurred during the laterality selection task, irrespectively of the channel space dimensions. Figure 4.8 presents trial averages using again the IIR-CSP extractor carried on both left and right actions for participants Sub01 and Sub08 of the laterality dataset. Ideally, the first spatial filter dyad should function asymmetrically. For a given action, the output of a first filter should produce features associable with this action, while the opposite output filter should remain completely invariant. For the given second action, their roles should be then reversed. Precise inflexions were marked by the β ERD on the expected contralateral side while minimal ERD were noted on the ipsilateral side. The α band also tended to de-synchronise with laterality selection. The evaluation ISF and MI metrics on the individual features revealed the importance of mid-range frequencies β and α and higher frequencies such as γ_1 and γ_3. Further details for other feature extractions method are given in Appendix D. Furthermore, the participants in Figure

4.6 were chosen to illustrate a general trend observed across the sampled population. Figure 4.10a and Figure 4.11a indicate each participant's positioning within the cohort based on the average MI of the extracted features. The deviation of the ERS and ERD from the respective band baseline varied significantly from one participant to another. In better cases, γ_2 ERS could peak at 4 standard deviations above while β ERD could be suppressed at 1 standard deviation below. In some participant recordings, no significant deviations of any band activity (except for the lower θ) were witnessable.

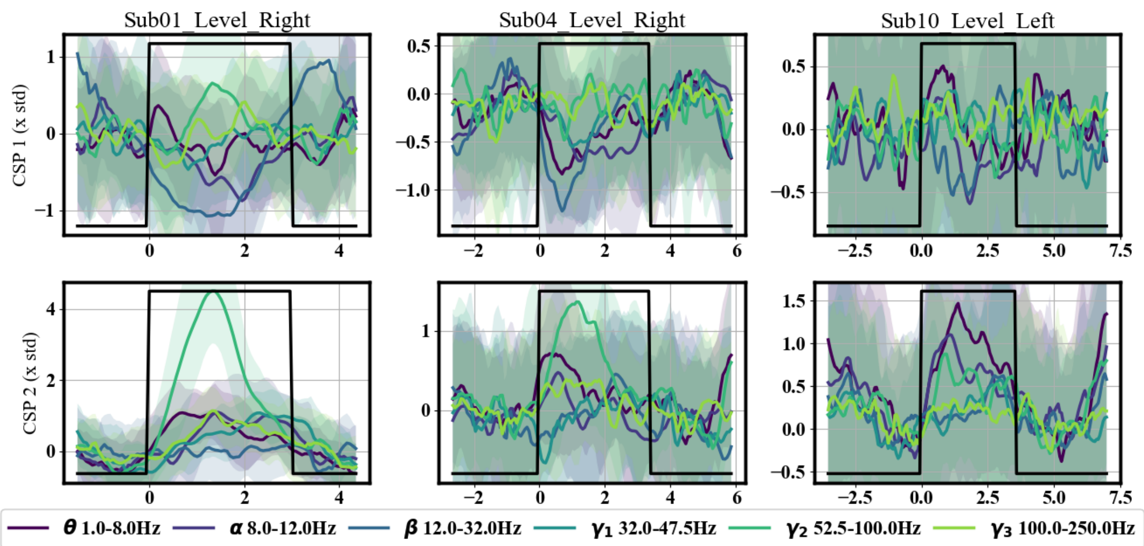


Figure 4.6 - Example trial average response for the IIR-CSP extractor; One active and one rest components were selected ($N_{c_0} = 1, N_{c_1} = 1$); An epoch length of 250ms ($n_o = 3$, for $f_d = 16$Hz) was used; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' dataset (Appendix A.A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid black line indicates the action to detect; Figure is also present in Appendix D.

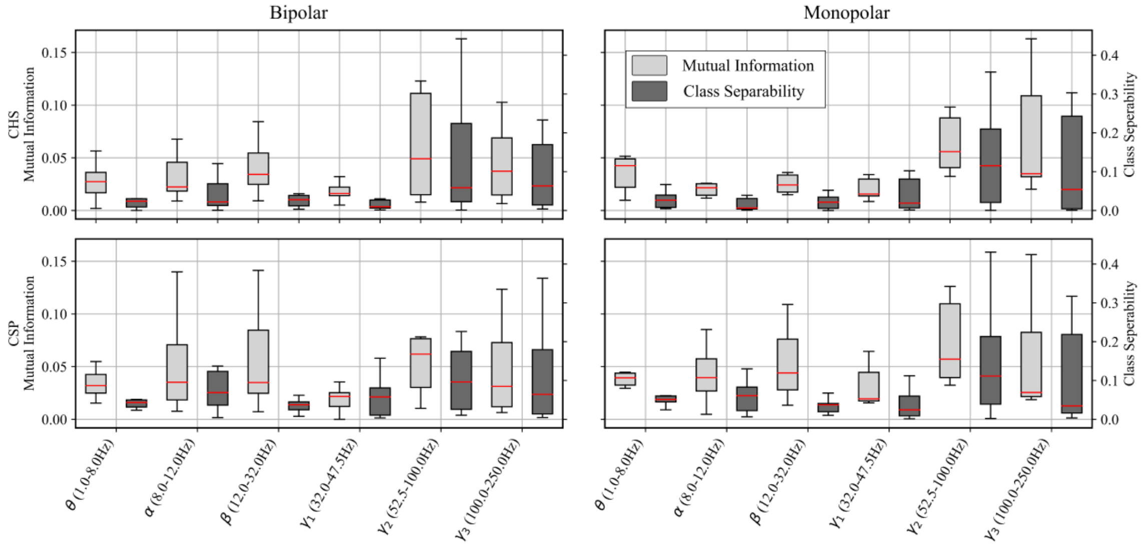


Figure 4.7 - ISF and MI metrics comparison between the IIR-CSP and IIR-CHS for the 'Single Hand Grip' dataset (Appendix A.A); For the CHS method two channels were selected; For the CSP method one rest and one action components were selected; Only the maximum ISF and MI across the channel pair were included in the distributions; Metrics were computed for monopolar and bipolar dataset. Default settings were used for extraction.

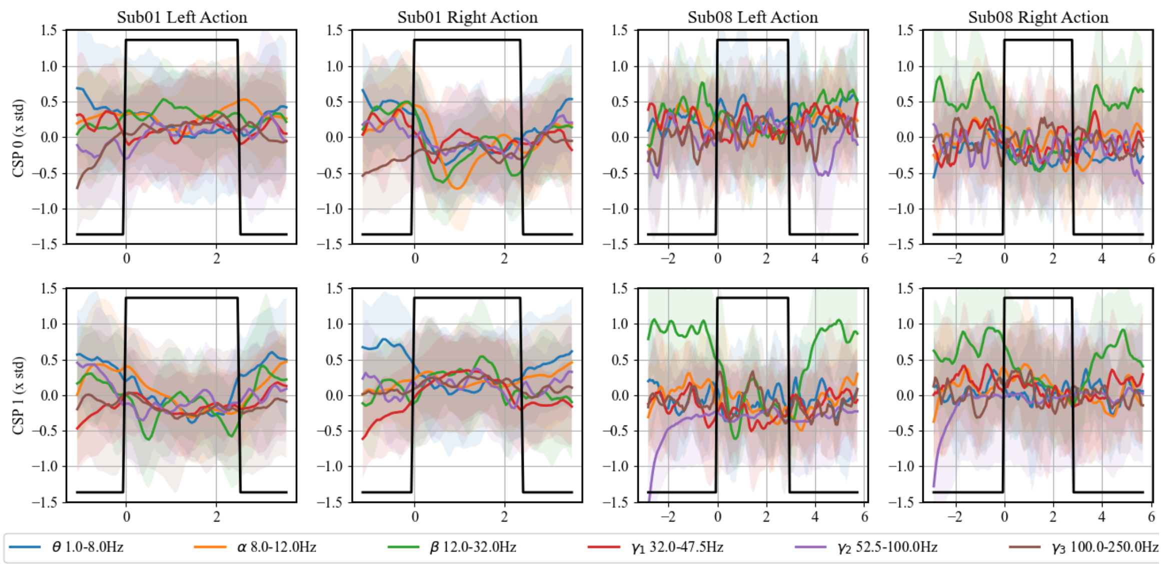


Figure 4.8 - Example trial average response for the IIR-CSP extractor in laterality detection task; One left and one right components were selected ($N_{c_0} = 1, N_{c_1} = 1$); An epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$) was used; The extraction was completed for 2 representative subjects of the 'Dual Hand Pinch' dataset (Appendix A.B) for the action detection task; 26 and 12 left trials were respectively averaged; 26 and 14 right trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid black line indicates part of signal considered for decoding.

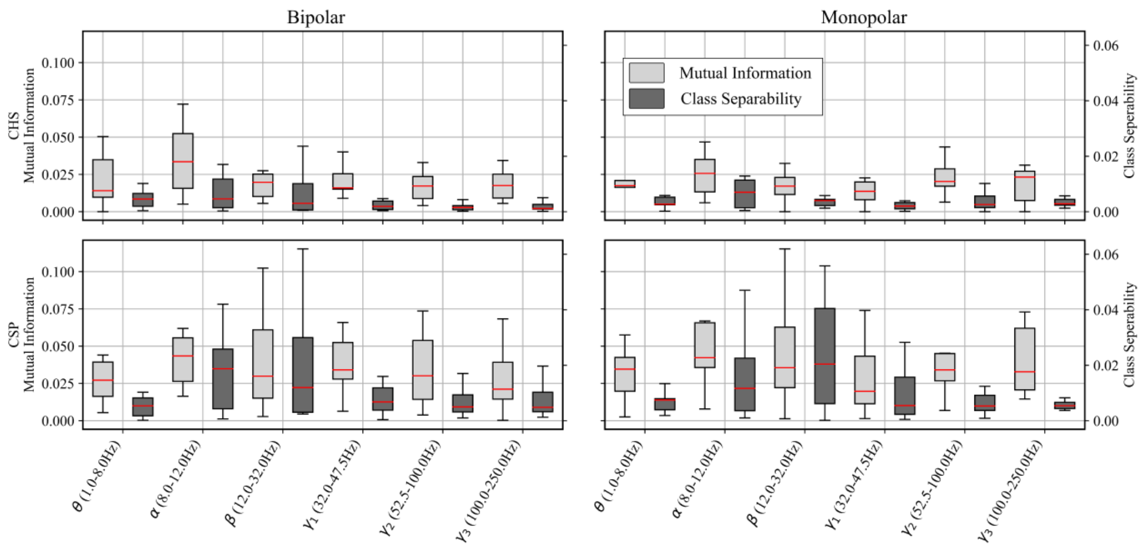


Figure 4.9 - ISF and MI metrics comparison between the IIR-CSP and IIR-CHS for the 'Dual Hand Pinch' dataset (Appendix A.B); For the CHS method two channels were selected; For CSP one left and one right components were selected; Only the maximum ISF and MI across the channel pair were included in the distributions; Metrics were computed for monopolar and bipolar dataset; Default settings were used for extraction.

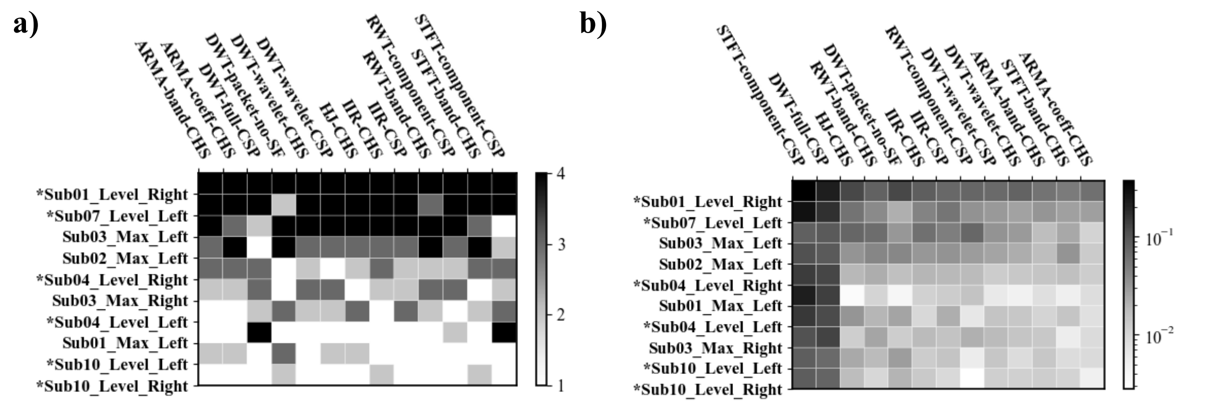


Figure 4.10 – Feature screening results for the action detection task using the 'Single Hand Grip' dataset (Appendix A.A) and every feature extraction method; **a)** Quartile ranking based on MI feature averages; Each participant MI average were assigned to the 1, 2, 3 and 4th quartile; **b)** Absolute ranking based on MI features averages; (*) indicates participants with available monopolar recordings; Default parameter settings were used in every feature extractor.

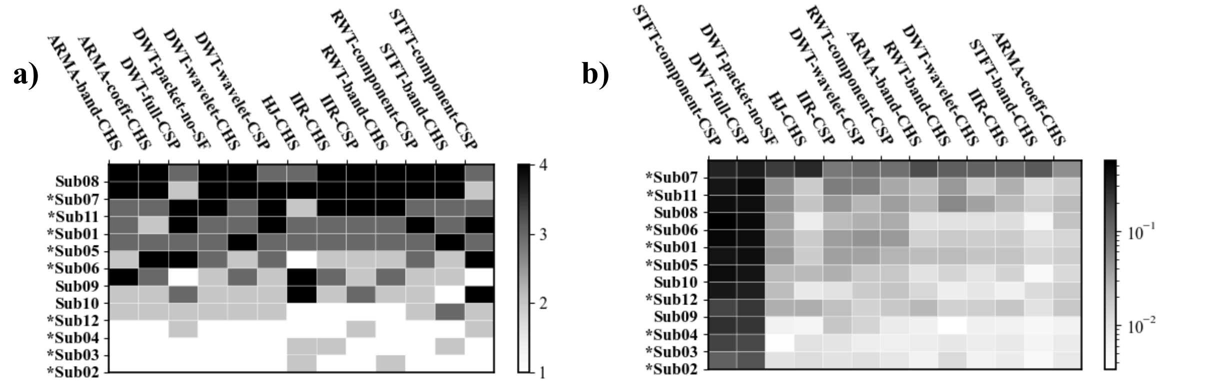


Figure 4.11 - Feature screening results for the laterality selection task using the 'Dual Hand Pinch' dataset (Appendix A.B) and every feature extraction method; **a)** Quartile ranking based on MI feature averages; Each participant MI average were assigned to the 1, 2, 3 and 4th quartile; **b)** Absolute ranking based on MI features averages; (*) indicates participants with available monopolar recordings; Default parameter settings were used in every feature extractor.

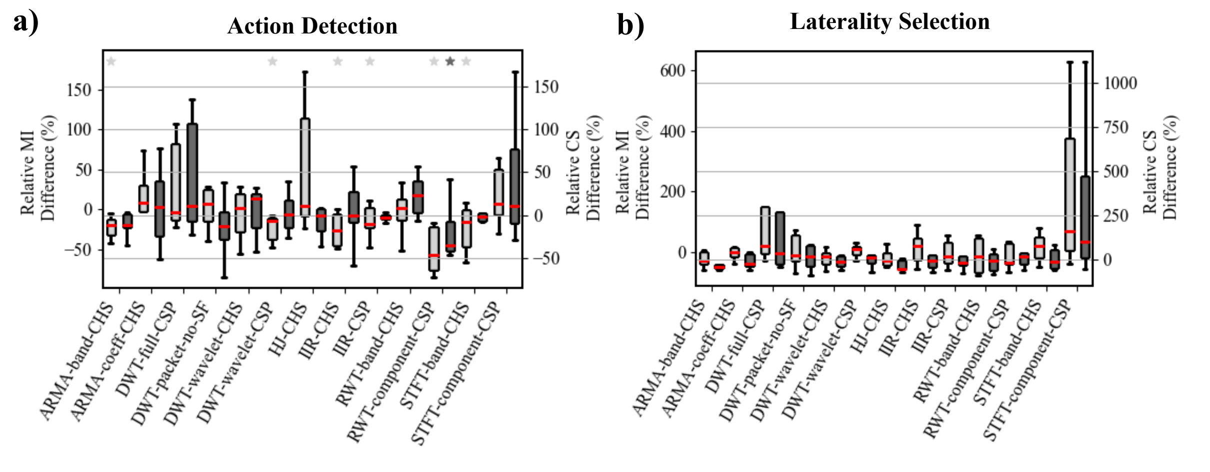


Figure 4.12 - Relative difference in average MI (**light grey**) and CS (**dark grey**) between features extracted with bipolar and that extracted in monopolar settings for **a)** the action detection problem (Appendix A.A) and for **b)** the laterality detection problem (Appendix A.B); (*) indicates a statistically significant degradation of those metrics using a Wilcoxon rank test ($p < 0.05$); Default parameter settings were used in every feature extractor.

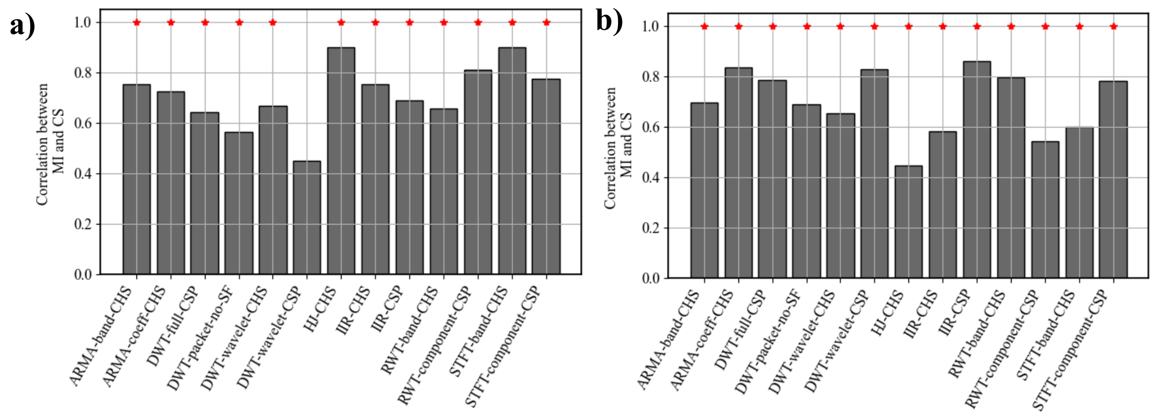


Figure 4.13 – Comparison between MI and CS screening metrics using Spearman ranked correlation; **a)** For the action detection problem (Appendix A.A); **b)** For laterality selection problem (Appendix A.B); (*) indicates a statistically correlation between those metrics using a Wilcoxon rank test ($p < 0.05$); Default parameter settings were used in every feature extractor.

In the less extreme cases, the ERS and ERD were present and especially marked at the action onset but not necessarily sustained throughout.

Generally, participants stayed within the same quartile, irrespectively of the extraction method. Figure 4.10b and Figure 4.11b compiled the broad range of feature quality found in the overall population sample. For example, the most informative feature sets were typically at least one order of magnitude superior, then the lesser informative ones in terms of MI averages. Therefore, the population samples could be characterised as significantly heterogeneous given the variability of the individual features found in each recording. As discussed in Chapter 3.A, this heterogeneity was to be expected. Some variance in average MI was observed when conferring to the extraction method (in their untuned settings). Some extraction methods tended to yield feature sets possessing significantly more MI, in particular the STFT and DWT-full CSP based methods. The translation into actual performance remained to prove, as this could have been the result of CSP overfitting, especially for participants who otherwise exhibited feature sets with low MI. Lastly, some feature metrics could be succinctly altered depending on whether the direct monopolar recording or bipolar digital construction was employed. It was noted that the bipolar differential settings enhanced the β activity, whereas the high-γ activity was attenuated. The relative differences between the monopolar and bipolar average MI and CS were computed for every feature extractor (Figure 4.12). The hypothesis stating the digital bipolar construction may destroy information was evaluated using a Wilcoxon test (p-value < 0.05). Some statistically significant population average drops were found when using the bipolar construction in some action detection cases but none in the laterality selection cases. Therefore, there was no immediate reason to further employ monopolar recordings over bipolar ones. Maintaining the bipolar settings also meant more uniformity over the decoding experiments, as some recordings were only made using the differential recording channels. The average MI and CS relationship was checked using Spearman's ranked correlation (p-value < 0.05). Those two metrics were always tightly correlated (at least above 0.45), which confirmed that they measured similar properties of the feature set (Figure 4.13).

2. BO tuned Action Detection Decoder

Figure 4.14 reports the performance results for the BO tuned action detection decoders. In terms of cross-validation, most performances median GM and TPR laid within the 0.65-0.75 range. Respective median FPR laid complementarily within the 0.20-0.35 range. The best median performance was obtained using the ARMA-band-CHS-LR decoder with a median averaged GM of 0.731.

Figure 4.15, Figure 4.16, Figure 4.17 and Figure 4.18 report various decoder evaluation criteria that assess the impact of BO tuning (Figure 3.4) and the complexity of the tuned solution. These are respectively model overfitting r_o, tuning improvement I, total number of features N_f used for decoding and total analysis window τ_d. Results and interpretations of the statistical analysis of performance and evaluation criteria are presented in Table 4.2.

Figure 4.19 present the total index of the parameters resulting from the sensitivity analysis of the BO surrogate models $\hat{F}$ employed for model tuning. Overall, it was observed that the most important parameters were the threshold T and the dead band parameter r_T. This relevance was to be expected as both parameters control the decoder output directly and thereby the performance criterion. Ill-set thresholds could lead to entirely un-operational decoders. Secondly, classifier parameters were systematically relevant. The regularisation constant for the LR classifier was more important than all other decoder or extractor specific parameters. There were then parameters relevant in more specific cases, such as the DWT lattice parameters with the LDA or n_o with the IIR based extractors. It was noted that the channel selection parameters N_c, N_{c_0}, N_{c_1} and the feature window parameter N_w were not as relevant to decoding performance as initially anticipated for most classifiers, except in the case of the NB. In the latter instance, it was as if the tuning algorithm was trying to compensate for the lack of regularisation tightening by leveraging other parameters.

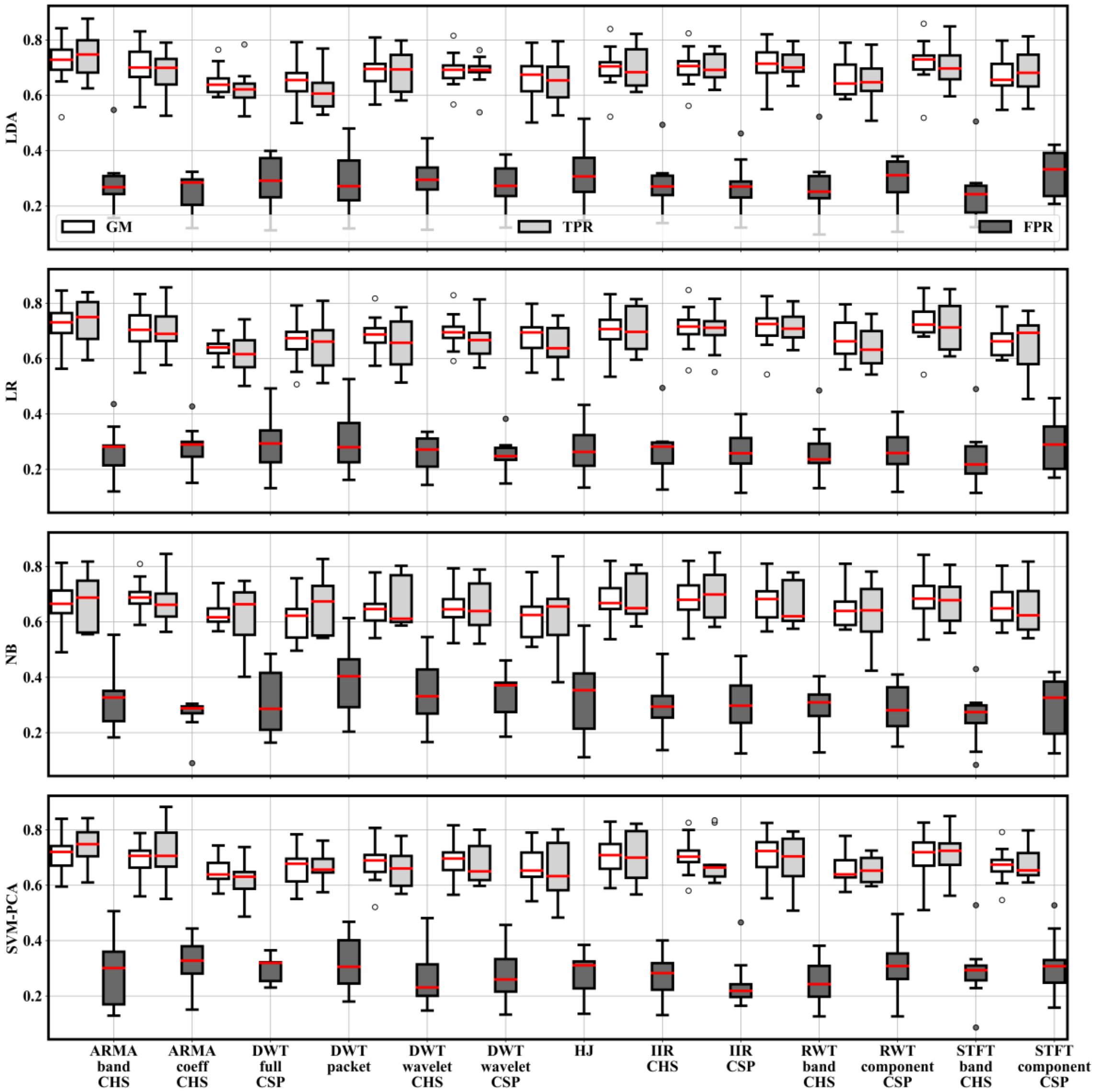


Figure 4.14 – Tuned performance distributions for the action detection task; All participant performances were included in the distribution (Appendix A.A); Post BO tuning cross-validation fold averages (3-46) of true-positive rate TPR, false positive FPR and their geometric mean GM (4-15) are reported; GM was the performance metric optimised by the BO parameter search; All extractor and classifier pairs are reported.

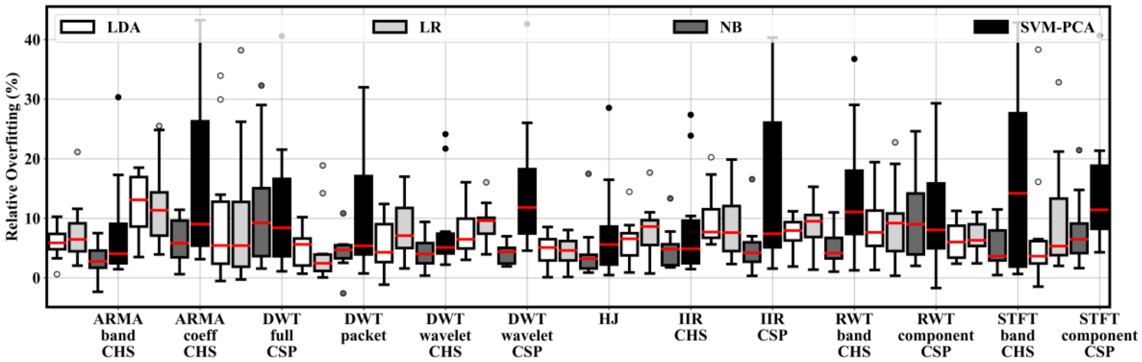


Figure 4.15 – Model overfitting (r_o) distributions for the action detection task; All participant performances were included in the distribution (Appendix A.A); All extractor and classifier pairs are reported; Overfit was computed according to (3-47) after BO decoder tuning and using cross-fold averages training and validation GM.

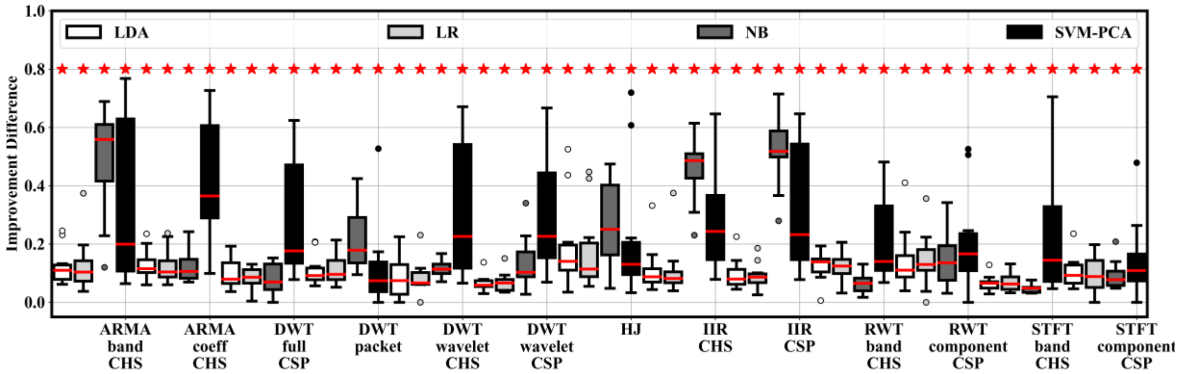


Figure 4.16 – Tuning improvement I distributions for the action detection task; All participant performances were included in the distribution (Appendix A.A); All extractor and classifier pairs are reported; Improvement was computed according to (3-11) using performance (cross-validation GM average) with default parameters ρ_0 and that with optimum parameters ρ^* found through BO; (*) indicates a significant improvement found (Wilcoxon ranked test, p-value < 0.05).

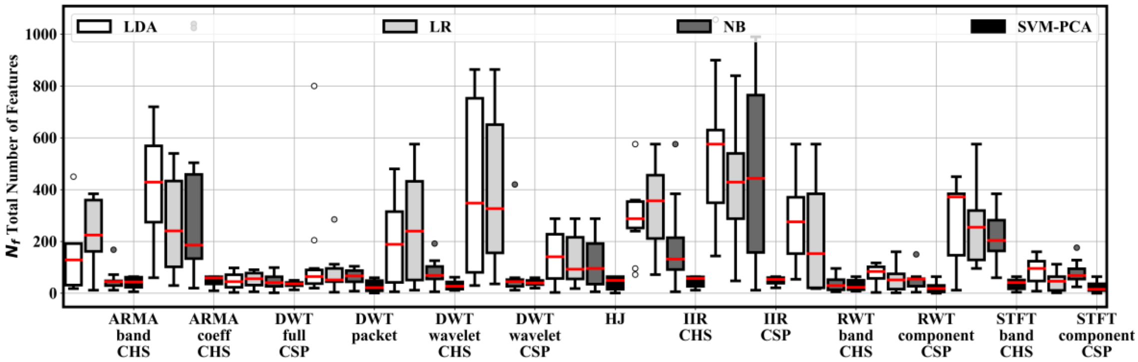


Figure 4.17 – Number of features used for decoding N_f distributions for the action detection task; All participant performances were included in the distribution (Appendix A.A); All extractor and classifier pairs are reported; N_f was computed according to (3-40) after BO decoder tuning using optimum parameters (in particular N_c and N_w).

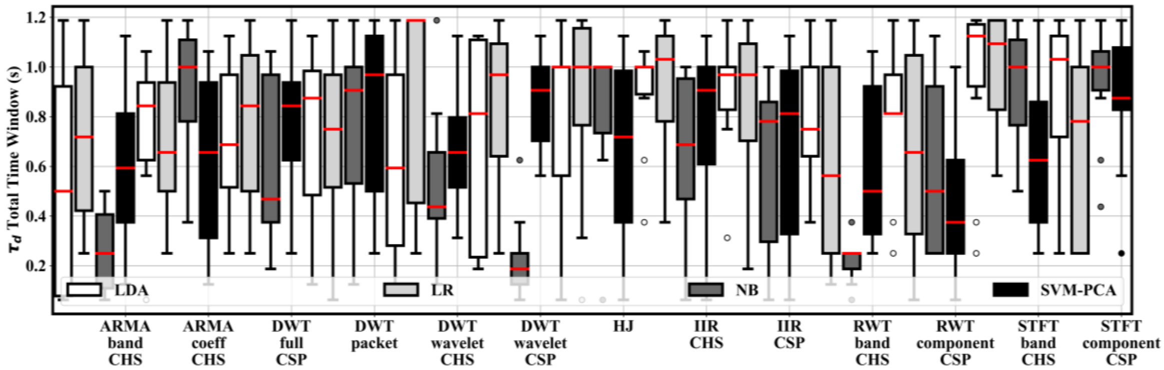


Figure 4.18 –Total analysis window length τ_d distributions for the action detection task; All participant performances were included in the distribution (Appendix A.A); All extractor and classifier pairs are reported; τ_d was computed according to (3-39) after BO decoder tuning using optimum parameters (in particular n_o and N_w).

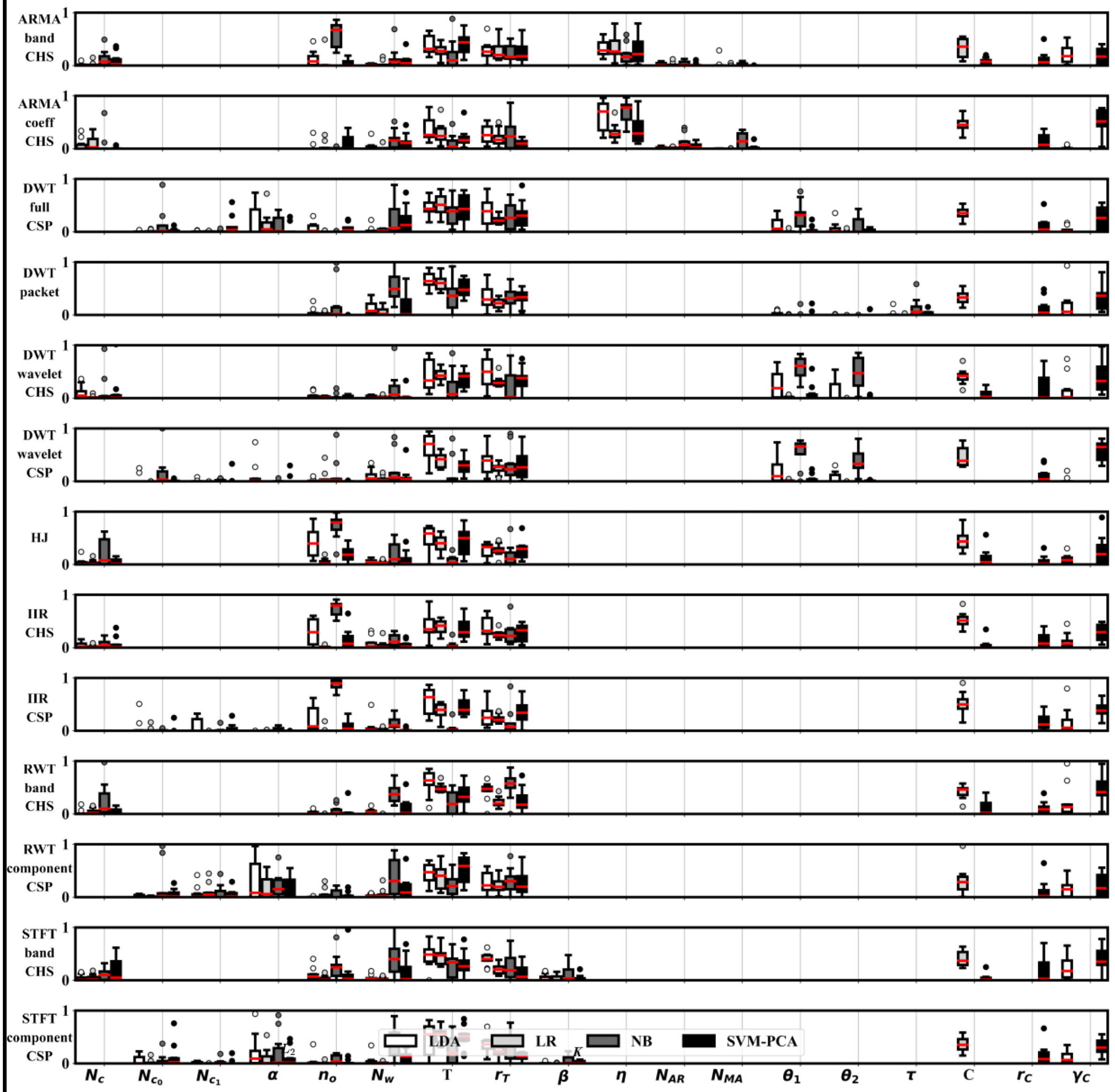


Figure 4.19 –Sensitivity index distributions for the action detection task; All participant performances were included in the distribution (Appendix A.A); All extractor and classifier pairs are reported; The index were obtained from a sensitivity analysis of the BO surrogate models $\hat{F}$ which maps ρ to the GM; Points from the search spaced were sampled using the Saltelli method; Sensitivity index were estimated from the samples using the Sobol method; Total indexes (addition of first order indexes and second order interactions) are reported; Extractors are presented on each row; parameters are organised in following ordered groups: channel and spatial filter parameters ($N_c, N_{c_0} N_{c_1} \alpha_{L_2}$), overlap n_o, feature extraction window N_w, thresholds (T, r_t), extractor specific parameters (STFT β_k , ARMA η, N_{AR}, N_{MA} and DWT Θ_1, Θ_2) and lastly estimator parameters (C, r_c, γ_c); Indexes leaning towards 1 indicate significant importance to the tuning problem whereas those leaning towards 0 have lesser relevance.

Table 4.2 – Statistical analysis of the different decoder evaluation criteria collected after BO tuning for the action detection task; Criteria include performance, overfitting, tuning improvement, total number of features and analysis window; Selection influence of extractor and estimator on criteria was evaluated using first an ANOVA then post-hoc Tukey pairwise tests; Significant Tukey test outcomes (<) indicate that the criteria was more important for one compared item to the other; (=) indicates that no significant difference occurred; Tukey test have been selectively reported into sub-groups of interests and to make comparisons across specific methods; All not reported test had resulted in no significant differences; An interpretation of the results is also provided; (IQR) stands for inter-quartile range, (ANOVA) for analysis of variance.

Metric	Two-Way ANOVA Test			Tukey Pairwise Test				Interpretation
	Exogenous Variable	p-value	Reject Null Hypothesis	Sub-Group	Reported and Compared Item			
Performance (GM)	Estimators	< 0.01	Yes	All Estimators	NB	<	LR	Performance depended on the type of estimator. There was no significant performance difference between the LR, LDA and SVM-PCA classifiers, except the NB classifier, which comparatively underperformed.
					NB	<	LDA	
					NB	<	SVM-PCA	
	Extractors	< 0.01	Yes	CHS	HJ-CHS	<	STFT-band-CHS	Performance depended on the type of extractor. For CHS methods, the performance gap was only significant between the STFT-band and HJ extractors. Top performant methods ARMA and STFT-band-CHS performed equally well. There was no significant performance difference between the two IIR or ARMA method variants; a significant decrease in performance when using the CSP component approach with both the RWT and STFT instead of using band definition and CHS; no significant performance difference between DWT approaches. However, DWT-full-CSP and DWT-packet variants underperformed relative to the best-performing methods such as the ARMA and STFT-band-CHS.
				IIR	IIR-CHS	=	IIR-CSP	
				ARMA	ARMA-band-CHS	=	ARMA-coeff-CHS	
				STFT	STFT-component-CSP	<	STFT-band-CHS	
				RWT	RWT-component-CSP	<	RWT-band-CHS	
				DWT	DWT-full-CSP	=	DWT-wavelet-CHS	
					DWT-packet	=	DWT-wavelet-CHS	
					DWT-wavelet-CSP	=	DWT-wavelet-CHS	
				ARMA vs STFT	ARMA-band-CHS	=	STFT-band-CHS	
				DWT vs STFT	DWT-packet	<	STFT-band-CHS	
					DWT-wavelet-CSP	<	ARMA-band-CHS	
				DWT vs ARMA	DWT-packet	<	ARMA-band-CHS	
					DWT-wavelet-CSP	<	ARMA-band-CHS	
Overfitting (r_o)	Estimators	< 0.01	Yes	All Estimators	NB	<	SVM-PCA	Overfit depended on the type of estimator. The SVM-PCA estimator significantly overfitted the data than most other estimators; the NB classifier overfitted the less even compared to other classifiers such as the LR. More complex estimators are thus more likely to overfit the data.
					LR	<	SVM-PCA	
					LDA	<	SVM-PCA	
					NB	<	LR	
	Extractors	< 0.01	Yes	HS vs other	HJ	<	ARMA-coeff-CHS	Certain estimators could influence overfitting. In particular, DWT-full-CSP and ARMA overfitted the data more than the HJ. Overfitting was not, however, excessively worse with the CSP-based method as initially posited.
					HJ	<	DWT-full-CHS	

Metric	Two Way ANOVA Test			Tukey Pairwise Test				Interpretation
	Exogenous Variable	p-value	Reject Null Hypothesis	Sub-Group	Reported and Compared Item			
Improvement (I)	Estimators	< 0.01	Yes	All Estimators	LR	<	SVM-PCA	Tuning improvement depended on the type of estimator. LDA and LR experienced a comparable low level of improvement, whereas NB and SVM-PCA required a high level of tuning. The SVM-PCA median improvement was more widely spread, all laying in the 0.5-0.35 and all bearing large IQR (up to 0.5), and default parameters lead to poor initial performance. Modifying extractor parameters could also significantly improve NB performance.
					LDA	<	SVM-PCA	
					LR	<	NB	
					LDA	<	NB	
	Extractors	< 0.01	Yes	ARMA vs others	DWT-full-CSP	<	ARMA-band-CHS	Tuning improvement depended on the type of extractor. ARMA and IIR-based methods required much tuning; conversely, tuning of STFT and RWT extractors did not significantly improve model performance. Default parameters could also be close to optimum. Although the DWT extractors were the most parameterised, extractor tuning had a modest impact on performance improvement.
					DWT-packet	<	ARMA-band-CHS	
					DWT-wavelet-CHS	<	ARMA-band-CHS	
					DWT-wavelet-CSP	<	ARMA-band-CHS	
					RWT-band-CHS	<	ARMA-band-CHS	
					STFT-band-CHS	<	ARMA-band-CHS	
					STFT-component-CSP	<	ARMA-band-CHS	
				IIR vs other	STFT-band-CHS	<	IIR-CHS	
					STFT-component-CSP	<	IIR-CHS	
					STFT-band-CHS	<	IIR-CSP	
					STFT-component-CSP	<	IIR-CSP	
					RWT-band -CHS	<	IIR-CSP	

Metric	Two Way ANOVA Test			Tukey Pairwise Test				Interpretation
	Exogenous Variable	p-value	Reject Null Hypothesis	Sub-Group	Reported and Compared Item			
Number of Features (N_f)	Estimators (except SVM-PCA)	< 0.01	Yes	NB, LDA, LR	LR	=	LDA	The number of selected features depended on the type of estimator. LR and LDA classifiers selected significantly more features than the NB classifier. The LR and LDA recruited many channels or components for decoding (> 60% typically). In the LR case, features could be selected from this large space through L_1-norm regularisation. For the NB, limiting the number of channels may have helped regularisation.
					NB	<	LDA	
					NB	<	LDA	
	Extractors	< 0.01	Yes	STFT	STFT-component-CSP	<	STFT-band-CHS	The number of features selected depended on the type of feature extractors used, particularly the channel management strategy used. The STFT-component approach significantly reduced the number of features used relative to the STFT-band approach, but this was not the case with the RWT extractor. Because of the variable adjustment of N_{AR} and N_{MA} parameters, more coefficient features were used with the ARMA extractor. IIR-CSP and DWT-wavelet-CSP used significantly more features than their respective variants since every component selected added a fixed number of bands or wavelet coefficients (with a maximum of $2N_{ch}$ components). The DWT-wavelet-CHS remained relatively lean in comparison to the DWT-full-CSP and -packet. Overall, HJ, DWT methods, and full spectra approach CSP (DWT-full-CSP, STFT and RWT-component-CSP) helped reduce the feature space size.
				RWT	RWT-component-CSP	=	RWT-band-CHS	
				ARMA	ARMA-band-CHS	<	ARMA-coeff-CHS	
				IIR	IIR-CHS	<	IIR-CSP	
				DWT	DWT-packet	<	DWT-wavelet-CPS	
					DWT-full-CSP	<	DWT-wavelet-CPS	
					DWT-packet	=	DWT-wavelet-CHS	
					DWT-full-CSP	=	DWT-wavelet-CHS	

Metric	Two Way ANOVA Test			Tukey Pairwise Test				Interpretation
	Exogenous Variable	p-value	Reject Null Hypothesis	Sub-Group	Reported and Compared Item			
Analysis Window (τ_d)	Estimators	< 0.01	Yes	All Estimators	NB	<	LR	The estimator type did affect the length of the analysis window. LR and LDA classifiers selected significantly longer than average windows, whereas NB selected smaller ones. Window size could have acted as a regularisation parameter in the latter case.
					NB	<	LDA	
	Extractors	< 0.01	Yes	ARMA vs STFT	ARMA-band-CHS	<	STFT-band-CHS	Analysis window length depended on the type of extractor chosen. ARMA and RWT methods had the shortest windows (< 0.6s typically), whereas STFT had the significantly longer ones (> 0.8s typically). Other methods had median windows between 0.6 and 0.9s with mixed extraction (controlled by n_o) and feature window sizes (controlled by N_w). In particular, IIR, RWT, HJ, and ARMA-based extractors typically preferred very short extraction windows, whereas DWT and STFT had longer ones. The STFT systematically selected long extraction windows (0.25s) for higher spectral resolution and temporal smoothing.
					ARMA-band-CHS	<	STFT-component-CSP	
				RWT vs STFT	RWT-band-CHS	<	STFT-band-CHS	
					RWT-band-CHS	<	STFT-component-CSP	

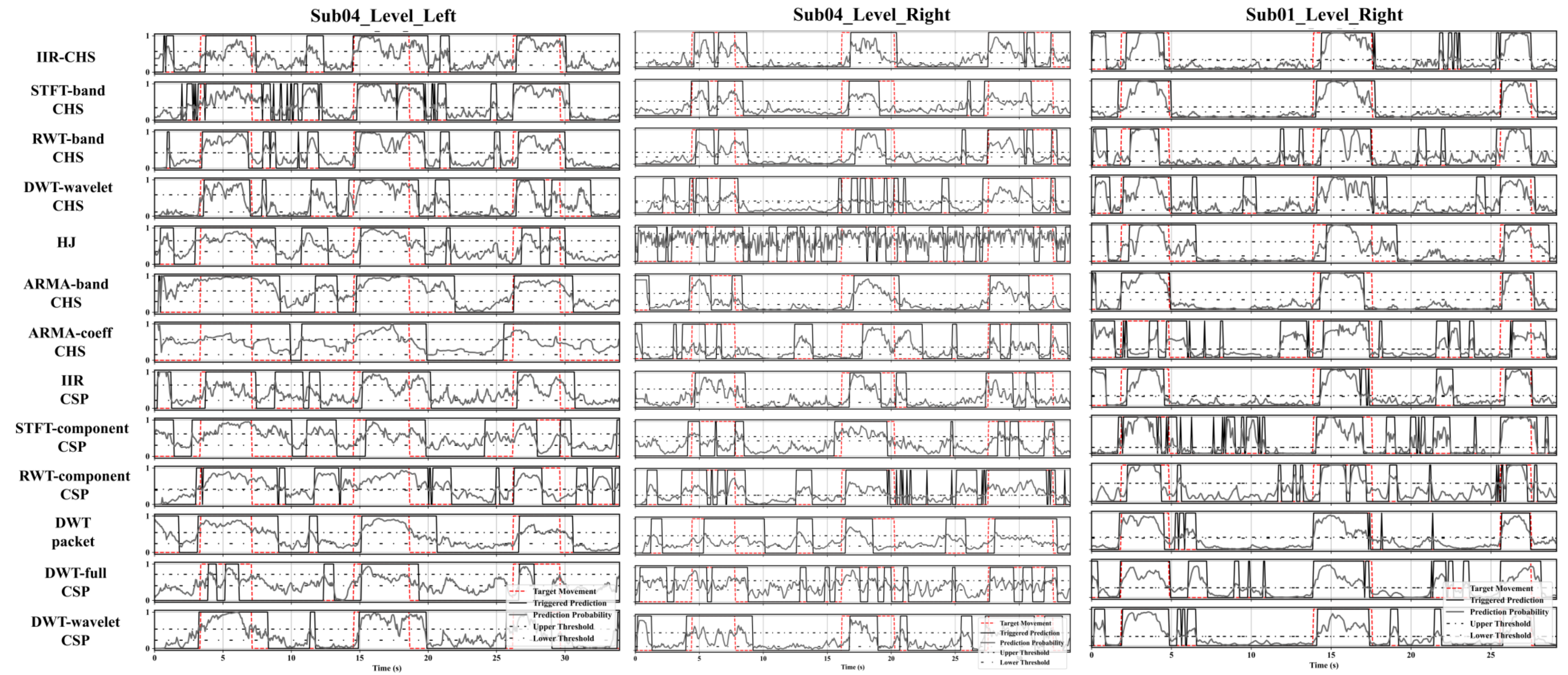


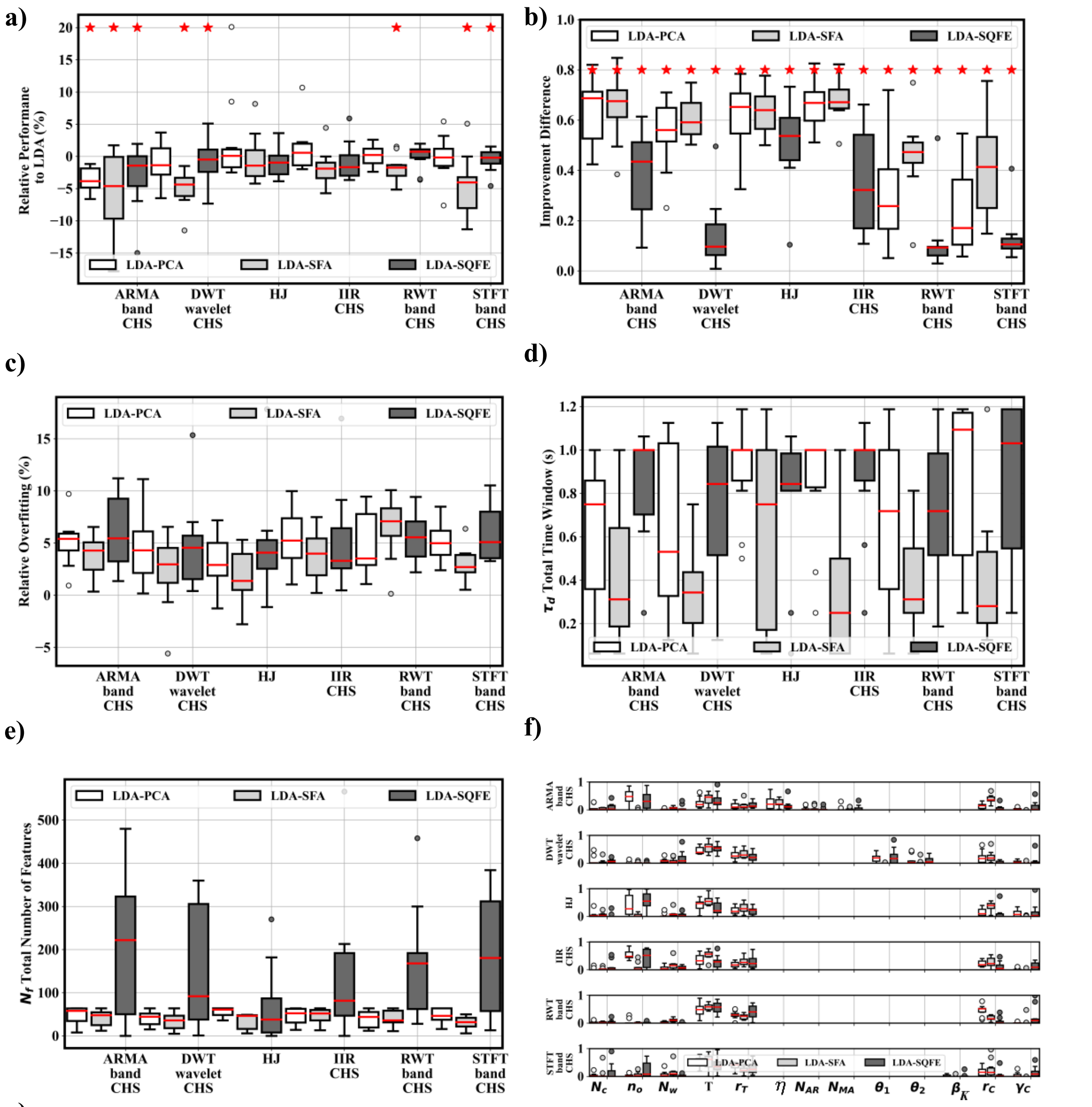
Figure 4.20 - Example of decoded trace for action detection for three participants (Sub04_Level_Left, Sub04_Level_Rigth and Sub01_Level_Rigth, Appendix A.A) using the LR classifier combined with all extractors; The average prediction of the decoder fitted during cross-validation was employed to make prediction on a set aside test data segment; The (**Red**) red dotted line is the action target; The (**Black)** solid line the prediction; The (**Grey**) solid line the classifier probability; The (**Dashed Black**) lines the trigger thresholds.

Figure 4.20 shows different decoded test segments for different extractors and the LR classifier. The settings trade-off for the double-threshold system is there apparent. A higher trigger threshold prevented a false transition caused by spurious activity but at the same time could delay the transition, especially if the extractor-classifier system was too slow. A lower reset threshold, below spurious activity level, enabled better-aligned detection timing with the action. A larger dead band was also effective at preventing state transition in the case of a sudden drop of the classifier prediction while the action was sustained. The threshold settings were also dependent on the type of probability range generated by the classifier, and a smaller certainty range permitted a smaller dead-band for ill-set transition rejection.

3. Dimensionality Reduction Methods

Feature dimensionality reduction technics were investigated for the LDA classifier specifically. L1-norm regularisation already served to induce sparsity in the regression coefficients and could allow for post-fitting pruning in the LR classifier. The use of the SVM algorithm for detection could not be further justified given that no significant performance gains were obtained with the RBF kernel, and PCA reduction was already implemented. Lastly, the NB did not match the performance of the other classifiers. Three different LDA and dimensionality reduction tandems were optimised again using BO: LDA-PCA, LDA-SFA and LDA-SQFE. The CS criterion employed in the SQFE method corresponded the most naturally to the LDA fitting problem (Fukunaga, 1990). Only CHS methods were employed in the feature extractor as it was concluded that the CSP methods and the DWT-packet were already acting as effective feature selectors.

Figure 4.21 a reports the performance of those new decoders relative to that previously obtained for the LDA decoder. It was first observed that no overall performance gains were achieved, apart from some anecdotal cases. Median performance losses laid between 0 and 5%. The loss was only significant in a few cases, particularly associated with the ARMA and STFT-band-CHS extractors (Wilcoxon rank test, p-value < 0.05). The typical improvement gap (Figure 4.21b) between the performance with default parameters and that with optimal parameters widened. Like in the case of the SVM-PCA, BO played a more significant role in adjusting the parameters when starting with a less informed set of default settings. Improvement depended on the dimensionality reduction methods (ANOVA, p-value < 0.05). SQFE, in particular, required significantly less tuning than PCA or SFA methods (Tukey pairwise test, p-value < 0.05). This result provided further evidence that optimisation helped set the number of selected parameters. Overfitting was maintained low (Figure 4.21c) and with no significant variance among decoders. Median sat below 7.5% in all cases which was comparable to what was previously obtained. The coupling between different parameters was as well surprising. Figure 4.21g, e and d present the distributions for some relevant parameters, the total number of features used for decoding and the total analysis window. The feature epoch window length N_w seemed for the most part, unaffected, apart from in the case of the LDA-SFA, where those were significantly reduced (Tukey



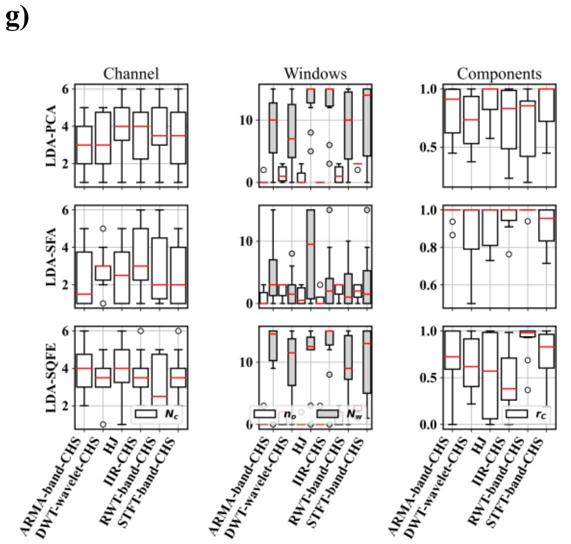


Figure 4.21 – Dimensionality reduction impact on action detection; All participant performances were included in the distribution (Appendix A.A); All CHS extractors were paired with LDA classifiers; **a)** Performance of LDA coupled with dimensionality reduction relative to with no dimensionality reduction scheme after BO tuning; (*) indicates that relative performance is significantly reduced (Wilcoxon paired rank test p-value < 0.05); **b)** Tuning improvement I between default parameter ρ_o and optimum parameter ρ^* found through BO; (*) indicates that tuning significantly increased performance (Wilcoxon paired rank test p-value < 0.05; **c)** Model overfit r_o computed after BO decoder tuning and using cross-fold averages training and validation GM; **d)** Decoder total analysis window length τ_d and **e)** Total number of features N_f employed for decoding after feature selection and BO tuning; **f)** Sensitivity index resulting from a sensitivity analysis of the BO surrogate models $\hat{F}$ for every tuned parameter; **g)** Relevant parameter distributions from ρ^* obtained after BO tuning.

pairwise test p-value < 0.05). In that sense, the SFA stage made the feature window redundant. The LDA-SFA algorithm preferred to rely on extremely short windows recruit fewer decoding channels but include the entirety of its components (Figure 4.21g). The PCA and the SQFE preferred constructing components and sample features from a larger channel space and longer time windows. The PCA and SFA approach required at least 40 features in every case in half of the sampled population, while SQFE used far more features overall. Importance of each tuning parameters through sensitivity analysis of the BO surrogate models is presented in Figure 4.21f. The compression ratio r_c appeared to hold more importance to tuning when using the PCA or SFA than when using the SQFE. Other parameter held similar levels of importance than what had been previously observed without any form of dimensionality reduction.

Finally, the SFA tended to let through more noisy components than other methods, resulting in unreliable decoder outputs (Figure 4.22).

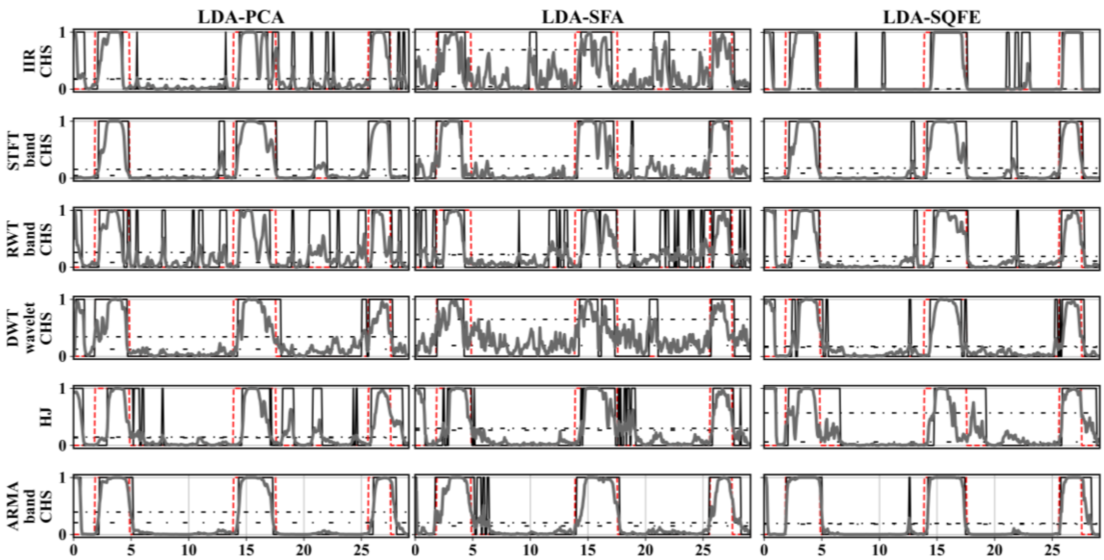


Figure 4.22 - Decoder trajectory examples with Sub1_Level_Right (Appendix A.A) and the different LDA decoding schemes; The (**Red**) dotted lines is the target signal; The (**Black**) solid line the decoder output; The (**Grey**) solid line the classifier probability output; The (**Dashed Black**) lines the trigger thresholds.

4. BO tuned Laterality Selection Decoder

Cross-validation accuracy performance for laterality selection is presented in Figure 4.23. Maximum performances obtained were superior to 0.85 and the lowest below 0.5. The performances were heterogeneous with large IQRs distributions. Most median performances remained low, typically between 0.6 and 0.7. Performance depended again on both the choice of extractor and classifier (AVONA, p-value < 0.05).

Figure 4.24, Figure 4.25, Figure 4.26 and Figure 4.27 report various decoder evaluation criteria that assess the impact of BO tuning (Figure 3.4) and the complexity of the tuned solution. These are again respectively model overfitting r_o, tuning improvement I, total number of features N_f used for

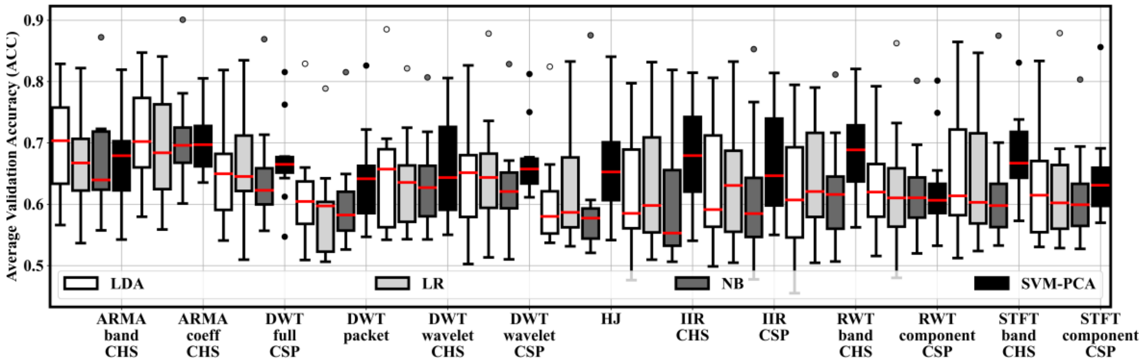


Figure 4.23 - Tuned performance distributions for the laterality selection task; All participant performances were included in the distribution (Appendix A.B); Post BO tuning cross-validation fold average (3-46) accuracy ACC is report and was the performance metric optimised by the BO parameter search; All extractor and classifier pairs are reported.

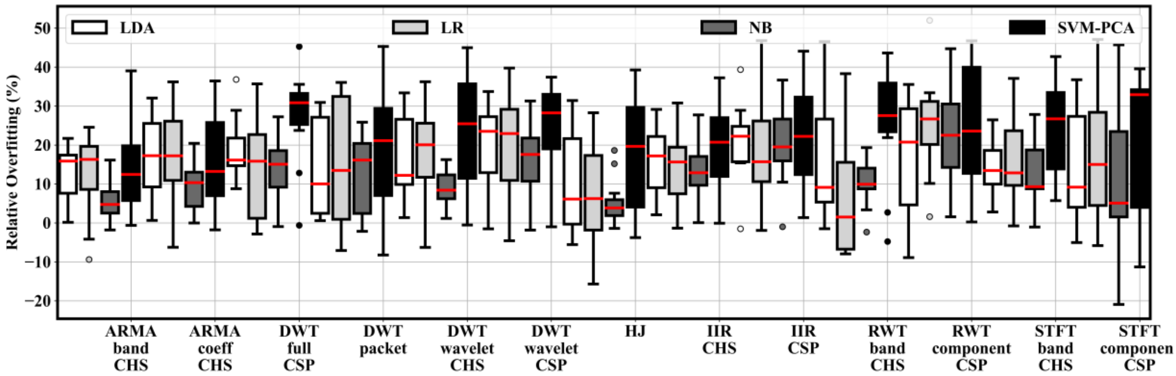


Figure 4.24 - Model overfitting (r_o) distributions for the laterality selection task; All participant performances were included in the distribution (Appendix A.B); All extractor and classifier pairs are reported; Overfit was computed according to (3-47) after BO decoder tuning and using cross-fold averages training and validation ACC.

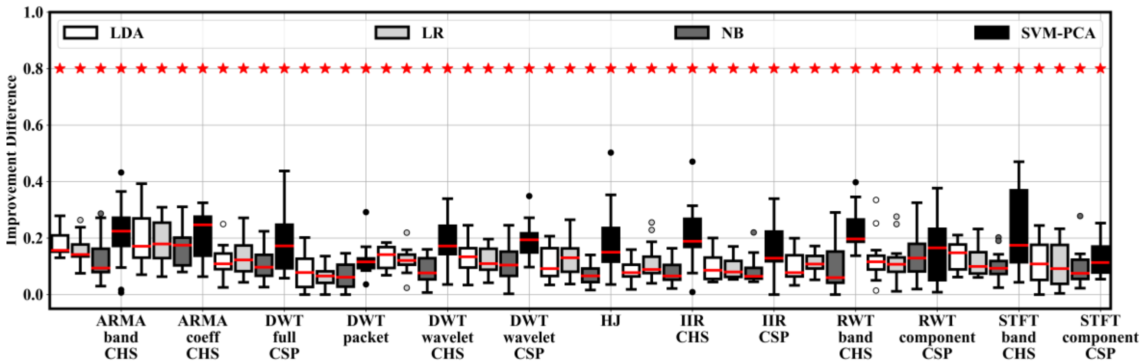


Figure 4.25 - Tuning improvement I distributions for the laterality selection task; All participant performances were included in the distribution (Appendix A.B); All extractor and classifier pairs are reported; Improvement was computed according to (3-11) using performance (cross-validation ACC average) with default parameters ρ_0 and that with optimum parameters ρ^* found through BO; (*) indicates a significant improvement found (Wilcoxon ranked test, p-value < 0.05).

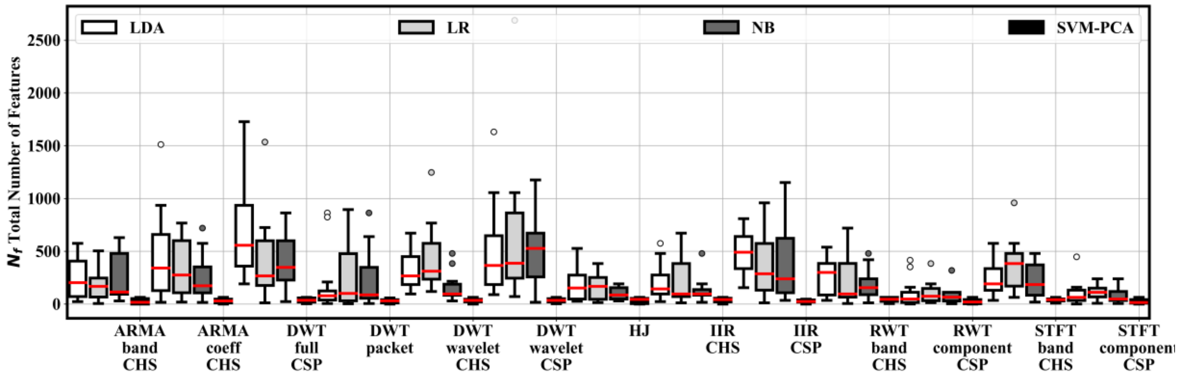


Figure 4.26 - Number of features used for decoding N_f distributions for the laterality selection task; All participant performances were included in the distribution (Appendix A.B); All extractor and classifier pairs are reported; N_f was computed according to (3-40) after BO decoder tuning using optimum parameters (in particular N_c and N_w).

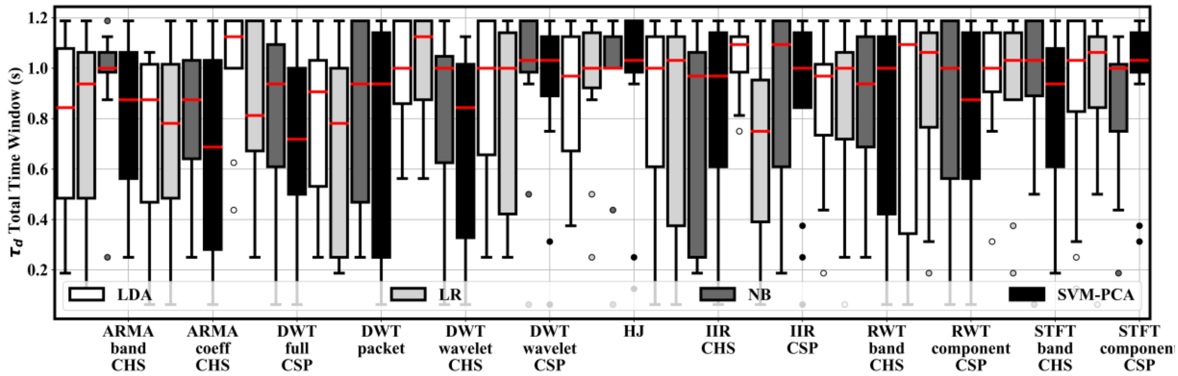


Figure 4.27 - Total analysis window length τ_d distributions for the laterality selection task; All participant performances were included in the distribution (Appendix A.B); All extractor and classifier pairs are reported; τ_d was computed according to (3-39) after BO decoder tuning using optimum parameters (in particular n_o and N_w).

decoding and total analysis window τ_d. Results and interpretations of the statistical analysis of performance and evaluation criteria are presented in Table 4.3.

Sensitivity analysis of the BO surrogate models was again carried (Figure 4.28). The parameter relevance distribution varied from the action decoding task. The number of selected N_c channels and, to a lesser extent, the number of selected components N_{c_o}, N_{c_1} were more relevant. This accrued relevance also holds for the number of overlaps n_o and N_w. All extractor parameters, AMRA decay rate η and term parameters, lattice parameters Θ_1, Θ_2, and STFT window shape factor β_K were significantly more important in all cases. Lastly, classifier parameters upheld their importance.

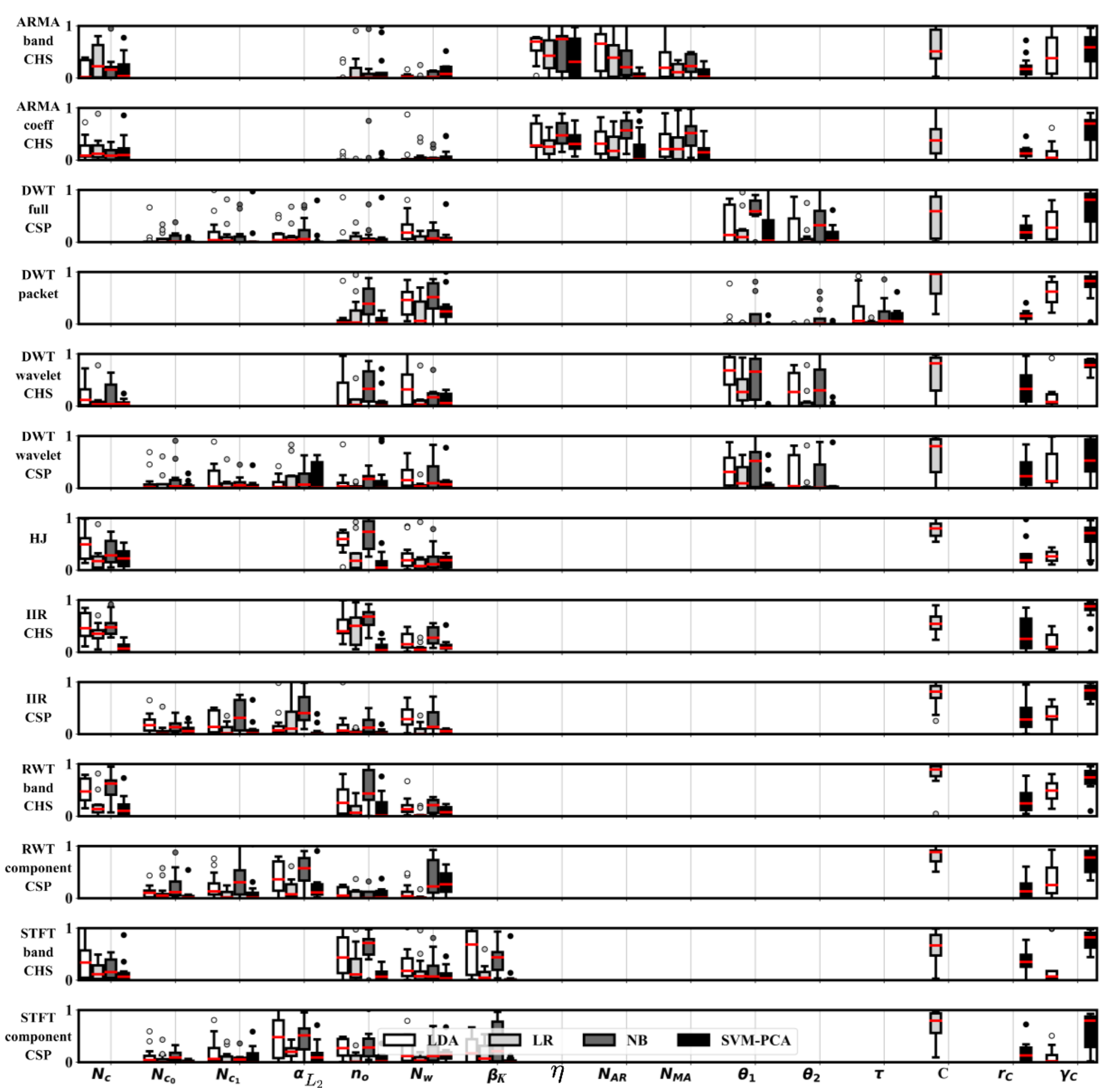


Figure 4.28 - Sensitivity index distributions for the laterality selection task; All participant performances were included in the distribution (Appendix A.B); All extractor and classifier pairs are reported; The index were obtained from a sensitivity analysis of the BO surrogate models $\hat{F}$ which maps ρ to the ACC; Points from the search spaced were sampled using the Saltelli method; Sensitivity index were estimated from the samples using the Sobol method; Total indexes (addition of first order indexes and second order interactions) are reported; Extractors are presented on each row; parameters are organised in following ordered groups: channel and spatial filter parameters ($N_c, N_{c_0} N_{c_1} \alpha_{L_2}$), overlap n_o, feature extraction window N_w, extractor specific parameters (STFT β_k , ARMA η, N_{AR}, N_{MA} and DWT Θ_1, Θ_2) and lastly estimator parameters (C, r_c, γ_c); Indexes leaning towards 1 indicate significant importance to the tuning problem whereas those leaning towards 0 have lesser relevance.

Table 4.3 – Statistical analysis of the different decoder evaluation criteria collected after BO tuning for the laterality selection task; Criteria include performance, overfitting, tuning improvement, total number of features and analysis window; Selection influence of extractor and estimator on criteria was evaluated using first an ANOVA then post-hoc Tukey pairwise tests; Significant Tukey test outcomes (<) indicate that the criteria was more important for one compared item to the other; (=) indicates that no significant difference occurred; Tukey test have been selectively reported into sub-groups of interests and to make comparisons across specific methods; All not reported test had resulted in no significant differences; An interpretation of the results is also provided.

Metric	Two Way ANOVA Test			Tukey Pairwise Test				Interpretation
	Exogenous Variable	p-value	Reject Null Hypothesis	Sub-Group	Reported and Compared Item			
Performance (ACC)	Estimators	< 0.01	Yes	All Estimators	NB	<	SVM-PCA	Performance depended on the type of estimator. The SVM-PCA here outperformed both LR and NB but not the LDA, and it helped deliver better performances than other classifiers with most extractor pairings.
					LR	<	SVM-PCA	
					LDA	=	SVM-PCA	
	Extractors	< 0.01	Yes	ARMA-coeff-CHS vs rest	ARMA-band-CHS	=	ARMA-coeff-CHS	Performance depended on the type of extractor. ARMA extractors, in particular, the ARMA-coeff-CHS, significantly surpassed most other extractor methods, aside from a few exceptions, such as the DWT-full-CSP and the DWT-wavelet-CSP. These two methods performed relatively well compared to others and their previous performance in the action detection task.
					STFT-band-CHS	<	ARMA-coeff-CHS	
					STFT-component-CSP	<	ARMA-coeff-CHS	
					RWT-band-CHS	<	ARMA-coeff-CHS	
					RWT-component-CSP	<	ARMA-coeff-CHS	
					IIR-CHS	<	ARMA-coeff-CHS	
					IIR-CSP	<	ARMA-coeff-CHS	
					DWT-wavelet-CHS	<	ARMA-coeff-CHS	
					DWT-wavelet-CSP	=	ARMA-coeff-CHS	
					DWT-full-CSP	=	ARMA-coeff-CHS	
					DWT-packet	<	ARMA-coeff-CHS	
					HJ	<	ARMA-coeff-CHS	
Overfitting (r_o)	Estimators	< 0.01	Yes	All Estimators	NB	<	SVM-PCA	Overfit depended on the choice of estimator. The SVM-PCA estimator significantly overfitted the data more than most other estimators. Nonetheless, it can be argued that the non-linearity was put to better use given the substantial performance gains, as other estimators tended to overfit the data than in the action detection task
					LR	<	SVM-PCA	
					LDA	<	SVM-PCA	
	Extractors	< 0.01	Yes	HJ vs other	HJ	<	RWT-component-CSP	Although overfit depended on the extractor type, variance across extractors was overall limited; Only certain CSP extractors significantly overfitted more the data than the HJ and ARMA-band-CHS methods. ARMA-coeff-CHS, in particular, did not overfit the data excessively as it previously did in the action detection task.
					HJ	<	IIR-CSP	
					HJ	<	DWT-wavelet-CSP	
				ARMA vs other	ARMA-band-CHS	<	RWT-component-CSP	
					ARMA-band-CHS	<	IIR-CSP	
					ARMA-band-CHS	<	DWT-wavelet-CSP	

Metric	Two Way ANOVA Test			Tukey Pairwise Test				Interpretation
	Exogenous Variable	p-value	Reject Null Hypothesis	Sub-Group	Reported and Compared Item			
Improvement (I)	Estimators	< 0.01	Yes	All Estimators	NB	<	SVM-PCA	Tuning improvement depended on the type of estimator. The SVM-PCA benefitted more significantly from parameter tuning than any other estimator, and tuning had a limited impact on these other estimators.
					LR	<	SVM-PCA	
					LDA	<	SVM-PCA	
	Extractors	< 0.01	Yes	ARMA-coeff-CHS vs rest	ARMA-band-CHS	=	ARMA-coeff-CHS	Tuning improvement depended on the choice of estimator. The ARMA-coeff-CHS, in particular, benefited the most from parameter tuning. Tuning is thus partly responsible for the good performance of this extractor.
					STFT-band-CHS	=	ARMA-coeff-CHS	
					STFT-component-CSP	<	ARMA-coeff-CHS	
					RWT-band-CHS	<	ARMA-coeff-CHS	
					RWT-component-CSP	=	ARMA-coeff-CHS	
					IIR-CHS	<	ARMA-coeff-CHS	
					IIR-CSP	<	ARMA-coeff-CHS	
					DWT-wavelet-CHS	<	ARMA-coeff-CHS	
					DWT-wavelet-CSP	=	ARMA-coeff-CHS	
					DWT-full-CSP	<	ARMA-coeff-CHS	
					DWT-packet	<	ARMA-coeff-CHS	
					HJ	<	ARMA-coeff-CHS	

Metric	Two Way ANOVA Test			Tukey Pairwise Test				Interpretation
	Exogenous Variable	p-value	Reject Null Hypothesis	Sub-Group	Reported and Compared Item			
Number of Features (N_f)	Estimators	> 0.01	No					The number of features used for decoding did not depend on the type of estimator (excluding the SVM-PCA).
	Extractors	< 0.01	Yes	DWT vs other	STFT-band-CHS	<	DWT-full-CSP	The number of decoding features depended on the extractor type, and STFT and RWT-component-CSP remained very effective at compressing the feature space. As with the action detection problem, the DWT-wavelet-CSP used significantly more features than other methods. The DWT-full-CSP also required a similarly large amount of features, noticeably more than for action detection. This increase in the number of features could partly explain the good performance of the DWT-full-CSP in the laterality discrimination problem.
					STFT-component-CSP	<	DWT-full-CSP	
					RWT-band-CHS	<	DWT-full-CSP	
					RWT-component-CSP	<	DWT-full-CSP	
					IIR-CHS	<	DWT-full-CSP	
					DWT-packet	<	DWT-full-CSP	
					HJ	<	DWT-full-CSP	
					STFT-component-CSP	<	DWT-wavelet-CSP	
					RWT-band-CHS	<	DWT-wavelet-CSP	
					RWT-component-CSP	<	DWT-wavelet-CSP	
					IIR-CHS	<	DWT-wavelet-CSP	
					DWT-packet	<	DWT-wavelet-CSP	
					HJ	<	DWT-wavelet-CSP	
				ARMA vs other	STFT-component-CSP	<	ARMA-coeff-CHS	
					RWT-component-CSP	<	ARMA-coeff-CHS	
Analysis Window (τ_d)	Estimators	> 0.05	No					The analysis window size was independent of the type of estimator or extractor. It could be argued that long analysis windows are not required for this decoding problem.
	Extractors	> 0.05	No					

5. Best Model and Screen Metric Relation to Performance

Figure 4.29 presents the best decoders for each participant recording and each task. ARMA-band-CHS, STFT-band-CHS and some IIR types were the preferred extractors for action detection, and ARMA-coeff delivered the higher performance in the laterality selection. The type of classifier varied significantly for each participant. Participants' performance followed a similar ranking as in the pre-screen for the action detection task. This similarity was not as true in the laterality selection case. In both cases selecting individual decoder models significantly increased the group median performance. Figure 4.30 and Figure 4.31, reports the relationship between the total feature metrics (average feature MI and CS) obtained in the pre-screening of the recordings with the final tuned performance estimated using Spearman ranking correlation test (p-value < 0.05). In the action detection task, a meaningful relationship was found for almost every case, with least at 0.4 for MI and 0.25 for CS. Strong relationships were also found in the laterality selection case for the best performant extractors. The lack of a significant relationship between the DWT-full-CSP and STFT-component-CSP extractor would confirm those extractors tended distort the relationship between the signals and the motor states. It was expected that the initial quality of the features could be a strong predictor of the end-performance. This dependence on the initial feature separability suggested that the impact of tuning could only be limited and that individual performance ceilings existed. It could be argued that other factors (extractor, classifier choice, optimum parameterisation) had a more limited impact on performance gain.

6. Case Study: Decoder Tuning using Gradient Backpropagation

Results for the backpropagation decoder tuning can be found in more details in Martineau et al. (2020). Decoders were trained for only action-detection and action-detection combined with laterality selection. In summary, it was observed that backpropagation could be employed to tune a simple, multi-stage decoder. Investigation of the weights (Figure 4.32e) of the feature epoch stage used to generate the probabilities p to the targe state y, showed that the expected mapping between the frequency band activity and the bilateral states were correctly formed. Positive γ and negative β were assigned to the action initiation and holding, while dissymmetry in β activity corresponded to laterality changes. Combining action detection and laterality selection caused a reduction in performance. This drop was not linked to excessive confusion between the left and right states (Figure 4.32d), but an overall deterioration in detection ability (Figure 4.32g) and lesser stable state transitions. Moreover, the backpropagation tuning of the spatial filter and the taper appeared to have helped increase performance. The spatial filter configuration departed significantly from its initialisation state. Given the size of the taper weight space relative to that of the dataset, the effectiveness of the backpropagation optimisation was more questionable.

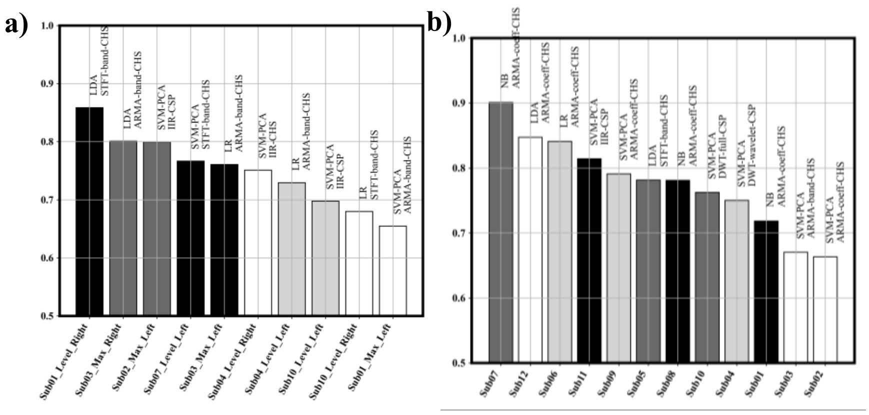


Figure 4.29 – Best decoders **a)** for detection for each participant and **b)** for laterality selection ; The extractor and classifier combination yielding the best tuned performance (either fold average validation GM for action detection or ACC for laterality selection) for each individual participant recordings were retained.

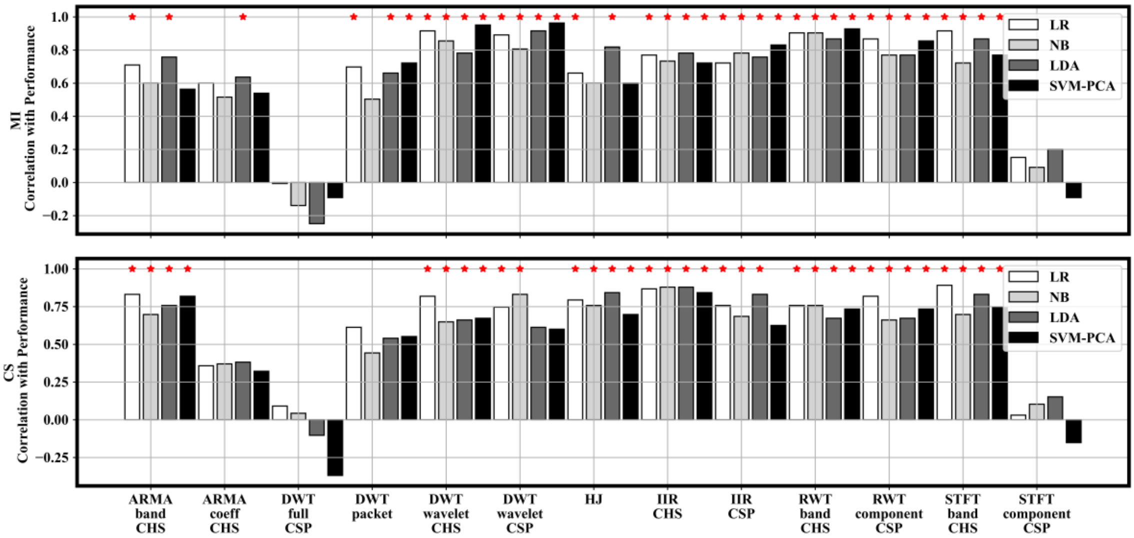


Figure 4.30 - Correlation between feature pre-screening metric and final action detection performance; (*) indicates a significant relation (Spearman rank correlation, p-value < 0.05); Pre-screening metric were obtained using default parameters and final performance with tuned parameters.

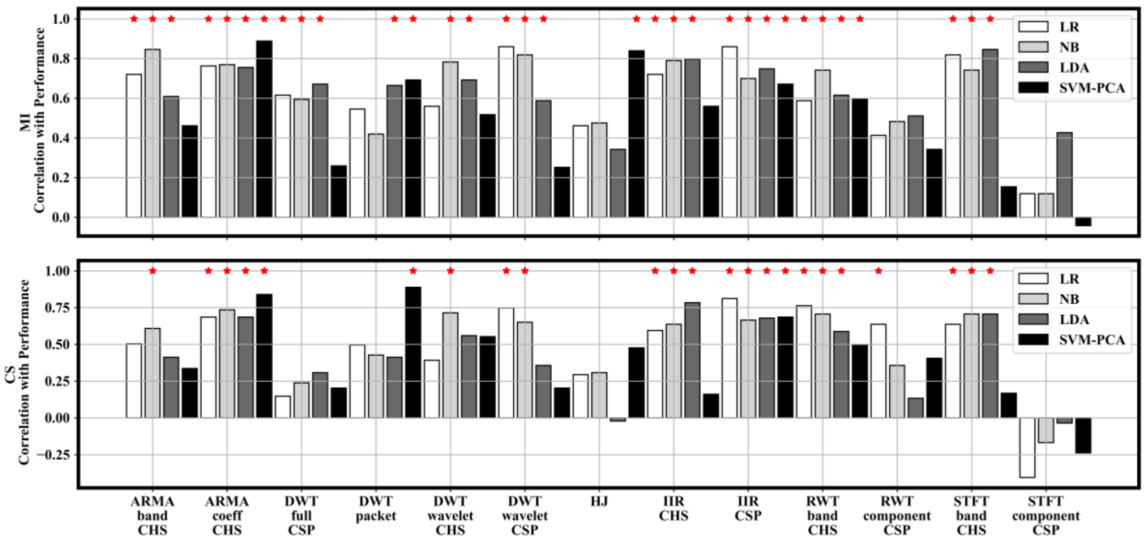


Figure 4.31 - Correlation between feature pre-screening metric and final laterality selectin performance; (*) indicates a significant relation (Spearman rank correlation, p-value < 0.05); Pre-screening metric were obtained using default parameters and final performance with tuned parameters.

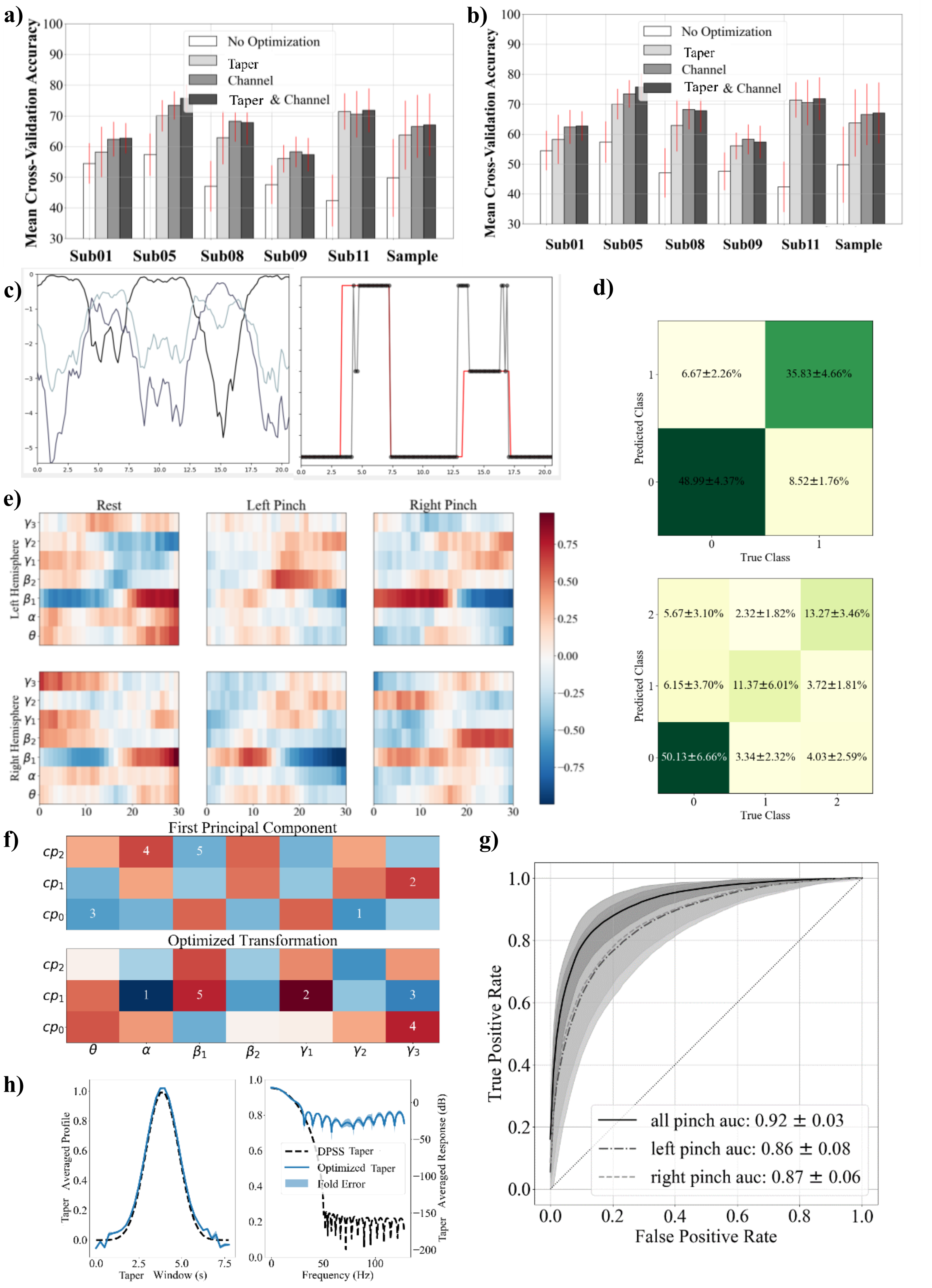


Figure 4.32 - © 2020 IEEE, Reprinted, with permission, from Martineau et al . (2020); **a)** Performance for action detection without laterality selection for every optimisation approach (none, spatial filter, taper, or both) and selected participants (Appendix A.B); **b)** Performance for action detection with laterality selection; **c)** Example of decoder trajectory using full optimisation in a combined action detection and laterality selection problem; **d)** Confusion matrices for 2 -class detection problem and 3-class detection and laterality selection using full optimisation; **e)** Optimised Wiener filter weight for action detection and laterality selection; **f)** Change in spatial filter weight after optimisation (number indicate top 5 weights); **g)** ROC curve for fully optimised model; **h)** Minor effect of taper optimisation.

G. Discussion

1. Relevance of Findings and Comparison to Past Studies

For the BO tuned decoder experiments, the detection rate median 0.75-0.25 (TPR, FPR) performance threshold was reached, and for laterality selection, a median rate of 0.7 accuracy. Results that decoder possessing over 0.7 median validation GM showed overall sustained detection throughout the action, some robustness to spurious activity and stable state transitions even in the nosier cases. This performance sufficiently justifies that they could be hypothetically deployable in specific applications and possibly controllable by users in an asynchronous fashion. Again, asynchronous decoding from STN-LFP was completed in both task instances while respecting the necessary constraint for real-time deployment. A limited number of studies has so far addressed this problem. The obtained action detection rate exceeded that of Niketeghad et al. (2015, 2018), showing some improvement in approach, but remained inferior to Mace et al. (2014) and Zaker et al. (2016). It is worth underlying that in the last instances, the first study was limited to a very small sample size and performed the detection of large body movement. The second study employed a complex methodology that may require signification modification to operate online. A fair comparison is thereby challenging to carry. The laterality selection performance was significantly inferior to the past offline studies (which surpassed a 0.8 accuracy threshold) (Mamun et al., 2015) but, for the first time, was obtained with asynchronous decoding, completed through sustained actions and under online constraints. This work thereby further comes to emphasises the importance to aby by those constraints to obtain realistic performance estimates. Lastly, compared to pathological state detection from STN-LFP decoders, the resulting performance was slightly inferior to that of Bakstein et al. (2012) or Yao et al. (2020). The result, however, suffered from the same heterogeneity raised by Bakstein et al. (2012). Compared to other deep-brain recording targets, in the case of action detection, STN-LFP was inferior to thalamic LFP decoding, where the 0.85-0.15 detection threshold had already been surpassed (He et al., 2021). Achieving accurate and robust motor state decoding may warrant some re-assessment relative to those findings.

Moreover, the obtained performance results were supported by systematic parameter tuning, which was unprecedented for this class of neural decoders. It was demonstrated that decoder tuning systematically and significantly increased performance. Those results, particularly with more complex decoder algorithms, warrant further adoption of BCI tuning and support that BO is a practical solution to that problem. The study emphasises the need for customised BCI parameters for both the user and the decoding task. Nonetheless, the performance improvement was limited depending on the extractor and classifier pair and the type of task. In case that, the default settings were sufficiently, BO then assisted only in fine-tuning, in some cases. It was shown that some parameters throughout the pipeline could be leveraged to increase performance better than others. Although classifier parameters remained the most important, the study did emphasise the importance of other parameters and other practical decoder characteristics such as the total number of features or the total analysis window. The NB case also showed that other decoder parameters could be leveraged to increase performance. Extractors preferred some specific parameter settings, notably regarding to the extraction window length. Lastly, the parameter relevance also varied depending on the decoding task.

In addition, the tuning platform has proven to be a valuable investigational tool. First, it allowed the systematic comparison of numerous algorithms for feature extraction, dimensionality reduction, channel selection and spatial filtering in a novel manner that was never carried out for deep-brain neural decoders. Those comparisons can help settle discrepancies between the disparate approaches to deep brain LFP decoding, especially regarding to deployment. Secondly, it permitted to address questions surrounding model complexity. The paradigm that reduced model complexity, primarily by limiting the number of total features or channels employed for decoding, led to increased model performance, was found here to be questionable. Reduced selection of channels, features or spatial filter components led to marginal performance decreases. Studies that have included all decoding channels have, for instance, delivered robust performances (He et al., 2020). As described by Flint et al. (2012), the relation between feature number, or channel, used for decoding, and performance can be non-convex and instead converge monotonically towards a maximal performance with steadily diminishing performance returns at every step increase in feature number. As discussed in Chapter 3.H, a multi-objective or constraint optimisation limiting model complexity is most likely required. Thirdly, the tuning approach helped evaluate the involvement of every parameter in the decoder architecture and can now serve for decoder re-design.

The BO approach has proved to be better adapted to the dataset characteristics than the gradient-backpropagation approach. Although it yielded an interesting proof of concept, the small dataset size made the gradient-backpropagation impractical and unreliable despite mitigation elements (primarily through the small pipeline design). More design limitations are discussed in Martineau et al. (2020). Nonetheless, it was the first time that action detection and laterality selection were carried out simultaneously and asynchronously within a single decoder architecture. The method could be revisited as the dataset, and the parameter space grows. Gradient-backpropagation will them computationally and practically scale more favourably than small sequentially fitted pipelines. Population participant decoder models rather than individual parameter models is likely the best incentive for this approach.

The principal limiting factor in performance was the overall quality of the features extracted from each recording. It would be reasonable to posit that this variance in feature quality is likely the source of the heterogeneity of the performance results. Attempts to optimise, and improve the SNR of the feature space using different extractors, features selection technics, BO or gradient-backpropagation have not yet been entirely satisfactory. The choice between a 'small feature/large channel' space against the 'large feature/small channel' space paradigm remains un-obvious. Adding as many features and channels as possible seems necessary to catch the signals with the best SNR in each recording. This outcome reduces the prospect of a more computationally efficient and compact decoder algorithm for an STN-LFP BCI system. Although more could be done to reduce noise and inter-trial variances (or denominator term of the SNR), heterogeneity in signal modulation (or numerator term of the SNR) cannot be obviously directly increased. In other words, in some recording cases, the biomarkers associated with motor states could simply be absent or too weak, and no further processing measures can be taken. Predicting a participant's performance using a pre-screening metric could help formulate reasonable inclusions criteria for studies in the meantime.

2. Selected Extractors and Classifiers, and Improvements to the Method

ARMA-band or STFT-band extractors were the extractors that led to the best performance and should be retained. Method improvements discussed in Chapter 3.H could be made specially to remedy the lengthy analysis window required by the STFT. Using just the ARMA coefficient may be justified for laterality selection given its good performance. Given its higher computational charge, the RWT delivered comparable performances but not enough improvement to justify its deployment. Even though the IIR extractors also worked reasonably well, more optimisation of the filter bank could also be done. HJ parameters typically lead to underperformance and should only be used as complementary features. Lastly, the DWT in all its declinations did not lead to particularly good solutions compared to other extractors. Performance was not insensitive to lattice parameters, but the optimisation of the mother wavelet was insufficient to overtake other methods.

CSP methods need to be re-assess. In combination with the IIR or DWT-wavelet extractor did not appear to offer significant performance gains nor reduction in the total number of employed features. STFT, DWT and RWT-component did not deliver the performance initially measured in the pre-screen stage. This gap could not be in part explained by overfitting but CSP components have shown inferior performances in the past (Gruenwald et al., 2019). Nonetheless, the sparsity of the representation remains an attractive attribute that could motivate further improvement to the method. Regularisation needs to be fully theorised for the complex-valued CSP, and the pseudo-covariance matrices should also be considered for a complete generalisation to non-circular complex data (Park et al., 2014). Technics such as spectrally weighted CSPs would avoid passing through the complex domain and still deliver similar results (Tomioka et al., 2006). The covariance matrix estimation could benefit from shrinkage regularisation (Ledoit & Wolf, 2003). The iterative singular-value decomposition algorithm proposed by Wang et al. (2020) would also offer a faster, more robust and regularised approach to the decomposition.

CHS worked as intended as it delivered some of the best performances and saved much manual labour in selecting the best recording channels. Both PCA and SQFE were effective dimensionality reduction technics. Moreover, CHS and SQFE only explore one branch of the permutation tree. Improved search technics could be implemented to find optimal subsets of features (Fukunaga, 1990). Further investigation of the SFA could also be warranted for short analysis window decoders.

Moreover, results showed that LR and LDA classifiers were best-suited for action detection. In contrast, SVMs were inadequate due to the computational cost of fitting on long-longitudinal data segments, high overfitting rate, and lack of prediction probability. The usage of SVMs is better justified for laterality selection. Selecting good estimators for a small and noisy longitudinal dataset is challenging. As summarised by Dempster et al. (2020), most state estimators achieving state-of-the-art accuracy suffer from very high computationally complexity and significant training times. There is a broader interest in expanding the set of tools for timeseries state estimation, and new methods are regularly made available. Classifier ensemble approaches appear to be promising (Gruenwald et al., 2019; Merk et al., 2021; Yao et al., 2020) and data balancing technics should also be further investigated.

Although the dual-threshold solution was simple and effective, more could be done to support a more robust state transition. Hidden Markov Models (HMM) were considered as they have been regularly employed in decoders for this purpose. However, their implementation was judged not trivial for timeseries classification as HMM are unsupervised. Obermaier et al. (2001) and Zaker et al. (2015) mapped the state to features using gaussian mixtures, K-means clustering or optimised jointly with the transition matrix using an expectation-minimisation algorithm. Ghimatgar et al. (2019) and Valstar & Pantic (2007) employed HMM models as post-processor on their classifier outputs. Recurrent neural networks were also considered as they have shown to be effective at capturing LFP temporal dynamics (Ahmadi et al., 2019), but preliminary results did not show any significant performance gains over other methods. Custom state machines were also investigated (Martineau, 2017). For detection, a pair of classifiers is used to detect the onset separately from the offset. The onset is arguably the more sensitive to impulsive feature types. The inherent problems of this method were defining the training windows around the state transition and the effects of state latching in the case of a missed offset. With BO tuning, state machines could still be revisited to implement more complex decoders. Two action detection decoders could be fitted to detect separately left, and right actions and a third laterality selection classifier could be employed as a tie-breaker. An overall objective function compound could then be optimised:

$$S = \sqrt{GM_L GM_R} \tag{4-17}$$

where GM_L is the action detection rate for the left-hand and GM_R for the right-hand. Fuzzy logic control could be lastly applied instead of thresholds, and BO could further help tune the various membership functions (Camara, Warwick, et al., 2015).

Another possible improvement could be the parameterisation of the output frequency f_d of the decoder as done by Golshan et al. (2018) or Shah et al. (2018). Reducing the refreshing output rate would mechanically increase the length of the extraction windows, reduce the number of timesteps to fit, and potentially increase performance when the SNR of the features is poor. Again the trade-off between slower signal processing and improved detection rate or accuracy needs to measure, and information transfer rate metrics may be one possible way of doing so (Singla et al., 2014).

Overall, the strategy for decoder tuning should be revisited. It may be more effective to distinguish two stages of tuning. The first stage would aim at building the best feature space representation of the signals, which would be optimised according to one a rapid metric function $M(X, y)$. It would be possible to start from a far larger feature space and distil it using either spatial filter or feature selection methods. Emphasis on the trade-off between representativeness of the feature space (M) and its sparsity (N_f) should be then measured.

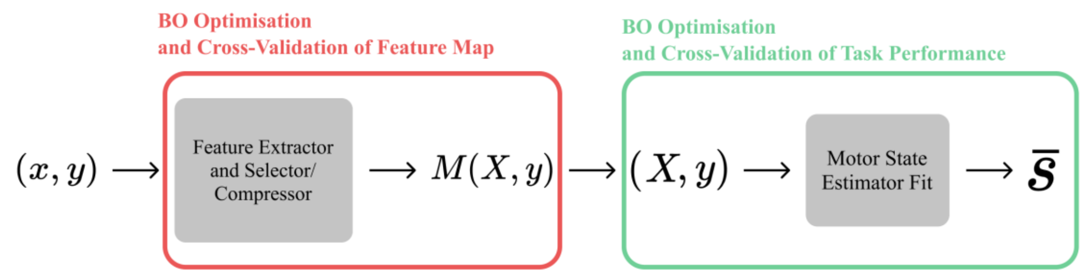


Figure 4.33 - Proposed two stage optimisation function for decoder tuning; First stage is dedicated to feature representation optimisation using metric function M and particularly to enforce sparsity by minimising N_f; Second stage is dedicated to task performance score S using the optimal feature map.

The second stage would focus on the state estimator fitting, optimising the performance function and robustness of the state transition (S) (Figure 4.33). This approach would be beneficial to optimise feature maps generated in mass by random convolutions (Dempster et al., 2020) or rectified DWT (Chapter 3.H). Alternatively further strategy to employ end-to-end optimisation with a gradient-backpropagation approach should be explored. The introduction of convolution layers could help bolster the feature extraction process (Golshan et al., 2020; Lawhern et al., 2018; Xie et al., 2018) provided enough data is made available for training.

H. Chapter Summary

Discrete asynchronous motor state estimation was achieved using STN-LFP during isometric contraction tasks. This work further demonstrates that STN-LFP signals can serve as an asynchronous and endogenous modality for decoding. Voluntary action can be detected, and bilateral electrode recording can select lateral end-effectors from real-time data. Nonetheless, achieving satisfying detection levels and correct selection rate was not a trivial endeavour, and real-world applications should be discussed with caution. The principal limitation encountered was the heterogeneous nature of the recorded dataset. Most strong ERS/ERD activity was embedded in some signal recordings but not others. This variability in signal strength translated into variability in the overall performance of the decoder. Therefore, not every participant would be a suitable candidate for an STN-LFP driven BCI. However, it was shown that those candidates could be selected through a pre-screening evaluation.

In this chapter, pre-processing and feature extraction methods were validated, and the unified approach to decoder parameterisation and tuning was tested for discrete state decoding. Extractors and state estimators could be selected based on the yielded population and individual performances, task type, and optimum settings. It was demonstrated that the different tuning technics (BO and gradient-backpropagation) could help automatise BCI calibration and help investigate the different optimisation choices. The work and results presented further demonstrate the versatility of the BO algorithm for BCI applications. The information provided by the parameter distribution and the surrogate models can help rethink the design of the decoder and its parameterisation, principally by re-adjusting search interval, re-defining the tuning function, or removing irrelevant parameters. As a product of this work, a repository of technics has been constituted, which can help to improve decoding performance further and tackle more complex decoding problems.

Chapter 5. Continuous State Decoding

A. Introduction

This chapter investigates STN-LFP neural rhythm modulations in response to variable adjustments in delivery rate of grip force. First, two paradigms, decoupling rate of change of force, or yank and grip force, are presented. Offline cross-participant analysis was carried to understand the overall behaviour of the neural signals. Continuous decoding strategies for yank and grip force were then explored.

1. Motivations

As discussed in Chapter 2.F, current continuous decoding paradigms have all focused on isometric grip force actions usually paced by visual cues (Merk et al., 2021; Shah et al., 2018; Tan et al., 2016). These cues indicate a target level that the participant needs to match as fast as possible. A fast response thus involves a large and short positive yank impulse followed by a similar negative yank impulse during action release. Different biomarkers may correspond better temporally to these two significant mechanical events than the sustained change in holding a given force level (Anzak et al., 2012). Again significant evidence gathered by Vaillancourt et al. (2004) has suggested that activity in the STN corresponds to self-regulating adjustments in the delivery rate of force. In theory, associating the correct biomarker to the correct motor state could enhance decoding performance.

However, yank decoding has so far been of limited use. For instance, Shah et al. (2018) concluded that combined yank and force regressions in a Kalman filter did not help to improve decoding performance for this specific type of action. Given the ballistic nature of the yank profile required to match different force levels rapidly, this outcome is not all surprising. Yank is intrinsically coupled to the demanded force in this paradigm, depends on the participant's biomechanical ability to quickly deliver the desired amount of force (Johansson & Cole, 1992) and does not constitute a state variable that requires dynamic adjustments. It may, therefore, be that yank is automatically delivered and not directly regulated by the STN (Vaillancourt et al., 2004).

Complex motor actions may require a more dynamical delivery of force. The grip profiles needed to be rethought to evoke LFP activity patterns corresponding to a modification in force rate separately from steady-state force levels. Several hypotheses, or supposed behaviours to be tested, describing a very simple yank and force-dependent LFP model, were formulated to orient the development of this new paradigm:

- **Hypothesis 1:** for a straightforward force and yank decoupled model, it should be expected that an arbitrary LFP biomarker signal x_i is only dependent on the yank rate $\dot{y}$. Those arbitrary features could be qualified of ‘impulsive’ (Martineau, 2017).

- **Hypothesis 2:** conversely, if x_i is not dependent on the yank, then it is only dependent on the force level y.
- **Hypothesis 3**: the mapping from $\dot{y}$ to x_i should be odd. In other words, negative gradients $\dot{y} < 0$ are dissociable from positive gradients $\dot{y} > 0$.
- **Hypothesis 4**: x_i is independent from the tracking error.

Hypotheses 1 and 2 attributed features to yank decoding and others to force decoding independently. A first alternative would be that x_i is not dependent on force nor yank. Except in the case of absent or feint ERS/ERD and given past findings, this seemed to be the least probable outcome. A second alternative could be that a feature x_i is both dependent on force and yank. The relation between x_i, $\dot{y}, y$ would hence be more a kind to that of a non-homogeneous differential equation model where x_i is effectively a source term driving the response (Tan et al., 2016). Although more complex, such a relationship should be perfectly drawable.

Furthermore, in the case of velocity decoding with MEA, each spike unit encodes motion velocity at the cellular level using preferred direction projection (Georgopoulos et al., 1986). It is assumed that in the simplest case of single-axial motion, at least one spike unit is required to encode strictly positive velocities, and at least a second spike unit encodes strictly negative velocities. The straightforward integration of the firing rate of both those units allows a direct mapping back to the position state. This symmetrical decoupling was deemed far less evident for STN-LFP. Finding spatial patterns encoding negative and positive yanks was unlikely given the available number of channels for recording. If positive and negative yank were encoded, it would be most likely through a temporal variation of x_i. For instance, a given neural rhythm x_i would synchronise with positive yanks and desynchronise with a negative yank (and vice-versa), while remaining idle at a steady baseline power when the force enters a steady state. The relationship between x_i and $\dot{y}$ would be described by an odd mapping. In comparison to hypotheses 1 and 2, the implication of the alternative to hypothesis 3 was thus far more problematic. An even relationship could be, for instance, that only the absolute yank $|\dot{y}|$ or positive yank $\dot{y} > 0$ is encoded. x_i would not alter synchronisation or desynchronisation in respect to $\dot{y}$. The integration of yank back to force would then become more complex. It would possibly have to rely on some features for negative yank and others for positive yank to reconstruct the complete trajectory.

Lastly, hypothesis 4 would confirm that the LFP recorded in STN are associated with a forward, rather than a feedback, control mechanism of the yank or force (Vaillancourt et al., 2004, 2007). The alternative would imply that ERS and ERDs would significantly decrease as the tracking error would diminish because of the learning process (Kristeva et al., 2007), rendering the STN-LFP less informative as the task learning converges.

2. Technical Investigation

As previously stated, new force profiles offered a new opportunity to investigate continuous state decoding tasks. Decoding those grip profiles would demonstrate that fine control could be achieved using an STN-LFP drive. The technical investigation pursued the same objectives set in Chapter 3 and Chapter 4. The decoder parameterisation, tuning, channel selection and feature pre-screening techniques were adapted for continuous state decoding. The asynchrony constraint was maintained, and decoders were tuned using BO, for both force and yank. Given the performance outcomes of the discrete state decoders, more emphasis was placed on testing different linear and non-linear estimators to predict either force or yank (Merk et al., 2021) rather than different feature extractors.

The outcome of both preliminary and decoding analyses could also help explain the decoding limitations observed for action detection and laterality selection. If it is confirmed that it is easier to map LFP features back to yank, this may warrant a reconsideration of those decoding paradigms (Martineau, 2017).

B. Decoupled Yank and Force Paradigm

Therefore, a first step was to re-imagine grip profiles to associate STN-LFP biomarkers with different demanded yank and force levels so to verify hypotheses 1 to 4.

1. Paradigm Description

The paradigm proposed by Vaillancourt et al. (2004) enforced an 'internal' generation of the force profile, where the users were trained to enforce different force development rates by themselves. For simplicity, the test participants were directly guided 'externally' through a grip profile using visual feedback (Fischer et al., 2017). Enforcing 'internality' would have been time-consuming and demanding on the DBS participants, limited the type of profiles that could be studied, and made it harder to synchronise different trials together to perform trial averaging. Participants were still given a series of mock trials to get acquainted with the task. Again Kühn et al. (2004) demonstrated that STN-LFP neural rhythms are modulated in self-pased tasks, and Spraker et al. (2007) that visual tracking is not a significant confounding factor. The endogenous hypotheses could be safely be maintained (Nicolas-Alonso & Gomez-Gil, 2012).

A time function $u \in \mathbb{R}^{N_k}$ was employed to generate a trajectory for a target cursor. The participants were tasked to match their grip force with the target using a second cursor to gauge how much grip force was applied. The position of the second cursor was normalised against the participants’ MVC.

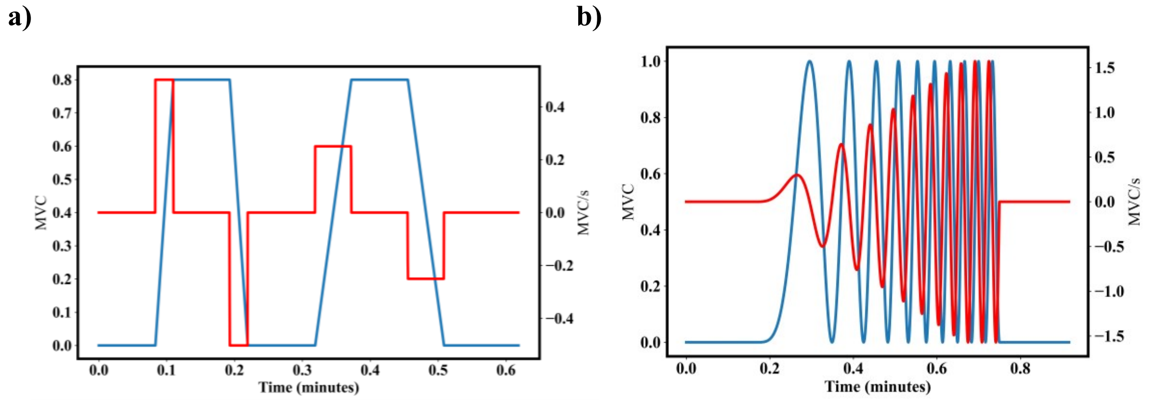


Figure 5.1 – Force tracking trajectories for proposed decoupled yank-force paradigm; (**Blue**) line indicates the demanded force level; (**Red**) line indicates the associated demanded yank. **a)** Two example ramp trials, first with high ascending (positive) and descending (negative) gradients, second with lower gradients; Gradients are held constants before and after force holding; Held force levels can also vary across trials; **b)** Example of chirp trial; Sinusoid with accelerating rhythm, transitioning from slow controlled force levels to fast and short pulses; Average yank level linearly increases with sinusoid while force level is held constant.

The first grip reference profile was inspired from Vaillancourt et al. (2004) and denominated as a 'ramp' profile. The ramp profile consisted of three stages (Figure 5.1a): a linear ascending ramp, a holding plateau, and a descending ramp. The holding plateau was introduced to compare the LFP response when the force level reached a steady-state with maintained yank in the ascension and descent. Adding the descending ramp to incite both positive and negative gradients in the task was essential to test hypothesis 3. For a single trial, this trapezoidal trajectory was described as:

$$u[k] = \begin{cases} \frac{g}{f_s}k & 0 \le k < k_1 \\ u_L & k_1 \le k \le k_2 \\ u_L - \frac{g}{f_s}(k - k_2) & k_2 < k \le k_2 + k_1 \end{cases} \tag{5-1}$$

$$k_1 = f_s \frac{u_L}{g} \tag{5-2}$$

$$k_2 = f_s \left(\frac{u_L}{g} + \tau_L \right) \tag{5-3}$$

where f_s is the sampling frequency of the trajectory, g is the gradient of the ramps (in unit force per second), u_L is the plateau height (in unit force), τ_L the duration of the plateau and, k_1,k_2 the plateau index. For convenience, u_L is normalised against the MVC of the participants such that $u[k] \in [0,1]$. The trapezoid enforces a simple yank sequence $(g, 0, -g)$ as depicted in Figure 5.1a. The symmetry of the ramp made it straightforward to test hypothesis 3 by simply comparing the LFP power features recorded during the ascending section to those recorded during the descending section. Each ramp profile instance could be characterised by the two principal parameters (g, u_L) each respectively controlling the steady yank and the force levels independently. The state-space $(y, \dot{y})$ was sampled discretely with different trapezoidal units to test hypotheses 1, 2 and 3. Three gradients were chosen such that $g \in \{0.125, 0.25, 0.5\}$ MVC/s, i.e., slow, medium, fast. Two different force levels were then chosen such that $u_L \in \{0.4, 0.8\}$ MVC and held for $\tau_L = 5$s to obtain low and high holding units. The resulting 6 conditions were repeated at least 15 times, and 5s were allowed between each trial.

Albeit its straightforwardness, the limitation of the ramp was that the yank did not have to be continuously adjusted during the task but selected and executed in the correct sequence. A second profile was hence developed to create a more dynamic force tracking task based on a sinusoidal chirp trajectory (Figure 5.1):

$$u[k] = \frac{\mathrm{A}}{2}(\sin(\phi[k]) + 1) \tag{5-4}$$

where A is the demanded amplitude, calibrated to be 0.8 of a participant's MVC and ϕ is the phase function of the signal such that:

$$\phi[k] = \frac{2\pi}{\tau_C f_s^2} \sum_{k=0}^{\lfloor \tau_C f_S \rfloor - 1} k \tag{5-5}$$

where f_s is the sampling frequency and τ_c the duration of the chirp, set to 30s. The effect researched was the temporal linear trend of the chirp derivative. The overall required yank increased linearly with each chirp cycle requiring a continual adjustment of the delivered rate:

$$\dot{u}[k] = \frac{A\pi}{\tau_c f_s} k \cos(\phi[k]) \tag{5-6}$$

The chirp thereby evaluated how the LFP modulations transitioned from a slow and precise control to a more demanding and ballistic regime. To test hypotheses 2 and 3, the state space $(y, \dot{y})$ with this trajectory could be continuously sampled. Moreover, if its baseline was adjusted, the yank trajectory was more a kind of a train of positive impulse with linearly increasing amplitude. Unlike the trapezoid trajectories, the intermittence between positive and negative gradient regimes was not as marked and may have been easier to map with an even function.

2. Paradigm Analysis

The datasets presented Appendix A.C was collected using both chirp and ramp profiles. An offline recording analysis was carried in Matlab (The Math Works, Inc. MATLAB. Version 2017a) using the Fieldtrip toolbox (Oostenveld et al., 2011) to test the hypotheses. 6 recordings produced from 4 participants were employed in the analysis. Two participants were tested for both their left and right hand.

The usual pre-processing steps were carried out. First, drift and mains artefacts were attenuated using high-pass and notch filters. The dynamometer and target recording timeseries were directly resampled at 16Hz. Once pre-processed, the recordings were segmented and sorted by conditions (six ramp conditions and the additional chirp condition). Each condition was composed of a triplet (x, u, y) where $x \in \mathbb{R}^{N_k \times N_{trial} \times N_{ch}}$ was the LFP timeseries, $u, y \in \mathbb{R}^{\frac{N_k}{R} \times N_{trial}}$ were the target and dynamometer arrays, respectively. N_k is the length of the trial, N_{trial} the total number of trials, and N_{ch} the total number of channels. In practice, each trial had slightly different lengths. A common time vector, t, was defined such that $t \in \mathbb{R}^{\frac{N_k}{R}}$ to synchronise the series with $t[k] = 0$ corresponding to the onset of the gripping action. Trials were selected to mitigate inter-subject variability. First, those with inadequate tracking errors were removed. The tracking error e was defined as:

$$e[k] = u[k] - y[k] \tag{5-7}$$

Each trial was characterised with a single tracking performance scalar, here chosen to be the mean-square error (MSE):

$$MSE = \frac{1}{N_k} \sum_{k=0}^{N_k - 1} e[k]^2 \tag{5-8}$$

The MSE of the trials were collected in a population-wide vector to automatically tag outliers. Each trial in x was then visually inspected for motion artefacts in the LFP signals. The union of both motion artefact and MSE outliers were removed prior to the time-frequency analysis.

Time-frequency analysis was carried with the Fieldtrip CWT-Morlet implementation, as defined in (3-22),(3-23), (3-24), yielding the power array, or spectrogram, $x_p \in \mathbb{R}^{\frac{N_k}{R} \times N_{trial} \times N_l}$. 48 wavelets per octave were employed between 1 and 512Hz. Fieldtrips automatically performed the resampling along t. An FWHM of 1000ms to 50ms was employed along the wavelet scale. A 2D 6x6 square image median spatial filter was also used to remove background noise and spurious activity. The normalisation of the ERS/ERD activity was also done in every trial to account for baseline drift. Given the temporal mean reference activity, $x_r \in \mathbb{R}^{N_{trial} \times N_{ch} \times N_c}$, obtained in the second leading to the trial onset, the activity gain in dB, x_d, was given by:

$$x_d[k, l, i, j] = 10 \log_{10} \left(\frac{x_p[k, l, i, j]}{x_r[l, i, j]} \right) \tag{5-9}$$

For each participant and each condition recording, channels were then visually selected based on β ERD and γ ERS. The population spectrogram for each condition was obtained by stacking the channel and trial dimensions of x_d for all participants and taking the trial average (3-2).

The analysis of the LFP features in the trapezoidal trials consisted in taking temporal-spectral averages to grid each spectrogram x_d into frequency bands and into rising, holding and dropping action sections. The individual spectral-temporal bin μ_i was given by:

$$\mu_i = \frac{1}{(k_2^* - k_1^*)|S_b|} \sum_{k=k_2^*}^{k_1^*} \sum_{l \in S_i} x_d[k, \dots, l] \tag{5-10}$$

where k_2^* and k_1^* are the segmenting index (such that $(k_1^*, k_2^*) = (0, k_1)$ for the rising edge, (k_1, k_2) for holding and $(k_2, k_1 + k_2)$ for the drop), and S_i is the set definition of the i^{th} band (3-20). The powerbands were adjusted following a visual inspection of the spectrograms. The lower frequencies were broke-down in two such that δ (1-4Hz) and θ (4-6Hz), the β domain such that β_1 (10-20Hz) and β_2 (20-30Hz). α was redefined between 6-10Hz. The high-frequency γ was extended to near the Nyquist frequency such that γ_3 (90-400Hz) and two sub-bands were defined in the lower γ domain such that γ_1 (30-60Hz) and γ_2 (60-90Hz). Individual channels were treated as individual trials such that $\mu_i \in \mathbb{R}^{(N_{trial}N_{ch})}$. All μ_i were concatenated across all participants.

A two-way ANOVA was carried to test hypotheses 1 and 2. Every μ_i (across rising, holding, and resting segments and for every frequency band) was analysed, and the resulting F-statistic probabilities were recorded. To test hypothesis 3, the means of the rising and dropping segments μ_i distributions were analysed using a paired t-test. The t-statistics probability was recorded for every condition and every band.

In the chirp case, the LFP signals were again partitioned in N_{bd} frequency bands and transformed x_d in to $x_{bd} \in \mathbb{R}^{\frac{N_k}{R} \times N_{trial} \times N_{ch} \times N_{bd}}$. The yank adjustment in the trajectory followed two timescales: a short timescale to track the period of the oscillating target; a slow time to respond to the progressively accelerating tempo of the oscillations. A fourth-order polynomial was employed to detrend the band signals:

$$x_r[k, \ldots] = x_{bd}[k, \ldots] - \sum_{i=0}^{4} a[\ldots, i]k^i \tag{5-11}$$

where x_r is the residual array and $a \in \mathbb{R}^{N_{trial} \times N_{ch} \times N_{bd} \times 5}$ the polynomial coefficient array for every trial, channel and band. The polynomial trend and residual averages could then be visually inspected.

Lastly, a cross-correlation analysis was carried on x_{bd} for the ensemble of ramp and yank conditions to measure the strength of a linear relationship between the LFP features and motor states in time. The motor state variables included the grip force y, the yank $\dot{y}$, the absolute yank $|\dot{y}|$, the tracking error e and the absolute tracking error $|e|$. A maximum lag/lead of 5s was allowed. Signals were smoothed and normalised prior to the analysis. Cross-correlation was also measured in-between y, $\dot{y}$ and e.

C. Continuous State Estimation with Sequential Fit and BO Tuning Approach

A similar workflow to Chapter 4 was adopted for continuous state estimation. The relationship strength between the STN-LFP features and the continuous state space was rapidly measured. Continuous state estimators based on regression algorithms were selected, and decoders were tuned using BO.

1. Measuring Feature Space and State Space Relation Strength

Measuring the relation between the feature X and state space $(y, \dot{y})$ was again deemed necessary given the anticipated heterogeneity of the recordings. Pearson's correlation was first considered to measure simple linear relationships. Assuming that X was demeaned and normalised, for the state variable y (or $\dot{y}$), the correlation vector $c \in \mathbb{R}^{N_f}$ (for every N_f feature) was given by:

$$c[l] = \frac{\sum_{k=0}^{\frac{N_k}{R}-1} X[k,l](y[k] - \bar{y})}{\sqrt{\sum_{k=0}^{\frac{N_k}{R}-1} (y - \bar{y})^2}} \tag{5-12}$$

were $\bar{y}$ is the mean of y. The coefficient of multiple correlations (J. Cohen et al., 2003), MR, was then employed to measure the combined relationship of the features in X with y:

$$MR = \sqrt{c^T S_X^{-1} c} \tag{5-13}$$

where S_X is the internal correlation matrix of X. MR became of interest since it resembles closely the CS criterion (4-2) and could thus be employed in the SQFE algorithm. Given its versatility, the MI criterion was again implemented. MI can be employed to measure feature strength for both discrete and continuous variables (Pedregosa et al., 2011; Ross, 2014). It was again used for channel selection.

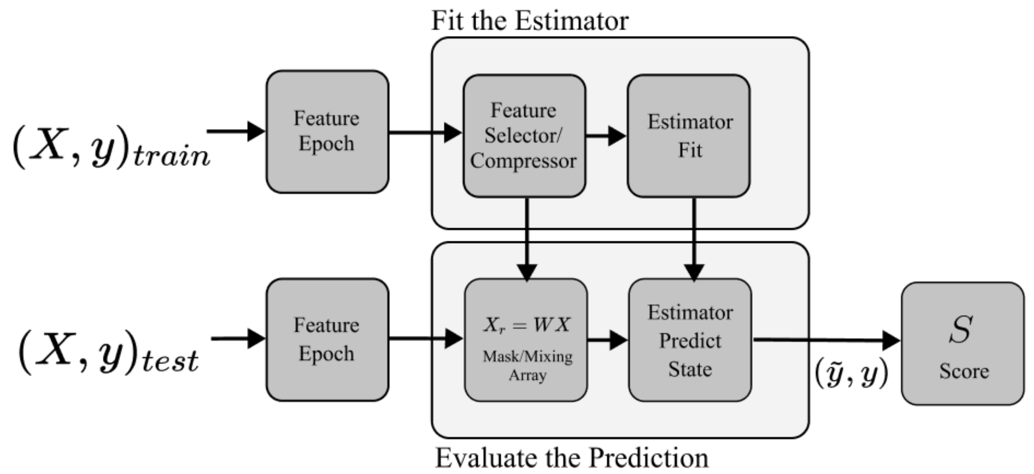


Figure 5.2 - Simple continuous estimator pipeline with optional feature selection stage; y and $\dot{y}$ are interchangeable; Both training and testing branches are maintained; Pipeline is duplicated during cross-validation; Estimator consists of a regressor (RR, ENR, KNR, SVR, SVR-RBF, GBTR); The scoring function S was the coefficient of determination r^2; State transition stage and combination of decoded y and $\dot{y}$ were not at first included.

2. Regression Estimators

The methodology presented in Chapter 4.C and the models presented in Chapter 3.G were again followed to organise the continuous state estimator (Figure 5.2). Again, a similar structure was employed for the pipeline, including feature epoching, possible dimensionality reduction, and followed by regression estimation.

First, linear regressive models were investigated. Both **Ridge Regression 1)** (RR) and **Elastic-Net Regression 2)** (ENR) regression were investigated (Pedregosa et al., 2011). RR imposed L_2-norm (or Tikhonov) regularisation on the optimisation of w whilst ENR combined both L_2 and L_1-norm sparse regularisation. An hyper-parameter α_{L_2} was thus needed to control the strength of the L_2-norm in both cases. A ratio r_{L_1} balanced the contributions of L_2-norm and L_1-norm regularisation in the ENR case. Both methods have already been employed in STN-LFP regression (Merk et al., 2021; Shah et al., 2018).

Other linear alternatives and non-linear regression technics were also tested: **K-Nearest Neighbour Regression 3)** Nearest neighbour algorithms are non-parametics and versatile algorithms. They have been tested for discrete state decoding both for classification (He et al., 2021) and for clustering (Zaker et al., 2016). The k-nearest neighbour regression (KNR) is one other algorithm variant. It required the tuning of the number of neighbours n_k employed for prediction. **Linear and Non-Linear Support Vector Regression 4,5)** the robustness, versalibility and accessibility of support-vector algorithms can also serve regression problems (Pedregosa et al., 2011; Smola & Scholkopf, 2004). As in classification, kernel transformations in particular permited to easily tackle non-linear problems. Two algorithm variants were thus tested: a linear support vector regression (SVR) and radial-kernel basis version (SVR-RBF) again combined with PCA for dimensionality reduction. Regularisation was controlled by parameter C, and the radial-kernel basis reach was controlled with a parameter γ_c. **Gradient Boosted Tree Regression 6**) gradient boosting algorthims have become important tools to optimise ensembles of small classifier, such as decisions trees (Friedman, 2001), to solve non-linear problems and deliver superior performances on small and medium datasets. Merk et al. (2021) showed promising decoding performances when applying them to ECoG regression for grip force decoding. Gradient-boosting tree regression (GBTR) was thus implemented. It remained however an advance algorithm which involved many hyper-parameters: learning rate η, maximum depth n_d, number of estimators n_e and subsample rate r_s. GBTR was also employed with PCA to avoid prolongued fitting times.

Lastly, the coefficient of determination, r^2, was employed as a scoring metric for BO tuning (Merk et al., 2021):

$$r^2(y,\tilde{y}) = 1 - \frac{\sum_{k=0}^{N_k/R-1}(y[k]-\tilde{y}[k])^2}{\sum_{k=0}^{N_k/R-1}(y[k]-\bar{y}[k])^2} \quad (5\text{-}14)$$

3. BO Tuned Decoders

Decoders were fitted and tuned on sections of both the ramp and the chirp dataset. Only the first 6s of the ramps were selected has it truncated every trial to the same length. Slower conditions took 6s seconds to reach the force plateau while faster ones would not have enough time reach the descending ramp. Offline analysis indicated that hypothesis 3 was most likely not valid and that the most changes in STN activity occurred in the ascending ramps. Only the first four slow periods of the chirp actions were decoded. The second half of the trajectory was judged possibly to ballistic and dynamic for decoding. Offline analysis also showed large trend changes towards the tail of the chirp, suggesting that individual ERS/ERD may be harder to assign to successive grip impulses.

Table 5.1 - Pipeline parameters contained in ρ and tuned with BO for continuous state decoders including: parameter description and stage assignment, default values stored in ρ_o; search interval for every dimensioned searched from which ρ is drawn; Dimension searched through a log mapping (Y) or through a linear mapping (N).

Stage	Parameter Symbol	Description	Default Value	Search Interval	Log Scale
Channel Selector	N_c	Number of selected Channel	2	$[\![1, N_{ch}]\!]$	N
Feaure Extraction	n_o	Number of overlaps	0	$[\![0,1]\!]$	N
	β_K	Kaiser window parameter	5	[0,15]	N
	λ	Box-Cox exponent	0	[0,2]	N
	r_σ	Ratio of process to observation ratio for Kalman filter	5×10^{-5}	$[1\times10^{-5},100]$	Y
	N_w	Length of the feature window	0	$[\![0,7]\!]$	N
Continuous State Estimator	α_{L_2}	RR and ENR L_2-norm regularisation constant	0.01	[0,100]	N
	r_{L_1}	ENR L_1-norm regularisation constant	0	[0,1]	N
	n_k	KNR number of neighbours	5	$[\![3,15]\!]$	N
	C	SVR Regularsation constant	1	$[1\times10^{-3},5]$	Y
	r_c	PCA compression ratio	0.1	[0,1]	N
	γ_c	SVR-RBF kernel parameter	0.1	$[1\times10^{-3},5]$	Y
	η	GBTR learning rate	0.1	$[1\times10^{-2},0.6]$	Y
	n_e	GBTR number of estimators	100	$[\![10,500]\!]$	N
	n_d	GBTR tree maximum depth	3	$[\![2,10]\!]$	N
	r_s	GBTR subsample ratio	1.0	[0.1,1]	N

Moreover, STN-LFP signals were pre-processed as outlined in Chapter 3.C and resampled at 512Hz. The decoder frequency output f_d was set to 4Hz with a maximum feature window (controlled by N_w) of 2s and a maximum feature extraction window of 500ms (controlled by n_o). Reducing f_d and allowing for larger windows was deemed necessary to eliminate noise from the feature signals. Given its good performance in the past decoding tasks and its simplicity, only the STFT feature extractor was tested to extract power band signals with a second order Kalman filter (3-37). The filter had the most effect when r_σ was set to a minimum and nearly inactive when r_σ was set to a maximum. Only certain power bands were selected for decoding, namely θ (4-6Hz), β_2 (20-30Hz), γ_1 (30-60Hz), γ_2 (60-90Hz) and γ_3 (90-400Hz). α and lower β_1 were removed after condisdering findings from Khawaldeh et al. (2020), suggesting that bursting activity in the 10-24Hz subdomain may overshadow activity relatable to motion. Pre-action activity was particularly strong in both bands. The δ band was also excluded to mitigate contribution of low frequency motion artefacts. Lastly the Box-Cox transform (3-34) was employed to alter the transformation of the STFT power signals from logarithmic to square powered. Merk et al. (2021), Shah et al. (2018) and Tan et al. (2016) have all preferred employing the raw power signals and logarithmic transformation had yet to be tested.

D. Results

Overall, the cross-participant paradigm analysis was consistent with previous findings. High β_2 ERD and high γ ERD responded to changes in executed force level. Those bands also had some specific couplings to yank. More importantly, the result showed that lower frequency bands δ to β_1 were mostly reactive to changes in yank and not to force. Evidence suggested that distinguishing negative from positive yank was difficult. The analysis and feature prescreening showed that the feature signals quality was overall relatively poor. Decoding force and yank continuously in these dynamic tasks were thus very challenging. Most linear models failed to produce viable decoding results. Nonetheless, given enough data, the advance non-linear models performed relatively well and appeared to be able to decode some properties of the motor state variables. Lastly, yank decoding tended to be inferior to force decoding.

1. Offline Cross-Participant Paradigm Analysis

Figure 5.3a presents the population average spectrograms resulting from the preliminary CWT analysis of the LFP signals for each ramp condition. Most ERS modulations did not exceed 1.5dB and ERD modulations, -3dB, which was consistent with previous studies studying isometric contractions. Tan et al. (2016) reported a maximal β ERD 80% inferior (or with a power decay of approximately -2.5dB) and a maximal γ ERS of 60% superior (or approximately 2dB) to the baseline power. The most visibly apparent feature was the suppression of the low-frequency δ throughout the action. Significant activation of the band also occurred at the action onset and offset but was not necessarily sustained thoughtout the ascending ramp. This impulsive activation propagated through all low bands, θ and α, and medium bands, β_1 and β_2. For stepper ramp conditions, this activation pattern tented to be more pronounced. θ activity was very pronounced at the action onset, sustained throughout both ramps and re-ignited at the action offset. After a pre-action onset burst, α activity was more dampened. Bursts were present following certain trajectory dis-continuities, such as the ascending and descending ramp apexes and towards motion onset. Small β bursts coincided with the low-frequency discharges then followed by a ERS in both the lower β_1 and higher β_2 band subdivisions and typically pervaded through the action. Regain of activity was perceptible mid action but as consistent. The ERS patterns were more emphasised in stepper and higher conditions. The high-frequency activity was weak and perceiving it was more difficult as ERS patterns could be as apparent as the stationary noise. Nonetheless bursts of γ_1 were marked at the onset of the action and during the ascending ramp. Their timing seemed to

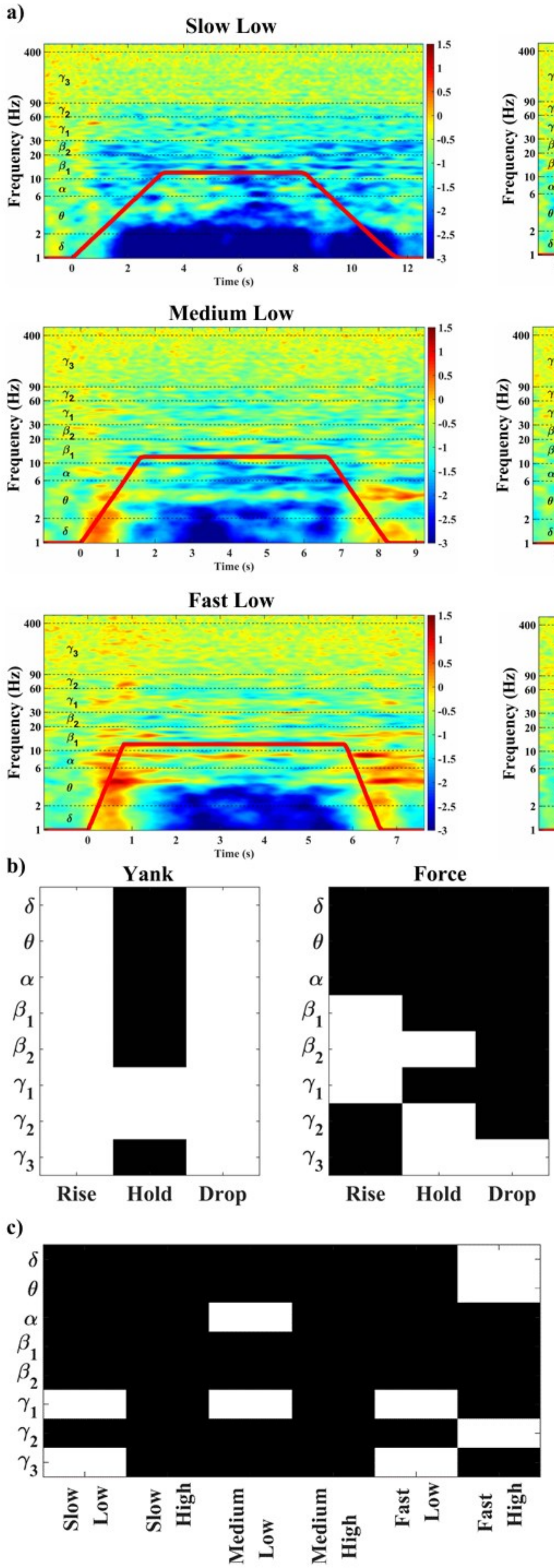


Figure 5.3 - Analysis results from the ramp tracking paradigm; **a)** Population average spectrogram for each of the ramp conditions (High and Low levels, Slow, Medium and Fast gradients); (**Red**) lines are the target force trajectories; All 6 participant recordings from the 'Gradient' dataset were included (Appendix A.C); Spectrogram was obtained from Morlet CWT analysis (48 wavelets per octave spanning 1 to 512Hz, FWHM linearly set from 1000ms to 50ms); Trials were checked for artefacts and abnormal tracking performance; After smoothing and normalisation their were averaged per condition across participants and channels; All colormaps are presented in decibels; **b)** ANOVA results per band activity for the different stage of ramp tracking (Rise, Hold, Drop) carried across gradient conditions (Slow, Medium and Fast) and force conditions (Low and High); Mean signal activity was averaged temporally and spectrally in the corresponding segment and band from trial spectrogram (5-10); Results are presented in a matrix; (**White**) cell indicate a meaningful change in signal baseline (p-value < 0.05); **c)** Results for the paired t-test between mean activity in the rising segment and the drop for every force, yank conditions, in every activity bands of interest; Results are presented in a matrix; (**White**) cell indicate a meaningful change in signal baseline (p-value < 0.05).

coincide with the initial low frequency bursting. γ_1 and γ_2 could as well be suppressed throughout the holding plateau of the action, but not in every condition.

The ANOVA results testing the six different trapezoidal conditions are presented in Figure 5.3b. The result showed that all features mean activity varied significantly depending on the ascending gradient yank. The same held in the case of descending gradient. Only the mean activity in the γ_1 and γ_2 during the holding phase were dependent on the yank selected to reach the holding point. The mean activity for each of the lower frequencies δ, α and θ was found to be completely independent of force level throughout the action. In this regard, δ, α and θ satisfied entirely hypothesis 1. Mean activity for β_1, β_2 and γ_1 was dependent on the force level to be reached during the ascending ramp. However, only the mean activity for β_2, γ_2 and γ_3 varied significantly during the holding phase. γ_3 was also varied during the descending phase.

The paired t-test results are shown in Figure 5.3c for every band and condition. It was observed that activity distribution did not vary significantly in most cases. This test further reinforced what was already observed in the spectrogram: activation patterns appeared mostly symmetric across negative and positive sustained yank trajectories.

Furthermore, mean activity trends and fluctuations are also presented in Figure 5.5 alongside the average population spectrogram for the chirp condition Figure 5.4. Regarding the chirp spectrogram, the large δ activity bursts were not present. However, suppression of the band activity could still be observed and was marked in a concave activity trend Figure 5.5. Comparatively small regain of activity coincided with slow yank peaks and accelerated action. θ bursts were still clearly present. The rhythm of the burst seemed to lead the force application wave and coincide with the peak yank, visible in both the spectrogram and the variation trace. The coincidence was more difficult to discern as the latency between peak yank and peak force diminished, but the activity trend was stable throughout the action. An important α and β_1 bursts were present prior to the action onset and led on θ . Bursts in both domains were intermittently observed during the action and appeared to lead on the yank peaks. Overall, the trend of α activity steadily increased with the tempo acceleration, whereas the β_1 was more stable throughout and possessed a slight dip during the first force period, most likely associated to the ERD observable in the spectrogram. The β_1 ERD was, however, weak in comparison to the β_2 which was engaged shortly after motion onset and seemed steadily maintained throughout the action. A steady trend lowered from the action onset captured the persistence of the ERD and seemed only lightly affected by the acceleration of the action cycle. The variation of β_2 were small relative to the maximal depth of the ERS trend and inconsistent with either the force or peak yanks. γ_1 activity clusters were visible during the first few force peaks and accumulated as the tempo of the action increased. This accumulation was further captured in the band trend and precisely matched the chirp ramp. A similar behaviour could be attributed to γ_2 with the addition of an ERS occurring prior to the development of

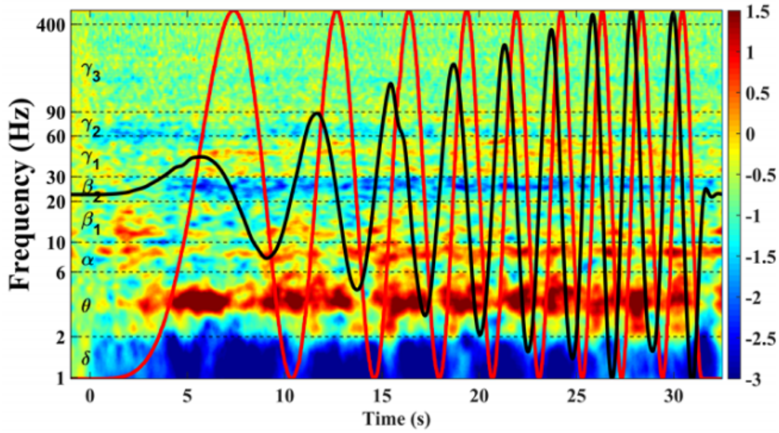


Figure 5.4 - Analysis results from the chirp paradigm; Population average spectrogram; Spectrogram was obtained from Morlet CWT analysis (48 wavelets per octave spanning 1 to 512Hz, FWHM linearly set from 1000ms to 50ms); Trials were checked for artefacts and abnormal tracking performance; After smoothing and normalisation there were averaged, across participants and channels; (**Red**) line is the target force trajectory; (**Black**) line is the target force yank trajectory; All colormaps are presented in decibels.

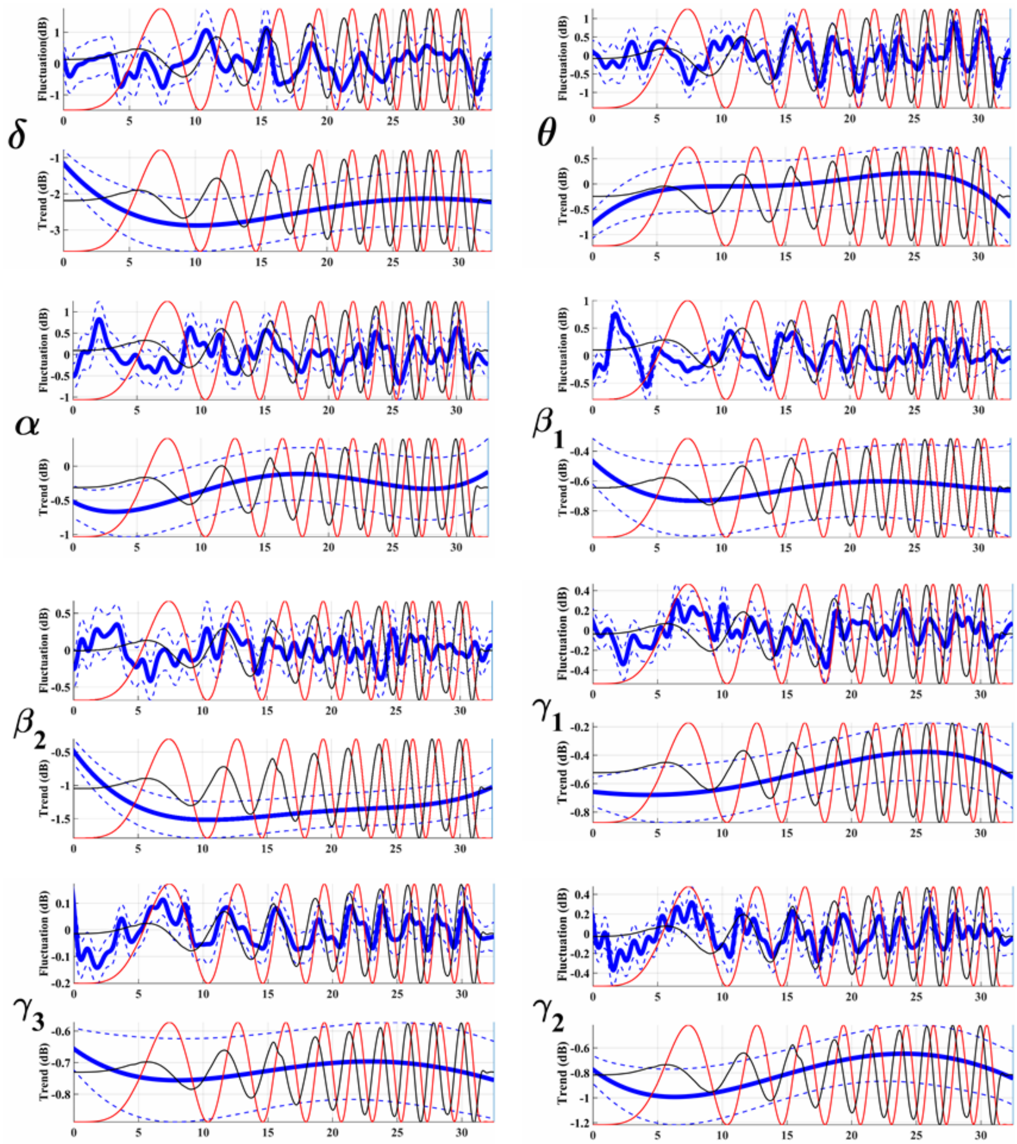


Figure 5.5 - Cross participant trends and residuals for every frequency band (thick solid **Blue** line) whilst chirp trajectory tracking; (**Dotted Blue**) lines indicate a 5% confidence interval; (**Red**) lines are the target trajectory; (**Black**) lines are the yank target trajectory; Trial trends were estimated from every trial spectrogram using a fourth polynomial previously smoothed, normalised and spectrally averaged in the band of interest; Trial fluctuations were estimated by subtracting trend; Trends and fluctuations were averaged; All average trends and fluctuations are presented in decibels.

the yank. This ERS pattern was then repeated between the first force peaks, and then small activity bursts were observed in the later stage of the action. Those bursts appear to occur both at the peak yank and the peak force. Although more feint, a similar patent extended to γ_3. γ_3 activity peaks coincided very precisely with demanded yank peaks.

Lastly, Figure 5.6 presents the outcome of the correlation analysis between the different power bands of x_b and the different motor states for the chirp and the concatenated ramp conditions. Overall correlation for a given feature and motor state-pair did not exceed a maximum absolute correlation value of 0.1. The correlation of band activity with direct grip force y was surprising in ramp conditions. The cross-correlation dropped across all bands at zero lag. This drop was less important in higher frequency bands than in lower frequency. Correlations also then transitioned from positive to negative. Moreover, correlations with the yank $\dot{y}$ and absolute yank $|\dot{y}|$ were more substantial than that of the force. In all bands, the correlation was strongest when the feature signal possessed a slight lead (typically inferior to 500ms) over the yank. This lead would suggest that activity bursts occurred prior to the yank onset. In the case of β_1 and β_2 the peak correlation was followed by a drop to a negative peak correlation lagging behind the yank signal. This negative correlation could be attributed to the β ERD. The correlation with the error e was typically small, excepted with γ_2 and γ_3. Correlation between those two variables was sharply onset as band signal started lagging. The cross-correlation with the absolute error $|e|$ corresponded to those observed with $|\dot{y}|$.

In the chirp condition, correlation with y depicted relations more akin to the expected LFP ERS/ERD. Negative correlations were observed with δ, β_1, β_2; positive correlation $\theta, \alpha, \gamma_1, \gamma_2, \gamma_3$. Yank correlations were significant for θ and the γ bands. The correlation patterns with e and $|e|$ were neither consistent nor significantly strong. However, the error signal was strongly correlated to both the force and the yank in both ramps and chirp conditions. A slight lead of the error signal was observed in the chirp while remaining in phase in the ramp case. It would stand to reason that rapid yank could lead to significant error, especially in the accelerating transition of the chirp task.

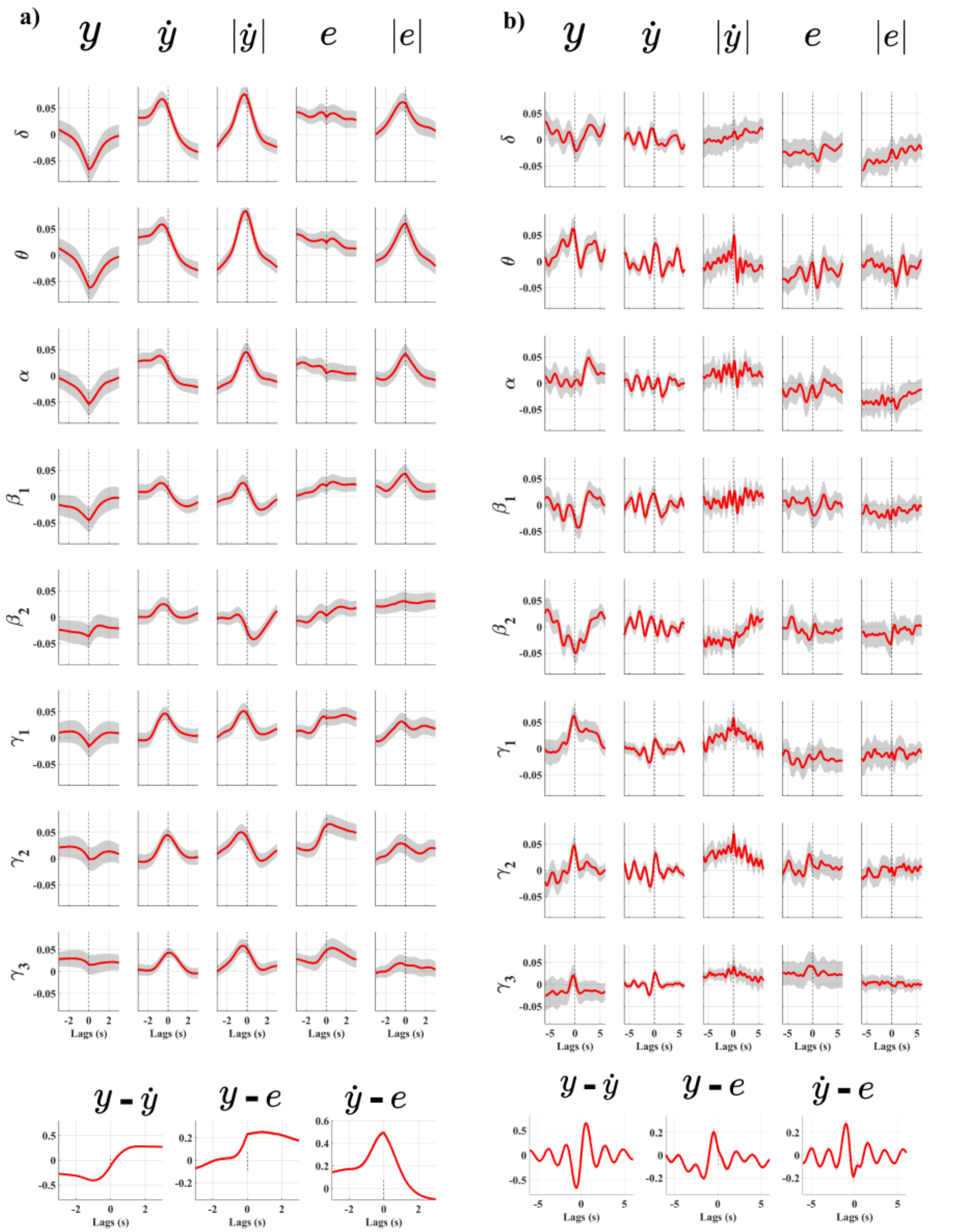


Figure 5.6 - Overall population correlation analysis averages for **a)** the overall trapezoid condition and **b)** the chirp condition; (**Red**) lines are the correlation function averaged across trials and participants; (**Grey**) boundary indicates a 5% confidence interval; Cross-correlation was estimated for every trial spectrogram smoothed, normalised and spectrally averaged in the band of interest; A maximum lag/lead of 5s was allowed; Band activity was correlated the grip force y, the yank $\dot{y}$, the absolute yank $|\dot{y}|$, the tracking error e and the absolute tracking error $|e|$.

2. Feature Screening

Features were extracted with the STFT-band-CHS extractor with default settings for prescreening. Figure 5.7 presents distributions of individual feature correlations, MI, MR and mean MI were computed for both yank and force. Albeit weak in all instances, feature and state relationships were consistent across ramps and chirp conditions. Common information content and correlations were lesser with yank and force. θ band correlated most significantly with force and γ_1 with yank. Like in the cross-participant correlation analysis, band power correlated almost exclusively with force negatively. These negative correlations were observed with in both the yank and the ramp. MR indicated that the cumulation of those smaller individual correlations may help to build a more substantial relationship between the feature set and the state variable, especially in the force case. Typical trial-averaged feature traced (Figure 5.8) showedc very weak and below noise level deviation of the underlying signals from the power baseline.

3. Chirp and Ramp Decoding

Figure 5.9a reports the distribution performance of the decoders in the chirp trajectory condition. Overall tuned performance was low: no median performance exceeded $0.05r^2$. The single best decoding performance was achieved using the ENR regressor and was slightly superior to 0.1. An ANOVA (p-value < 0.05) confirmed that linear, non-parametric and non-linear regressors performed equally poorly. There was no difference between yank and force decoding, except in two instances: yank was superior for RR estimators, whereas, for SVR-RBF-PCA, force was superior to yank. Tuning (Figure 5.9b) still significantly impacted performance. A post hoc test (pairwise Tukey, p-value < 0.05) confirmed that only the RR regressor required significantly tuning. The default settings lead systematically to overly poor linear fits. Instead of relative overfit, an absolute overfit metric ($s_{train} - s_{validate}$) was employed to avoid division by zeros or negative numbers. An ANOVA (p-value < 0.05) confirmed that overfit only depended on the type of regressor employed (Figure 5.9c). Overfit was maintained low for linear estimators (RR, ENR and SVR). Although more important, post-hoc tests (pairwise, Tukey p-value < 0.05) showed that the GBTR-PCA was not overfitting significantly the data. On the other hand, this was not the case with the KNR and, in particular, the SVR-RBF-PCA. This regressor mostly was shaping noise and was incapable of generalising correctly.

Furthermore, decoders performed better when fitted on the ramp dataset (Figure 5.9). Again the RR did not lead to any viable solutions. The remaining linear and KNR regressors reached a median performance between 0.05 and 0.1 for force decoding, SVR-RBF-PCA, 0.2 and GBTR-PCA just under 0.5. An ANOVA (p-value < 0.05) confirmed that the choice of regressor was a determinant factor. Median yank decoding performance never exceeded 0.1. The gap between yank and force decoding was only significant in the GBTR-PCA and SVR-RBF-PCA (Wilcoxon p-value < 0.05). Again tuning helped increase performance in all instances (Wilcoxon p-value < 0.05), in particular, this increase was

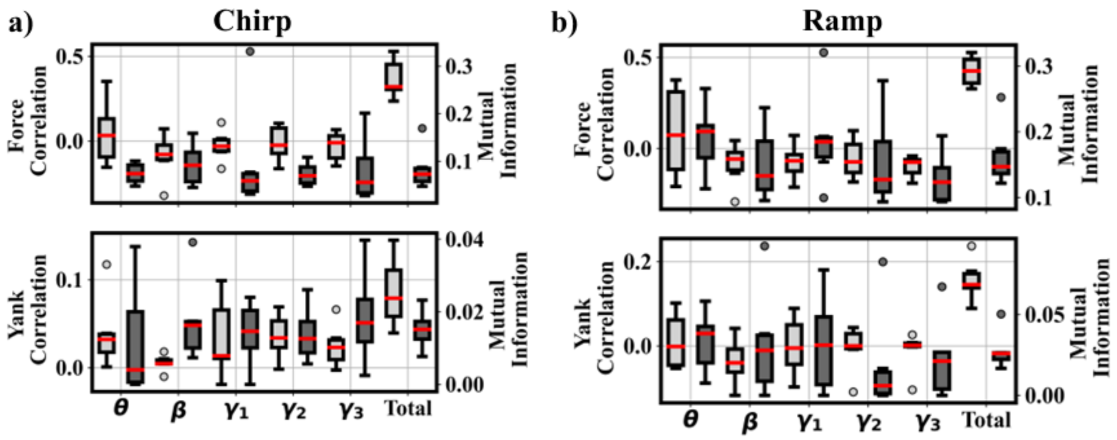


Figure 5.7 - Pre-screening results of the 'Gradient Recoding' dataset (Appendix A.C) for **a)** chirp and for **b)** ramps paradigm; Features were extracted using the STFT-band-CHS extractor using default parameters and modified band definition; An epoch length of 500ms ($n_o = 1$, for $f_d = 4$Hz) was used; Decoder output frequency was downgraded to 4Hz; Metric functions were applied to the aggregated trials; Individual feature correlations (**Dim Grey**) and MI (**Dark Grey**) are reported; Total metrics, multi-regression MR and average MI are also reported; Pre-screening was carried with force labels y and yank labels $\hat{y}$.

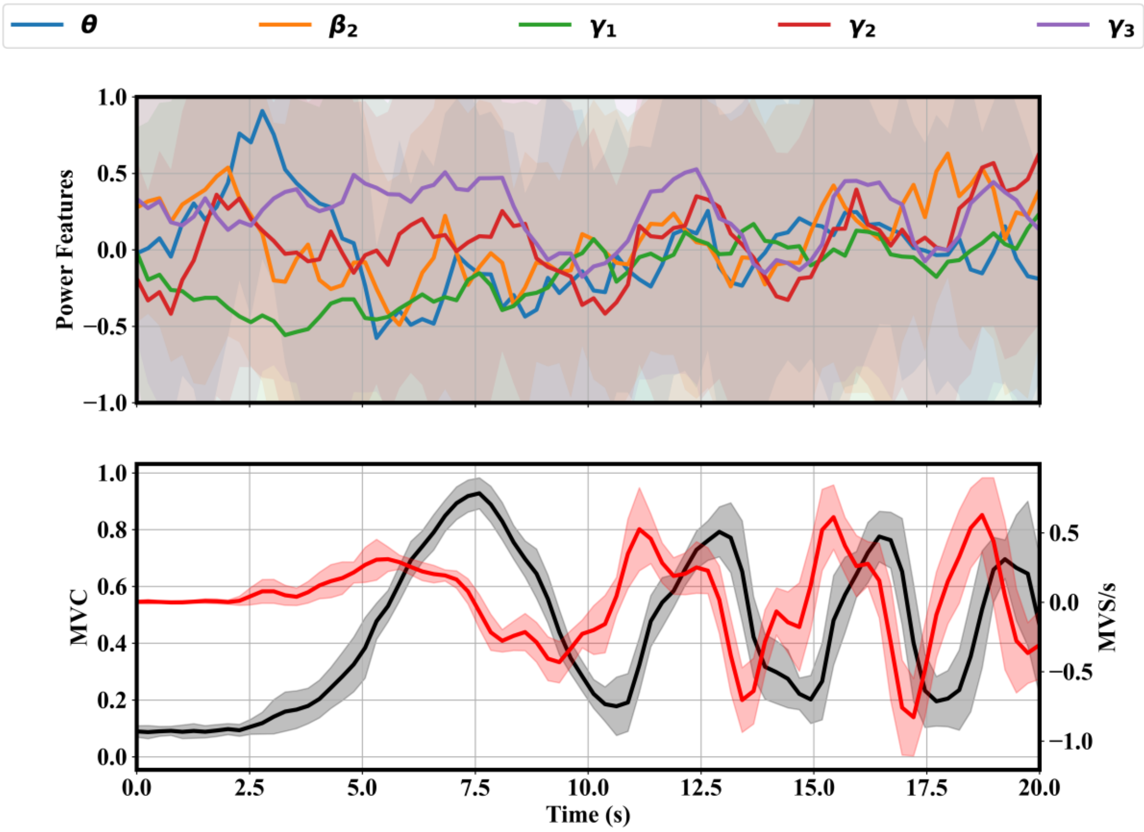


Figure 5.8 - Example of trial averages of power features for the chirp condition; Features were extracted using the STFT-band-CHS extractor using default parameters; An epoch length of 500ms ($n_o = 1$, for $f_d = 4$Hz) was used; Decoder output frequency was downgraded to 4Hz; 18 trials from Sub10L were averaged (Appendix A.C); Shade areas indicate one standard deviation from the trial average; (**Black**) line is the averaged tracked force; (**Red**) line is the averaged tracked yank.

significantly more for the RR regressor during force decoding (ANOVA, pairwise Tukey p-value 0.05). Overfitting was comparable for the most part to chirp decoding. KNR, GBTR-PCA and SVR-RBF-PCA tended to significantly more overfit the data than the linear regressors (ANOVA, pairwise Tukey p-value 0.05). The GBTR-PCA delivered most of the best individual participant performances (Figure 5.13a). Examples of decoded force ramp trajectories are presented in Figure 5.10 and of decoded yank trajectories in Figure 5.11, using this regressor. For force decoding, the regressor captured a single template trajectory that matches each ramp condition sufficiently. Although the decoder had some difficulties scaling the higher force level actions fully, it could still adopt the right ramp and plateau gradients when necessary. The absence of robust mappings between the feature space and the motor state led to over-regularisation and invariant decoder outputs for yank decoding. Deviations from the baseline were minimal, and scaling was excessively poor. Still, some small deviations coincided with the larger yank changes.

Figure 5.9d and h holds the parameter importance distributions for ramp and chirp decoding. Channel selection and feature extractor parameters, in particular the Box-Cox constant λ, the number of overlaps between STFT windows n_o and the size of the window length N_w were most important for linear models (RR, ENR and SVR). In contrast, those parameters were insignificant for the KNR and other non-linear models. Performance was then more dependent on regressor specific parameters. In particular, the tuning of the GBTR-PCA regressor only depended on one parameter, the learning rate η.

As in Chapter 4., the pre-screening metrics (average MI and MR) were correlated to the final performances. Non-significant relations were found using the MI-based metric, but the MR coefficient was predictive of some force decoding performance outcome, specifically for the GBTR-regressor.

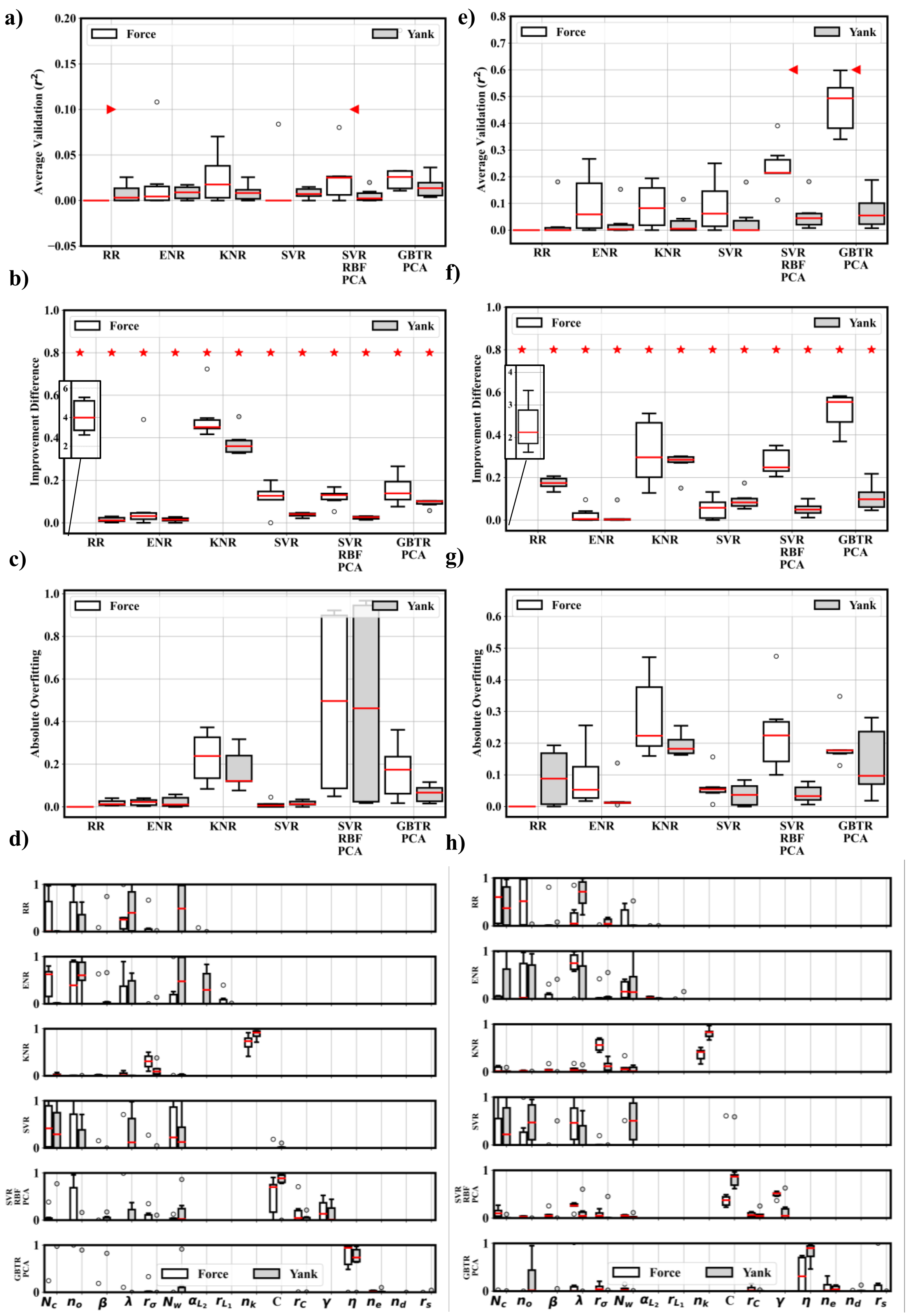


Figure 5.9 - Decoding results for chirp and ramp after BO tuning; **a)** Chirp average validation r^2distributions for all decoders, for yank and force decoding; (**Red >**) indicates significant difference (Wilcoxon test p-value < 0.05); direction of the triangle points towards the superior state; negative r^2 scores were set to zero; **b)** Chirp BO tuning improvement distributions; (*) indicates a significant difference (Wilxocon test p-value < 0.05); **c)** Chirp absolute overfit distributions; **d)** Chirp parameter importance distributions resulting from sensitivity analysis of BO surrogate models; **e)** Ramp average validation r^2distributions for all decoders, for yank and force decoding; **f)** Ramp BO tuning improvement distributions; **g)** Ramp absolute overfit distributions; **h)** Ramp parameter importance distributions.

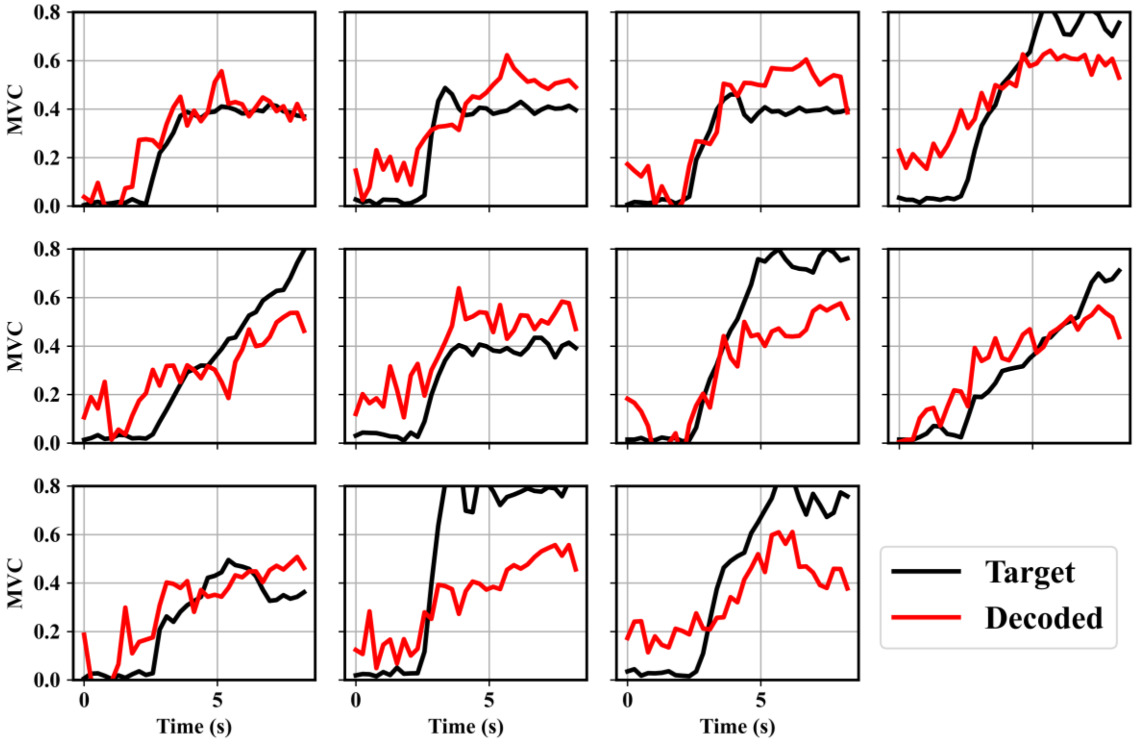


Figure 5.10 - Examples of decoded force ramp trajectories using the STFT-band-CHS extractor and GBTR-PCA regressor; Optimal parameters found with BO are used; Recording were made on participant Sub11R (Appendix A.C); Decoding is completed on a set of test trials, not use for BO tuning or estimator fitting.

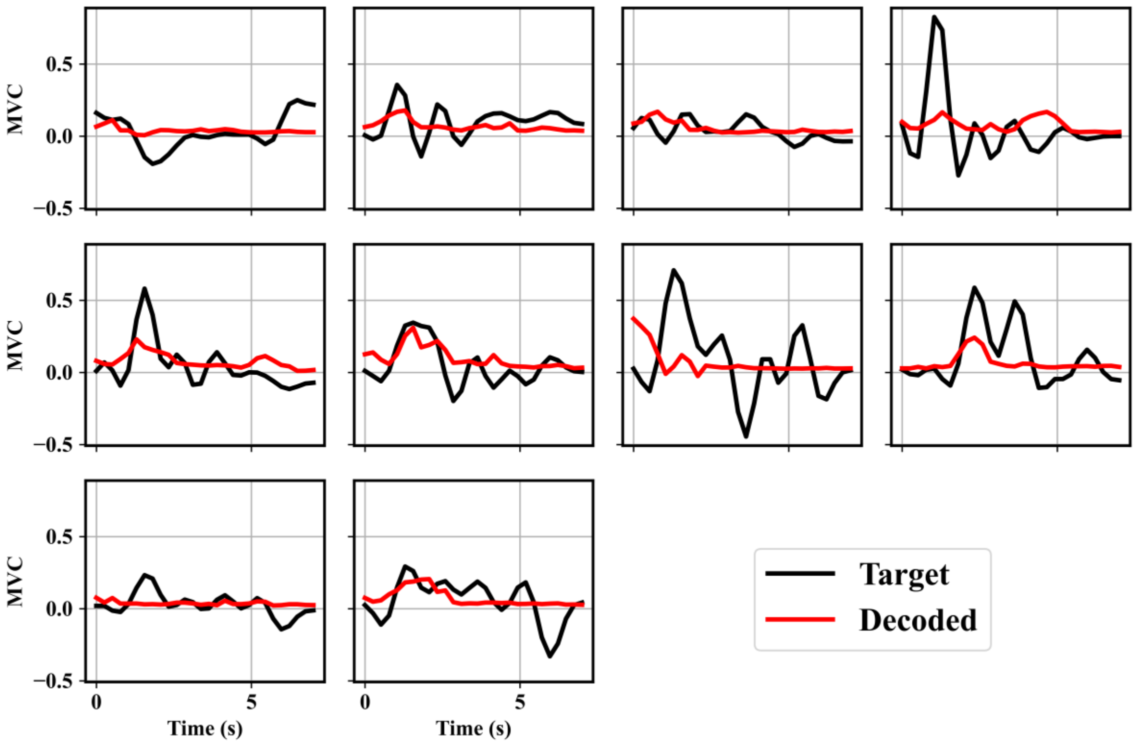


Figure 5.11 - Examples of decoded yank ramp trajectories using the STFT-band-CHS extractor and GBTR-PCA regressor; Optimal parameters found with BO are used; Recording were made on participant Sub11R (Appendix A.C); Decoding is completed on a set of test trials, not use for BO tuning or estimator fitting.

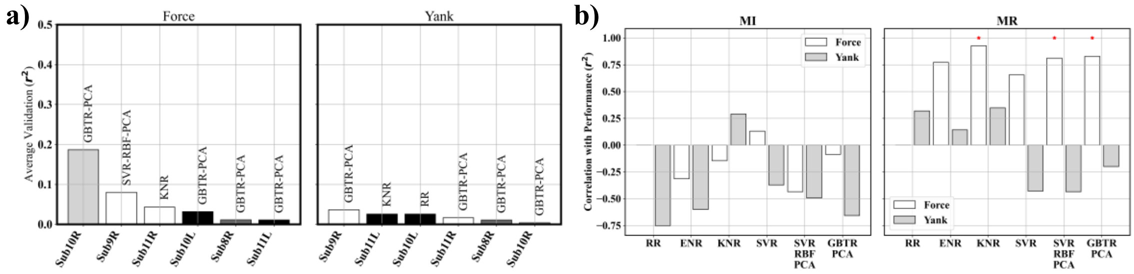


Figure 5.12 – Model selection; **a)** Best regressor for chirp decoding; **b)** Spearman correlation coefficient between feature pre-screening metric and final tuned performance; (*) indicates a significant relation (p-value < 0.05).

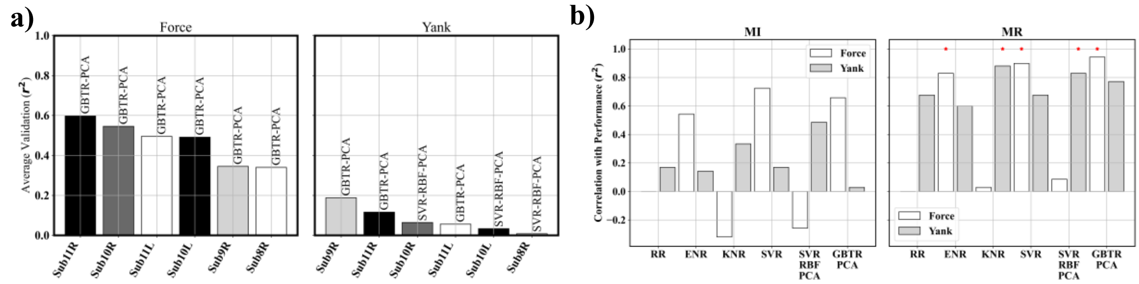


Figure 5.13 – Model selection; **a)** Best regressor for the ramp decoding b) Spearman correlation coefficient between feature pre-screening metric and final tuned performance; (*) indicates a significant relation (p-value < 0.05).

E. Discussion

The paradigm did help deepen the understanding of STN-LFP behaviours under variable task dynamics, and sufficient data were gathered to discuss, at least preliminary, the validity of hypothesis 1 through 4. Decoding those STN-LFP was, on the other hand, not as insightful as many models failed to fit the data. Non-linear models did help deliver some adequate performances. Although promising, the usage of such opaque models still requires caution and further investigation.

1. Insights in STN-LFP and Isometric Action Dynamics

Several classes of features could be distinguished. The ANOVA and trend results supported clearly that some power features, namely δ, θ, α and arguably β_1, were mostly impulsive, i.e., sensitive almost exclusively to yank application and thus satisfied hypothesis 1. However, all these biomarkers did not coincide with yank impulses in the same way. θ appeared to be in phase with the yank. This relation is consistent with BCI decoders which employ single-feature low-frequency to estimate dynamic motor states such as velocity (Ahmadi et al., 2019; Bradberry et al., 2010). In contrast, the temporal relation α appeared to include a temporal lead and was accentuated in more pronounced yank impulses. This relation was already observed in Tan et al. (2013). It could include a sensory-motor response to movement initiation once the cue is detected and the yank selected. β_1 could be further nuanced as a transition band exhibiting characteristics of both its α neighbour, notably its pre-action burst and β_2, its action ERD. It could also not be ruled out that the pre-action discharges could be pathologic in nature (Khawaldeh et al., 2020).

Furthermore, there was not any exclusively force coupled feature satisfying hypothesis 2. The best feature representative of force could have arguably been β_2. Compared to γ activity, the band sustained an ERS throughout most ramp and chirp actions and was sensitive to the force level, especially in the rising and holding phase of the ramp conditions. Nonetheless, even then β_2 still responded differently given different ramp gradients. The minor pre-action bursts and the leading positive correlation would indicate coupling of this band to initial yank onset, like with β_1 and possibly α. Results also showed that γ activity was strongly coupled to yank and also force but more subtly. For instance, the γ_2 ERD event observed prior to the chirp grip development (Figure 5.4, from t=0 to about t=5s) could be interpreted as volitional motor regularisation. By the time of the α pre-action burst, the initial motion of the target cursor should have been perceived and a motor plan selected. In this motor plan, γ_2 would be the marker of action inhibition, preventing an over-reaction to the cue change and over-shooting the target on its first slow initial ascension. γ overall activity also tended to build-up linearly with the acceleration of the chirp as if it was driving the consecutive release of faster actions. In the ramp tasks, γ_2 and γ_3 mean activity also varied in the holding phase, depending on the yank selected to reach said level. Therefore, dynamic modelling could be necessary to fully describe the relationship between STN-LFP mid and high-frequency features and motor states.

Evidence was gathered against hypothesis 3, especially in the ramp conditions. No significantly asymmetric activity patterns in the ramp condition were attributable to negative gradients. Given that correlations with yank were comparable to those with absolute yank, it is most likely that only positive gradients were encoded in the features. The distinction between positive and negative yank mattered less in the chirp condition. A linear temporal correction to the baseline could transform the chirp yank signal into a train of strictly positive impulses more easily mappable to activity burst. In any case, a non-linear strategy mapping from feature to yank is likely necessary.

The tracking error was more strongly correlated to the two motor states than any features. There was no further evidence that the tracking error could be driving the STN LFP and thus no reasons to reject hypothesis 4. Until a causal relation can be directly established, it appears sensible to maintain this clause at the very least as a working assumption. After all, neither the ramp nor the chirp paradigms were truly designed to test it.

2. Paradigm Limitations

Several caveats could be underlined. Again, the SNR of the LFP was a factor limiting the outcome of the analysis. Given the limited number and the heterogeneity of recordings, including more participants could further help reduce uncertainty surrounding the exact behaviour of the STN-LFP signals. The study only included 6 participant recordings which was below the mean sample size found in other studies (Chapter 2). The low SNR also made it challenging to measure correlation as well as γ activity reliably. Meanwhile, low frequencies, most notably δ and less so θ, were prone to mechanical artefacts. Activity patterns in those domains should be interpreted with more caution.

There were some un-avoidable biases in the ramp study. Because of their variable length, each rise and fall ramp section could introduce biases in the ANOVA. The only way to combine constant gradients with different force levels was to introduce variable ramp lengths across the different yank conditions. Although no design biases were identified with the chirp condition, some participants did find an unforeseen strategy to beat the tracking paradigm. As the tempo increased, the participants naturally increased their strength above 0.8, saturating the cursor signal to deliver more yank. Increased activity in the accelerated portion of the chirp trials could also be associated with some increased in force rather than only yank. This behavioural response further illustrated how difficult yank and force decoupling can be in practice.

Lastly, as described and employed in other studies (Kühn et al., 2004), pre-cues, signalling the onset of the target cursor movement, could have been employed to decouple activity patterns, notably in the α and β_1, domain associable with pre-action and action onset. It is finally worth mentioning, that, as discussed in Chapter 2.D, the STN-LFP most likely encodes effort rather than actual force. Given the isometric nature of the grip contractions and the rigid mechanical interface of the dynamometers, a sensible measure of action power was not feasible. Mechanical power is the product of the applied force and the contact interface rate of deformation (or spring contraction velocity). Power could allow revisiting hypothesis 3 to treat positive and negative power delivery differently. A compliant and variable stiffness interface would have to be developed to test such paradigms.

3. Continuous State Decoding from STN-LFP

The cross-participant paradigm analysis suggested that a dynamical model, including both regressions of force and yank, would be necessary for complete force decoding. The initial objective was to design a Kalman filter-based decoder. The decoder would be constructed on a state-space coupling of yank and force using a second-order mass-spring model such as the one proposed by Gupta et al. (2013). The observation space would then consist of single feature regressions from yank and force (Li et al., 2009). Unfortunately, this approach had to be pre-emptively abandoned, given the difficulties in fitting simple linear models to the data. Methods such as the unscented Kalman filter (Li et al., 2009) could have been employed to integrate non-linear models. However, the poor performance of yank decoding models could not justify pursuing such a complex approach.

Although BO overall helped increase performance, tuning above all else tended to produce over-regularised models because of the lack of a clear relationship between features and state variables. Feature prescreening and the paradigm analysis pointed to low feature SNR in individual trials as one potential problem hindering the robust fitting of linear models. The GBTR-PCA regressor seemed to have been able to work around this problem. Although the performance results are encouraging, they should be interpreted with caution. Non-linear models are non-transparent, making them difficult to interpret. The relation between MR and the GBTR-PCA would suggest that the regressor still exploits information buried within the feature space. What is most immediately warranted is acquiring more patient data, especially to try and find participants with strong overall γ ERS during motor actions (Tan et al., 2016) and re-assess the creation of informative linear models. Comparisons between force and yank decoding could then be more conclusive.

Moreover, the difference between chirp and ramp decoding could be due to the quantity of data employed to fit the regressors. The chirp condition only employed less than 20 trials, while at least more than 90 were employed for the ramp. Nevertheless, Merk et al. (2021) employed even more trials and did not see such improvement. Further refinement of the training set for those decoders is required, including positive and negative ramps and baseline segments to increase robustness. A transition to deep learning is possibly warranted as datasets significantly increase in size, and it would permit further comparison of non-linear dynamic regressors such as recurrent neural networks (Ahmadi et al., 2019). Cross-participant models could also be beneficial (Lawhern et al., 2018), as GBTR-PCA results increased the median performance of the entire sample and not just of a subset of participants.

F. Chapter Summary

This work contributes to the problem of continuous force decoding from STN-LFP recordings. Two paradigms imposing variable rates of change for grip force delivery were for the first time proposed for STN-LFP decoding. The force tracking paradigm decoupled force and yank (i.e. the force first derivative) using several ramp conditions with variable gradients and holding force levels. A chirp task was employed to test the gradual and continuous increase of the rate of force delivery in time. A cross-participant and offline analysis across a small sample of recordings converged with previous findings and helped further detail the behaviour of the STN-LFP. Lower frequency bands, δ to β_1 were exclusively responsive under variable yank regimes. β and γ bands have more subtle coupled dynamics with both yank and force levels.

Decoding STN-LFP signals under dynamic task constraints was very challenging. So far, attempts to create informative and transparent dynamic linear models have not been successful. Non-linear methods, however, permitted some successful decoding of ramp trajectories. Results suggest that force decoding remains superior to yank decoding, but more data gathering is required to draw definite conclusions.

Chapter 6. Conclusion

This chapter summarises the complete body of work produced for this thesis and reviews contributions and relevant findings. A discussion on the limits of STN-LFP decoding is started to assert prospective applications and possible future work avenues.

A. Thesis Summary

This work has proposed innovative solutions to design and assemble STN-LFP neuronal decoders, such as pipeline parameterisation and tuning. The adopted sequential fitting approach combined with BO has permitted to handle various decoding problems, both discrete and continuous, with small-scale machine learning trained on heterogeneous datasets and, lastly, to make relevant comparisons between different decoding algorithms. Results have shown that decoder tuning increased performance even with little prior knowledge of the problem. In cases where initial decoder parameters were poorly guessed, tuning could find functioning solutions. In others, the optimisation scheme still allowed for fine parameter tuning. The tuning process also provided indirect measures of decoder parameter relevance which can be employed to simplify the search space progressively. Combined with other metrics, it formed a set of design evaluation criteria that could be employed to re-iterate a decoder design for a given application. Ultimately, the sequential and BO tuning approach provides a unified and concise framework for more evolved and complex decoder structures (Figure 6.1), such as combined continuous and discrete detection with multiple conjoint tuning schemes for feature representation and performance optimisation.

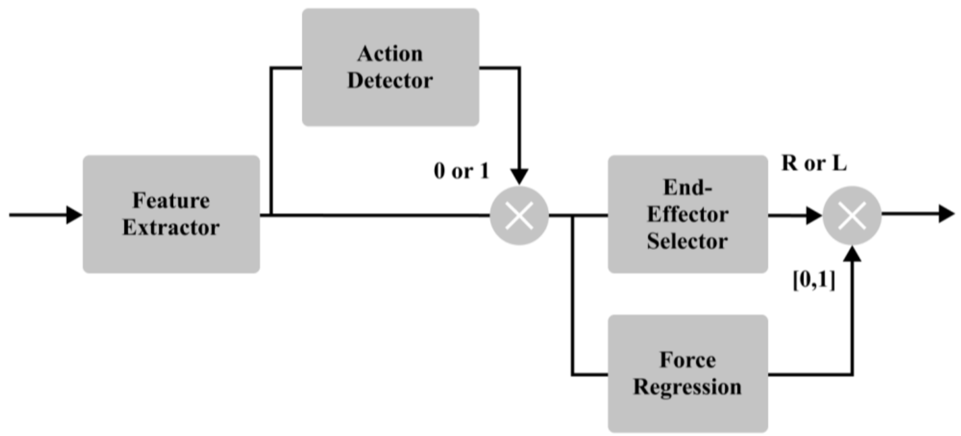


Figure 6.1 – Example of pipeline combining a single feature extractor and multiple decoders which are employed to construct a joint decoded state space; The first state estimator detects if an action is performed and gate information to subsequent estimators; A laterality classifier and force regressor estimate separately the end-effector and how much this end-effector is activated; Both pieces of information are then combined and transmitted to a device target.

In addition, this work has further contributed to the study of end-to-end backpropagation driven optimisation and argued why it should be considered to design even simple decoding algorithms. Although the provided results were insightful, the method was ill-suited for small datasets. In part, the optimisation of a large filter bank to improve the feature space representation was fundamentally inadequate. The method was still forward-looking as it should scale more favourably than other traditional machine learning techniques on larger datasets, with increased recording length and participant sample size. Deep-learning and end-to-end optimisation should also permit the creation of cross-user decoders that could help reduce performance heterogeneity (Golshan et al., 2020; Lawhern et al., 2018).

Moreover, this work demonstrated that STN-LFP decoding of discrete motor states could be achieved under asynchronous and real-time constraints. The result showed that action onset, holding and rest could be effectively detected even in noisy signal conditions. Evidence that asynchronous decoding was possible had been sparse. This work is also the first to report asynchronous laterality selection using two different approaches. Asynchronous decoding is key for control-oriented interfacing problems, particularly for closed-loop DBS. The task performance obtained fell within the range of what had been previously reported while being realistic and estimated for real-world deployment. Those also result from an extensive and unprecedented comparison of decoding algorithms.

Investigating new force tracking paradigms has elucidated past established relations between STN-LFP biomarkers and grip force generation in more detail. A cross-participant analysis has shown that lower STN-LFP frequencies bands are most sensitive to the instantaneous rate of change of force. β and γ have more complicated temporal dynamics, which vary depending on which force level is selected and how fast it is delivered. Continuous decoding of force and yank was performed with these new paradigms, but the resulting performances were significantly more limited than those obtained for discrete state decoding, as most of the regressor employed could not correctly fit the data. Certain non-linear regressors managed to overcome this problem in some instances and could map the force trend adopted in individual trials. Overall, the results appeared to have been superior for force decoding than for yank. More recordings are likely required to confirm this initial observation.

Despite the benefits of decoder pipeline optimisation and the test of multiple algorithmic combinations, performance gains have always reached a ceiling. This performance ceiling was different from one participant to the other due to the heterogeneity of the dataset, with examples of good and bad performances. After ruling out the algorithmic choice, the model complexity, the choice of feature representation, past evidence and current results suggest that the information contained in each STN-LFP recording is very variable. The disparity in information content leads to significant variability in extracted feature discriminability, which was later shown to affect performance. Methods proposed to alleviate this problem, notably by enhancing the quality of the feature space, have not been entirely satisfactory. No significant leverage on feature SNR and discriminability was found and acted upon by the BO and the backpropagation algorithm. Methods, such as the proposed rectified wavelet decomposition or curated random-kernel convolutions (Dempster et al., 2020), could potentially help increase discriminability, break individual performance ceilings, and close inter-subject performance gaps. However, even if successful alongside, for example, more advanced high performance deep learning methods, those would remain far from being conventional, indicating further that STN-LFP decoding remains non-trivial in practice. Lastly, although low feature SNR was somewhat manageable in the case of discrete decoding, evidence collected here and in at least one other study (Merk et al. 2021) showed that it could seriously impede the fitting of continuous state estimators.

B. Prospective Applications for STN-LFP Decoding

This work suggests that motor states can be inferred from STN-LFP signals under real-time constraints for controlling a peripheral device. However, before addressing which range of BCI applications could benefit from STN-LFP decoding, it is necessary to emphasise some of the limitations encountered.

1. Limitations

The low feature discriminability and SNR are currently the principal limiting factors encountered within this work and documented in past studies (Merk et al., 2021; Niketeghad et al., 2018). Even if technical solutions can help mitigate the effect of noise, these additional efforts may be in vain if there is not any significantly measurable neural-rhythm present in the recorded signals. Although offline studies demonstrated promising biomarker correlations, those can afford to rely upon trial-averaging and other offline technics to prompt even the weaker signals. This argument can be difficult to make, since DBS signals are invasively recorded, they can elicit a higher degree of enthusiasm and expectation. It is thus doubly important to formulate some hypothesis as to why this problem exists and how it could be overcome.

First, it is now known that synchrony sources activated in the STN during motor processes are very focal, and it is necessary to carefully choose the correct electrode contacts (Anzak et al., 2012; Tan et al., 2016, 2013). Although this work proposed and validated solutions to automatise this process, low-performance ceilings were always reached for some participants, even with the stronger signals selected. It could be possible that in those instances, the activity focal is not captured by the recording probe.

Certain LFP properties and the localisation of the source relative to the DBS electrodes may be related to this problem. Theoretical modelling by Lempka & McIntyre (2013) has suggested that the spherical reach of a given contact scales linearly with the distance to the probe and can be as large as several millimetres radially. For example, a reach field of 5mm could cover a volume up to approximately 525mm^3 which should be more than enough to encompass a sizeably large STN (Emmi et al., 2020). However, Lempka & McIntyre (2013) emphasised that the LFP strength is not due to individual or small clusters of cells. Their contribution should maintain a source relationship that is inversely proportional to the square of their distance from the electrode. Instead, the field is perpetuated by the sheer number of cells accumulated within the neighbouring volume. This scaling might be favourable in recording large hyper-synchronous ensembles caused by PD β deregulation (Pogosyan et al., 2010) or large movement, as hypothesised by Mace et al. (2014). Lempka & McIntyre (2013) and Maling et al. (2018) have further showed that these fields are bounded by the degree of local synaptic correlation or synchrony. This bound implies that the signal emanating from smaller local synchronous ensembles risks being buried by larger one. Lempka & McIntyre (2013) have lastly demonstrated that bipolar settings help contain the recording reach locally, suggesting that most of the stronger signals acquired in this study resulted from a source close to the contact pair and not from afar field.

Furthermore, the design of DBS electrodes may not be optimised for this problem, as it serves specifications associated with stimulation and not recording. It is well established that smaller electrodes should provide stronger signals (Lempka & McIntyre, 2013). Modelling by Kent & Grill (2014) has showed that reducing the lead diameter and ring contact length could help increase signal power and avoid spatial population averaging effects. More importantly, most DBS electrodes possessing ring contacts lack circumferential spatial resolutions. Recordings with high-density directional DBS electrodes have supported the hypothesis that synchronisation patterns are spatially distributed in the STN, most notably in the β range (Bour et al., 2015; S. Zhang et al., 2018), drawing further parallels with what had already been observed in the cortex (Heldman et al., 2006). High density and directional recording arrays are likely to enhance STN-LFP recordings (Bour et al., 2015; Khawaldeh et al., 2020; S. Zhang et al., 2018). It is still not clear whether they could capture preferred-synchronous directions emanating from the micro-structures of the STN (DeLong & Wichmann, 2010; Turner & Anderson, 1997). Experiments involving movement directionality could be highly insightful in this regard.

Secondly, the pathological effects of PD on STN electrophysiology should have been more carefully considered. Pathologic β hyper-synchrony can dominate the generation of LFP, permeate through the STN (Lempka & McIntyre, 2013; Pogosyan et al., 2010), and make the isolation of synchronisation sources more difficult. In particular, α and β bursts have been showed to significantly compromise the decoding of motion (Khawaldeh et al., 2020). The progression of PD can potentially aggravate the deregulation of those rhythms. A strong pathological β synchronisation can translate into weaker ERD. The feature most important to movement decoding is the pro-kinetic γ which is unfortunately suppressed because of the disease (Brown, 2007). An additional difficulty arises from the effects of medication which influence directly β and γ rhythms. The nature of the LFP neuro-dynamics should thus temporally change throughout the day with medication potency (Sand et al., 2021). On and off medication decoding models may have to be trained to respond to those changes. This problem goes against the initial arguments made which supported DBS for its longer-term recording reliability and should be further address for applications which require consistent longitudinal recording.

Thirdly, it may be worth considering alternative DBS targets to develop peripheral control interfaces and other BCI applications. DBS treatment is progressively extending in scope and is also becoming a standard of care to treat essential tremor (Lozano et al., 2019). In the latter case, DBS electrodes are implanted in the thalamus ventral intermediate nucleus, where pallidal and cerebellar output fibres are redirected towards the cortex. Performance obtained from decoding ventral intermediate recorded LFP has been very promising (He et al., 2021; Tan et al., 2019) and arguably superior to what has been accomplished in the STN. Although a thorough comparison needs to be completed, several advantages can be theorised. The thalamus is the final output of the CBGTC, through which a richer set of motor parameters could be conveyed to the cortex for final motor execution. Essential tremor does not induce any anti-kinetic β rhythm, which may interfere with signal acquisition and diminish the strength of pro-kinetic γ. More hypothetically, stronger synchronisation signals maybe easier to measure in thalamus. Pfurtscheller & Lopes da Silva, (1999) points to cortical-thalamic feedback loops, in particular thalamic relay cells, as partly responsible for generating cortical ERS/ERD. It could be, that the ventral intermediate target is closer in proximity to important local circuits driving synchronous activity throughout the CNS.

Fourthly, the decoding task required simple motor actions and assumed that the BG behaved almost exclusively as a feedforward amplifier organ, in essence, that its activity would remain invariant under deviation from the original motor plan. Again, the pinch and grip dynamometers are stable, predictable, and passive interfaces. Uncertain elements were purposefully not introduced in any of the visual feedback so that the participants would not have to adapt their internal motor plans based on error corrections. The BG is also an important feedback organ. Correction signals have been measured in the STN, specifically in the β band (Kühn et al., 2004; Tan et al., 2014), as well as in EEG (Kristeva et al., 2007). More broadly, sensorimotor β rhythms are though to be context specific and to depend on the task dynamics (Engel and Fries, 2010; Kristeva et al., 2007; Omlor et al., 2011). It is likely to behave in ways that have not initially been considered.

This last point raises a significant caveat applicable not exclusively to STN-LFP decoders but also to any BCI problems in general. Certain motor functions may overlap with the same electrophysiological phenomena exploited for decoding. Even in constrained experiments, during recording, participants, for instance, need to readjust their posture, move their heads naturally and tend to enjoy engaging in conversations. Although the latter can be essential to keep them motivated and see the experimental task through, speech and other movements will leave discernible signatures in the STN-LFP (Golshan et al., 2018). In time, there can also be a dissociation of the STN-LFP from the relationship initially established with the desired decoded state. These neuronal decoders could require extensive fitting across different dynamic motor tasks longitudinally in time with potential regular recalibration (Grado et al., 2018).

2. Proposed Applications

This work has shown that it is still possible to address some asynchronous interfacing control problems. Bulk movement detection from STN-LFP for closed-loop DBS should be possible, provided a gradual improvement in feature discriminability and source localisation. Gait cycle parameter decoding might be a promising avenue (Fischer et al., 2018; Tan et al., 2018). As suggested by Mace et al. (2014), gait, low limb-locomotion and postural control, turning, balancing, standing, and seating may engage many different muscle groups and CNS processes. Synchronous activity induced by gross motion should help exceed the minimal signal to noise threshold required for decoding and may also be sufficiently large to be systematically in reach of at least one recording contact. Wearable motion sensors could be employed for data labelling (Huo et al., 2020). DBS therapy has shortcomings in treating gait-related symptoms (Lozano et al., 2019). The resulting state prediction machine could inform assistive peripheral devices such as an exoskeleton (Grimmer et al., 2019). The exoskeleton function could help deliver extra assistive power to guide motion, prevent falls during gait-freezing events, regain balance, or restore standard gait patterns (Grimmer et al., 2019; Moreno and Pons, 2008).

Furthermore, STN-LFP decoding could be employed for other general-purpose applications. At first, the real-time and BCI asynchronous constraints could be relaxed. Robust offline processing can instead be employed to treat the signals in pre-defined analysis windows (Cassidy et al., 2002). Directional motor tasks should be the object of future studies.

An expected outcome of the laterality selection decoding models was the significant dissymmetric α and β rhythm modulations across hemispheres (Chapter 4.F) (Martineau et al., 2020). Those modulations correspond to those employed in EEG cursor control paradigms (Wolpaw and McFarland, 2004). In addition, these modulations are still present in STN-LFP during imagined tasks (Fischer et al., 2017) and are controllable by users with feedback training (He et al., 2019). First discrete state decoding could be employed to predict one of four directions of a 2D plane (up, down, left, right). The user could be asked to image the displacement of a centre cursor towards one targeted direction. Decoding would occur offline after the cue is given. The decoded state would then be displayed using a simple colour scheme to confirm either a correct or incorrect prediction. Similar trials would be repeated with decoder parameters updated iteratively to track changing modulation patterns resulting from neurofeedback. This paradigm would also give insights into activity related to error ERS/ERD in the LFPs after displaying the prediction cue (Tan et al., 2014). EEG comparison could also be employed for validation. Real-time asynchronous and continuous decoding would be progressively introduced depending on the outcomes of prior validation experiments. Training PD patients through such interactive tasks to regain some control over their internal β rhythms may prove to have long term therapeutic effects, which could also be studied (Daly & Wolpaw, 2008; He et al., 2019).

C. Future Work

Certain milestones still need to be reached to validate the proposed decoding strategies fully, and final decoding prototypes need to be thoroughly tested online on a new sample of DBS participants. The prescreening strategies can be employed as selection and exclusion criteria for this new study to rapidly assess whether individually recorded signal biomarkers could permit decoding and prioritise recording time among participants. Pseudo-real time constraints can also be simulated from recordings for system validation.

Moreover, work on a related collaborative project was also carried out. As previously discussed, it is believed that β rhythms in the cortex and, hypothetically, the STN, is dependent on the current rest state (i.e. no movements). β activity should be altered as the CNS compensates for any disruptive external forces and resist deviation from its holding state (Engel and Fries, 2010; Kristeva et al., 2007; Omlor et al., 2011). The CNS employs several strategies to maintain posture, such as mechanical impedance joint modulation (or the stiffening of limbs through co-contraction) (Burdet et al., 2006). Parallels were suggested with β hyper-synchronisation and its relation to specific PD symptoms such as Brakynesia and rigidity (Brown, 2007). It is possible that β activity could also be a biomarker in the volitional regulation of limb stiffness. Measuring such a relation would further deepen the understanding of PD electrophysiology and could permit the decoding of a new type of motor state important to the control of a robotic device.

Inducing measurable changes in limb stiffness requires the interaction with mechanical action and an unpredictable environment. A portable powered interface presented in Figure 6.2 was designed and constructed as part of this collaboration. The prototype device handle could deliver specific torque impulses at unpredictable time intervals to induce the appropriate motor response. Paradigms to measure the modified β response are still being tested on healthy participants using EEG measurements. Preliminary results indicate that a steady increase in the number of perturbations seems to induce a gradual increase of β activity which is then suppressed and reset when movement is required.

Future work, therefore, includes the completion of these dynamic task paradigms and their testing on DBS participants. The resulting recording would again over new opportunities for pursuing BCI research dedicated limb rigidity decoding.

D. Final Remarks

As with any other invasive implant technology, substantial challenges remain to be overcome and much more is left to be accomplished to make the more promising applications a reality. Deep brain neural decoders could empower a future generation of assistive medical devices and improve the quality of life of its recipients. This work has contributed to advancing toward that goal by further framing the control interfacing problem and inspire solutions to address it.

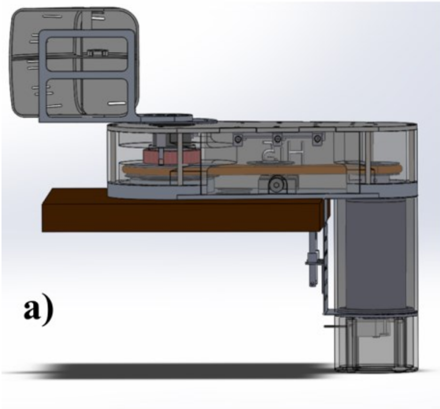
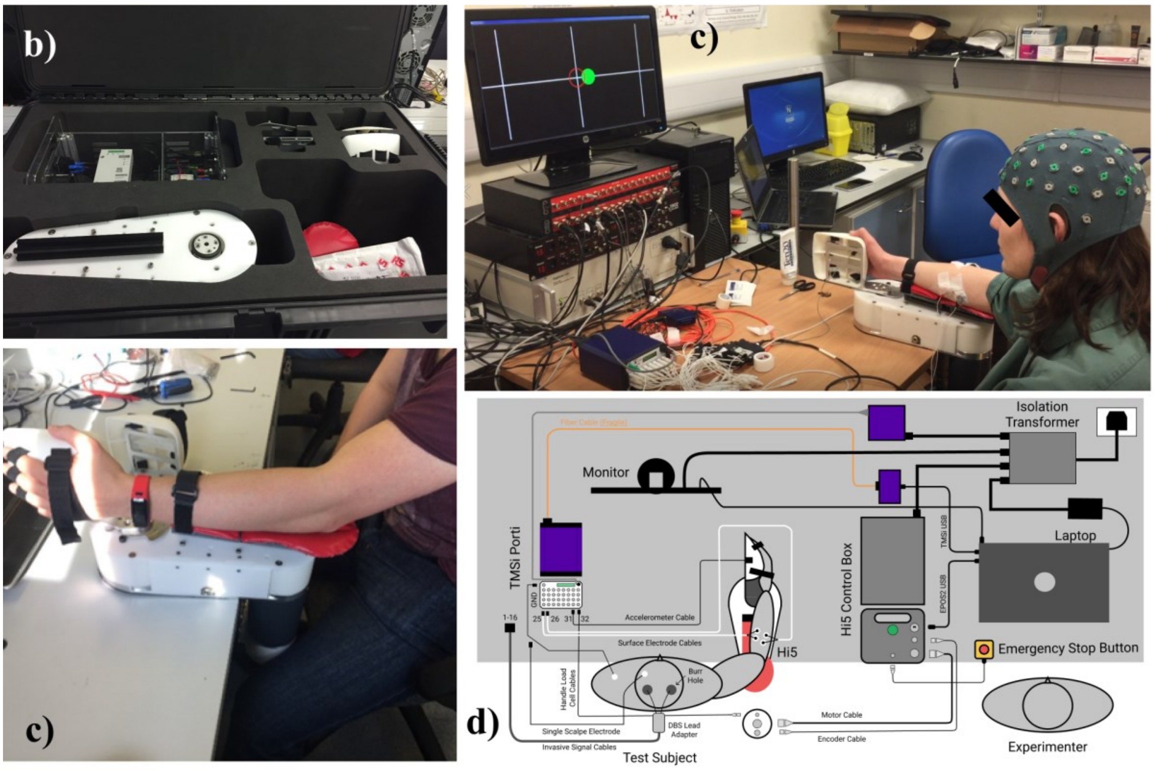


Figure 6.2 - Current ongoing work related to electrophysiological biomarkers measurement in motor interaction tasks with dynamically active and unpredictable environment; **a)** Model of the device which includes a haptic handle driven by a powerful motor through a direct belt drive; Handle torque is measured by a load cell; **b)** Disassembled kit for transport; **c)** Participant installation on the haptic device with their elbow and forearm placed a cushioned pad, hand and wrist secured with straps; **d)** Example of are recording EEG validation session where task instruction is validated on-screen; green marker indicates the current position of the handle; **e)** Detailed planned setup for DBS participant.



Bibliography

Abosch, A., Lanctin, D., Onaran, I., Eberly, L., Spaniol, M., Ince, N.F., 2012. Long-term recordings of local field potentials from implanted deep brain stimulation electrodes. Neurosurgery 71, 804–814. https://doi.org/10.1227/NEU.0b013e3182676b91

Adrian, E.D., Matthews, B.H.C., 1934. The Berger rhythm: potential changes from the occipital lobes in man. Brain 57, 355–385. https://doi.org/10.1093/brain/57.4.355

Ahmadi, N., Constandinou, T.G., Bouganis, C.S., 2019. Decoding Hand Kinematics from Local Field Potentials Using Long Short-Term Memory (LSTM) Network, in: Proceedings of the International IEEE/EMBS Conference on Neural Engineering (NER), San Francisco CA USA, 20-23 March 2019, IEEE. pp. 415–419. https://doi.org/10.1109/NER.2019.8717045

Anzak, A., Tan, H., Pogosyan, A., Foltynie, T., Limousin, P., Zrinzo, L., Hariz, M., Ashkan, K., Bogdanovic, M., Green, A.L., Aziz, T., Brown, P., 2012. Subthalamic nucleus activity optimizes maximal effort motor responses in Parkinson's disease. Brain 135, 2766–2778. https://doi.org/10.1093/brain/aws183

Avestruz, A.T., Santa, W., Carlson, D., Jensen, R., Stanslaski, S., Helfenstine, A., Denison, T., 2008. A 5 μw/channel spectral analysis IC for chronic bidirectional brainmachine interfaces. IEEE J Solid-State Circuits 43, 3006–3024. https://doi.org/10.1109/JSSC.2008.2006460

Aziz, T.Z., Nandi, D., Parkin, S., Liu, X., Giladi, N., Bain, P., Gregory, R.G., Joint, C., Scott, R.B., Stein, J.F., 2002. Targeting the subthalamic nucleus. Stereotact Funct Neurosurg 77, 87–90. https://doi.org/10.1159/000064602

Aziz, T.Z., Peggs, D., Sambrook, M.A., Crossman, A.R., 1991. Lesion of the subthalamic nucleus for the alleviation of 1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine (MPTP)-induced parkinsonism in the primate. Movement Disorders 6, 288–292. https://doi.org/10.1002/MDS.870060404

Bakstein, E., Burgess, J., Warwick, K., Ruiz, V., Aziz, T., Stein, J., 2012. Parkinsonian tremor identification with multiple local field potential feature classification. J Neurosci Methods 209, 320–330. https://doi.org/10.1016/j.jneumeth.2012.06.027

Baltadjieva, R., Giladi, N., Gruendlinger, L., Peretz, C., Hausdorff, J.M., 2006. Marked alterations in the gait timing and rhythmicity of patients with de novo Parkinson's disease. European Journal of Neuroscience 24, 1815–1820. https://doi.org/10.1111/j.1460-9568.2006.05033.x

Baranauskas, G., Maggiolini, E., Vato, A., Angotzi, G., Bonfanti, A., Zambra, G., Spinelli, A., Fadiga, L., 2012. Origins of 1/f 2 scaling in the power spectrum of intracortical local field potential. J Neurophysiol 107, 984–994. https://doi.org/10.1152/jn.00470.2011

Bashashati, H., Ward, R.K., Bashashati, A., 2016a. Bayesian optimization of BCI parameters, in: Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, Vancouver BC Canada, 15-18 May 2016, IEEE. IEEE, pp. 1–5. https://doi.org/10.1109/CCECE.2016.7726801

Bashashati, H., Ward, R.K., Bashashati, A., 2016b. User-customized brain computer interfaces using Bayesian optimization. J Neural Eng 13. https://doi.org/10.1088/1741-2560/13/2/026001

Bedard, C., Destexhe, A., 2012. Modelling local field potentials and their interaction with the extracellular medium, in: Brette, R., Destexhe, A. (Eds.), Handbook of Neural Activity Measurement. Cambridge University Press, pp. 136–188.

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. Lancet Neurol 18, 1112–1122. https://doi.org/10.1016/S1474-4422(19)30321-7

Benabid, A.L., Pollak, P., Hoffmann, D., Gervason, C., Hommel, M., Perret, J.E., de Rougemont, J., Gao, D.M., 1991. Long-term suppression of tremor by chronic stimulation of the ventral intermediate thalamic nucleus. The Lancet 337, 403–406. https://doi.org/10.1016/0140-6736(91)91175-T

Bergstra, J., Ca, J.B., Ca, Y.B., 2012. Random Search for Hyper-Parameter Optimization Yoshua Bengio, Journal of Machine Learning Research. https://doi.org/10.5555/2188385.2188395

Beric, A., Kelly, P.J., Rezai, A., Sterio, D., Mogilner, A., Zonenshayn, M., Kopell, B., 2002. Complications of deep brain stimulation surgery, in: Stereotactic and Functional Neurosurgery. Karger Publishers, pp. 73–78. https://doi.org/10.1159/000064600

Borton, D.A., Dawes, H.E., Worrell, G.A., Starr, P.A., Denison, T.J., 2020. Developing Collaborative Platforms to Advance Neurotechnology and Its Translation. Neuron 108, 286–301. https://doi.org/10.1016/j.neuron.2020.10.001

Bour, L.J., Lourens, M.A.J., Verhagen, R., De Bie, R.M.A., Van Den Munckhof, P., Schuurman, P.R., Contarino, M.F., 2015. Directional recording of subthalamic spectral power densities in Parkinson's disease and the effect of steering deep brain stimulation. Brain Stimul 8, 730–741. https://doi.org/10.1016/j.brs.2015.02.002

Bradberry, T.J., Gentili, R.J., Contreras-Vidal, J.L., 2010. Reconstructing three-dimensional hand movements from noninvasive electroencephalographic signals. Journal of Neuroscience 30, 3432–3437. https://doi.org/10.1523/JNEUROSCI.6107-09.2010

Brechet, L., Lucas, M.F., Doncarli, C., Farina, D., 2007. Compression of biomedical signals with mother wavelet optimization and best-basis wavelet packet selection. IEEE Trans Biomed Eng 54, 2186–2192. https://doi.org/10.1109/TBME.2007.896596

Brittain, J.S., Halliday, D.M., 2011. Optimal spectral tracking—Adapting to dynamic regime change. J Neurosci Methods 195, 111–115. https://doi.org/10.1016/J.JNEUMETH.2010.11.017

Brown, P., 2007. Abnormal oscillatory synchronisation in the motor system leads to impaired movement. Curr Opin Neurobiol. https://doi.org/10.1016/j.conb.2007.12.001

Brown, P., 2003. Oscillatory nature of human basal ganglia activity: Relationship to the pathophysiology of parkinson's disease. Movement Disorders 18, 357–363. https://doi.org/10.1002/mds.10358

Brown, P., 2000. Cortical drives to human muscle: the Piper and related rhythms. Prog Neurobiol 60, 97–108. https://doi.org/10.1016/S0301-0082(99)00029-5

Brown, P., Marsden, C., 1998. What do the basal ganglia do? The Lancet 351, 1801–1804. https://doi.org/10.1016/S0140-6736(97)11225-9

Burdet, E., Tee, K.P., Mareels, I., Milner, T.E., Chew, C.M., Franklin, D.W., Osu, R., Kawato, M., 2006. Stability and motor adaptation in human arm movements. Biol Cybern 94, 20–32. https://doi.org/10.1007/s00422-005-0025-9

Cagnan, H., Denison, T., McIntyre, C., Brown, P., 2019. Emerging technologies for improved deep brain stimulation. Nat Biotechnol 37, 1024–1033. https://doi.org/10.1038/s41587-019-0244-6

Cai, Y., Chow, M.Y., Lu, W., Li, L., 2010. Evaluation of distribution fault diagnosis algorithms using ROC curves. IEEE PES General Meeting, PES 2010 1–6. https://doi.org/10.1109/PES.2010.5588154

Camara, C., Isasi, P., Warwick, K., Ruiz, V., Aziz, T., Stein, J., Bakštein, E., 2015a. Resting tremor classification and detection in Parkinson's disease patients. Biomed Signal Process Control 16, 88–97. https://doi.org/10.1016/j.bspc.2014.09.006

Camara, C., Warwick, K., Bruña, R., Aziz, T., del Pozo, F., Maestú, F., 2015b. A Fuzzy Inference System for Closed-Loop Deep Brain Stimulation in Parkinson's Disease. J Med Syst 39. https://doi.org/10.1007/s10916-015-0328-x

Canessa, A., Palmisano, C., Isaias, I.U., Mazzoni, A., 2020. Gait-related frequency modulation of beta oscillatory activity in the subthalamic nucleus of parkinsonian patients. Brain Stimul 13, 1743–1752. https://doi.org/10.1016/j.brs.2020.09.006

Capogrosso, M., Milekovic, T., Borton, D., Wagner, F., Moraud, E.M., Mignardot, J.B., Buse, N., Gandar, J., Barraud, Q., Xing, D., Rey, E., Duis, S., Jianzhong, Y., Ko, W.K.D., Li, Q., Detemple, P., Denison, T., Micera, S., Bezard, E., Bloch, J., Courtine, G., 2016. A brain-spine interface alleviating gait deficits after spinal cord injury in primates. Nature 539, 284–288. https://doi.org/10.1038/nature20118

Cassidy, M., Mazzone, P., Oliviero, A., Insola, A., Tonali, P., Di Lazzaro, V., Brown, P., 2002. Movement-related changes in synchronization in the human basal ganglia. Brain 125, 1235–1246. https://doi.org/10.1093/brain/awf135

Chakravarthy, V.S., Joseph, D., Bapi, R.S., 2010. What do the basal ganglia do? A modeling perspective. Biol Cybern. https://doi.org/10.1007/s00422-010-0401-y

Chen, Y., Gong, C., Hao, H., Guo, Y., Xu, S., Zhang, Y., Yin, G., Cao, X., Yang, A., Meng, F., Ye, J., Liu, H., Zhang, J., Sui, Y., Li, L., 2019. Automatic Sleep Stage Classification Based on Subthalamic Local Field Potentials HHS Public Access. IEEE Trans Neural Syst Rehabil Eng 27, 118–128. https://doi.org/10.1109/TNSRE

Clinical commissioning policy : deep brain stimulation (DBS) in movement disorders, 2013. . NHS England.

Cohen, J., Cohen, P., West, S.G., Aiken, L., 2003. Multiple Regression/Correlation with k Independent Variables, in: Applied Multiple Regression: Correlation Analysis for the Behavioral Sciences. Taylor and Francis.

Cohen, M.X., 2018. A better way to define and describe Morlet wavelets for time-frequency analysis. Neuroimage 199, 81–86. https://doi.org/10.1101/397182

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. The Lancet 381, 557–564. https://doi.org/10.1016/S0140-6736(12)61816-9

Daly, J.J., Wolpaw, J.R., 2008. Brain-computer interfaces in neurological rehabilitation. Lancet Neurol 7, 1032–1043. https://doi.org/10.1016/S1474-4422(08)70223-0

DeLong, M., Wichmann, T., 2010. Changing views of basal ganglia circuits and circuit disorders. Clin EEG Neurosci 41, 61–7. https://doi.org/10.1126/scisignal.2001449

Dempster, A., Petitjean, F., Webb, G.I., 2020. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min Knowl Discov 34, 1454–1495. https://doi.org/10.1007/s10618-020-00701-z

Destexhe, A., Bedard, C., 2013. Local field potential [WWW Document]. Scholarpedia. https://doi.org/10.4249/scholarpedia.10713

Ding, Y., Galiana, I., Asbeck, A., Quinlivan, B., De Rossi, S.M.M., Walsh, C., 2014. Multi-joint actuation platform for lower extremity soft exosuits. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong China, 29 September 2019, IEEE 1327–1334. https://doi.org/10.1109/ICRA.2014.6907024

Dosen, S., Muceli, S., Dideriksen, J.L., Romero, J.P., Rocon, E., Pons, J., Farina, D., 2015. Online tremor suppression using electromyography and low-level electrical stimulation. IEEE Transactions on Neural Systems and Rehabilitation Engineering 23, 385–395. https://doi.org/10.1109/TNSRE.2014.2328296

Doya, B.K., Kimura, H., Kawato, M., 2001. Neural Mechanisms of Learning and Control. IEEE Control Systems Magazine 21, 42–54.

Doya, K., 2000. Complementary roles of basal ganglia and cerebellum in learning and motor control. Curr Opin Neurobiol 10, 732–739. https://doi.org/10.1016/S0959-4388(00)00153-7

Dunovan, K., Lynch, B., Molesworth, T., Verstynen, T., 2015. Competing basal ganglia pathways determine the difference between stopping and deciding not to go. Elife. https://doi.org/10.7554/ELIFE.08723

Emmi, A., Antonini, A., Macchi, V., Porzionato, A., De Caro, R., 2020. Anatomy and Connectivity of the Subthalamic Nucleus in Humans and Non-human Primates. Front Neuroanat 14. https://doi.org/10.3389/fnana.2020.00013

Engel, A.K., Fries, P., 2010. Beta-band oscillations-signalling the status quo? Curr Opin Neurobiol 20, 156–165. https://doi.org/10.1016/j.conb.2010.02.015

Falzon, O., Camilleri, K.P., Muscat, J., 2010. Complex-valued spatial filters for task discrimination, in: Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Buenos Aires Argentina, 31 August - 4 September 2010, IEEE. IEEE, pp. 4707–4710. https://doi.org/10.1109/IEMBS.2010.5626381

Farina, D., do Nascimento, O.F., Lucas, M.-F., Doncarli, C., 2007. Optimization of wavelets for classification of movement-related cortical potentials generated by variation of force-related parameters. J Neurosci Methods 162, 357–363. https://doi.org/10.1016/J.JNEUMETH.2007.01.011

Feliot, P., Bect, J., Vazquez, E., 2017. A Bayesian approach to constrained single- and multi-objective optimization. Journal of Global Optimization 67, 97–133. https://doi.org/10.1007/s10898-016-0427-3

Ferrari, A., Ginis, P., Nieuwboe, A., Greenlaw, R., Muddiman, A., Chiari, L., 2016. Handling gait impairments of persons with Parkinson's disease by means of real-time biofeedback in a daily life environment, in: International Conference on Smart Homes and Health Telematics (ICOST)), Wuhan China, 25-27 May 2016, Springer. Springer Verlag, pp. 250–261. https://doi.org/10.1007/978-3-319-39601-9_22

Fischer, P., Chen, C.C., Chang, Y.J., Yeh, C.H., Pogosyan, A., Herz, D.M., Cheeran, B., Green, A.L., Aziz, T.Z., Hyam, J., Little, S., Foltynie, T., Limousin, P., Zrinzo, L., Hasegawa, H., Samuel, M., Ashkan, K., Brown, P., Tan, H., 2018. Alternating modulation of subthalamic nucleus beta oscillations during stepping. Journal of Neuroscience 38, 5111–5121. https://doi.org/10.1523/JNEUROSCI.3596-17.2018

Fischer, P., Pogosyan, A., Cheeran, B., Green, A.L., Aziz, T.Z., Hyam, J., Little, S., Foltynie, T., Limousin, P., Zrinzo, L., Hariz, M., Samuel, M., Ashkan, K., Brown, P., Tan, H., 2017. Subthalamic nucleus beta and gamma activity is modulated depending on the level of imagined grip force. Exp Neurol 293, 53–61. https://doi.org/10.1016/J.EXPNEUROL.2017.03.015

Flash, T., Hogan, N., 1985. The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model. The Journal of Neuroscience 5, 1688–1703.

Flesher, S.N., Collinger, J.L., Foldes, S.T., Weiss, J.M., Downey, J.E., Tyler-Kabara, E.C., Bensmaia, S.J., Schwartz, A.B., Boninger, M.L., Gaunt, R.A., 2016. Intracortical microstimulation of human somatosensory cortex. Sci Transl Med 8, 1–11. https://doi.org/10.1126/scitranslmed.aaf8083

Flint, R.D., Lindberg, E.W., Jordan, L.R., Miller, L.E., Slutzky, M.W., 2012. Accurate decoding of reaching movements from field potentials in the absence of spikes. J Neural Eng 9. https://doi.org/10.1088/1741-2560/9/4/046006

Foffani, G., Bianchi, A.M., Baselli, G., Priori, A., 2005. Movement-related frequency modulation of beta oscillatory activity in the human subthalamic nucleus. Journal of Physiology 568, 699–711. https://doi.org/10.1113/jphysiol.2005.089722

Foffani, G., Bianchi, A.M., Priori, A., Baselli, G., 2004. Adaptive autoregressive identification with spectral power decomposition for studying movement-related activity in scalp EEG signals and basal ganglia local field potentials. J Neural Eng 1, 165–173. https://doi.org/10.1088/1741-2560/1/3/006

Frazier, P.I., 2018. A Tutorial on Bayesian Optimization.

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. Ann Stat 29, 1189–1232. https://doi.org/10.1214/aos/1013203451

Fukunaga, K., 1990. Feature Extraction and Linear Mapping for Classification, in: Introduction to Statistical Pattern Recognition. Academic Press, pp. 441–506.

Gagnon-Turcotte, G., Keramidis, I., Ethier, C., De Koninck, Y., Gosselin, B., 2019. A Wireless Electro-Optic Headstage with a 0.13-μm CMOS Custom Integrated DWT Neural Signal Decoder for Closed-Loop Optogenetics. IEEE Trans Biomed Circuits Syst 13, 1036–1051. https://doi.org/10.1109/TBCAS.2019.2930498

Gardner, J., 2013. A history of deep brain stimulation: Technological innovation and the role of clinical assessment tools. Soc Stud Sci 43, 707–728. https://doi.org/10.1177/0306312713483678

Georgopoulos, A.P., DeLong, M.R., Crutcher, M.D., 1983. Relations between parameters of step-tracking movements and single cell discharge in the globus pallidus and subthalamic nucleus of the behaving monkey. J Neurosci 3, 1586–1598.

Georgopoulos, A.P., Schwartz, A.B., Kettner, R.E., 1986. Neuronal population coding of movement direction. Science (1979) 233, 1416–1419. https://doi.org/10.1126/science.3749885

Ghimatgar, H., Kazemi, K., Helfroush, M.S., Aarabi, A., 2019. An automatic single-channel EEG-based sleep stage scoring method based on hidden Markov Model. J Neurosci Methods 324. https://doi.org/10.1016/j.jneumeth.2019.108320

Gianfelici, F., Biagetti, G., Crippa, P., Turchetti, C., 2007. Multicomponent AM-FM representations: An asymptotically exact approach. IEEE Trans Audio Speech Lang Process 15, 823–837. https://doi.org/10.1109/TASL.2006.889744

Golshan, H.M., Hebb, A.O., Hanrahan, S.J., Nedrud, J., Mahoor, M.H., 2018. A hierarchical structure for human behavior classification using STN local field potentials. J Neurosci Methods 293, 254–263. https://doi.org/10.1016/j.jneumeth.2017.10.001

Golshan, H.M., Hebb, A.O., Hanrahan, S.J., Nedrud, J., Mahoor, M.H., 2017. An FFT-based synchronization approach to recognize human behaviors using STN-LFP signal, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans LA USA, 5-9 March 2017, IEEE. pp. 979–983. https://doi.org/10.1109/ICASSP.2017.7952302

Golshan, H.M., Hebb, A.O., Hanrahan, S.J., Nedrud, J., Mahoor, M.H., 2016. A Multiple Kernel Learning approach for human behavioral task classification using STN-LFP signal, in: Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Orlando FL USA, 16-20 August 2016, IEEE. IEEE, pp. 1030–1033. https://doi.org/10.1109/EMBC.2016.7590878

Golshan, H.M., Hebb, A.O., Mahoor, M.H., 2020. LFP-Net: A Deep Learning Framework to Recognize Human Behavioral Activities using Brain STN-LFP Signals. J Neurosci Methods 335, 108621. https://doi.org/10.1016/j.jneumeth.2020.108621

Grado, L.L., Johnson, M.D., Netoff, T.I., 2018. Bayesian adaptive dual control of deep brain stimulation in a computational model of Parkinson's disease. PLoS Comput Biol. https://doi.org/10.1371/journal.pcbi.1006606

Grado, L.L., Johnson, M.D., Netoff, T.I., 2017. The Sliding Windowed Infinite Fourier Transform. IEEE Signal Process Mag 34, 183–188. https://doi.org/10.1109/MSP.2017.2718039

Grimmer, M., Riener, R., Walsh, C.J., Seyfarth, A., 2019. Mobility related physical and functional losses due to aging and disease - a motivation for lower limb exoskeletons. J Neuroeng Rehabil 16. https://doi.org/10.1186/S12984-018-0458-8

Gruenwald, J., Kapeller, C., Kamada, K., Scharinger, J., Guger, C., 2017. Optimal bandpower estimation and tracking via Kalman filtering for real-time Brain-Computer Interfaces, in: Proceedings of the International IEEE/EMBS Conference on Neural Engineering (NER), Shanghai China 25-28 May 2017, IEEE. pp. 605–608. https://doi.org/10.1109/NER.2017.8008424

Gruenwald, J., Znobishchev, A., Kapeller, C., Kamada, K., Scharinger, J., Guger, C., 2019. Time-variant linear discriminant analysis improves hand gesture and finger movement decoding for invasive brain-computer interfaces. Front Neurosci 13. https://doi.org/10.3389/fnins.2019.00901

Gupta, A., Balasubramani, P.P., Chakravarthy, V.S., 2013. Computational model of precision grip in Parkinson's disease: a utility based approach. Front Comput Neurosci 7, 172. https://doi.org/10.3389/fncom.2013.00172

Han, M., Liu, X., 2013. Feature selection techniques with class separability for multivariate time series. Neurocomputing 110, 29–34. https://doi.org/10.1016/j.neucom.2012.12.006

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

He, S., Baig, F., Mostofi, A., Pogosyan, A., Debarros, J., Green, A.L., Aziz, T.Z., Pereira, E., Brown, P., Tan, H., 2021. Closed-Loop Deep Brain Stimulation for Essential Tremor Based on Thalamic Local Field Potentials. Movement Disorders 36, 863–873. https://doi.org/10.1002/mds.28513

He, S., Debarros, J., Khawaldeh, S., Pogosyan, A., Mostofi, A., Baig, F., Pereira, E., Brown, P., Tan, H., 2020. Closed-loop DBS triggered by real-time movement and tremor decoding based on thalamic LFPs for essential tremor, in: Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Montreal QC Canada, 20-24 July 2020, IEEE. pp. 3602–3605. https://doi.org/10.1109/EMBC44109.2020.9175433

He, S., Syed, E., Torrecillos, F., Tinkhauser, G., Fischer, P., Pogosyan, A., Pereira, E., Ashkan, K., Hasegawa, H., Brown, P., Tan, H., 2019. Beta Oscillation-Targeted Neurofeedback Training Based on Subthalamic LFPs in Parkinsonian Patients, in: International IEEE/EMBS Conference on Neural Engineering (NER), San Francisco CA USA, 20-23 March 2019, IEEE. pp. 81–84. https://doi.org/10.1109/NER.2019.8717176

He, Y., Eguren, D., Azorín, J.M., Grossman, R.G., Luu, T.P., Contreras-Vidal, J.L., 2018. Brain–machine interfaces for controlling lower-limb powered robotic systems. J Neural Eng 15, 021004. https://doi.org/10.1088/1741-2552/aaa8c0

Head, T., Kumar, M., Nahrstaedt, H., Louppe, G., Shcherbatyi, I., 2021. scikit-optimize/scikit-optimize. https://doi.org/10.5281/ZENODO.5565057

Heldman, D.A., Wang, W., Chan, S.S., Moran, D.W., 2006. Local field potential spectral tuning in motor cortex during reaching. IEEE Transactions on Neural Systems and Rehabilitation Engineering 14, 180–183. https://doi.org/10.1109/TNSRE.2006.875549

Herman, J., Usher, W., 2017. SALib: An open-source Python library for sensitivity analysis. Journal of Open Source Software. The Journal of Open Source Software 2. https://doi.org/10.21105/joss.00097

Herreras, O., 2016. Local Field Potentials: Myths and Misunderstandings. Front Neural Circuits 10. https://doi.org/10.3389/fncir.2016.00101

Hirschmann, J., Schoffelen, J.M., Schnitzler, A., van Gerven, M.A.J., 2017. Parkinsonian rest tremor can be detected accurately based on neuronal oscillations recorded from the subthalamic nucleus. Clinical Neurophysiology 128, 2029–2036. https://doi.org/10.1016/j.clinph.2017.07.419

Hsieh, H.L., Shanechi, M.M., 2016. Multiscale brain-machine interface decoders, in: Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 16-20 August 2016, Orlando FL USA, IEEE. IEEE, pp. 6361–6364. https://doi.org/10.1109/EMBC.2016.7592183

Hsieh, H.-L., Wong, Y.T., Pesaran, B., Shanechi, M.M., 2017. Multiscale Decoding for Reliable Brain-Machine Interface Performance Over Time. https://doi.org/10.0/Linux-x86_64

Huo, W., Angeles, P., Tai, Y.F., Pavese, N., Wilson, S., Hu, M.T., Vaidyanathan, R., 2020. A Heterogeneous Sensing Suite for Multisymptom Quantification of Parkinson's Disease. IEEE Transactions on Neural Systems and Rehabilitation Engineering 28, 1397–1406. https://doi.org/10.1109/TNSRE.2020.2978197

Islam, M.S., Khan, M.S., Deng, H., Mamun, K.A., 2014. Decoding movements from human deep brain local field potentials using radial basis function neural network, in: IEEE Symposium on Computer-Based Medical Systems, New York NY USA, 27-29 May 2014, IEEE. IEEE, pp. 105–108. https://doi.org/10.1109/CBMS.2014.77

Islam, M.S., Mamun, K.A., Deng, H., 2017. Decoding of human movements based on deep brain local field potentials using ensemble neural networks. Comput Intell Neurosci 2017, 1–16. https://doi.org/10.1155/2017/5151895

Izhikevich, E., 2007. Synchronization, in: Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. The MIT Press.

Jameel, A.S.M.M., Mamun, K.A. Al, 2017. An algorithm to decode movement and laterality from Deep Brain Local Field Potentials utilizing time and frequency domain features, in: IEEE Region Conference (TENCON), Singapore, 22-25 November 2016, IEEE. IEEE, pp. 409–413. https://doi.org/10.1109/TENCON.2016.7848031

Jamieson, K., Talwalkar, A., 2016. Non-stochastic best arm identification and hyperparameter optimization. International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz Spain, May 9-11 2016, Journal of Machine Learning Research 240–248.

Jenkinson, N., Kühn, A.A., Brown, P., 2013. Gamma oscillations in the human basal ganglia. Exp Neurol 245, 72–76. https://doi.org/10.1016/J.EXPNEUROL.2012.07.005

Johansson, R.S., Cole, K.J., 1992. Sensory-motor coordination during grasping and manipulative actions. Curr Opin Neurobiol 2, 815–823. https://doi.org/10.1016/0959-4388(92)90139-C

Kandel, E.R., Schwartz, J.H., Jessell, T.M., 2012a. The Basal Ganglia, in: Principal of Neural Science.

Kandel, E.R., Schwartz, J.H., Jessell, T.M., 2012b. Voluntary Movement: The Primary Motor Cortex, in: Principal of Neural Science.

Kapeller, C., Ogawa, H., Schalk, G., Kunii, N., Coon, W.G., Scharinger, J., Guger, C., Kamada, K., 2018. Real-time detection and discrimination of visual perception using electrocorticographic signals. J Neural Eng 15. https://doi.org/10.1088/1741-2552/aaa9f6

Kapeller, C., Schneider, C., Kamada, K., Ogawa, H., Kunii, N., Ortner, R., Pruckl, R., Guger, C., 2014. Single trial detection of hand poses in human ECoG using CSP based feature extraction, in: Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 26-30 August 2014, Chicago IL USA, IEEE. pp. 4599–4602. https://doi.org/10.1109/EMBC.2014.6944648

Kenney, C., Simpson, R., Hunter, C., Ondo, W., Almaguer, M., Davidson, A., Jankovic, J., 2007. Short-term and long-term safety of deep brain stimulation in the treatment of movement disorders. J Neurosurg 106, 621–625. https://doi.org/10.3171/jns.2007.106.4.621

Kent, A.R., Grill, W.M., 2014. Analysis of deep brain stimulation electrode characteristics for neural recording. J Neural Eng 11. https://doi.org/10.1088/1741-2560/11/4/046010

Khan, M.E., Dutt, D.N., 2007. An expectation-maximization algorithm based Kalman smoother approach for event-related desynchronization (ERD) estimation from EEG. IEEE Trans Biomed Eng 54, 1191–1198. https://doi.org/10.1109/TBME.2007.894827

Khawaldeh, S., Tinkhauser, G., Shah, S.A., Peterman, K., Debove, I., Khoa Nguyen, T.A., Nowacki, A., Lenard Lachenmayer, M., Schuepbach, M., Pollo, C., Krack, P., Woolrich, M., Brown, P., 2020. Subthalamic nucleus activity dynamics and limb movement prediction in Parkinson's disease. Brain 143, 582–586. https://doi.org/10.1093/brain/awz417

Korik, A., Sosnik, R., Siddique, N., Coyle, D., 2018. Decoding imagined 3D hand movement trajectories from EEG: Evidence to support the use of mu, beta, and low gamma oscillations. Front Neurosci 12, 1–16. https://doi.org/10.3389/fnins.2018.00130

Korik, A., Sosnik, R., Siddique, N., Coyle, D., 2016. 3D hand motion trajectory prediction from EEG mu and beta bandpower. Prog Brain Res 228, 71–105. https://doi.org/10.1016/BS.PBR.2016.05.001

Kornhuber, H.H., 1971. Motor functions of cerebellum and basal ganglia: the cerebellocortical saccadic (ballistic) clock, the cerebellonuclear hold regulator, and the basal ganglia ramp (voluntary speed smooth movement) generator. Kybernetik 8, 157–162. https://doi.org/10.1007/BF00290561

Kothe, C.A., Makeig, S., 2013. BCILAB: a platform for brain–computer interface development. J Neural Eng 10, 056014. https://doi.org/10.1088/1741-2560/10/5/056014

Kristeva, R., Patino, L., Omlor, W., 2007. Beta-range cortical motor spectral power and corticomuscular coherence as a mechanism for effective corticospinal interaction during steady-state motor output. Neuroimage. https://doi.org/10.1016/j.neuroimage.2007.03.025

Kühn, A.A., Tsui, A., Aziz, T., Ray, N., Brücke, C., Kupsch, A., Schneider, G.-H., Brown, P., 2009. Pathological synchronisation in the subthalamic nucleus of patients with Parkinson's disease relates to both bradykinesia and rigidity. Exp Neurol 215, 380–387. https://doi.org/10.1016/j.expneurol.2008.11.008

Kühn, A.A., Williams, D., Kupsch, A., Limousin, P., Hariz, M., Schneider, G.H., Yarrow, K., Brown, P., 2004. Event-related beta desynchronization in human subthalamic nucleus correlates with motor performance. Brain 127, 735–746. https://doi.org/10.1093/brain/awh106

Lafleur, K., Cassady, K., Doud, A., Shades, K., Rogin, E., He, B., 2013. Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain-computer interface. J Neural Eng 10. https://doi.org/10.1088/1741-2560/10/4/046003

Lawhern, V.J., Solon, A.J., Waytowich, N.R., Gordon, S.M., Hung, C.P., Lance, B.J., 2018. EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces. J Neural Eng 15, 056013. https://doi.org/10.1088/1741-2552/aace8c

Ledoit, O., Wolf, M., 2003. Honey, I Shrunk the Sample Covariance Matrix. Journal of Portfolio Management 30, 110–64.

Lemaître, G., Nogueira, F., Aridas, C.K., 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research 18, 1–5. https://doi.org/10.5555/3122009.3122026

Lempka, S.F., McIntyre, C.C., 2013. Theoretical Analysis of the Local Field Potential in Deep Brain Stimulation Applications. PLoS One. https://doi.org/10.1371/journal.pone.0059839

Leuthardt, E.C., Schalk, G., Wolpaw, J.R., Ojemann, J.G., Moran, D.W., 2004. A brain-computer interface using electrocorticographic signals in humans. J Neural Eng 1, 63–71. https://doi.org/10.1088/1741-2560/1/2/001

Levy, R., Ashby, P., Hutchison, W.D., Lang, A.E., Lozano, A.M., Dostrovsky, J.O., 2002. Dependence of subthalamic nucleus oscillations on movement and dopamine in Parkinson's disease. Brain 125, 1196–1209. https://doi.org/10.1093/brain/awf128

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. https://doi.org/10.1371/journal.pone.0006243

Limousin, P., Krack, P., Pollak, P., Benazzouz, A., Ardouin, C., Hoffmann, D.U., Benabib, A.L., 1998. Electrical stimulation of the subthalamic nucleus in advanced Parkinson's disease. New England Journal of Medicine 339, 1105–1111. https://doi.org/10.1056/NEJM199810153391603

Little, S., Pogosyan, A., Kuhn, A.A., Brown, P., 2012. Beta band stability over time correlates with Parkinsonian rigidity and bradykinesia. Exp Neurol 236, 383–388. https://doi.org/10.1016/j.expneurol.2012.04.024

Little, S., Pogosyan, A., Neal, S., Zavala, B., Zrinzo, L., Hariz, M., Foltynie, T., Limousin, P., Ashkan, K., FitzGerald, J., Green, A.L., Aziz, T.Z., Brown, P., 2013. Adaptive deep brain stimulation in advanced Parkinson disease. Ann Neurol 74, 449–457. https://doi.org/10.1002/ana.23951

Liu, X., Van der Spiegel, J., 2017. Brain-machine interface: Closed-loop bidirectional system design. Springer. https://doi.org/10.1007/978-3-319-67940-2

Lopez-Gordo, M.A., Sanchez Morillo, D., Pelayo Valle, F., 2014. Dry EEG electrodes. Sensors 14, 12847–12870. https://doi.org/10.3390/s140712847

Lotte, F., Guan, C., 2011. Regularizing common spatial patterns to improve BCI designs: Unified theory and new algorithms. IEEE Trans Biomed Eng 58, 355–362. https://doi.org/10.1109/TBME.2010.2082539

Loukas, C., Brown, P., 2004. Online prediction of self-paced hand-movements from subthalamic activity using neural networks in Parkinson's disease. J Neurosci Methods 137, 193–205. https://doi.org/10.1016/j.jneumeth.2004.02.017

Lozano, A.M., Lipsman, N., Bergman, H., Brown, P., Chabardes, S., Chang, J.W., Matthews, K., McIntyre, C.C., Schlaepfer, T.E., Schulder, M., Temel, Y., Volkmann, J., Krauss, J.K., 2019. Deep brain stimulation: current challenges and future directions. Nat Rev Neurol 15, 148–160. https://doi.org/10.1038/s41582-018-0128-2

Luciw, M.D., Jarocka, E., Edin, B.B., 2014. Multi-channel EEG recordings during 3,936 grasp and lift trials with varying weight and friction. Nature Scientific Data 1. https://doi.org/10.1038/sdata.2014.47

Luo, H., Huang, Y., Du, X., Zhang, Y., Green, A.L., Aziz, T.Z., Wang, S., 2018. Dynamic neural state identification in deep brain local field potentials of neuropathic pain. Front Neurosci 12. https://doi.org/10.3389/fnins.2018.00237

Mace, M., Yousif, N., Naushahi, M., Abdullah-Al-Mamun, K., Wang, S., Nandi, D., Vaidyanathan, R., 2014. An automated approach towards detecting complex behaviours in deep brain oscillations. J Neurosci Methods 224, 66–78. https://doi.org/10.1016/J.JNEUMETH.2013.11.019

Maling, N., Lempka, S.F., Blumenfeld, Z., Bronte-Stewart, H., McIntyre, C.C., 2018. Biophysical basis of subthalamic local field potentials recorded from deep brain stimulation electrodes. J Neurophysiol 120, 1932–1944. https://doi.org/10.1152/jn.00067.2018

Mamun, K.A., Huda, M.N., Mace, M., Lutman, M.E., Stein, J., Liu, X., Aziz, T., Vaidyanathan, R., Wang, S., 2012a. Pattern classification of deep brain local field potentials for brain computer interfaces, in: International Conference on Computer and Information Technology (ICCIT), Chittagong Bangladesh, 22-24 December 2012, IEEE. IEEE, pp. 518–523. https://doi.org/10.1109/ICCITechn.2012.6509740

Mamun, K.A., Mace, M., Lutman, M.E., Stein, J., Liu, X., Aziz, T., Vaidyanathan, R., Wang, S., 2015. Movement decoding using neural synchronization and inter-hemispheric connectivity from deep brain local field potentials. J Neural Eng 12, 056011. https://doi.org/10.1088/1741-2560/12/5/056011

Mamun, K.A., Mace, M., Lutman, M.E., Stein, J., Liu, X., Aziz, T., Vaidyanathan, R., Wang, S., 2012b. A robust strategy for decoding movements from deep brain local field potentials to facilitate brain machine interfaces, in: IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics, Rome Italy, 24-27 June 2012, IEEE . IEEE, pp. 320–325. https://doi.org/10.1109/BioRob.2012.6290708

Martineau, T., 2017. Controlling State Machines using Local Field Potential decoding in the Basal Ganglia. Imperial College London.

Martineau, T., He, S., Vaidyanathan, R., Brown, P., Tan, H., 2020. Optimizing Time-Frequency Feature Extraction and Channel Selection through Gradient Backpropagation to Improve Action Decoding based on Subthalamic Local Field Potentials, in: International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), 20-24 July 2020, Montreal QC Canada, IEEE. pp. 3023–3026. https://doi.org/10.1109/EMBC44109.2020.9175885

Martineau T, He S., Vaidyanathan R. and Tan H., 2023. Hyper-parameter tuning and feature extraction for asynchronous action detection from sub-thalamic nucleus local field potentials. Front. Hum. Neurosci.17:1111590. doi: 10.3389/fnhum.2023.1111590

Merk, T., Peterson, V., Lipski, W., Blankertz, B., Turner, R.S., Li, N., Horn, A., Kühn, A.A., Richardson, R.M., Neumann, W.-J., 2021. Electrocorticography is superior to subthalamic local field potentials for movement decoding in Parkinson's disease. bioRxiv 2021.04.24.441207. https://doi.org/10.1101/2021.04.24.441207

Merletti, R., Holobar, A., Farina, D., 2008. Analysis of motor units with high-density surface electromyography. Journal of Electromyography and Kinesiology 18, 879–890. https://doi.org/10.1016/j.jelekin.2008.09.002

Mohammed, A., Zamani, M., Bayford, R., Demosthenous, A., 2015. Patient specific Parkinson's disease detection for adaptive deep brain stimulation, in: Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), 25-29 August 2015, Milan Italy, IEEE. IEEE, pp. 1528–1531. https://doi.org/10.1109/EMBC.2015.7318662

Moreno, J.C., Pons, J.L., 2008. Case Study: Stance Stabilization During Gait Through Impedance Control, in: Pons, J.L. (Ed.), Wearable Robots Biomechatronics Exoskeleton. Wiley, pp. 158–161.

Mridha, M.F., Das, S.C., Kabir, M.M., Lima, A.A., Islam, M.R., Watanobe, Y., 2021. Brain-computer interface: advancement and challenges. Sensors 21, 1–46. https://doi.org/10.3390/s21175746

Musk, E., 2019. An integrated brain-machine interface platform with thousands of channels. J Med Internet Res 21, 1–14. https://doi.org/10.2196/16194

Nambu, A., Tokuno, H., Hamada, I., Kita, H., Imanishi, M., Akazawa, T., Ikeuchi, Y., Hasegawa, N., 1998. Excitatory cortical inputs to pallidal neurons via the subthalamic nucleus. Neurosci Res. https://doi.org/10.1016/s0168-0102(98)82646-1

Nandy, A., Alahe, M.A., Nasim Uddin, S.M., Alam, S., Nahid, A. Al, Awal, M.A., 2019. Feature extraction and classification of EEG signals for seizure detection, in: International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 10-12 January 2019, Dhaka Bangladesh, IEEE. IEEE, pp. 480–485. https://doi.org/10.1109/ICREST.2019.8644337

Neretti, N., Intrator, N., 2009. Gradient-based optimization of mother wavelets.

Nicolas-Alonso, L.F., Gomez-Gil, J., 2012. Brain computer interfaces, a review. Sensors 12, 1211–1279. https://doi.org/10.3390/s120201211

Niketeghad, S., Hebb, A.O., Nedrud, J., Hanrahan, S.J., Mahoor, M.H., 2018. Motor Task Detection from Human STN Using Interhemispheric Connectivity. IEEE Transactions on Neural Systems and Rehabilitation Engineering 26, 216–223. https://doi.org/10.1109/TNSRE.2017.2754879

Niketeghad, S., Hebb, A.O., Nedrud, J., Hanrahan, S.J., Mahoor, M.H., 2015. Motor task event detection using Subthalamic Nucleus Local Field Potentials, in: International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), 25-29 August 2015, Milan Italy, IEEE. IEEE, pp. 5553–5556. https://doi.org/10.1109/EMBC.2015.7319650

Niketeghad, S., Hebb, A.O., Nedrud, J., Hanrahan, S.J., Mahoor, M.H., 2014. Single trial behavioral task classification using subthalamic nucleus local field potential signals, in: International

Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 26-30 August 2014, Chicago IL USA, IEEE. pp. 3793–3796. https://doi.org/10.1109/EMBC.2014.6944449

Nordhausen, C.T., Maynard, E.M., Normann, R.A., 1996. Single unit recording capabilities of a 100 microelectrode array. Brain Res 726, 129–140. https://doi.org/10.1016/0006-8993(96)00321-6

Obermaier, B., Guger, C., Neuper, C., Pfurtscheller, G., 2001. Hidden Markov models for online classification of single trial EEG data. Pattern Recognit Lett 22, 1299–1309. https://doi.org/10.1016/S0167-8655(01)00075-7

Oertel, M.F., Schüpbach, W.M.M., Ghika, J.A., Stieglitz, L.H., Fiechter, M., Kaelin-Lang, A., Raabe, A., Pollo, C., 2017. Combined thalamic and subthalamic deep brain stimulation for tremor-dominant Parkinson's disease. Acta Neurochir (Wien) 159, 265–269. https://doi.org/10.1007/s00701-016-3044-5

Omlor, W., Patino, L., Mendez-Balbuena, I., Schulte-Monting, J., Kristeva, R., 2011. Corticospinal Beta-Range Coherence is Highly Dependent on the Pre-stationary Motor State. Journal of Neuroscience 31, 8037–8045. https://doi.org/10.1523/JNEUROSCI.4153-10.2011

Oostenveld, R., Fries, P., Maris, E., Schoffelen, J.M., 2011. FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. Comput Intell Neurosci 2011. https://doi.org/10.1155/2011/156869

OpenBCI Documentation [WWW Document], n.d. URL https://docs.openbci.com/FAQ/DocsUpdate/ (accessed 4.11.22).

Oppenheim, A. V., Schafer, R.W., 2010. The Time-Dependent Fourier Transform, in: Discrete-Time Signal Processing. Pearson, pp. 836–860.

Park, C., Took, C.C., Member, S., Mandic, D.P., 2014. Augmented Complex Common Spatial Patterns for Classification of Noncircular EEG From Motor Imagery Tasks. IEEE Transactions on Neural Systems and Rehabilitation Engineering 22, 1–10.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., Facebook, Z.D., Research, A.I., Lin, Z., Desmaison, A., Antiga, L., Srl, O., Lerer, A., 2017. Automatic differentiation in PyTorch.

Pedregosa, F., Michel, V., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Pedregosa, F., Varoquaux, G., Gramfort, A., Thirion, B., Grisel, O., Dubourg, V., Passos, A., Brucher, M., Perrot andÉdouardand, M., Duchesnay, A., Duchesnay, Fré., 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Pel, S.C., Tseng, C.C., 1997. IIR multiple notch filter design based on allpass filter. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 44, 133–136. https://doi.org/10.1109/82.554450

Pfurtscheller, G., Lopes da Silva, F.H., 1999. Event-related EEG/MEG synchronization and desynchronization: basic principles. Clinical Neurophysiology 110, 1842–1857. https://doi.org/10.1016/S1388-2457(99)00141-8

Pfurtscheller, G., Neuper, C., 2001. Motor imagery and direct brain-computer communication. Proceedings of the IEEE 89, 1123–1134. https://doi.org/10.1109/5.939829

Pikovsky, A., Rosenblum, M., 2007. Synchronization [WWW Document]. Scholarpedia. https://doi.org/10.4249/SCHOLARPEDIA.1459

Platt, 1999. Probablistic outputs for svm and comparison to regularized likelihood methods. Advances in Large Margin Classifiers 10, 61–74.

Pogosyan, A., Yoshida, F., Chen, C.C., Martinez-Torres, I., Foltynie, T., Limousin, P., Zrinzo, L., Hariz, M.I., Brown, P., 2010. Parkinsonian impairment correlates with spatially extensive subthalamic oscillatory synchronization. Neuroscience 171, 245–257. https://doi.org/10.1016/j.neuroscience.2010.08.068

Pons Jose, 2008. Introduction to Wearable Robotics, in: Biomechatronic. Wiley, pp. 1–17.

Powers, R., Etezadi-Amoli, M., Arnold, E.M., Kianian, S., Mance, I., Gibiansky, M., Trietsch, D., Alvarado, A.S., Kretlow, J.D., Herrington, T.M., Brillman, S., Huang, N., Lin, P.T., Pham, H.A., Ullal, A. v., 2021. Smartwatch inertial sensors continuously monitor real-world motor fluctuations in Parkinson's disease. Sci Transl Med 13, 1–12. https://doi.org/10.1126/scitranslmed.abd7865

Pozzi, N.G., Pacchetti, C., 2017. Back to the future: 30th anniversary of deep brain stimulation for Parkinson's disease. Funct Neurol. https://doi.org/10.11138/FNeur/2017.32.1.005

Raciti, L., Pignolo, L., Perini, V., Pullia, M., Porcari, B., Latella, D., Isgrò, M., Naro, A., Calabrò, R.S., 2022. Improving Upper Extremity Bradykinesia in Parkinson's Disease: A Randomized Clinical Trial on the Use of Gravity-Supporting Exoskeletons. J Clin Med 11. https://doi.org/10.3390/jcm11092543

Rao, R.P.N., 2013. Applications of BCIs, Brain-Computer Interfacing. Cambridge University Press. https://doi.org/10.1017/cbo9781139032803.017

Ren, J., Kezunovic, M., 2010. Use of recursive wavelet transform for estimating power system frequency and phasors, in: IEEE PES Transmission and Distribution Conference and Exposition: Smart Solutions for a Changing World, New Orleans LA USA, 19-22 April 2010, IEEE. IEEE, pp. 1–6. https://doi.org/10.1109/TDC.2010.5484474

Ren, J., Kezunovic, M., Stenbakken, G., 2011. Characterizing dynamic behavior of PMUs using step signals. European Transactions on Electrical Power 21, 1496–1508. https://doi.org/10.1002/etep.513

Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., Lécuyer, A., 2010. OpenViBE: An open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. Presence: Teleoperators and Virtual Environments 19, 35–53. https://doi.org/10.1162/PRES.19.1.35

Rieder, P., Götze, J., Nossek, J.A., Burrus, C.S., 1998. Parameterization of Orthogonal Wavelet Transforms and Their Implementation. IEEE Transactions on Proceedings of the Circuits and Systems - II: Analog and Digital signal Processing 45, 217–226. https://doi.org/10.1109/82.661654

Riviere, C.N., Reich, S.G., Thakor, N. V., 1997. Adaptive Fourier modeling for quantification of tremor. J Neurosci Methods 74, 77–87. https://doi.org/10.1016/S0165-0270(97)02263-2

Rivlin-Etzion, M., Marmor, O., Heimer, G., Raz, A., Nini, A., Bergman, H., 2006. Basal ganglia oscillations and pathophysiology of movement disorders. Curr Opin Neurobiol. https://doi.org/10.1016/j.conb.2006.10.002

Robinson, N., Vinod, A.P., Ang, K.K., Tee, K.P., Guan, C.T., 2013. EEG-based classification of fast and slow hand movements using wavelet-CSP algorithm. IEEE Trans Biomed Eng 60, 2123–2132. https://doi.org/10.1109/TBME.2013.2248153

Rocon, E., Pons, J L, 2008. Case Study: Upper Limb Tremor Suppression Through Impedance Control, in: Pons, José L. (Ed.), Wearable Robots: Biomechatronic Exoskeletons. Wiley, pp. 156–157.

Rosa, M., Scelzo, E., Locatelli, M., Carrabba, G., Levi, V., Arlotti, M., Barbieri, S., Rampini, P., Priori, A., 2017. Risk of Infection After Local Field Potential Recording from Externalized Deep Brain Stimulation Leads in Parkinson's Disease. World Neurosurg 97, 64–69. https://doi.org/10.1016/j.wneu.2016.09.069

Ross, B.C., 2014. Mutual information between discrete and continuous data sets. PLoS One 9. https://doi.org/10.1371/journal.pone.0087357

Rouse, A.G., Stanslaski, S.R., Cong, P., Jensen, R.M., Afshar, P., Ullestad, D., Gupta, R., Molnar, G.F., Moran, D.W., Denison, T.J., 2011. A chronic generalized bi-directional brain-machine interface. J Neural Eng 8. https://doi.org/10.1088/1741-2560/8/3/036018

Rozman, R., Kodek, D.M., 2007. Using asymmetric windows in automatic speech recognition. Speech Commun 49, 268–276. https://doi.org/10.1016/j.specom.2007.01.012

Ruiz, M.H., Rusconi, M., Brücke, C., Haynes, J.D., Schönecker, T., Kühn, A.A., 2014. Encoding of sequence boundaries in the subthalamic nucleus of patients with Parkinson's disease. Brain 137, 2715–2730. https://doi.org/10.1093/brain/awu191

Russell, B., Han, J., 2016. Jean Morlet and the Continuous Wavelet Transform. Consortium for Research in Elastic Wave Exploration Seismology 28, 1–15.

Rutledge, 1998. Wavelet Packets, in: Akay, M. (Ed.), Time Frequency and Wavelet in Biomedical Signal Processing. IEEE Press, pp. 368–371.

Sand, D., Rappel, P., Marmor, O., Bick, A.S., Arkadir, D., Lu, B.L., Bergman, H., Israel, Z., Eitan, R., 2021. Machine learning-based personalized subthalamic biomarkers predict ON-OFF levodopa states in Parkinson patients. J Neural Eng 18. https://doi.org/10.1088/1741-2552/abfc1d

Schwartz, A.B., 1994. Distributed motor processing in cerebral cortex. Curr Opin Neurobiol 4, 840–846. https://doi.org/10.1016/0959-4388(94)90132-5

Schwartz, A.B., Cui, X.T., Weber, D.J., Moran, D.W., 2006. Brain-Controlled Interfaces: Movement Restoration with Neural Prosthetics. Neuron. https://doi.org/10.1016/j.neuron.2006.09.019

Seabold, S., Perktold, J., 2010. Statsmodels : Econometric and Statistical Modeling with Python 92–96.

Sergio, L.E., 2002. Systematic Changes in Motor Cortex Cell Activity With Arm Posture During Directional Isometric Force Generation. J Neurophysiol 89, 212–228. https://doi.org/10.1152/jn.00016.2002

Sergio, L.E., Kalaska, J.F., 1998. Changes in the Temporal Pattern of Primary Motor Cortex Activity in a Directional Isometric Force Versus Limb Movement Task. J Neurophysiol 80, 1577–1583. https://doi.org/10.1152/jn.1998.80.3.1577

Shah, S.A., Tan, H., Brown, P., 2016. Decoding force from deep brain electrodes in Parkinsonian patients, in: Proceedings of the Annual International Conference of the IEEE Engineering in

Medicine and Biology Society, EMBS. IEEE, pp. 5717–5720. https://doi.org/10.1109/EMBC.2016.7592025

Shah, S.A., Tan, H., Tinkhauser, G., Brown, P., 2018. Towards Real-Time, Continuous Decoding of Gripping Force from Deep Brain Local Field Potentials. IEEE Transactions on Neural Systems and Rehabilitation Engineering 26, 1460–1468. https://doi.org/10.1109/TNSRE.2018.2837500

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N., 2016. Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE 104, 148–175. https://doi.org/10.1109/JPROC.2015.2494218

Shanechi, M.M., 2017. Brain-Machine Interface Control Algorithms. IEEE Transactions on Neural Systems and Rehabilitation Engineering 25, 1725–1734. https://doi.org/10.1109/TNSRE.2016.2639501

Shanechi, M.M., Carmena, J.M., 2013. Optimal feedback-controlled point process decoder for adaptation and assisted training in brain-machine interfaces. Proceedings of the International IEEE/EMBS Conference on Neural Engineering (NER), San Diego CA USA, 06-08 November 2013, IEEE 653–656. https://doi.org/10.1109/NER.2013.6696019

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. Nat Commun 8. https://doi.org/10.1038/ncomms13825

Silberstein, P., Pogosyan, A., Kühn, A.A., Hotton, G., Tisch, S., Kupsch, A., Dowsey-Limousin, P., Hariz, M.I., Brown, P., 2005. Cortico-cortical coupling in Parkinson's disease and its modulation by therapy. Brain 128, 1277–1291. https://doi.org/10.1093/brain/awh480

Singla, R., Devgan, A., Gogna, P., Batra, A., 2014. Optimizing Spatial filters for Robust EEG Single-Trial Analysis. Journal of Orthopaedic Surgery 22, 70–74. https://doi.org/10.1177/230949901402200118

Smola, A.J., Scholkopf, B., 2004. A tutotial on support vector regression. Stat Comput 14, 199–222.

Smulders, F.T.Y., ten Oever, S., Donkers, F.C.L., Quaedflieg, C.W.E.M., van de Ven, V., 2018. Single-trial log transformation is optimal in frequency analysis of resting EEG alpha. European Journal of Neuroscience 48, 2585–2598. https://doi.org/10.1111/ejn.13854

Spraker, M.B., Yu, H., Corcos, D.M., Vaillancourt, D.E., 2007. Role of individual basal ganglia nuclei in force amplitude generation. J Neurophysiol 98, 821–834. https://doi.org/10.1152/jn.00239.2007

Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W., 2012. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Trans Inf Theory 58, 3250–3265. https://doi.org/10.1109/TIT.2011.2182033

Stanslaski, S., Herron, J., Chouinard, T., Bourget, D., Isaacson, B., Kremen, V., Opri, E., Drew, W., Brinkmann, B.H., Gunduz, A., Adamski, T., Worrell, G.A., Denison, T., 2018. A Chronically Implantable Neural Coprocessor for Investigating the Treatment of Neurological Disorders. IEEE Trans Biomed Circuits Syst 12, 1230–1245. https://doi.org/10.1109/TBCAS.2018.2880148

Sun, Q., Zhao, D., Cheng, S., Hou, X., Zhao, X., Tian, Y., 2018. A feature extraction method for adaptive DBS using an improved EMD. International Journal of Neuroscience 128, 975–986. https://doi.org/10.1080/00207454.2018.1450253

Szendro, P., Vincze, G., Szasz, A., 2001. Pink-noise behaviour of biosystems. European Biophysics Journal 30, 227–231. https://doi.org/10.1007/s002490100143

Tan, H., Debarros, J., He, S., Pogosyan, A., Aziz, T.Z., Huang, Y., Wang, S., Timmermann, L., Visser-Vandewalle, V., Pedrosa, D.J., Green, A.L., Brown, P., 2019. Decoding voluntary movements and postural tremor based on thalamic LFPs as a basis for closed-loop stimulation for essential tremor. Brain Stimul 12, 858–867. https://doi.org/10.1016/j.brs.2019.02.011

Tan, H., Fischer, P., Shah, S.A., Vidaurre, D., Woolrich, M.W., Brown, P., 2018. Decoding Movement States in Stepping Cycles Based on Subthalamic LFPs in Parkinsonian Patients, in: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS. pp. 1384–1387. https://doi.org/10.1109/EMBC.2018.8512545

Tan, H., Pogosyan, A., Anzak, A., Ashkan, K., Bogdanovic, M., Green, A.L., Aziz, T., Foltynie, T., Limousin, P., Zrinzo, L., Brown, P., 2013. Complementary roles of different oscillatory activities in the subthalamic nucleus in coding motor effort in Parkinsonism. Exp Neurol 248, 187–195. https://doi.org/10.1016/j.expneurol.2013.06.010

Tan, H., Pogosyan, A., Ashkan, K., Cheeran, B., FitzGerald, J.J., Green, A.L., Aziz, T., Foltynie, T., Limousin, P., Zrinzo, L., Brown, P., 2015. Subthalamic nucleus local field potential activity helps encode motor effort rather than force in parkinsonism. J Neurosci 35, 5941–9. https://doi.org/10.1523/JNEUROSCI.4609-14.2015

Tan, H., Pogosyan, A., Ashkan, K., Green, A.L., Aziz, T., Foltynie, T., Limousin, P., Zrinzo, L., Hariz, M., Brown, P., 2016. Decoding gripping force based on local field potentials recorded from subthalamic nucleus in humans. Elife 5, 484–499. https://doi.org/10.7554/eLife.19089

Tan, H., Zavala, X.B., Pogosyan, A., Ashkan, K., Zrinzo, L., Foltynie, T., Limousin, P., Brown, P., 2014. Human Subthalamic Nucleus in Movement Error Detection and Its Evaluation during Visuomotor Adaptation. https://doi.org/10.1523/JNEUROSCI.3414-14.2014

Tarvainen, M.P., Hiltunen, J.K., Ranta-Aho, P.O., Karjalainen, P.A., 2004. Estimation of Nonstationary EEG with Kalman Smoother Approach: An Application to Event-Related Synchronization (ERS). IEEE Trans Biomed Eng 51, 516–524. https://doi.org/10.1109/TBME.2003.821029

Todorov, E., 2000. Direct cortical control of muscle activation in voluntary arm movements: a model. J Neurophysiol 80, 696–714.

Tomioka, R., Dornhege, G., Nolte, G., Blankertz, B., Aihara, K., Müller, K.-R., 2006. Spectrally Weighted Common Spatial Pattern Algorithm for Single Trial EEG Classification. Mathematical Engineering Technical Reports, Department of Mathematical Informatics, Graduate School of Information Science And Technology, University of Tokyo.

Torch Contributors, 2019. Pytorch Documentation [WWW Document]. URL https://pytorch.org/docs/stable/index.html (accessed 1.17.20).

Torres Valderrama, A., Oostenveld, R., Vansteensel, M.J., Huiskamp, G.M., Ramsey, N.F., 2010. Gain of the human dura in vivo and its effects on invasive brain signal feature detection. J Neurosci Methods 187, 270–279. https://doi.org/10.1016/J.JNEUMETH.2010.01.019

Toth, E., Kumar, S.S., Chaitanya, G., Riley, K., Balasubramanian, K., Pati, S., 2020. Machine learning approach to detect focal-onset seizures in the human anterior nucleus of the thalamus. J Neural Eng 17. https://doi.org/10.1088/1741-2552/abc1b7

Turner, R.S., Anderson, M.E., 1997. Pallidal discharge related to the kinematics of reaching movements in two dimensions. J Neurophysiol 77, 1051–1074.

Turner, R.S., Desmurget, M., Grethe, J., Crutcher, M.D., Grafton, S.T., 2003. Motor Subcircuits Mediating the Control of Movement Extent and Speed. J Neurophysiol 90, 3958–3966. https://doi.org/10.1152/jn.00323.2003

Vaidyanathan, P., 1993. Two Channel FIR Paraunitary QMF Banks, in: Multirate Systems and Filter Banks. Prentice Hall, pp. 298–314.

Vaillancourt, D.E., Mayka, M.A., Thulborn, K.R., Corcos, D.M., 2004. Subthalamic nucleus and internal globus pallidus scale with the rate of change of force production in humans. Neuroimage 23, 175–186. https://doi.org/10.1016/j.neuroimage.2004.04.040

Vaillancourt, D.E., Yu, H., Mayka, M.A., Corcos, D.M., 2007. Role of the basal ganglia and frontal cortex in selecting and producing internally guided force pulses. Neuroimage 36, 793–803. https://doi.org/10.1016/j.neuroimage.2007.03.002

Valstar, M.F., Pantic, M., 2007. Combined Support Vector Machines and Hidden Markov Models for Modeling Facial Action Temporal Dynamics, in: International Workshop on Human-Computer Interaction, Rio de Janeiro Brazil, 20 October, 2007 Springer. pp. 118–127.

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/SUPPL_FILE/NEJMOA1608085_DISCLOSURES.PDF

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., Contributors, S. 1. 0, 2019. SciPy 1.0--Fundamental Algorithms for Scientific Computing in Python.

Volpe, D., Giantin, M.G., Fasano, A., 2014. A wearable proprioceptive stabilizer (Equistasi®) for rehabilitation of postural instability in Parkinson’s disease: A phase II randomized double-blind, double-dummy, controlled study. PLoS One 9. https://doi.org/10.1371/journal.pone.0112065

Wang, B., Wong, C.M., Kang, Z., Liu, F., Shui, C., Wan, F., Chen, C.L.P., 2020. Common Spatial Pattern Reformulated for Regularizations in Brain–Computer Interfaces. IEEE Transactions on Cydernetics 51, 5008–5020.

Wang, S., Green, A., Stein, J., Aziz, T., Liu, X., 2008. Identifying rhythms of subthalamic neural oscillations in time-frequency domain. Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS), Vancouver BC Canada, 20-25 August 2008, IEEE 5724–5728. https://doi.org/10.1109/iembs.2008.4650514

Wang, T., Shoaran, M., Emami, A., 2018. Towards Adaptive Deep Brain Stimulation: LFP-Feature Analysis and Classification, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary AB Canada, 15-20 April 2018, IEEE. pp. 2536–2540.

Wang, X.W., Nie, D., Lu, B.L., 2014. Emotional state classification from EEG data using machine learning approach. Neurocomputing 129, 94–106. https://doi.org/10.1016/j.neucom.2013.06.046

Wang, Y., Veluvolu, K.C., Lee, M., 2013. Time-frequency analysis of band-limited EEG with BMFLC and Kalman filter for BCI applications. J Neuroeng Rehabil 10, 1–16. https://doi.org/10.1186/1743-0003-10-109

Wiskott, L., Sejnowski, T., 2002. Slow Feature Analysis: Unsupervised Learning of Invariances. Neural Comput 14, 715–770.

Wiskott-lab, 2021. sklearn-sfa [WWW Document]. Github. URL https://github.com/wiskott-lab/sklearn-sfa

Wodlinger, B., Downey, J.E., Tyler-Kabara, E.C., Schwartz, A.B., Boninger, M.L., Collinger, J.L., 2015. Ten-dimensional anthropomorphic arm control in a human brain−machine interface: difficulties, solutions, and limitations. J Neural Eng 12, 016011. https://doi.org/10.1088/1741-2560/12/1/016011

Wolpaw, J.R., Birbaumer, N., Mcfarland, D.J., Pfurtscheller, G., Vaughan, T.M., 2002. Brain-computer interfaces for communication and control. Clinical Neurophysiology 113, 767–791.

Wolpaw, J.R., McFarland, D.J., 2004. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. Proc Natl Acad Sci U S A 101, 17849–54. https://doi.org/10.1073/pnas.0403504101

Wong, C.H., Birkett, J., Byth, K., Dexter, M., Somerville, E., Gill, D., Chaseling, R., Fearnside, M., Bleasel, A., 2009. Risk factors for complications during intracranial electrode recording in presurgical evaluation of drug resistant partial epilepsy. Acta Neurochir (Wien) 151, 37–50. https://doi.org/10.1007/S00701-008-0171-7

Wright, J., Macefield, V.G., van Schaik, A., Tapson, J.C., 2016. A Review of Control Strategies in Closed-Loop Neuroprosthetic Systems. Front Neurosci 10, 312. https://doi.org/10.3389/fnins.2016.00312

Xie, Z., Schwartz, O., Prasad, A., 2018. Decoding of finger trajectory from ECoG using deep learning. J Neural Eng 15. https://doi.org/10.1088/1741-2552/aa9dbe

Yang, Y., Kamboh, A.M., Mason, A.J., 2014. A configurable realtime DWT-based neural data compression and communication VLSI system for wireless implants. J Neurosci Methods 227, 140–150. https://doi.org/10.1016/j.jneumeth.2014.02.009

Yao, L., Brown, P., Shoaran, M., 2020. Improved detection of Parkinsonian resting tremor with feature engineering and Kalman filtering. Clinical Neurophysiology 131, 274–284. https://doi.org/10.1016/j.clinph.2019.09.021

Yao, L., Brown, P., Shoaran, M., 2018. Resting Tremor Detection in Parkinson's Disease with Machine Learning and Kalman Filtering. Proceedings of the IEEE Biomedical Circuits and Systems

Conference (BioCAS) 2018, Cleveland OH USA, 17-19 October 2018, IEEE 10–13. https://doi.org/10.1109/BIOCAS.2018.8584721

Zaker, N., Dutta, A., Maurer, A., Zhang, J.J., Hanrahan, S., Hebb, A.O., Kovvali, N., Papandreou-Suppappola, A., 2015. Adaptive learning of behavioral tasks for patients with Parkinson's disease using signals from deep brain stimulation, in: Proceedings of the Asilomar Conference on Signals, Systems and Computers, Pacific Grove CA USA, 02-05 November 2014. pp. 208–212. https://doi.org/10.1109/ACSSC.2014.7094429

Zaker, N., Zhang, J.J., Hanrahan, S., Nedrud, J., Hebb, A.O., 2016. A new approach for automated detection of behavioral task onset for patients with Parkinson's disease using subthalamic nucleus local field potentials, in: Proceedings of the Asilomar Conference on Signals, Systems and Computers, Pacific Grove CA USA, 29 February 2016, IEEE. IEEE, pp. 780–784. https://doi.org/10.1109/ACSSC.2015.7421240

Zhang, H., 2004. The optimality of Naive Bayes, in: International Florida Artificial Intelligence Research Society Conference (FLAIRS), Miami Beach Florida USA. pp. 562–567.

Zhang, P., Huang, J., Li, W., Ma, X., Yang, P., Dai, J., He, J., 2019. Using High-Frequency Local Field Potentials from Multicortex to Decode Reaching and Grasping Movements in Monkey. IEEE Trans Cogn Dev Syst 11, 270–280. https://doi.org/10.1109/TCDS.2018.2869587

Zhang, S., Connolly, A.T., Madden, L.R., Vitek, J.L., Johnson, M.D., 2018. High-resolution local field potentials measured with deep brain stimulation arrays. J Neural Eng 15, 046019. https://doi.org/10.1088/1741-2552/aabdf5

Zhang, S., Green, A., Smith, P.P., 2013. An automatic classifier of pain scores in chronic pain patients from local field potentials recordings, in: Proceedings of the International IEEE/EMBS Conference on Neural Engineering (NER), San Diego CA USA, 06-08 November 2013, IEEE. IEEE, pp. 1194–1197. https://doi.org/10.1109/NER.2013.6696153

Zhu, B., Farivar, M., Shoaran, M., 2020. ResOT: Resource-Efficient Oblique Trees for Neural Signal Classification. IEEE Trans Biomed Circuits Syst 14, 692–704. https://doi.org/10.1109/TBCAS.2020.3004544

Appendix A: STN-LFP Recording Datasets

The data used in this work was provided by the Tan group from the MRC Brain Network Dynamics Unit at the University of Oxford. All experimental recording were conducted according to the Declaration of Helisinki and approved by the local ethics committees (Oxford C Research Ethics Committee 18/SC/0006). All participants gave informed consent. Data was shared anonymously, with no personal sensitive or identifiable information as allowed per the ethical protocol. Part of the anonymised patient information had already been previously published (Tan et al. 2013, 2015, 2016). The Tan group has made several similar datasets openly accessible in the data sharing platform hosted by the MRC Brain Network Dynamics Unit in accordance with the Medical Research Council Data Sharing Policy (version 2.2), and the Concordat on Open Research Data. This platform can be used to make publicly available any of the following datasets.

A. Single Hand Grip Recordings

Table A.1 summarises the participants and the characteristics for every recording belonging to the first dataset, used for action detection decoding in Chapter 4. All participants had recently undergone DBS surgery targeting the STN for the treatment of PD and had externalised leads for recording. The recording procedure is described in Tan et al. (2015, 2016) along with further participant details (such as medication state during recording). STN-LFP were recorded while participants were performing gripping isometric contractions at the pace of a visual or auditory cue. The grip force was recorded with a dynamometer (Hand Grip Dynamometer, Biometrics Ltd.). The dataset includes two subsets, the first where participants were asked to perform their respective maximum voluntary contraction (MVC) (noted as Max) and the second where they were asked to maintain a given force level as a percentage of their respective MVC (noted as Level). The STN-LFP and dynamometer signals were acquired with a TMSi Porti (TMSi Ltd., the Netherlands). The recordings were sampled at 2048Hz and a 22bit quantisation (adding to a resolution of 71.526 nV/bit) for STN-LFP signals.

Table A.1 - Summary of single hand grip recordings used for action detection decoding; (**Right**) or (**Left**) indicate which hand was employed; (**Max**) or (**Level**) indicate if the MVC was demanded or variable levels of force; Recording details, number of trials, number of channel recorded (in contra or ipsi-lateral STN), detection class balance (grip force against rest), average force level are reported; All recording were acquired at 2048Hz with the TMSi Porti; Recordings include STN-LFP signals and isometric grip contractions reading from an hand held dynamometer.

Participant ID	Recording Duration (min)	Contra-Lateral Channel	Ipsi-Lateral Channel	Number of Trials	Mean (±std) Grip Duration (s)	Ratio of Grip to Rest	Mean (±std) Grip Level (relative to MVC)
Sub01_Level_Right	6.38	4	4	26	2.95±0.64	0.20	0.33±0.19
Sub01_Max_Left	3.82	4	3	19	5.95±1.54	0.49	0.69±0.17
Sub02_Max_Left	3.99	4	0	21	5.33±0.22	0.47	0.63±0.13
Sub03_Max_Left	5.62	4	3	22	5.37±0.19	0.35	0.82±0.12
Sub03_Max_Right	6.42	3	4	24	5.28±0.67	0.33	0.68±0.15
Sub04_Level_Left	8.54	4	4	34	5.10±1.33	0.34	0.37±0.26
Sub04_Level_Right	7.54	4	4	34	3.61±0.91	0.27	0.29±0.21
Sub07_Level_Left	4.28	4	4	20	3.25±0.16	0.25	0.55±0.28
Sub10_Level_Left	9.21	4	2	49	3.30±1.72	0.29	0.39±0.24
Sub10_Level_Right	8.39	2	4	40	3.79±1.34	0.30	0.44±0.28
Cohort (mean±std)	6.42±1.88			29±9	4.22±1.51	0.33±0.09	0.48±0.27

B. Dual Hand Pinch Recordings

Table A.2 summarises the participants and the characteristics for every recording belonging to the second dataset, used for laterality selection and backpropagation experiment decoding in Chapter 4. The collection procedure is described in Fischer et al. (2017). STN-LFP were recorded while participants were performing pinch isometric contractions at the pace of a visual cue. The grip force was recorded with two pinch dynamometers held in each hand (Pinch Dynamometer, Biometrics Ltd). The STN-LFP and dynamometer signals were acquired with a TMSi Porti. The recordings were sampled at 2048Hz.

Table A.2 - Summary of the dual hand dataset used for laterality selection decoding; Recording details, number of trials, number of channel recorded (in right and left STN), detection class balance (grip force against rest), average force level are reported; All recording were acquired at 2048Hz with the TMSi Porti; Recordings include STN-LFP signals and two isometric pinch contractions reading from a left and right hand held dynamometer.

Participant ID	Recording Duration (min)	Left Channel	Right Channel	Left Trials	Right Trials	Ratio Left/Right	Grip Duration	Ratio of Grip to Rest	Mean (±std) Grip Level (relative to MVC)
Sub01	4.99	8	8	26	26	1.00	2.40±0.33	0.42	0.47±0.14
Sub02	10.25	3	3	37	33	0.89	3.86±0.86	0.44	0.44±0.16
Sub03	10.29	3	3	33	34	0.97	3.29±0.71	0.36	0.44±0.16
Sub04	9.35	3	3	31	29	0.94	3.32±0.71	0.36	0.45±0.18
Sub05	3.70	4	4	13	12	0.92	3.26±1.07	0.37	0.50±0.07
Sub06	3.19	4	4	13	14	0.93	3.69±1.73	0.52	0.60±0.13
Sub07	3.44	4	4	13	11	0.85	3.27±0.52	0.38	0.42±0.15
Sub08	4.37	4	4	12	14	0.86	3.33±1.42	0.33	0.50±0.13
Sub09	5.20	4	4	22	22	1.00	3.13±1.14	0.44	0.47±0.15
Sub10	5.19	4	4	23	23	1.00	3.07±0.28	0.45	0.47±0.13
Sub11	3.74	8	8	17	16	0.94	3.02±0.36	0.44	0.41±0.12
Sub12	5.83	4	4	25	19	0.76	2.81±1.49	0.35	0.40±0.13
Cohort (mean±std)	5.80±2.53			19±10	18±9	0.92±0.07	3.21±1.02	0.40±0.05	0.46±0.15

C. Gradient Recordings

Table A.3 summarises the participants and the characteristics for every recording belonging to the third dataset used to study the combined force and yank paradigm and continuous state decoding in Chapter 5. The grip force was recorded with a grip dynamometer held in a single hand. The STN-LFP and dynamometer signals were acquired with a TMSi Porti. The recordings were sampled at 1024Hz. A recording software enabled to display at a rate of at least 20Hz a target cursor moving according to the pre-planned trajectory and tracking cursor to provided visual feedback using the dynamometer reading (with some normalisation and smoothing). The software upsampled both the target and the tracking cursors signals and saved them simultaneously and synchronously with the LFP data. Trial information (demanded ramp level, gradient, chirp acceleration ramp) were also recorded.

Table A.3 - Summary of the ramps and chirp tracking task paradigm dataset used for continuous state decoding and offline analysis; Recording details, number of trials for ramps and chirp trajectories, number of channel recorded (in contra and ipsi-STN hemispheres) are reported; All recording were acquired at 1024Hz with the TMSi Porti; Recordings include STN-LFP signals, isometric grip contractions reading from an hand held dynamometer, tracker target, tracker position (fed back from the dynamometer) and trial information identifier.

Participant ID	Ramp Recording Duration (min)	Ramp Trials	Chirp Recording Duration (min)	Chirp Trials	Contra-Lateral Channel	Ipsi-Lateral Channel
Sub8R	32.34	108	13.75	18	4	4
Sub9R	32.80	108	15.52	20	4	4
Sub11R	33.69	114	15.13	20	4	4
Sub11L	31.37	108	13.35	18	4	4
Sub10R	32.92	111	21.76	29	4	4
Sub10L	31.91	108	13.53	18	4	4
Cohort (mean±std)	32.50±0.75	110±2	15.51±2.91	20±4		

Prior to Sub8R several recordings other recordings were made using an earlier version of the ramps and chirp paradigm, all presented in Table A.4. Unfortunately, not enough trial iterations were recorded for each condition to obtained significant results after analysis and the enhanced common-mode-suppression functionality of the TMSi Porti had not been engaged resulting in recordings excessively corrupted by mains artefact noise.

Table A.4 - Summary of ramps and chirp tracking task paradigm dataset rejected for continuous state decoding; Recording details, number of trials for ramps and chirp trajectories, number of channel recorded (in contra and ipsi-STN hemispheres) are reported; All recording were acquired at 1024Hz with the TMSi Porti; Recordings include STN-LFP signals, isometric grip contractions reading from an hand held dynamometer, tracker target, tracker position (fed back from the dynamometer) and trial information identifier.

Participant ID	**Ramp Recording Duration (min)**	**Ramp Trials**	**Chirp Recording Duration (min)**	**Chirp Trials**	**Contra-Lateral Channel**	**Ipsi-Lateral Channel**
Sub1.1L	7.60	30	3.42	3	8	8
Sub1.1R	7.22	30	4.38	3	8	8
Sub2L	7.13	30	4.84	3	8	8
Sub3L	11.66	50	6.71	5	4	4
Sub3R	12.77	50	7.16	5	4	4
Sub4L	4.45	24	6.58	5	4	4
Sub4R	11.85	50	6.80	5	4	4
Sub4.1L	11.90	50	6.73	5	4	4
Sub4.1R	11.57	50	6.59	5	4	4
Sub5L	11.93	50	6.59	5	4	4
Sub5R	11.57	50	6.79	5	4	4
Cohort (mean±std)	9.97±2.67	42±10	6.05±1.18	4±1		

Appendix B: BO Selection of Surrogate Model and Acquisition Function

A. Comparisons and Reference to Random Search

Bayesian Optimisation (BO) was implemented using the Scikit-Optimize (Head et al., 2021) library. Scikit-Optimize proposes several surrogate models, including gaussian-process (GP) (with Matern-Kernel) and gradient-boosted regression trees (GBTR) models (Head et al., 2021), and different types of acquisition function (expected improvement (EI), probability of improvement (PI), and lower bound confidence (LBC)). For the GP, the GP-edge function was also available as a fusion of the EI, PI and LBC acquisition functions. Which combination was of surrogate model and acquisition function would be preferrable was not immediately apparent. A simple point of comparison to evaluate the efficiency of BO was random search (RS). RS independently samples parameter vectors ρ randomly from the search space S_ρ at each iteration. Bergstra et al. (2012) has emphasised that RS is already a robust method to tune hyperparameters. RS only needs to sample ρ^* from a sub-space space $\mathcal{S}_\rho^* \subset \mathcal{S}_\rho$ of near-optimal solutions. Given a volumetric ratio r_S between $\mathcal{S}_\rho$ and $\mathcal{S}_\rho^*$, the total number of iteration N_i required to sample at least one ρ^*, such that $\rho^* \epsilon\ \mathcal{S}_\rho^*$, with a chance of success α is given:

$$N_i \geq \frac{\log(1-\alpha)}{\log(1-\mathrm{r_s})} \tag{B-1}$$

As a reference, a coarse random search of 60 iterations had a approximatively a 95% chance of finding a solution in a target space possessing 5% of the initial volume of $\mathcal{S}_p$. Hence, even if the BO behaved purely as an RS, the chances of stumbling on a near-optimal were relatively high. BO was, however to expected to differ from RS in two ways. First BO methods should converge asymptotically towards a maximum score s^* faster and higher s^* than RS. By building the surrogate model, BO should be capable of identifying potential sub-spaces like $\mathcal{S}_\rho^*$. BO could then knowingly exploit them as the search refined itself. Secondly, BO should be capable overall to make better-informed decisions at every iteration step. The decision ability of the algorithm could be measured with the notion of cumulated regret R_i. At each i^{th} search step, the cumulated regret was given by:

$$\begin{cases} R_i = R_{i-1} + (s_i^* - F(x, y, \rho)) \quad \forall\, i = 1, \dots, N_i \\ R_o = 0 \end{cases} \tag{B-2}$$

where F is the pipeline fitting function using the dataset (x, y), ρ_i the current parameter vector being sampled, the s_i^* the highest score found on or prior to the i^{th} iteration. Decoders for the action detection task presented in Chapter 3.D were tuned using RS, and BO setup with combinations of GP, or GBTR and EI, PI, LCB, or GP-edge acquisition functions. The STFT-band-CHS extractor and LR classifier were employed. The decoders were tuned on three participant datasets (Sub01_Level_Right, Sub04_Level_Right, Sub10_Level_Left from the ‘Single Hand Grip ’ dataset, Appendix A.A) 5 times each. 60 iterations were employed for each optimisation run.

B. Results and Discussion

Figure B.1 and Figure B.2 show the convergence and cumulated regret for both GP and GBTR methods, respectively. Default parameters were used in the first iteration, and no random initial sampling was employed to seed the surrogate model fitting. It was observed that RD could, in some instances, start faster than BO methods but would ultimately be overtaken. RD maked poorer decisions as the BO methods began a more refined parameter space search. Regret appeared to scale linearly with the iteration number for RD. PI acquisition function has appeared to perform better than others in either the GP or GBTR case, converging fast and accumulating less regret at each iteration. The uncertainty in the convergence path of RD could be as well substantial compared to BO. A single BO run is thereby more likely to yield a better solution in a few iterations systematically. As the GBTR did not seem to offer any significant advantage, it was decided to maintain GP surrogate model as it is more commonly employed throughout other studies (Bashashati et al., 2016b). The GP-edge-function was as well-chosen as it performed similarly to the PI acquisition function and would allow more flexibility in cases where the PI function is not as effective.

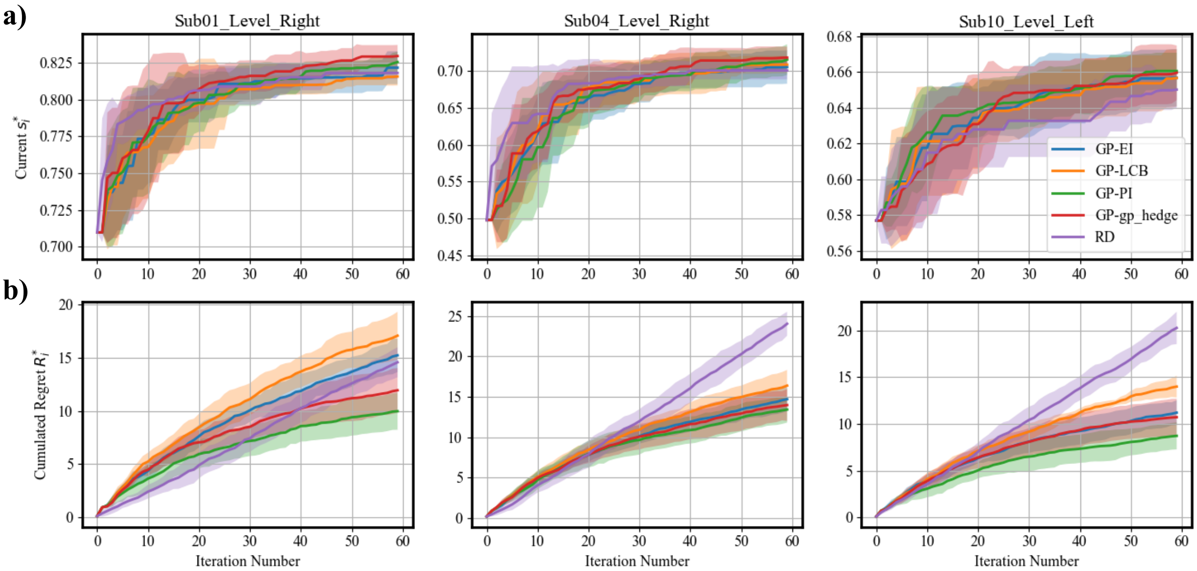


Figure B.1 - Convergence of different BO algorithms on 3 participant recordings from the 'Single Hand Grip' dataset (Appendix A) using GP surrogate models; **a)** Current performance score at the ith iteration; **b)** Cumulated regret at the ith iteration; Decoder (a STFT-band-CHS extractor paired with a LR classifier) were tuned 5 times for 60 iterations with all tuning methods (RD or GP combined with a given acquisition function); Solid line indicate a the ith iteration mean and shaded area indicate the ith iteration standard deviation across the 5 repetition of the optimisation process.

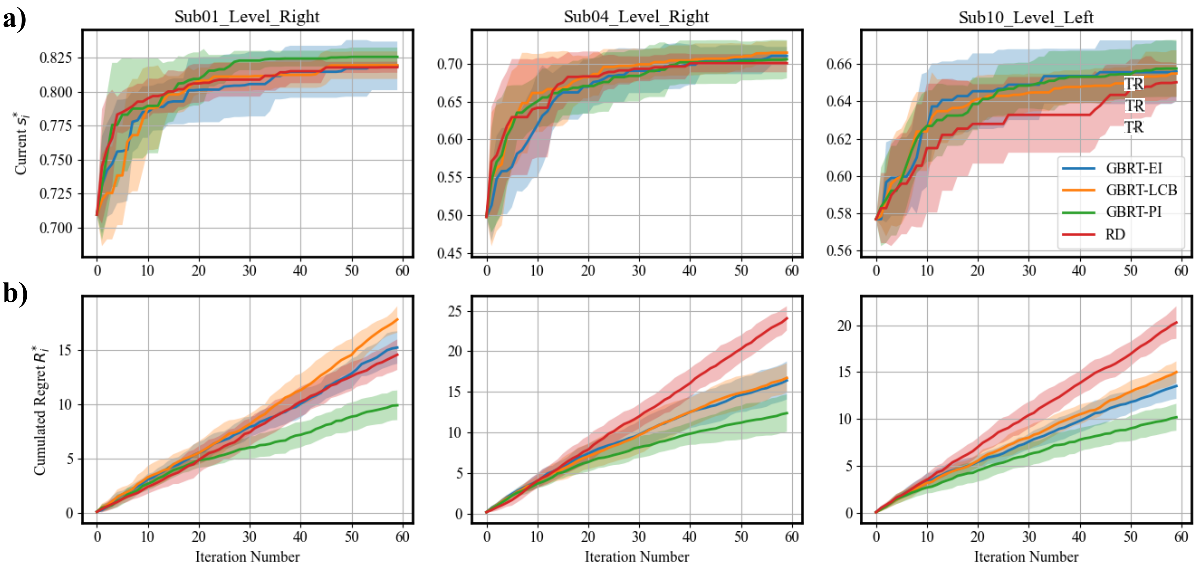


Figure B.2 - Convergence of different BO algorithms on 3 participant recordings using GBTR surrogate models; **a)** Current score at the ith iteration; **b)** Cumulated regret at the ith iteration; Decoder (a STFT-band-CHS extractor paired with a LR classifier) were tuned 5 times for 60 iterations with all tuning methods (RD or GBTR combined with a given acquisition function); Solid line indicate a the ith iteration mean and shaded area indicate the ith iteration standard deviation across the 5 repetition of the optimisation process.

Appendix C: Signal Processing

A. Example of Outlier Rejection Strategy

The proposed artefact rejection algorithm consisted in smoothing the rectified envelope of the pre-processed signal $x \in \mathbb{R}^{N_k \times N_{ch}}$ using an exponential average signal such that:

$$x_r[k] = (1-\lambda)x_r[k-1] + \lambda|x[k]| \tag{C-1}$$

where λ is a forgetting constant and x_r is the envelope signal. The robust covariance outlier algorithm implemented in Scikit-Learn (Pedregosa et al., 2011) was employed as an outlier detector. An example is given in Figure C.1.

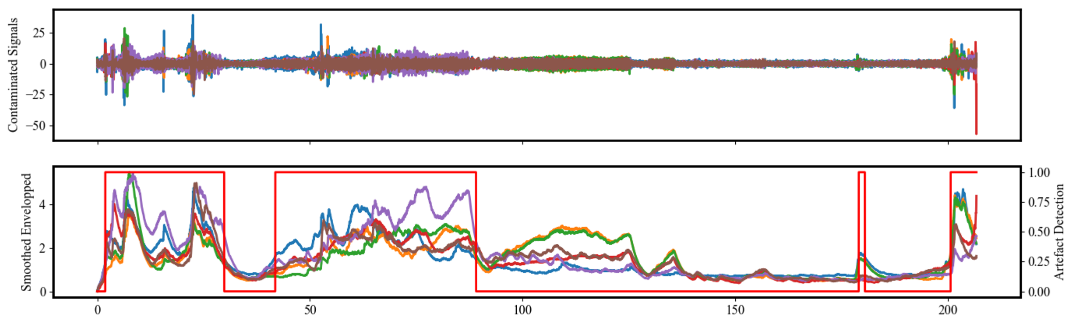


Figure C.1 - Example of artefacts and outlier detection for Sub07 of the 'Dual Hand Pinch' dataset (Appendix A.B); Recording is heavily polluted by large artefacts; Recording were pre-processed prior to the application of the artefact removal scheme; Solid red line indicates flags portion of the recording which are likely to be comminated by artefacts; λ was set to 0.999; Contamination was set to 0.44 in the robust covariance algorithm.

B. Direct Filter Formulation

For any arbitrary filter defined with the coefficient array pair(a, b), the update equation of a direct filter was given by:

$$\begin{aligned} x_{bd}[k] &= b[0]x[k] + z[0] \\ z[0] &= b[1]x[k] + z[1] - a[1]x_{bd}[k] \\ z[1] &= b[2]x[k] + z[2] - a[2]x_{bd}[k] \\ &\dots \\ z[N_a - 2] &= b[N_a - 1]x[k] - a[N_a - 1]\, x_{bd}[k] \end{aligned} \tag{C-2}$$

where x is the input signal, x_{bd} the filtered signal, N_a the size of a and z the memory of the filter. The direct form only required a single memory array z which could properly initiated to minimise edge effects. Array broadcasting was used to define the filter memory and compute stacks filters (a, b) across multiple dimensions in software (Harris et al., 2020). For instance, if N_{ch} channels and N_{bd} were required then $z \in \mathbb{R}^{(N_a-1)\times N_{ch}\times N_{bd}}$.

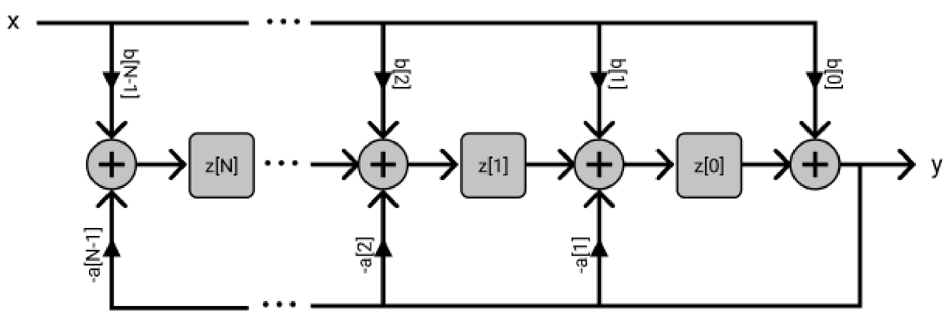


Figure C.2 – Simple direct realisation of linear filters used to implement IIR filter banks; This also includes a complex IIR filter bank for the RWT.

C. Epoch Buffer and Operators

The epoching operation was described as a simultaneous downsample and windowing of the signal. For a signal x the epoched signal x_e was given by:

$$x_e[k, \dots, l] = x[Rk - l] \ \forall \ l = 0,1,2 \dots L - 1 \qquad \text{(C-3)}$$

where R is the downsampling ratio, and L is the window epoch length. Managing the memory of the epoching process was less straightforward than a conventional filter. It was be achieved using a buffer array z. The following pseudo-code describes how x was intermittently stored in x_e:

Initiate x_e as an empty array

Initiate z as empty buffer array

for $k = 0, 1, 2 \dots N_k - 1$

    append $x[k]$ to z

    if size$(z) == L$

        append z to $x_e[k/R, \dots, 0{:}\,L - 1]$

        $z \leftarrow z[R{:}\,L - 1]$

Figure C.3 - Simple algorithm for buffering epoch windows; x is progressively cut by strides R using z; Every time z is L long, a new epoch is appended to x_e.

L was defined to be congruent to R to allow for chunk processing. Let $n_o \in \mathbb{N}$ be the number of overlaps between epoch windows such that:

$$L(R, n_o) = n_o R + 1 \qquad \text{(C-4)}$$

Given that the estimated power is temporally centred within the window, the transformed automatically introduces a temporal delay of half the extraction window length. Consequently n_o is constrained by τ_y such that:

$$n_o \leq \lfloor 2 f_d \tau_y \rfloor \tag{C-5}$$

For power extraction, n_o set L and helped control the smoothness at the expense of signal lag. n_o was thereby to be assigned as a hyper-parameter during model tuning. Several epoch operators were defined and employed in different feature extractors, such as the **1)** epoch mean:

$$\bar{x}_e[k] = \frac{1}{L(R, n_o)} \sum_{l=0}^{L(R,n_o)-1} x_e[k, \dots, l] \tag{C-6}$$

2) the epoch variance:

$$Var_e(x, R, n_o) = \frac{1}{L(R, n_o) - 1} \sum_{n=0}^{L(n_o)-1} (x_e[k, \dots, n] - \bar{x}[k])^2 \tag{C-7}$$

and **3)** squared sum of the epoch:

$$SS_e(x, R, n_o) = \sum_{n=0}^{L(R,n_o)-1} |x_e[k, \dots, n]|^2 \tag{C-8}$$

The difference C-6 and C-7 were, in practice, succinct. The first might be preferred for its formalism and the latter for its efficiency. Also, contrary to $\mathrm{Var_e}$ and $\mathrm{SS_e}$ is also defined when $R = 1$ and $n_o = 0$. Lastly, to remove spurious activity in epoch, Mace et al. (2014) also proposed the usage of generalised statistical filters where the square elements of the epoch x_e are ranked and then linearly combined. The most straightforward instance of statistical filters, i.e. the median operator, was tested, but no particular improvement over other epoch operators was observed.

D. QMF Lattice

The filter coefficient stored in the QMF filter defined the mother wavelet function ψ used for the decomposition. The QMF should use a short pair of real-valued FIR decomposition that minimise the overall latency and computational overhead. Wavelet functions that required long decomposition filters, such as the Meyer wavelet were not considered. The lattice QMF parameterisation proposed by Vaidyanathan, (1993) was implemented. The lattice structure is depicted in Figure C.4.

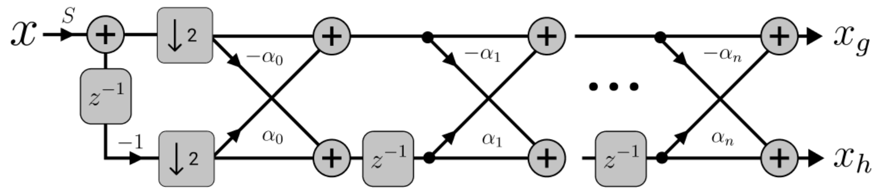


Figure C.4 - Lattice realisation of the QMF filter; After scaling every even timestep value of the incoming signal x is passed on to the upper branch and every odd value, to the lower branch; The resulting vector in sequentially rotated by an angle $\tan^{-1}(\alpha_i)$ at each lattice stage; The rotated vector forms the output pair of the filter.

It compacted two filters into one and allowed for an efficient computational realisation (Vaidyanathan, 1993). The lattice structure contained N_α rotation stages, each controlled by a parameter $\alpha_i \in \mathbb{R}$. The scaling factor s was given by:

$$s = \frac{s^*}{\prod_{i=0}^{N_\alpha - 1}(1 + \alpha_i^2)^{\frac{1}{2}}} \tag{C-9}$$

where $s^* \in \{-1,1\}$.

The following algorithm (presented in pseudo-code, Figure C.5) was designed to allow for efficient computation of the lattice filter realisation and efficient usage of the filter memory:

Initiate empty output arrays x_g, x_h of size $\frac{N_k}{2} - 1$

Initiate empty memory array z of size N_α

for $k = 0,2,4 \dots \frac{N_k}{2} - 1$

$z \leftarrow \text{cat}(-sx[k+1], z)$

<u>*Top branch computation*</u>

h $\leftarrow$ cumulated sum of $\text{cat}(sx[k], \alpha z)$

<u>*Bottom branch computation*</u>

$z \leftarrow -\alpha h[0{:}N_\alpha] + z$

$x_g[k/2] \leftarrow h[N_\alpha + 1]$

$x_h[k/2] \leftarrow z[N_\alpha]$

$z \leftarrow z[0{:}N_\alpha - 1]$

Figure C.5 - Proposed lattice realisation pseudo code; The intermediate computational nodes used to compute x_g are temporarily stored in a array h; h is used to update the memory state z and the lower branch of the lattice which yields x_h.

Again, broadcasting was employed to ease computation and manage parallel channel and hidden dimensions. The lattice algorithm was validated against a Simulink (The Math Works, Inc. MATLAB. Version 2017a) model. Using Matlab's (The Math Works, Inc. MATLAB. Version 2017a) wavelet and symbolic toolbox, lattice parameter solutions were found for common wavelets (db2, db3, db4, db5, and coif1, Table A.1). Backpropagation and BO strategies were studied to optimise the lattice parameters. To bound the search domain of the lattice parameters for BO, the following transformation was applied to each α_i coefficients:

$$\Theta_i = \tan^{-1}(\alpha_i) \tag{C-10}$$

A three-tap wavelet was employed. The first parameter was eliminated by using the constraint:

$$\Theta_0 = \frac{\pi}{4} - \Theta_1 - \Theta_2 \tag{C-11}$$

This yielded two tuning hyper-parameters $\Theta_1, \Theta_2 \in \left]-\frac{\pi}{2}, \frac{\pi}{2}\right[$ as defined in (Farina et al., 2007). $(28.64°, -6.04°)$ were used as default parameters as they were shown to parameterise the db3 wavelet (Rieder et al., 1998). It was also possible to use gradient backpropagation through the QMF lattice (Neretti & Intrator, 2009). Although this was achieved, the method required more validation and development.

Table C.1 – Approximated lattice parameters for Daubechies' (**db**) and coiflet (**coif**) wavelet estimated using symbolic computing in Matlab and fitting of known direct-form filter coefficients.

Mother Wavelet	s^*	Θ_0	Θ_1	Θ_2	Θ_3	Θ_4
db2	1	-60.00	15.00			
db3	1	-67.59	28.64	-6.04		
db4	1	-72.14	39.04	-14.54	2.63	
db5	1	-76.18	45.63	-22.38	7.33	-0.44
coif1	-1	77.85	69.30	-12.15		

E. Wavelet Packet Decomposition

Given a QMF node pruning function h_l at a level l, splitting a signal x into x_g, x_h (Rutledge, 1998). The decomposition in the packet implementation was achieved thanks to a local metric function M:

$$m_i = M(\mathrm{SS_e}(x_i, R_l, n_o), y_l)$$
$$h_l(x, x_g, x_h) = \begin{cases} x & \text{if } m \geq m_g + m_h \\ x_g, x_h & \text{if } m \leq m_g + m_h \end{cases} \tag{C-12}$$

where y_l is the label array correctly down-sampled to match the rate of the current and preceding levels. The choice of M was restricted to metrics conserving a meaningful interpretation after addition. The mutual information MI metric was chosen in this respect.

Moreover, it made sense to further take advantage of M to filter out any x, x_g, x_h which had little informative value. h_l was thereby be followed by a threshold function h_τ such that:

$$h_\tau(x_i) = \begin{cases} x_i \text{ if } m_i \geq \tau \\ \emptyset \text{ otherwise} \end{cases} \tag{C-13}$$

where τ is a selection threshold. If $h_\tau \circ h_l$ returned x, then x was correctly appended to the coefficient set $\mathcal{S}_c$. If it returned either x_g, x_h or both, those were sent to the next level for further decomposition. During model fitting, each node transformation function was stored as parameters without the need to re-evaluate M at a later stage (i.e., the decomposition was not dynamic). Lastly, the decomposition was achieved entirely independently in each channel. Therefore, the DWT-packet performed three functions simultaneously: channel selection, feature extraction, and feature selection.

F. ARMA RLS Update

The ARMA RLS filter update equations was derived from Tarvainen et al. (2004). First, the pre-fit residual was computed:

$$\tilde{e}[k] = x[k] - \varphi_k^T \tilde{\theta}[k-1] \tag{C-14}$$

where $\tilde{e} \in \mathbb{R}^{N_k}$ is the estimated residual error, $\tilde{\theta} \in \mathbb{R}^{N_k \times (N_{AR}+N_{MA})}$ parameter vector and $\varphi_k \in \mathbb{R}^{N_{AR}+N_{MA}}$ the observation vector at time k. The parameter vector was then updated:

$$\tilde{\theta}[k] = \tilde{\theta}[k-1] + K_k \tilde{e}[k] \tag{C-15}$$

where $K_k \in \mathbb{R}^{N_{AR}+N_{MA}}$ is the Kalman gain of the filter. For the RLS adaptive algorithm, K_k was then updated according to the parameter η:

$$K_k = \frac{P_{k-1}\varphi_k}{(\varphi_k^T P_{k-1} \varphi_k + \eta)} \tag{C-16}$$

where $P_k \in \mathbb{R}^{(N_{AR}+N_{MA}) \times (N_{AR}+N_{MA})}$ is the state covariance estimated at time k. P_k was itself updated according to:

$$P_k = \frac{1}{\eta}\left(I - K_k \varphi_k^T\right) P_{k-1} \tag{C-17}$$

Broadcasting (Harris et al., 2020) was employed to duplicate those operations across channel dimensions.

G. Random Walk Kalman Filter

The single feature Kalman filter update equations were derived from Gruenwald et al. (2017). For a feature signal and smoothed hidden stage signal $x, x_s \in \mathbb{R}^{N_k}$, the state variance P_k and Kalman gain K_k at every timestep were given by:

$$P_k = P_{k-1}(1 - K_{k-1}) + \sigma_w^2 \tag{C-18}$$

$$K_k = \frac{P_k}{P_k + \sigma_v^2} \tag{C-19}$$

The prediction of the hidden state was then updated:

$$x_s[k] = x_s[k] + K_k(x - x_s[k]) \tag{C-20}$$

Broadcasting (Harris et al., 2020) was employed to duplicate those operations across feature and channel dimensions.

H. Second Order Kalman Filter

The single feature second order Kalman update equations were derived from Yao et al. (2020). If $\tau = 1/f_d$, then the transition A and covariance Q_w were defined as:

$$A = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}$$
$$Q = \begin{bmatrix} \frac{\tau^3}{3} & \frac{\tau^2}{2} \\ \frac{\tau^2}{2} & \tau \end{bmatrix} \tag{C-21}$$

The state variance P_k and Kalman gain K_k were update such that:

$$P_k = AP_kA^T + r_\sigma Q \tag{C-22}$$

$$\begin{aligned} K_k^1 &= P_k[0,0]/(P_k[0,0] + 1) \\ K_k^2 &= P_k[1,0]/(P_k[0,0] + 1) \end{aligned} \tag{C-23}$$

The state update were then given by:

$$\hat{x}_s = x_s[k-1] + \tau v[k-1]$$
$$x_s[k] = \hat{x}_s + +K_k^1(x[k] - \hat{x}_s) \tag{C-24}$$
$$v[k] = v[k-1] + K_k^2(x[k] - v[k-1])$$

The state variance was finally updated a posteriori:

$$P_{k+1} = \left(I - \left[K_k^1, K_k^2\right]^T [1,0]\right) P_k \tag{C-25}$$

Broadcasting (Harris et al., 2020) was employed to duplicate those operations across feature and channel dimensions in software.

Appendix D: Example of Feature Signals

A. IIR Extractors

Figure D.1 presents the layout for the IIR-CHS extractor, and Figure D.2 typical feature signals obtained with this extractor in an action detection task for three different participants from the 'Single Hand Grip' dataset (Appendix A). Participants were chosen to represent the variability of signal response across the cohort (Sub01_Level_Right had a 'strong' LFP activity response, Sub04_Level_Right a 'medium' response, and Sub10_Level_Left had a weak response).

What first stood out was the strong ERS response in γ_2 (4 standard deviations above the average signal activity) for Sub01_Level_Right. The CHS algorithm concluded that the ERS was the most informative feature and thus placed the corresponding channel at the top of the selection list. A second selected channel contained a strong ERD response in β (1 standard deviation below average activity) followed by a marked rebound. γ_2 was also present but less marked (about 1 standard deviation above average activity). The strength of the γ_2 ERS, of the β ERD and their respective coincidence with action development (admitting the latency of the IIR filter) were very specific to Sub01_Level_Right. Feature signal responses in other participants were far more minute. γ_2 ERS and β ERD were also present in Sub04_Level_Right recordings in the two most informative channels. The peak γ_2 ERS, however, was significantly weaker (of the order of 1 standard deviation above average activity). Although the β ERD was relatively strong (1 standard deviation below average activity), its response was more transient and was only really sustained through half of the action. θ activity was also noted, with activity peaks at the onset of the action. Sub04_Level_Right possessed a response far more representative of most other recordings, with feature signal trial-average deviations raising or falling just on the edges of the average signal activity. Lastly, in some participants, such as Sub10_Level_Left, these deviations were not perceptible and hardly measurable, apart maybe for low-frequency θ activity. Such participants had thus a very weak recorded STN-LFP activity.

Nevertheless, this example still demonstrated that the CHS algorithm could sort channels effectively by prioritizing the channels with the strongest synchronisation patterns, whether these patterns were present within a single channel (such as for Sub04_Level_Righ) or dispersed across multiple ones (such as for Sub10_Level_Right). The IIR filter bank, followed by the power estimation and smoothing scheme, could correctly partition the LFP spectra into its fundamental components. The obtained feature signals were fairly smooth, and no excessive processing latency was noted.

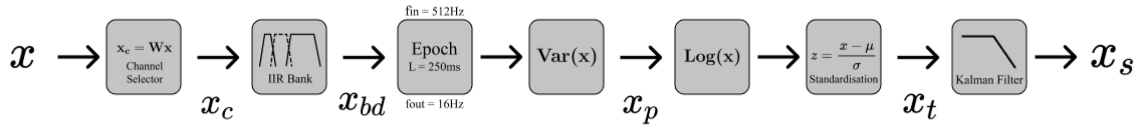


Figure D.1 - IIR-CHS extractor setup; Relevant channels are first selected from x (N_c out of N_{ch} channels are left into x_c); The IIR filter bank extracts different frequency bands from x_c (a total of $N_c N_{bd}$ band signals are extracted into x_{bd}); The signal is then epoched (using n_o overlaps); The epoch variance is used to estimate the signal power x_p; Log-transform and standardisation normalises the distributions in x_p; A Random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; fitting the extractor requires finding a mask array W to select channels, μ, σ for standardisation and σ_w for the Kalman filter; Tuning requires setting the parameters N_c, n_o (and λ in the case of a Box-Cox transformer).

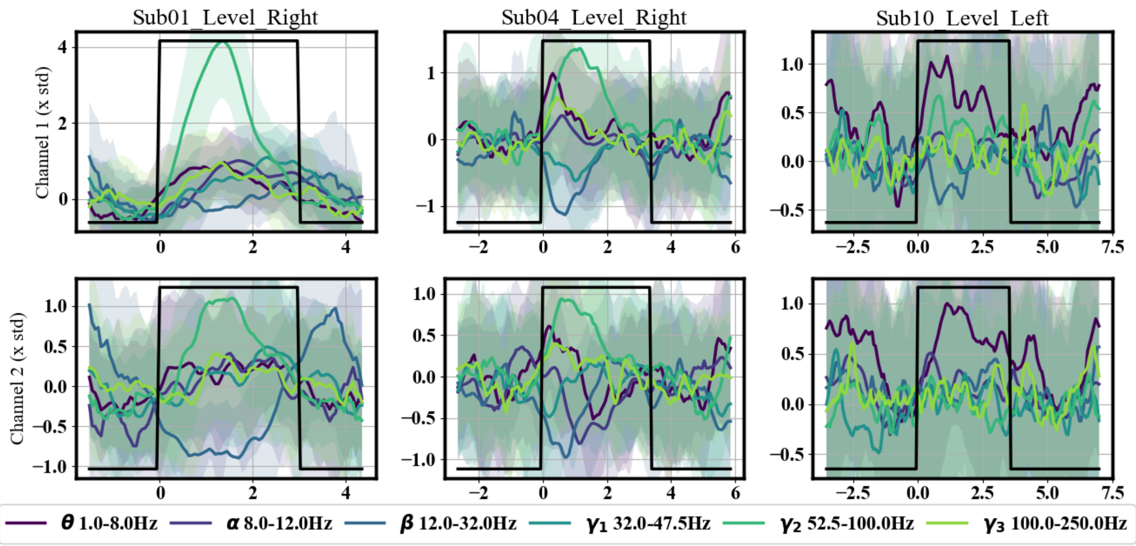


Figure D.2 - Example of trial average responses for the IIR-CHS extractor; 2 channels were selected ($N_c = 2$); An epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$) was used; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' (Appendix A.A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) line indicates the action to detect.

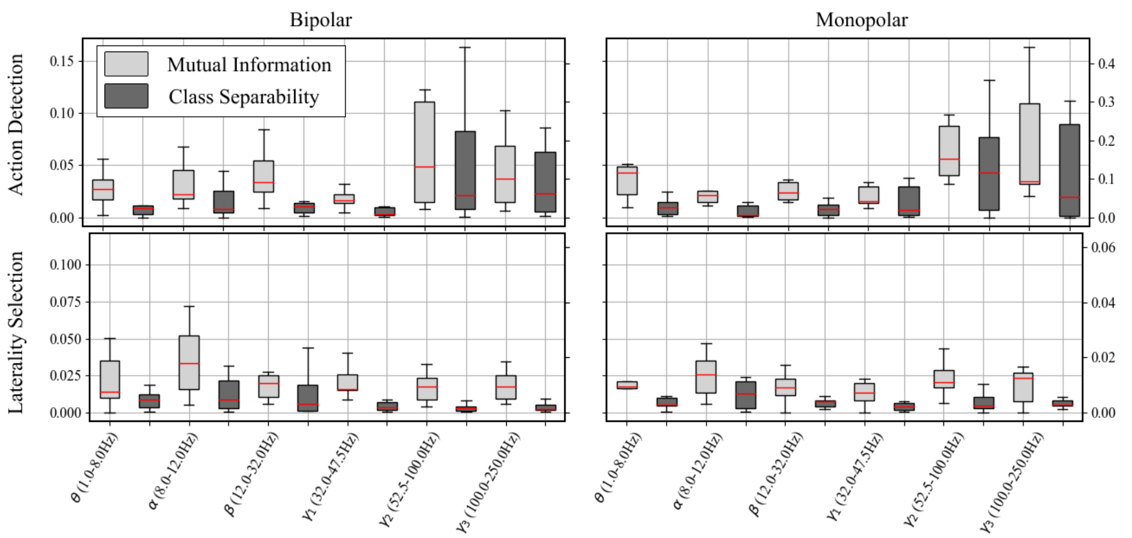


Figure D.3 - Pre-screening metric distributions (mutual-information and class-separability) measured with the IIR-CHS extractor; Metrics were extracted with the extractor default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A); measures were taken both in bipolar monopolar configuration; The maximum metric value was chosen between the two selected channels.

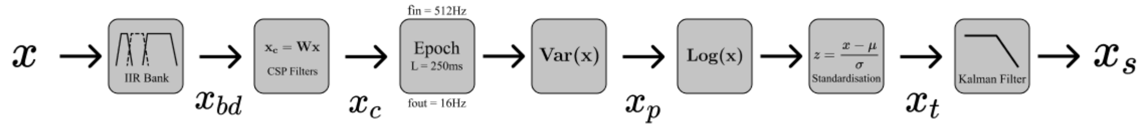


Figure D.4 - IIR-CSP extractor setup; The IIR filter bank extracts different frequency bands from x (a total of $N_{ch}N_{bd}$ band signals are extracted into x_{bd}); The bands are recombined by the CSP spatial filters ($N_{bd}(N_{c_0} + N_{c_1})$ components contained in x_c); The signal is then epoched (using n_o overlaps); The epoch variance is used to estimate the signal power x_p; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting the extractor requires finding a mask array W to select channels, μ, σ for standardisation and σ_w for the Kalman filter; Tuning requires setting the parameters N_{c_0}, N_{c_1}, n_o. (and λ in the case of a Box-Cox transformer).

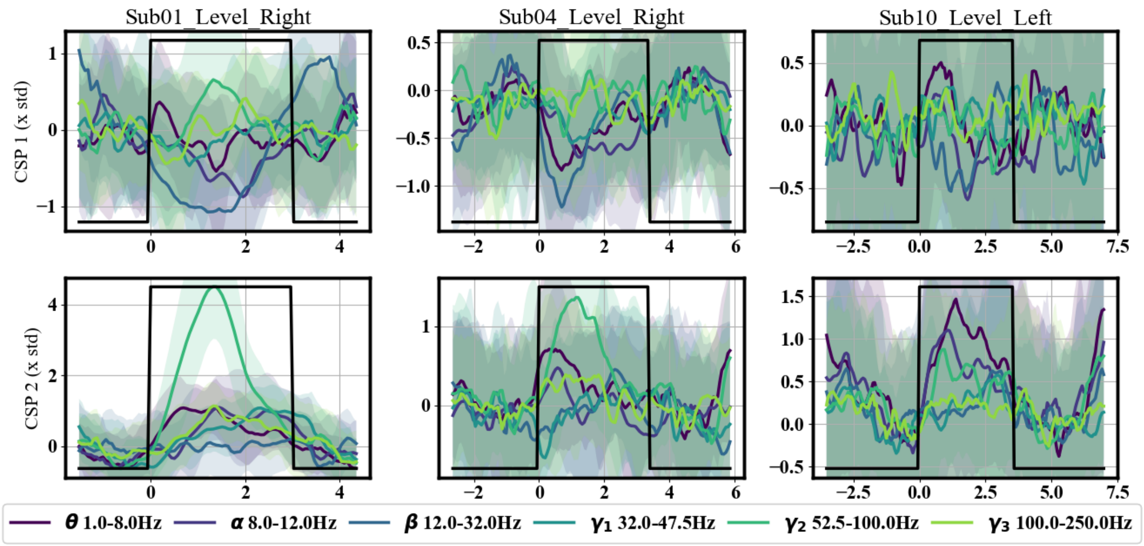


Figure D.5 – Example of trial average responses for the IIR-CSP extractor; one active and one rest components were selected ($N_{c_0} = 1, N_{c_1} = 1$); An epoch length of 250ms ($n_o = 3$, for $f_d = 16$Hz) was used; The extraction was completed for 3 representative subjects of the 'Single Hand Grip Recording' (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) line indicates the action to detect.

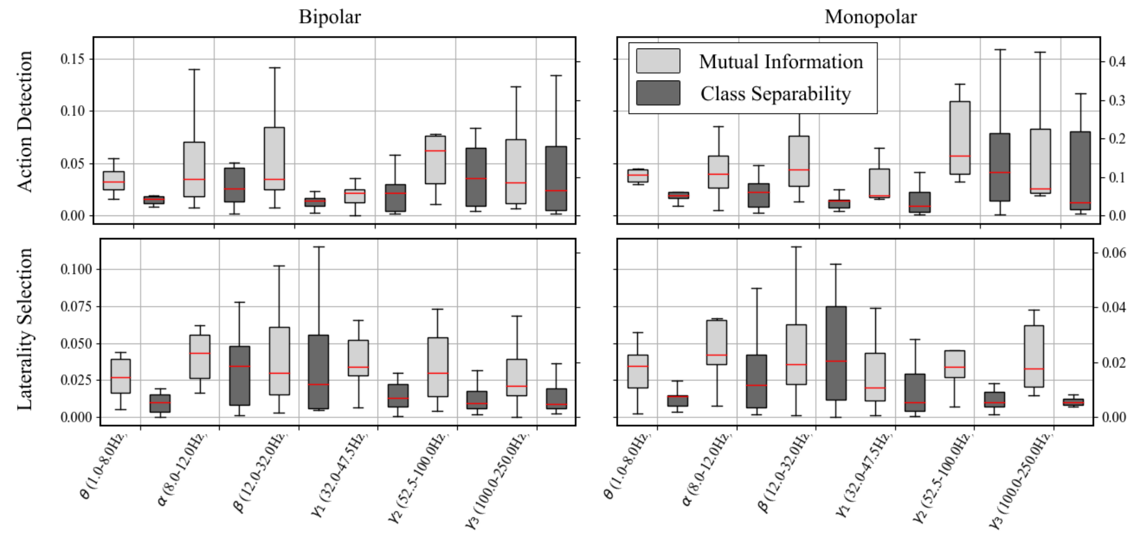


Figure D.6 - Pre-screening metric distributions (mutual-information and class-separability) measured with the IIR-CSP extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A); Measures were taken both in bipolar monopolar configuration; The maximum metric value was chosen between the two extracted CSP components.

The distribution of metrics measuring the relationship to the state to be decoded (Chapter 4.C) is also presented in Figure D.3. These metrics indicated that, overall, in the entire cohort, the narrow band fine γ_2 was the most informative feature (or at least the one feature with the best SNR) for action detection, followed by high-frequency γ_3 (albeit no significant ERS was observed in the three examples given). θ, α and β activity seemed to provide some information but to a far lesser extent. It was noted that low γ_1 contained no information relevant to decoding. For laterality selection, α bad activity was the only truly informative feature, which was a particular shortcoming of this extractor. It was lastly conjectured that bipolar settings appeared to isolate better low-frequency whilst being more destructive to higher frequencies.

Figure D.4 presents the layout for the IIR-CSP extractor, and Figure D.5 typical feature signals obtained with this extractor in an action detection task. The CSP filter could sort ERD and ERS according to 'rest' (C_o) or 'active' (C_1) components respectively (Chapter 4.C, in the case of laterality selection C_o components correlate with left action and C_1 with right actions). For Sub01_Level_Right and Sub04_Level_Right, β and α ERD were assigned to the 'rest' CSP filter. γ_2 and θ ERS were assigned with the 'active' filter. No ERD was observed in Sub10_Level_Left whilst the θ and γ_2 ERS were somewhat visible. CSP filtering did not appear to have affected either the temporality of the feature signal or their respective SNR. On the contrary, spatial filtering resulted in more informative features (Figure D.6). For action detection, α and β features were more informative and more comparable to γ_2 and γ_3 features for action detection. $\alpha, \beta, \gamma_1, \gamma_2$, and γ_3 with CSP decomposition were also informative of laterality. The information gain across multiple features was a significant advantage of the IIR-CSP method. Again the noted drawback was that all channels must be passed through the IIR bank before CSP mixing, whereas with CHS, once the selection is completed, only a restricted number of channels need to be partitioned.

B. STFT Extractors

Figure D.7 presents the layout for the STFT-band-CHS extractor, and Figure D.8 typical feature signals obtained with this extractor in an action detection task. Overall, the extracted feature signal exhibited similar relationships to the decoded target that those extracted with the IIR-CHS extractor. However, it was noted that the produced signals were distinctively more irregular and noisier (despite utilising a 250ms extraction window). The peaks of identifiable ERS/ERD patterns were less prominent (just above 3 standard deviations for γ_2 ERS and below 0.5 standard deviations for the β ERD). Nonetheless, response transients appeared to be shorter and more abrupt, particularly for the extraction of lower-frequency activity in the θ band. When inspecting relationship metrics (Figure D.9), α, β and γ_2 stood again as the most informative features for action detection, and the α, β and γ_1 the most important for laterality selections.

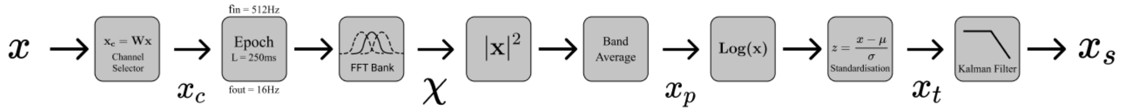


Figure D.7 - STFT-band-CHS extractor setup; Relevant channels are first selected from x (N_c out of N_{ch} channels are left into x_c); x_c is then epoched (using n_o overlaps); Each epoch is projected upon a Fourier basis to form χ; The signal power x_p estimated by taking the square complex norm of the resulting Fourier coefficients; Band activity is estimated by averaging together coefficients belonging to a pre-defined band interval; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting the extractor requires finding a mask array W to select channels, μ, σ for standardisation and σ_w for the Kalman filter; tuning requires setting the parameters N_c, n_o, β_k which controls the window shape (and λ in the case of a Box-Cox transformer).

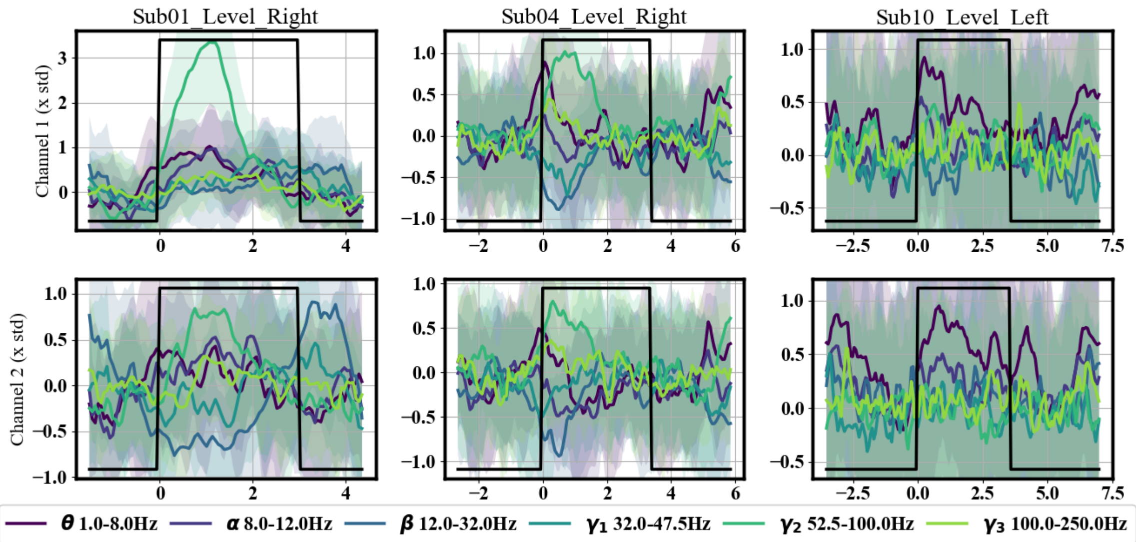


Figure D.8 - Example of trial average responses for the STFT-band-CHS extractor; 2 channels were selected ($N_c = 2$); an epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$) was used; The window shape factor β_k was set to 5; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; solid (**Black**) lines indicate the action to detect.

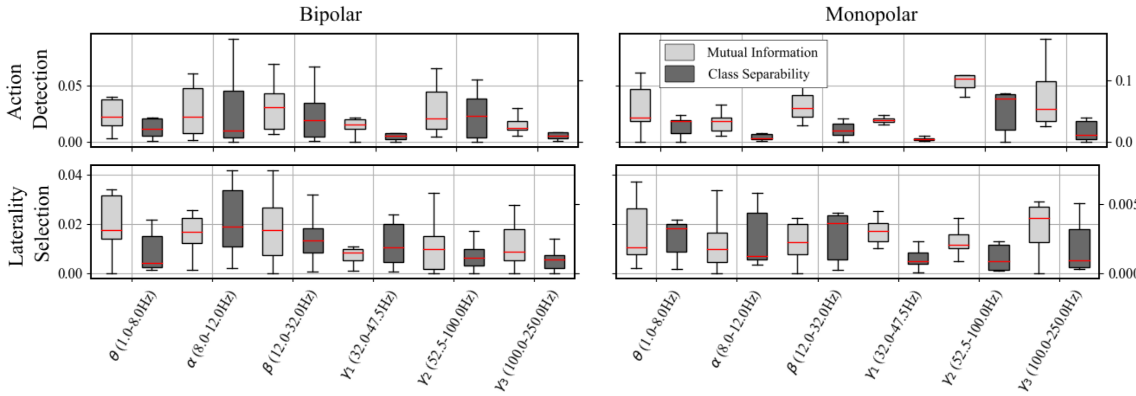


Figure D.9 - Pre-screening metric distributions (mutual-information and class-separability) measured with the STFT-band-CHS extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip Recordings' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A); Measures were taken both in bipolar monopolar configuration; The maximum metric value was chosen between the two selected channels.

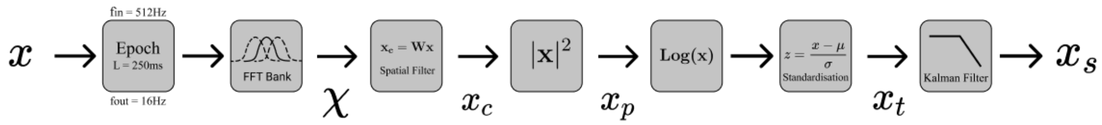


Figure D.10 - STFT-component-CSP extractor setup; x is directly epoched (using n_o overlaps); Each epoch is projected upon a Fourier basis for every N_{ch} channels to form χ; All resulting Fourier coefficients across all channels ($(L(n_o)+1)/2 \times N_{ch}$) are mixed by a CSP spatial filter matrix to extract; x_p is then obtained by selecting $N_{c_0} + N_{c_1}$ components from x_c and taking the square complex norm; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting requires finding the CSP mixing matrix W, μ, σ for standardisation and σ_w for the Kalman filter; Tuning requires setting the parameters N_{c_0}, N_{c_1}, n_o, β_k which controls the window shape, (and λ in the case of a Box-Cox transformer).

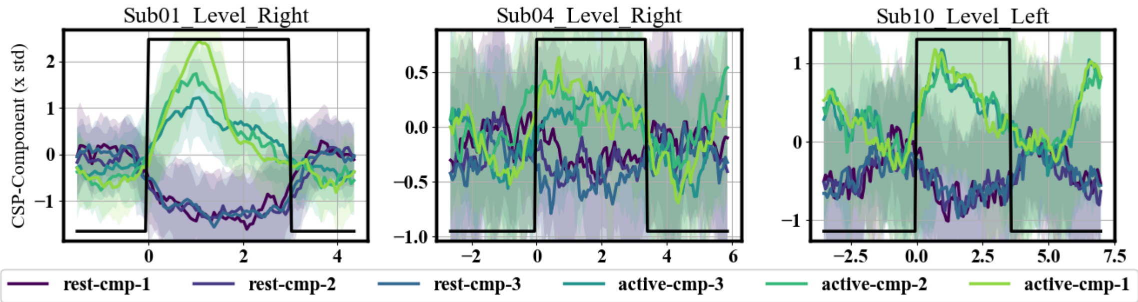


Figure D.11 - Example trial average response extracted for the STFT-component-CSP extractor; One active and one rest components were selected ($N_{c_0} = 3, N_{c_1} = 3$); An epoch length of 250ms ($n_o = 3$, for $f_d = 16$Hz) was used; The window shape factor β_k was set to 5; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) lines indicate the action to detect.

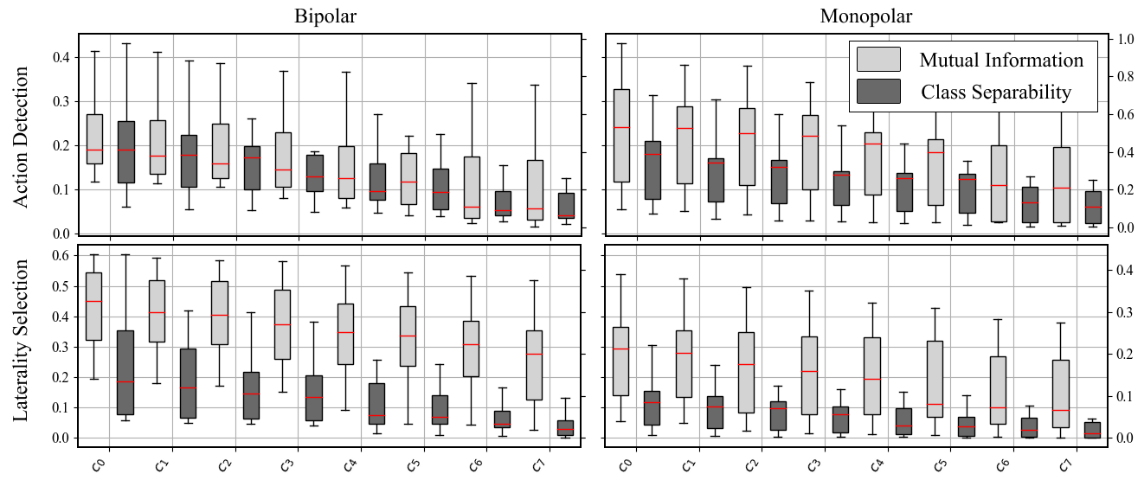


Figure D.12 - Pre-screening metric distributions (mutual-information and class-separability) measured with the STFT-component-CSP extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A); Measures were taken both in bipolar monopolar configuration; All CSP components are reported.

Figure D.10 presents the layout for the STFT-component-CSP extractor, and Figure D.11 typical feature signals obtained with this extractor in an action detection task. This complex CSP variant was fundamentally different from that of the IIR extractor, as the spatial filters remixed the complete signal decomposition in χ rather than individual frequency bands across different channels. The CSP was capable of constructing components matching the action temporality, in all three examples, with 'rest' components fully depressed throughout the action and 'action' components sustained. Peak activity in active components also seemed to coincide with the action onset. Peak responses were found to be significantly lower (at best 2 standard deviations above for active components and 1 standard deviation below for rest components) than what was measured in signals extracted with the IIR method; irregularities previously associated with the STFT were also present, and the SNR was more degraded in weaker responses. Although the noise aspect of the feature may not have been enhanced, the signal was overall reinforced. The top extracted components were far more informative than any other feature proposed (Figure D.12). It was also conjectured that component informativeness diminished monotonously (alternating rest and active components) with its ranking. Considering preliminary evidence, the proposed extraction algorithm was very promising as it solved three important problems: the transformation of the raw signal into an interpretable component space closely relatable to a variable to be decoded; good generalisation across participants; and a combination of an efficient extraction algorithm with a sparse representation of the signals.

C. RWT Extractors

Figure D.13 presents the layout for the RWT-band-CHS extractor, and Figure D.14 typical feature signals obtained with this extractor in an action detection task. Although the produced feature signals resembled those of the IIR-CHS extractor at first glance, two main differences were noted. The feature signals were smoother, with good overall peak response across every example. Low-frequency resolution was improved in comparison to IIR and STFT methods. The θ band was most informative on action detection or laterality selection (Figure D.15). For Sub01_Level_Right, the first channel selected was not the one containing the very large γ_2 ERS but the one containing a strong θ ERS. These ERS also stood from other band activity in Sub10_Level_Left. In addition to its compact recursive computation form, the quality of the features produced by the RWT extractor made it an interesting alternative to the IIR and STFT methods.

Figure D.16 presents the layout for the RWT-component-CSP, the same complex CSP variant previously applied to the STFT algorithm. Figure D.17 presents typical feature signals obtained with this extractor in an action detection task. Similar results to that of the STFT-component-CSP were obtained with yet again some noted differences. High ERS and ERD peaks were maintained in Sub01_Level_Right and Sub04_Level_Right. However, the temporality of the components did not re-

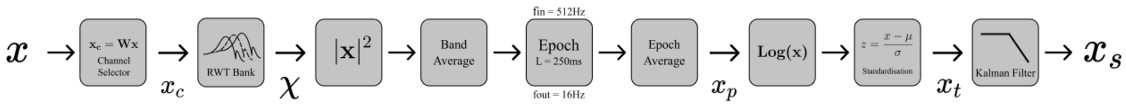


Figure D.13 - RWT-band-CHS extractor setup; Relevant channels are first selected from x (N_c out of N_{ch} are left in x_c); The signal is then project upon a wavelet basis (32 wavelet per octave) using a complex IIR filter bank to form χ; x_p is estimated by taking the square complex norm of χ; Epoch averages of power signals are taken for down sampling and smoothing; Band activity is estimated by averaging together coefficients belonging to a pre-defined band interval; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting the extractor requires finding a mask array W to select channels, μ, σ for standardisation and σ_w for the Kalman filter; Tuning requires setting the parameters N_c, n_o (and λ in the case of a Box-Cox transformer).

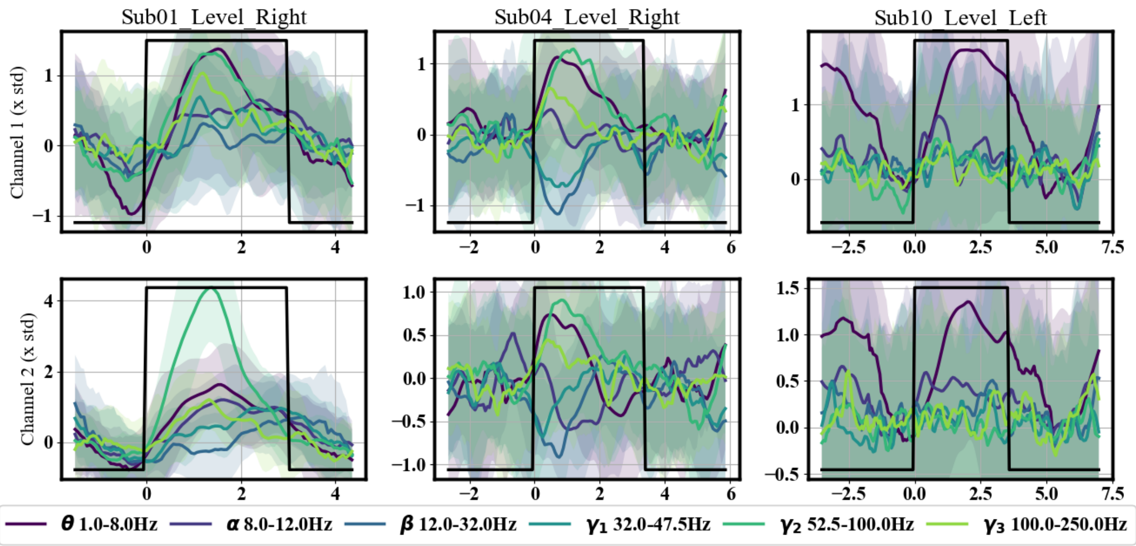


Figure D.14 - Example trial average response for the RWT-band-CHS extractor; 2 channels were selected ($N_c = 2$); an epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$); 32 wavelet per octave were used; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) lines indicate the action to detect.

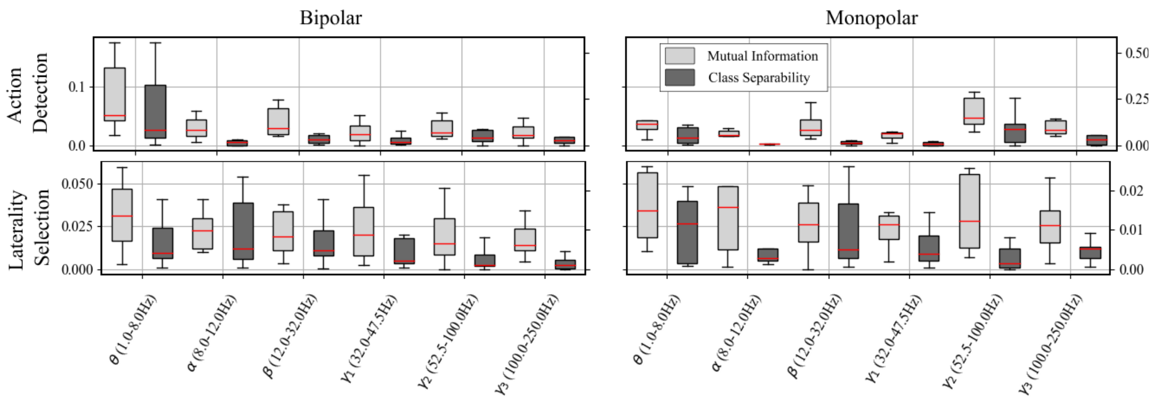


Figure D.15 - Pre-screening metric distributions (mutual-information and class-separability) measured with the RWT-band-CHS extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A); Measures were taken both in bipolar monopolar configuration; The maximum metric value was chosen between the two selected channels.

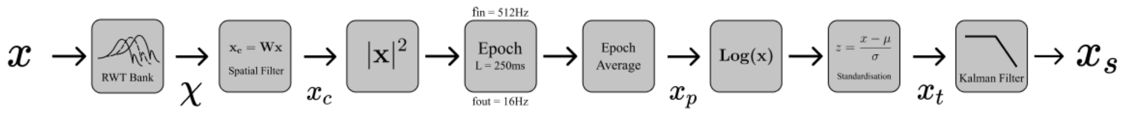


Figure D.16 - RWT-component-CSP extractor setup; x is directly projected upon a wavelet basis (32 wavelet per octave) using a complex IIR filter bank to form χ; The wavelet coefficients in χ are mixed by a CSP spatial filter matrix across all channels ($N_l N_{ch}$); x_p is then obtained from the selected $N_{c_o} + N_{c_1}$ components from x_c, taking the square complex Epoch averages of power signals are taken for down sampling and smoothing; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting requires finding the CSP mixing matrix W, μ, σ for standardisation and σ_w for the Kalman filter; tuning requires setting the parameters N_{c_o}, N_{c_1} and n_o (and λ in the case of a Box-Cox transformer).

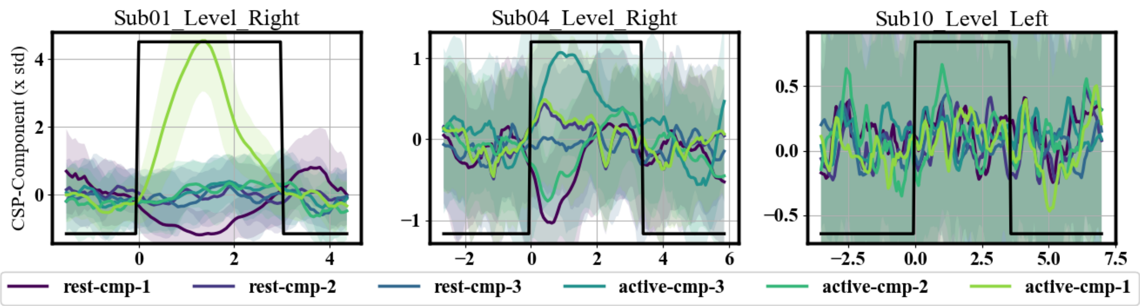


Figure D.17 - Example trial average response extracted for the RWT-component-CSP extractor; One active and one rest components were selected ($N_{c_0} = 3, N_{c_1} = 3$); An epoch length of 250ms ($n_o = 3$, for $f_d = 16$Hz) was used; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) lines indicate the action to detect.

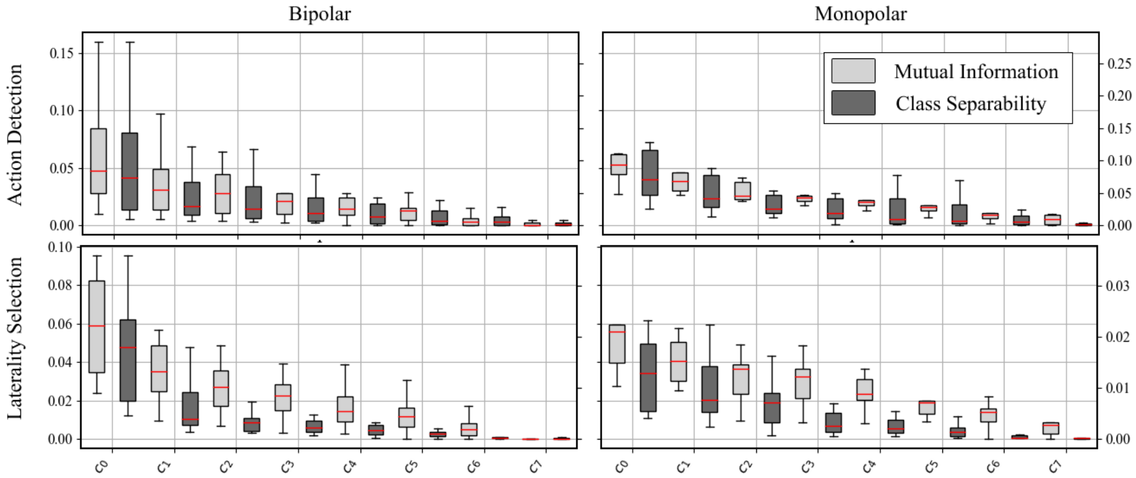


Figure D.18 - Pre-screening metric distributions (mutual-information and class-separability) measured with the RWT-component-CSP extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A); Measures were taken both in bipolar monopolar configuration. All CSP components are reported.

align as much as it previously did with the STFT-component extractor but instead seems to accentuate the ERS/ERD patterns observed in the RWT-band-CSP. Surprisingly, none of the components generated for Sub10_Level_Left deviated significantly from the base activity. Decay of the component informativeness was faster than that of the STFT-component-CSP function (Figure D.18). These last observations suggests that the RWT-component-CSP could be a more effective compressive algorithmic combination, less prone to overfitting the signal representation χ and thus yield noise-shaped components. Alternatively, it could also mean that the CSP application is more destructive than other methods.

D. DWT Extractors

Figure D.19 presents the layout for the DWT-wavelet -CHS extractor, and Figure D.20 typical feature signals obtained with this extractor in an action detection task; Figure D.22 and Figure D.23 that of the DWT-wavelet-CSP. Both extractors operated very much as the IIR-CHS and IIR-CSP, respectively. The signal decomposition instead occurred by recursively filtering the lower-frequency output x_g. The yield coefficient signals mirrored the previously obtained band signals: level 1 contained high γ_3 activity; level 2, mid γ_2; level 3, γ_1; level 4, β; the details on level 5, α; the approximation on level 5, θ. In all, the obtained feature signal quality and temporality were comparable to what was obtained with the IIR-based extractors, possibly with lesser pronounced ERS and ERD peak responses and sharper prominence of the lower frequency components in the last approximation coefficient. Information distribution across wavelet coefficients (Figures D.21 and D.24) was also comparable to what was previously obtained with the other extractors. The point of interest for using the DWT was again to use a parameterised wavelet shape to adapted neuronal signal signature, low computation cost and use alternative tree decomposition structures (Chapter 3.E).

Figure D.25 presents the layout for the DWT-full-CSP extractor, and Figure D.26 typical feature signals obtained with this extractor in an action detection task. Like with the STFT, the full-tree DWT decomposition spanned the frequency space linearly. Here a CSP was again used to mix all resulting coefficients. Unlike with the DWT-wavelet or DWT-packet decompositions, all coefficients were completely downsampled to f_d. The resulting feature signals were similar to those obtained with the STFT-component-CSP (even more so than the RWT-component-CSP). The same monotonic decay in component informativeness was also noted (Figure D.27). The fact that similar results could be obtained with a real (as opposed to complex) decomposition and a mixing matrix across both channel and coefficient dimensions made this variant particularly interesting, especially noting that the efficiency of implementation of DWT algorithms could be later leveraged.

Figure D.28 presents the layout for the DWT-packet extractor, and Figure D.29 typical feature signals obtained with this extractor in an action detection task. The DWT packet differed from all other

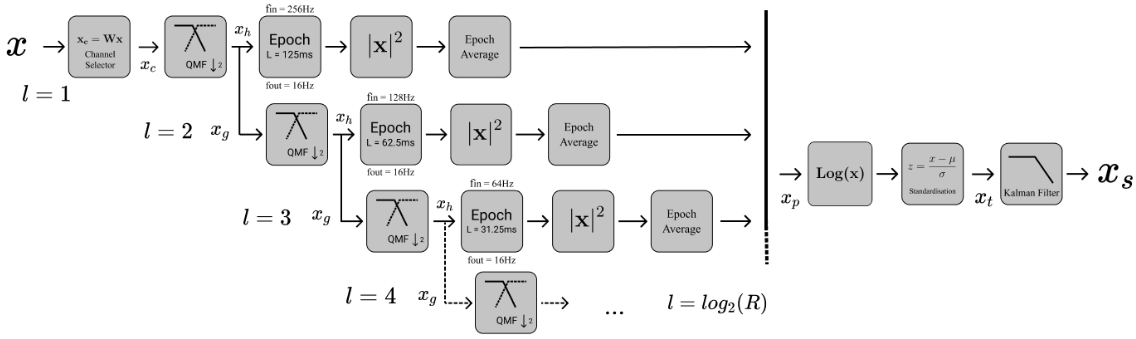


Figure D.19 - DWT-wavelet-CHS extractor setup; Relevant channels are first selected from x (N_c out of N_{ch} are left in x_c); x_c is first split using a QMF pair on level $l = 1$; Low-frequency signal x_g is passed on to next level $l + 1$ to be further decomposed; The high-frequency detail signal x_g is epoched (using n_o overlaps) at an input frequency $\frac{f_d}{2^l}$, squared and averaged; The decomposition of x_g occurs on $\log_2(R)$ levels; All power signals coefficients are collected into x_p; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting the extractor requires finding a mask array W to select channels, μ, σ for standardisation and σ_w for the Kalman filter; Tuning requires setting the parameters N_c, n_o, Θ, which controls the wavelet shape (and λ in the case of a Box-Cox transformer).

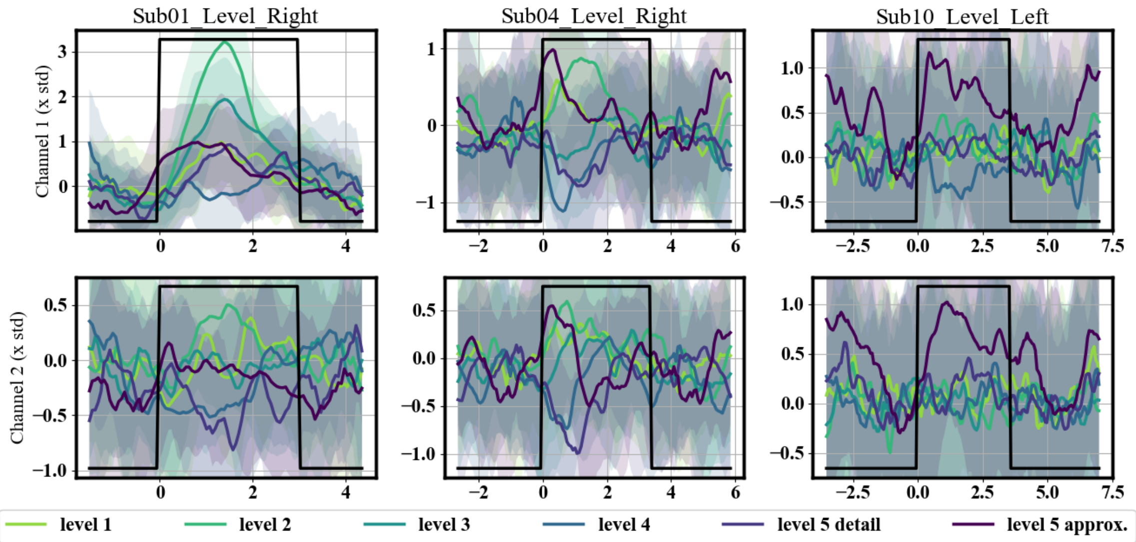


Figure D.20 - Example trial average response for the DWT-wavelet-CHS extractor; 2 channels were selected ($N_c = 2$); an epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$) was used; Θ was set to reproduce a db3 wavelet; The extraction was completed for 3 representative subjects of the 'Single Hand Grip R' (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) lines indicate the action to detect.

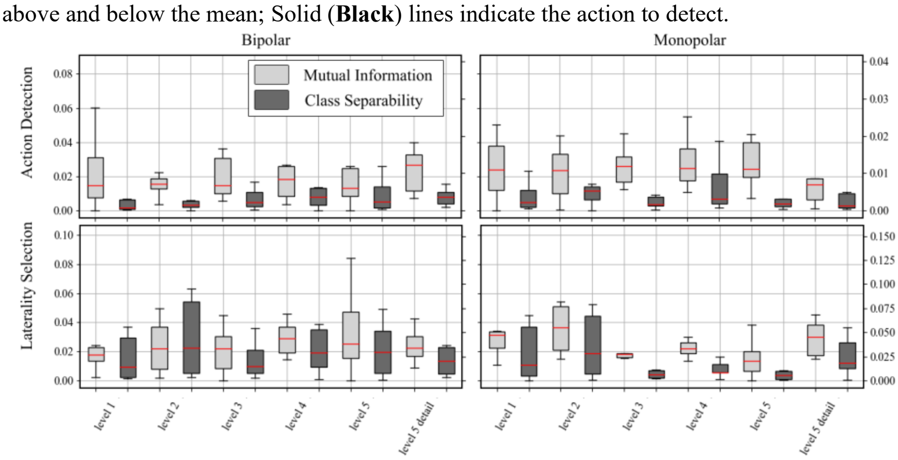


Figure D.21 - Pre-screening metric distributions (mutual-information and class-separability) measured with the DWT-wavelet-CHS extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A).

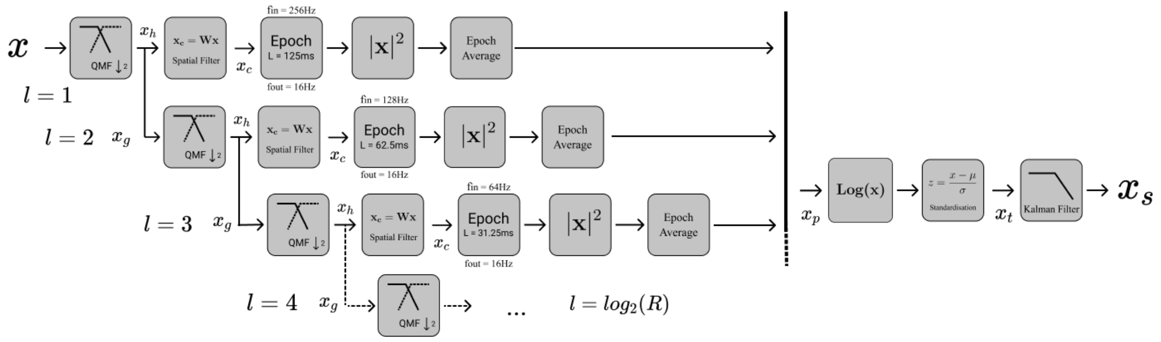


Figure D.22 - DWT-wavelet-CSP extractor setup; x is first split using a QMF pair on level $l = 1$; Low-frequency signal x_g is passed on to next level $l + 1$ to be further decomposed; The high-frequency detail signal x_g is decomposed into x_c using a CSP filter and epoched (using n_o overlaps) at an input frequency $\frac{f_d}{2^l}$, squared and averaged; The decomposition of x_g occurs on $\log_2(R)$ levels; All power signals coefficients are collected into x_p; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting requires finding the CSP mixing matrix W, μ, σ for standardisation and σ_w for the Kalman filter; Tuning requires setting the parameters N_c, n_o, Θ, which controls the wavelet shape (and λ in the case of a Box-Cox transformer).

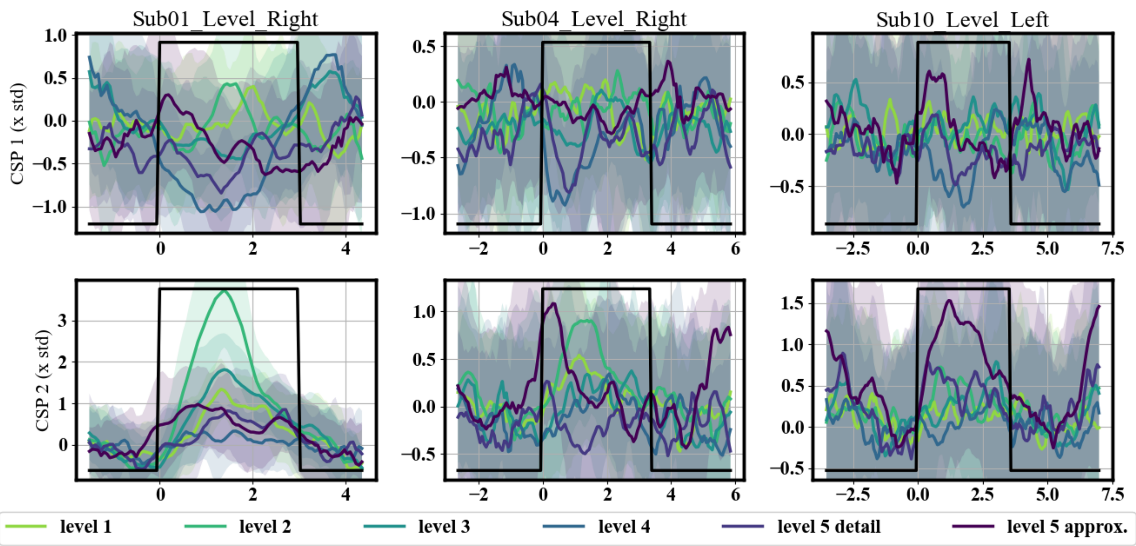


Figure D.23 - Example trial average response for the DWT-wavelet-CSP extractor; One active and one rest components were selected ($N_{c_0} = 1, N_{c_1} = 1$); An epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$) was used; Θ was set to reproduce a db3 wavelet; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' dataset (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) lines indicate the action to detect.

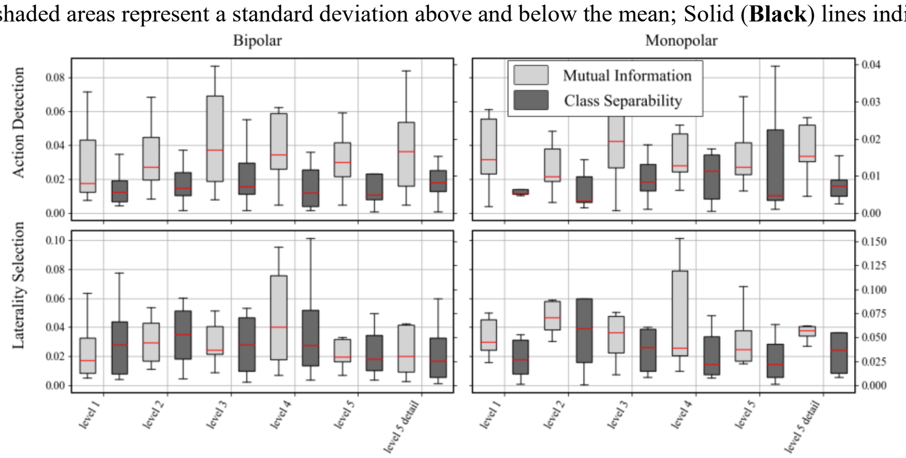


Figure D.24 - Pre-screening metric distributions (mutual-information and class-separability) measured with the DWT-wavelet-CSP extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A).

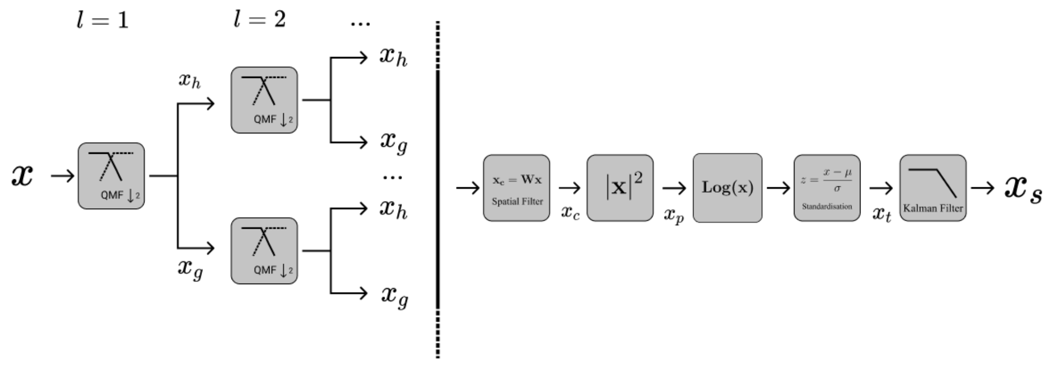


Figure D.25 - DWT-full-CSP extractor setup; x is first split using a QMF pair on level $l = 1$; High and low frequency coefficients x_g and x_h occurs at every level expanding the tree by 2^l coefficients until $\log_2(R)$ levels have been reached; All coefficients across all channels are decomposed using a CSP filter; x_c is squared to obtain x_p; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting requires finding the CSP mixing matrix W, μ, σ for standardisation and σ_w for the Kalman filter; tuning requires setting the parameters N_c, n_o, Θ, which controls the wavelet shape (and λ in the case of a Box-Cox transformer).

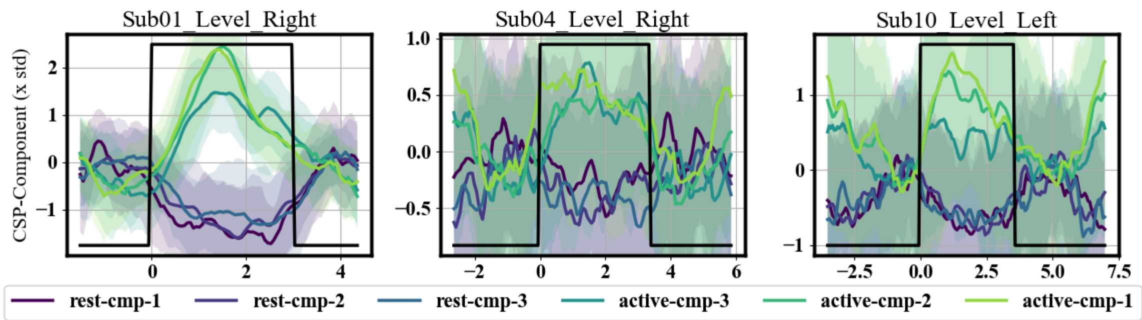


Figure D.26 - Example trial average response for the DWT-full-CSP extractor; One active and one rest components were selected ($N_{c_0} = 0, N_{c_1} = 1$); An epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$) was used; Θ was set to reproduce a db3 wavelet; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' dataset (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) line indicates the action to detect.

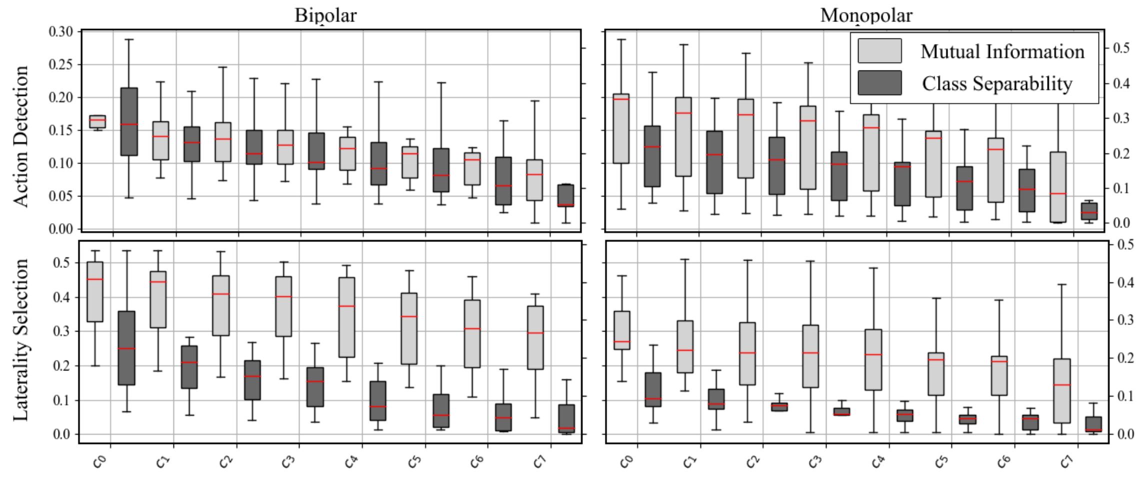


Figure D.2 -7 Pre-screening metric distributions (mutual-information and class-separability) measured with the DWT-full-CSP extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A); Measures were taken both in bipolar monopolar configuration; All CSP components are reported.

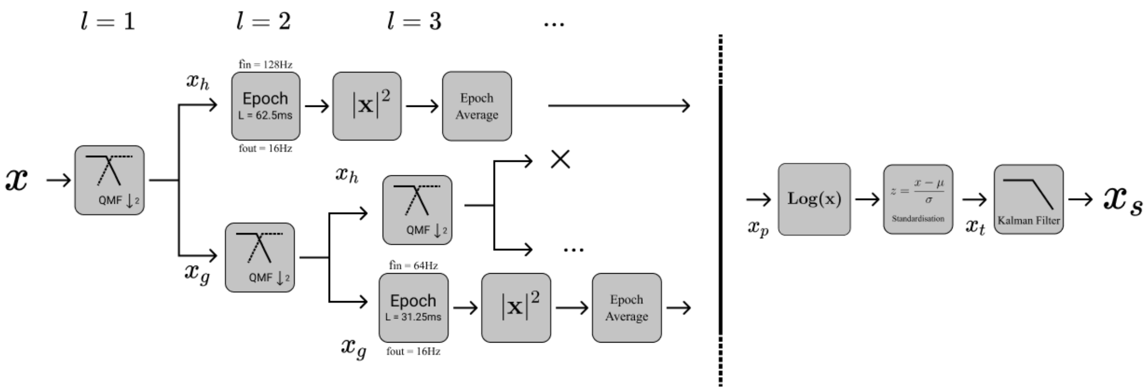


Figure D.28 - DWT-packet extractor setup; For every N_{ch} channels x is first split using a QMF pair on level $l = 1$; Each tree node is decomposed adaptively according to the data label y and a metric function M to halt decomposition when new information ceases to be produced; Un-informative tree leaves are also directly pruned through thresholding (×); Remaining coefficients were epoched (using n_o overlaps) at an input frequency $\frac{f_d}{2^l}$, squared and averaged; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting requires finding μ, σ for standardisation and σ_w for the Kalman filter; Tuning requires setting the parameters N_c, n_o, Θ, which controls the wavelet shape, τ, the pruning threshold (and λ in the case of a Box-Cox transformer).

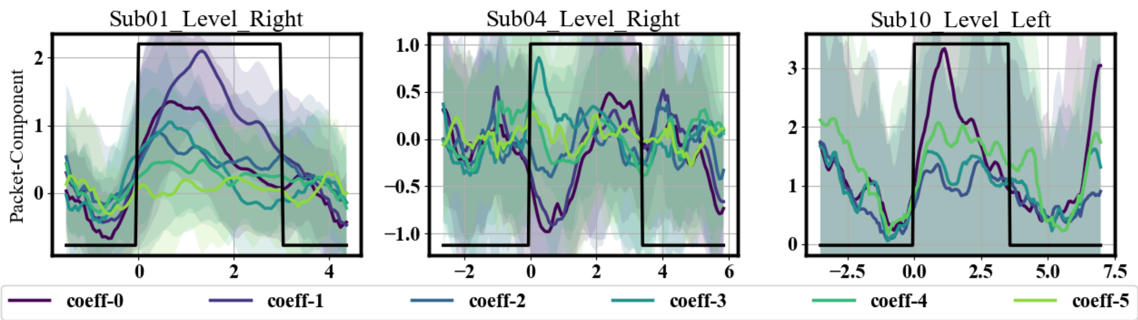


Figure D.29 - Example trial average response for the DWT-packet extractor; mutual information with y was used has metric function; τ was set to 0 (no thresholding); Θ was set to reproduce a db3 wavelet; the extraction was completed for 3 representative subjects of the 'Single Hand Grip' dataset (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid **(Black)** line indicates the action to detect.

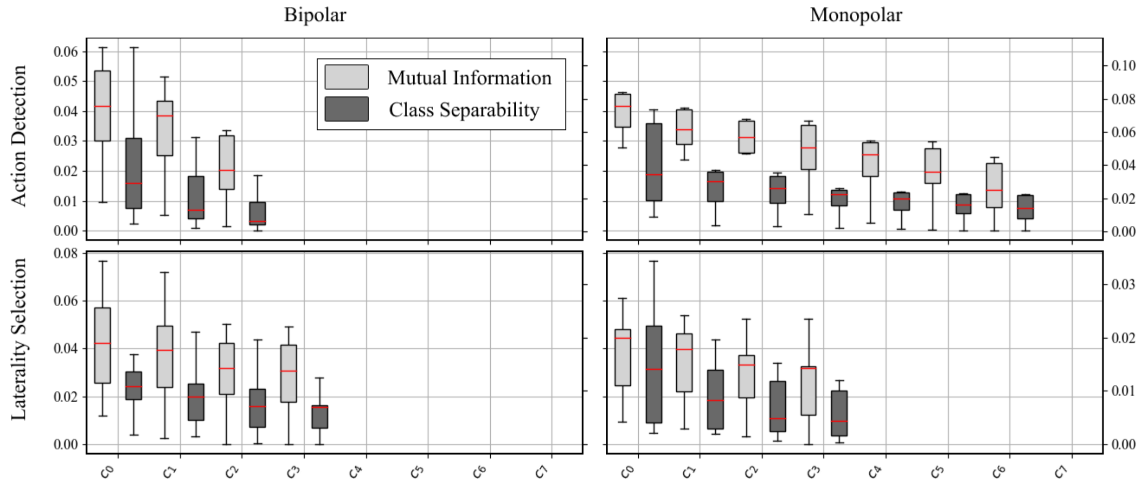


Figure D.30 - Pre-screening metric distributions (mutual-information and class-separability) measured with the DWT-packet extractor; Metrics were extracted with the default settings (see above) for every feature (only if every coefficient was available for every participant) and for every participant in the 'Single Hand Grip Recordings' dataset for action detection and in the 'Dual Hand Grip Recordings' dataset for laterality selection (Appendix A); Measures were taken both in bipolar monopolar configuration; Only distributions wehre coefficients were available for all participant recordings are shown.

methods, as channel selection, feature selection and decomposition were all performed conjointly through a sparse decomposition tree. The coefficients selected varied differently from one case participant to another. For Sub01_Level_Right, only ERS patterns were captured; for Sub04_Level_Right, the θ ERS and β ERD, and Sub10_Level_Left, the θ ERS. Coefficients were ranked according to their information content (Figure D.30). Pre-screening metric strengths were more similar to extractors that used the CHS method rather than CSP.

E. ARMA Extractors

Figure D.31 presents the layout for the ARMA-band-CHS extractor, and Figure D.32 typical feature signals obtained with this extractor in an action detection task. Although more irregular, under default settings, the temporal layout of the signals was comparable to that obtained with the IIR-CHS with the same selected top channels. ERS peaks were even larger (the γ_2 peak reached 6 standard deviations above average activity for Sub01_Level_Rights). Feature informativeness was notably improved in the β and α bands for action detection and in γ_1 and β for laterality selection (Figure D.33). This extractor estimated using an RLS adaptive filter an instantaneous parametric representation of the signal $\bar{\theta}_c$. In the ARMA-band-CHS configuration, band power x_p was estimated from $\bar{\theta}_c$. A point of interest was that the RLS filter also acted as an efficient low-pass filter that produced relatively smooth feature signals (thus removing the need for post-hoc Kalman filtering, Figure D.31). The tuning of single smoothing parameter η could help find a good SNR trade-off.

Another interesting variant of the extractor was the ARMA-coeff-CHS, presented in Figure D.34. Figure D.35 shows typical feature signals obtained with this extractor in an action detection task. $\bar{\theta}_c$ was directly used as a feature vector instead of x_p, which automatically helped simplify the extraction process. However, the temporal dynamics and the relationship strength to the decoded states (Figure 36) of the ARMA coefficients were more complicated to interpret. It was noted that the coefficient signals were mostly irregular, and the peak deviations from the baseline were modest (barely exceeding 2 standard deviations in the best case). The power operator could have acted as a form of dimensionality reduction and amplified part of the coefficient signals together. The signal decomposition was still dependent on the settings (N_{AR}, N_{MA}), which were later tuned.

F. Hjorth Extractor

Figure D.31 presents the layout for the HJ-CHS extractor, and Figure D.32 typical feature signals obtained with this extractor in an action detection task. The Hjorth parameters provided comparable yet different signal dynamics to what has been observed thus far. The deviations of the feature signals from baseline activity were relatively low, even in what was observed as strong responses. There was a visible coupling between activity and complexity, whilst the mobility mirrored these signals. The deviation

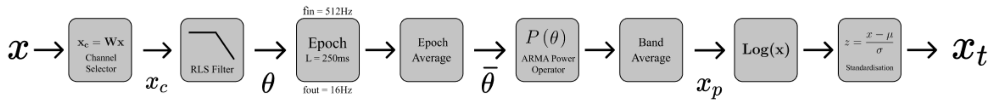


Figure D.31 - ARMA-band-CHS extractor setup; Relevant channels are first selected from x (N_c out of N_{ch} channels are left into x_c); ARMA coefficients θ are estimated using an RLS filter, epoched (using n_o overlaps) and averaged; The signal spectra is estimated and partitioned into different frequency bands from x_c (a total of $N_c N_{bd}$ band signals are extracted into x_{bd}); Log-transform and standardisation normalises the distributions in x_p; Fitting the extractor requires finding a mask array W to select channels, μ, σ for standardisation; Tuning requires setting the parameters N_c, n_o, η which controls the smoothing of the ARMA estimation (and λ in the case of a Box-Cox transformer).

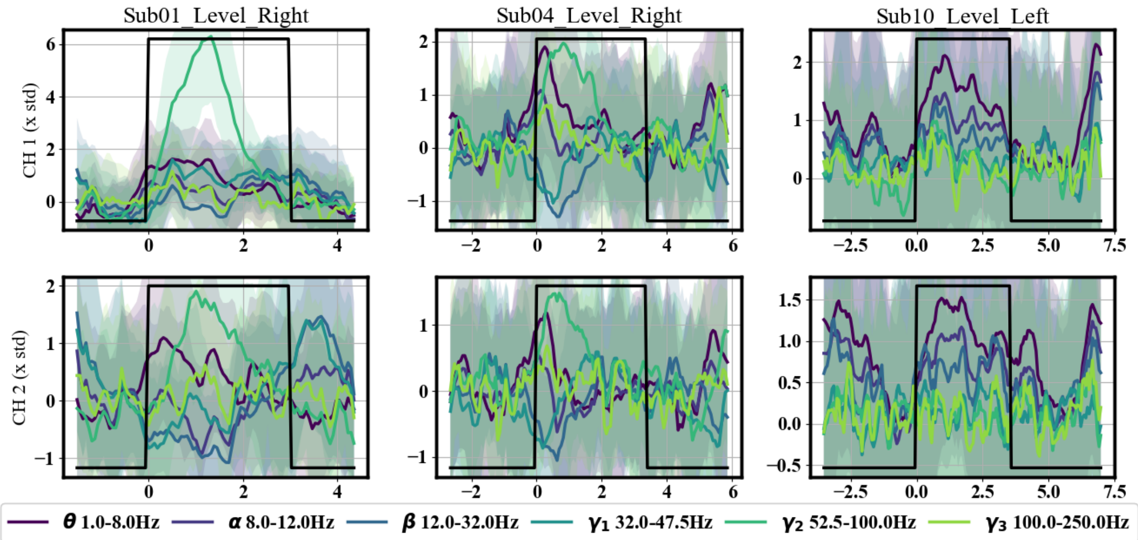


Figure D.32 - Example trial average response for the ARMA-band-CHS extractor; 2 channels were selected ($N_c = 2$); an epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$) was used; 6 AR coefficients, 2 MA coefficients and rate η set to 0.01 was used; The extraction was completed for 3 representative subjects of the 'Single Hand Grip Recordings' (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) lines indicate the action to detect.

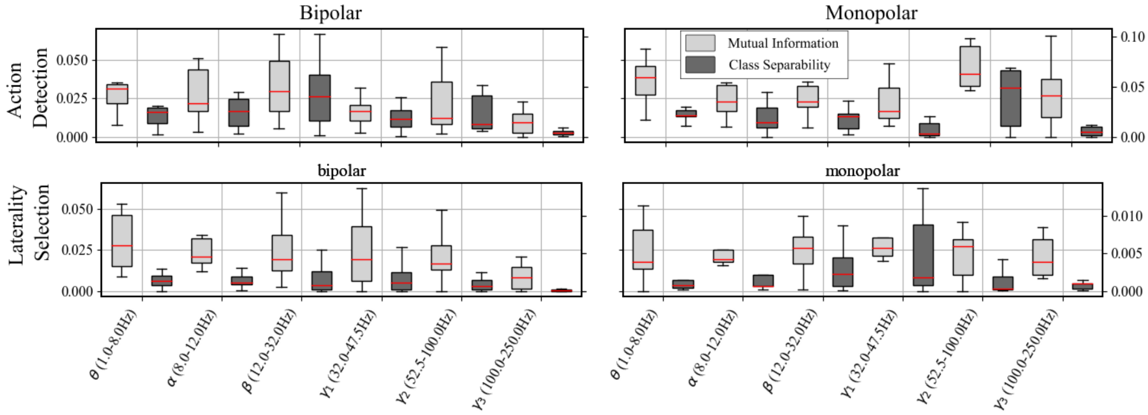


Figure D.33 - Pre-screening metric distributions (mutual-information and class-separability) measured with the ARMA-band-CHS extractor; metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A); Measures were taken both in bipolar monopolar configuration; The maximum metric value was chosen between the two selected channels.

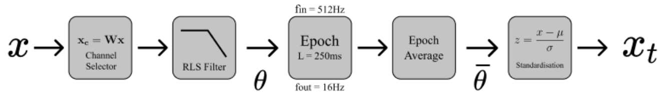


Figure D.34 - ARMA-coeff-CHS extractor setup; Relevant channels are first selected from x (N_c out of N_{ch} channels are left into x_c); ARMA coefficients θ are estimated using an RLS filter, Epoched (using n_o overlaps) and averaged; Standardisation centres the distributions in $\bar{\theta}$; Fitting the extractor requires finding a mask array W to select channels, μ, σ for standardisation; Tuning requires setting the parameters N_c, n_o and η which controls the smoothing of the ARMA.

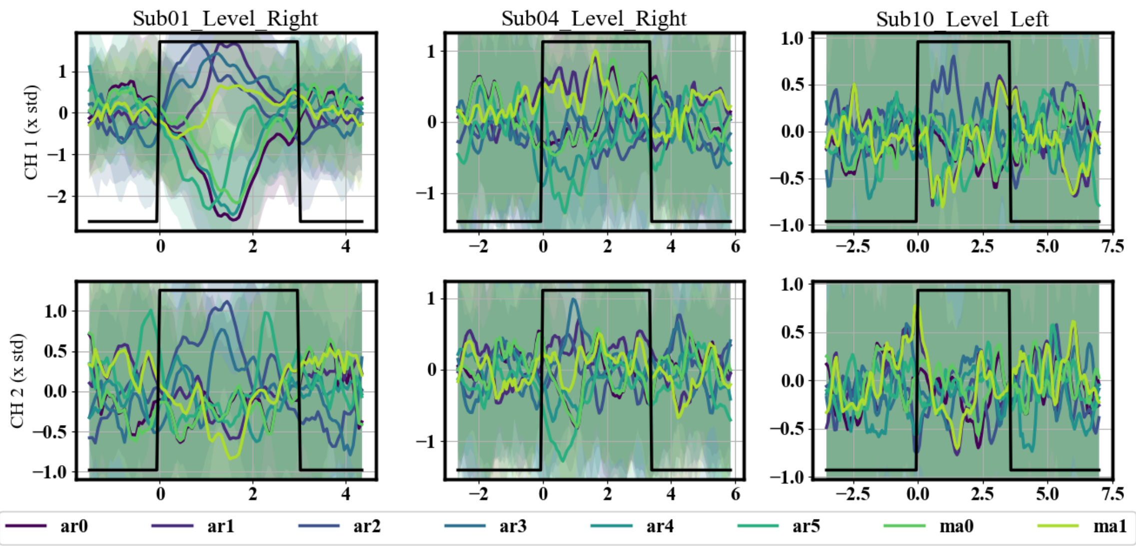


Figure D.35 - Example trial average response for the ARMA-coeff-CHS extractor; 2 channels were selected ($N_c = 2$); an epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$) was used; 6 AR coefficients, 2 MA coefficients and rate η set to 0.01 was used; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' dataset (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) lines indicates the action to detect.

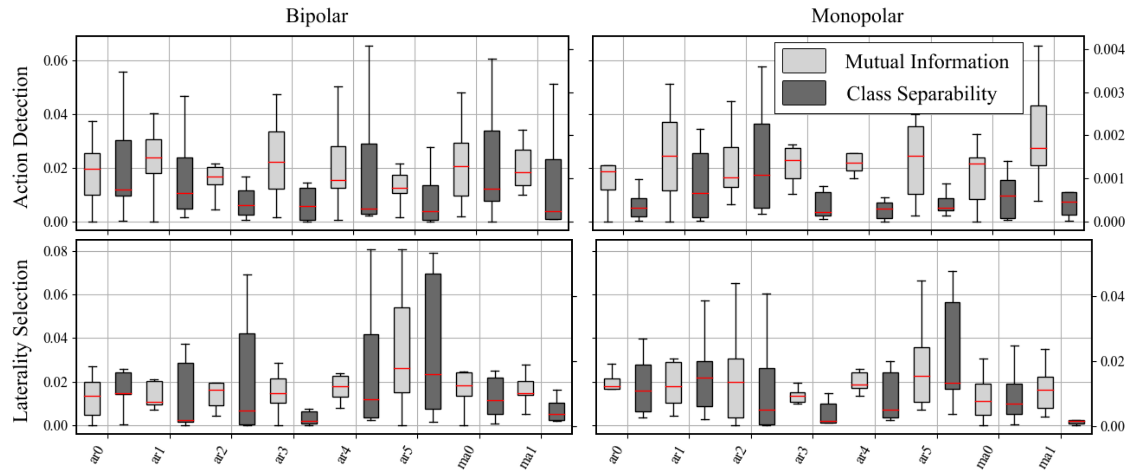


Figure D.36 - Pre-screening metric distributions (mutual-information and class-separability) measured with the ARMA-coeff-CHS extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip' dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A); Measures were taken both in bipolar monopolar configuration; The maximum metric value was chosen between the two selected channels.

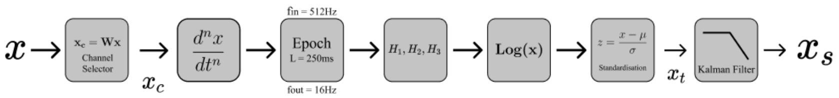


Figure D.37 - HJ-CHS extractor setup; Relevant channels are first selected from x (N_c out of N_{ch} channels are left into x_c); First and second derivatives are estimated for every channel in x_c; The signals are then epoched (using n_o overlaps) and Hjorth parameters (activity complexity and mobility) are computed; Log-transform and standardisation normalises the distributions in x_p; A random-walk first order Kalman filter is applied the transformed power x_t for further signal smoothing yielding the extractor output x_s; Fitting the extractor requires finding a mask array W to select channels, μ, σ for standardisation and σ_w for the Kalman filter; Tuning requires setting the parameters N_c, n_o (and λ in the case of a Box-Cox transformer).

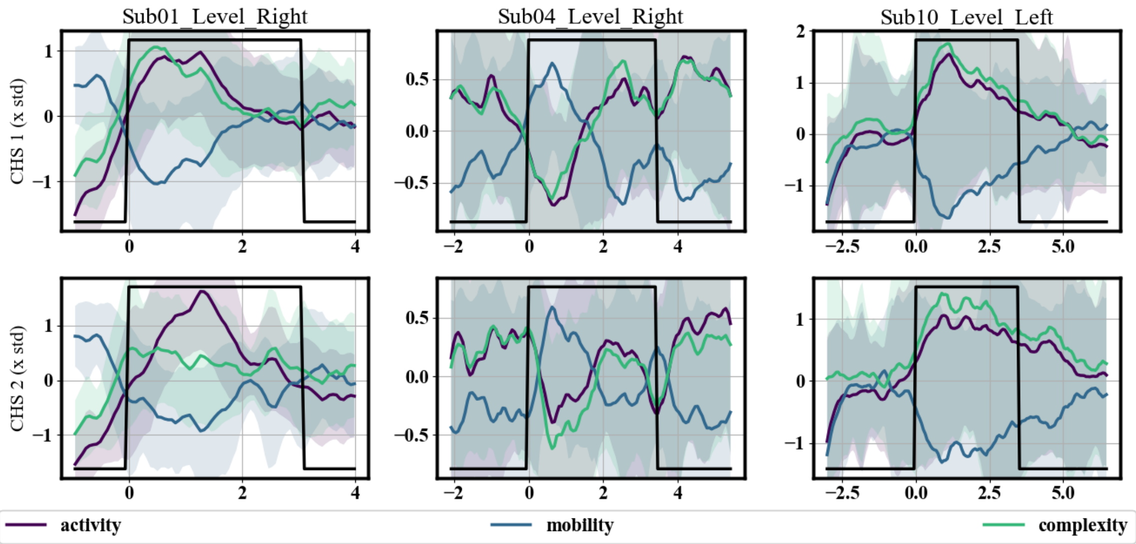


Figure D.38 - Example trial average response for the HJ-CHS extractor; 2 channels were selected ($N_c = 2$); An epoch length of 250ms ($n_o = 3$, for $f_d = 16Hz$) was used; The extraction was completed for 3 representative subjects of the 'Single Hand Grip' dataset (Appendix A) for the action detection task; 26, 34 and 49 trials were respectfully averaged; Solid lines represent trial mean and shaded areas represent a standard deviation above and below the mean; Solid (**Black**) lines indicate the action to detect.

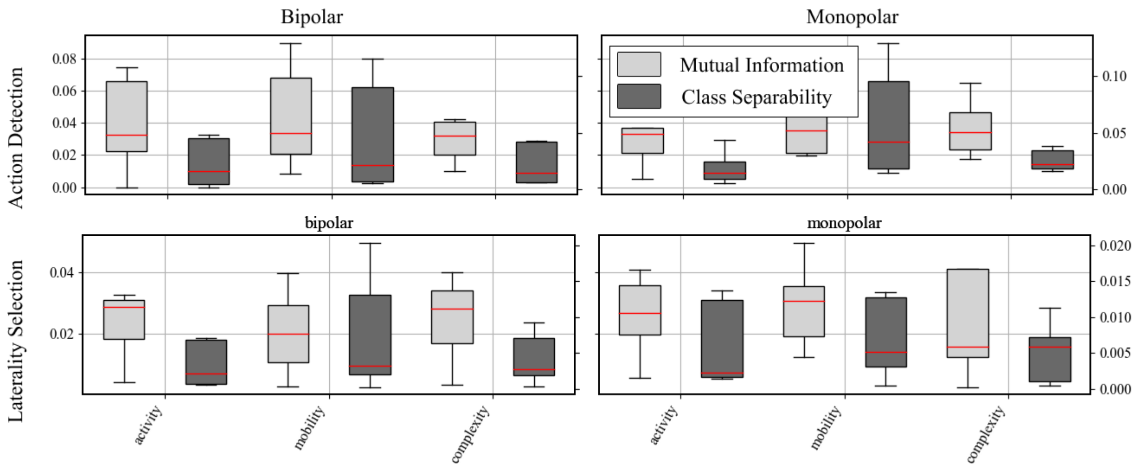


Figure D.39 - Pre-screening metric distributions (mutual-information and class-separability) measured with the HJ-CHS extractor; Metrics were extracted with the default settings (see above) for every feature and for every participant in the 'Single Hand Grip dataset for action detection and in the 'Dual Hand Pinch' dataset for laterality selection (Appendix A). Measures were taken both in bipolar monopolar configuration; The maximum metric value was chosen between the two selected channels.

direction of these the activity, complexity pair and mobility was conjectured to be case specific. In Sub01_Level_Right and Sub10_Level_Left, the mobility was suppressed, whilst, in Sub04_Level_Right, it was active. Activity and complexity in this instance followed more the trajectory of a β ERD rather than a γ or θ ERS (and vice versa). The ERD pattern in Sub04_Level_Right was noted to have a relatively large bandwidth in comparaison to the two other example recordings, ranging from α, β and even partly to γ_1. The corresponding θ peak was also particularly narrow. Variability in ERS/ERD features distribution across the signal spectra could be responsible for diametrically opposed Hjorth parameter signal temporal behaviours.